

# TOSHIBA

16-Bit  
Microcontroller  
TLCS-900  
MCU Series (1)

1 9 9 5

TMP96C141A F  
TMP96C041AF  
TMP96CM40F  
TMP96PM40F  
TMP96031N/F  
TMP96C081F



**NU HORIZONS ELECTRONICS CORP.**

<http://www.nuhorizons.com>

2070 RINGWOOD AVENUE  
SAN JOSE, CA 95131

(408) 434-0800  
FAX (408) 434-0935

D A T A B O O K

**Published in March, 1995**

Toshiba products described in this document are not authorized for use as critical components in life support systems without the written consent of the appropriate officer of Toshiba America, Inc. Life support systems are either systems intended for surgical implant in the body or systems which sustain life. A critical component is any component of a life support system whose failure to perform may cause a malfunction or failure of the life support system, or may affect its safety or effectiveness.

The information in this document has been carefully checked and is believed to be reliable. However, no responsibility can be assumed for inaccuracies that may not have been caught. All information in this document is subject to change without prior notice. Furthermore, Toshiba cannot assume responsibility for the use of any license under the patent rights of Toshiba or any third parties.

This technical data may be controlled under U.S. Export Administration Regulations and may be subject to the approval of the U.S. Department of Commerce prior to export. Any export or re-export, directly or indirectly, in contravention of the U.S. Export Administration Regulations is strictly prohibited.

Brand names and product names mentioned herein may be trademarks or registered trademarks of their respective companies.

---

## Preface

The TLCS 900 microcontroller family of products is described in the following documents:

### TLCS 900 Series CPU Architecture

The TLCS 900 user manual describes the 16-bit microcontroller TLCS 900 series system architecture. The scope of this document includes common topics to all derivatives of this family of products, such as CPU core operating modes, register organization, addressing modes, instruction set, timings and differences between the TLCS 90 and TLCS 900 series.

In addition, several databooks are available that cover the device specific topics such as package specifications and electrical characteristics. Depending on the TLCS 900 LSI device selected, the appropriate databook should be used in conjunction with the user manual. At the time of print, the following data books are available:

### TLCS 900 MCU Series (1)

describes:   TMP96C041AF  
              TMP96CM40F  
              TMP96PM40F  
              TMP96C031N/F  
              TMP96C081F

### TLCS 900/L MCU Series (2)

describes:   TMP93M40F  
              TMP93M40AF  
              TMP93M41F  
              TMP93M41AF  
              TMP96CS40F  
              TMP96PS40F

### TLCS 900/H MCU Series (3)

describes:   TMP95C061F





Chapter 1 - LSI Devices

TMP96C141AF

1. Outline and Device Characteristics .....	MCU900- 1
2. Pin Assignment and Functions .....	MCU900- 3
2.1 Pin Assignment .....	MCU900- 3
2.2 Pin Names and Functions .....	MCU900- 4
3. Operation .....	MCU900- 7
3.1 CPU .....	MCU900- 7
3.2 Memory Map .....	MCU900- 8
3.3 Interrupts .....	MCU900- 9
3.4 Standby Function .....	MCU900- 21
3.5 Functions of Ports .....	MCU900- 23
3.6 Chip Select/Wait Control .....	MCU900- 47
3.7 8-bit Timers .....	MCU900- 54
3.8 8-bit PWM Timer .....	MCU900- 70
3.9 16-bit Timers .....	MCU900- 84
3.10 Stepping Motor Control/Pattern Generation Port .....	MCU900-103
3.11 Serial Channel .....	MCU900-116
3.12 Analog/Digital Converter .....	MCU900-139
3.13 Watchdog Timer (Runaway Detecting Timer) .....	MCU900-144
4. Electrical Characteristics .....	MCU900-150
4.1 Absolute Maximum .....	MCU900-150
4.2 DC Characteristics .....	MCU900-151
4.3 AC Electrical Characteristics .....	MCU900-152
4.4 A/D Conversion Characteristics .....	MCU900-155
4.5 Serial Channel Timing - I/O Interface Mode .....	MCU900-155
4.6 Timer/Counter Input Clock .....	MCU900-155
4.7 Interrupt Operation .....	MCU900-156
4.8 Timing Chart for I/O Interface Mode .....	MCU900-157
4.9 Timing Chart for Bus Request/BUS Acknowledge .....	MCU900-158
4.10 Typical Characteristics .....	MCU900-159
5. Table of Special Function Registers (SFRs) .....	MCU900-160
6. Port Section Equivalent Circuit Diagram .....	MCU900-174
7. Guidelines and Restrictions .....	MCU900-178

**TMP96C041AF**

1. Outline and Device Characteristics .....	MCU900-179
2. Pin Assignment and Functions .....	MCU900-181
2.1 Pin Assignment .....	MCU900-181
2.2 Pin Names and Functions .....	MCU900-182
3. Operation .....	MCU900-185
3.1 CPU .....	MCU900-185
3.2 Memory Map .....	MCU900-185
3.3 Bus Release Function .....	MCU900-187
3.4 Serial Function .....	MCU900-188
4. Electrical Characteristics .....	MCU900-189
4.1 Absolute Maximum .....	MCU900-189
4.2 DC Characteristics .....	MCU900-190
4.3 AC Electrical Characteristics .....	MCU900-191
4.4 A/D Conversion Characteristics .....	MCU900-194
4.5 Serial Channel Timing - I/O Interface Mode .....	MCU900-194
4.6 Timer/Counter Input Clock (TI0, TI4, TI5, TI6, TI7) .....	MCU900-194
4.7 Interrupt Operation .....	MCU900-195
4.8 Timing Chart for I/O Interface Mode .....	MCU900-196
4.9 Timing Chart for Bus Request/BUS Acknowledge .....	MCU900-197
5. TMP96C141AF/TMP96C041AF Differences .....	MCU900-198

**TMP96CM40F**

1. Outline and Device Characteristics .....	MCU900-199
2. Pin Assignment and Functions .....	MCU900-201
2.1 Pin Assignment .....	MCU900-201
2.2 Pin Names and Functions .....	MCU900-202
3. Operation .....	MCU900-205
3.1 CPU .....	MCU900-205
3.2 Memory Map .....	MCU900-205
4. Electrical Characteristics .....	MCU900-207
4.1 Absolute Maximum .....	MCU900-207
4.2 DC Characteristics .....	MCU900-208
4.3 AC Electrical Characteristics .....	MCU900-209
4.4 A/D Conversion Characteristics .....	MCU900-212
4.5 Serial Channel Timing - I/O Interface Mode .....	MCU900-212
4.6 Timer/Counter Input Clock (TI0, TI4, TI5, TI6, TI7) .....	MCU900-212
4.7 Interrupt Operation .....	MCU900-213
4.8 Timing Chart for I/O Interface Mode .....	MCU900-214
4.9 Timing Chart for Bus Request/BUS Acknowledge .....	MCU900-215
4.10 Typical Characteristics .....	MCU900-216
5. TMP96C141AF/TMP96CM40F/TMP96PM40F Differences .....	MCU900-216

**TMP96PM40F**

1. Outline and Device Characteristics .....	MCU900-217
2. Pin Assignment and Functions .....	MCU900-219
2.1 Pin Assignment .....	MCU900-219
2.2 Pin Names and Functions .....	MCU900-220
3. Operation .....	MCU900-223
3.1 MCU Mode .....	MCU900-223
3.2 PROM Mode .....	MCU900-225
4. Electrical Characteristics .....	MCU900-227
4.1 Absolute Maximum .....	MCU900-227
4.2 DC Characteristics .....	MCU900-228
4.3 AC Electrical Characteristics .....	MCU900-229
4.4 A/D Conversion Characteristics .....	MCU900-232
4.5 Serial Channel Timing - I/O Interface Mode .....	MCU900-232
4.6 Timer/Counter Input Clock (TI0, TI4, TI5, TI6, TI7) .....	MCU900-232
4.7 Interrupt Operation .....	MCU900-233
4.8 Timing Chart for I/O Interface Mode .....	MCU900-234
4.9 Timing Chart for Bus Request/BUS Acknowledge .....	MCU900-235
4.10 Read Operation (PROM Mode) .....	MCU900-236
4.11 Programming Operation (PROM Mode) .....	MCU900-236
4.12 Read Operation Timing Chart (PROM Mode) .....	MCU900-237
4.13 Programming Operation Timing Chart (PROM Mode) .....	MCU900-237
4.14 Typical Characteristics .....	MCU900-238
5. TMP96C141AF/TMP96CM40F/TMP96PM40F Differences .....	MCU900-239

**TMP96C031N/TMP96C031F**

1. Outline and Device Characteristics .....	MCU900-241
2. Pin Assignment and Functions .....	MCU900-243
2.1 Pin Assignment .....	MCU900-243
2.2 Pin Names and Functions .....	MCU900-244
3. Operation .....	MCU900-246
3.1 CPU .....	MCU900-246
3.2 Memory Map .....	MCU900-247
3.3 Interrupts .....	MCU900-249
3.4 Standby Function .....	MCU900-261
3.5 Port Functions .....	MCU900-263
3.6 Chip Select/Wait Control .....	MCU900-282
3.7 8-bit Timers .....	MCU900-300
3.8 16-bit Timers .....	MCU900-320
3.9 Stepping Motor Control/Pattern Generation Port .....	MCU900-337

3.10	Serial Channel .....	MCU900-350
3.11	A/D Converter .....	MCU900-375
3.12	Watchdog Timer .....	MCU900-379
3.13	DRAM Controller .....	MCU900-385
4.	Electrical Characteristics .....	MCU900-396
4.1	Absolute Maximum .....	MCU900-396
4.2	DC Characteristics .....	MCU900-397
4.3	AC Electrical Characteristics .....	MCU900-398
4.4	DRAM Control A/C Characteristics .....	MCU900-401
4.5	A/D Conversion Characteristics .....	MCU900-403
4.6	Serial Channel Timing - I/O Interface Mode .....	MCU900-403
4.7	Timer/Counter Input Clock .....	MCU900-403
4.8	Interrupt Operation .....	MCU900-404
4.9	Timing Chart for I/O Interface Mode .....	MCU900-405
4.10	Timing Chart for Bus Request/BUS Acknowledge .....	MCU900-406
5.	Table of Special Function Registers (SFRs) .....	MCU900-407
6.	Port Section Equivalent Circuit Diagram .....	MCU900-422
7.	Guidelines and Restrictions .....	MCU900-426
<b>TMP96C081F</b>		
1.	Outline and Device Characteristics .....	MCU900-427
2.	Pin Assignment and Functions .....	MCU900-429
2.1	Pin Assignment .....	MCU900-429
2.2	Pin Names and Functions .....	MCU900-430
3.	Operation .....	MCU900-434
3.1	CPU .....	MCU900-434
3.2	Memory Map .....	MCU900-435
3.3	Interrupts .....	MCU900-436
3.4	Standby Function .....	MCU900-449
3.5	Functions of Ports .....	MCU900-442
3.6	Chip Select/Wait Controller .....	MCU900-481
3.7	8-bit Timers .....	MCU900-490
3.8	8-bit PWM Timer .....	MCU900-519
3.9	16-bit PWM Timers .....	MCU900-523
3.10	Stepping Motor Control/Pattern Generation Port .....	MCU900-543
3.11	Serial Channel .....	MCU900-556
3.12	Analog/Digital Converter .....	MCU900-579
3.13	Watchdog Timer .....	MCU900-584
3.14	Direct Memory Access Controller .....	MCU900-591
4.	Electrical Characteristics .....	MCU900-632

4.1 Absolute Maximum.....	MCU900-622
4.2 DC Characteristics .....	MCU900-623
4.3 AC Electrical Characteristics.....	MCU900-624
4.4 A/D Conversion Characteristics.....	MCU900-629
4.5 Serial Channel Timing - I/O Interface Mode .....	MCU900-629
4.6 Timer/Counter Input Clock .....	MCU900-629
4.7 Interrupt Operation .....	MCU900-630
4.8 Timing Chart for I/O Interface Mode .....	MCU900-631
4.9 Timing Chart for Bus Request/BUS Acknowledge .....	MCU900-632
4.10 DMAC .....	MCU900-633
5. Table of Special Function Registers (SFRs).....	MCU900-635
6. Port Section Equivalent Circuit Diagram .....	MCU900-653
7. Guidelines and Restrictions .....	MCU900-658
<b>Chapter 2</b>	
<b>TLCS-900 Application Circuit .....</b>	<b>APL900- 1</b>
<b>Chapter 3</b>	
<b>Package Dimensions .....</b>	<b>PKG900- 1</b>

Notes

## **Chapter 1 TLCS-900 LSI Devices**





### CMOS 16-bit Microcontroller

#### TMP96C141AF

#### 1. Outline and Device Characteristics

The TMP96C141AF is high-speed advanced 16-bit microcontroller developed for controlling medium to large-scale equipment.

The TMP96C141AF is housed in an 80-pin flat package. Device characteristics are as follows:

- (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 (1.6 $\mu$ s/2 bytes @ 20MHz)
  - (2) Minimum instruction execution time
    - 200ns @ 20MHz
- (3) Internal RAM: 1K byte
  - Internal ROM: None
  - (4) External memory expansion
    - Can be expanded up to 16M bytes (for both programs and data).
    - Can mix 8- and 16-bit external data buses.
      - …Dynamic data bus sizing
  - (5) 8-bit timers: 2 channels
  - (6) 8-bit PWM timers: 2 channels
  - (7) 16-bit timers: 2 channels
  - (8) Pattern generators: 4 bits, 2 channels
  - (9) Serial interface: 2 channels
  - (10) 10-bit A/D converter: 4 channels
  - (11) Watchdog timer
  - (12) Chip select/wait controller: 3 blocks
  - (13) Interrupt functions
    - 3 CPU interrupts: … SWI instruction, privileged violation, and illegal instruction
    - 14 internal interrupts  7-level priority can be set.
    - 6 external interrupts
  - (14) I/O ports
  - (15) Standby function : 3 halt modes (RUN, IDLE, STOP)

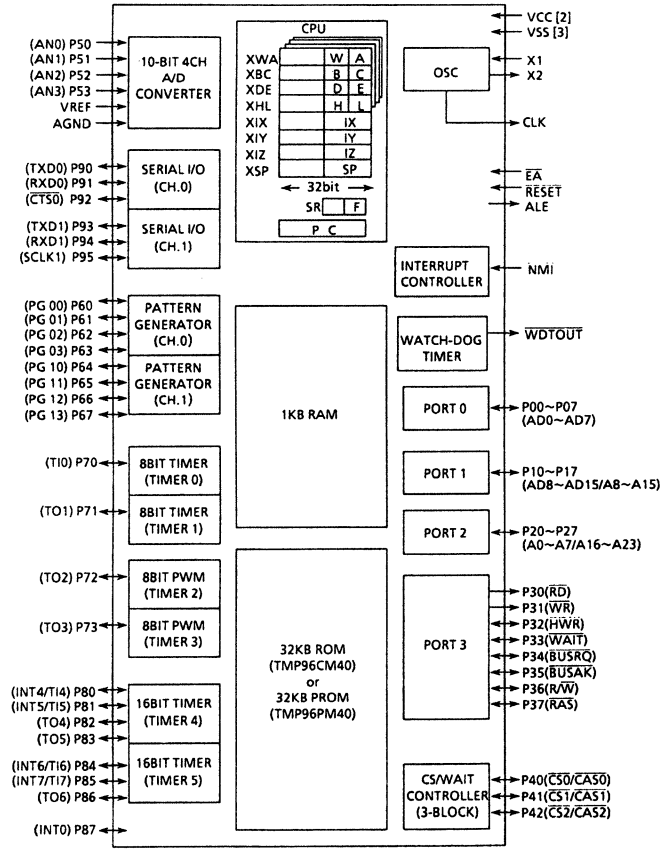


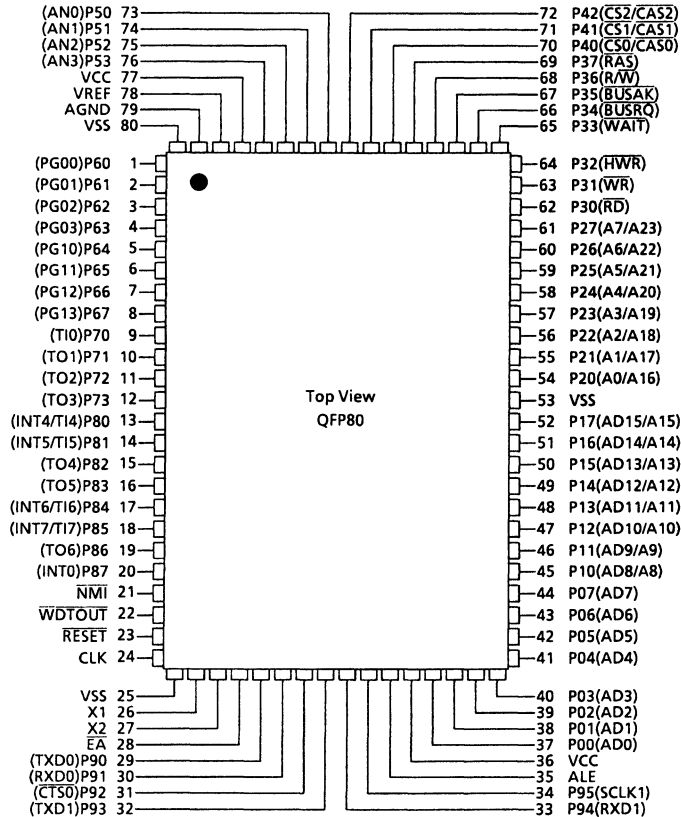
Figure 1. TMP96C141AF Block Diagram

**2. Pin Assignment and Functions**

The assignment of input/output pins for TMP96C141AF, their name and outline functions are described below.

**2.1 Pin Assignment**

Figure 2.1 shows pin assignment of TMP96C141AF.



Note : Because the TMP96C141AF has an external ROM, P00 to P17 pins are fixed to AD0 to AD15; P30 to RD; and P31 to WR.

**Figure 2.1 Pin Assignment (80-pin QFP)**

## 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
P00 ~ P07 AD0 ~ AD7	8	I/O Tri-state	Port 0: I/O port that allows I/O to be selected on a bit basis Address / data (lower): 0 - 7 for address / data bus
P10 ~ P17 AD8 ~ AD15 A8 ~ A15	8	I/O Tri-state Output	Port 1: I/O port that allows I/O to be selected on a bit basis 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 RD	1	Output Output	Port 30: Output port Read: Strobe signal for reading external memory
P31 WR	1	Output Output	Port 31: Output port Write: Strobe signal for writing data on pins AD0 -7
P32 HWR	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 - 15
P33 WAIT	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P34 BUSRQ	1	I/O Input	Port 34: I/O port (with pull-up resistor) Bus request: Signal used to request high impedance for AD0 - 15, A0 - 23, RD, WR, HWR, R/W, RAS, CS0, CS1, and CS2 pins. (For external DMAC)
P35 BUSAK	1	I/O Output	Port 35: I/O (with pull-up resistor) Bus acknowledge: Signal indicating that AD0 - 15, A0 - 23, RD, WR, HWR, R/W, RAS, CS0, CS1, and CS2 pins are at high impedance after receiving BUSRQ. (For external DMAC)
P36 R/W	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0, write cycle.
P37 RAS	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs RAS strobe for DRAM.
P40 CS0 CAS0	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs CAS strobe for DRAM when address is within specified address area.

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

Pin Name	Number of Pins	I/O	Functions
P41 CS1 CAS1	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs $\overline{\text{CAS}}$ strobe for DRAM if address is within specified address area.
P42 CS2 CAS2	1	I/O Output Output	Port 42: I/O port (with pull-up resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs $\overline{\text{CAS}}$ strobe for DRAM if address is within specified address area.
P50 ~ P53 AN0 ~ AN3	4	Input Input	Port 5: Input port Analog input: Input to A/D converter
VREF	1	Input	Pin for reference voltage input to A/D converter
AGND	1	Input	Ground pin for A/D converter
P60 ~ P63 PG00 ~ PG03	4	I/O Output	Ports 60 - 63: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 00 - 03
P64 ~ P67 PG10 ~ PG13	4	I/O Output	Ports 64 - 67: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 10 - 13
P70 T10	1	I/O Input	Port 70: I/O port (with pull-up resistor) Timer input 0: Timer 0 input
P71 T01	1	I/O Output	Port 71: I/O port (with pull-up resistor) Timer output 1: Timer 0 or 1 output
P72 T02	1	I/O Output	Port 72: I/O port (with pull-up resistor) PWM output 2: 8-bit PWM timer 2 output
P73 T03	1	I/O Output	Port 73: I/O port (with pull-up resistor) PWM output 3: 8-bit PWM timer 3 output
P80 T14 INT4	1	I/O Input Input	Port 80: I/O port (with pull-up resistor) Timer input 4: Timer 4 count/capture trigger signal input Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge
P81 T15 INT5	1	I/O Input Input	Port 81: I/O port (with pull-up resistor) Timer input 5: Timer 4 count/capture trigger signal input Interrupt request pin 5: Interrupt request pin with rising edge
P82 T04	1	I/O Output	Port 82: I/O port (with pull-up resistor) Timer output 4: Timer 4 output pin
P83 T05	1	I/O Output	Port 83: I/O port (with pull-up resistor) Timer output 5: Timer 4 output pin

Pin Name	Number of Pins	I/O	Functions
P84 TI6 INT6	1	I/O Input Input	Port 84: I/O port (with pull-up resistor) Timer input 6: Timer 5 count/capture trigger signal input Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
P85 TI7 INT7	1	I/O Input Input	Port 85: I/O port (with pull-up resistor) Timer input 7: Timer 5 count/capture trigger signal input Interrupt request pin 7: Interrupt request pin with rising edge
P86 TO6	1	I/O Output	Port 86: I/O port (with pull-up resistor) Timer output 6: Timer 5 output pin
P87 INT0	1	I/O Input	Port 87: I/O port (with pull-up resistor) Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P90 TXD0	1	I/O Output	Port 90: I/O port (with pull-up resistor) Serial send data 0
P91 RXD0	1	I/O Input	Port 91: I/O port (with pull-up resistor) Serial receive data 0
P92 CTS0	1	I/O Input	Port 92: I/O port (with pull-up resistor) Serial data send enable 0 (Clear to Send)
P93 TXD1	1	I/O Output	Port 93: I/O port (with pull-up resistor) Serial send data 1
P94 RXD1	1	I/O Input	Port 94: I/O port (with pull-up resistor) Serial receive data 1
P95 SCLK1	1	I/O I/O	Port 95: I/O port (with pull-up resistor) Serial clock I/O 1
WD $\overline$ TOUT	1	Output	Watchdog timer output pin
NMI	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output: Outputs $\lceil X1 + 4 \rceil$ clock. Pulled-up during reset.
$\overline$ EA	1	Input	External access: 0 should be inputted with TMP96C141AF 1, with TMP96CM40F/TMP96PM40F.
ALE	1	Output	Address latch enable
RESET	1	Input	Reset: Initializes LSI. (With pull-up resistor)
X1/X2	2	I/O	Oscillator connecting pin
VCC	2		Power supply pin (+ 5V)
VSS	3		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 TMP96C141AF device.

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

#### 3.1 CPU

The TMP96C141AF 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 TMP96C141AF that are not described in that manual.

##### 3.1.1 Reset

To reset the TMP96C141AF, the  $\overline{\text{RESET}}$  input must be kept at 0 for at least 10 system clocks (10 states: 1 $\mu$ s 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.)

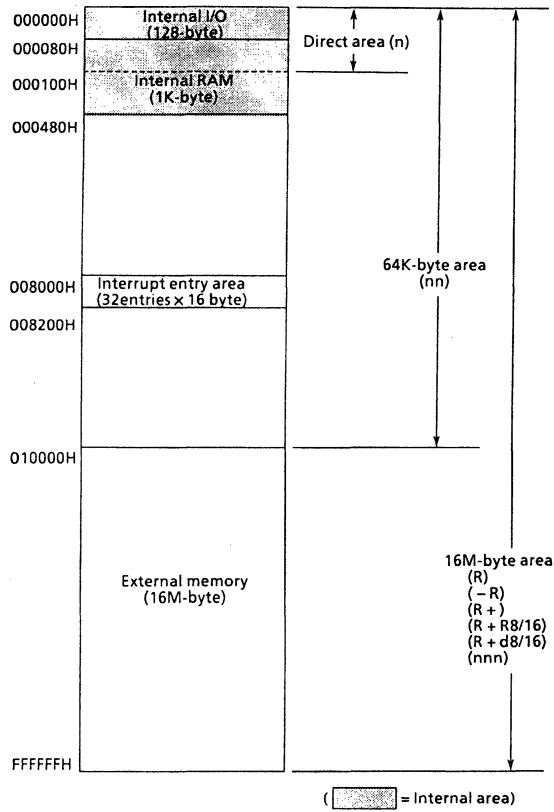
When reset is released, instruction execution starts from address 8000H. CPU internal registers other than the above are not changed.

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.

**3.2 Memory Map**

Figure 3.2 is a memory map of the TMP96C141AF.



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

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

The TMP96C141AF have altogether the following 23 interrupt sources:

- Interrupts from the CPU...3  
(Software interrupts, privileged violations, and Illegal (undefined) instruction execution)
- Interrupts from external pins (NMI, INTO, and INT4 to 7)...6
- Interrupts from built-in I/Os...14

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 (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

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

(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 micro DMA processing mode. High-speed micro DMA is a mode used by the CPU to automatically transfer byte or word data. It enables the CPU to process interrupts such as data saves to built-in I/Os at high speed.

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

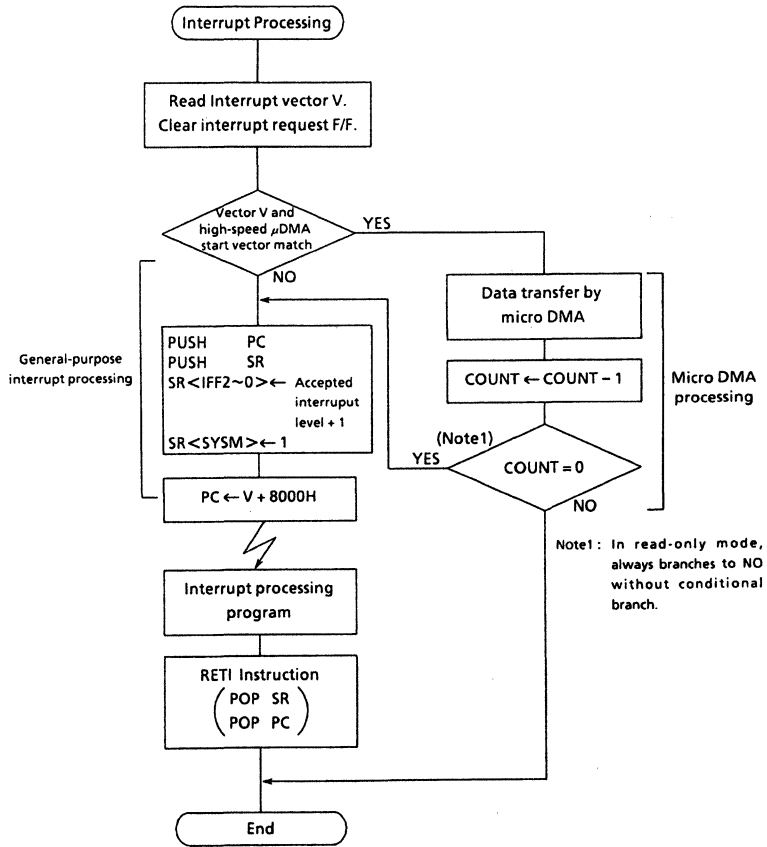


Figure 3.3 (1) Interrupt Processing Flowchart

### 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.

In minimum mode, all the above processing is completed in ██████████ (1.5μs @ 20MHz). In maximum mode, it is completed in 17 states.

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) TMP96C141AF Interrupt Table

Default Priority	Type	Interrupt Source	Vector Value "V"	Start Address	High-Speed Micro DMA Start Vector
1	Non-Maskable	Reset, or SW10 instruction	0 0 0 0 H	8 0 0 0 H	—
2		INTPREV: Privileged violation, or SW11	0 0 1 0 H	8 0 1 0 H	—
3		INTUNDEF: Illegal instruction, or SW12	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		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	INTO 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		INTTR6: 16-bit timer 5 (TREG6)	0 1 6 0 H	8 1 6 0 H	16H
23		INTTR7: 16-bit timer 5 (TREG7)	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
-		(Reserved)	0 1 D 0 H	8 1 D 0 H	1DH
-	(Reserved)	0 1 E 0 H	8 1 E 0 H	1EH	
-	(Reserved)	0 1 F 0 H	8 1 F 0 H	1FH	

### 3.3.2 High-Speed Micro DMA

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

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

**(1) High-Speed Micro DMA Operation**

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

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

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 micro DMA. 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 micro DMA processing.

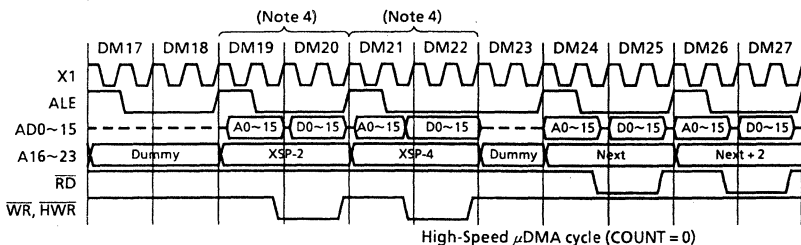
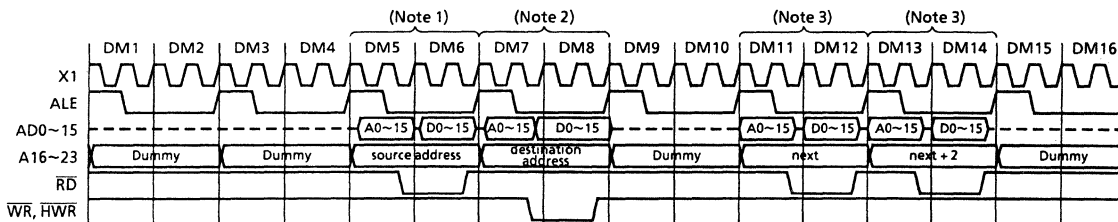
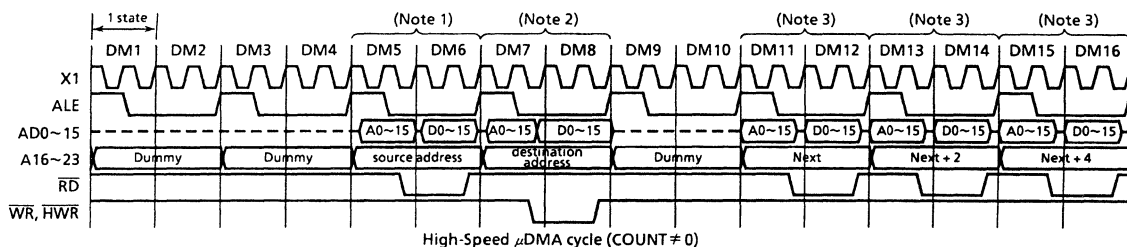
There are two data transfer modes: one-byte mode and one-word mode. Incrementing, decrementing, and fixing the transfer source/destination address after transfer can be done in both modes. Therefore data can easily be transferred between I/O and memory and between I/Os. For details of transfer modes,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

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

A the data transferred by the  $\mu$ DMA function, the transfer nter was decreased.

When this counter is "0"H, the processor operates general interrupt processing. At this time if the same channel of interrupt is required next interrupt, the transfer counter starts from 65536.

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

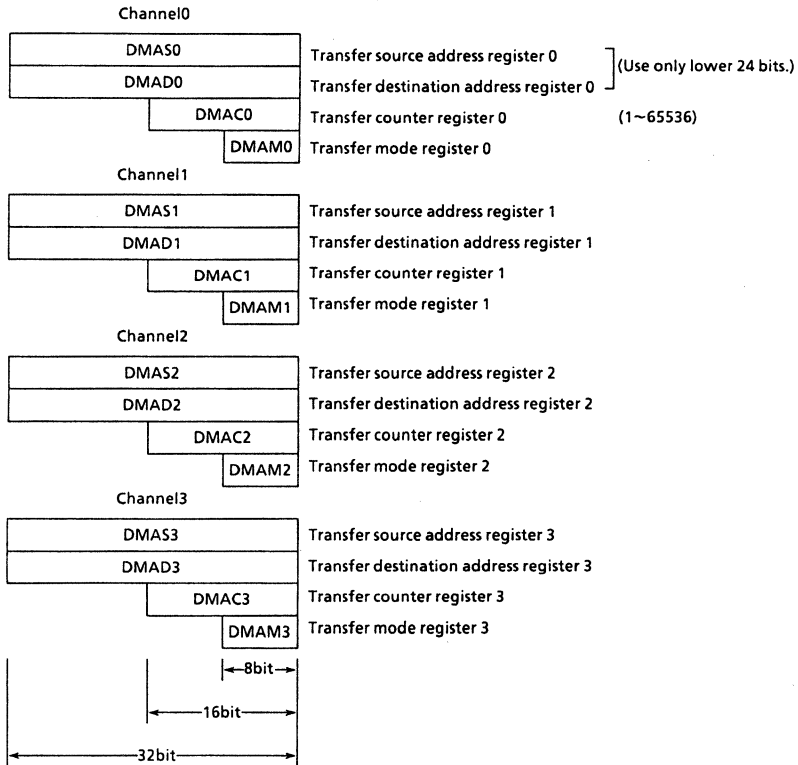


(Note1) This is added 2 states the case of the bus width of source address area is 8bit  
 (Note2) This is added 2 states the case of the bus width of destination address area is 8bit

(Note3) This may be a dummy cycle with instruction queue buffer.  
 (Note4) This is added 2 states the case of the bus width of stack address area is 8bit

The following timing chart is a high-speed  $\mu$ DMA cycle of the Transfer Address Increment mode (the other mode except the Read-only mode is same as this) (Condition: MIN mode, 16bit Bus width for 16M Byte, 0 wait)

**(2) Register Configuration (CPU Control Register)**



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

(3) Transfer Mode Register Details

(DMAM0~3)

0	0	0	0	Mode	Note: When specifying values for this register, set the upper 4 bits to 0.
↓					
↓					execution time (Min.) @20MHz
Z: 0 = byte transfer, 1 = word transfer					

0	0	0	Z	Transfer destination address INC mode ..... for I/O to memory (DMADn +) ← (DMASn) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	0	1	Z	Transfer destination address DEC mode ..... for I/O to memory (DMADn -) ← (DMASn) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	0	Z	Transfer source address INC mode ..... for I/O to memory (DMADn) ← (DMASn +) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	1	Z	Transfer source address DEC mode ..... for I/O to memory (DMADn) ← (DMASn -) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	0	Z	Fixed address mode ..... I/O to I/O (DMADn) ← (DMASn) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	1	0	Read-only mode ..... for DRAM refresh Dummy ← (DMASn) ; Reads 4 bytes. DMASn ← DMASn + 4 ; Increments lower word only. DMACn ← DMACn - 1	14 states (1.4 μs)
1	0	1	1	Counter mode ..... for interrupt counter DMASn ← DMASn + 1 DMACn ← DMACn - 1 if DMACn = 0 then INT.	11 states (1.1 μs)

(1 state = 100ns)

This condition is 16-bit bus width and 0 wait of source/destination address space.

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 accessed by high-speed μDMA. Do not use undefined codes for transfer mode control.

## &lt;Usage of read only mode (DRAM refresh)&gt;

When the hardware configuration is as follows:

DRAM mapping size:	= 1MB
DRAM data bus size:	= 8 bits
DRAM mapping address range:	= 100000H to 1FFFFFFH

Set the following registers first; refresh is performed automatically.

## ① Register initial value setting

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

## ② Timer Setting

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

## ③ Interrupt controller setting

Set the timer interrupt mask h other interrupt mask. Write the above timer interrupt vector value in the High-Speed μDMA start vector register, DMA0V.

(Operation description)

The DRAM data bus is an 8-bit bus and the micro DMA is in read-only mode (4 bytes), so refresh is performed four times per interrupt.

When a 512 refresh/8ms DRAM is connected, DRAM refresh is performed sufficiently if the micro DMA is started every  $15.625\mu\text{s} \times 4 = 62.4\mu\text{s}$  or less, since the timing is 15.625μs/refresh.

(Overhead)

Each processing time by the micro DMA is 1.8μs (18 states) @ 20MHz with an 8-bit data bus.

In the above example, the micro DMA is started every 62.5μs,  $1.8\mu\text{s}/62.5\mu\text{s} = 0.029$ ; thus, the overhead is 2.9%.

### 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 prior-

ity 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 INTO interrupt request, set the register after the XXXXXXXXXX as follows.

INTE0AD ← ---- 0 --- Zero-clears the INTO 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 (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 micro other DMA 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 micro DMA processing (see Table 3.3 (1)), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter registers (e.g., DMAS and DMAD) prior to the micro DMA processing.



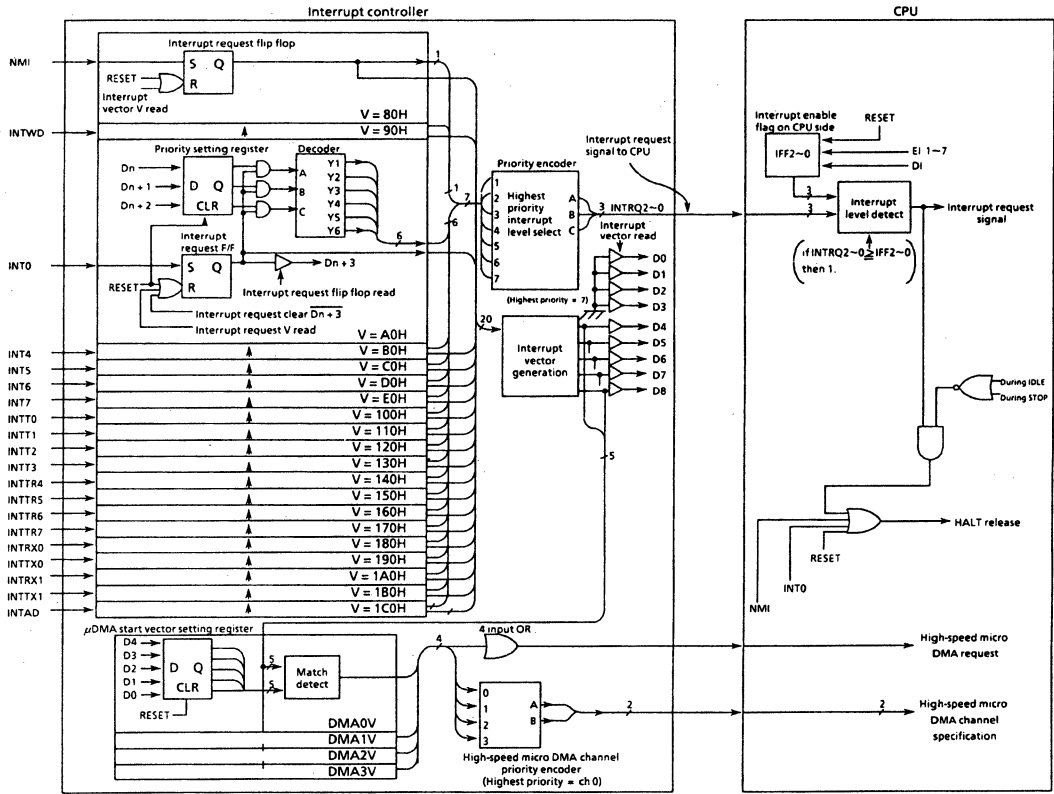


Figure 3.3.3 (1) Block Diagram of Interrupt Controller

(1) Interrupt Priority Setting Register

(Read-modify-write prohibited.)

Symbol	Address	7	6	5	4	3	2	1	0
INTE0AD	0070H	INTAD				INT0			
		IADC	IADM2	IADM1	IADM0	IOC	IOM2	IOM1	IOM0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
INTE45	0071H	INT5				INT4			
		ISC	ISM2	ISM1	ISM0	I4C	I4M2	I4M1	I4M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
INTE67	0072H	INT7				INT6			
		I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
INTE10	0073H	INTT1 (Timer1)				INTT0 (Timer0)			
		IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
INTEPW10	0074H	INTT3 (Timer3/PWM1)				INTT2 (Timer2/PWM0)			
		IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
INTE54	0075H	INTTR5 (TREG5)				INTTR4 (TREG4)			
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
INTE76	0076H	INTTR7 (TREG7)				INTTR6 (TREG6)			
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
		R/W		W		R/W		W	
		0	0	0	0	0	0	0	0
INTE50	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
INTE51	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

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

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

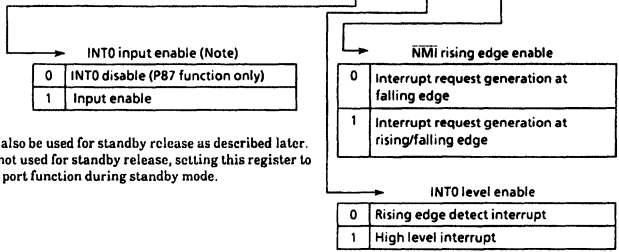
IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request.	Clears interrupt request flag.
1	Indicates interrupt request.	----- Don't care -----

(2) External Interrupt Control

Interrupt Input Mode Control Register

IIMC (007BH)	bit Symbol	7	6	5	4	3	2	1	0
	Read/Write						IOIE W	IOLE W	NMIREE W
	After reset						0	0	0
	Function						1: INTO input enable	0: INTO edge mode 1: INTO level mode	1: Can be operated in NMI rising edge.

Read-modify-write is prohibited.



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

Setting of External Interrupt Pin Functions

Interrupt	Pin name	Mode	Setting method
NMI	—	Falling edge	IIMC<NMIREE> = 0
		Rising and falling edges	IIMC<NMIREE> = 1
INT0	P87	Rising edge	IIMC<IOLE> = 0, <IOIE> = 1
		Level	IIMC<IOLE> = 1, <IOIE> = 1
INT4	P80	Rising edge	T4MOC<CAP12M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T4MOD<CAP12M1,0> = 1,0
INT5	P81	Rising edge	—
INT6	P84	Rising edge	T5MOC<CAP34M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T5MOD<CAP34M1,0> = 1,0
INT7	P85	Rising edge	—

**(3) High-Speed Micro DMA Start Vector**

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the interrupt vector with each channel's micro DMA start vector (bits 4 to 8 of the interrupt vec-

tor). When both match, the interrupt is processed in micro DMA 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.

**Micro DMA0 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA0V (007CH)	/			DMA0V8	DMA0V7	DMA0V6	DMA0V5	DMA0V4
Read / Write	W							
After reset				0	0	0	0	0

**Micro DMA1 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA1V (007DH)	/			DMA1V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4
Read / Write	W							
After reset				0	0	0	0	0

**Micro DMA2 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA2V (007EH)	/			DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4
Read / Write	W							
After reset				0	0	0	0	0

**Micro DMA3 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA3V (007FH)	/			DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4
Read / Write	W							
After reset				0	0	0	0	0

**(4) Notes**

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

while reading the interrupt vector after accepting the interrupt. If so, the CPU would read the default vector 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 TMP96C141AF 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.

	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 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 for 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 NMI or INTO 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 INTO pin. Processing resumes from address next to HALT instruction.
IDLE	Can only be released by NMI or INTO pin. After standby mode is released, interrupt processing starts.	↑
STOP	↑ (Note)	↑

Table 3. 4 (1) Pin States in STOP Mode

Pin Name	I/O	96C141AF		96CM40/96PM40	
		DRVE = 0	DRVE = 1	DRVE = 0	DRVE = 1
P0	Input mode/AD0 - 7 Output mode	- x	- x	- -	- Output
P1	Input mode/AD8 - 15 Output mode/AB - 15	- x	- x	- -	- Output
P2	Input mode Output mode/A0 - 7, A16 - 23	PD* PD*	PD* Output	PD* PD*	PD* Output
P30 (RD), P31 (WR)	Output	-	"1" Output	-	Output
P32 - P37	Input mode Output mode	PU PU	PU Output		
P40, P41	Input mode Output mode	PU* PU*	PU Output		
P42 (CS2/CAS2)	Input mode Output mode	PD* PD*	PD Output		
P5	Input	-	-		
P6	Input mode Output mode	PU* PU*	PU Output		
P7	Input mode Output mode	PU* PU*	PU Output		
P80 - P86	Input mode Output mode	PU* PU*	PU Output		
P87 (INT0)	Input mode Output mode	PU PU	PU Output		
P9	Input mode Output mode	PU* PU*	PU Output		
NMI	Input	Input	Input		
WDTOU	Output	Output	Output		
ALE	Output	"0"	"0"		
CLK	Output	-	"1"		
RESET	Input	Input	Input		
EA	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 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.

x: Cannot set.

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 Functions of Ports**

The TMP96CM40F/TMP96PM40F has 65 bits for I/O ports. The TMP96C141AF, TMP96C041AF has 47 bits for I/O ports because Port0, Port1, P30, and P31 are dedicated pins for AD0 to 7, AD8 to

15,  $\overline{RD}$ , and  $\overline{WR}$ .

These port pins have I/O functions for the built-in CPU and internal I/Os as well as general-purpose I/O port functions. Table 3.5 lists the function of each port pin.

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

**Table 3.5 Functions of Ports**

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port0	P00 - P07	8	I/O	-	Bit	AD0 - AD7
Port1	P10 - P17	8	I/O		Bit	AD8 - AD15/ A8 - A15
Port2	P20 - P27	8	I/O	↓	Bit	A0 - A7/ A16 - A23
Port 3	P30	1	Output	-	(Fixed)	$\overline{RD}$
	P31	1	Output	-	(Fixed)	$\overline{WR}$
	P32	1	I/O	↑	Bit	$\overline{HWR}$
	P33	1	I/O	↑	Bit	$\overline{WAIT}$
	P34	1	I/O	↑	Bit	$\overline{BUSRQ}$
	P35	1	I/O	↑	Bit	$\overline{BUSAK}$
	P36	1	I/O	↑	Bit	$\overline{R/W}$
	P37	1	I/O	↑	Bit	$\overline{RAS}$
Port4	P40	1	I/O		Bit	$\overline{CS0}$ / $\overline{CAS0}$
	P41	1	I/O	↑	Bit	$\overline{CS1}$ / $\overline{CAS1}$
	P42	1	I/O	↓	Bit	$\overline{CS2}$ / $\overline{CAS2}$
Port5	P50 - P53	4	Input	-	(Fixed)	AN0 - AN3
Port6	P60 - P67	8	I/O	↑	Bit	PG00 - PG03, PG10 - PG13
Port7	P70	1	I/O	↑	Bit	T10
	P71	1	I/O	↑	Bit	TO1
	P72	1	I/O	↑	Bit	TO2
	P73	1	I/O	↑	Bit	TO3
Port8	P80	1	I/O	↑	Bit	T14/INT4
	P81	1	I/O	↑	Bit	T15/INT5
	P82	1	I/O	↑	Bit	TO4
	P83	1	I/O	↑	Bit	TO5
	P84	1	I/O	↑	Bit	T16 / INT6
	P85	1	I/O	↑	Bit	T17 / INT7
	P86	1	I/O	↑	Bit	TO6
	P87	1	I/O	↑	Bit	INT0
Port9	P90	1	I/O	↑	Bit	TXD0
	P91	1	I/O	↑	Bit	RXD0
	P92	1	I/O	↑	Bit	$\overline{CTS0}$
	P93	1	I/O	↑	Bit	TXD1
	P94	1	I/O	↑	Bit	RXD1
	P95	1	I/O	↑	Bit	SCLK1

Resetting makes the port pins listed below function as general-purpose I/O ports.

I/O pins programmable for input or output function as input ports.

To set port pins for built-in functions, a program is required.

Since the TMP96C141AF has an external ROM, some ports are permanently assigned to the CPU.

- P00 ~ P07 → AD0 ~ AD7
- P10 ~ P17 → AD8 ~ AD15
- P30 →  $\overline{RD}$
- P31 →  $\overline{WR}$

Bus release function

The TMP96C141AF has the internal pull-up and pull-down resistors to fix the bus control signals at bus release.

Table 3.5 (1) shows the pin condition at bus release (BUSAK = "L").

Pin Name	Pin state at bus release	
	Port mode	Function mode
P00 - P07 (AD0 - AD7) P10 - P17 (AD8 - AD15)	No status change (these pins are not "Hz")	These pins are "Hz".
P30 ( $\overline{RD}$ ) P31 ( $\overline{WR}$ )	↑	These pins are "Hz". ("Hz" status after these pins are driven to high level.)
P32 ( $\overline{HWR}$ ) P37 ( $\overline{RAS}$ )	↑	The output buffer is "OFF" after these pins are driven high. These pins are added in the internal resistor of pull-up. It's no relation for the value of output latch.
P36 ( $\overline{R/W}$ ) P40 ( $\overline{CS0/CAS0}$ ) P41 ( $\overline{CS1/CAS1}$ )	↑	↑
P42 ( $\overline{CS2/CAS2}$ )	↑	(*) ↑
P20 - P27 (A16 - A23) P42 ( $\overline{CS2/CAS2}$ )	↑	The output buffer is "OFF" after these pins are driven high. These pins are added in the internal resistor of pull-down. It's no relation for the value of output latch.

(\*) P42 has the resistor of programmable pull-down, but when the bus are released, P42 pin is added a resistor of pull-up.

That is, when it is used for bus release ( $\overline{BUSAK} = "0"$ ), the pins of below need pull-up or pull-down resistor for an external circuit.

P00 - P07 (AD07)

P10 - P17 (AD8 - AD15)

P30 (RD)

P31 (WR)

When the bus is released, both internal memory and internal I/O cannot be accessed. But the internal I/O continues to run. Therefore, be careful about releasing time and set the selection time WDT.

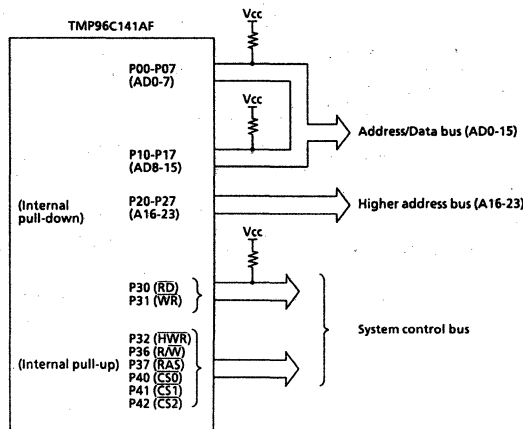


Figure 3.5. Example of external bus interface using bus release function.



**3.5.1 Port 0 (P00 - P07)**

Port 0 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using control register P0CR to 0 and sets Port 0 to input mode.

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

With the TMP96C141AF/TMP96C041AF, which comes with an external ROM, Port 0 always functions as an address data bus (AD0 to 7) regardless of the value set in control register P0CR.

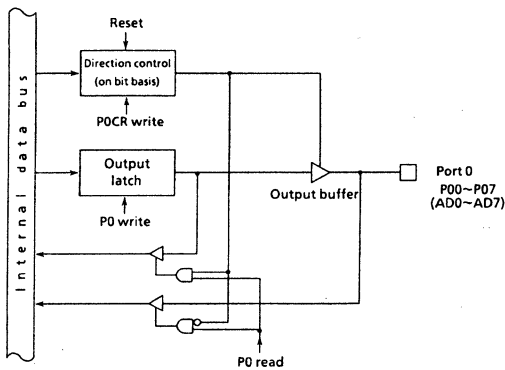


Figure 3.5 (1). Port 0

**3.5.2 Port 1 (P10 - P17)**

Port 1 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using control register P1CR and function register P1FC. Resetting resets all bits of output latch P1, control register P1CR, and function register P1FC to 0 and sets Port 1 to input mode.

In addition to functioning as a general purpose I/O port, Port 1 also functions as an address data bus (AD8 to 15) or an address bus (A8 to 15).

With the TMP96C141AF/TMP96C041AF, which comes with an external ROM, Port 1 always functions as an address data bus (AD8 to 15) regardless of the value set in control register P1CR.

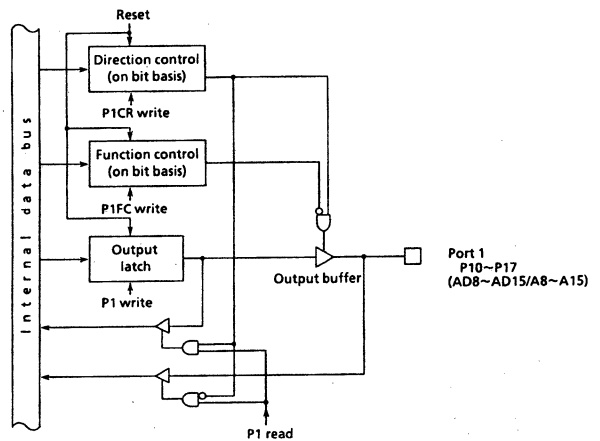


Figure 3.5 (2). Port 1

Port 0 Register

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

Port 0 Control Register

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

Port 0 I/O setting

0	Input
1	Output

Port 1 Register

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

Port 1 Control Register

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

Port 1 Function Register

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

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

Port 1 function setting

	P1FC <P1XF>	0	1
P1CR <P1XC>	0	Input port	Address data bus (AD15-8)
	1	Output port	Address bus (A15-8)

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

Figure 3.5 (3). Registers for Ports 0 and 1

**3.5.3 Port 2 (P20 - 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. To disconnect the pull-down resistor, write 1 in the output latch.

In addition to functioning as a general-purpose I/O port, Port 2 also functions as an address data bus (A0 to 7) and an address bus (A16 to 23).

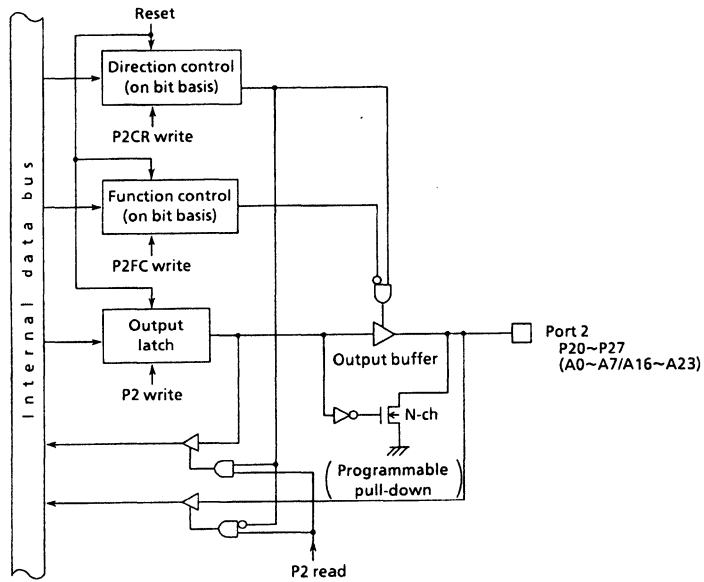


Figure 3.5 (4). Port 2

Port 2 Register

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

Port 2 Control Register

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

Port 2 Function Register

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

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

Port 2 function setting

	P2FC <P2XF>	
P2CR <P2XC>	0	1
0	Input	address bus (A7-0)
1	Output	address bus (A23-16)

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

Figure 3.5 (5). Registers for Port 2

### 3.5.4 Port 3 (P30 - P37)

Port 3 is an 8-bit general-purpose I/O port.

I/O can be set on a bit basis, but note that P30 and P31 are used for output only. I/O is set using control register P3CR and function register P3FC. Resetting resets all bits of output latch P3, control register P3CR (bits 0 and 1 are unused), and function register P3FC to 0. Resetting also outputs 1 from P30 and P31, sets P32 to P37 to input mode, and connects a pull-up resistor.

In addition to functioning as a general-purpose I/O port, Port 3 also functions as an I/O for the CPU's control/status signal.

With the TMP96C141AF, when P30 pin is defined as  $\overline{RD}$  signal output mode ( $\langle P30F \rangle = 1$ ), clearing the output latch register  $\langle P30 \rangle$  to 0 outputs the  $\overline{RD}$  strobe (used for the pseudo static RAM) from the P30 pin even when the internal address area is accessed.

If the output latch register  $\langle P30 \rangle$  remains 1, the  $\overline{RD}$  strobe signal is output only when the external address area is accessed.

With the TMP96C141AF/TMP96C041AF, which comes with an external ROM, Port 30 outputs the  $\overline{RD}$  signal; P31, the  $\overline{WR}$  signal, regardless of the values set in function registers P30F and P31F.

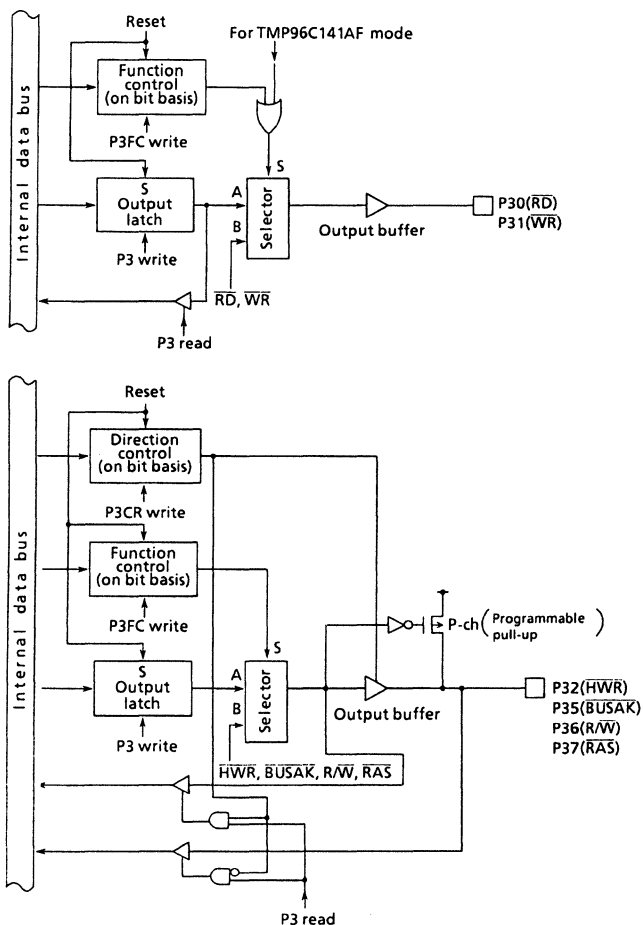


Figure 3.5 (6). Port 3 (P30, P31, P32, P35, P36, P37)

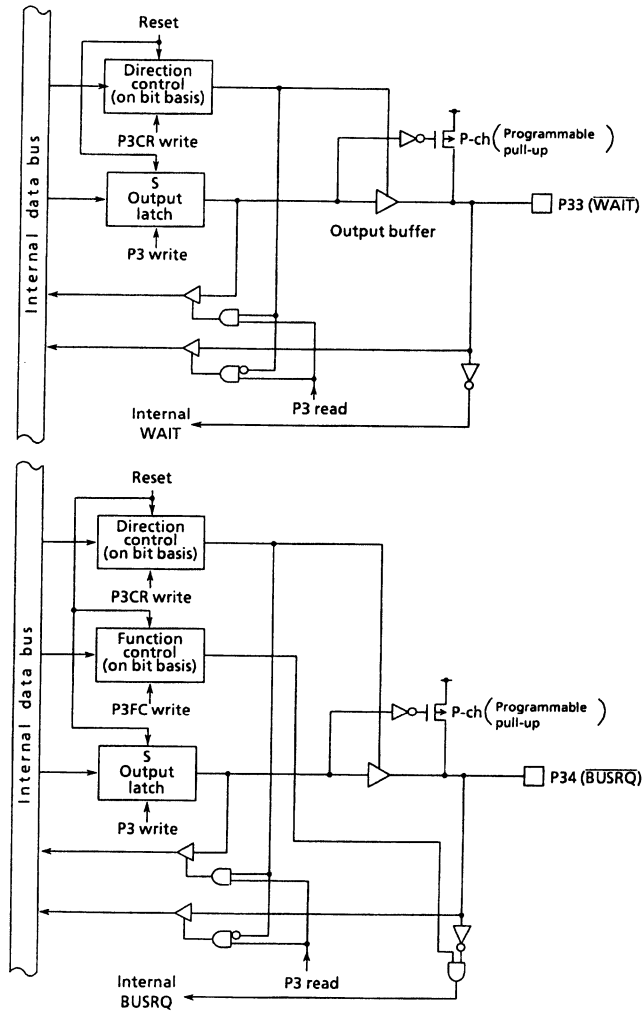
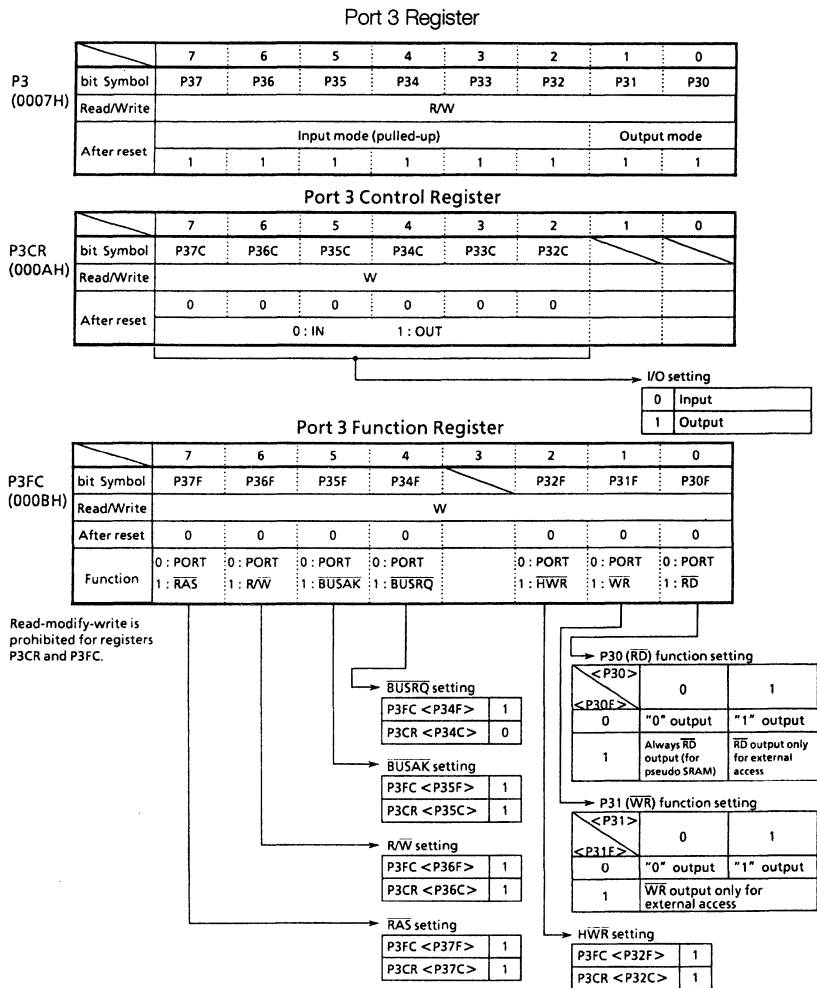


Figure 3.5 (7). Port 3 (P33, P34)



Note: When P33/WAIT pin is used as a WAIT pin, set P#CR <P33C> to "0" and Chip Select/Wait control register.

Figure 3.5 (8). Registers for Port 3



### 3.5.5 Port 4 (P40 - P42)

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

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

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

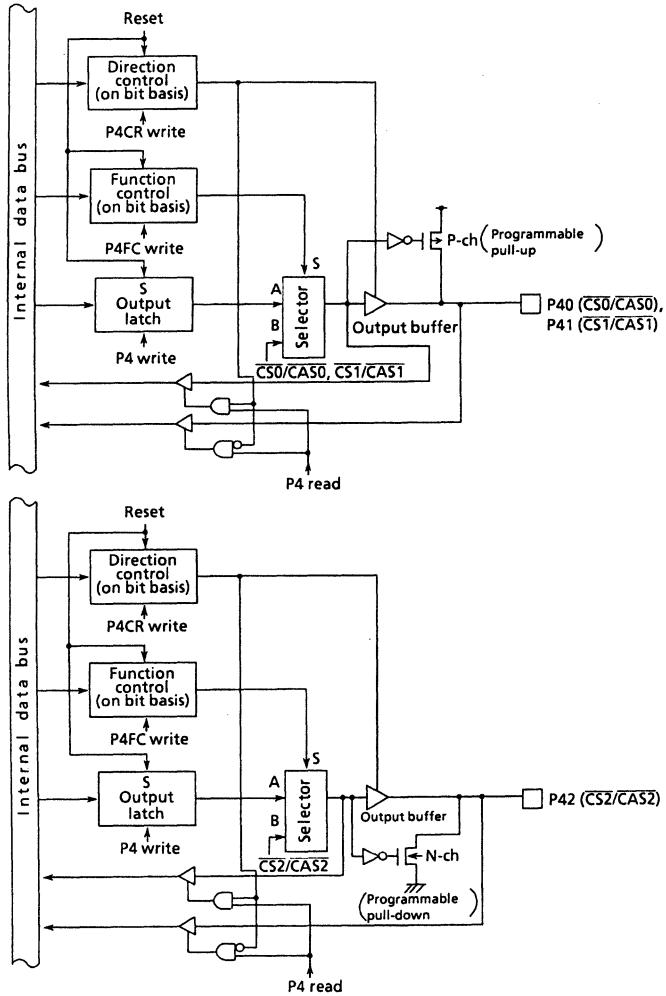
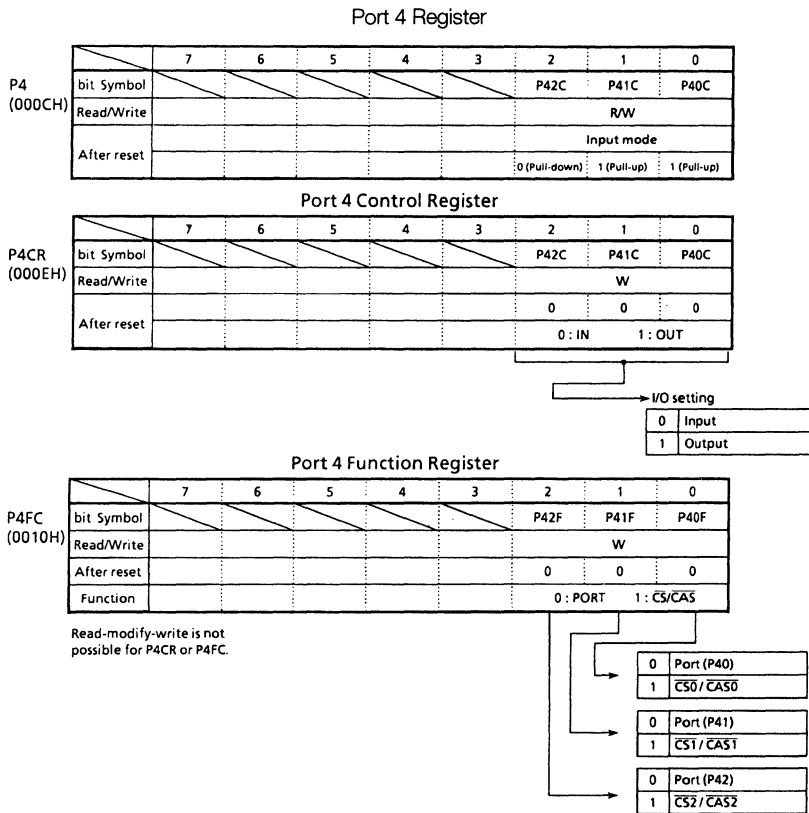


Figure 3.5 (9). Port 4

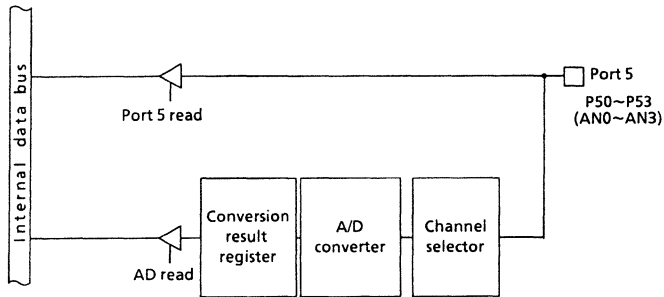


Note: To output chip select signal ( $\overline{CS0}/\overline{CAS0}$  to  $\overline{CS2}/\overline{CAS2}$ ), set the corresponding bits of the control register P4CR and the function register to P4FC. The BOCS, B1CS, and B2CS registers of the chip select/wait controller are used to select the  $\overline{CS}/\overline{CAS}$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

Figure 3.5 (10). Registers for Port 4

**3.5.6 Port 5 (P50 - P53)**

Port 5 is a 4-bit input port, also used as an analog input pin.



**Figure 3.5 (11). Port 5**

**Port 5 Register**

		7	6	5	4	3	2	1	0		
P5 (000DH)	bit Symbol	/			/			P53	P52	P51	P50
	Read/Write	R									
	After reset	Input mode									

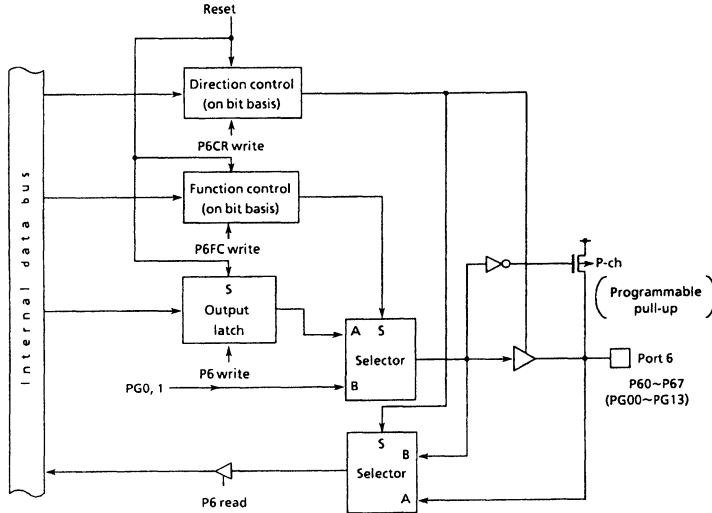
Note: The input channel selection of A/D Converter is set by A/D Converter mode register ADMOD2.

**Figure 3.5 (12). Registers for Port 5**

**3.5.7 Port 6 (P60 - P67)**

Port 6 is an 8-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 6 as an input port and connects a pull-up resistor. It also sets all bits of the output latch to 1. In addition to functioning as a general-purpose I/O port, Port 6 also functions as a pattern generator PG0/PG1 output. PG0 is

assigned to P60 to P63; PG1, to P64 to P67. Writing 1 in the corresponding bit of the port 6 function register (P6FC) enables PG output. Resetting resets the function register P6FC value to 0, and sets all bits to ports.



**Figure 3.5 (13). Port 6**

Port 6 Register

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

Port 6 Control Register

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

Port 6 I/O setting

0	Input
1	Output

Port 6 Function Register

	7	6	5	4	3	2	1	0
bit Symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : PORT		1 : PG1-OUT		0 : PORT		1 : PG0-OUT	

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

Port 6 function setting

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

Figure 3.5 (14). Registers for Port 6

**3.5.8 Port 7 (P70 - P73)**

Port 7 is a 4-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 7 as an input port and connects a pull-up resistor. In addition to functioning as a general-purpose I/O port, Port 70 also functions as an input clock pin T10; Port

71 as an 8-bit timer output (TO1), Port 72 as a PWM0 output (TO2), and Port 73 as a PWM1 output (TO3) pin. Writing 1 in the corresponding bit of the Port 7 function register (P7FC) enables output of the timer. Resetting resets the function register P7FC value to 0, and sets all bits to ports.

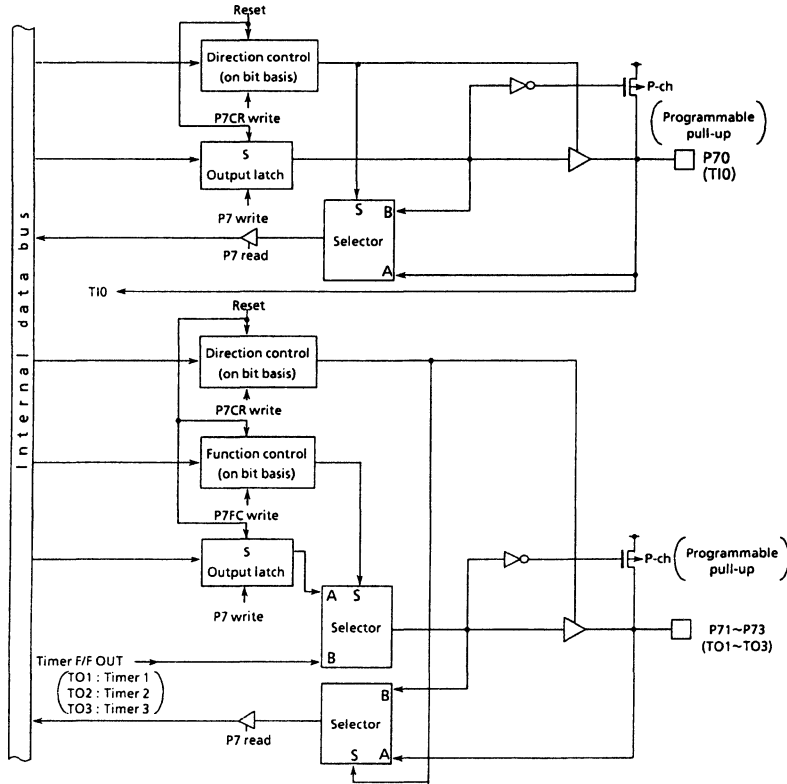
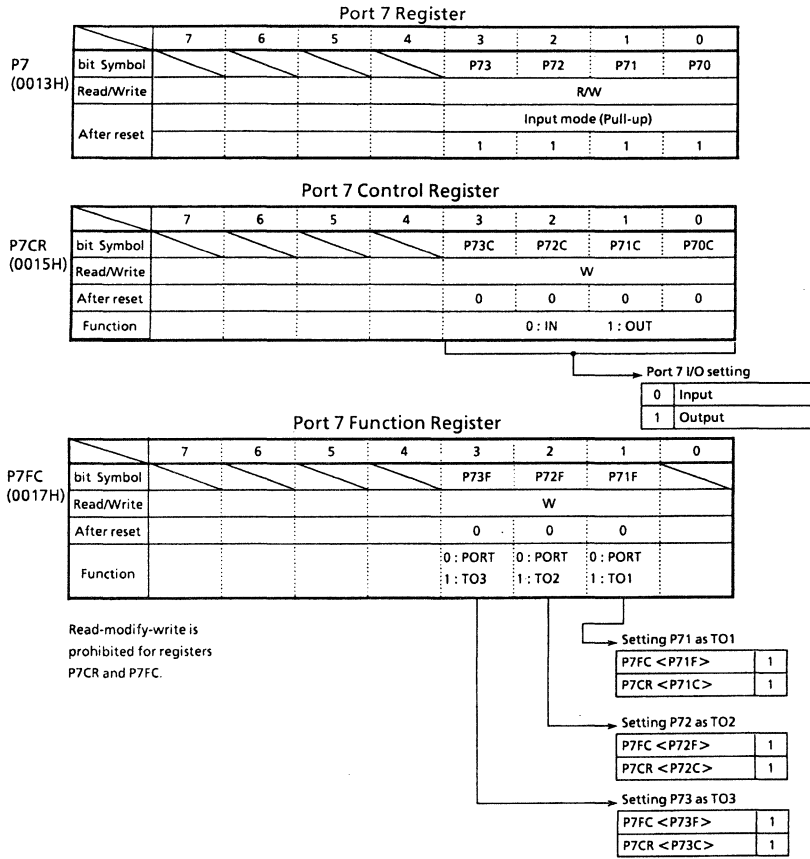


Figure 3.5 (15). Port 7



Note : P70/T10 pin does not have a register changing PORT/FUNCTION.  
 For example, when it is used as an input port (P70), the input signal for P70 is inputted to 8bit Timer 0 as a timer input 0 (T10).

**Figure 3.5 (16). Registers for Port 7**



**3.5.9 Port 8 (P80 - P83)**

Port 8 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis. Resetting sets Port 8 as an input port and connects a pull-up resistor. It also sets all bits of the output latch register P8 to 1. In addition to functioning as a general-purpose I/O port, Port 8 also functions as an input for 16-bit timer 4 and 5

clocks, an output for 16-bit timer F/F 4, 5 and 6 output, and an input for INTO. Writing 1 in the corresponding bit of the Port 8 function register (P8FC) enables those functions. Resetting resets the function register P8FC value to 0, and sets all bits to ports.

(1) P80 ~ P86

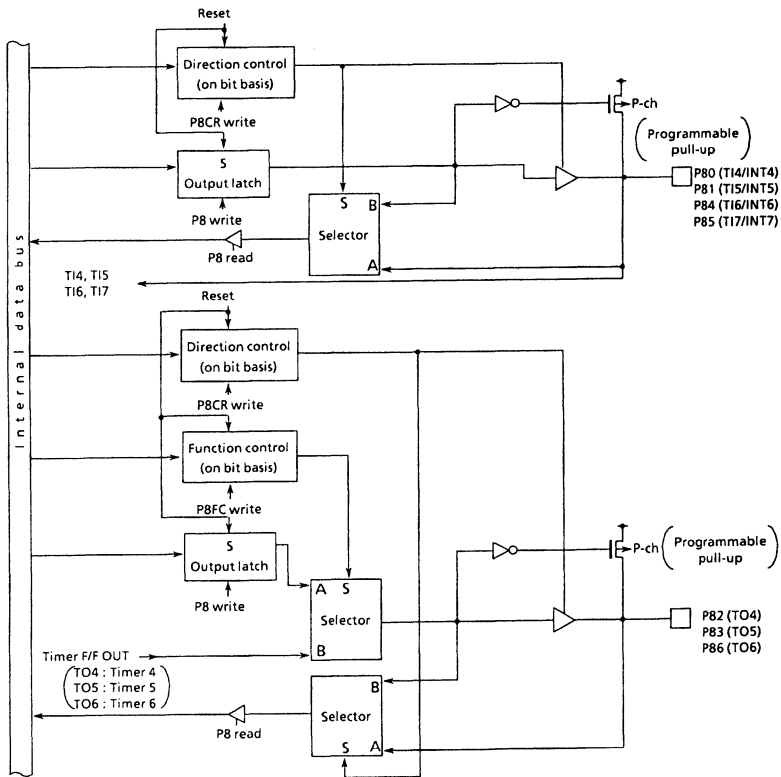


Figure 3.5 (17). Port 8 (P80 - P86)

(2) P87 (INT0)

an INT0 pin for external interrupt request input.

Port 87 is a general-purpose I/O port, and also used as

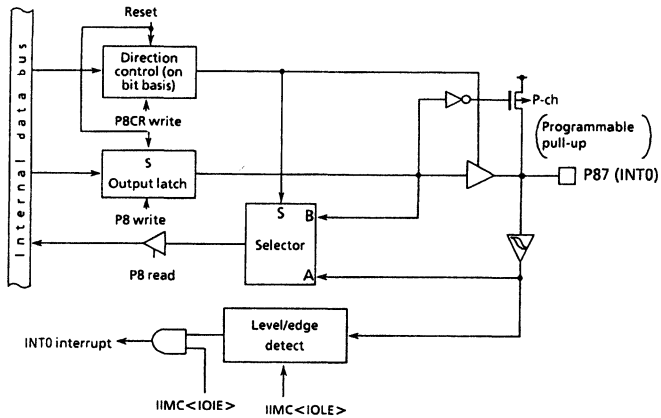


Figure 3.5 (18). Port 87

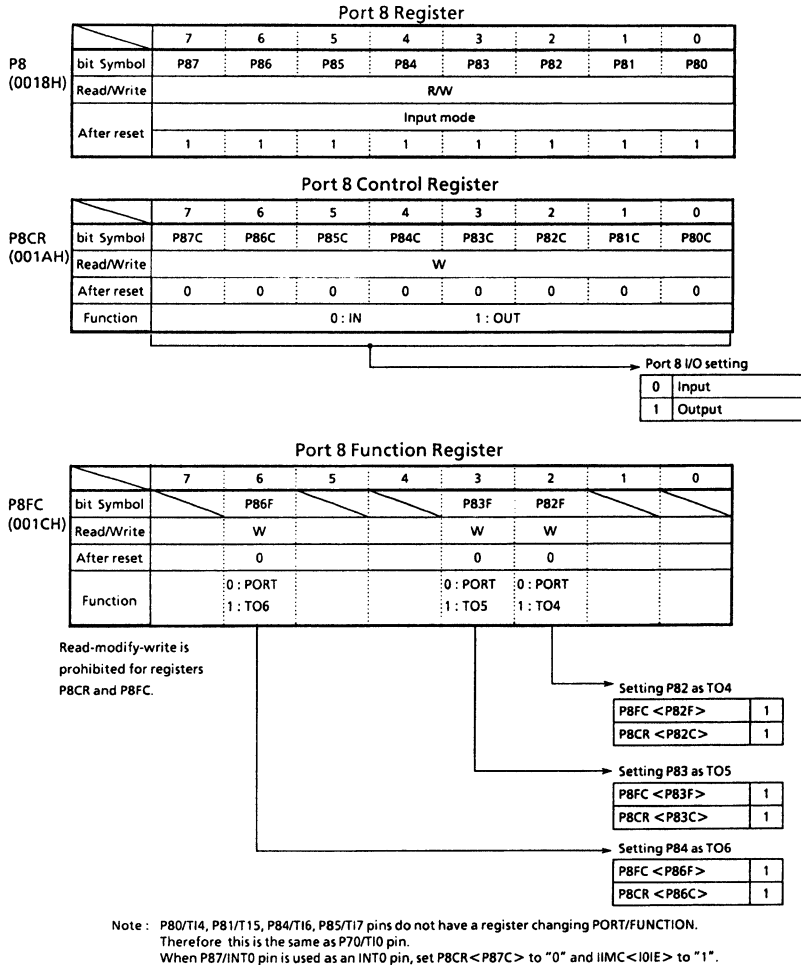


Figure 3.5 (19). Registers for Port 8

**3.5.10 Port 9 (P90 - P95)**

Port 9 is a 6-bit general-purpose I/O port. I/Os can be set on a bit basis. Resetting sets Port 9 to an input port and connects a pull-up resistor. It also sets all bits of the output latch register to 1.

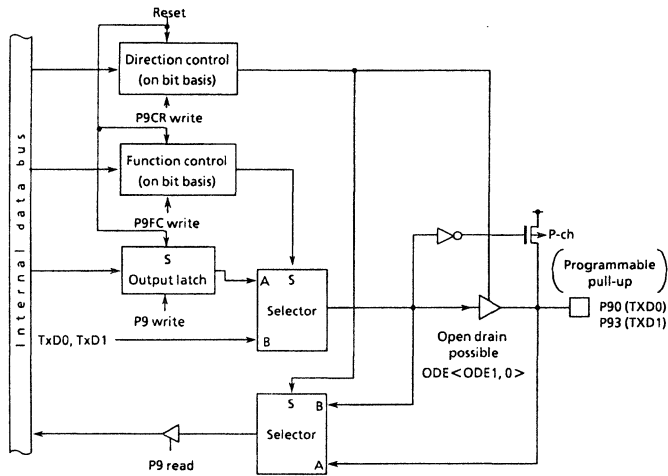
In addition to functioning as a general-purpose I/O port, Port 9 can also function as an I/O for serial channels 0 and 1. Writing 1 in the corresponding bit of the port 9 function register (P9FC) enables this function.

Resetting resets the function register value to 0 and sets all bits to ports.

- (1) Port 90 and 93 (TXD0/TXD1)

Ports 90 and 93 also function as serial channel TXD output pins in addition to I/O ports.

They have a programmable open drain function.



**Figure 3.5 (20). Ports 90 and 93**

- (2) Ports 91 and 94 (RXD0, 1) input pins for serial channels.

Ports 91 and 94 are I/O ports, and also used as RXD

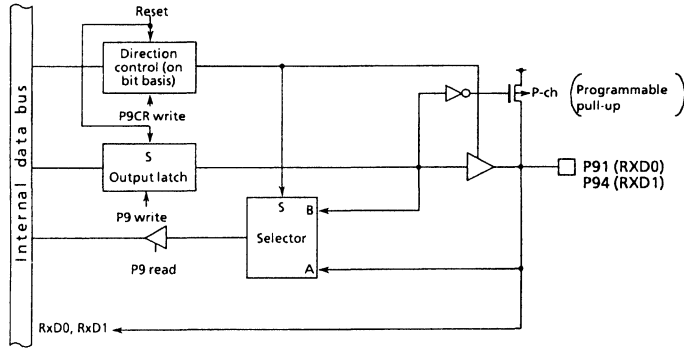


Figure 3.5 (21). Ports 91 and 94

- (3) Port 92 ( $\overline{\text{CTS}}$ /SCKL0) for serial channel0; additionally, the CTS0 pin, and also as a SCKL0 I/O pin.

Port 92 is an I/O port. It is also used as a  $\overline{\text{CTS}}$  input pin

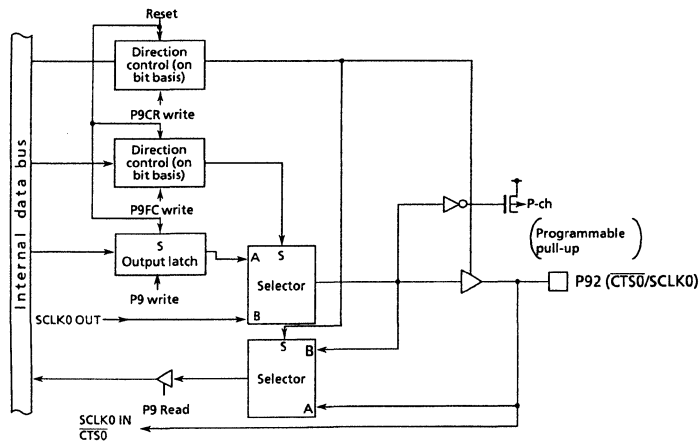


Figure 3.5 (22). Port 92

(4) Port 95 (SCLK)

an SCLK I/O pin for serial channel 1.

Port 95 is a general-purpose I/O port. It is also used as

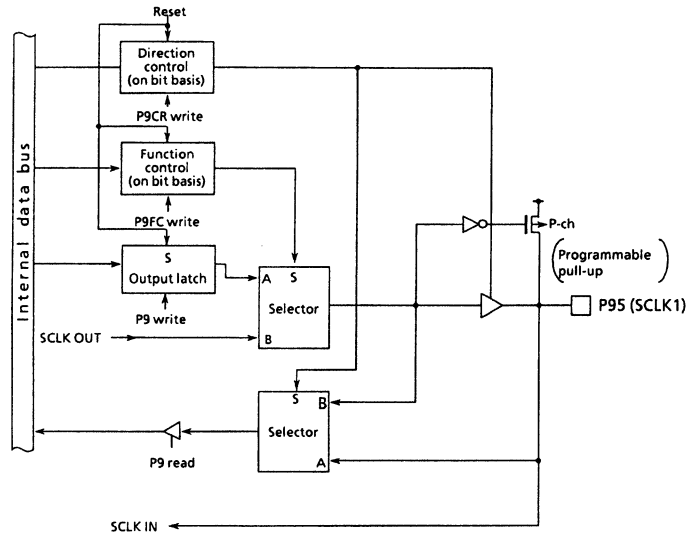
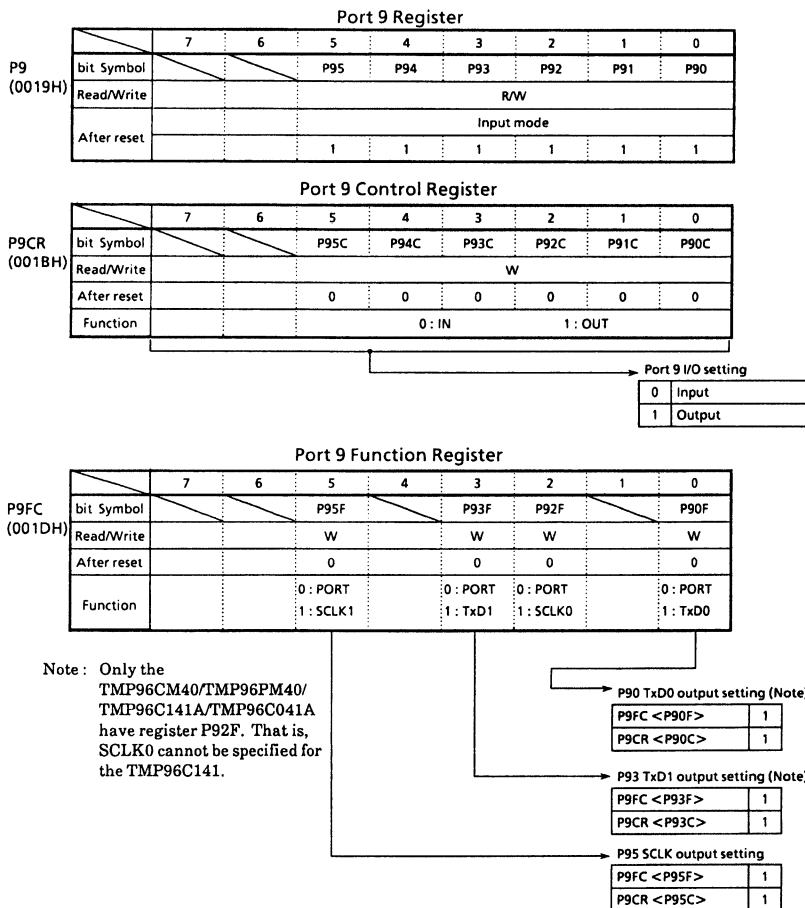


Figure 3.5 (23). Port 95



Note : To set the TxD pin to open drain, write 1 in bit 0 (for TxD0 pin) or bit 1 (for TxD1 pin) of the ODE register.  
 P91/RXD0, P94/RXD1 pins do not have a register changing PORT/FUNCTION.  
 Therefore this is the same as P70/II0 pin.

Figure 3.5 (24). Registers for Port 9

### 3.6 Chip Select/Wait Control

TMP96C141AF has a built-in chip select/wait controller used to control chip select ( $\overline{CS0}$  -  $\overline{CS2}$  pins), wait ( $\overline{WAIT}$  pin), and data bus size (8 or 16 bits) for any of the three block address areas.

#### 3.6.1 Control Registers

Table 3.6 (1) shows control registers

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

##### (1) Enable

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

##### (2) System only specification

Control register bit 6 (B0SYS, B1SYS, and B2SYS) 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 should not be accessed in normal mode (i.e., for system mode only memory data for the operating system).

##### (3) CS/CAS Waveform select

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

Resetting clears bit 5 to 0.

##### (4) Data bus size select

Bit 4 (B0BUS, B1BUS, and B2BUS) of the control reg-

ister is used to specify data bus size. Setting this bit to 0 accesses the memory in 16-bit data bus mode; setting it to 1 accesses the memory in 8-bit data bus mode.

Changing data bus size depending on the access address is called dynamic bus sizing. Table 3.6 (2) shows the details of the bus operation.

##### (5) Wait control

Control register bits 3 and 2 (B0W1, 0; B1W1, 0; B2W1, 0) are used to specify the number of waits. Setting these bits to 00 inserts a 2-state wait regardless of the  $\overline{WAIT}$  pin status. Setting them to 01 inserts a 1-state wait regardless of the  $\overline{WAIT}$  status. Setting them to 10 inserts a 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).

##### (6) Address area specification

Control register bits 1 and 0 (B0C1, 0; B1C1, 0; B2C1, 0) are used to specify the target address area. Setting these bits to 00 enables settings ( $\overline{CS}$  output, Wait state, Bus size, etc.) as follows:

- \* CS0 setting enabled when 7F00H to 7FFFH is accessed.
- \* CS1 setting enabled when 480H to 7FFFH is accessed.
- \* CS2 setting enabled when 8000H to 3FFFFFFFH is accessed, for the TMP96C141, which does not have a built-in ROM.

CS2 setting enabled when 10000H to 3FFFFFFFH is accessed for the TMP96CM40/TMP96PM40, which has built-in ROM/PROM

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



Table 3.6 (1) Chip Select/Wait Control Register

Code	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block0 CS/WAIT control register	0068H	B0E	B0SYS	B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			1 : CS/CAS Enable	1 : SYSTEM only	0 : $\overline{CS0}$ 1 : $\overline{CAS0}$	0 : 16-bit Bus 1 : 8-bit Bus	00 : 2WAIT 01 : 1WAIT 10 : 1WAIT + n 11 : 0WAIT	00 : 7F00H ~ 7FFFH 01 : 400000H ~ 10 : 800000H ~ 11 : C00000H ~		
B1CS	Block1 CS/WAIT control register	0069H	B1E	B1SYS	B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			1 : CS/CAS Enable	1 : SYSTEM only	0 : $\overline{CS1}$ 1 : $\overline{CAS1}$	0 : 16-bit Bus 1 : 8-bit Bus	00 : 2WAIT 01 : 1WAIT 10 : 1WAIT + n 11 : 0WAIT	00 : 480H ~ 7FFFH 01 : 400000H ~ 10 : 800000H ~ 11 : C00000H ~		
B2CS	Block2 CS/WAIT control register	006AH	B2E	B2SYS	B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W	W	W	W	W	W	W	W
			1	0	0	0	0	0	0	0
			1 : CS/CAS Enable	1 : SYSTEM only	0 : $\overline{CS2}$ 1 : $\overline{CAS2}$	0 : 16-bit Bus 1 : 8-bit Bus	00 : 2WAIT 01 : 1WAIT 10 : 1WAIT + n 11 : 0WAIT	00 : 8000H ~ 01 : 400000H ~ 10 : 800000H ~ 11 : C00000H ~		

Note: With only block 2, enable (16-bit data bus, 2-wait mode) after reset.

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 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 - b0
		16 bits	2n + 0	xxxxx	b7 - b0
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 - b0
		16 bits	2n + 1	b7 - b0	xxxxx
16 bits	2n + 0 (even number)	8 bits	2n + 0 2n + 1	xxxxx xxxxx	b7 - b0 b15 - b8
		16 bits	2n + 0	b15 - b8	b7 - b0
	2n + 1 (odd number)	8 bits	2n + 1 2n + 2	xxxxx xxxxx	b7 - b0 b15 - b8
		16 bits	2n + 1 2n + 2	b7 - b0 xxxxx	xxxxx b15 - b8
32 bits	2n + 0 (even number)	8 bits	2n + 0 2n + 1 2n + 2 2n + 3	xxxxx xxxxx xxxxx xxxxx	b7 - b0 b15 - b8 b23 - b16 b31 - b24
		16 bits	2n + 0 2n + 2	b15 - b8 b31 - b24	b7 - b0 b23 - b16
	2n + 1 (odd number)	8 bits	2n + 1 2n + 2 2n + 3 2n + 4	xxxxx xxxxx xxxxx xxxxx	b7 - b0 b15 - b8 b23 - b16 b31 - b24
		16 bits	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.

**3.6.2 Chip Select Image**

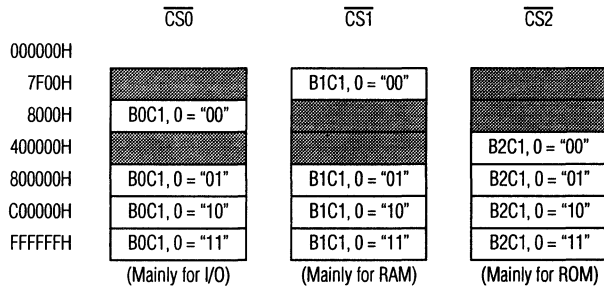
An image of the actual chip select is shown below. Out of the whole memory area, address areas that can be specified are divided into four parts. Addresses from 000000H to 3FFFFFFH are divided differently: 7F00H to 7FFFH is specified for CS0; 480H to 7FFFH, for CS1; and 8000H to 3FFFFFFH, for CS2. The reason is that a device other than ROM (i.e., RAM or I/O) might be connected externally.

The addresses 7F00 to 7FFFH (256 bytes) for CS0 are mapped mainly for possible expansions to external I/O.

The addresses 480H to 7FFFH (approximately 31K

bytes) for CS1 are mapped there mainly for possible extensions to external RAM.

The addresses 8000H to 3FFFFFFH (approximately 4Mbytes) for CS2 are mapped mainly for possible extensions to external ROM. After reset, CS2 is enabled in 16-bit bus and 2-wait. With the TMP96C141AF, which does not have a built-in ROM, the program is externally read at address 8000H in this setting (16-bit bus, 2-wait). With the TMP96CM40F/TMP96PM40F, which has a built-in ROM, addresses from 8000H to FFFFFFFH are used as the internal ROM area; CS2 is disabled in this area. After reset, the CPU reads the program from the built-in ROM in 16-bit bus, 0-wait mode.



Supplement 1: Access priority is highest for built-in I/O, then built-in memory, and lowest for the chip select/wait controller.

Supplement 2: External areas other than CS0 to CS2 are accessed in 16-bit data bus (0 wait) mode.

When using the chip select/wait controller, do not specify the same address area more than once. (However, when addresses 7F00H - 7FFFH for CS0 and 480H - 7FFFH for CS1 are specified, in other words, specifications overlap, only the CS0 setting/pin is active.)

3.6.3 Example of Usage

Figure 3.6 (1) is an example in which an external memory is con-

nected to the TMP96C141AF. In this example, a ROM is connected using 16-bit Bus; a RAM is connected using 8-bit Bus.

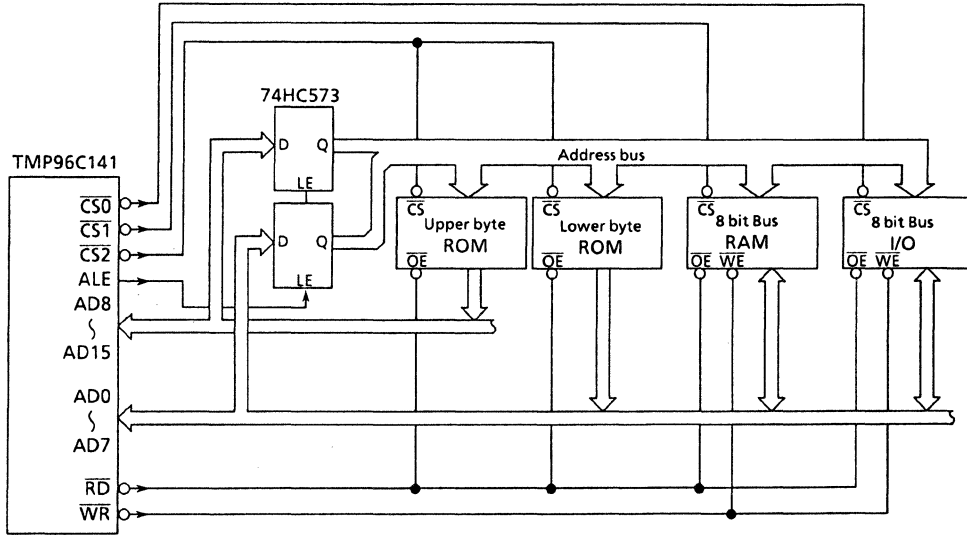


Figure 3.6 (1). Example of External Memory Connection (ROM = 16 bits, RAM and I/O = 8 bits)

Resetting sets pins  $\overline{CS0}$  to  $\overline{CS2}$  to input port mode.  $\overline{CS0}$  and  $\overline{CS1}$  are set high due to an internal pull-up resistor;  $\overline{CS2}$ ,

low due to an internal pull-down resistor. The program used to set these pins is as follows:

```

P4CR EQU 0EH
P4FC EQU 10H
B0CS EQU 68H
B1CS EQU 69H
B2CS EQU 6AH
LD (B0CS), 90H ; CS0 = 8 bits, 2WAIT, 7F00H ~ 7FFFH
LD (B1CS), 9CH ; CS1 = 8 bits, 0WAIT, 480H ~ 7EFFH
LD (B2CS), 84H ; CS2 = 16 bits, 1WAIT, 8000H ~ 3FFFFFH
LD (P4CR), 07H )  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS2}$  output mode setting
LD (P4FC), 07H
    
```

**3.6.4 How to Start with an 8-Bit Data Bus**

Resetting sets the 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:

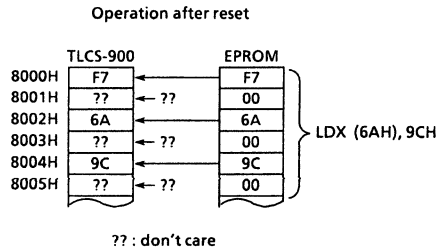
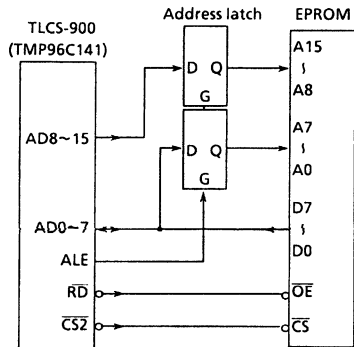
```

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 - 3FFFFFFH) is accessed in 8-bit data bus mode without any problem.

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



### 3.7 8-bit Timers

The TMP96C141AF contains two 8-bit timers (timers 0 and 1), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timer. The following four operating modes are provided for the 8-bit timers.

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

stant cycle) output mode (1 timer)

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

Each interval timer consists of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Besides, one timer flip-flop (TFF1) is provided for pair of timer 0 and timer 1.

Among the input clock sources for the interval timers, the internal clocks of  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$ , and  $\phi T256$  are obtained from the 9-bit prescaler shown in Figure 3.7 (2).

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

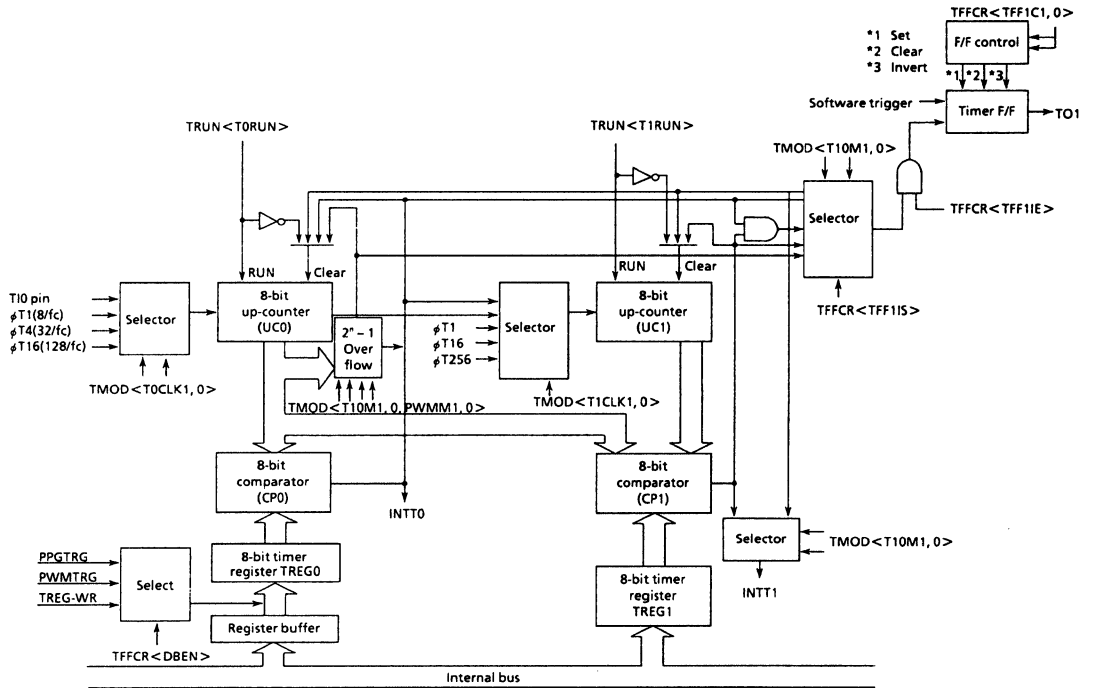


Figure 3.7 (1). Block Diagram of 8-Bit Timers (Timers 0 and 1)

① Prescaler

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 four types of clock:

$\phi T1$ ,  $\phi T4$ ,  $\phi T16$ , and  $\phi T256$ .

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	16MHz	20MHz
$\phi T1 (8/fc)$	0.5 $\mu s$	0.4 $\mu s$
$\phi T4 (32/fc)$	2.0 $\mu s$	1.6 $\mu s$
$\phi T16 (128/fc)$	8.0 $\mu s$	6.4 $\mu s$
$\phi T256 (2048/fc)$	128 $\mu s$	102 $\mu s$

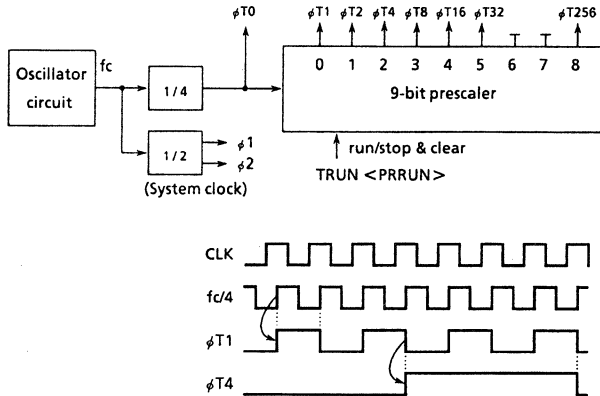


Figure 3.7 (2). Prescaler

② Up-counter

This is an 8-bit binary counter which counts up by the input clock pulse specified by TMOD.

The input clock of timer 0 is selected from the external clock from T10 pin and the three internal clocks  $\phi T1$  (8/fc),  $\phi T4$  (32/fc), and  $\phi T16$  (128/fc), according to the set value of TMOD register.

The input clock of timer 1 differs depending on the operation mode. When set to 16-bit timer mode, the overflow output of timer 0 is used as the input clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks  $\phi T1$  (8/fc),  $\phi T16$  (128/fc), and  $\phi T256$  (2048/fc) as well as the comparator output (match detection signal) of timer 0 according to the set value of TMOD register.

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

When TMOD <T10M1, 0> = 00 and TMOD <T1CLK1, 0> = 01,  $\phi T1$  (8/fc) becomes the input of timer 1 (8 bit timer mode).

Operation mode is also set by TMOD register. When reset, it is initialized to TMOD <T01M1, 0> = 00 whereby the up-counter is placed in the 8-bit timer mode.

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



③ Timer register

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

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

The timer flip-flop control register TFFCR <DBEN> bit controls whether the double buffer structure in the

TREG0 should be enabled or disabled. It is disabled when <DBEN> = 0 and enabled when they are set to 1.

In the condition of double buffer enable state, the data is 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. Therefore, during timer mode, the double buffer cannot be used.

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

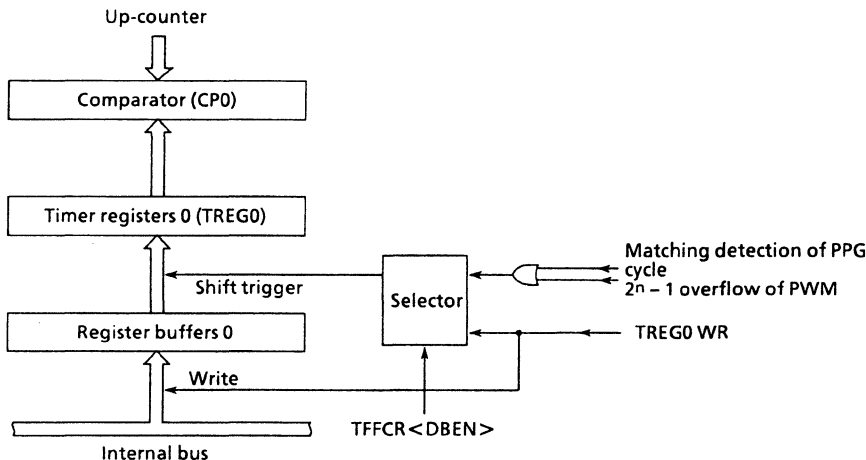


Figure 3.7 (3). Configuration of Timer Register 0

Note : Timer register and the register buffer are allocated to the same memory address. When <DBEN> = 0, the same value is written in

the register buffer as well as the timer register, while when <DBEN> = 1 only the register buffer is written.

The memory address of each timer register is as follows.

TREG0: 000022H

TREG1: 000023H

All registers are write-only and cannot be read.

④ Comparator

A comparator compares the value in the up-counter with the values to which the timer register is set. When they match, the up-counter is cleared to zero and an interrupt signal (INTT0, INTT1) is generated. If the timer

flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

⑤ Timer flip-flop (timer F/F: TFF1)

The status of the timer flip-flop is inverted by the match detect signal (comparator output) of each interval timer and the value can be output to the timer output pins TO1 (also used as P71).

A timer F/F is provided for a pair of timer 0 and timer 1 and is called TFF1. TFF1 is output to TO1 pin.

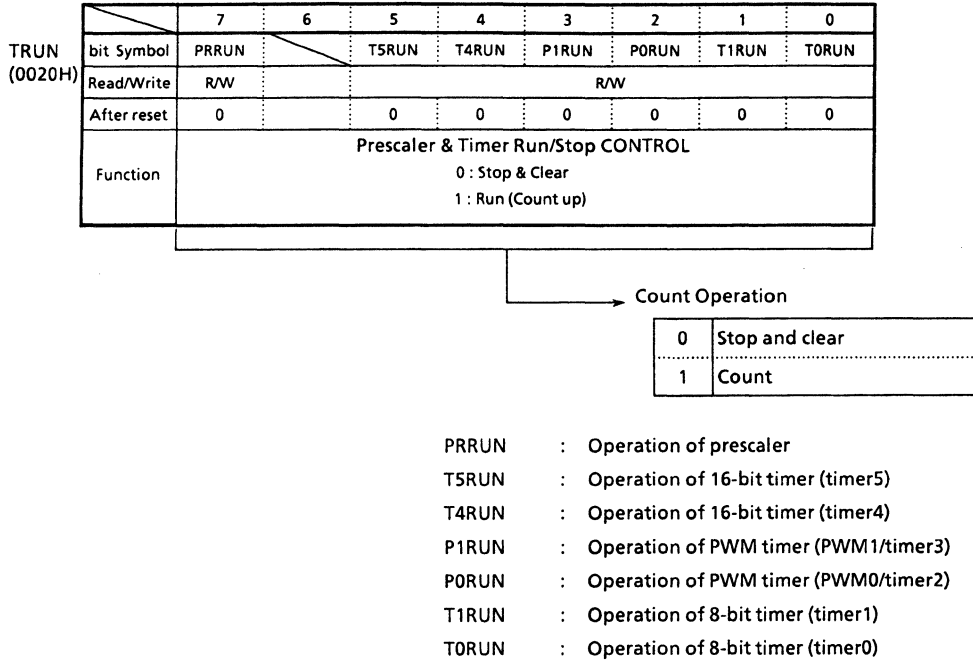


Figure 3.7 (4). Timer Operation Control Register (TRUN)

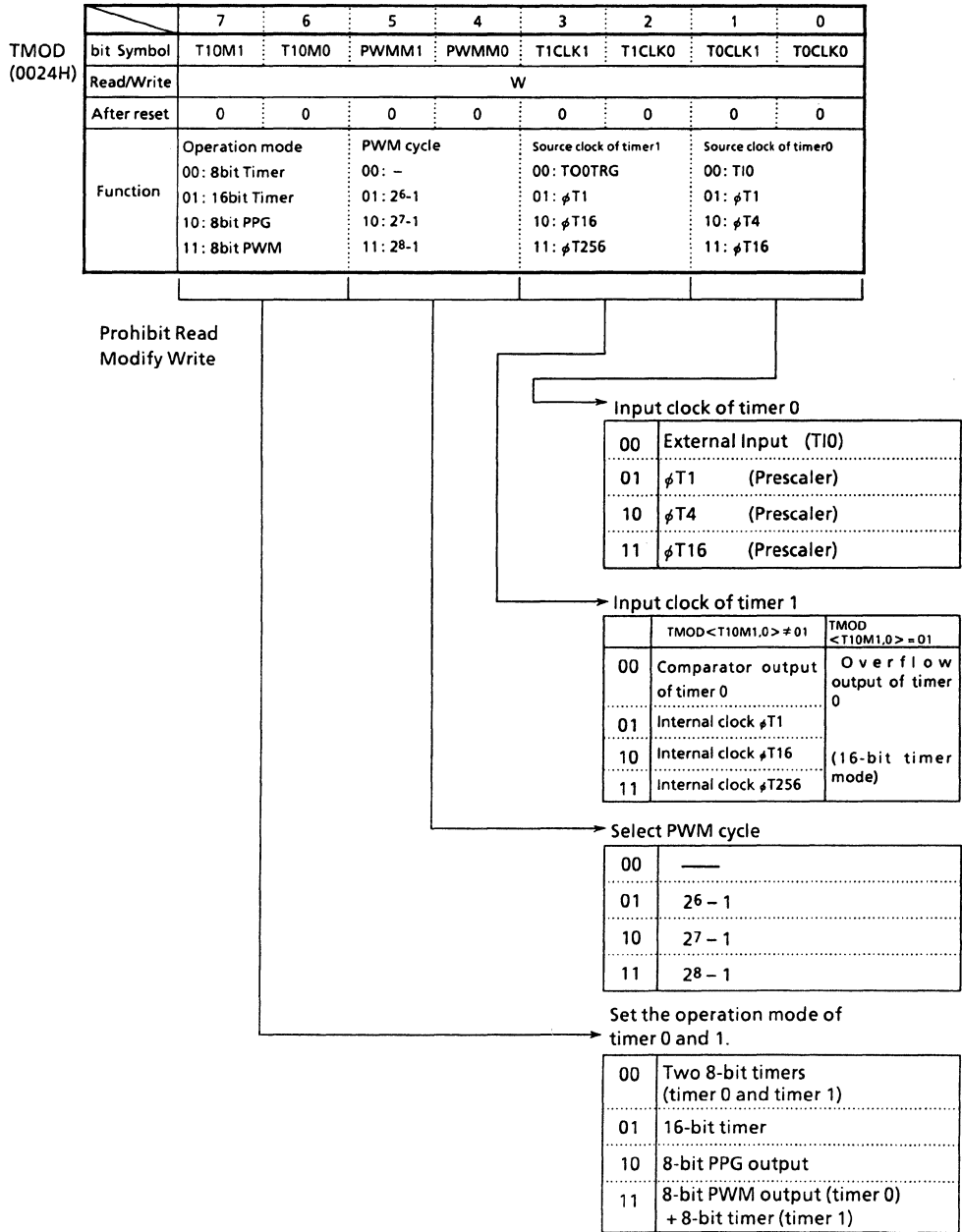


Figure 3.7 (5). Timer Mode Control Register (TMOD)

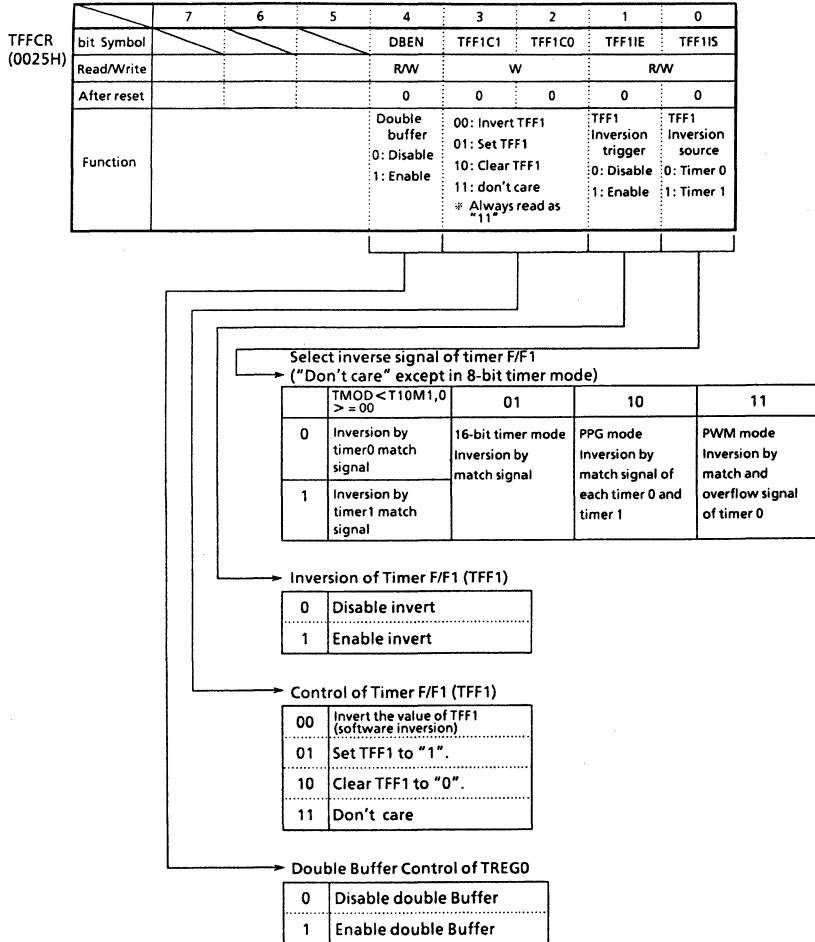


Figure 3.7 (6). Timer Flip-Flop Control Register (TFFCR)

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

(1) 8-bit timer mode

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

① Generating interrupts in a fixed cycle

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

Example: To generate timer 1 interrupt every 40 microseconds at  $f_c = 16\text{ MHz}$ , set each register in the following manner.

	MSB							LSB
	7	6	5	4	3	2	1	0
TRUN	← -	x	-	-	-	-	0	-
TMOD	← 0	0	x	x	0	1	-	-
TREG1	← 0	1	0	1	0	0	0	0
INTET10	← 1	1	0	1	-	-	-	-
TRUN	← 1	x	-	-	-	-	1	-

Stop timer 1, and clear it to "0".

Set the 8-bit timer mode, and select  $\phi T1$  ( $0.5\mu s @ f_c = 16\text{MHz}$ ) as the input clock.

Set the timer register at  $40\mu s \phi T1 = 50H$ .

Enable INTT1, and set it to "Level 5".

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 $f_c = 16\text{MHz}$ )	Resolution	Interrupt Cycle (at $f_c = 20\text{MHz}$ )	Resolution
$\phi T1$ (8/ $f_c$ )	$0.5\mu s \sim 128\mu s$	$0.5\mu s$	$0.4\mu s \sim 102.4\mu s$	$0.4\mu s$
$\phi T4$ (32/ $f_c$ )	$2\mu s \sim 512\mu s$	$2\mu s$	$1.6\mu s \sim 409.6\mu s$	$1.6\mu s$
$\phi T16$ (128/ $f_c$ )	$8\mu s \sim 2.048\text{ms}$	$8\mu s$	$6.4\mu s \sim 1.638\text{ms}$	$6.4\mu s$
$\phi T256$ (2048/ $f_c$ )	$128\mu s \sim 32.708\text{ms}$	$128\mu s$	$102.4\mu s \sim 2.621\text{ms}$	$128\mu s$

Note: The input clock of timer 0 and timer 1 are different from as follows:

Timer 0: T10 input,  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$

Timer 1: Match Output of Timer 0,  $\phi T1$ ,  $\phi T16$ ,  $\phi T256$

② Generating a 50% duty square wave pulse

The timer flip-flop (TFF1) is inverted at constant intervals, and its status is output to timer output pin (TO1).

TO1 pin at  $f_c = 16\text{MHz}$ , set each register in the following procedures. Either timer 0 or timer 1 may be used, but this example uses timer 1.

Example: To output a  $3.0\mu\text{s}$  square wave pulse from

		7	6	5	4	3	2	1	0	
TRUN	←	-	x	-	-	-	-	0	-	Stop timer 1, and clear it to "0".
TMOD	←	0	0	x	x	0	1	-	-	Set the 8-bit timer mode, and select $\phi\text{T1}$ ( $0.5\mu\text{s}$ @ $f_c = 16\text{MHz}$ ) as the input clock.
TREG1	←	0	0	0	0	0	0	1	1	Set the timer register at $3.0\mu\text{s} + \phi\text{T1} + 2 = 3$ .
TFFCR	←	-	-	-	-	1	0	1	1	Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.
P7CR	←	x	x	x	x	-	-	1	-	) Select P71 as TO1 pin.
P7FC	←	x	x	x	x	-	-	1	x	
TRUN	←	1	x	-	-	-	-	1	-	Start timer 1 counting.

Note: x; don't care    -; no change

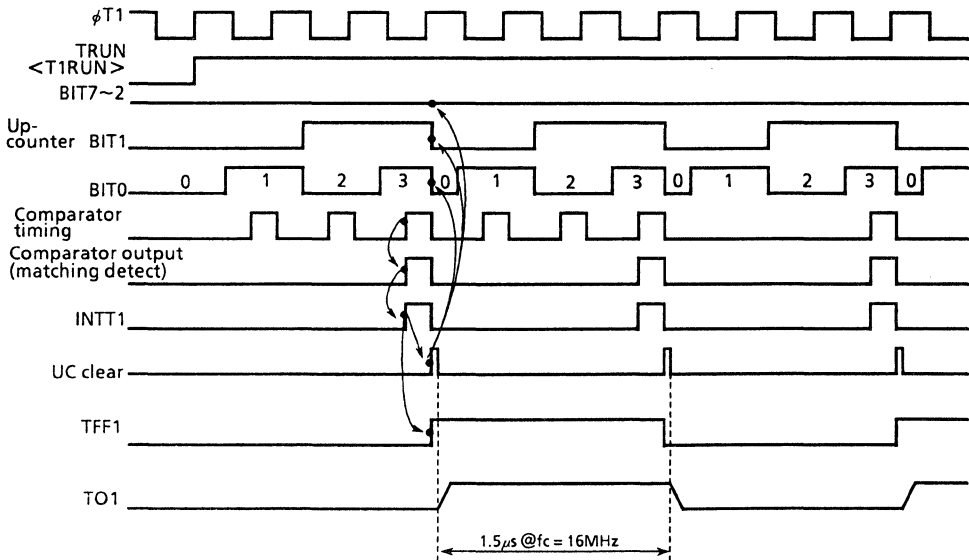


Figure 3.7 (7). Square Wave (50% Duty) Output Timing Chart

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

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

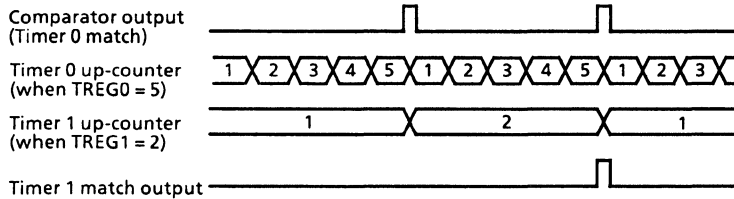


Figure 3.7 (8). Timer 1 Count Up by Timer 0

- ④ Output inversion with software

Note: The value of timer register cannot be read.

The value of timer flip-flop (TFF1) can be inverted, independent of timer operation.

Writing "00" into TFFCR <TFF1C1, 0> (memory address: 000025h of bit 3 and bit 2) inverts the value of TFF1.

- ⑤ Initial setting of timer flip-flop (TFF1)

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".

- (2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of timer 0 and timer 1.

To make a 16-bit interval timer by cascade connecting timer 0 and timer 1, set timer 0/timer 1 mode register TMOD <T10M1, 0> to "0, 1".

When set in 16-bit timer mode, the overflow output of timer 0 will become the input clock of timer 1, regardless of the set value of TMOD <T1CLK1, 0>. Table 3.7 (2) shows the relation between the cycle of timer (interrupt) and the selection of input clock.

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

Input Clock	Interrupt Cycle (at fc = 16MHz)	Resolution	Interrupt Cycle (at fc = 20MHz)	Resolution
$\phi T1$ (8/fc)	0.5 $\mu$ s ~ 32.786ms	0.5 $\mu$ s	0.4 $\mu$ s ~ 26.214ms	0.4 $\mu$ s
$\phi T4$ (32/fc)	2 $\mu$ s ~ 131.072ms	2 $\mu$ s	1.6 $\mu$ s ~ 104.857ms	1.6 $\mu$ s
$\phi T16$ (128/fc)	8 $\mu$ s ~ 524.288ms	8 $\mu$ s	6.4 $\mu$ s ~ 419.430ms	6.4 $\mu$ s

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

Setting example: To generate an interrupt INTT1 every 0.5 seconds at  $f_c = 16\text{MHz}$ , set the following values for timer registers TREG0 and TREG:

When counting with input clock of  $\phi T16$  ( $8\mu\text{s}$  @  $16\text{MHz}$ )

$$0.5 \text{ sec} \div 8\mu\text{s} = 62500 = \text{F424H}$$

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

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

With the timer 1 comparator, the match detect signal is output at each comparator timing when up-counter UC1 and TREG1 values match. When the match detect signal is output simultaneously from both comparators of timer 0 and timer 1, the up-counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If inversion is enabled, the value of the timer flip-flop TFF1 is inverted.

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

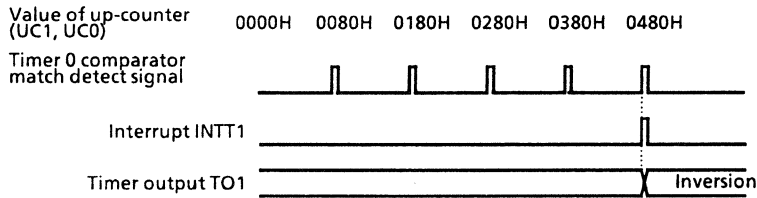


Figure 3.7 (9). Output Timer by 16-Bit Timer Mode

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

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

Timer 0 outputs pulse to TO1 pin (also used as P70). In this mode, a programmable square wave is generated by inverting timer output each time the 8-bit up-

counter (UC0) matches the timer registers TREG0 and TREG1.

However, it is required that the set value of TREG0 is smaller than that of TREG1.

Though the up-counter (UC1) of timer 1 is not used in this mode, UC1 should be set for counting by setting  $\text{TRUN} <T1\text{RUN}>$  to 1.

Figure 3.7 (11) shows the block diagram for this mode.



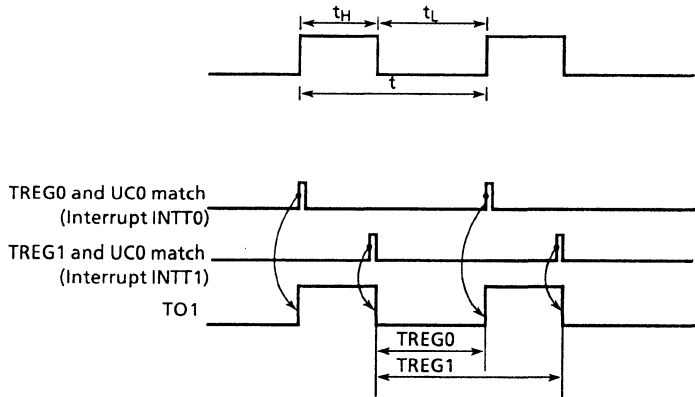


Figure 3.7 (10). 8-Bit PPG Output Waveforms

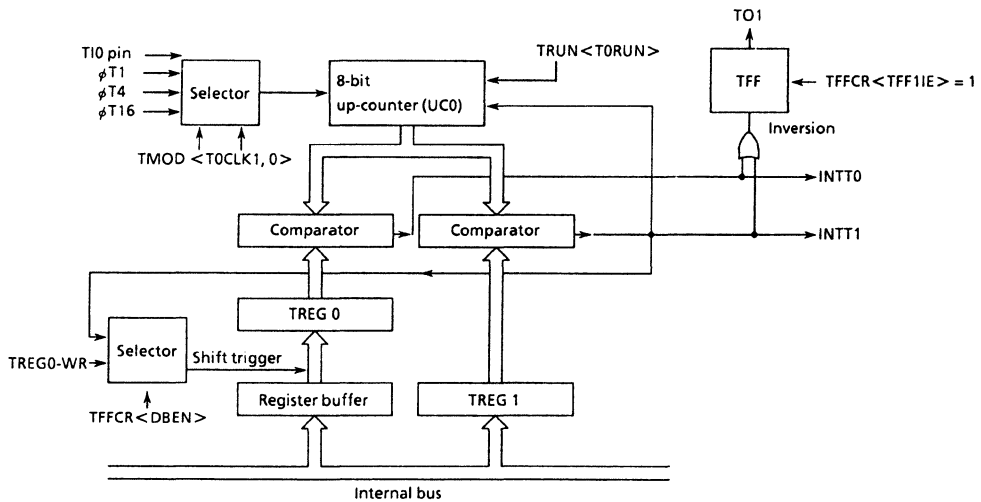


Figure 3.7 (11). Block Diagram of 8-Bit PPG Output Mode

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

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

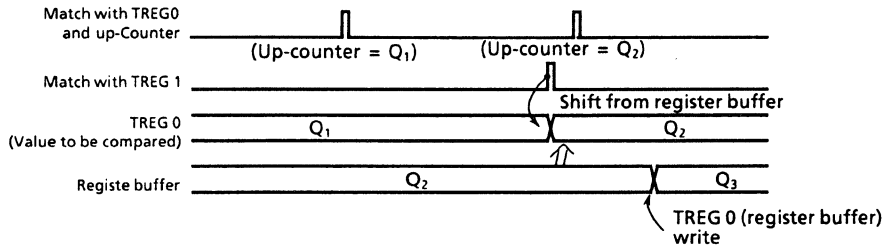
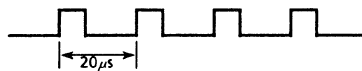


Figure 3.7 (12). Operation of Register Buffer

Example: Generating 1/4 duty 50KHz pulse @  $f_c = 16\text{MHz}$



- Calculate the value to be set for timer register.  
 To obtain the frequency 50KHz, the pulse cycle  $t$  should be:  $t = 1/50\text{KHz} = 20\mu\text{s}$ .  
 Given  $\phi T1 = 0.5\mu\text{s}$  @  $16\text{MHz}$ ,  
 $20\mu\text{s} + 0.5\mu\text{s} = 40$   
 Consequently, to set the timer register 1 (TREG1) to

TREG1 = 40 = 28H and then duty to 1/4,  $t \times 1/4 = 20\mu\text{s} \times 1/4 = 5\mu\text{s}$   
 $5\mu\text{s} + 0.5\mu\text{s} = 10$   
 Therefore, set timer register 0 (TREG0) to TREG0 = 10 = 0AH.

		7	6	5	4	3	2	1	0	
TRUN	←	-	x	-	-	-	-	0	0	Stop timer 0, and clear it to "0".
TMOD	←	1	0	x	x	x	x	0	1	Set the 8-bit PPG mode, and select $\phi T1$ as input clock.
TREG0	←	0	0	0	0	1	0	1	0	Write "0AH".
TREG1	←	0	0	1	0	1	0	0	0	Write "28H".
TFFCR	←	-	-	-	1	0	1	1	x	Sets TFF1 and enables the inversion and double buffer enable.
										Writing "10" provides negative logic pulse.
P7CR	←	x	x	x	x	-	-	1	-	) Set P71 as TO1 pin.
P7FC	←	x	x	x	x	-	-	1	x	
TRUN	←	1	x	-	-	-	-	1	1	Start timer 0 and timer 1 counting.

Note : x; don't care    -; no change

(4) 8-bit PWM Output mode

This mode is valid only for timer 0. In this mode, maximum 8-bit resolution of PWM pulse can be output.

PWM pulse is output to TO1 pin (also used as P71) when using timer 0. Timer 1 can also be used as 8-bit timer.

Timer output is inverted when up-counter (UC0) matches the set value of timer register TREG0 or when  $2^n - 1$  ( $n = 6, 7, \text{ or } 8$ ; specified by T01MOD <PWM01,

$0 >$ ) counter overflow occurs. Up-counter UC0 is cleared when  $2^n - 1$  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  $\neq 0$ )

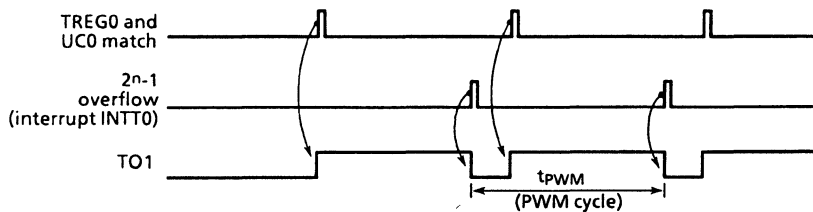


Figure 3.7 (13). 8-Bit PWM Waveforms

Figure 3.7 (14) shows the block diagram of this mode.

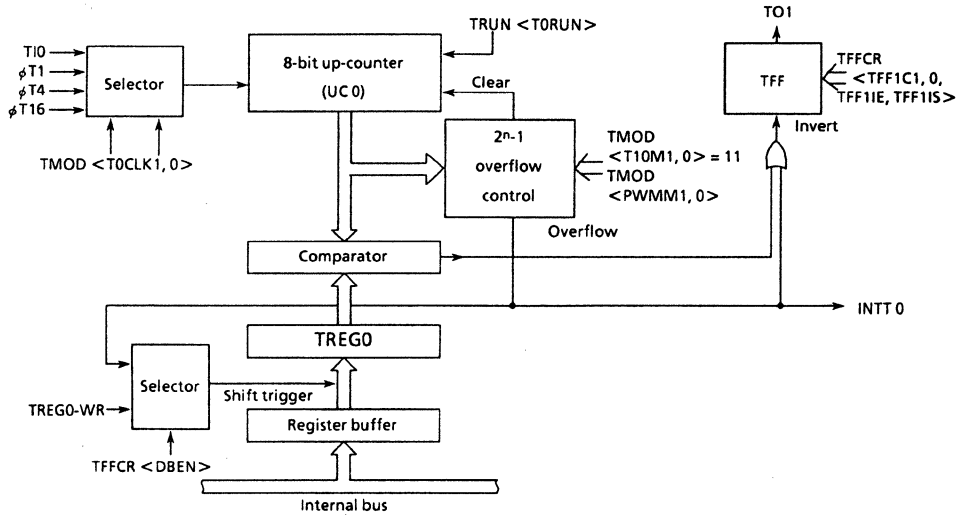


Figure 3.7 (14). Block Diagram of 8-Bit PWM Mode

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

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

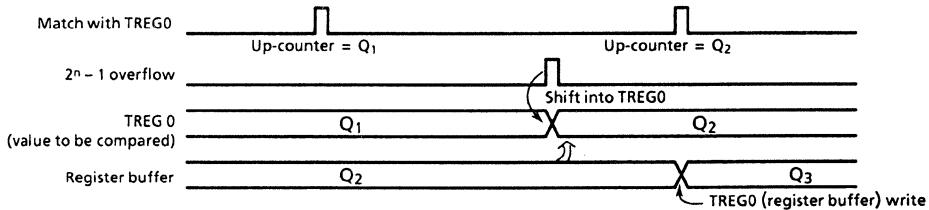
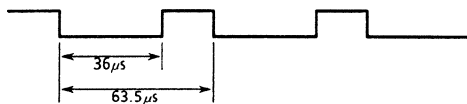


Figure 3.7 (15). Operation of Register Buffer

Example: To output the following PWM waves to TO1 pin at  $f_c = 16\text{MHz}$ .



To realize  $63.5\mu\text{s}$  of PWM cycle by  $\phi T1 = 0.5\mu\text{s}$  ( $f_c = 16\text{MHz}$ ),

$$63.5\mu\text{s} + 0.5\mu\text{s} = 127 = 2^7 - 1$$

Consequently,  $n$  should be set to 7.

As the period of low level is  $36\mu\text{s}$ , for  $\phi T1 = 0.5\mu\text{s}$ , set the following value for TREG0:

$$36\mu\text{s} + 0.5\mu\text{s} = 72 = 48H$$

	MSB							LSB		
	7	6	5	4	3	2	1	0		
TRUN	←	-	x	-	-	-	-	0	Stop timer 0, and clear it to "0".	
TMOD	←	1	1	1	0	-	-	0	1	Set 8-bit PWM mode (cycle: $2^7 - 1$ ) and select $\phi T1$ as the input clock.
TREG0	←	0	1	0	0	1	0	0	0	Write "48H".
TFFCR	←	x	x	x	x	1	0	1	x	Clears TFF1, enables the inversion and double buffer.
P7CR	←	x	x	x	x	-	-	1	-	) Set P71 as the TO1 pin.
P7FC	←	x	x	x	x	-	-	1	x	
TRUN	←	1	x	-	-	-	-	-	1	Start timer 0 counting.

Note: x; don't care      -; no change

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

	PWM Cycle (@ $f_c = 16\text{MHz}$ )			PWM Cycle (@ $f_c = 20\text{MHz}$ )		
	$\phi T1$	$\phi T4$	$\phi T16$	$\phi T1$	$\phi T4$	$\phi T16$
$2^6 - 1$	31.5 $\mu\text{sec}$ (31.7kHz)	126msec (7.9kHz)	0.50 $\mu\text{sec}$ (1.9kHz)	25.2 $\mu\text{sec}$ (39.0kHz)	100 $\mu\text{sec}$ (10.0kHz)	0.40msec (2.4kHz)
$2^7 - 1$	63.5 $\mu\text{sec}$ (15.7kHz)	254msec (3.9kHz)	1.01 $\mu\text{sec}$ (0.98kHz)	50.8 $\mu\text{sec}$ (19.7kHz)	203 $\mu\text{sec}$ (4.9kHz)	0.81msec (1.2kHz)
$2^8 - 1$	127 $\mu\text{sec}$ (7.8kHz)	510msec (1.9kHz)	2.04 $\mu\text{sec}$ (0.49kHz)	102 $\mu\text{sec}$ (9.80kHz)	408 $\mu\text{sec}$ (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**

Register Name	TMOD				TFFCR
	T10M	PWMM	T1CLK	TOCLK	TFF1IS
Function	Timer Mode	PWMO Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select
16-bit timer mode	01	-	-	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	-
8-bit timer x 2 channels	00	-	Lower timer match : $\phi T1, \phi T16, \phi T256$ (00, 01, 10, 11)	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	0 : Lower timer output 1 : Upper timer output
8-bit PPG x 1 channel	10	-	-	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	-
8-bit PWM x 1 channel	11	$2^6 - 1, 2^7 - 1, 2^8 - 1$ (01, 10, 11)	-	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	-
8-bit timer x 1 channel	11	-	$\phi T1, \phi T16, \phi T256$ (01, 10, 11)	-	Output disabled

Note: -; don't care

### 3.8 8-Bit PWM Timer

The TMP96C141AF/TMP96CM40F/TMP96PM40F has two built-in 8-bit PWM timers (timers 2 and 3).

They have two operating modes.

- 8-bit PWM (pulse width modulation: variable duty at fixed interval) output mode
- 8-bit interval timer mode

Figure 3.8 (1) is a block diagram of 8-bit PWM timer (timers 2 and 3).

PWM timers consist of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Two timer flip-flops (TFF2 for timer 2 and TFF3 for timer 3) are provided.

Input clocks  $\phi P1$ ,  $\phi P4$ , and  $\phi P16$  for the PWM timers can be obtained using the built-in prescaler.

PWM timer operating mode and timer flip-flops are controlled by four control registers (P0MOD, P1MOD, PFFCR, and TRUN).

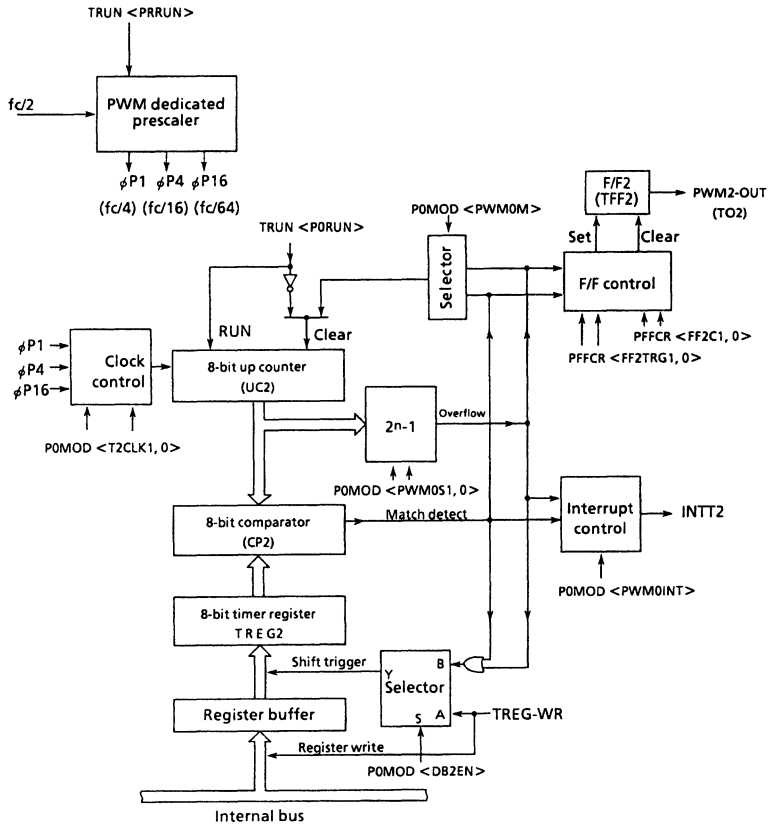


Figure 3.8 (1). Block Diagram of 8-Bit PWM Timer 0 (Timer 2)

Note: Block diagram for 8-bit PWM timer 1 (timer 3) is the same as the above diagram.

① Prescaler

Generates input clocks dedicated to PWM timers by further dividing the fundamental clock (fc) after it has been divided by 2 (fc/2). Since the register used to control the prescaler is the same as the one for other timers, the prescaler cannot be operated independently.

The PWM timer uses three input clocks:  $\phi/P1$ ,  $\phi/P4$ , and  $\phi/P16$ .

Like the 9-bit prescaler described in the 8-bit timer section, this prescaler can be counted/stopped using bit 7 <PRRUN> of the timer operation control register TRUN. Setting <PRRUN> to 1 starts counting; setting it to 0 zero-clears and stops counting. Resetting clears <PRRUN> to 0, which clears and stops the prescaler.

Dedicated Prescaler Cycle

	16MHz	20MHz
$\phi P1$ (4/fc)	250ns	200ns
$\phi P4$ (16/fc)	1 $\mu$ s	800ns
$\phi P16$ (64/fc)	4 $\mu$ s	3.2 $\mu$ s

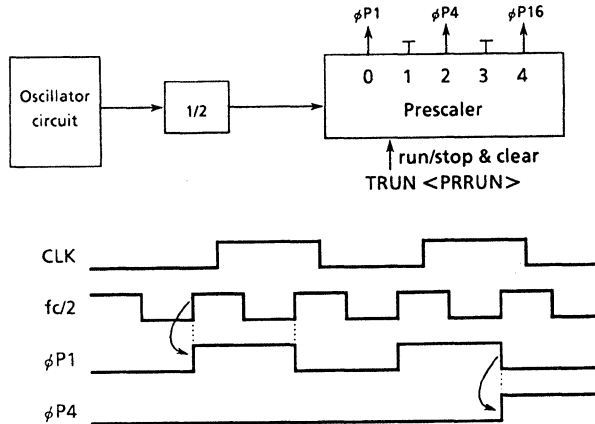


Figure 3.8 (2). Prescaler

② Up-counter

An 8-bit binary counter which counts up using the input clock specified by PWM mode register (P0MOD or P1MOD).

The input clock for the PWM0/PWM1 is selected from the internal clocks  $\phi P1$ ,  $\phi P4$ , and  $\phi P16$  (PWM dedicated prescaler output) depending on the value set in the P0MOD/P1MOD register.

Operating mode is also set by P0MOD and P1MOD registers. At reset, they are initialized to P0MOD <PWM0M> = 0 and P1MOD <PWM1M> = 0, thus, the up-counter is in PWM mode. In PWM mode, the up-counter is cleared when a  $2^N - 1$  overflow occurs; in timer mode, the up-counter is cleared at compare and

match.

Count/stop and clear of the up-counter can be controlled for each PWM timer using the timer operation control register TRUN. Resetting clears all up-counters and stops timers.

③ Timer registers

Two 8-bit registers used for setting an interval time. When the value set in the timer registers (TREG 2 and 3) matches the value in the up-counter, the match detect signal of the comparator becomes active.

Timer registers TREG2 and TREG3 are each paired with register buffer to make a double buffer structure.



TREG2 and TREG3 are controlled double buffer enable/disable by P0MOD <DB2EN> and P1MOD <DB3EN> : disabled when <DB2EN>/<DB3EN> = 0, enabled when <DB2EN>/<DB3EN> = 1.

Data is transferred from register buffer to timer when a  $2^n - 1$  overflow occurs in the PWM mode, or when compare and match occurs in 8-bit timer mode. That is, with a PWM timer, the timer mode can be operated

in double buffer enable state, unlike timer mode for timers 0 and 1.

At reset, <DB2EN>/<DB3EN> is initialized to 0 to disable double buffer. To use double buffer, write the data in the timer register at first, then set <DB2EN>/<DB3EN> to 1, and write the following data in the register buffer.

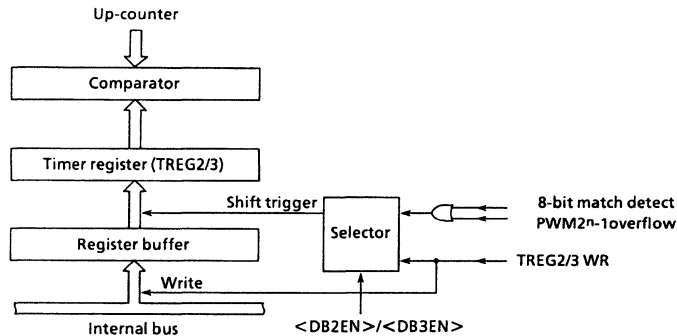


Figure 3.8 (3). Structure of Timer Registers 2 and 3

Note: The timer register and register buffer are allocated to the same memory address. When <DB2EN>/<DB3EN> = 0, the same value is written to both register buffer and timer register. When <DB2EN>/<DB3EN> = 1, the value is written to the register buffer only.

Memory addresses of the timer registers are as follows:

TREG2 : 000026H

TREG3 : 000027H

Both timer registers are write only; however, register buffer values can be read when reading the above addresses.

④ Comparator

Compares the value in the up-counter with the value in the timer register (TREG2/TREG3). When they match,

the comparator outputs the match detect signal. A timer interrupt (INTT2/INTT3) is generated at compare and match if the interrupt select bit <PWM01NT>/<PWM1NT> of the mode register (P0MOD/P1MOD) is set to 1. In timer mode, the comparator clears the up-counter to 0 at compare and match. It also inverts the value of the timer flip-flop if timer flip-flop invert is enabled.

⑤ Timer flip-flop

The value of the timer flip-flop is inverted by the match detect signal (comparator output) of each interval timer or  $2^n - 1$  overflow. The value can be output to the timer output pin TO2/TO3 (also used as P72/P73).

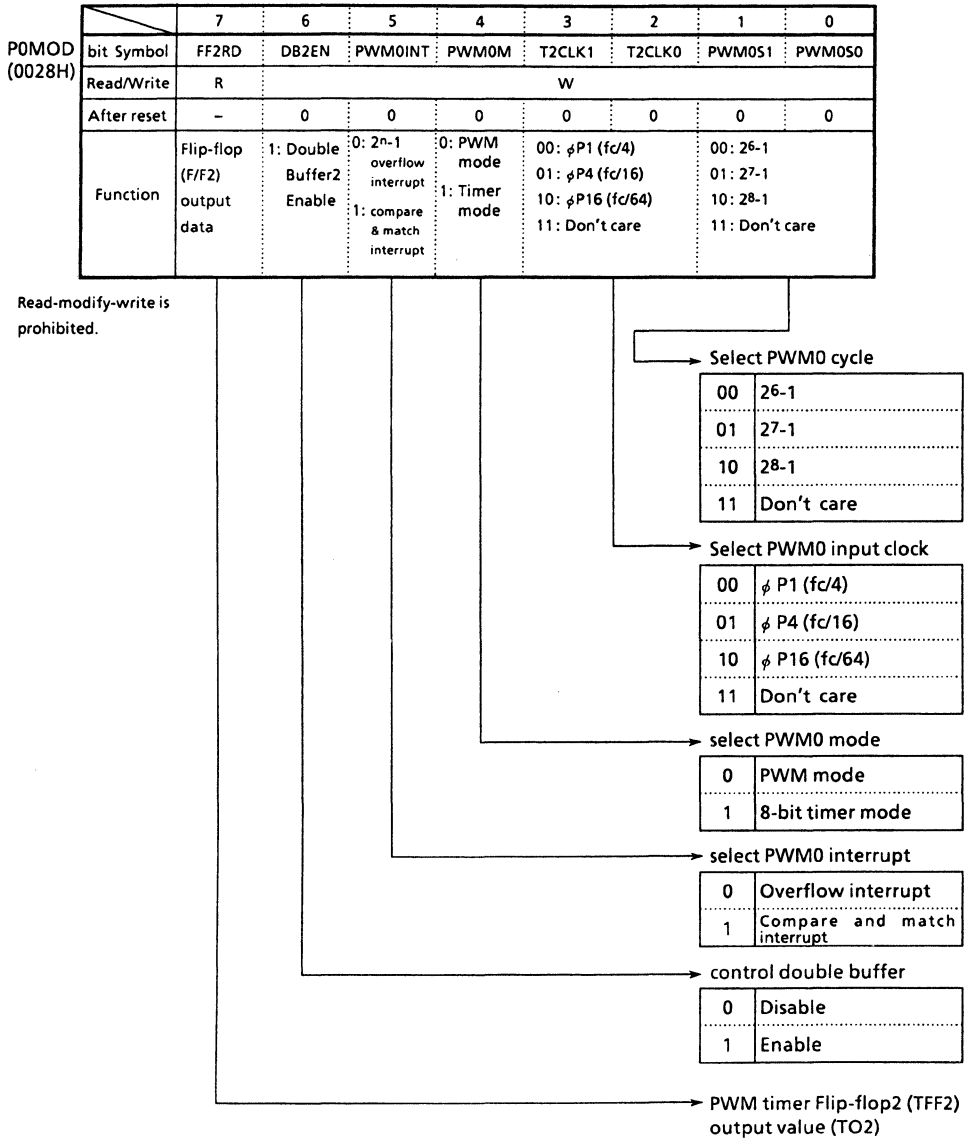


Figure 3.8 (4). 8-Bit PWM0 Mode Control Register

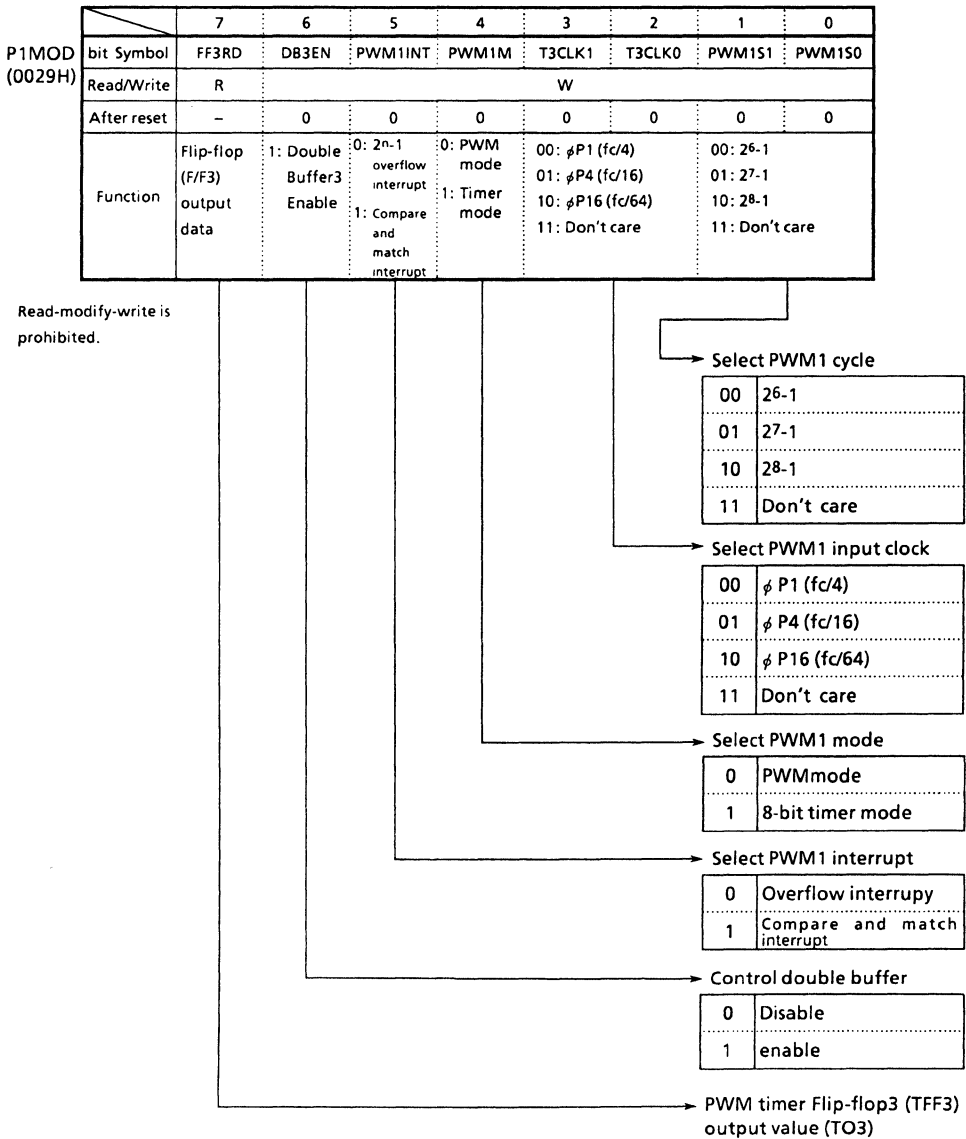


Figure 3.8 (5). 8-Bit PWM1 Mode Control Register

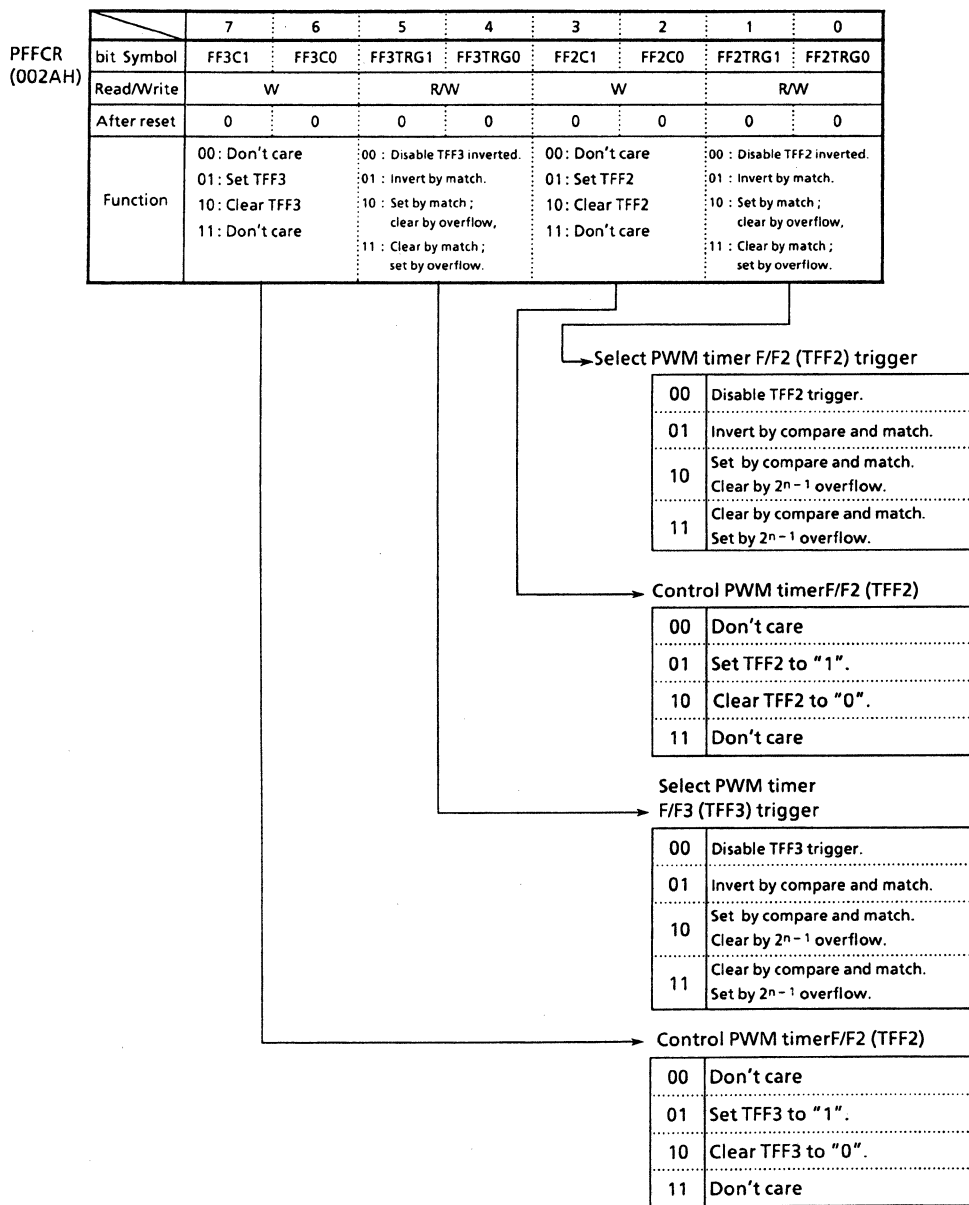


Figure 3.8 (6). 8-Bit PWM F/F Control Register

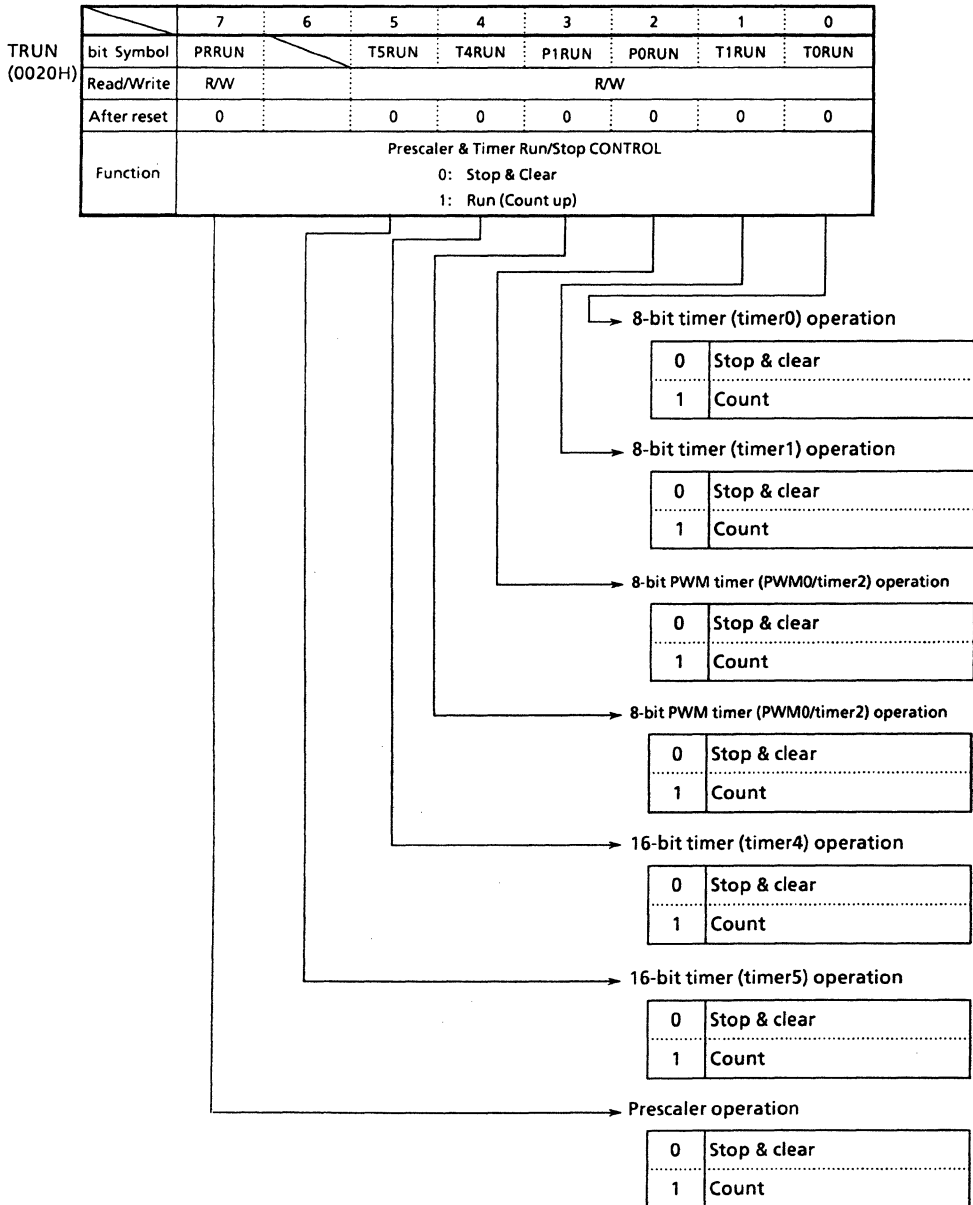


Figure 3.8 (7). Timer Operation Control Register (TRUN)

The following explains PWM timer operations.

(1) PWM timer mode

Both PWM timers can output 8-bit resolution PWM independently. Since both timers operate in exactly the same way, PWM0 is used for purposes of explanation. PWM output changes under the following two conditions.

Condition 1:

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

Condition 2:

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

The up-counter (UC2) is cleared by a  $2^n - 1$  counter overflow.

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

PWM timing

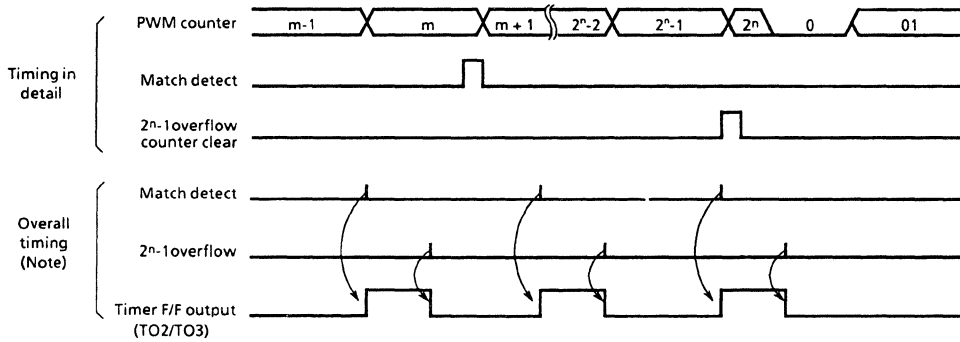


Figure 3.8 (8). Output Waves in PWM Timer Mode

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

Figure 3.8 (9) is a block diagram of this mode.

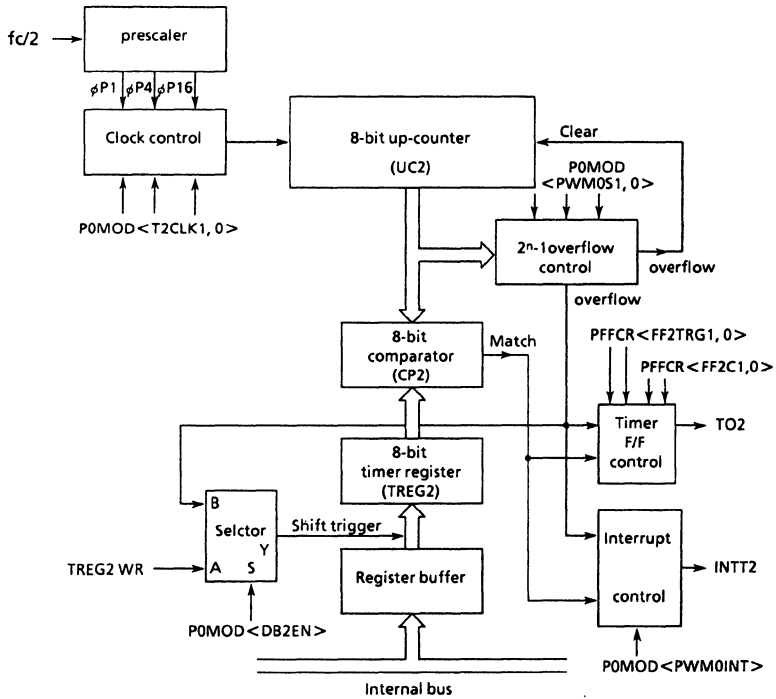


Figure 3.8 (9). Block Diagram of PWM Timer Mode (PWM0)

In this mode, enabling double buffer is very useful. The register buffer value shifts into TREG2 when a  $2^n - 1$  overflow is detected, when double buffer is enabled.

Using double buffer makes handling small duty waves easy.

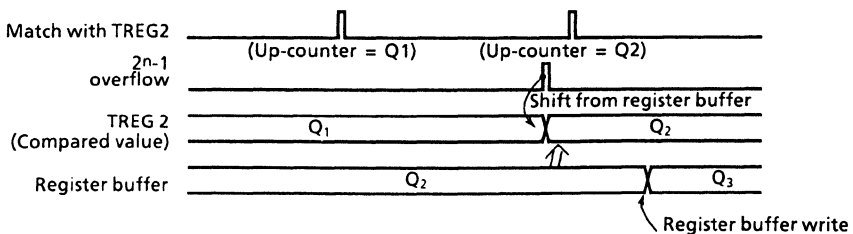
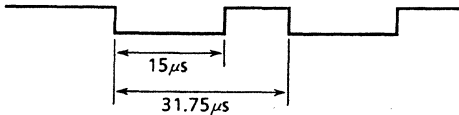


Figure 3.8 (10). Register Buffer Operation

Example: To output the following PWM waves to TO2 pin using PWM0 at  $f_c = 16\text{MHz}$ .



To implement  $31.75\mu\text{s}$  PWM cycle by  $\phi P1 = 0.25\mu\text{s}$  ( $f_c = 16\text{MHz}$ )

$$31.75\mu\text{s} + 0.25\mu\text{s} = 127 = 2^7 - 1.$$

Consequently, set n to 7.

Since the low level cycle =  $15\mu\text{s}$ ; for  $\phi P1 = 0.25\mu\text{s}$

$$15\mu\text{s} + 0.25 = 60 = 3\text{CH}$$

set the 3CH in TREG2.

	7	6	5	4	3	2	1	0	
TRUN	← -	x	-	-	-	0	-	-	Stops PWM0 and clears it to 0.
POMOD	← -	0	0	0	0	0	0	1	Sets PWM ( $2^7 - 1$ ) mode, input clock $\phi P1$ , overflow interrupt, and disables double buffer.
TREG2	← 0	0	1	1	1	1	0	0	Writes 3CH.
POMOD	← -	1	0	0	0	0	0	1	Enables double buffer.
PFFCR	← -	-	-	-	0	1	1	1	Sets TFF2 and a mode where TFF2 is set by compare and match, and cleared by overflow.
P7CR	← x	x	x	x	-	1	-	-	) Sets P72 as the TO2 pin.
P7FC	← x	x	x	x	-	1	-	x	
TRUN	← 1	x	-	-	-	1	-	-	Starts PWM0 counting.

Note: x; don't care      -; no change

Table 3.8 (1) PWM Cycle and  $2^n - 1$  Counter Setting

	Formula	16MHz			20MHz		
		$\phi P1$	$\phi P4$	$\phi P16$	$\phi P1$	$\phi P4$	$\phi P16$
$2^6 - 1$	$2^6 - 1 - \phi Pn$	15.8μsec (63kHz)	63.0μsec (16kHz)	252μsec (3.9kHz)	12.6μsec (79kHz)	50.4μsec (20kHz)	201μsec (4.9kHz)
$2^7 - 1$	$2^7 - 1 - \phi Pn$	31.8μsec (31kHz)	127.0μsec (7.9kHz)	508μsec (1.9kHz)	25.4μsec (39kHz)	101.6μsec (9.8kHz)	406μsec (2.5kHz)
$2^8 - 1$	$2^8 - 1 - \phi Pn$	63.8μsec (16kHz)	255.0μsec (3.9kHz)	1020μsec (0.98kHz)	51.0μsec (20kHz)	204.0μsec (4.9kHz)	816μsec (1.2kHz)



(2) 8-bit timer mode

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

① Generating interrupts at a fixed interval

To generate timer 2 interrupt (INTT2) at a fixed interval using PWM0 timer, first stop PWM0, then set the operating mode, input clock, and interval in the P0MOD and TREG2 registers. Next, enable INTT2 and start counting PWM0.

Example: To generate a timer 2 interrupt every 40μs at fc = 16MHz, set registers as follows:

		7	6	5	4	3	2	1	0
TRUN	←	-	x	-	-	-	0	-	-
P0MOD	←	x	0	1	1	0	0	x	x
TREG2	←	1	0	1	0	0	0	0	0
INTEPW10	←	-	-	-	-	1	1	0	0
TRUN	←	1	x	-	-	-	1	-	-

- Stops PWM0 and clears it to 0.
- Sets 8-bit timer mode and selects φP1 (0.25μs) and compare interrupt.
- Sets 40μs/0.25μs = A0H in timer register.
- Enables INTT2 and sets interrupt level 4.
- Starts PWM0 counting.

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

Select an input clock using the table below.

**Table 3.8 (2) Interrupt Cycle and Input Clock Selection using 8-Bit Timer Mode**

Input Clock	Interrupt Cycle (at fc = 16MHz)	Resolution	Interrupt Cycle (at fc = 20MHz)	Resolution
φP1 (4/fc)	0.25μs ~ 64μs	0.25μs	0.2μs ~ 51.2μs	0.2μs
φP4 (16/fc)	1μs ~ 256μs	1μs	0.8μs ~ 204.8μs	0.8μs
φP16 (64/fc)	4μs ~ 1024μs	4μs	3.2μs ~ 819.2μs	3.2μs

Note: To generate interrupts in 8-bit timer mode, bit 5 (interrupt control bit <PWM01NT>/<PWM1NT> of P0MOD/P1MOD) must be set to 1.

② Generating a 50% square wave

value to the timer output pin (TO2).

To generate a 50% square wave, invert the timer flip-flop at a fixed interval and output the timer flip-flop

Example: To output a 3.0μs square wave at  $f_c = 16\text{MHz}$  from TO2 pin, set register as fol-

lows:

		7	6	5	4	3	2	1	0
TRUN	←	-	x	-	-	-	0	-	-
POMOD	←	x	0	1	1	0	0	x	x
TREG2	←	0	0	0	0	0	1	1	0
PFFCR	←	-	-	-	-	1	0	0	1
P7CR	←	x	x	x	x	-	1	-	-
P7FC	←	x	x	x	x	-	1	-	x
TRUN	←	1	x	-	-	-	1	-	-

Stops PWM0 and clears it to 0.

Sets 8-bit timer mode and selects  $\phi P1$  (0.25μs) as the input clock.

Sets  $3.0\mu\text{s}/0.25\mu\text{s}/2 = 6$  in the timer register.

Clears TFF2 to 0 and inverts using comparator output.

) Sets P72 as the TO2 pin.

Note: x; don't care    -; no change

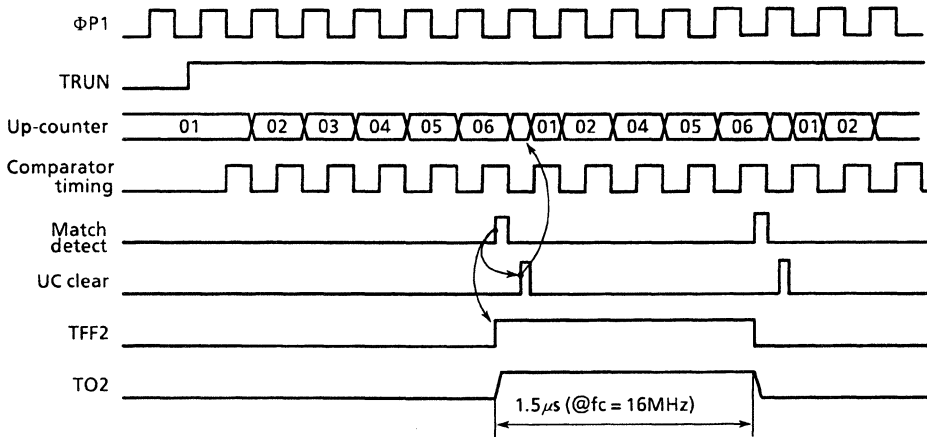


Figure 3.8 (11). Square Wave (50% Duty) Output Timing Chart

This mode is as shown in Figure 3.8 (12) below.

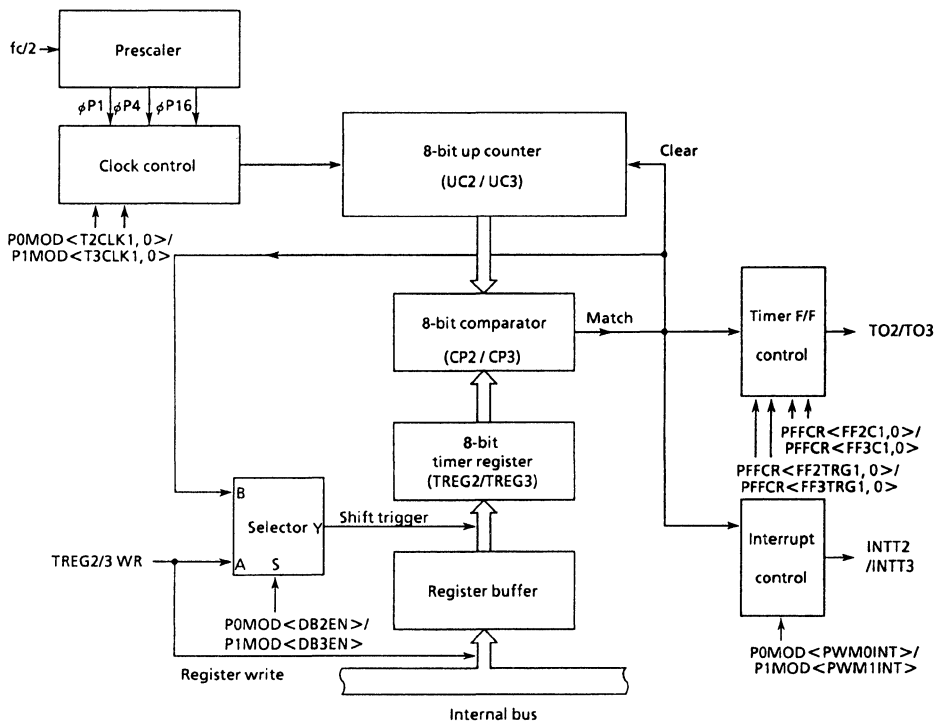


Figure 3.8 (12). Block Diagram of 8-Bit Timer Mode

### 3.9 16-Bit Timer

The TMP96C141AF has two (timer 4 and timer 5) multifunctional 16-bit timer/event counter with the following operation modes.

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

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

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

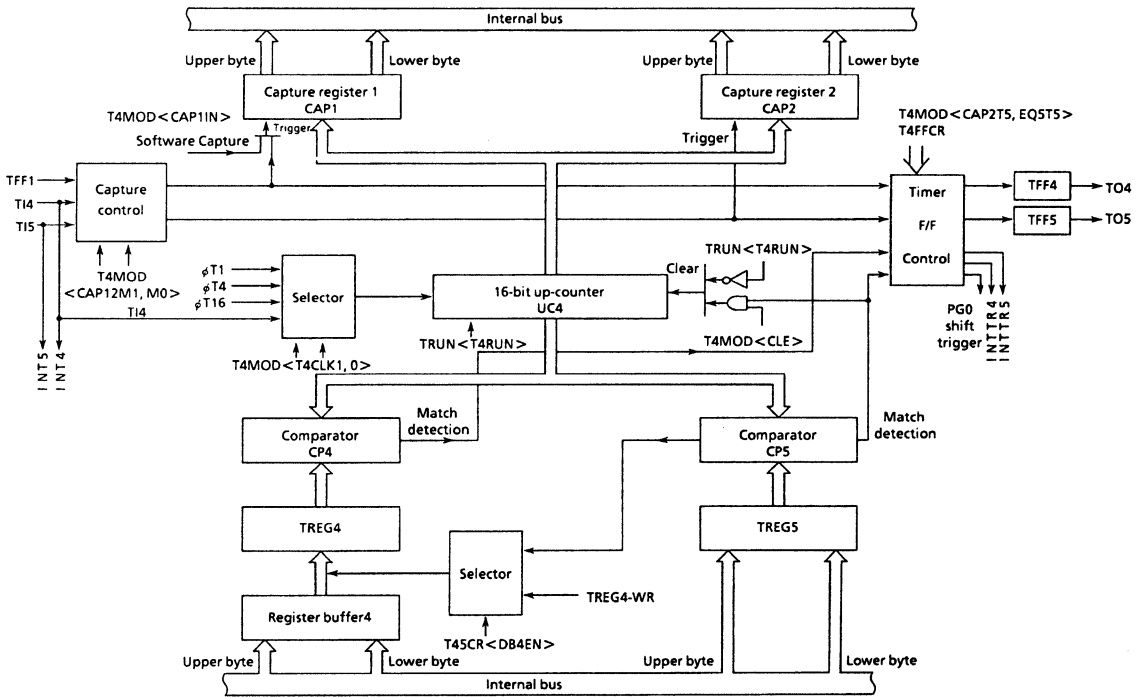


Figure 3.9 (1). Block Diagram of 16-Bit Timer (Timer 4)

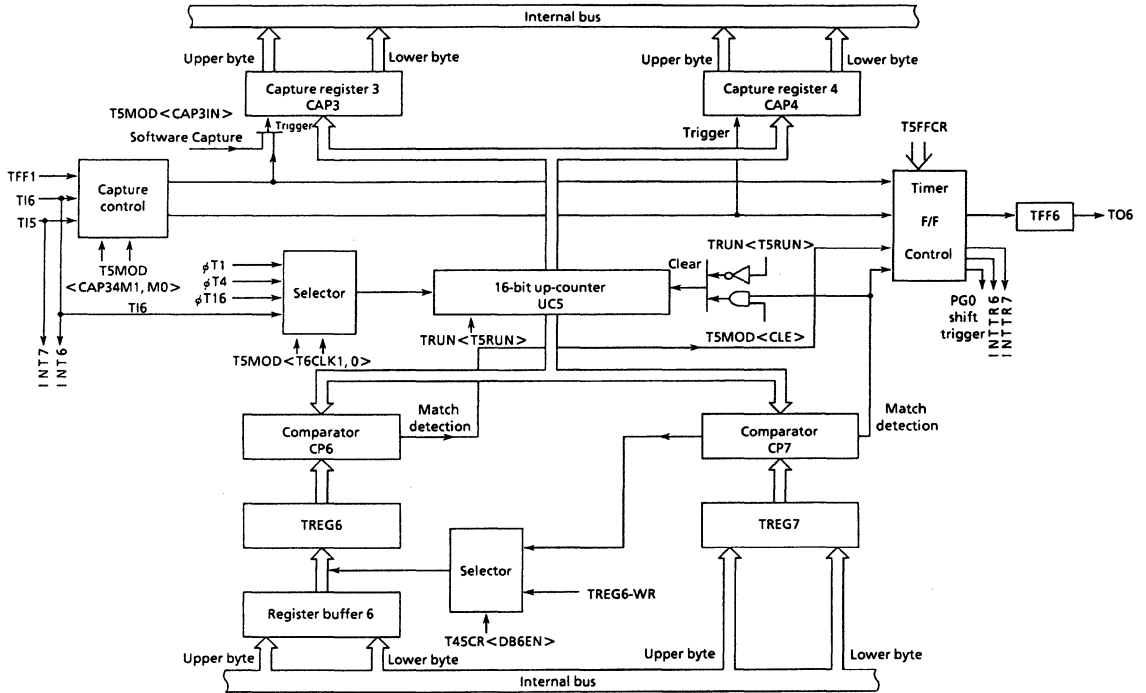


Figure 3.9 (2). Block Diagram of 16-Bit Timer (Timer 5)

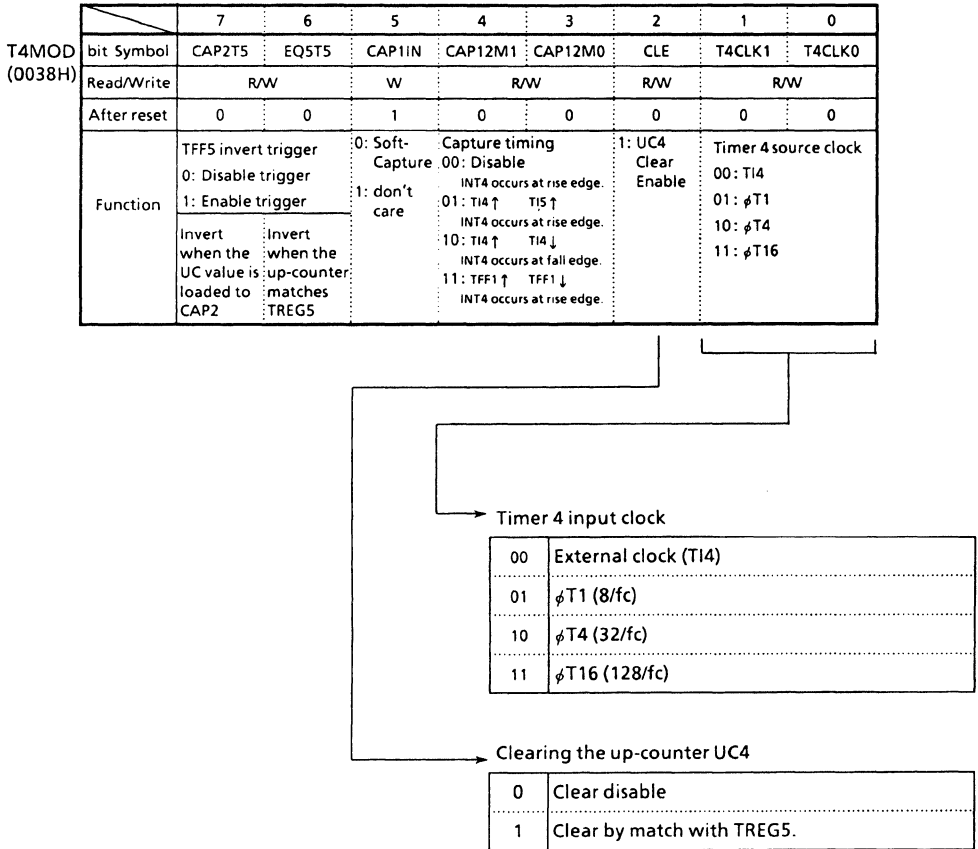
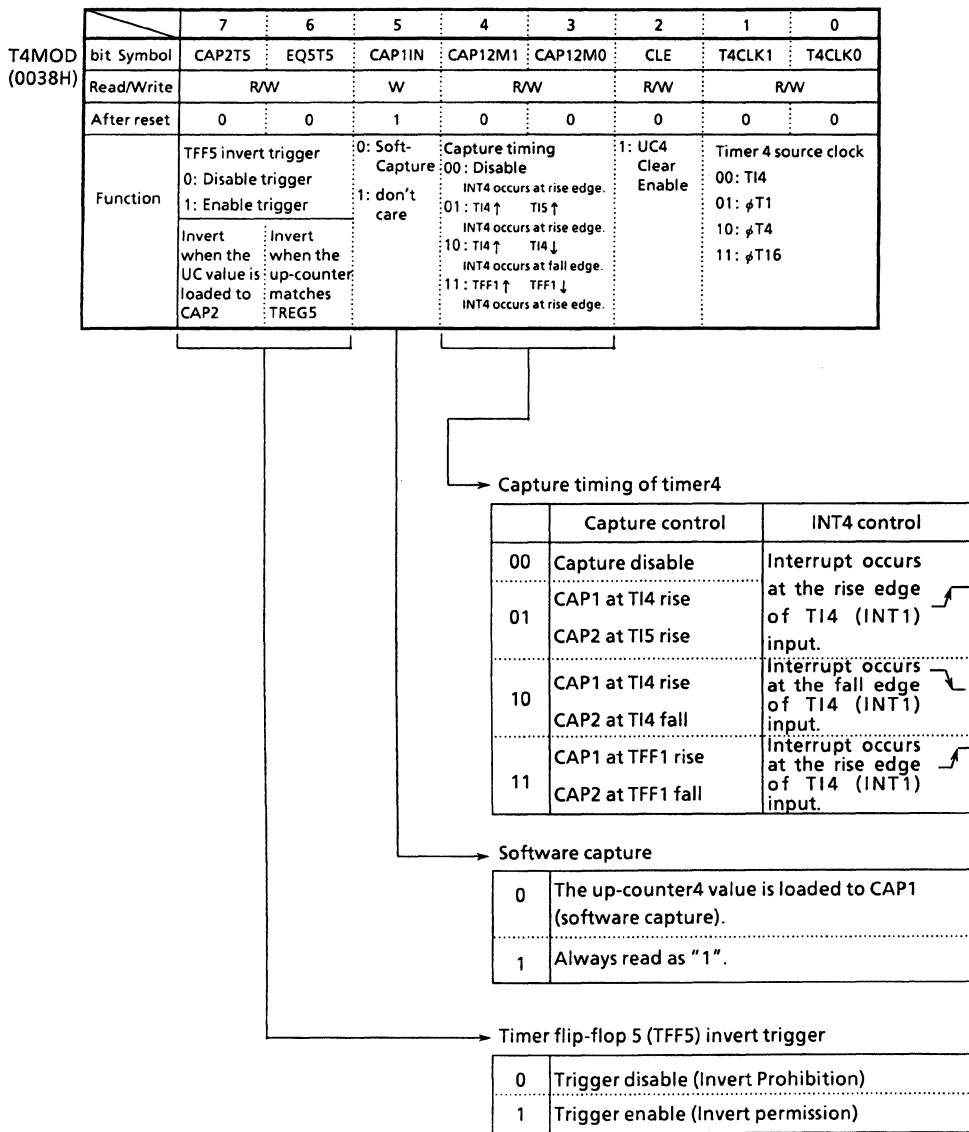


Figure 3.9 (3). 16-Bit Timer Mode Controller Register (T4MOD) (1/2)



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

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



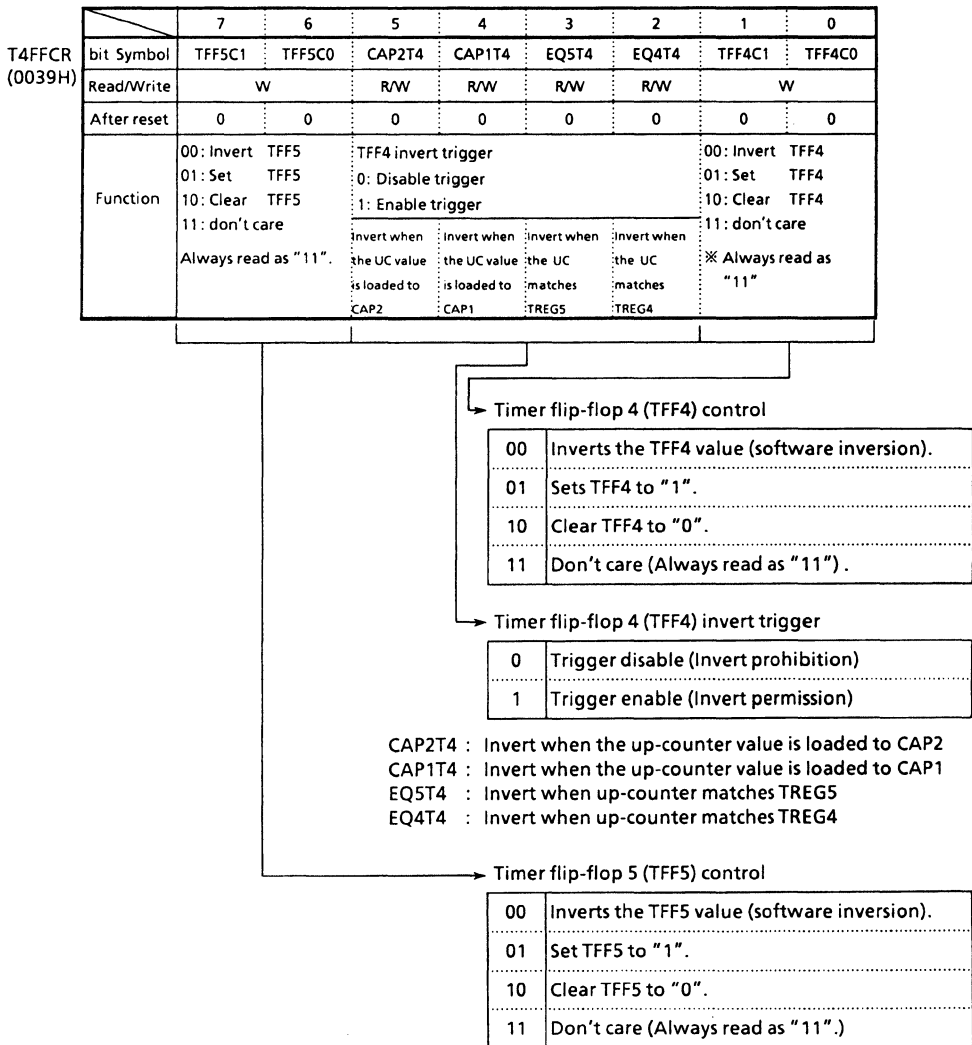


Figure 3.9 (5). 16-Bit Timer 4 F/F Control (T4FFCR)

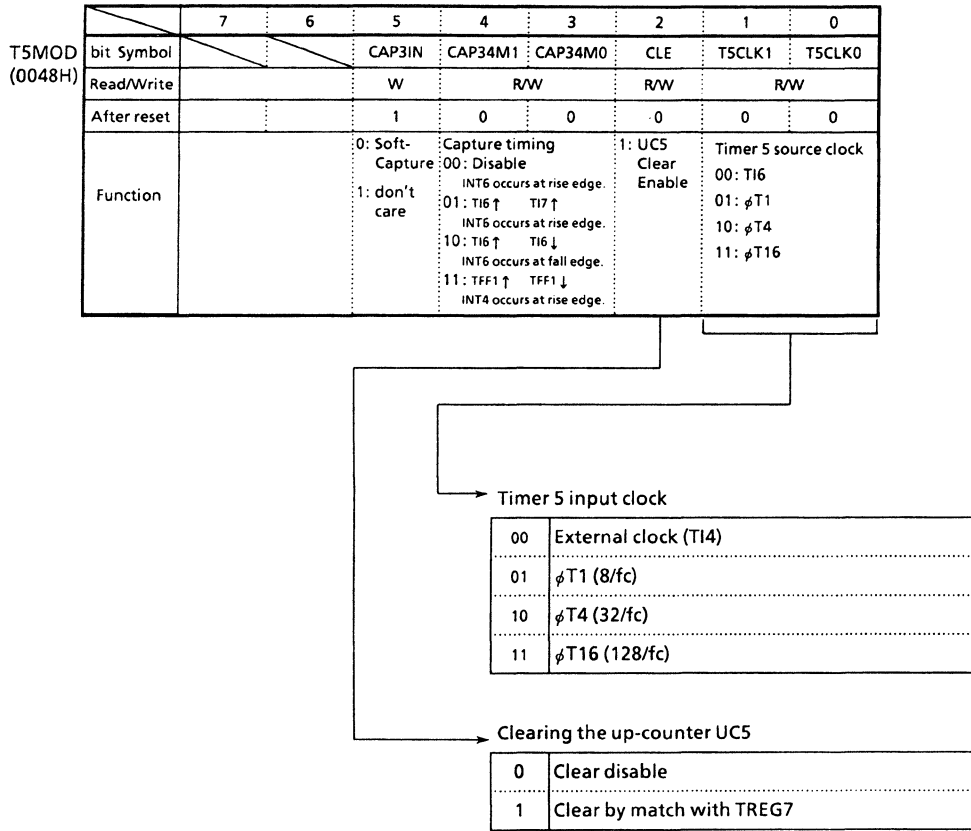


Figure 3.9 (6). 16-Bit Timer Mode Control Register (T5MOD) (1/2)

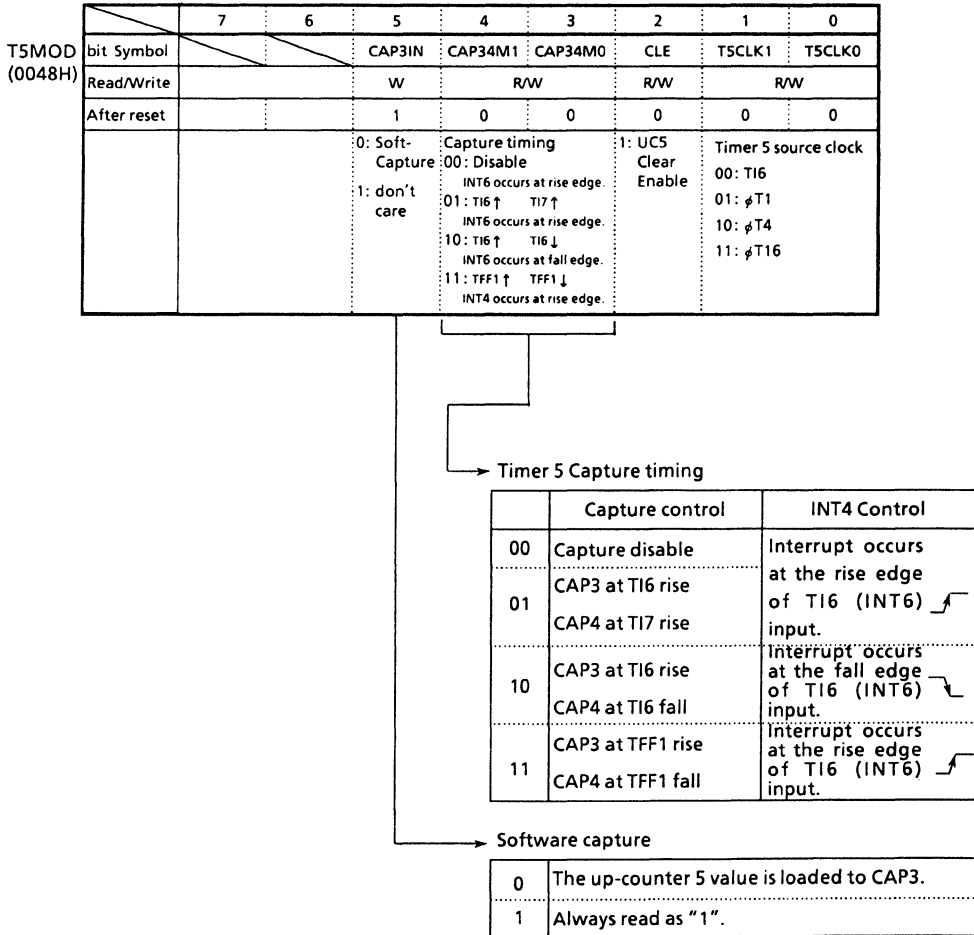
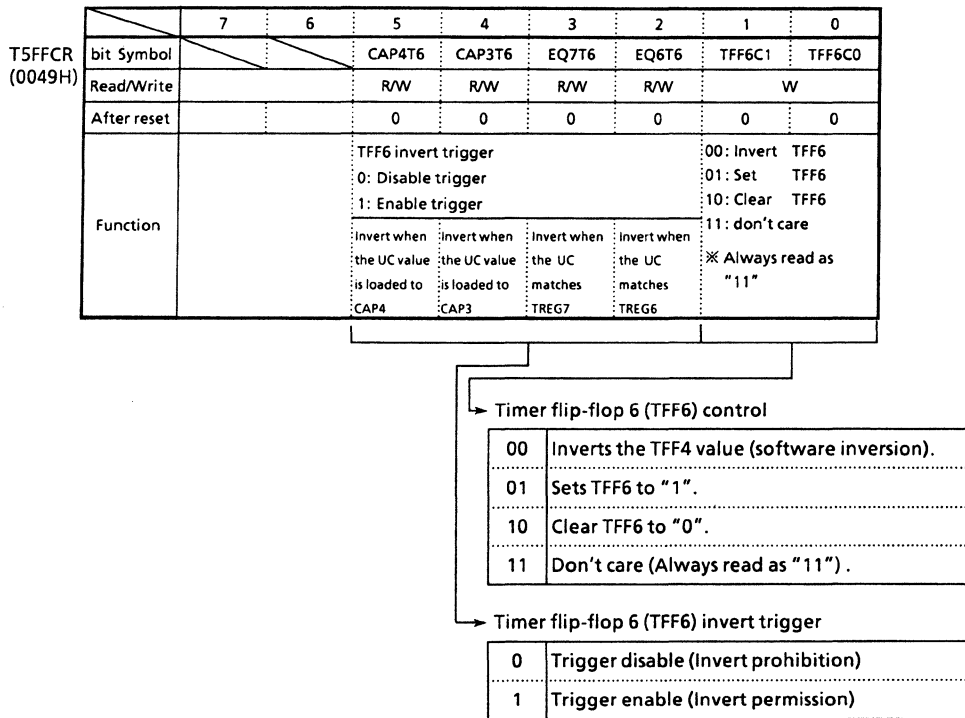


Figure 3.9 (7). 16-Bit Timer Control Register (T5MOD) (2/2)



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

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

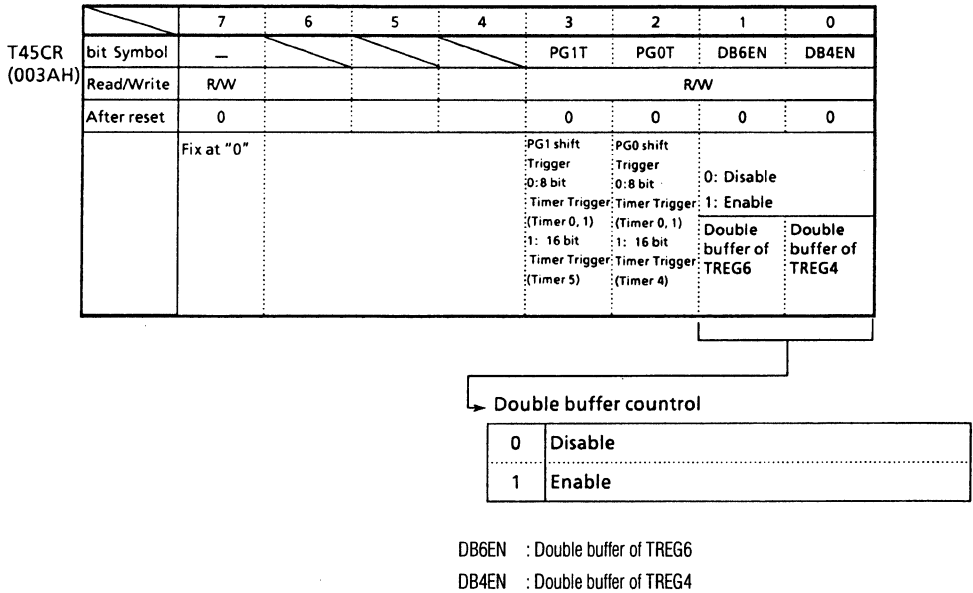


Figure 3.9 (9). 16-Bit Timer (Timer 4, 5) Control Register (T45CR)

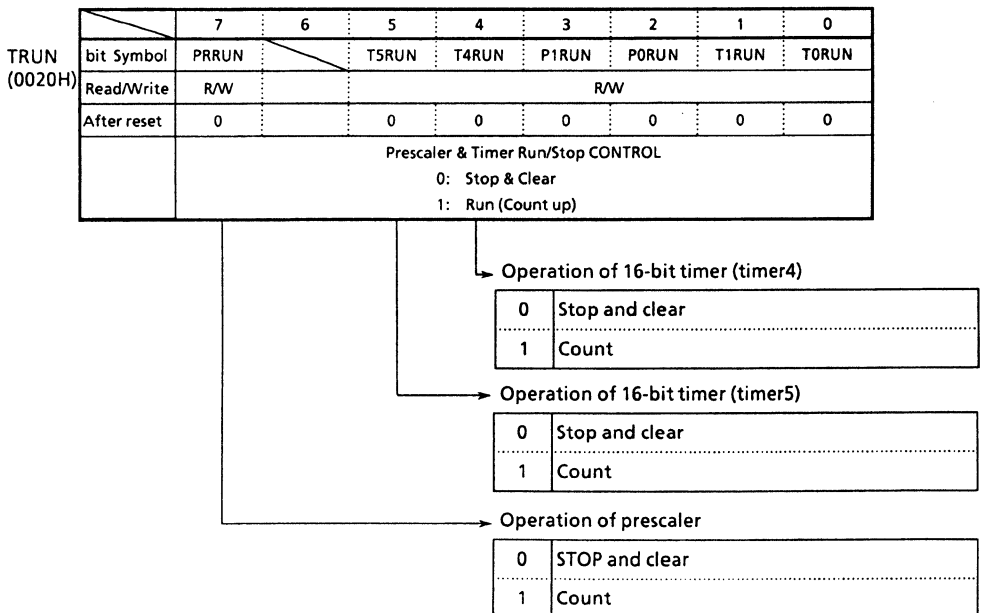


Figure 3.9 (10). Timer Operation Control Register (TRUN)

① Up-counter (UC4/UC5)

UC4/UC5 is a 16-bit binary counter which counts up according to the input clock specified by T4MOD <T4CLK1, 0> or T5MOD <T5CLK1, 0> register.

As the input clock, one of the internal clocks  $\phi$  T1 (8/fc),  $\phi$  T4 (32/fc), and  $\phi$  T16 (128/fc) from 9-bit prescaler (also used for 8-bit timer), and external clock from TI4 pin (also used as P80/INT4 pin) or TI6 (also used as P84/INT6 pin) can be selected. When reset, it will be initialized to <T4CLK1, 0>/<T5CLK1, 0> = 00 to select TI4/TI6 input mode. Counting or stop and clear of the counter is controlled by timer operation control register TRUN <T4RUN, T5RUN>.

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

timer register TREG5, TREG7. The "clear enable/disable" is set by T4MOD <CLE> and T5MOD <CLE>.

If clearing is disabled, the counter operates as a free-running counter.

② Timer Registers

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

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

TREG4

Upper 8 bits	Lower 8 bits
000031H	000030H

TREG5

Upper 8 bits	Lower 8 bits
000033H	000032H

TREG6

Upper 8 bits	Lower 8 bits
000041H	000040H

TREG7

Upper 8 bits	Lower 8 bits
000043H	000042H

TREG4 and TREG6 timer register is of double buffer structure, which is paired with register buffer. The timer control register T45CR <DB4EN, DB6EN> controls whether the double buffer structure should be enabled or disabled. : disabled when <DB4EN, DB6EN> = 0, while enabled when <DB4EN, DB6EN> = 1.

When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up-counter (UC4/UC5) and timer register TREG5/TREG7.

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

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

③ Capture Register

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

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

CAP 1

Upper 8 bits	Lower 8 bits
000035H	000034H

CAP 2

Upper 8 bits	Lower 8 bits
000037H	000036H

CAP 3

Upper 8 bits	Lower 8 bits
000045H	000044H

CAP 4

Upper 8 bits	Lower 8 bits
000047H	000046H

④ Capture Input Control

This circuit controls the timing to latch the value of up-counter UC4/UC5 into (CAP1, CAP2)/(CAP3, CAP4).

The latch timing of capture register is controlled by register T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0>.

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

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

- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 01  
Data is loaded to CAP1, CAP3 at the rise edge of TI4 pin (also used as P80/INT4) and TI6 pin (also used as P84/INT6) input, while data is loaded to CAP2, CAP4 at the rise edge of TI5 pin (also used as P81/INT5 and TI7 pin (also used as P85/INT7) input. (Time difference measurement)
- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 10  
Data is loaded to CAP1 at the rise edge of TI4 pin input and to CAP3 at the rise edge of TI6, while to CAP2, CAP4 at the fall edge. Only in this setting, interrupt INT4/INT6 occurs at fall edge. (Pulse width measurement)
- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 11  
Data is loaded to CAP1, CAP3 at the rise edge of timer flip-flop TFF1, while to CAP2, CAP4 at the fall edge.  
Besides, the value of up-counter can be loaded to capture registers by software. Whenever "0" is written in T4MOD <CAPIN>, T5MOD <CAP3IN> the current value of up-counter will be loaded to capture register CAP1/CAP3. It is necessary to keep the prescaler in RUN mode (TRUN <PRRUN> to be "1").

⑤ Comparator

These are 16-bit comparators which compare the up-counter UC4/UC5 value with the set value of (TREG4, TREG5)/(TREG6, TREG7) to detect the match. When a match is detected, the comparators generate an interrupt (INTT4, INTT5)/(INTT6, INTT7) respectively. The up-counter UC4/UC5 is cleared only when UC4/UC5

		7	6	5	4	3	2	1	0	
TRUN	←	-	x	-	0	-	-	-	-	
INTET54	←	1	1	0	0	1	0	0	0	
T4FFCR	←	1	1	0	0	0	0	1	1	
T4MOD	←	0	0	1	0	0	1	*	*	
					(** = 01, 10, 11)					
TREG5	←	*	*	*	*	*	*	*	*	
		*	*	*	*	*	*	*	*	
TRUN	←	1	x	-	1	-	-	-	-	

Note: x; don't care      -; no change

(2) 16-bit Event Counter Mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TI4/ TI6 pin input) as the input clock. To read the value of the

matches TREG5/TREG7. (The clearing of up-counter UC4/UC5 can be disabled by setting T4MOD <CLE>/ T5MOD <CLE> = 0.)

⑥ Timer Flip-Flop (TFF4/TFF6)

This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable/enable of inversion can be set for each element by T4FFCR <CAP2T4, CAP1T4, EQ5T4, EQ4T4>/T6FFCR <CAP4T6, CAP3T6, EQ7T6, EQ6T6>. TFF4/TFF6 will be inverted when "00" is written in T4FFCR <TFF4C1, 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 TFF4/TFF6 can be output to the timer output pin TO4 (also used as P82) and TO6 (also used as P86).

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).

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

(1) 16-bit Timer Mode

Timer 4 and 5 operate independently.

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

Generating interrupts at fixed intervals:

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

Stop timer 4.

Enable INTTR5 and sets interrupt level 4. Disables

INTTR4.

Disable trigger.

Select internal clock for input and disable the capture function.

Set the interval timer (16 bits).

Start timer 4.

counter, first perform "software capture" once and read the captured value.

The counter counts at the rise edge of TI4/TI6 pin input. TI4/TI6 pin can also be used as P80/INT4 and P84/INT6.

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

	7	6	5	4	3	2	1	0		
TRUN	←	-	x	-	0	-	-	-	Stop timer 4.	
P8CR	←	-	-	-	-	-	-	0	Set P80 to input mode.	
INTET54	←	1	1	0	0	1	0	0	Enable INTTR5 and sets interrupt level 4, while disables INTTR4.	
T4FFCR	←	1	1	0	0	0	0	1	1	Disable trigger.
T4MOD	←	0	0	1	0	0	1	0	0	Select T14 as the input clock.
TREG5	←	*	*	*	*	*	*	*	Set the number of counts (16 bits).	
TRUN	←	1	x	-	1	-	-	-	Start timer 4.	

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

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

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

The PPG mode is obtained by inversion of the timer

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

$$(\text{Set value of TREG4}) < (\text{Set value of TREG5})$$

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

Note: x; don't care    -; no change

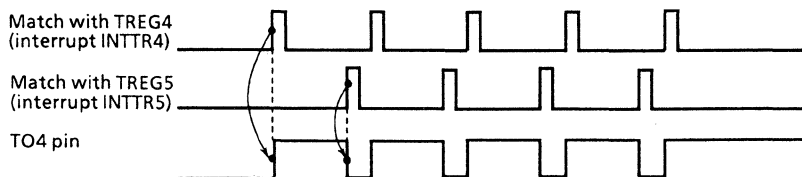


Figure 3.9 (11). Programmable Pulse Generation (PPG) Output Waveforms



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

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

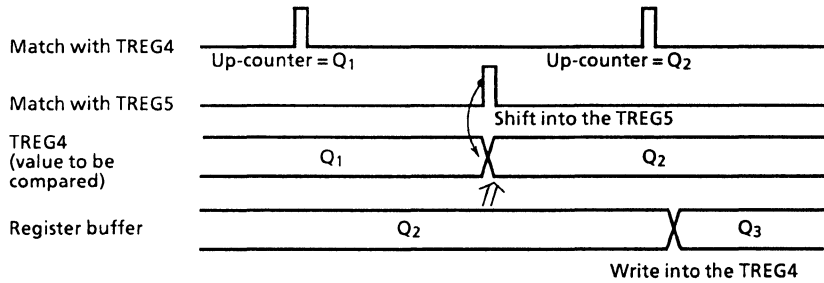


Figure 3.9 (12). Operation of Register Buffer

Shows the block diagram of this mode.

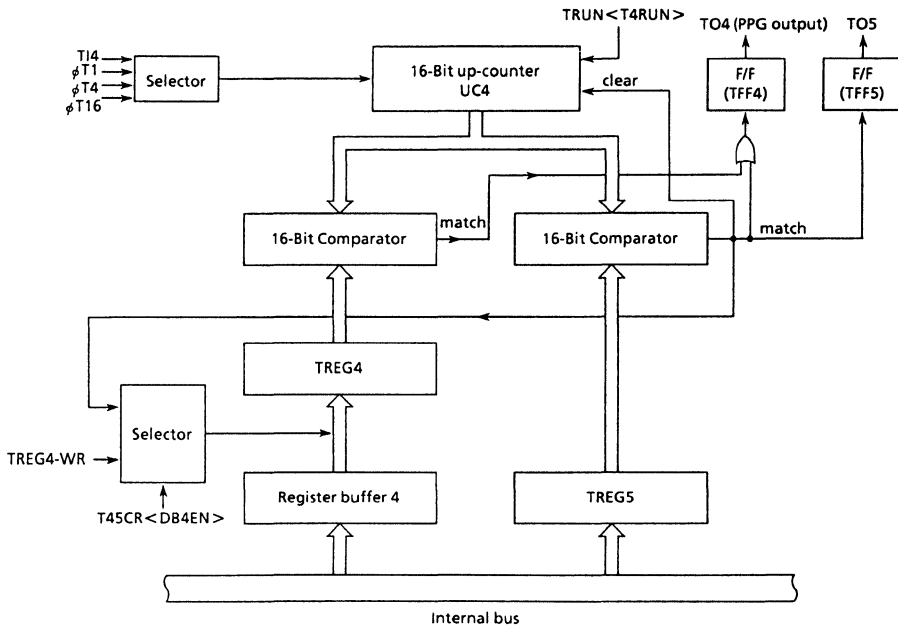


Figure 3.9 (13). Block Diagram of 16-Bit PPG Mode

(4) Application Examples of Capture Function

① One-Shot Pulse Output from External Trigger Pulse

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 TFF4 status to TO4 pin can be enabled or disabled. Combined with interrupt function, they can be applied in many ways, for example:

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

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

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

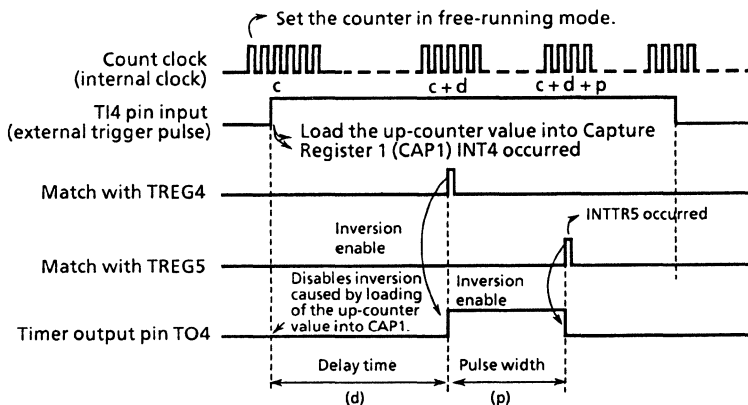
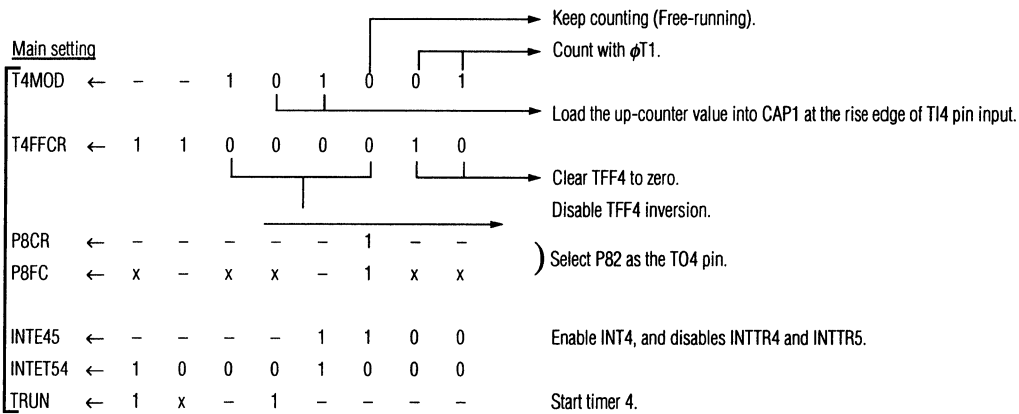
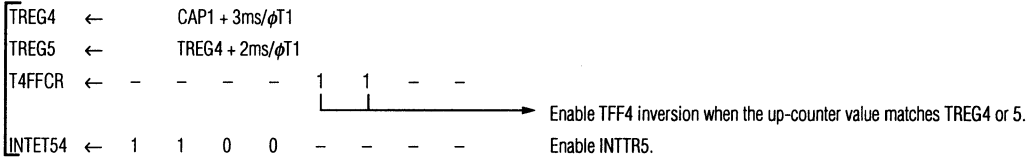


Figure 3.9 (14). One-Shot Pulse Output (with Delay)

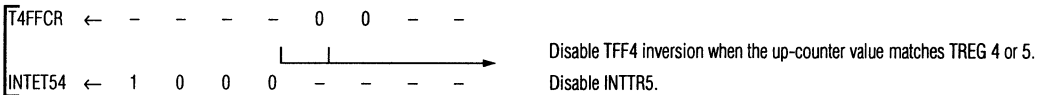
Setting Example: To output 2ms one-shot pulse with 3ms delay to the external trigger pulse to TI4 pin.



**Setting of INT4**



**Setting of INT5**



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

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

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

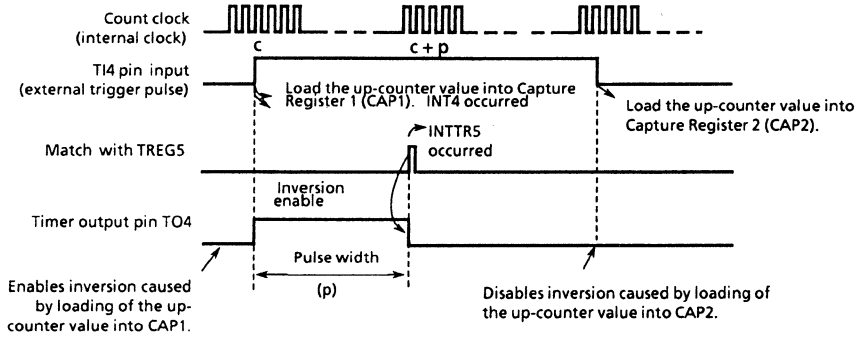


Figure 3.9 (15). One-Shot Pulse Output (without Delay)

② Frequency Measurement

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

The TI4 pin input should be selected for the input clock of Timer 4. The value of the up-counter is loaded

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

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

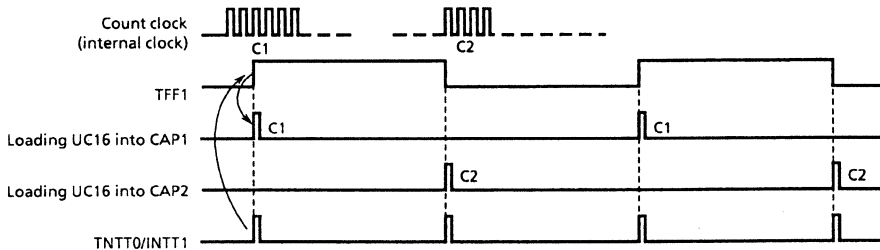


Figure 3.9 (16). Frequency Measurement

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

ence between CAP1 and CAP2 is 100, the frequency will be  $100/0.5 \text{ [sec.]} = 200 \text{ [Hz]}$ .

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  $100 \times 0.8 = 80$  microseconds.

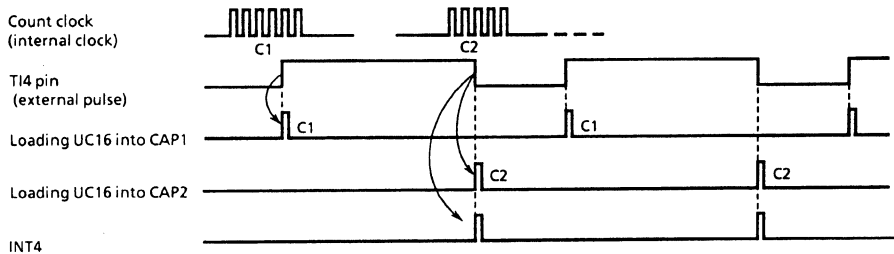


Figure 3.9 (17). Pulse Width Measurement

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

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

④ Time Difference Measurement

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

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

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

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

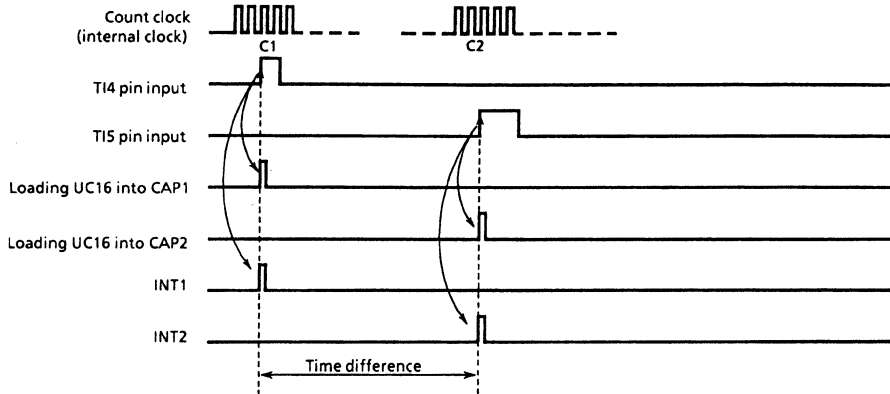


Figure 3.9 (18). Time Difference Measurement

(5) Different Phased Pulses Output Mode

In this 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.

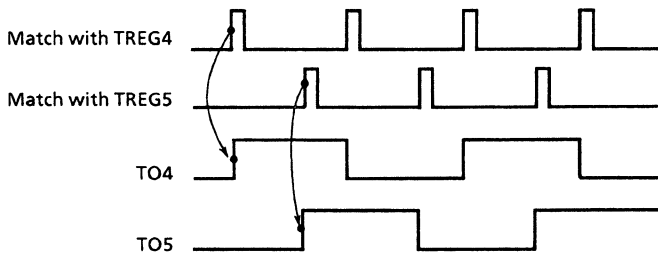


Figure 3.9 (19). Phase Output

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

	16MHz	20MHz
$\phi T1$	1.024msec	0.819msec
$\phi T4$	4.096msec	3.277msec
$\phi T16$	16.38 msec	13.11 msec

**3.10 Stepping Motor Control/Pattern Generation Port**

The TMP96C141AF has two channels (PG0 and PG1) of 4-bit hardware stepping motor control/pattern generation (herein after called PG) which actuate in synchronization with the (8-bit/16-bit) timers. The PG (PG0 and PG1) are shared in 8-bit I/O ports P6.

Channel 0 (PG0) is synchronous with 8-bit timer 0 or timer 1, 16-bit timer 5, to update the output.

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

the PG port.

PG0 and PG1 can be used independently.

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

Differences between PG0 and PG1

	PG0	PG1
Trigger Signal	from Timer 4	from Timer 5

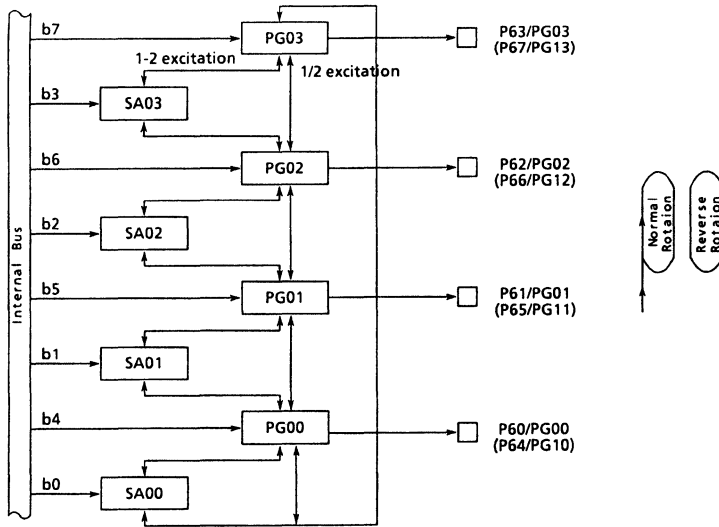


Figure 3.10 (1). Port 6/PG Circuit

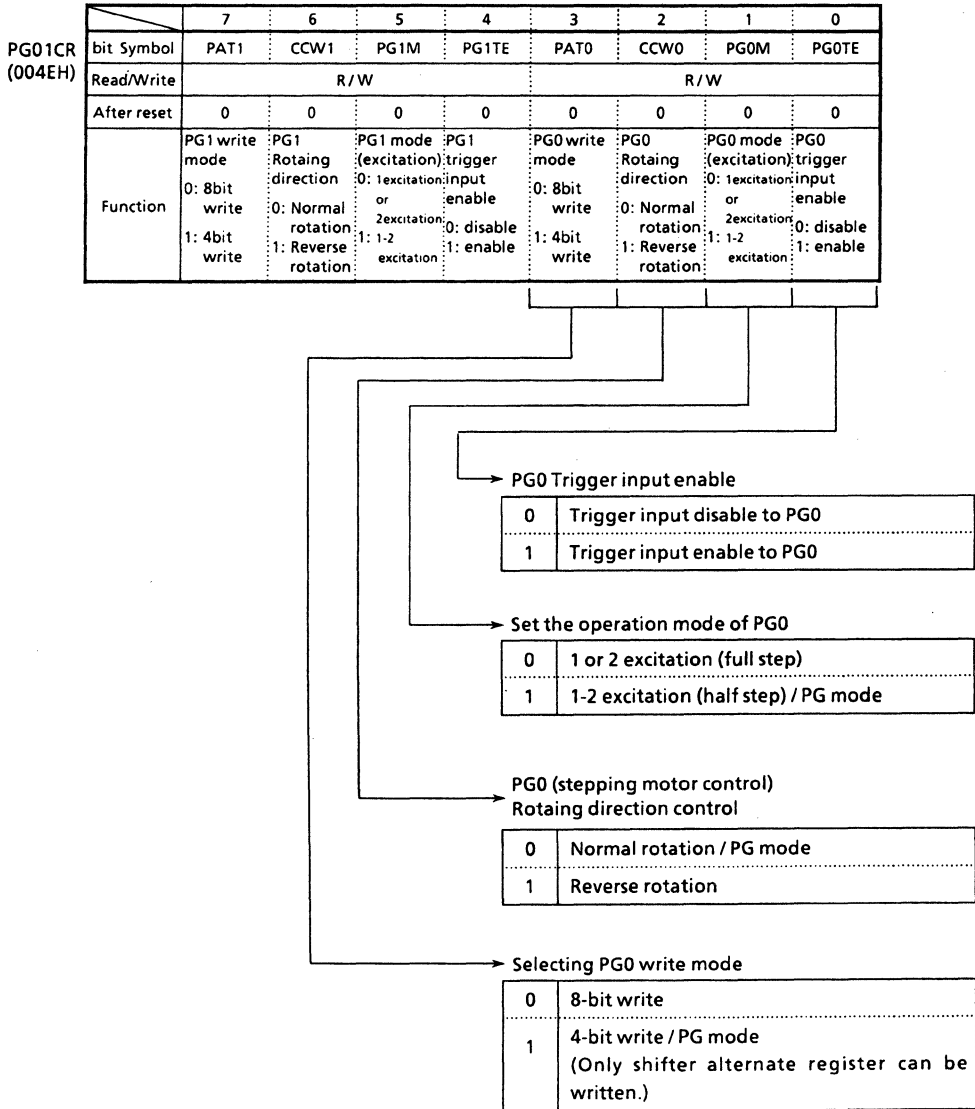


Figure 3.10 (2a). Pattern Generation Control Register (PG01CR)



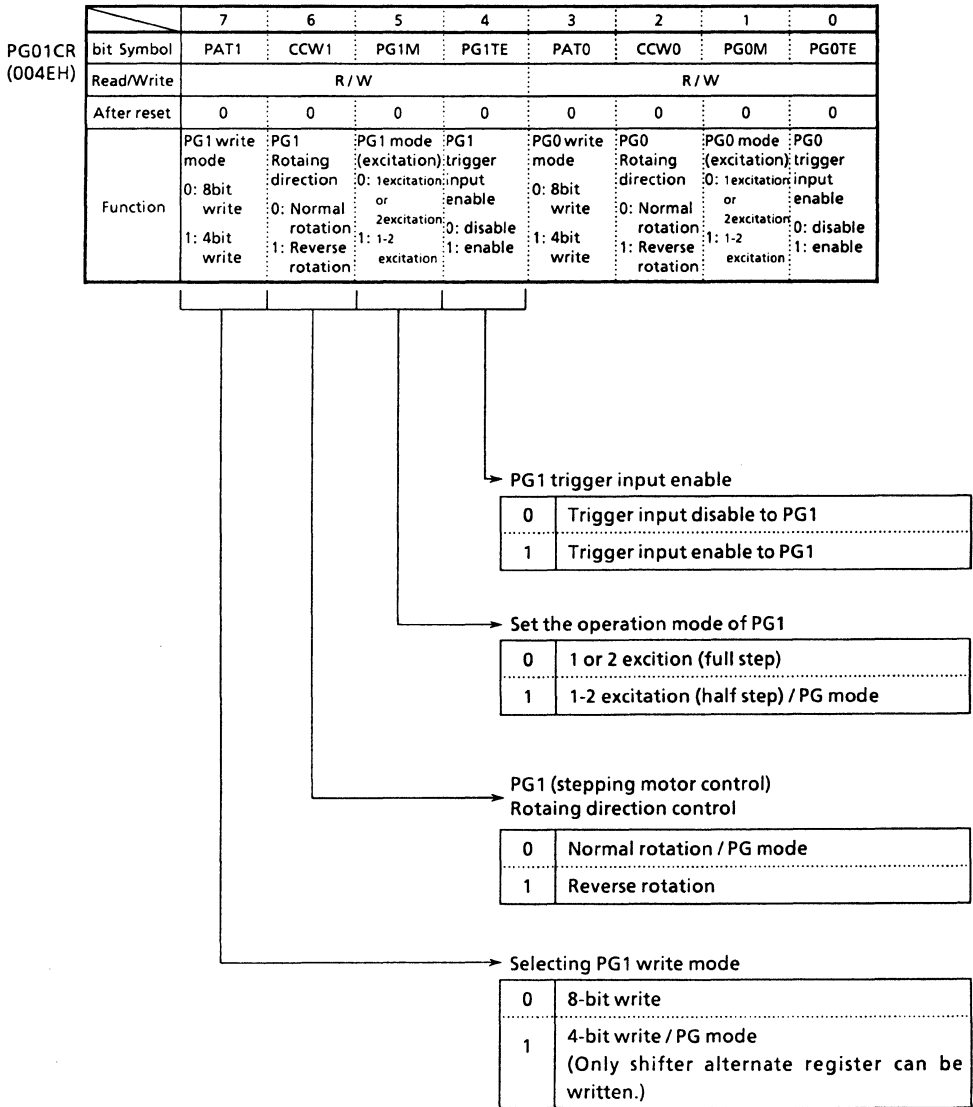


Figure 3.10 (2b). Pattern Generation Control Register (PG01CR)

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
PG0REG (004CH)	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 3.10 (3). Pattern Generation 0 Register (PG0REG)**

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
PG1REG (004DH)	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 3.10 (4). Pattern Generation 1 Register (PG1REG)**

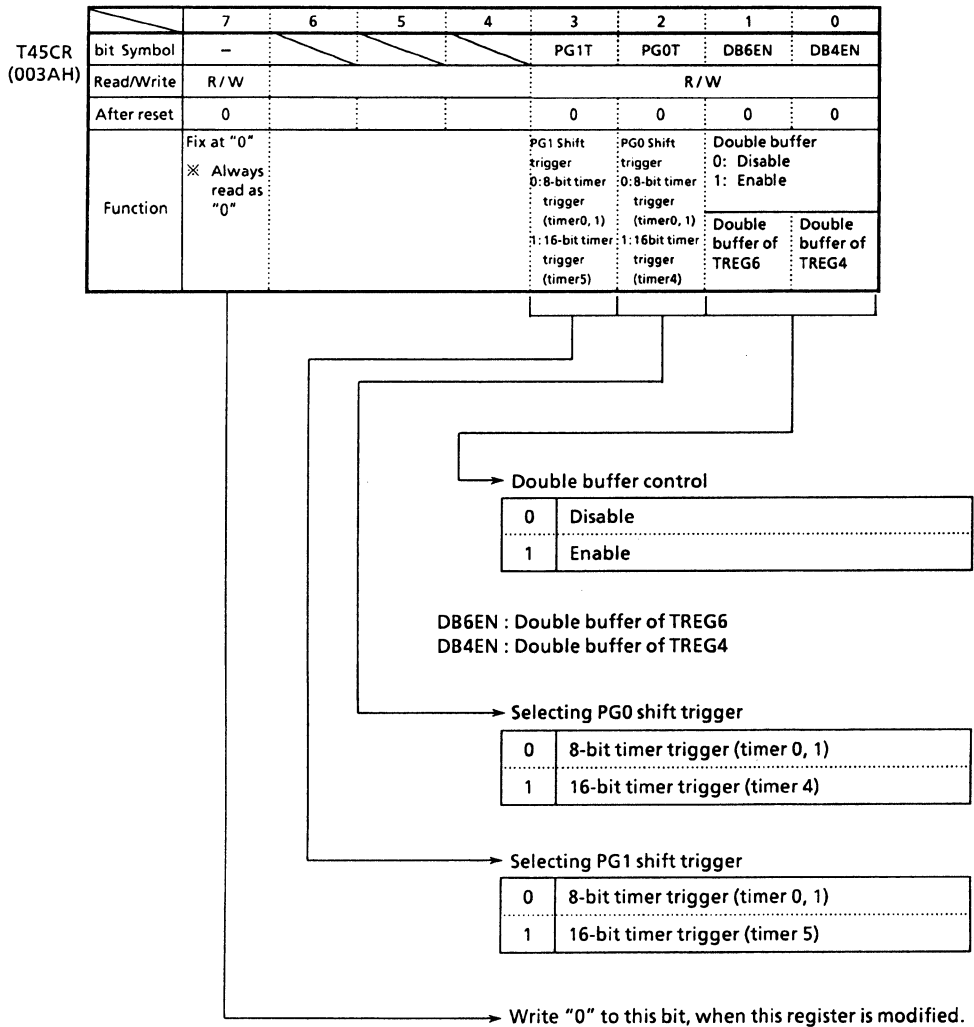


Figure 3.10 (5). 16-bit Timer Trigger Control Register (T45CR)

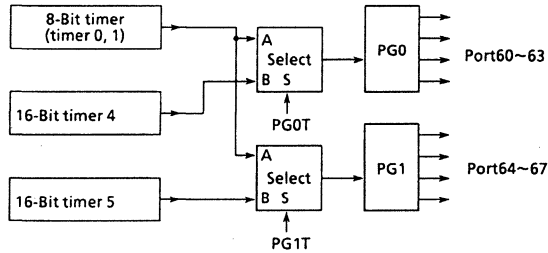


Figure 3.10 (6). Connection of Timer and Pattern Generator

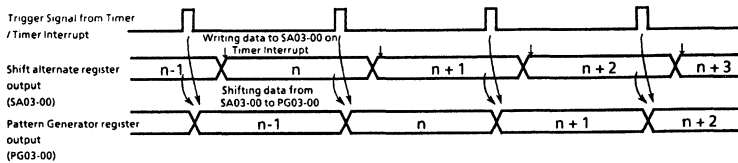
(1) Pattern Generation Mode

PG functions as a pattern generation according to the setting of PG01CR <PAT1>/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 P6FC, any port pin can be assigned to pattern generator output.

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



Example of pattern generation mode

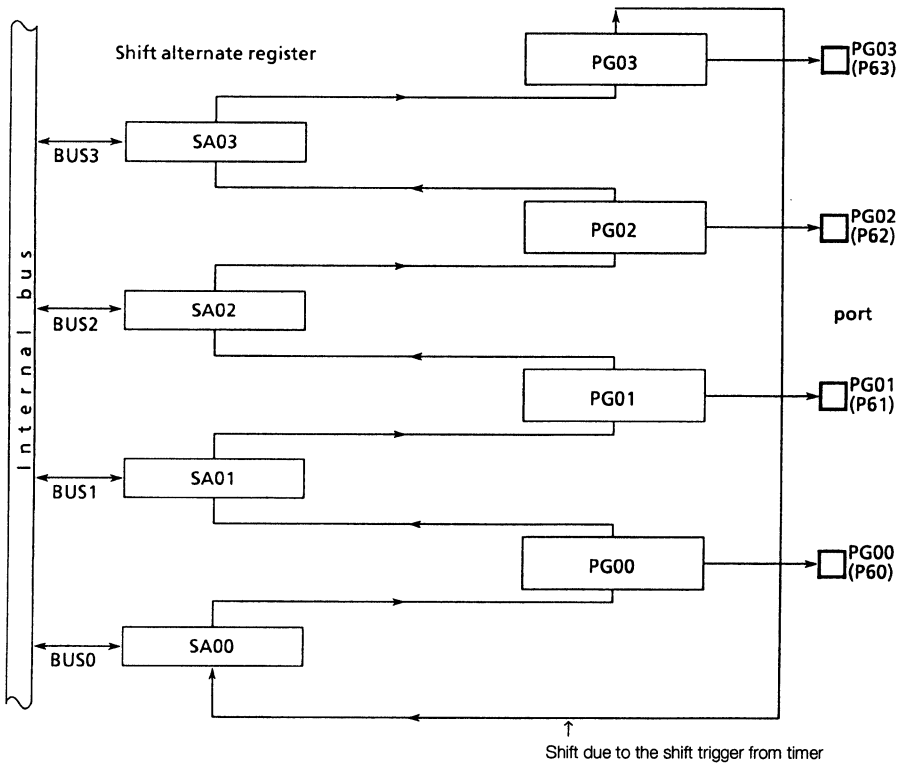


Figure 3.10 (7). Pattern Generation Mode Block Diagram (PG0)

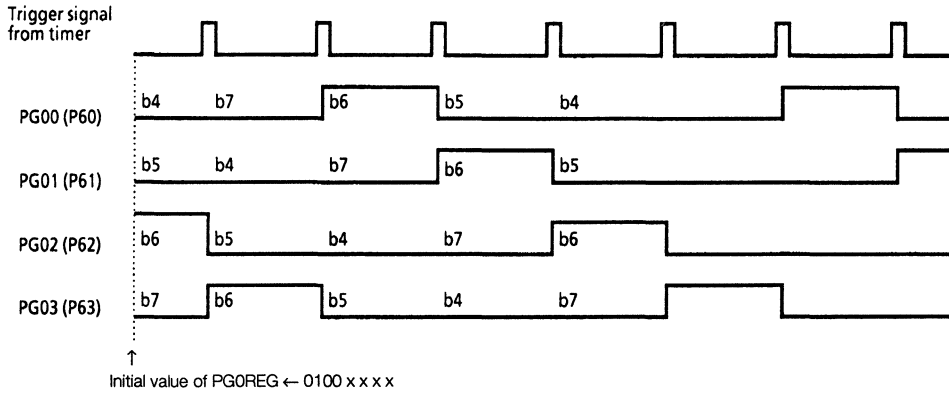
In this pattern generation mode, only writing the output latch is disabled by hardware, but other functions do the same operation as 1-2 excitation in stepping motor control port

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

(2) Stepping Motor Control Mode

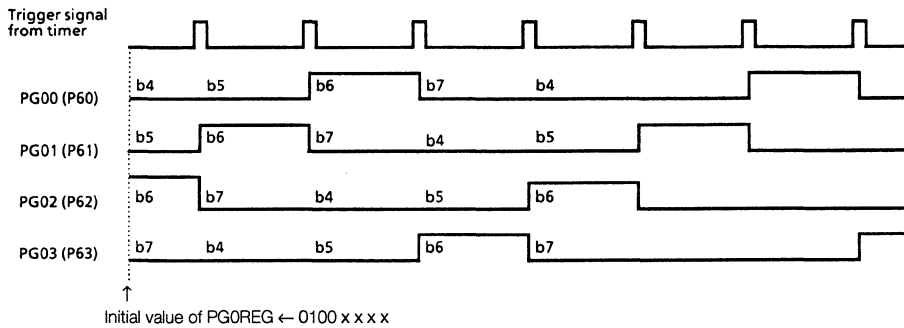
① 4-phase 1-Step/2-Step Excitation

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



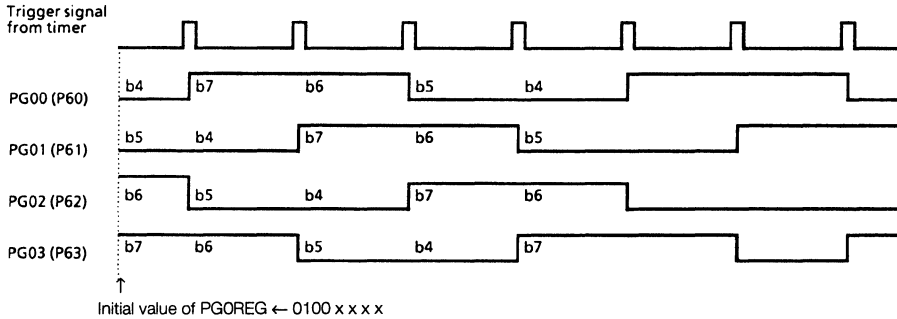
Note: bn indicates the initial value of PGOREG ← b7 b6 b5 b4 x x x x

① Normal Rotation



② Reverse Rotation

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



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

The operation when channel 0 is selected is explained below.

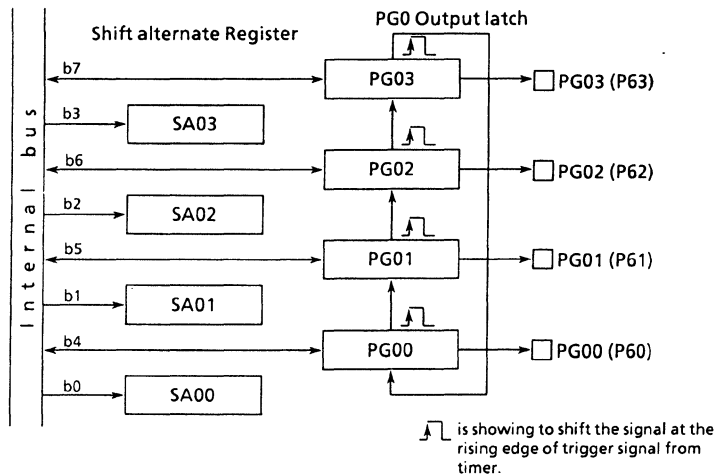
The output latch of PG0 (also used as P6) is shifted at the rising edge of the trigger signal from the timer to be output to the port.

The direction of shift is specified by PG01CR <CCW0>: Normal rotation (PG00 → PG01 → PG02 → PG03) when <CCW0> is set to "0"; reverse rotation (PG00 ← PG01 ← PG02 ← PG03) when "1". 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 3.10 (10) shows the block diagram.

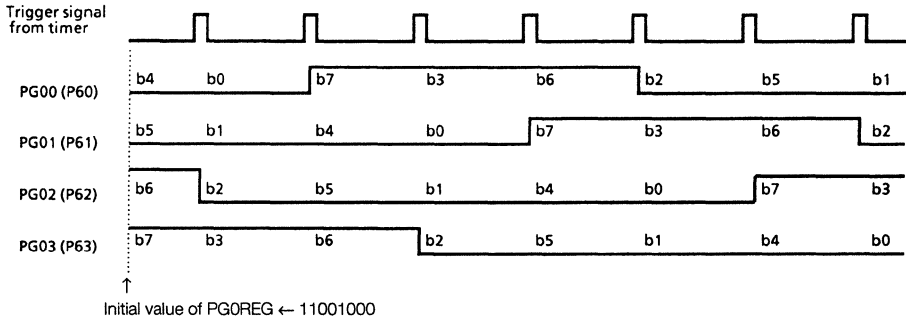


**Figure 3.10 (10). Block Diagram of 4-Phase 1-Step Excitation/2-Step Excitation (Normal Rotation)**

② 4-Phase 1-2 Step Excitation

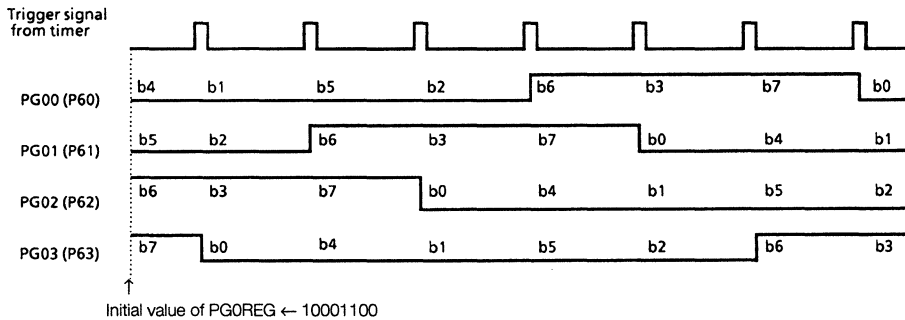
phase 1 -2 step excitation when channel 0 is selected.

Figure 3.10 (11) shows the output waveforms of 4-



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

① Normal Rotation



② Reverse Rotation

Figure 3.10 (11). Output Waveforms of 4-Phase 1-2 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 3.10 (12) shows the block diagram.

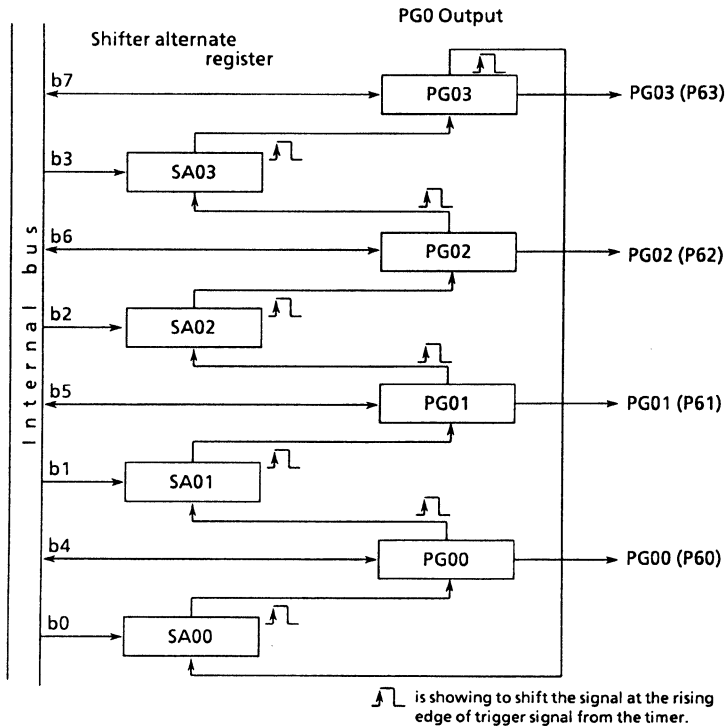


Figure 3.10 (12). Block Diagram of 4-Phase 1-2 Step Excitation (Normal Rotation)

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

timer 0 is selected, set each register as follows:

		7	6	5	4	3	2	1	0
TRUN	←	-	x	-	-	-	-	-	0
TMOD	←	0	0	x	x	-	-	0	1
TFFCR	←	x	x	x	0	1	0	1	0
TREG0	←	*	*	*	*	*	*	*	*
P6CR	←	-	-	-	-	1	1	1	1
P6FC	←	-	-	-	-	1	1	1	1
PG01CR	←	-	-	-	-	0	0	1	1
PGOREG	←	1	1	0	0	1	0	0	0
TRUN	←	1	-	-	-	-	-	-	1

- Stop timer 0, and clears it to zero.
- Set 8-bit timer mode and selects  $\phi T1$  as the input clock of timer 0.
- Clear TFF1 to zero and enables the inversion trigger by timer 0.
- Set the cycle in timer register.
- Set P60 – P63 bits to the output mode.
- Set P60 – P63 bits to the PG output.
- Select PG0 4-phase 1 - 2 step excitation mode and normal rotation.
- Set an initial value.
- Start timer 0.

Note: x; don't care      -; no change

(3) Trigger Signal From Timer

equal to the trigger signal of timer flip-flop (TFF1, TFF4, TFF5, and TFF6) and differs as shown in Table 3.10 (1) depending on the operation mode of the timer.

The trigger signal from the timer which is used by PG is not

**Table 3.10 (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/Timer 5. In this case, the PG shift trigger signal from the 16-bit timer is output only when the up-counter UC4/UC5 value matches TREG5/TREG7.

When using a trigger signal from Timer 4, set either

T4FFCR <EQ5T4> or T4MOD <EQ5T5> to "1" and a trigger is generated when the value in UC4 and the value in TREG5 match. When using a trigger signal from Timer 5, set T5FFCR <EQ7T6> to 1. Generates a trigger when the value in UC5 and the value in TREG7 match.

(4) Application of PG and Timer Output

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

PPG mode will be explained below.

To drive a stepping motor, in addition to the value of each phase (PG output), synchronizing signal is often required at the timing when excitation is changed over. In this application, port 6 is used as a stepping motor control port to output a synchronizing signal to the TO1 pin (shared by P71).

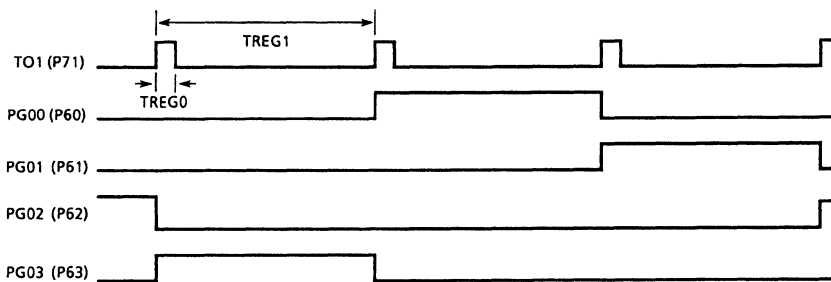


Figure 3.10 (13). Output Waveforms of 4-Phase 1-Step Excitation

Setting example:

	7	6	5	4	3	2	1	0	
TRUN	← -	← -	← -	← -	← -	← -	← 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	← x	← x	← x	← x	← -	← -	← 1	← -	) Assign P71 as TO1.
P7FC	← x	← x	← x	← x	← -	← -	← 1	← x	
P6CR	← -	← -	← -	← -	← 1	← 1	← 1	← 1	) Assign P60 - 63 as PG0.
P6FC	← -	← -	← -	← -	← 1	← 1	← 1	← 1	
PG01CR	← -	← -	← -	← -	← 0	← 0	← 0	← 1	Set PG0 in 4-phase 1-step excitation mode.
PGOREG	← *	← *	← *	← *	← *	← *	← *	← *	Set an initial value.
TRUN	← 1	← x	← -	← -	← -	← -	← 1	← 1	Start timer 0 and timer 1.

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

**3.11 Serial Channel**

The TMP96C141AF contains two serial I/O channels for full duplex asynchronous transmission (UART)

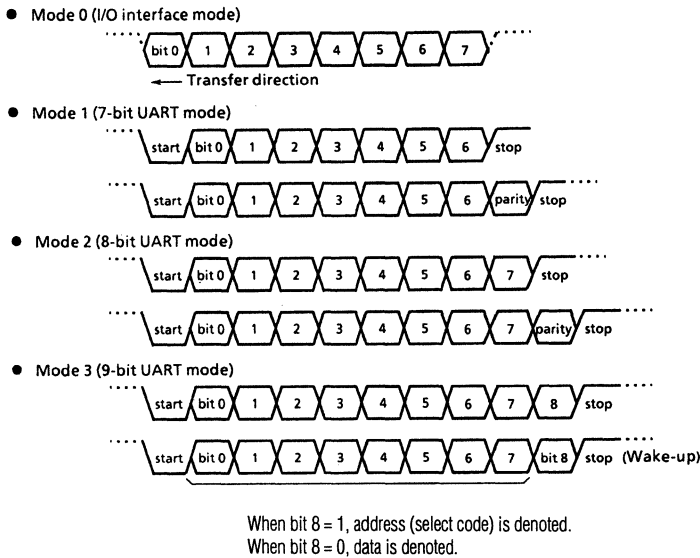
as well as for I/O extension.

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.
- Note: TMP96C141AF/TMP96C041AF/  
TMP96CM40F/TMP96PM40F with  
Channel 0 and 1.
- 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 3.11 (1) shows the data format (for one frame) in each mode.



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

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

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

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

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

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

detected to be normal at least twice in three samplings.

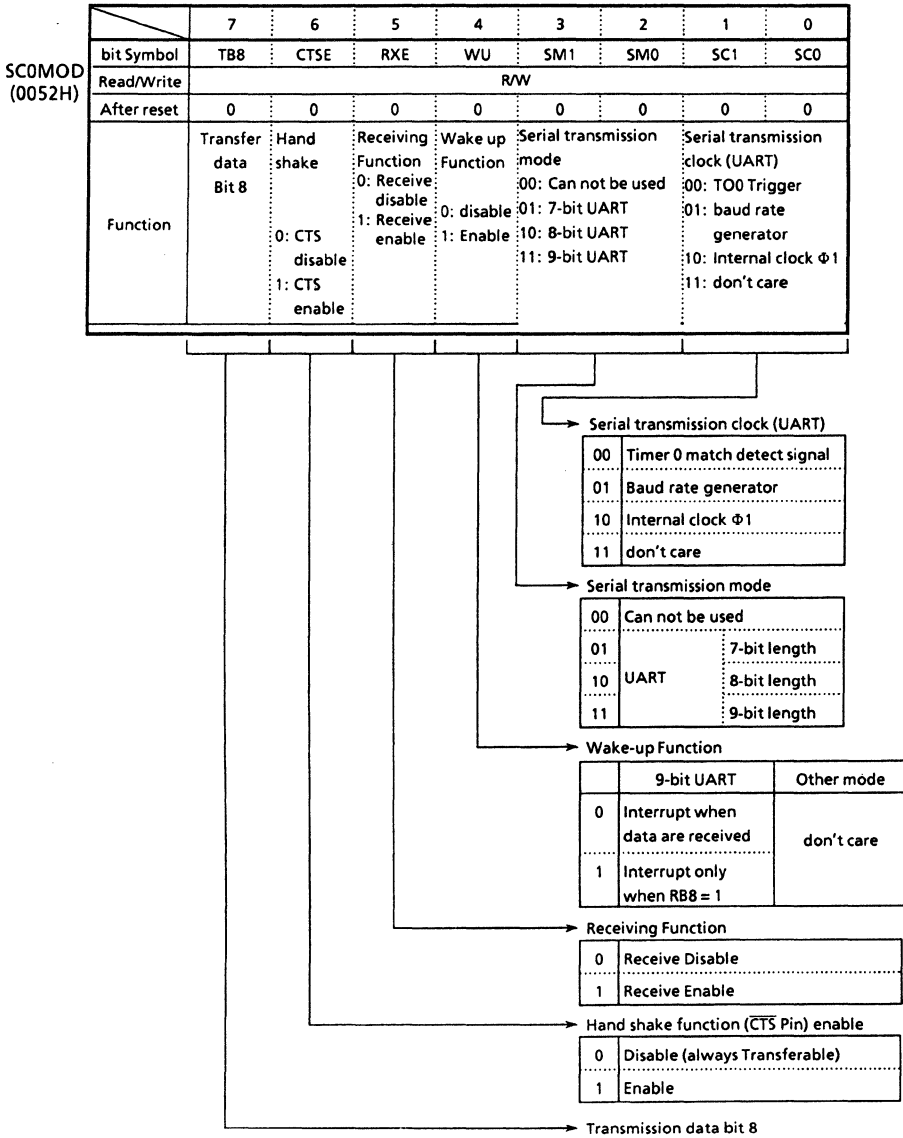
When the transmission buffer becomes empty and requests the CPU to send the next transmission data, or when data is stored in the receiving data register and the CPU is requested to read the data, INTTX or INTRX interrupt occurs. Besides, if an overrun error, parity error, or framing error occurs during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

The serial channel 0/1 includes a special baud rate generator, which can set any baud rate by dividing the frequency of four clocks ( $\phi T0$ ,  $\phi T2$ ,  $\phi T8$ , and  $\phi T32$ ) from the internal prescaler (shared by 8-bit/16-bit timer) by the value 2 to 16.

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

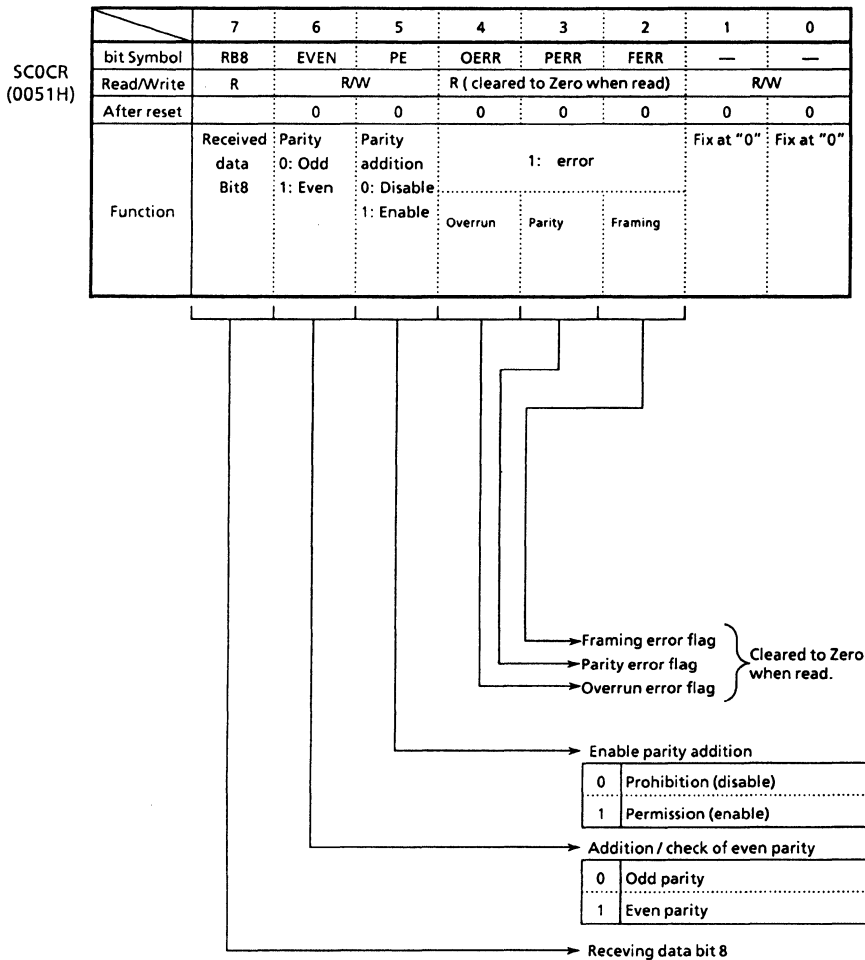
### 3.11.1 Control Registers

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



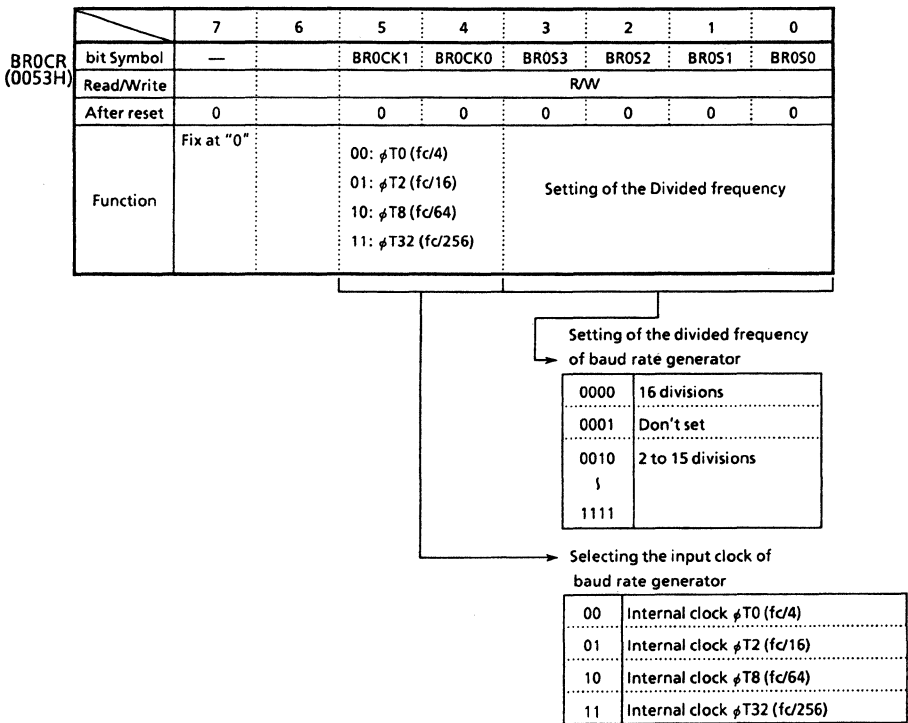
Note: There is SC1MOD (56H) in Channel 1

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



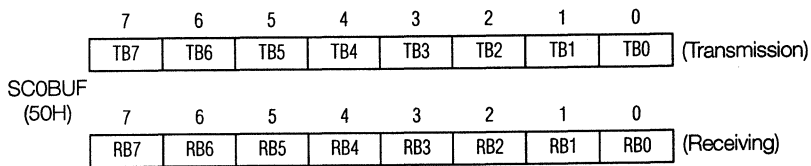
Note: Serial control register for channel 1 is SC1CR (55H).  
 As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

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



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

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



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



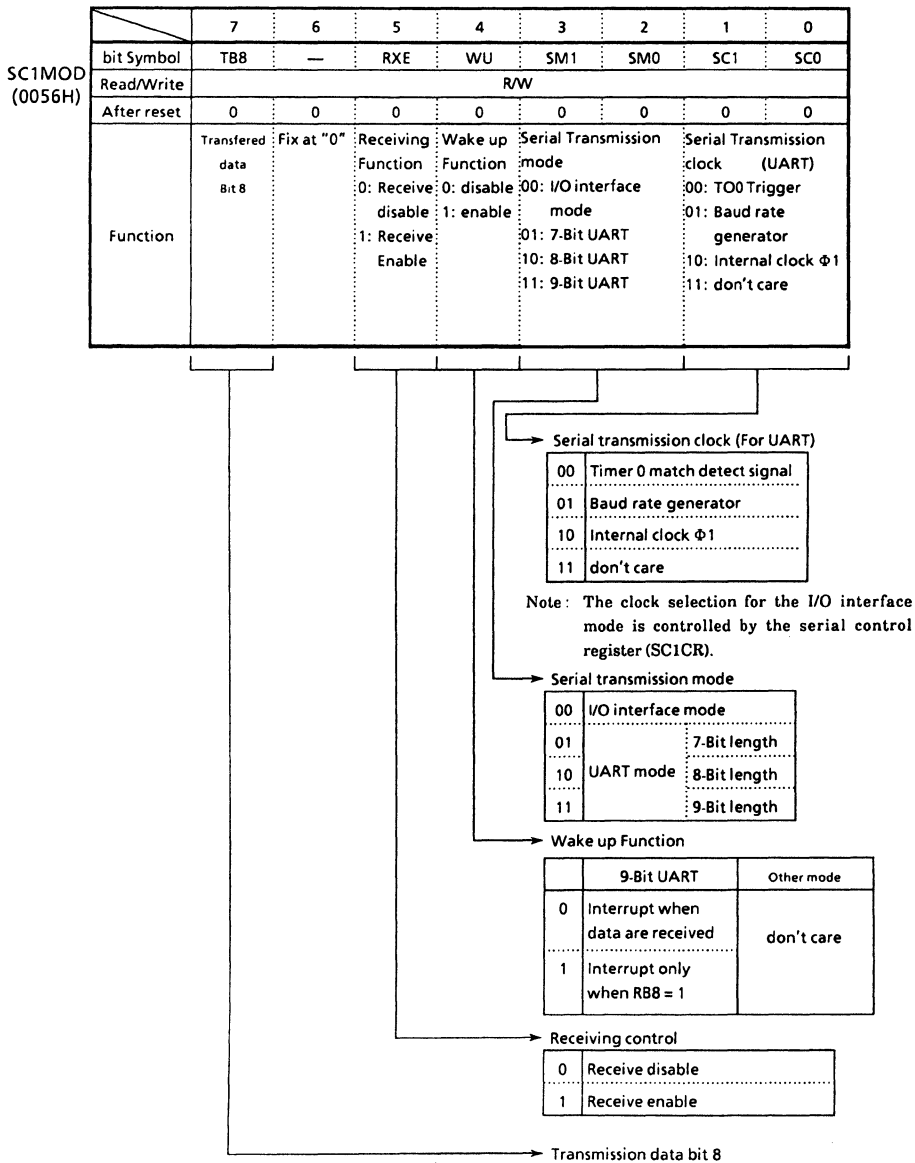
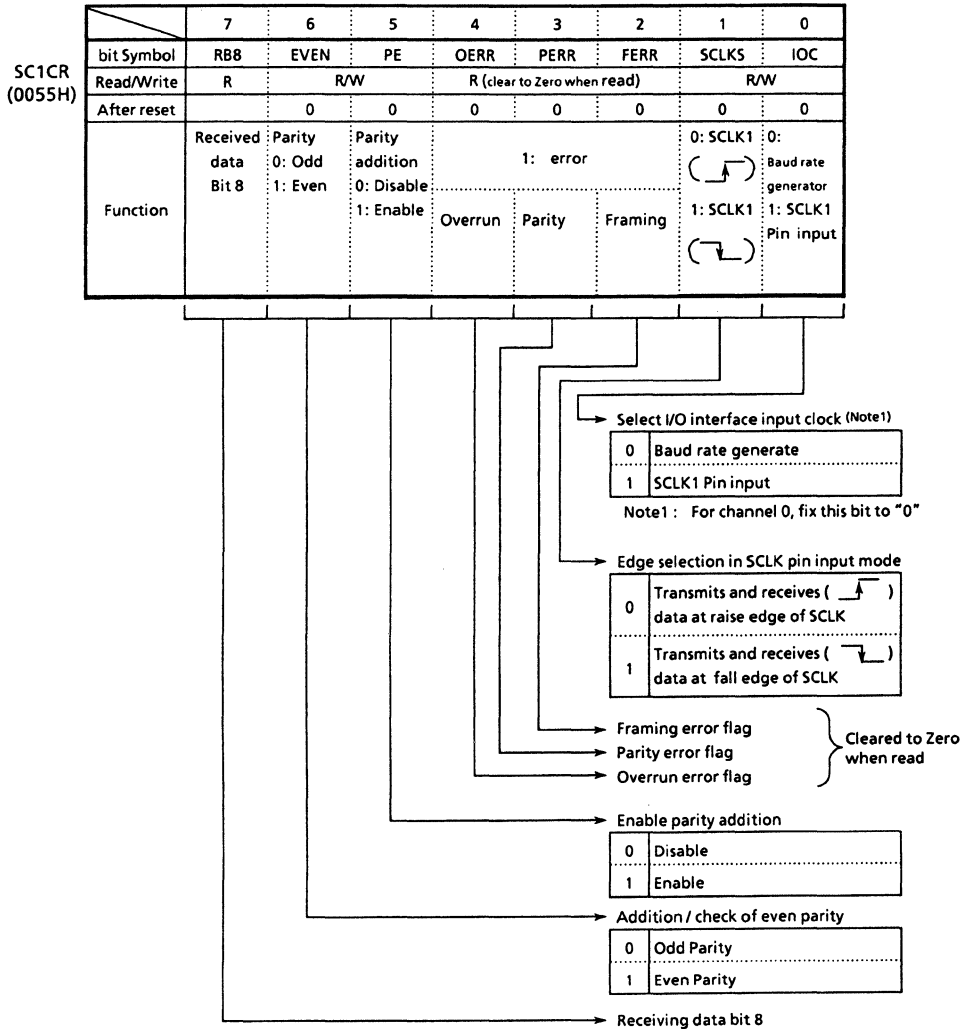
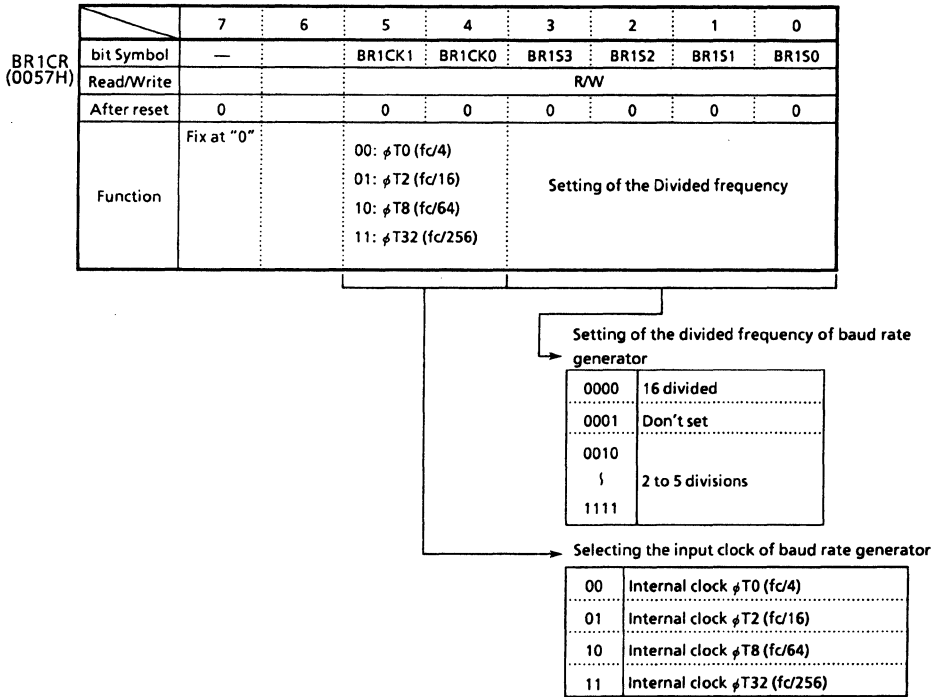


Figure 3.11 (6). Serial Mode Control Register (Channel 1, SC1MOD)



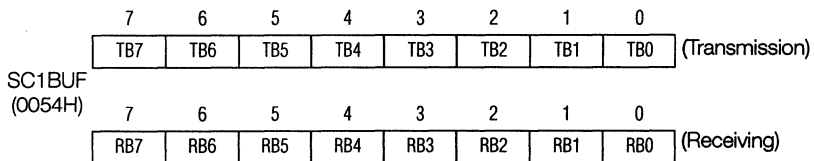
Note: As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

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



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

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



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

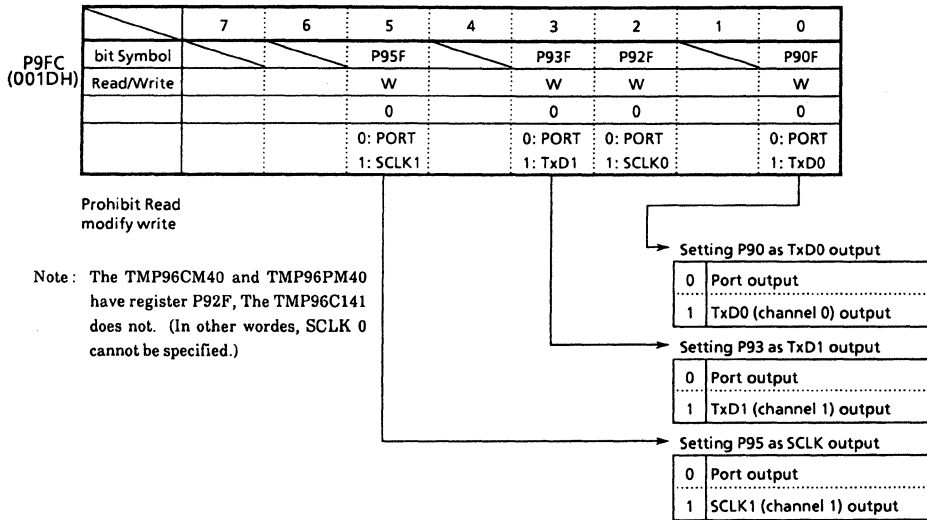
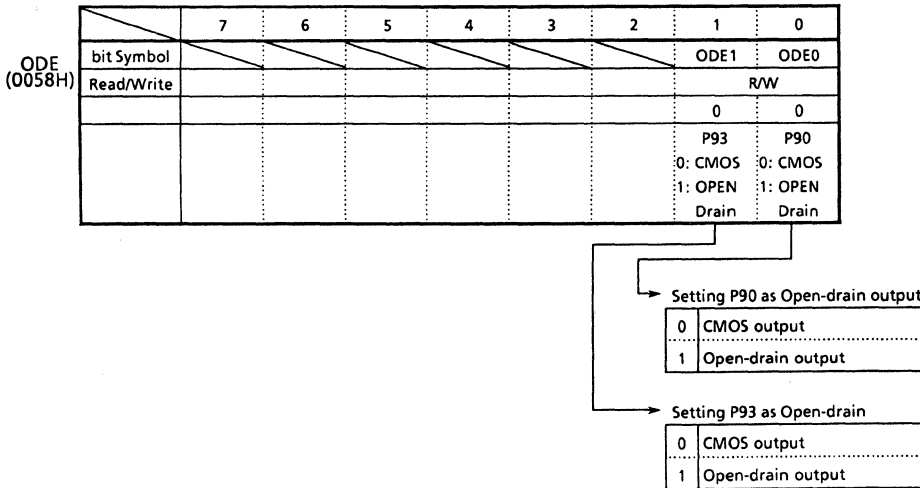


Figure 3.11 (10). Port 9 Function Register (P9FC)



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

3.11.2 Configuration

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

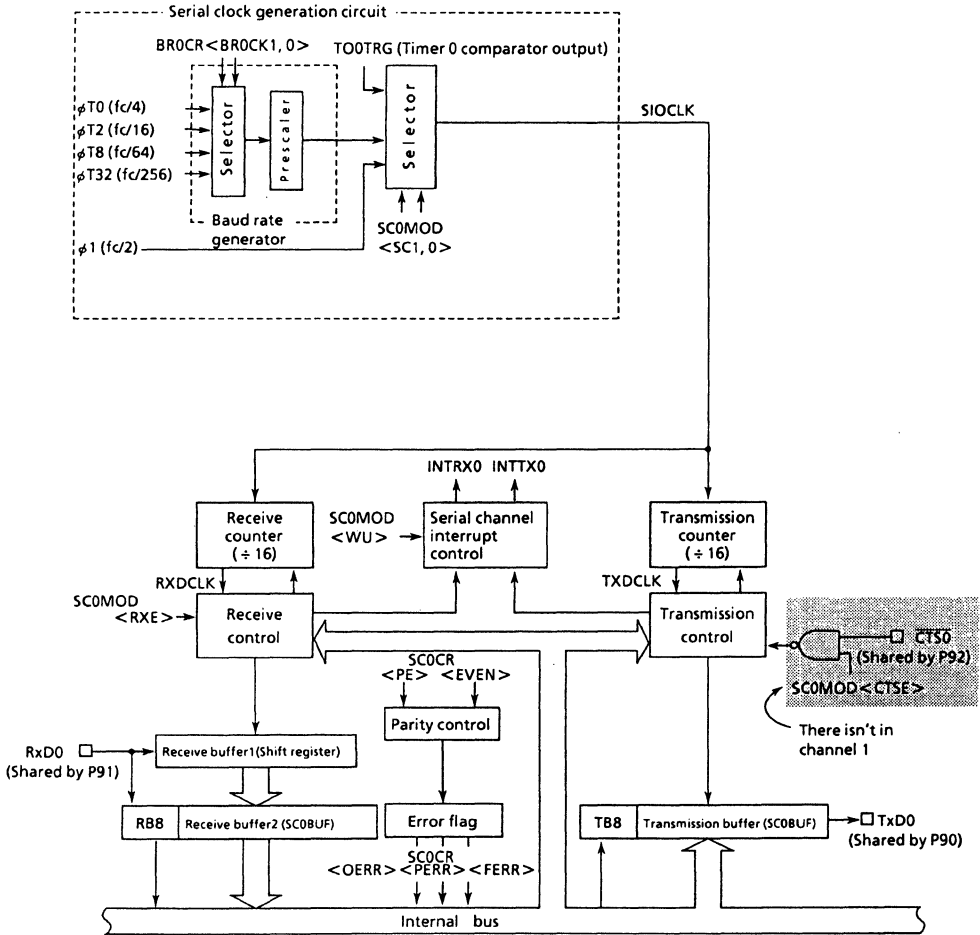


Figure 3.11 (12). Block Diagram of the Serial Channel 0

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

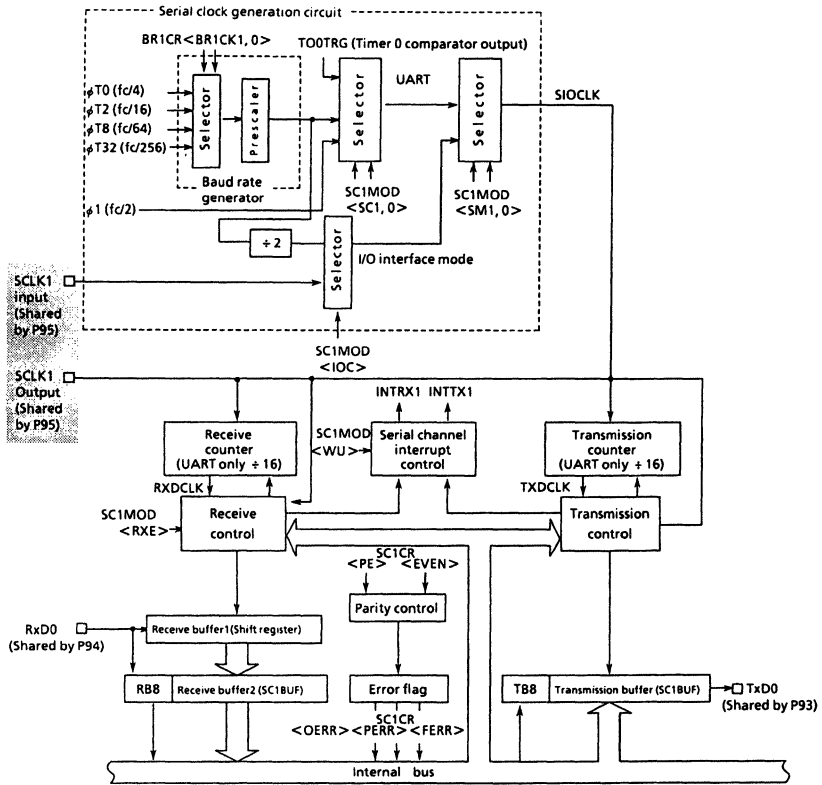


Figure 3.11 (13). Block Diagram of the Serial Channel 1

① Baud Rate Generator

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

The input clock to the baud rate generator,  $\phi T0$  ( $fc/4$ ),  $\phi T2$  ( $fc/16$ ),  $\phi T8$  ( $fc/64$ ), or  $\phi T32$  ( $fc/256$ ) is generated by the 9-bit prescaler which is shared by the timers. One

of these input clocks is selected by the baud rate generator control register 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

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

● I/O interface mode

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

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

$$\phi T0 = fc/4$$

$$\phi T2 = fc/16$$

$$\phi T8 = fc/64$$

$$\phi T32 = fc/256$$

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

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

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

Also with 8-bit timer 0, the serial channel can get a transfer rate. Table 3.11 (2) shows an example of baud rate using timer 0.

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

fc [Mhz]	Input Clock					Unit (kbps)
	Frequency Divisor	$\phi T0$ (fc/4)	$\phi T2$ (fc/16)	$\phi T8$ (fc/64)	$\phi T32$ (fc/256)	
9.830400	2	76.800	19.200	4.800	1.200	
↑	4	38.400	9.600	2.400	0.600	
↑	8	19.200	4.800	1.200	0.300	
↑	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 3.11 (2) Selection of Transfer Rate (1) (When Timer 0 (Input Clock  $\phi T1$ ) 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):

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

↑  
(When timer 0 (input clock  $\phi T1$ ) is used)

Input clock of timer 0

$$\phi T1 = f_c/8$$

$$\phi T4 = f_c/32$$

$$\phi T16 = f_c/128$$

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.

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

When in SCLK output mode with the setting of SC1CR <IOC> = "0", the basic clock will be generated by dividing by 2 the output of the baud rate generator as described before. When in SCLK input mode with the setting of 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.

### 2) Asynchronous Communication (UART) mode

According to the setting of SC0CR and SC1CR <SC1, 0>, the above baud rate generator clock, internal clock  $\phi 1$  (500 Kbps @  $f_c = 16$  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. Sixteen pulses of SIOCLK are used for receiving one bit of data, and the data bit is sampled three times at 7th, 8th and 9th clock.

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

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

## ④ Receiving Control

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

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

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



2) Asynchronous Communication (UART) mode

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

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

⑤ Receiving Buffer

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

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

the receiving buffer 1. However, unless the receiving buffer 2 (SC0BUF/SC1BUF) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of the receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC0CR <RB8> SC1CR <RB8> are still preserved.

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 TxCLK every 16 clock pulses.

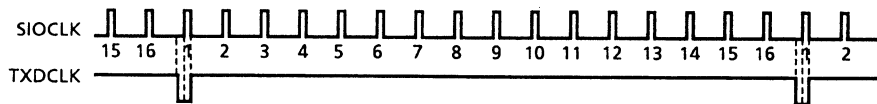


Figure 3.11 (14). Generation of Transmission Clock

⑦ Transmission Controller

1) 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.

2) Asynchronous Communication (UART) mode

When transmission data is written in the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TxCLK, 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 SCOMOD <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.

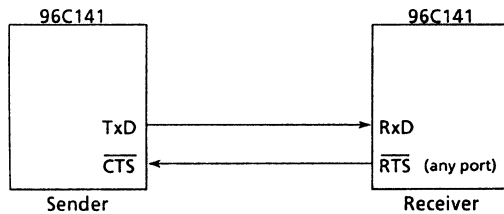
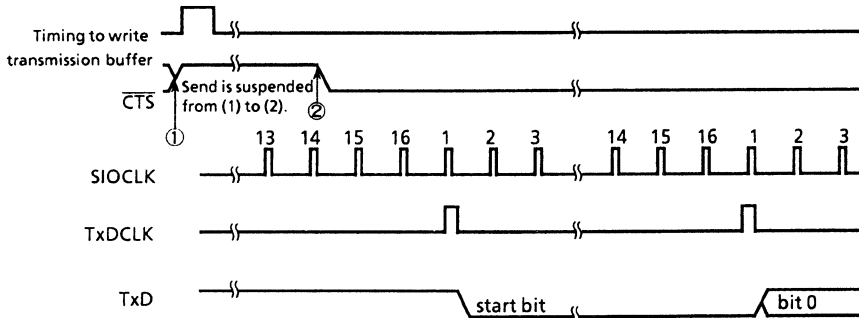


Figure 3.11 (15). Handshake Function



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

Figure 3.11 (16). Timing of  $\overline{\text{CTS}}$  (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

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

Transmitting

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

2) I/O Interface mode

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

3.11.3 Operational Description

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

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

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

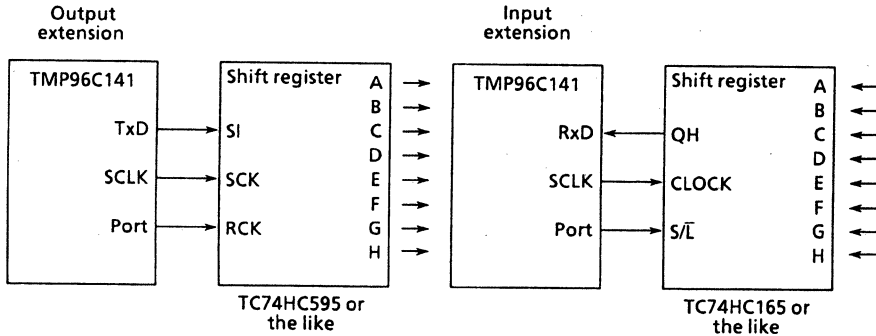


Figure 3.11 (17). Example of SCLK Output Mode Connection

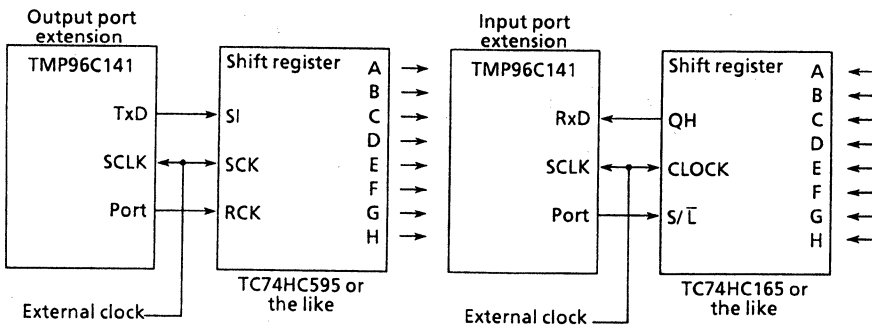


Figure 3.11 (18). Example of SCLK Input Mode Connection

① Transmission

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

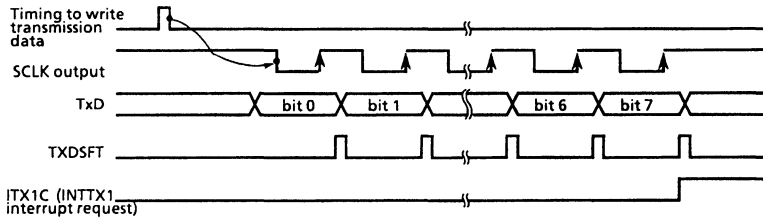


Figure 3.11 (19) Transmitting Operation in I/O Interface Mode (SCLK Output Mode)

In SCLK input 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.

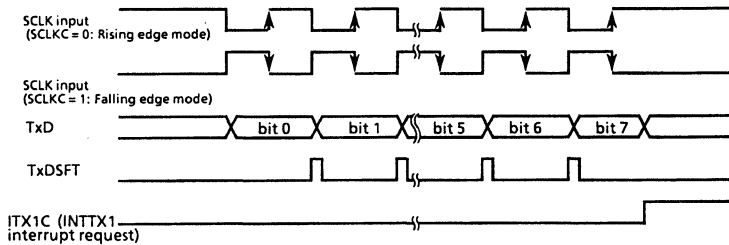


Figure 3.11 (20). Transmitting Operation in I/O Interface Mode (SCLK Input Mode)

② Receiving

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

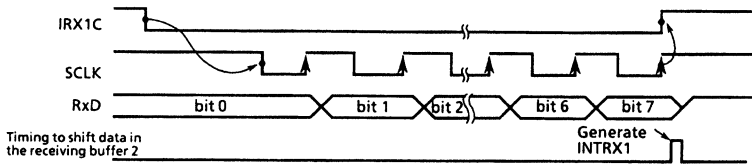


Figure 3.11 (21). Receiving Operation in I/O Interface Mode (SCLK Output Mode)

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

data will be shifted in the receiving buffer 2 (SC1BUF) at the timing shown below, and INTES1 <IRX1C> will be set again to generate INTRX1 interrupt.

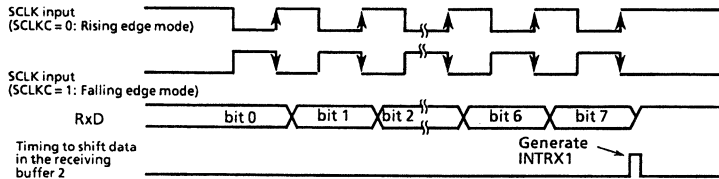


Figure 3.11 (22). Receiving Operation in I/O Interface Mode (SCLK Input Mode)

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

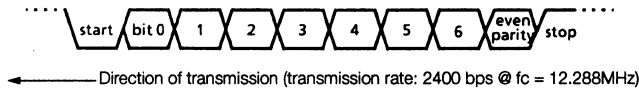
(2) Mode 1 (7-bit UART Mode)

The 7-bit mode can be set by setting serial channel mode register SCOMOD <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	
P9CR	←	x	x	-	-	-	-	-	1	) Select P90 as the TxD pin.
P9FC	←	x	x	-	x	-	x	x	1	
SCOMOD	←	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.
BROCR	←	0	x	1	0	0	1	0	1	Set transfer rate at 2400 bps.
TRUN	←	1	x	-	-	-	-	-	-	Start the prescaler for the baud rate generator.
INTES0	←	1	1	0	0	-	-	-	-	Enable INTTX0 interrupt and sets interrupt level 4.
SC0BUF	←	*	*	*	*	*	*	*	*	Set data for transmission.

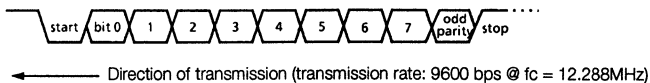
Note: x; don't care -; no change

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

The 8-bit UART mode can be specified by setting SCOMOD <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
P9CR	← x	x	-	-	-	-	0	-
SC0MOD	← -	0	1	x	1	0	0	1
SC0CR	← x	0	1	x	x	x	0	0
BROCR	← 0	x	0	1	0	1	0	1
TRUN	← 1	x	-	-	-	-	-	-
INTES0	← -	-	-	-	1	1	0	0

- Select P91 (Rx/D) as the input pin.
- Enable receiving in 8-bit UART mode.
- Add an odd parity.
- Set transfer rate at 9600 bps.
- Start the prescaler for the baud rate generator.
- Enable INTTX0 interrupt and sets interrupt level 4.

Interrupt processing

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

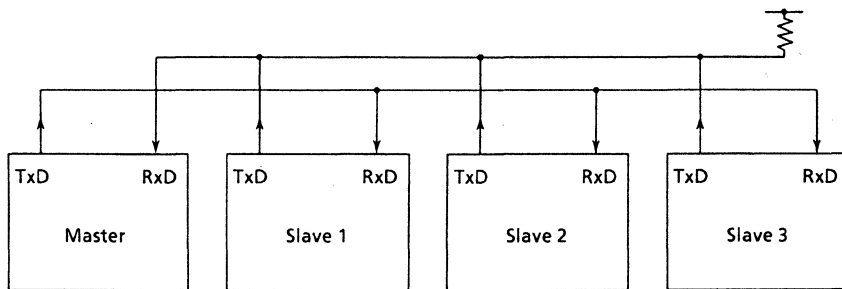
Note: x; don't care -; no change

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

The 9-bit UART mode can be specified by setting SC0MOD <SM1, 0> / SC1MOD <SM1, 0> to "11". In this mode, parity bit cannot be added. For transmission, the MSB (9th bit) is written in SCMOD <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: Tx/D pin of the slave controllers must be in open drain output mode.

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

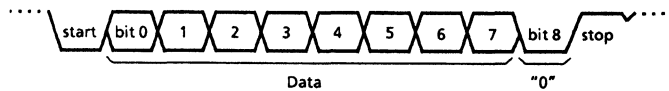


Protocol

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



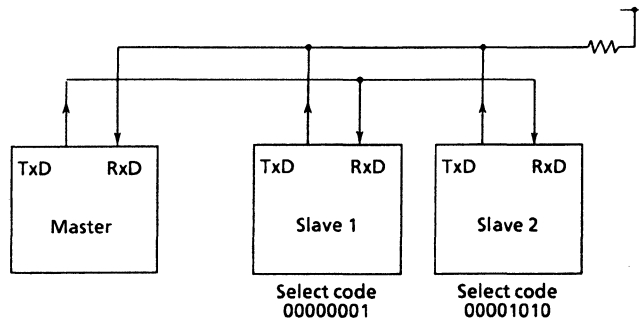
- ④ Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.
- ⑤ The master controller transmits data to the specified slave controller whose SC0MOD <WU>/SC1MOD <WU> bit is cleared to "0." The MSB (bit 8) <TB8> is cleared to "0".



- 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  $\phi 1$  ( $f_c/2$ ) as the transfer clock.



Since serial channels 0 and 1 operate in exactly the same

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

• Setting the master controller

Main setting

P9CR	←	x	x	-	-	-	-	0	1	) Select P90 as TxD pin and P91 as RxD pin.
P9FC	←	x	x	-	x	-	x	x	1	
INTES0	←	1	1	0	0	1	1	0	1	
SCOMOD	←	1	0	1	0	1	1	1	0	Set $\phi 1$ ( $f_c/2$ ) 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

SCOMOD	←	-	0	-	-	-	-	-	-	Set TB8 to "0".
SC0BUF	←	*	*	*	*	*	*	*	*	Set data for transmission.

• Setting the slave controller 2

Main setting

P9CR	←	x	x	-	-	-	-	0	1	) Select P91 as RxD pin and P90 as TxD pin (open drain output).
P9FC	←	x	x	-	x	-	x	x	1	
ODE	←	x	x	x	x	x	x	-	1	
INTES0	←	1	1	0	1	1	1	1	0	Enable INTRX0 and INTTX0.
SCOMOD	←	0	0	1	1	1	1	1	0	Set <WU> to "1" in the 9-bit UART transmission mode with transfer clock $\phi 1$ ( $f_c/2$ ).

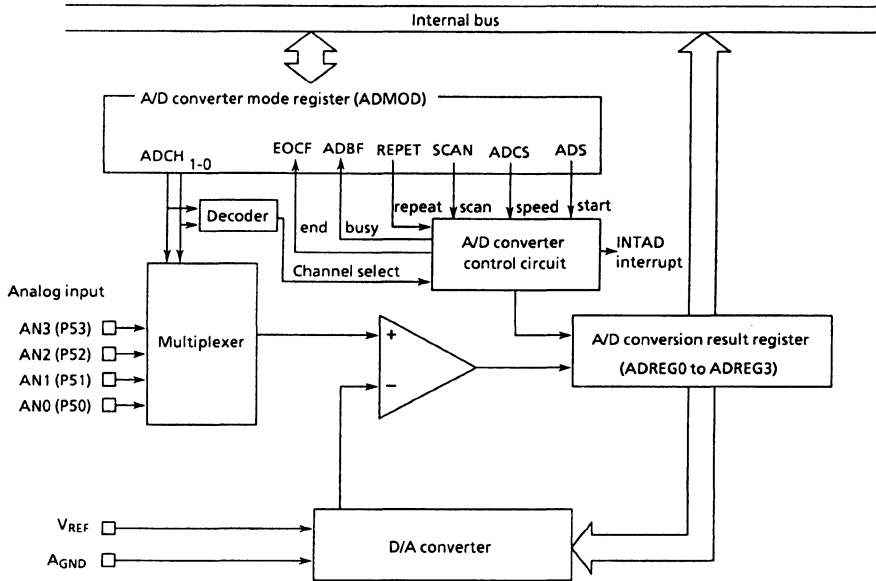
INTRX0 interrupt

Acc ← SC0BUF										
If Acc = Select Code										
Then SCOMOD4	←	-	-	-	-	0	-	-	-	Clear <WU> to "0".

**3.12 Analog/Digital Converter**

The TMP96C141AF contains a high-speed analog/digital converter (A/D converter) with 4-channel analog input that features 10-bit successive approximation.

Figure 3.12 (1) shows the block diagram of the A/D converter. The 4-channel analog input pins (AN3 to AN0) are shared by input-only P5 and so can be used as input port.



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

Note: 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.

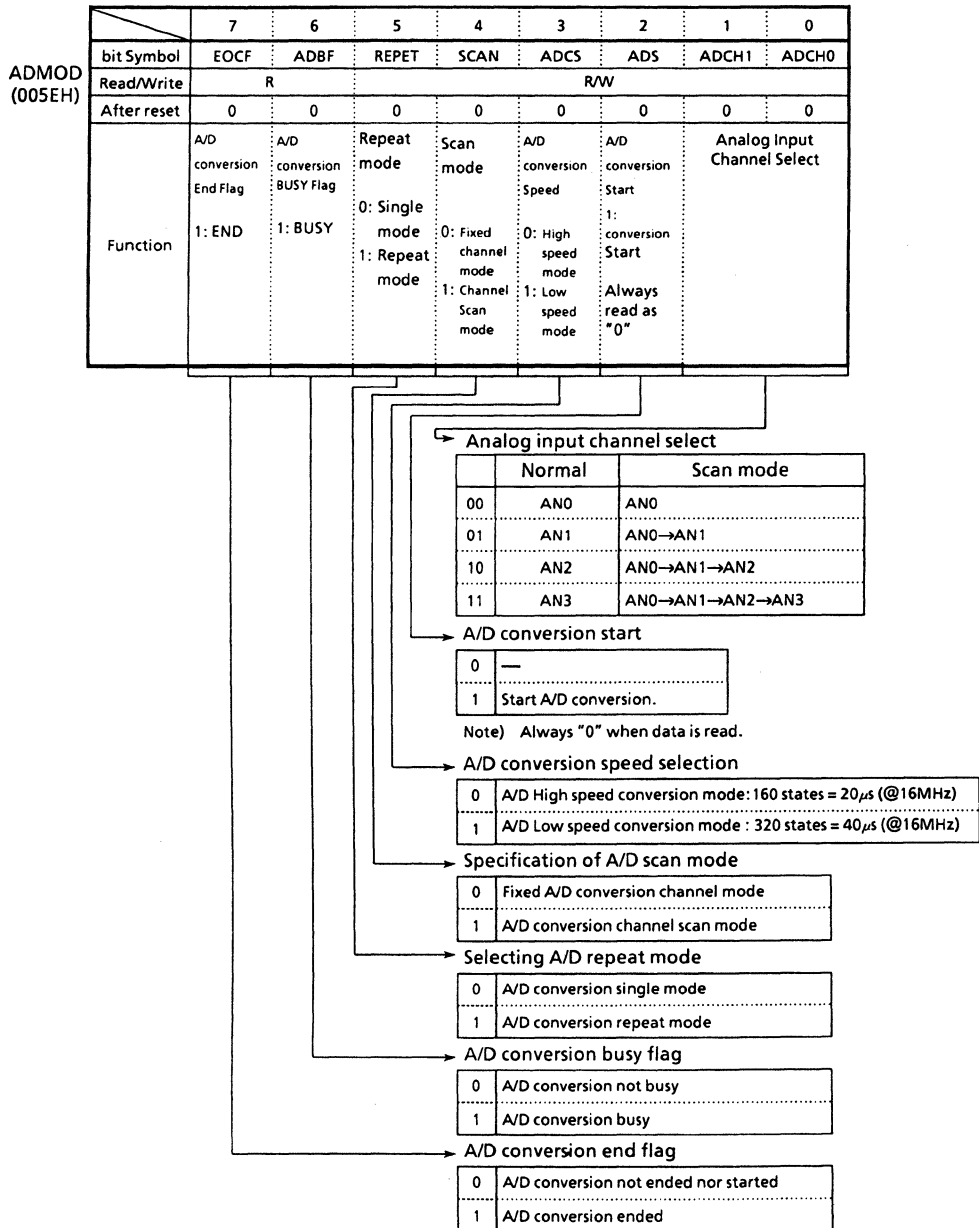


Figure 3.12 (2). A/D Control Register

	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
ADREG0L (0060H)	bit Symbol	ADR01	ADR00					
	Read/Write	R						
	After reset	Undefined		1	1	1	1	1
	Function	Lower 2 bits of A/D result for AN0 are stored.						

	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>	
ADREG0H (0061H)	bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN0 are stored.							

	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
ADREG1L (0062H)	bit Symbol	ADR11	ADR10					
	Read/Write	R						
	After reset	Undefined		1	1	1	1	1
	Function	Lower 2 bits of A/D result for AN1 are stored.						

	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>	
ADREG1H (0063H)	bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN1 are stored.							

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

ADREG2L (0064H)		7	6	5	4	3	2	1	0
	bit Symbol	ADR21	ADR20						
	Read/Write	R							
	After reset	Undefined		1	1	1	1	1	1
	Function	Lower 2 bits of A/D result for AN2 are stored.							

ADREG2H (0065H)		7	6	5	4	3	2	1	0
	bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN2 are stored.							

ADREG3L (0066H)		7	6	5	4	3	2	1	0
	bit Symbol	ADR31	ADR30						
	Read/Write	R							
	After reset	Undefined		1	1	1	1	1	1
	Function	Lower 2 bits of A/D result for AN3 are stored.							

ADREG3H (0067H)		7	6	5	4	3	2	1	0
	bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN3 are stored.							

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

### 3.12.1 Operation

#### (1) Analog Reference Voltage

High analog reference voltage is applied to the VREF pin, and low analog reference voltage is applied to AGND pin.

The reference voltage between VREG and AGND is divided by 1024 using ladder resistance, and compared with the analog input voltage for A/D conversion.

#### (2) Analog Input Channels

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

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

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

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

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

##### • A/D conversion repeat mode

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

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

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

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

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

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

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

INTAD routine											
WA	←	ADREG3								Read ADREG3L and ADREG3H values and writes to WA (16 bit).	
WA	>	>	6								Right-shifts WA six times and writes 0 in upper bits.
(00FF10H)	←	WA								Writes contents of WA in memory at FF10H.	
When the analog input voltage of AN0 – AN2 pins is A/D converted in high speed conversion channel scan repeat mode.											
INTE0AD	←	1	0	0	-	-	-	-	-	Disable INTAD.	
ADMOD	←	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

### 3.13 Watchdog Timer (Runaway Detecting Timer)

The TMP96C141AF is containing watchdog timer of Runaway detecting.

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

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

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



3.13.1 Configuration

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

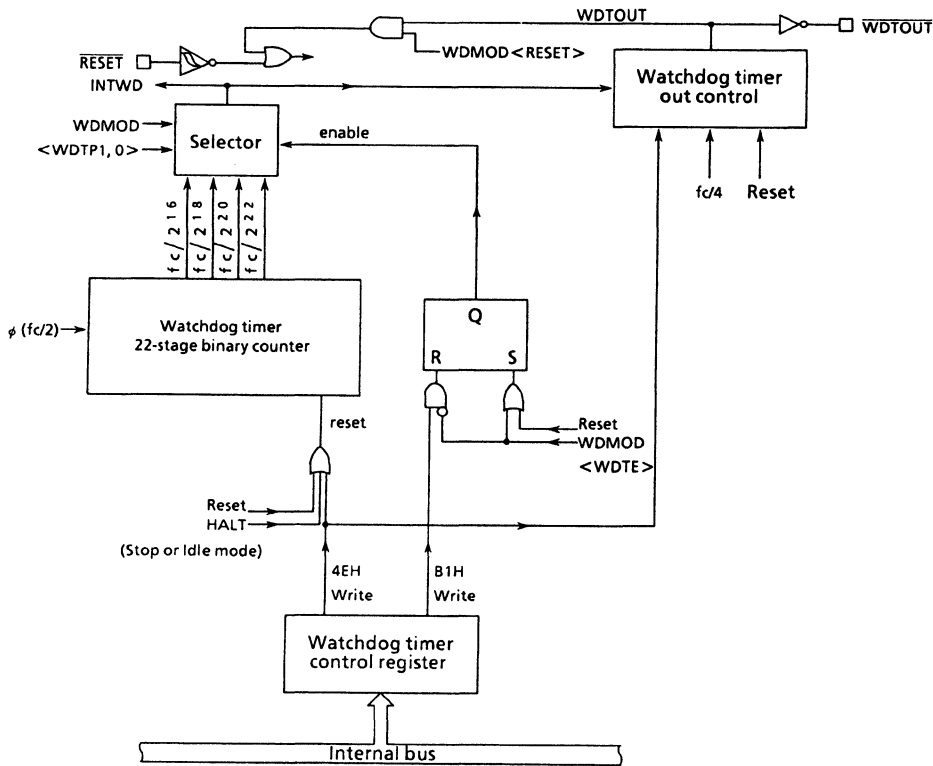


Figure 3.13 (1). Block Diagram of Watchdog Timer

The watchdog timer is a 22-stage binary counter which uses  $\phi$  ( $f_c/2$ ) as the input clock. There are four outputs from the binary counter:  $2^{16}/f_c$ ,  $2^{18}/f_c$ ,  $2^{20}/f_c$ , and  $2^{22}/f_c$ . 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. The watchdog timer out pin is set to 1 by clearing the watchdog timer (by writing a clear code 4EH in the WDCR register). 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 $\mu$ s @ 20MHz) and resets itself.

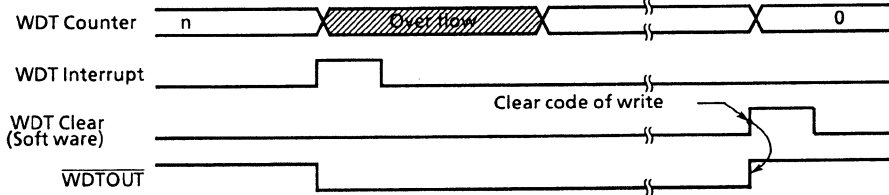


Figure 3.13 (2). Normal Mode

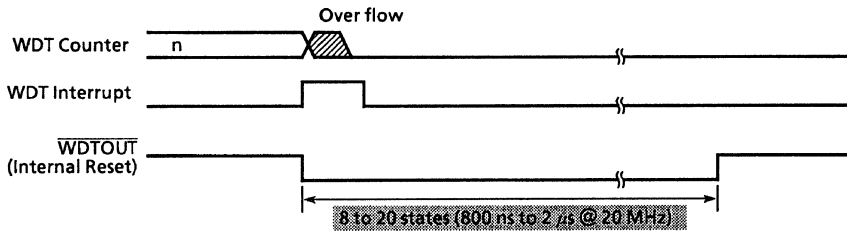


Figure 3.13 (3). Reset Mode

**3.13.2 Control Registers**

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

(1) Watchdog Timer Mode Register (WDMOD)

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

This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD <WDTP1, 0> = 00 when reset, and therefore  $2^{16}/fc$  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.

- Disable control

WDMOD	←	0	-	-	-	-	-	x	x
WDCR	←	1	0	1	1	0	0	0	1

Clear WDMOD <WDTE> to "0".

Write the disable code (B1H).

- Enable control  
Set WDMOD <WDTE> to "1".
- 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).

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

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

- ③ Watchdog timer out reset connection <RESCR>

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

(2) Watchdog Timer Control Register (WDCR)

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

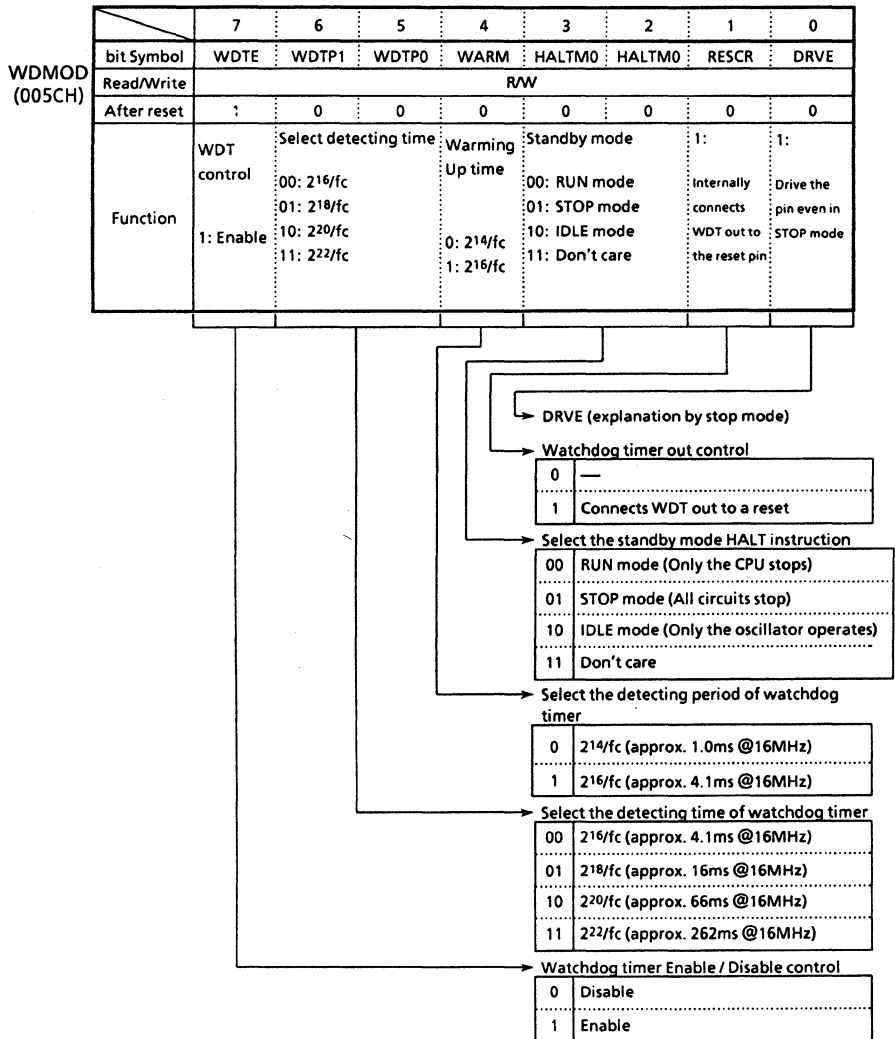


Figure 3.13 (4). Watchdog Timer Mode Register

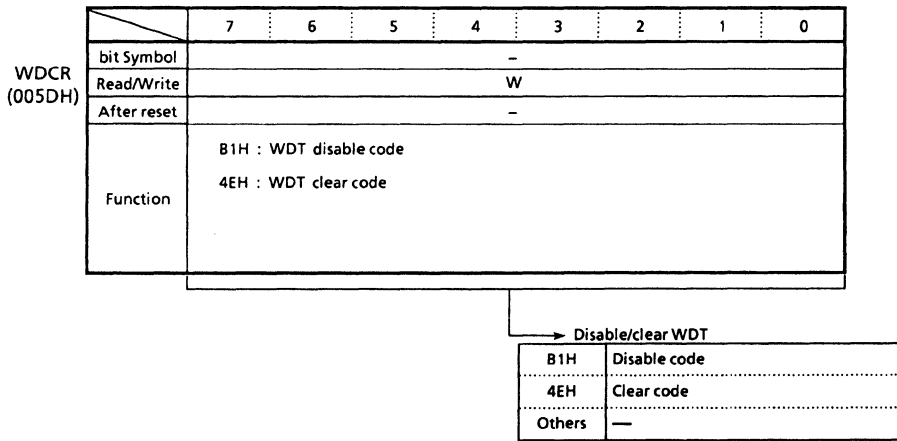


Figure 3.13 (5). Watchdog Timer Control Register

### 3.13.3 Operation

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

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

The watchdog timer restarts operation immediately after resetting is released.

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

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

Example:

- ① Clear the binary counter

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

- ② Set the watchdog timer detecting time to  $2^{18}/f_c$

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:  $2^{16}/f_c$ )

WDMOD ← - - - 1 0 1 x x      Set the STOP mode.

Executes HALT command

Execute HALT instruction. Set the standby mode.

## 4. Electrical Characteristics

### 4.1 Absolute Maximum (TMP96C141AF)

Symbol	Parameter	Rating	Unit
$V_{cc}$	Power Supply Voltage	-0.5 ~ 6.5	V
$V_{IN}$	Input Voltage	-0.5 - $V_{cc} + 0.5$	V
$\Sigma I_{OL}$	Output Current (total)	100	mA
$\Sigma I_{OH}$	Output Current (total)	-100	mA
PD	Power Dissipation ( $T_a = 70^\circ\text{C}$ )	600	mW
T SOLDER	Soldering Temperature (10s)	260	$^\circ\text{C}$
T STG	Storage Temperature	-65 ~ 150	$^\circ\text{C}$
T OPR	Operating Temperature	-20 ~ 70	$^\circ\text{C}$

### 4.2 DC Characteristics (TMP96C141AF)

$V_{cc} = 5V \pm 10\%$ ,  $T_a = -20 \sim 70^\circ\text{C}$  (Typical values are for  $T_a = 25^\circ\text{C}$  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	
$V_{IL2}$	RESET, NMI, INTO (P87)	-0.3	$0.25V_{cc}$	V	
$V_{IL3}$	$\overline{EA}$	-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	
$V_{IH2}$	RESET, 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	$I_{OL} = 1.6\text{mA}$
$V_{OH}$	Output High Voltage	2.4		V	$I_{OH} = -400\mu\text{A}$
$V_{OH1}$		$0.75V_{cc}$		V	$I_{OH} = -100\mu\text{A}$
$V_{OH2}$		$0.9V_{cc}$		V	$I_{OH} = -20\mu\text{A}$
$I_{DAR}$	Darlington Drive Current (8 Output Pins max.)	-1.0	-3.5	mA	$V_{EXT} = 1.5V$ $R_{EXT} = 1.1K\Omega$
$I_{LI}$	Input Leakage Current	0.02 (Typ)	$\pm 5$	$\mu\text{A}$	$0.0 \leq V_{in} \leq V_{cc}$
$I_{LO}$	Output Leakage Current	0.05 (Typ)	$\pm 10$	$\mu\text{A}$	$0.2 \leq V_{in} \leq V_{cc} - 0.2$
$I_{cc}$	Operating Current (RUN)	26 (Typ)	50	mA	$t_{osc} = 16\text{MHz}$
	IDLE	10	10	mA	
	STOP ( $T_a = -20 \sim 70^\circ\text{C}$ )	1.7 (Typ)	50	$\mu\text{A}$	$0.2 \leq V_{in} \leq V_{cc} - 0.2$
	STOP ( $T_a = 0 \sim 50^\circ\text{C}$ )	0.2 (Typ)	10	$\mu\text{A}$	$0.2 \leq V_{in} \leq V_{cc} - 0.2$
$V_{STOP}$	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	$V_{IL2} = 0.2V_{cc}$ $V_{IH2} = 0.8V_{cc}$
R RST	RESET Pull Up Register	50	150	$K\Omega$	
C IO	Pin Capacitance		10	pF	$t_{osc} = 1\text{MHz}$
$V_{TH}$	Schmitt Width RESET, NMI, INTO (P87)	0.4	1.0 (Typ)	V	
R K	Pull Down/Up Register	50	150	$K\Omega$	

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

4.3 AC Electrical Characteristics (TMP96C141AF)  $V_{CC} = 5V \pm 10\%$ ,  $T_a = -20 \sim 70^\circ C$  (4MHz ~ 20MHz)

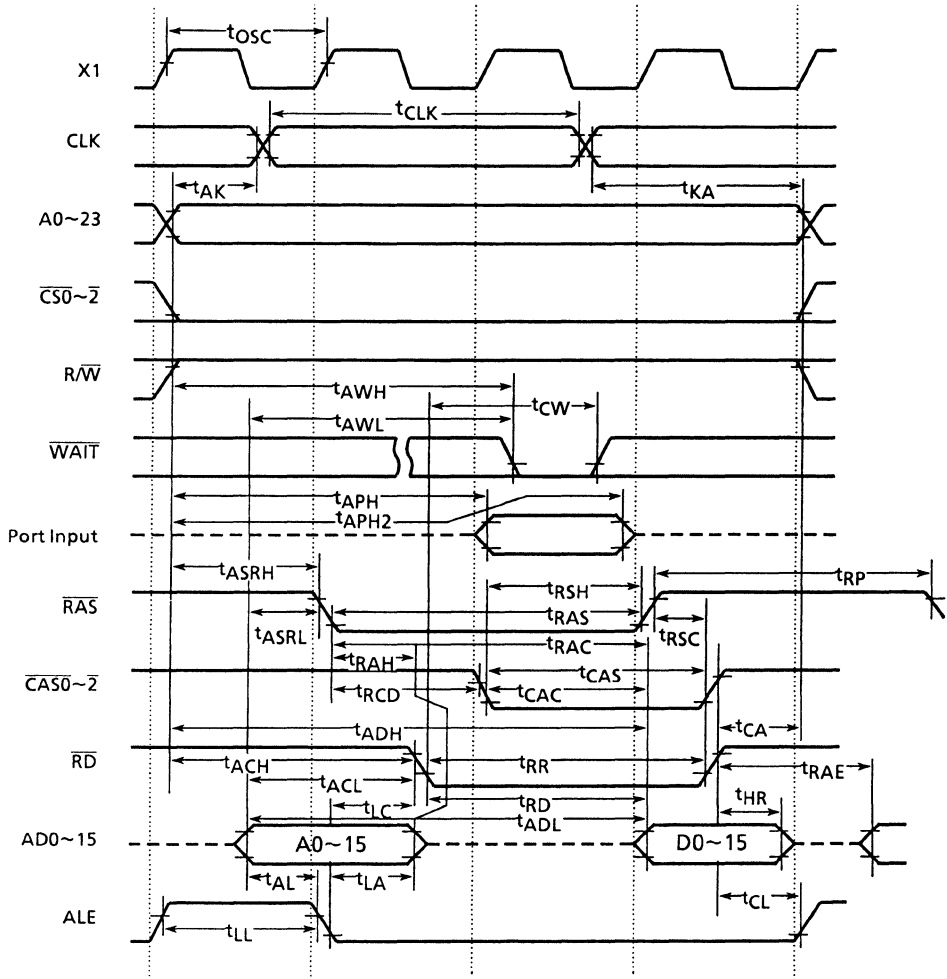
No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	$t_{OSC}$	Osc. Period (= x)	50	250	62.5		50		ns
2	$t_{CLK}$	CLK width	2x - 40		85		60		ns
3	$t_{AK}$	A0 - 23 Valid $\rightarrow$ CLK Hold	0.5x - 20		11		5		ns
4	$t_{KA}$	CLK Valid $\rightarrow$ A0 - 23 Hold	1.5x - 70		24		5		ns
5	$t_{AL}$	A0-15 Valid $\rightarrow$ ALE fall	0.5x - 15		16		10		ns
6	$t_{LA}$	ALE fall $\rightarrow$ 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 $\rightarrow$ $\overline{RD}/\overline{WR}$ fall	0.5x - 30		1		-5		ns
9	$t_{CL}$	$\overline{RD}/\overline{WR}$ rise $\rightarrow$ ALE rise	0.5x - 20		11		5		ns
10	$t_{ACL}$	A0 - 15 Valid $\rightarrow$ $\overline{RD}/\overline{WR}$ fall	x - 25		38		25		ns
11	$t_{ACH}$	A0 - 23 Valid $\rightarrow$ $\overline{RD}/\overline{WR}$ fall	1.5x - 50		44		25		ns
12	$t_{CA}$	$\overline{RD}/\overline{WR}$ rise $\rightarrow$ A0 - 23 Hold	0.5x - 20		11		5		ns
13	$t_{ADL}$	A0 - 15 Valid $\rightarrow$ D0 - 15 input		3.0x - 45		143		105	ns
14	$t_{ADH}$	A0 - 23 Valid $\rightarrow$ D0 - 15 input		3.5x - 65		154		110	ns
15	$t_{RD}$	$\overline{RD}$ fall $\rightarrow$ 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 $\rightarrow$ D0 - 15 Hold	0		0		0		ns
18	$t_{RAE}$	$\overline{RD}$ rise $\rightarrow$ 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 $\rightarrow$ $\overline{WR}$ rise	2.0x - 50		75		50		ns
21	$t_{WD}$	$\overline{WR}$ rise $\rightarrow$ D0 - 15 Hold	0.5x - 10		21		15		ns
22	$t_{AEH}$	A0 - 23 Valid $\rightarrow$ $\overline{WAIT}$ input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	$t_{AWL}$	A0 - 15 Valid $\rightarrow$ $\overline{WAIT}$ input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	$t_{CW}$	$\overline{RD}/\overline{WR}$ fall $\rightarrow$ $\overline{WAIT}$ Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	$t_{APH}$	A0 - 23 Valid $\rightarrow$ PORT input		2.5x - 120		80		36	ns
26	$t_{APH2}$	A0 - 23 Valid $\rightarrow$ PORT Hold	2.5x + 50		206		175		ns
27	$t_{CP}$	$\overline{WR}$ rise $\rightarrow$ PORT Valid		200		200		200	ns
28	$t_{ASRH}$	A0 - 23 Valid $\rightarrow$ $\overline{RAS}$ fall	1.0x - 40		23		10		ns
29	$t_{ASRL}$	A0 - 15 Valid $\rightarrow$ $\overline{RAS}$ fall	0.5x - 15		16		10		ns
30	$t_{RAC}$	$\overline{RAS}$ fall $\rightarrow$ D0 - 15 input		2.5x - 70		130		86	ns
31	$t_{RAH}$	$\overline{RAS}$ fall $\rightarrow$ A0 - 15 Hold	0.5x - 15		16		10		ns
32	$t_{RAS}$	$\overline{RAS}$ Low width	2.0x - 40		85		60		ns
33	$t_{RP}$	$\overline{RAS}$ High width	2.0x - 40		85		60		ns
34	$t_{RSH}$	$\overline{CAS}$ fall $\rightarrow$ $\overline{RAS}$ rise	1.0x - 35		28		15		ns
35	$t_{RSC}$	$\overline{RAS}$ rise $\rightarrow$ $\overline{CAS}$ rise	0.5x - 25		6		0		ns
36	$t_{RCD}$	$\overline{RAS}$ fall $\rightarrow$ $\overline{CAS}$ fall	1.0x - 40		23		10		ns
37	$t_{CAC}$	$\overline{CAS}$ fall $\rightarrow$ D0 - 15 input		1.5x - 65		29		10	ns
38	$t_{CAS}$	$\overline{CAS}$ Low width	1.5x - 30		64		40		ns

## AC Measuring Conditions

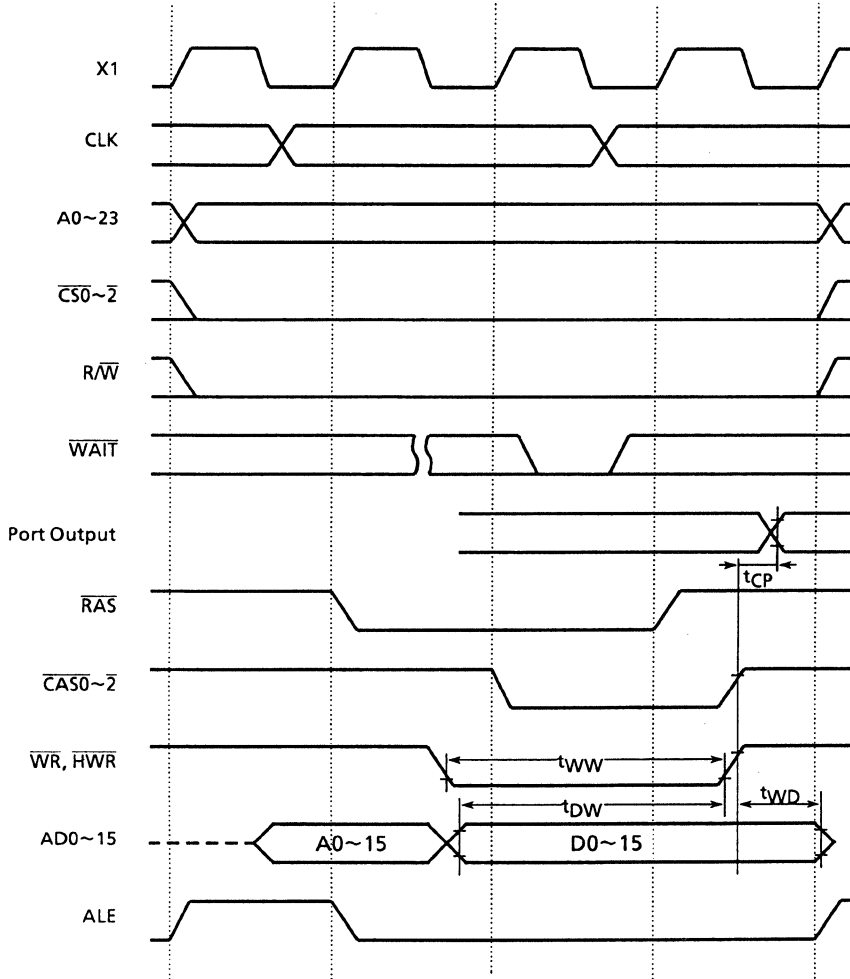
- Output Level: High 2.2V /Low 0.8V,  $CL50pF$   
(However  $CL = 100pF$  for AD0 ~ AD15, AD0 ~ AD23, ALE,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{HWR}$ ,  $R\overline{W}$ , CLK,  $\overline{RAS}$ ,  $\overline{CAS0} \sim \overline{CAS2}$ )
- Input Level: High 2.4V /Low 0.45V (AD0 ~ AD15)  
High 0.8V<sub>CC</sub> /Low 0.2V<sub>CC</sub> (Except for AD0 ~ AD15)



(1) Read Cycle



(2) Write Cycle



4.4 A/D Conversion Characteristics (TMP96C141AF)

$V_{CC} = 5V \pm 10\%$  TA = -20 ~ 70°C

Symbol	Parameter		Min	Typ	Max	Unit
V <sub>REF</sub>	Analog reference voltage		V <sub>CC</sub> - 1.5	V <sub>CC</sub>	V <sub>CC</sub>	V
A <sub>GND</sub>	Analog reference voltage		V <sub>SS</sub>	V <sub>SS</sub>	V <sub>SS</sub>	
V <sub>AIN</sub>	Analog input voltage range		V <sub>SS</sub>		V <sub>CC</sub>	
I <sub>REF</sub>	Analog current for analog reference voltage			0.5	1.5	mA
Error (Quantize error of ±0.5 LSB not included)	4 ≤ f <sub>c</sub> ≤ 16MHz	Low speed conversion mode		±1.5 (TBD)	±4.0	LSB
		High speed conversion mode		±3.0 (TBD)	±6.0	
	16 ≤ f <sub>c</sub> ≤ 20MHz	Low speed conversion mode		±1.5 (TBD)	±4.0	
		High speed conversion mode		±4.0 (TBD)	±8.0	

4.5 Serial Channel Timing - I/O Interface Mode

$V_{CC} = 5V \pm 10\%$  TA = -20 ~ 70°C

(1) SCLK Input Mode

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

(2) SCLK Output Mode

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

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

$V_{CC} = 5V \pm 10\%$  TA = -20 ~ 70°C

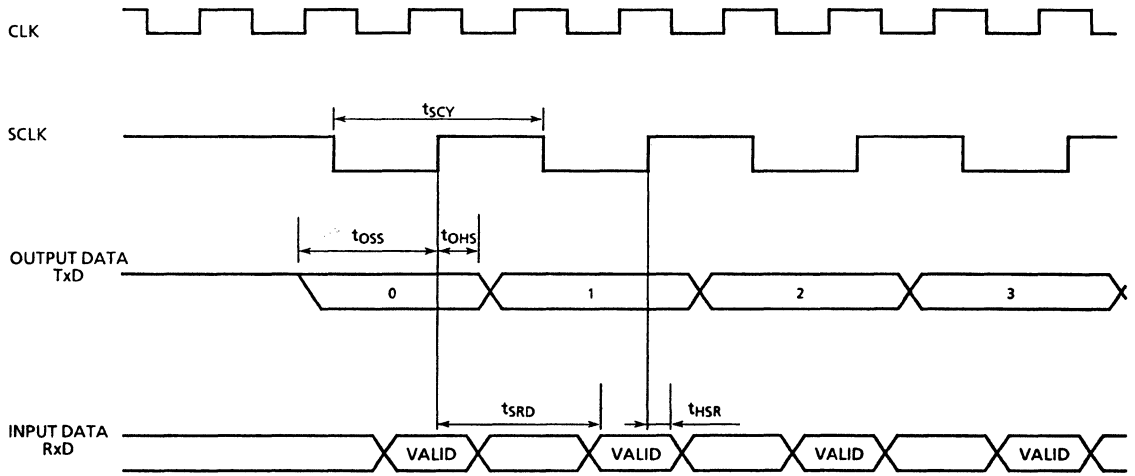
Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
t <sub>VCK</sub>	Clock cycle	8x + 100		600		500		ns
t <sub>VCKL</sub>	Low level clock pulse width	4x + 40		290		240		ns
t <sub>VCKH</sub>	High level clock pulse width	4x + 40		290		240		ns

## 4.7 Interrupt Operation

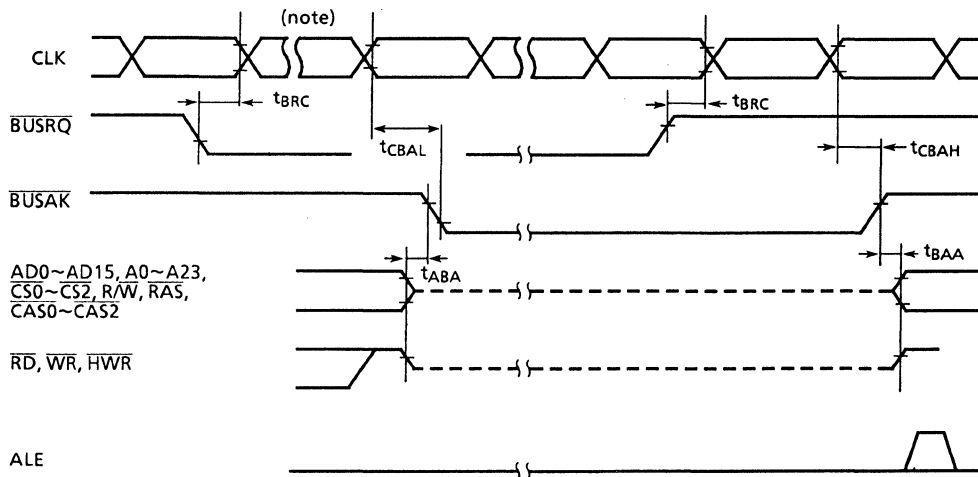
$$V_{CC} = 5V \pm 10\% \quad T_a = -20 \sim 70^\circ\text{C}$$

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{\text{INTAL}}$	NMI, INTO Low level pulse width	4x		250		200		ns
$t_{\text{INTAH}}$	NMI, INTO High level pulse width	4x		250		200		ns
$t_{\text{INTBL}}$	INT4 ~ INT7 Low level pulse width	$8x + 100$		600		500		ns
$t_{\text{INTBH}}$	INT4 ~ INT7 High level pulse width	$8x + 100$		600		500		ns

4.8 Timing Chart for I/O Interface Mode



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



Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{BRC}$	$\overline{\text{BUSRQ}}$ setup time for CLK	120		120		120		ns
$t_{CBAL}$	CLK→ $\overline{\text{BUSAK}}$ falling edge		$1.5x + 120$		214		195	ns
$t_{CBAH}$	CLK→ $\overline{\text{BUSAK}}$ rising edge		$0.5x + 40$		71		65	ns
$t_{ABA}$	Output buffer is off to $\overline{\text{BUSAK}}$	0	80	0	80	0	80	ns
$t_{BAA}$	$\overline{\text{BUSAK}}$ output buffer is on.	0	80	0	80	0	80	ns

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

Note 2: This line only shows the output buffer is off-states. They don't indicate the signal levels are fixed. After the bus is released, the signal level is kept dynamically before the bus is released by the external capacitance. Therefore, to fix the signal level by an external resistance under the bus is releasing, the design must be carefully because of the level-fix will be delayed. The internal programmable pull-up/pull-down resistance is switched active by the internal signal.

4.10 Interrupt Operation

$V_{CC} = 5V$ ,  $T_a = -25^{\circ}C$ , unless otherwise noted

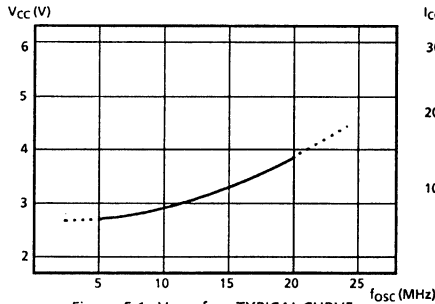


Figure 5.1  $V_{CC} - f_{osc}$  TYPICAL CURVE

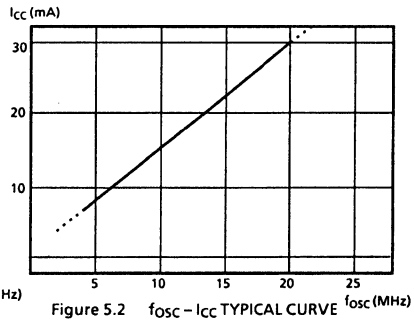


Figure 5.2  $f_{osc} - I_{CC}$  TYPICAL CURVE

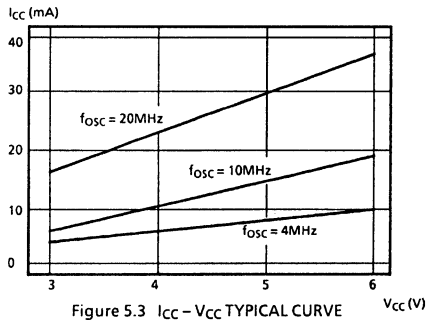


Figure 5.3  $I_{CC} - V_{CC}$  TYPICAL CURVE

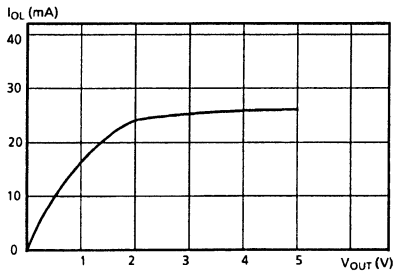


Figure 5.4  $V_{OUT} - I_{OL}$  TYPICAL CURVE

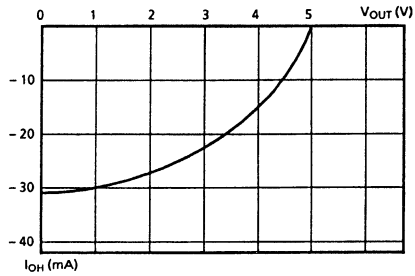


Figure 5.5  $V_{OUT} - I_{OH}$  TYPICAL CURVE

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

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

- (1) I/O port
- (2) I/O port control
- (3) Timer control
- (4) Pattern Generator control
- (5) Watch Dog Timer control
- (6) Serial Channel control
- (7) A/D converter control
- (8) Interrupt control
- (9) Chip Select/Wait Control

Configuration of the table

Symbol	Name	Address	7	6	5	4	3	2	1	0

→bit Symbol

→Read / Write

→Initial value afrer reset

→Remarks



Table 5 I/O Register Address Map

Address	Name	Address	Name	Address	Name	Address	Name
000000H	P0	20H	TRUN	40H	TREG6L	60H	ADREG0L
1H	P1	21H		41H	TREG6H	61H	ADREG0H
2H	POCR	22H	TREG0	42H	TREG7L	62H	ADREG1L
3H		23H	TREG1	43H	TREG7H	63H	ADREG1H
4H	P1CR	24H	TMOD	44H	CAP3L	64H	ADREG2L
5H	P1FC	25H	TFFCR	45H	CAP3H	65H	ADREG2H
6H	P2	26H	TREG2	46H	CAP4L	66H	ADREG3L
7H	P3	27H	TREG3	47H	CAP4H	67H	ADREG3H
8H	P2CR	28H	POMOD	48H	T5MOD	68H	BOCS
9H	P2FC	29H	P1MOD	49H	T5FFCR	69H	B1CS
AH	P3CR	2AH	PFFCR	4AH		6AH	B2CS
BH	P3FC	2BH		4BH		6BH	
CH	P4	2CH		4CH	PG0REG	6CH	
DH	P5	2DH		4DH	PG1REG	6DH	
EH	P4CR	2EH		4EH	PG01CR	6EH	
FH		2FH		4FH		6FH	
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H		31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BR0CR	73H	INTET10
14H	P6CR	34H	CAP1L	54H	SC1BUF	74H	INTEPW10
15H	P7CR	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P6FC	36H	CAP2L	56H	SC1MOD	76H	INTET76
17H	P7FC	37H	CAP2H	57H	BR1CR	77H	INTES0
18H	P8	38H	T4MOD	58H	ODE	78H	INTES1
19H	P9	39H	TFF4CR	59H		79H	
1AH	P8CR	3AH	T45CR	5AH		7AH	
1BH	P9CR	3BH		5BH		7BH	IIMC
1CH	P8FC	3CH		5CH	WDMOD	7CH	DMA0V
1DH	P9FC	3DH		5DH	WDCR	7DH	DMA1V
1EH		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		
P0	PORT0	00H	P07	P06	P05	P04	P03	P02	P01	P00		
			R/W									
			Input mode									
			Undefined									
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10		
			R/W									
			Input mode									
			0	0	0	0	0	0	0	0		
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20		
			R/W									
			Input mode									
			0	0	0	0	0	0	0	0		
P3	PORT3	07H	P37	P36	P35	P34	P33	P32	P31	P30		
			R/W									
			Input mode						Output mode			
			1	1	1	1	1	1	1	1		
P4	PORT4	0CH						P42	P41	P40		
			R/W									
			Input mode									
								0	1	1		
P5	PORT5	0DH					P53	P52	P51	P50		
			R									
			Input mode									
P6	PORT6	12H	P67	P66	P65	P64	P63	P62	P61	P60		
			R/W									
			Input mode									
			1	1	1	1	1	1	1	1		
P7	PORT7	13H					P73	P72	P71	P70		
			R/W									
			Input mode									
								1	1	1	1	
P8	PORT8	18H	P87	P86	P85	P84	P83	P82	P81	P80		
			R/W									
			Input mode									
			1	1	1	1	1	1	1	1		
P9	PORT9	19H			P95	P94	P93	P92	P91	P90		
			R/W									
			Input mode									
					1	1	1	1	1	1		

Note: When P30 pin is defined as  $\overline{RD}$  signal output mode (P30F = 1), clearing the output latch register P30 to "0" outputs the  $\overline{RD}$  strobe from P30 pin for PSRAM, even when the internal address is accessed. If the output latch register P30 remains "1", the  $\overline{RD}$  strobe is output only when the external address is accessed.

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

Note: With the TMP96C141A/TMP96C141A/TMP96C041A, which requires an external ROM, PORT0 functions as AD0 to AD7; PORT1, AD8 to AD15; P30, the RD signal; P31, the WR signal, regardless of the values set in P0CR, P1CR, P1FC, P30F and P31F.

I/O Port Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6CR	PORT6 Control	14H (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P7CR	PORT7 Control	15H (Prohibit RMW)					P73C	P72C	P71C	P70C
			W							
							0	0	0	0
			0: IN 1: OUT							
P6FC	PORT6 Function	16H (Prohibit RMW)	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			0	0	0	0	0	0	0	0
			0: PORT 1: PG1 - OUT				0: PORT 1: PGO - OUT			
P7FC	PORT7 Function	17H (Prohibit RMW)					P73F	P72F	P71F	
			W							
							0	0	0	
							0: PORT 1: TO3	0: PORT 1: TO2	0: PORT 1: TO1	
P8CR	PORT8 Control	1AH (Prohibit RMW)	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P9CR	PORT9 Control	1BH (Prohibit RMW)			P95C	P94C	P93C	P92C	P91C	P90C
			W							
					0	0	0	0	0	0
			0: IN 1: OUT							
P8FC	PORT8 Function	1CH (Prohibit RMW)		P86F			P83F	P82F		
				W			W	W		
				0			0	0		
				0: PORT 1: TO6			0: PORT 1: TO5	0: PORT 1: TO4		
P9FC	PORT9 Function	1DH (Prohibit RMW)			P95F		P93F	P92F		P90F
					W		W	W		W
					0		0	0		0
					0: PORT 1: SCLK1		0: PORT 1: TxD1	0: PORT 1: SCLK0		0: PORT 1: TxDO

(3) Timer Control (1/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0		
TRUN	Timer Control	20H	PRRUN		T5RUN	T4RUN	P1RUN	PORUN	T1RUN	TORUN		
			R/W		R/W							
			0		0	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									
TMOD	8bit Timer Source CLK and MODE	24H (Prohibit RMW)	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	TOCLK1	TOCLK0		
			W									
			0	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 <sup>6</sup> - 1 10 : 2 <sup>7</sup> - 1 11 : 2 <sup>8</sup> - 1	PWM	00 : T00TRG 01 : φT1 10 : φT16 11 : φT256	00 : T10 Input 01 : φT1 10 : φT4 11 : φT16					
TFFCR	8bit Timer Flip-flop Control	25H				DBEN	TFF1C1	TFF1C0	TFF1IE	TFF1IS		
						R/W	W		R/W			
						0	0	0	0	0		
						1 : Double Buffer Enable	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care	1 : TFF1 Invert Enable	0 : Inverted by Timer 0			
TREG2	PWM Timer Register 2	26H	-									
			(R)/W (Can read double buffer values.)									
			Undefined									
TREG3	PWM Timer Register 3	27H	-									
			(R)/W (Can read double buffer values.)									
			Undefined									
POMOD	PWM0 MODE	28H (Prohibit RMW)	FF2RD	DB2EN	PWM0INT	PWM0M	T2CLK1	T2CLK0	PWM0S1	PWM0S0		
			R	W								
			-	0	0	0	0	0	0	0		
			TFF2 output value	1 : Double Buffer Enable	0 : Overflow Interrupt 1 : Compare/Match Interrupt	0 : PWM Mode 1 : Timer Mode	00 : φP1(tc/4) 01 : φP4(tc/16) 10 : φP16(tc/64) 11 : Don't care	00 : 2 <sup>6</sup> - 1 01 : 2 <sup>7</sup> - 1 10 : 2 <sup>8</sup> - 1 11 : Don't care				
P1MOD	PWM1 MODE	29H (Prohibit RMW)	FF3RD	DB3EN	PWM1INT	PWM1M	T3CLK1	T3CLK0	PWM1S1	PWM1S0		
			R	W								
			-	0	0	0	0	0	0	0		
			TFF3 output value	1 : Double Buffer Enable	0 : Overflow Interrupt 1 : Compare/Match Interrupt	0 : PWM Mode 1 : Timer Mode	00 : φ P1(tc/4) 01 : φ P4(tc/16) 10 : φP16(tc/64) 11 : Don't care	00 : 2 <sup>6</sup> - 1 01 : 2 <sup>7</sup> - 1 10 : 2 <sup>8</sup> - 1 11 : Don't care				

Timer Control (2/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PFFCR	PWM Flip-flop Control	2AH	FF3C1	FF3C0	FF3TRG1	FF3TRG0	FF2C1	FF2C0	FF2TRG1	FF2TRG0
			W		R/W		W		R/W	
			0	0	0	0	0	0	0	0
			00 : Don't care 01 : Set TFF3 10 : Clear TFF3 11 : Don't care		00 : Prohibit TFF3 Inverted 01 : Invert if matched 10 : Set if matched; Clear if overflowed 11 : Clear if matched; set if overflowed		00 : Don't care 01 : Set TFF2 10 : Clear TFF2 11 : Don't care		00 : Prohibit TFF2 Inverted 01 : Invert if matched 10 : Set if matched; Clear if overflowed 11 : Clear if matched; set if overflowed	
TREG4L	16-bit Timer Register 4L	30H (Prohibit RMW)	-							
			W							
			Undefined							
TREG4H	16-bit Timer Register 4H	31H (Prohibit RMW)	-							
			W							
			Undefined							
TREG5L	16-bit Timer Register 5L	32H (Prohibit RMW)	-							
			W							
			Undefined							
TREG5H	16-bit 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	16-bit 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 ↑ T15 ↑ 10: T14 ↑ T14 ↓ 11: TFF1 ↑ TFF1 ↓		1: UC4 Clear Enable	Source Clock 00: T14 01: φT1 10: φT4 11: φT16	

Timer Control (3/4)

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	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	DB6EN	DB4EN	
			R/W				R/W				
			0				0	0	0	0	
			Fix at "0"				PG1 shift trigger 0 : Timer 0, 1 1 : Timer 5	PG0 shift trigger 0 : Timer 0, 1 1 : Timer 4	1 : Double Buffer Enable		
TREG6L	16bit Timer Register 6L	40H (Prohibit RMW)	-								
			W								
			Undefined								
TREG6H	16bit Timer Register 6H	41H (Prohibit RMW)	-								
			W								
			Undefined								
TREG7L	16bit Timer Register 7L	42H (Prohibit RMW)	-								
			W								
			Undefined								
TREG7H	16bit Timer Register 7H	43H (Prohibit RMW)	-								
			W								
			Undefined								
CAP3L	Capture Register 3L	44H	-								
			R								
			Undefined								
CAP3H	Capture Register 3H	45H	-								
			R								
			Undefined								
CAP4L	Capture Register 4L	46H	-								
			R								
			Undefined								
CAP4H	Capture Register 4H	47H	-								
			R								
			Undefined								
T5MOD	16bit Timer 5 Source CLK and MODE	48H			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0	
			R/W							W	
			0	0	0	0	0	0	0	0	0
					0 : Soft- Capture 1 : Don't care	Capture Timing 00 : Disable 01 : T16 ↑ T17 ↑ 10 : T16 ↑ T16 ↓ 11 : TFF1 ↑ TFF1 ↓			1 : UC5 Clear Enable	Source Clock 00 : Invert TFF6 01 : Set TFF6 10 : Clear TFF6 11 : Don't care	

Timer Control (4/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0	
T5FFCR	16bit Timer 5 Flip-flop Control	49H			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0	
			R/W							W	
					0	0	0	0	0	0	
			TFF6 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable							00 : Invert TFF6 01 : Set TFF6 10 : Clear TFF6 11 : Don't care	

(4) Pattern Generator

Symbol	Name	Address	7	6	5	4	3	2	1	0	
PG0REG	PG0 Register	4CH (Prohibit RMW)	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00	
			W				R/W				
			0	0	0	0	Undefined				
PG1REG	PG1 Register	4DH (Prohibit RMW)	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10	
			W				R/W				
			0	0	0	0	Undefined				
PG01CR	PG0, 1 Control	4EH (Prohibit RMW)	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE	
			R/W								
			0	0	0	0	0	0	0	0	
			0 : 8bit write 1 : 4bit write	0 : Normal Rotation 1 : Reverse Rotation	0 : 4bit Step 1 : 8bit Step	PG1 trigger input enable 1 : Enable	0 : 8bit write 1 : 4bit write	0 : Normal Rotation 1 : Reverse Rotation	0 : 4bit Step 1 : 8bit 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 <sup>16</sup> /fc 01 : 2 <sup>18</sup> /fc 10 : 2 <sup>20</sup> /fc 11 : 2 <sup>22</sup> /fc		Warming up Time 0 : 2 <sup>14</sup> /fc 1 : 2 <sup>16</sup> /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								
			-								
			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	0		
			Receiving data bit 8	Parity 0 : Odd 1 : Even	1 : Parity Enable	1 : Error			0: SCLK0 1: SCLK0	1: Input SCLK0 pin (Note)		
						Overrun	Parity	Framing				
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 7bit 10 : UART 8bit 11 : UART 9bit		00 : T00 Trigger 01 : Baud rate generator 10 : Internal clock $\phi$ 1 11 : Don't care			
BR0CR	Baud Rate Control	53H	-		BROCK1	BROCK0	BR053	BR052	BR051	BR050		
			R/W	R/W								
			0		0	0	0	0	0	0		
			Fix at "0"		00 : $\phi$ 10 (fc/4) 01 : $\phi$ 12 (fc/16) 10 : $\phi$ 18 (fc/64) 11 : $\phi$ 32 (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	0		
			Receiving data bit 8	Parity 0 : Odd 1 : Even	1 : Parity Enable	1 : Error			0: SCLK1 1: SCLK1	1: Input SCLK1 pin		
						Overrun	Parity	Framing				
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 7bit 10 : UART 8-bit 11 : UART 9bit		00 : T00 Trigger 01 : Baud rate generator 10 : Internal clock $\phi$ 1 11 : Don't care			

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 : $\phi$ 10 (fc/4) 01 : $\phi$ 2 (fc/16) 10 : $\phi$ 8 (fc/64) 11 : $\phi$ 32 (fc/256)	Set frequency divisor 0 - F ("1" prohibited)					
ODE	Special Open Drain Enable	58H	-						ODE1	ODE0	
									R/W		
									0	0	
									1 : P93 Open-drain	1 : P90 Open-drain	

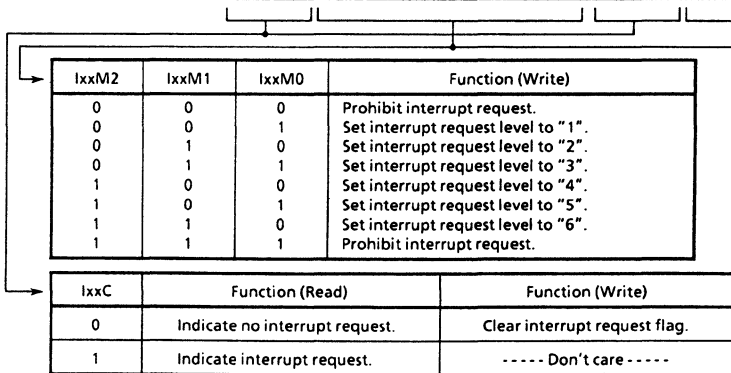
(7) A/D Converter Control

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

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

(8) Interrupt Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE-0AD	INTerrupt Enable 0 & A/D	70H (Prohibit RMW)	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE45	INTerrupt Enable 4/5	71H (Prohibit RMW)	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE67	INTerrupt Enable 6/7	72H (Prohibit RMW)	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE10	INTerrupt Enable Timer 1/0	73H (Prohibit RMW)	INTT1 (Timer 1)				INTT0 (Timer 0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE-PW10	INTerrupt Enable PWm 1/0	74H (Prohibit RMW)	INTT3 (Timer 3/PWM1)				INTT2 (Timer 2/PWM0)			
			IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE54	INTerrupt Enable Treg 5/4	75H (Prohibit RMW)	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE76	INTerrupt Enable Treg 7/6	76H (Prohibit RMW)	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE50	INTerrupt Enable Serial 0	77H (Prohibit RMW)	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE51	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



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	DMA0V5	DMA0V4			
			W							
			0	0	0	0	0			
DMA1V	DMA 1 request Vector	7DH (Prohibit RMW)	μDMA1 start vector							
			DMA01V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4			
			W							
			0	0	0	0	0			
DMA2V	DMA 2 request Vector	7EH (Prohibit RMW)	μDMA2 start vector							
			DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4			
			W							
			0	0	0	0	0			
DMA3V	DMA 3 request Vector	7FH (Prohibit RMW)	μDMA3 start vector							
			DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4			
			W							
			0	0	0	0	0			
IIMC	Interrupt Input Mode Control	7BH (Prohibit RMW)								
								IOIE	IOLE	NMIREE
								W	W	W
								0	0	0
								1: INTO input enable	0: INTO edge mode 1: INTO level mode	1: Operate even at NMI rise edge

(9) Chip Select/Wait Controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	68H (Prohibit RMW)	B0E	B0SYS	B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			1: CS Enable	1: SYSTEM only	0: $\overline{CS0}$ 1: $\overline{CAS0}$	0: 16bit Bus 1: 8bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT	00: 7F00H ~ 7FFFH 01: 400000H ~ 10: 800000H ~ 11: C00000H ~		
B1CS	Block 1 CS/WAIT control register	69H (Prohibit RMW)	B1E	B1SYS	B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			1: CS Enable	1: SYSTEM only	0: $\overline{CS1}$ 1: $\overline{CAS1}$	0: 16bit Bus 1: 8bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT	00: 480H ~ 7FFFH 01: 400000H ~ 10: 800000H ~ 11: C00000H ~		
B2CS	Block 2 CS/WAIT control register	6AH (Prohibit RMW)	B2E	B2SYS	B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			1: CS Enable	1: SYSTEM only	0: $\overline{CS2}$ 1: $\overline{CAS2}$	0: 16bit Bus 1: 8bit Bus	00: 2WAIT 01: 1WAIT 10: 1WAIT + n 11: 0WAIT	00: 8000H ~ 01: 400000H ~ 10: 800000H ~ 11: C00000H ~		

Note 1: After reset, only "Block 2" is set to enable.  
 → After reset, the program starts in 16-bit data bus, 2-wait state.

Note 2: These registers can be accessed XXXXXXXXXX

Note 3: TMP96C141A for internal RAM less is 80H ~ 7FFFH.

### 6. Port Section Equivalent Circuit Diagram

• Reading The Circuit Diagram

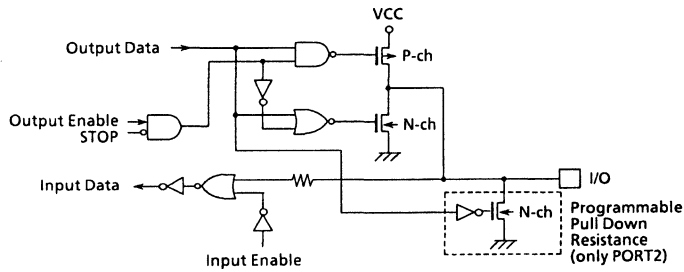
Basically, the gate singles written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The dedicated signal is described below.

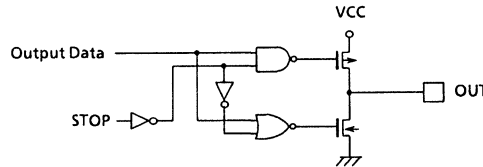
STOP: This signal becomes active "1" when the hold mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit [DRIVE] is set to "1", however, STP remains at "0".

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

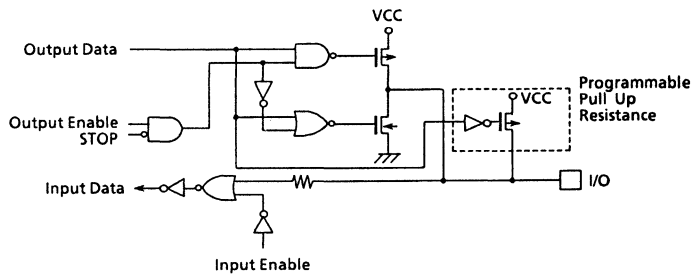
- PO (AD0 ~ AD7), P1 (AD8 ~ 15, A8 ~ 15), P2 (A2 - 23, A0 ~7)



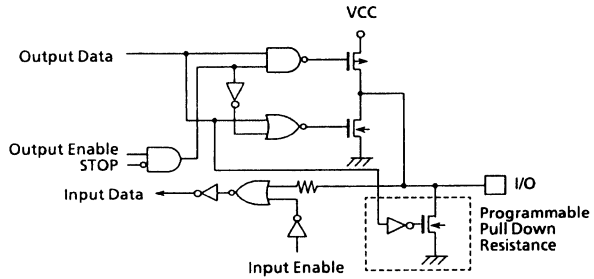
- P30 ( $\overline{RD}$ ), P31 ( $\overline{WR}$ )



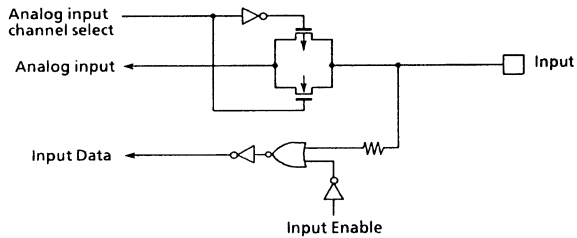
- P32 ~ 37, P40 ~ 41, P6, P7, P80 ~ 86, P91 ~ 92, P94 ~ 95



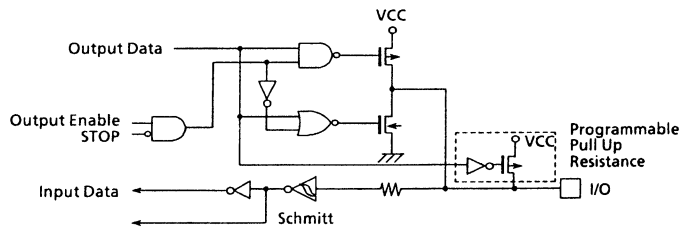
- P42 ( $\overline{CS2}$ ,  $\overline{CAS2}$ )



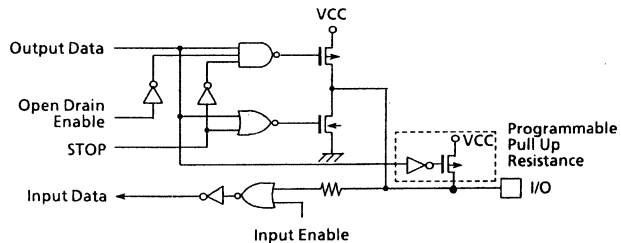
- P5 (AN0 ~ 3)



- P87 (INT0)



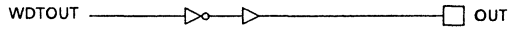
- P90 (TXD0), P93 (TXD1)



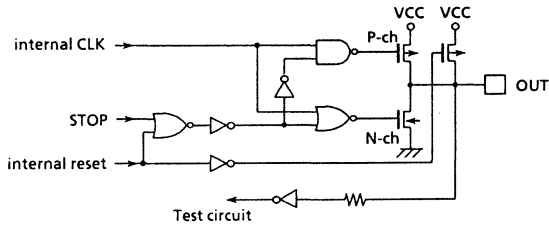
•  $\overline{\text{NMI}}$



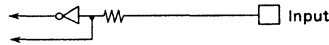
•  $\overline{\text{WDTOUT}}$



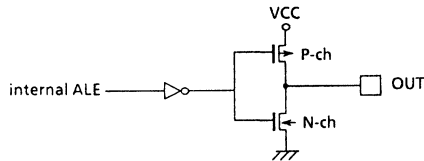
• CLK



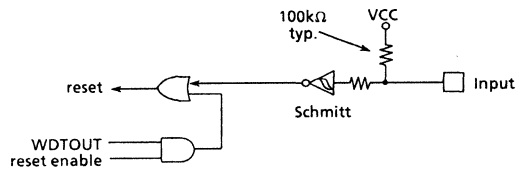
•  $\overline{\text{EA}}$ ,  $\text{AM8}/\overline{\text{T6}}$



• ALE

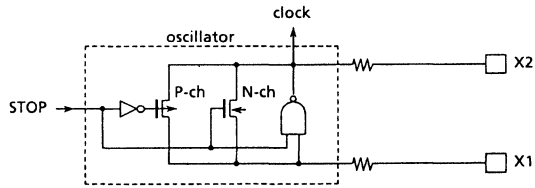


•  $\overline{\text{RESET}}$

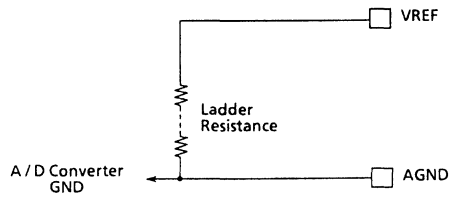




- X1, X2



- VREF, AGND



## 7. Care Points and Restriction

### (1) Special Expression

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

Symbol <Bit Symbol>

ex) TRUN <TRUN> ... Bit TORUN 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) The case of oscillation frequency is 20MHz.

### (2) Care Points

#### ① $\overline{EA}$ , 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 High Speed $\mu$ DMA (DRAM) refresh mode)

When the bus is released ( $\overline{BUSAK} = "0"$ ) for waiting to accept the interrupt, DRAM refresh is not performed because of the high speed  $\mu$ DMA is generated by an interrupt.

#### ④ 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.

#### ⑤ 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.

#### ⑥ Watch Dog Timer

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

#### ⑦ 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.

#### ⑧ CPU (High Speed $\mu$ DMA)

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.

### CMOS 16-bit Microcontrollers

#### TMP96C041AF

#### 1. Outline and Device Characteristics

The TMP96C041AF are high-speed advanced 16-bit micro-controllers developed for controlling medium to large-scale equipment. The TMP96C041AF has the improved bus release function, serial interface and RAMless for TMP96C141AF. Otherwise, the devices function in the same way.

The TMP96C041AF is housed in an 80-pin flat package and is pin compatible with TMP96C141F except the P92 (CTS0/SCLK0). Device characteristics are as follows:

- (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 (1.6 $\mu$ s/2 bytes @ 20MHz)
- (2) Minimum instruction execution time
  - 200ns @ 20MHz
- (3) Internal RAM: None  
Internal ROM: None
- (4) External memory expansion
  - Can be expanded up to 16M bytes (for both programs and data).
  - Can mix 8- and 16-bit external data buses.
- (5) 8-bit timers: 2 channels
- (6) 8-bit PWM timers: 2 channels
- (7) 16-bit timers: 2 channels
- (8) Pattern generators: 4 bits, 2 channels
- (9) Serial interface: 2 channels
- (10) 10-bit A/D converter: 4 channels
- (11) Watchdog timer
- (12) Chip select/wait controller: 3 blocks
- (13) Interrupt functions
  - 3 CPU interrupts: SWI instruction, privileged violation, and Illegal instruction
  - 14 internal interrupts
  - 6 external interrupts ] 7-level priority can be set.
- (14) I/O ports
  - 47pins
- (15) Standby function : 3 halt modes (RUN, IDLE, STOP)

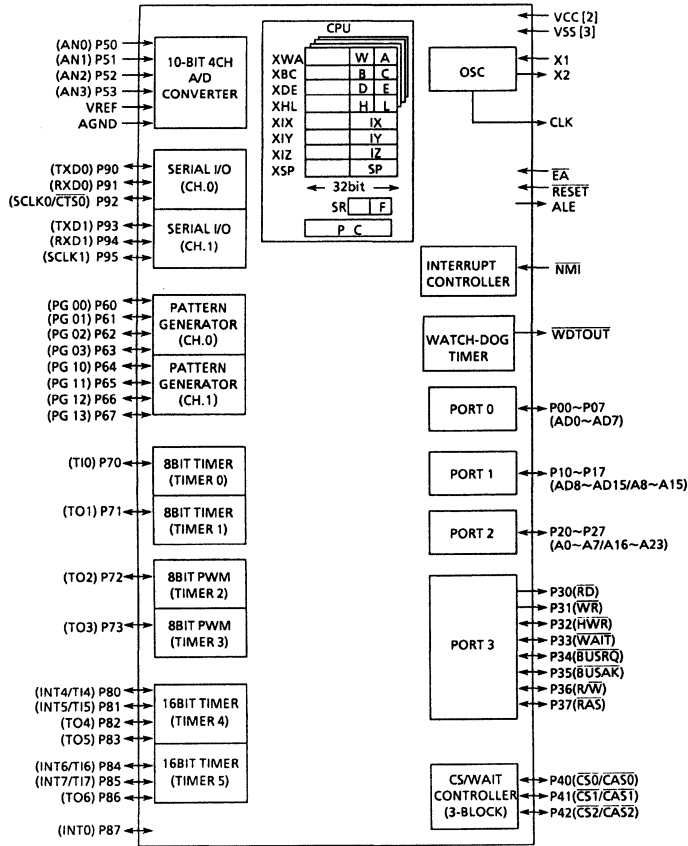


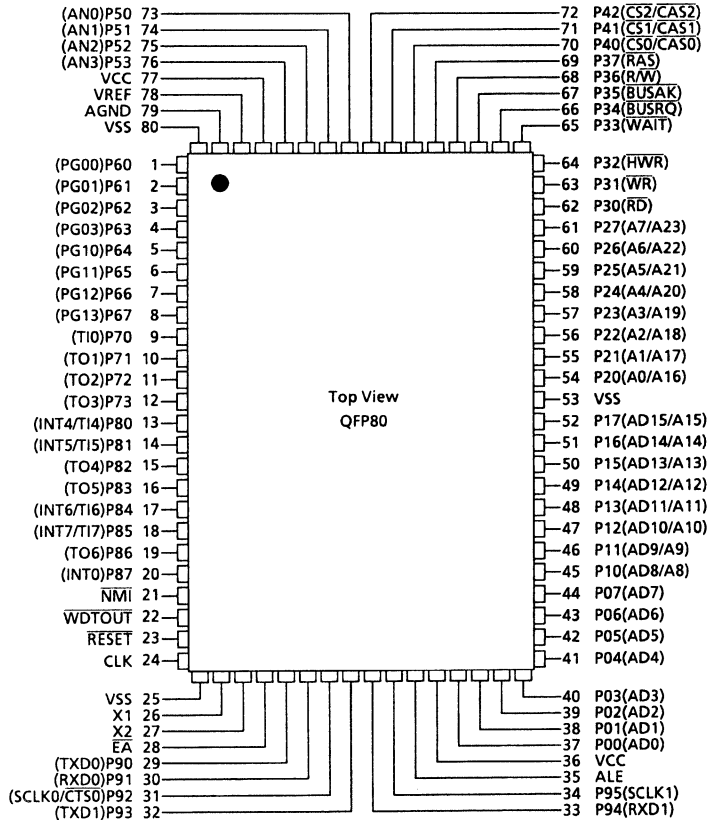
Figure 1. TMP96C041AF Block Diagram

**2. Pin Assignment and Functions**

The assignment of input/output pins for TMP96C041AF, their name and outline functions are described below.

**2.1 Pin Assignment**

Figure 2.1 shows pin assignment of TMP96C041AF.



Note : Because the TMP96C041AF has an external ROM, P00 to P17 pins are fixed to AD0 to AD15; P30 to RD; and P31 to WR.

**Figure 2.1. Pin Assignment (80-pin QFP)**

## 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
P00 – P07 AD0 – AD7	8	I/O Tri-state	Port 0: I/O port that allows I/O to be selected on a bit basis Address / data (lower): 0 - 7 for address / data bus
P10 – P17 AD8 – AD15 A8 – A15	8	I/O Tri-state Output	Port 1: I/O port that allows I/O to be selected on a bit basis 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 RD	1	Output Output	Port 30: Output port Read: Strobe signal for reading external memory
P31 WR	1	Output Output	Port 31: Output port Write: Strobe signal for writing data on pins AD0 - 7
P32 HWR	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 - 15
P33 WAIT	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P34 BUSRQ	1	I/O Input	Port 34: I/O port (with pull-up resistor) Bus request: Signal used to request high impedance for AD0 - 15, A0 - 23, RD, WR, HWR, R/W, RAS, CS0, CS1, and CS2 pins. (For external DMAC)
P35 BUSAK	1	I/O Output	Port 35: I/O (with pull-up resistor) Bus acknowledge: Signal indicating that AD0 - 15, A0 - 23, RD, WR, HWR, R/W, RAS, CS0, CS1, and CS2 pins are at high impedance after receiving BUSRQ. (For external DMAC)
P36 R/W	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0, write cycle.
P37 RAS	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs RAS strobe for DRAM.
P40 CS0 CAS0	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs CAS strobe for DRAM when address is within specified address area.

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

Pin Name	Number of Pins	I/O	Functions
P41 <u>CS1</u> CAS1	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs CAS strobe for DRAM if address is within specified address area.
P42 <u>CS2</u> CAS2	1	I/O Output Output	Port 42: I/O port (with pull-up resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs CAS strobe for DRAM if address is within specified address area.
P50 ~ P53 AN0 ~ AN3	4	Input Input	Port 5: Input port Analog input: Input to A/D converter
VREF	1	Input	Pin for reference voltage input to A/D converter
AGND	1	Input	Ground pin for A/D converter
P60 ~ P63 PG00 ~ PG03	4	I/O Output	Ports 60 - 63: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 00 - 03
P64 ~ P67 PG10 ~ PG13	4	I/O Output	Ports 64 - 67: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 10 - 13
P70 T10	1	I/O Input	Port 70: I/O port (with pull-up resistor) Timer input 0: Timer 0 input
P71 T01	1	I/O Output	Port 71: I/O port (with pull-up resistor) Timer output 1: Timer 0 or 1 output
P72 T02	1	I/O Output	Port 72: I/O port (with pull-up resistor) PWM output 2: 8-bit PWM timer 2 output
P73 T03	1	I/O Output	Port 73: I/O port (with pull-up resistor) PWM output 3: 8-bit PWM timer 3 output
P80 T14 INT4	1	I/O Input Input	Port 80: I/O port (with pull-up resistor) Timer input 4: Timer 4 count/capture trigger signal input Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge
P81 T15 INT5	1	I/O Input Input	Port 81: I/O port (with pull-up resistor) Timer input 5: Timer 4 count/capture trigger signal input Interrupt request pin 5: Interrupt request pin with rising edge
P82 T04	1	I/O Output	Port 82: I/O port (with pull-up resistor) Timer output 4: Timer 4 output pin
P83 T05	1	I/O Output	Port 83: I/O port (with pull-up resistor) Timer output 5: Timer 4 output pin

Note 1: Case of the settable CS2 and CAS2; when TMP96C041AF is bus release, this pin is not added the internal pull-down resistor but is added the internal pull-up resistor.

Pin Name	Number of Pins	I/O	Functions
P84 TI6 INT6	1	I/O Input Input	Port 84: I/O port (with pull-up resistor) Timer input 6: Timer 5 count/capture trigger signal input Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
P85 TI7 INT7	1	I/O Input Input	Port 85: I/O port (with pull-up resistor) Timer input 7: Timer 5 count/capture trigger signal input Interrupt request pin 7: Interrupt request pin with rising edge
P86 TO6	1	I/O Output	Port 86: I/O port (with pull-up resistor) Timer output 6: Timer 5 output pin
P87 INT0	1	I/O Input	Port 87: I/O port (with pull-up resistor) Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P90 TXD0	1	I/O Output	Port 90: I/O port (with pull-up resistor) Serial send data 0
P91 RXD0	1	I/O Input	Port 91: I/O port (with pull-up resistor) Serial receive data 0
P92 CTS0 SCLK0	1	I/O Input	Port 92: I/O port (with pull-up resistor) Serial data send enable 0 (Clear to Send) Serial clock I/O 0
P93 TXD1	1	I/O Output	Port 93: I/O port (with pull-up resistor) Serial send data 1
P94 RXD1	1	I/O Input	Port 94: I/O port (with pull-up resistor) Serial receive data 1
P95 SCLK1	1	I/O I/O	Port 95: I/O port (with pull-up resistor) Serial clock I/O 1
WDTOUT	1	Output	Watchdog timer output pin
NMI	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output: Outputs $[X1 + 4]$ clock. Pulled-up during reset.
EA	1	Input	External access: 0 should be inputted with TMP96C041AF.
ALE	1	Output	Address latch enable
RESET	1	Input	Reset: Initializes LSI. (With pull-up resistor)
X1/X2	2	I/O	Oscillator connecting pin
VCC	2		Power supply pin (+5V)
VSS	3		GND pin (0V)

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



### 3. Operation

This section describes in blocks the functions and basic operations of the TMP96C041AF device.

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

#### 3.1 CPU

The TMP96C041AF device has a built-in high-performance 16-bit CPU. (For CPU operation, see TLCS-900 CPU in the book Core Architecture User Manual.)

#### 3.2 Memory Map

The TMP96C041AF has two register modes. One is minimum mode; in this mode, the area of program memory is 64K bytes maximum. The other is maximum mode; in this mode, the area of the program memory is 16M bytes maximum.

Both minimum and maximum modes are the data memory area 16M bytes maximum.

That is, the program memory can locate 0H ~ FFFFFFFH in maximum mode.

##### (1) Internal I/O Devices

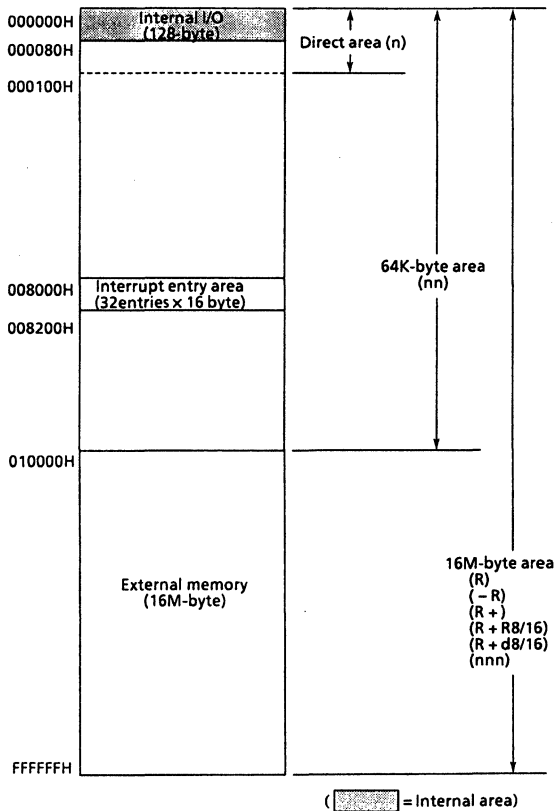
The TMP96C041AF uses the address space of 128 bytes for the internal I/O devices area. This area is located to 0H ~ 7FH.

The CPU can access the internal I/O devices area with using short instruction code of direct addressing mode.

Check the access area of each addressing mode and the memory map in Fig. 3.2 (1).

**3.2 Memory Map**

Figure 3.2 is a memory map of the TMP96C041AF.



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**

(2)  $\overline{CS1}$  Area (Chip Select/Wait Controller)

address area for  $\overline{CS1}$  (only B1C1.0 = "00"). Show the address area of  $\overline{CS1}$  in Fig. 3.2 (2).

The TMP96C041AF is expanded the part of the

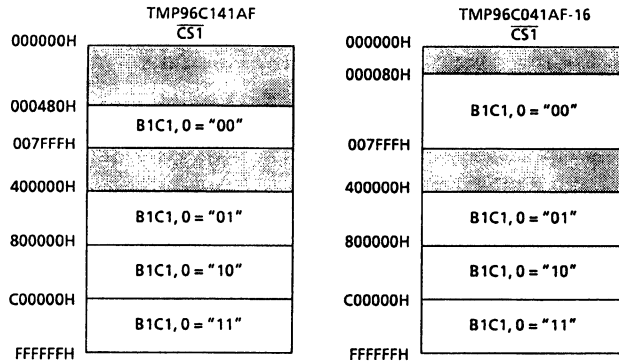


Fig. 3.2 (2).  $\overline{CS1}$  Address Area

3.3 Bus Release Function

The TMP96C041AF has the internal pull-up and pull down resistors to fix the bus control signals at bus release.

Show the table 3.3 of pin condition at bus release ( $\overline{BUSAK} = 0$ ).

Table 3.3 Pin Condition at Bus Release ( $\overline{BUSAK} = "L"$ )

Pin Name	Pin state at bus release	
	Port Mode	Function Mode
P00 - P07 (AD0 - AD7) P10 - P17 (AD8 - 15/AD8 - 15)	The status is no-change (these pins are not "Hz")	These pins are "Hz".
P30 ( $\overline{RD}$ ) P31 ( $\overline{WR}$ )	↑	These pins are "Hz". ("Hz" stays after these pins driven high level)
P32 ( $\overline{HWR}$ ) P37 ( $\overline{RAS}$ )	↑	The output buffer is "OFF" after these pins driven high. These pins are added the internal resistor of pull-up. It's no relation for the value of output latch.
P36 ( $\overline{R/\overline{W}}$ ) P40 ( $\overline{CS0/CAS0}$ ) P41 ( $\overline{CS1/CAS1}$ )	↑	↑
P20 - P27 (A16 - A23) P42 ( $\overline{CS2/CAS2}$ )	↑	(*) ↑
P20 - P27 (A16 - A23)	↑	The output buffer is "OFF" after these pins driven low. These pins are added the internal resistor of pull-up. It's no relation for the value of output latch.

That is, when it is used for bus release ( $\overline{BUSAK} = 0$ ), the pins of below need pull-up or pull-down resistor for an external circuit.

- P00 - P07 (AD0 AD7)
- P10 - P17 (AD8 AD15)

- P30 ( $\overline{RD}$ )
- P31 ( $\overline{WR}$ )

(\*) P42 has the resistor of programmable pull-down, but when the bus are released, P42 pin is added a resistor of pull-up.

For the bus release function; show a sample of external bus interface in the Figure 3.3 (1).  
 When the bus is released, both internal memory and internal I/O cannot be accessed. But the internal I/O continues

to run. So, the watchdog timer also continues to run. Therefore, be careful about bus releasing time and set the detection time of WDT.

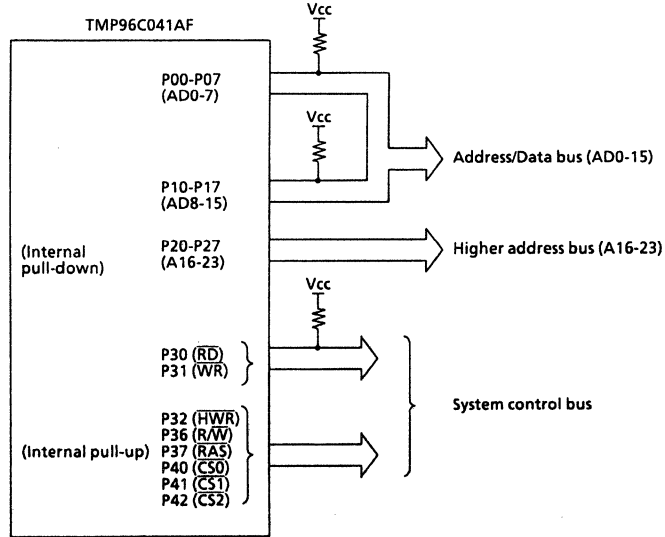


Figure 3.3 (1). Example of the Interface Circuit (Bus Releasing Function)

### 3.4 Serial Function

The TMP96C041AF has two Serial I/O devices. But channel 0 and channel 1 are same function except the handshake (CTS0

pin) function of the channel 0 and can use I/O interface mode. Show the part of TMP96C41AF in detail.

## 4. Electrical Characteristics

### 4.1 Absolute Maximum (TMP96C041AF)

Symbol	Parameter	Rating	Unit
$V_{CC}$	Power Supply Voltage	-0.5 ~ 6.5	V
$V_{IN}$	Input Voltage	-0.5 ~ $V_{CC} + 0.5$	V
$\Sigma I_{OL}$	Output Current (total)	100	mA
$\Sigma I_{OH}$	Output Current (total)	-100	mA
PD	Power Dissipation ( $T_a = 70^\circ\text{C}$ )	600	mW
T SOLDER	Soldering Temperature (10s)	260	$^\circ\text{C}$
T STG	Storage Temperature	-65 ~ 150	$^\circ\text{C}$
T OPR	Operating Temperature	-20 ~ 70	$^\circ\text{C}$

## 4.2 DC Characteristics (TMP96C041AF)

$V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ\text{C}$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ\text{C}$  (4 ~ 20MHz)  
 (Typical values are for  $T_A = 25^\circ\text{C}$  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	RESET, NMI, INTO (P87)	-0.3	$0.25V_{CC}$	V	
V IL3	$\overline{EA}$	-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	RESET, 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	IOL = 1.6mA
V OH	Output High Voltage	2.4		V	IOH = -400 $\mu$ A
V OH1		$0.75V_{CC}$		V	IOH = -100 $\mu$ A
V OH2		$0.9V_{CC}$		V	IOH = -20 $\mu$ A
I DAR	Darlington Drive Current (8 Output Pins max.)	-1.0	-3.5	mA	VEXT = 1.5V REXT = 1.1K $\Omega$
I LI	Input Leakage Current	TBD (Typ)	$\pm 5$	$\mu$ A	$0.0 \leq V_{in} \leq V_{CC}$
I LO	Output Leakage Current	TBD (Typ)	$\pm 10$	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
I <sub>CC</sub>	Operating Current (RUN)	TBD (Typ)	TBD	mA	$t_{osc} = 20\text{MHz}$
	IDLE	TBD (Typ)	10	mA	
	STOP ( $T_A = -20 \sim 70^\circ\text{C}$ )	TBD (Typ)	50	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
	STOP ( $T_A = 0 \sim 50^\circ\text{C}$ )	TBD (Typ)	10	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
V STOP	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	V IL2 = $0.2V_{CC}$ , V IH2 = $0.8V_{CC}$
R RST	RESET Pull Up Register	50	150	K $\Omega$	
C IO	Pin Capacitance		10	pF	$t_{osc} = 1\text{MHz}$
V TH	Schmitt Width RESET, NMI, INTO (P87)	0.4	TBD (Typ)	V	
R K	Pull Down/Up Register	50	150	K $\Omega$	

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

4.3 AC Electrical Characteristics (TMP96C041AF)

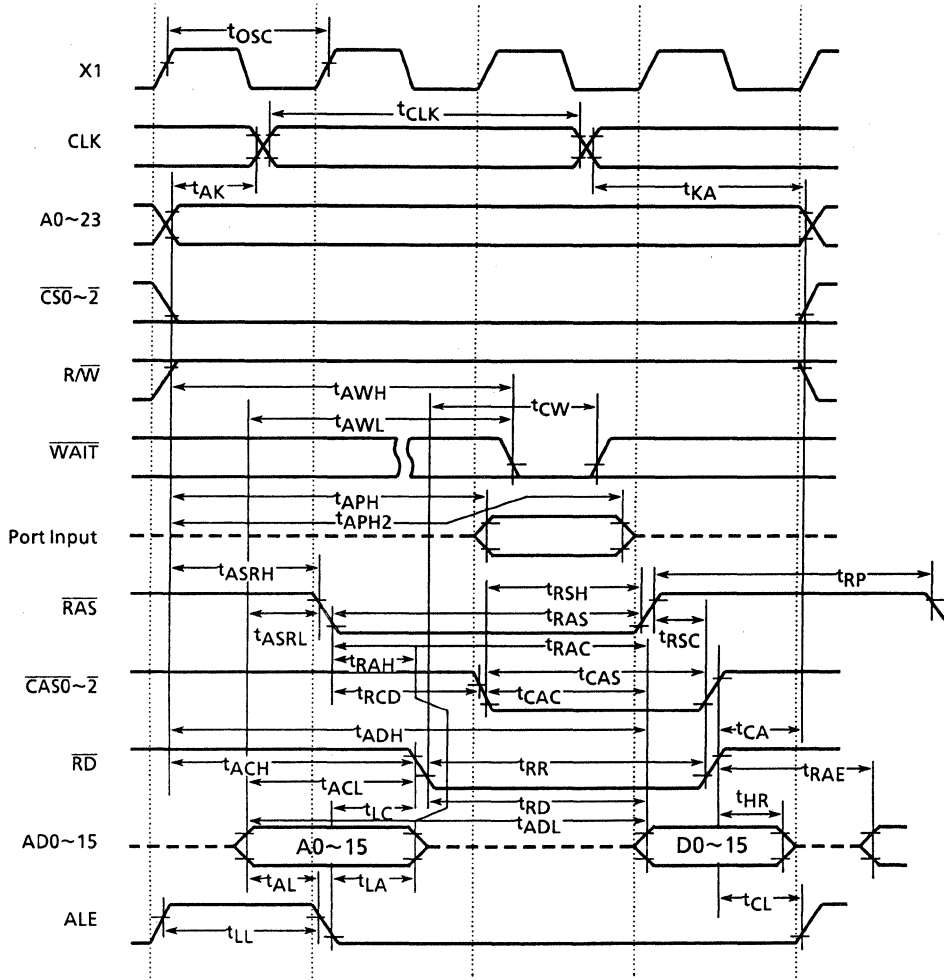
V<sub>CC</sub> = 5V±10% TA = -40 ~ 85°C (4 ~ 16MHz) TA = -20 ~ 70°C (4MHz ~ 20MHz)

No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t <sub>OSC</sub>	Osc. Period (= x)	50	250	62.5		50		ns
2	t <sub>CLK</sub>	CLK width	2x - 40		85		60		ns
3	t <sub>AK</sub>	A0 - 23 Valid→CLK Hold	0.5x - 20		11		5		ns
4	t <sub>KA</sub>	CLK Valid→A0 - 23 Hold	1.5x - 70		24		5		ns
5	t <sub>AL</sub>	A0-15 Valid→ALE fall	0.5x - 15		16		10		ns
6	t <sub>LA</sub>	ALE fall→A0 - 15 Hold	0.5x - 15		16		10		ns
7	t <sub>LL</sub>	ALE High width	x - 40		23		10		ns
8	t <sub>LC</sub>	ALE fall→RD/W <sub>R</sub> fall	0.5x - 30		1		-5		ns
9	t <sub>CL</sub>	R <sub>D</sub> /W <sub>R</sub> rise→ALE rise	0.5x - 20		11		5		ns
10	t <sub>ACL</sub>	A0 - 15 Valid→R <sub>D</sub> /W <sub>R</sub> fall	x - 25		38		25		ns
11	t <sub>ACH</sub>	A0 - 23 Valid→R <sub>D</sub> /W <sub>R</sub> fall	1.5x - 50		44		25		ns
12	t <sub>CA</sub>	R <sub>D</sub> /W <sub>R</sub> rise→A0 - 23 Hold	0.5x - 20		11		5		ns
13	t <sub>ADL</sub>	A0 - 15 Valid→D0 - 15 input		3.0x - 45		143		105	ns
14	t <sub>ADH</sub>	A0 - 23 Valid→D0 - 15 input		3.5x - 65		154		110	ns
15	t <sub>RD</sub>	R <sub>D</sub> fall→D0 - 15 input		2.0x - 50		75		50	ns
16	t <sub>RR</sub>	R <sub>D</sub> Low width	2.0x - 40		85		60		ns
17	t <sub>HR</sub>	R <sub>D</sub> rise→D0 - 15 Hold	0		0		0		ns
18	t <sub>RAE</sub>	R <sub>D</sub> rise→A0 - 15 output	x - 15		48		35		ns
19	t <sub>WW</sub>	W <sub>R</sub> Low width	2.0x - 40		85		60		ns
20	t <sub>DW</sub>	D0 - 15 Valid→W <sub>R</sub> rise	2.0x - 50		75		50		ns
21	t <sub>WD</sub>	W <sub>R</sub> rise→D0 - 15 Hold	0.5x - 10		21		15		ns
22	t <sub>AEH</sub>	A0 - 23 Valid→W <sub>AIT</sub> input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	t <sub>AWL</sub>	A0 - 15 Valid→W <sub>AIT</sub> input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	t <sub>CW</sub>	R <sub>D</sub> /W <sub>R</sub> fall→W <sub>AIT</sub> Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	t <sub>APH</sub>	A0 - 23 Valid→PORT input		2.5x - 120		80		36	ns
26	t <sub>APH2</sub>	A0 - 23 Valid→PORT Hold	2.5x + 50		206		175		ns
27	t <sub>CP</sub>	W <sub>R</sub> rise→PORT Valid		200		200		200	ns
28	t <sub>ASRH</sub>	A0 - 23 Valid→R <sub>AS</sub> fall	1.0x - 40		23		10		ns
29	t <sub>ASRL</sub>	A0 - 15 Valid→R <sub>AS</sub> fall	0.5x - 15		16		10		ns
30	t <sub>RAC</sub>	R <sub>AS</sub> fall→D0 - 15 input		2.5x - 70		130		86	ns
31	t <sub>RAH</sub>	R <sub>AS</sub> fall→A0 - 15 Hold	0.5x - 15		16		10		ns
32	t <sub>RAS</sub>	R <sub>AS</sub> Low width	2.0x - 40		85		60		ns
33	t <sub>RP</sub>	R <sub>AS</sub> High width	2.0x - 40		85		60		ns
34	t <sub>RSH</sub>	C <sub>AS</sub> fall→R <sub>AS</sub> rise	1.0x - 35		28		15		ns
35	t <sub>RSC</sub>	R <sub>AS</sub> rise→C <sub>AS</sub> rise	0.5x - 25		6		0		ns
36	t <sub>RCD</sub>	R <sub>AS</sub> fall→C <sub>AS</sub> fall	1.0x - 40		23		10		ns
37	t <sub>CAC</sub>	C <sub>AS</sub> fall→D0 - 15 input		1.5x - 65		29		10	ns
38	t <sub>CAS</sub>	C <sub>AS</sub> Low width	1.5x - 30		64		40		ns
39	t <sub>DS</sub>	D0 - 15 valid →C <sub>AS</sub> fall	0.5x - 15		16		10		ns

AC Measuring Conditions

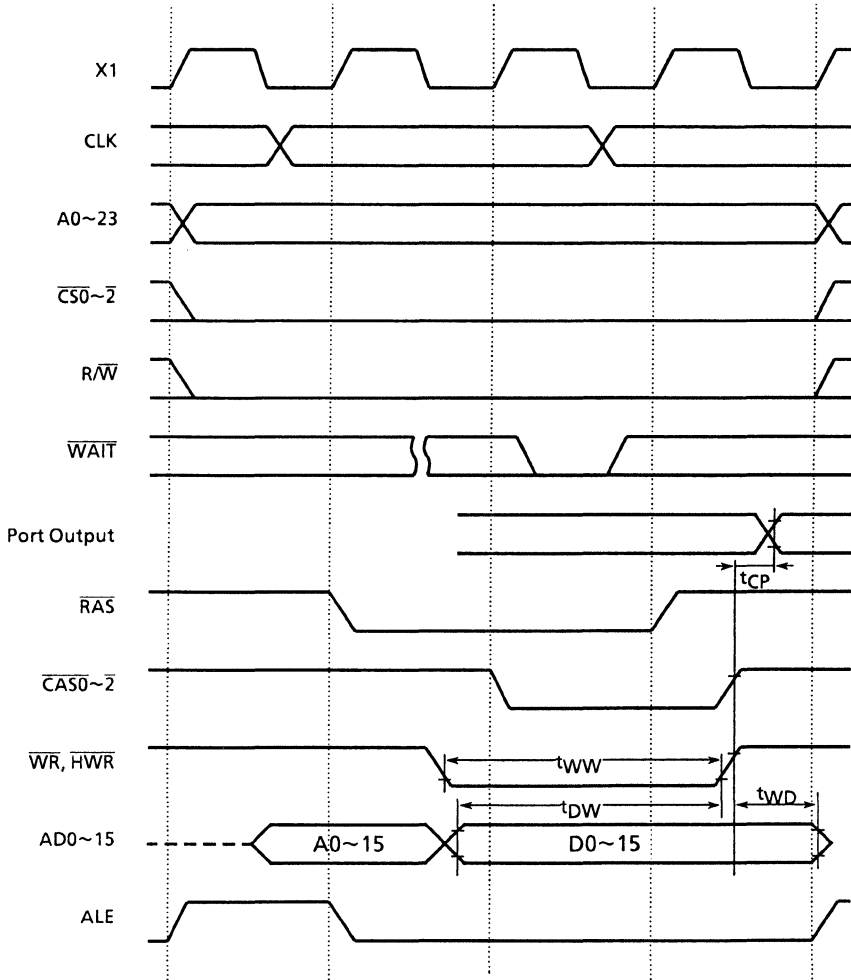
- Output Level: High 2.2V /Low 0.8V, CL50pF  
(However CL = 100pF for AD0 ~ AD15, AD0 ~ AD23, ALE, RD, WR, FWR, R/W, CLK, RAS, CAS0 ~ CAS2)
- Input Level: High 2.4V /Low 0.45V (AD0 ~ AD15)  
High 0.8V<sub>CC</sub> /Low 0.2V<sub>CC</sub> (Except for AD0 ~ AD15)

(1) Read Cycle





(2) Write Cycle



## 4.4 A/D Conversion Characteristics (TMP96C041AF)

 $V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ\text{C}$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ\text{C}$  (4 ~ 20MHz)

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			TBD	1.5	mA
Error (Quantize error of $\pm 0.5$ LSB not included)	$4 \leq f_c \leq 16\text{MHz}$	High speed conversion mode		TBD	$\pm 4.0$	LSB
		Low speed conversion mode		TBD	$\pm 6.0$	
	$16 \leq f_c \leq 20\text{MHz}$	High speed conversion mode		TBD	$\pm 4.0$	
		Low speed conversion mode		TBD	$\pm 6.0$	

## 4.5 Serial Channel Timing - I/O Interface Mode

 $V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ\text{C}$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ\text{C}$  (4 ~ 20MHz)

## (1) SCLK Input Mode

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

 $V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ\text{C}$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ\text{C}$  (4 ~ 20MHz)

## (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	$\mu\text{s}$
$t_{OSS}$	Output Data $\rightarrow$ Rising edge of SCLK	$t_{SCY} - 2x - 150$		725		550		ns
$t_{OHS}$	SCLK rising edge $\rightarrow$ Output Data hold	$2x - 80$		45		20		ns
$t_{HSR}$	SCLK rising edge $\rightarrow$ Input Data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge $\rightarrow$ effective data input		$t_{SCY} - 2x - 150$		725		550	ns

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

 $V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ\text{C}$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ\text{C}$  (4 ~ 20MHz)

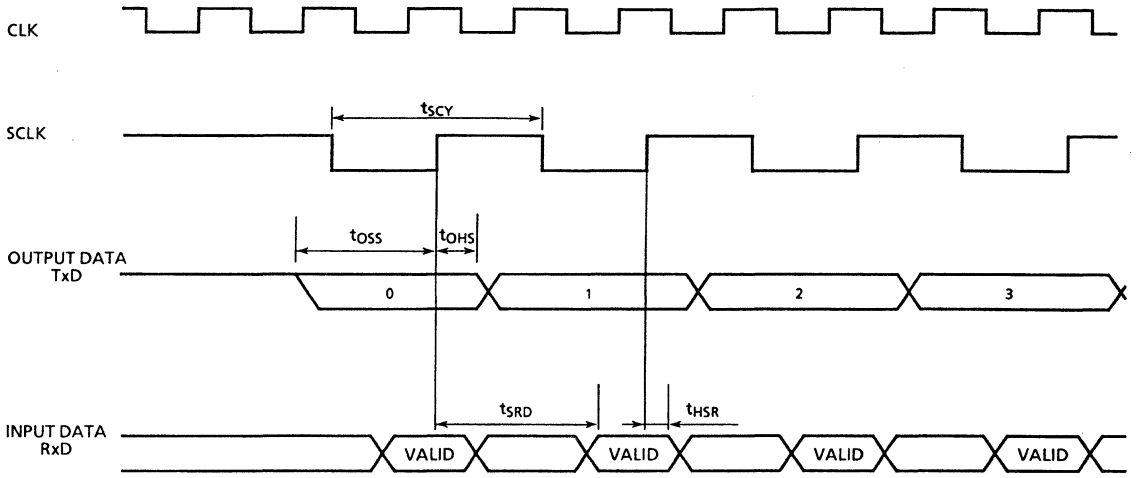
Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{VCK}$	Clock cycle	$8x + 100$		600		500		ns
$t_{VCKL}$	Low level clock pulse width	$4x + 40$		290		240		ns
$t_{VCKH}$	High level clock pulse width	$4x + 40$		290		240		ns

4.7 Interrupt Operation

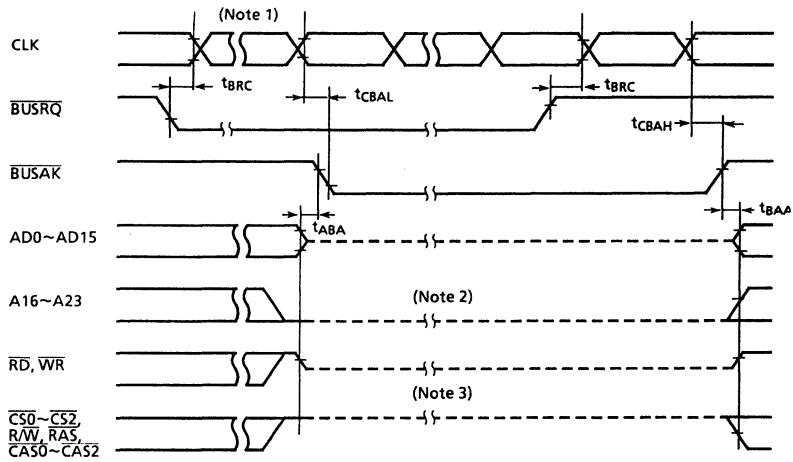
$V_{cc} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ C$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ C$  (4 ~ 20MHz)

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
$t_{INTBL}$	INT4 - INT7 Low level pulse width	$8x + 100$		600		500		ns
$t_{INTBH}$	INT4 - INT7 High level pulse width	$8x + 100$		600		500		ns

4.8 Timing Chart for I/O Interface Mode



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



Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{\text{BRC}}$	$\overline{\text{BUSRQ}}$ setup time for CLK	120		120		120		ns
$t_{\text{CBAL}}$	CLK→ $\overline{\text{BUSAK}}$ falling edge		$1.5x + 120$		245		220	ns
$t_{\text{CBAH}}$	CLK→ $\overline{\text{BUSAK}}$ rising edge		$0.5x + 40$		71		65	ns
$t_{\text{ABA}}$	Output buffer is off to $\overline{\text{BUSAK}}$	0	80	0	80	0	80	ns
$t_{\text{BAA}}$	$\overline{\text{BUSAK}}$ output buffer is on.	0	80	0	80	0	80	ns

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

Note 2: The internal programmable pull-up/pull-down resistance is added.

The internal programmable pull-up/pull-up resistance is added.

But the CS2/CAS2 pin does not have the internal programmable pull-up resistor. And in the condition of release, this pin is added the internal pull-up resistor.

## 5. Differences Between TMP96C141AF and TMP96C041AF

The devices TMP96C141AF and TMP96C041AF have much the same function, but they are different from following points.

Parameter	TMP96C141AF	TMP96C041AF
Internal RAM	1K byte	Does not exist
Pin condition at bus release	TMP96C141AF see Figure 3.3	TMP96C041AF see Figure 3.3
Mapping area of CS1 default setting (B1C1/0: 00)	480H ~ 7FFFH	80H ~ 7FFFH

### CMOS 16-bit Microcontrollers

#### TMP96CM40F

#### 1. Outline and Device Characteristics

The TMP96CM40F are high-speed advanced 16-bit micro-controllers developed for controlling medium to large-scale equipment. The TMP96C141AF does not have a ROM, the TMP96CM40F has a built-in ROM of 32K-byte, and the TMP96CPM40F has a built-in OTP of 32K-byte. TMP96CM40F is housed in an 80-pin flat package.

Device characteristics are as follows:

- (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 (1.6 $\mu$ s/2 bytes @ 20MHz)
- (2) Minimum instruction execution time
  - 200ns @ 20MHz
- (3) Internal RAM: 1Kbyte
  - Internal ROM: 32Kbyte
- (4) External memory expansion
  - Can be expanded up to 16M bytes (for both programs and data).
  - Can mix 8- and 16-bit external data buses.
- (5) 8-bit timers: 2 channels
- (6) 8-bit PWM timers: 2 channels
- (7) 16-bit timers: 2 channels
- (8) Pattern generators: 4 bits, 2 channels
- (9) Serial interface: 2 channels
- (10) 10-bit A/D converter: 4 channels
- (11) Watchdog timer
- (12) Chip select/wait controller: 3 blocks
- (13) Interrupt functions
  - 3 CPU interrupts... SWI instruction, privileged violation, and illegal instruction
  - 14 internal interrupts    7-level priority can be set.
  - 6 external interrupts
- (14) I/O ports
  - 65 pins
- (15) Standby function : 3 halt modes (RUN, IDLE, STOP)

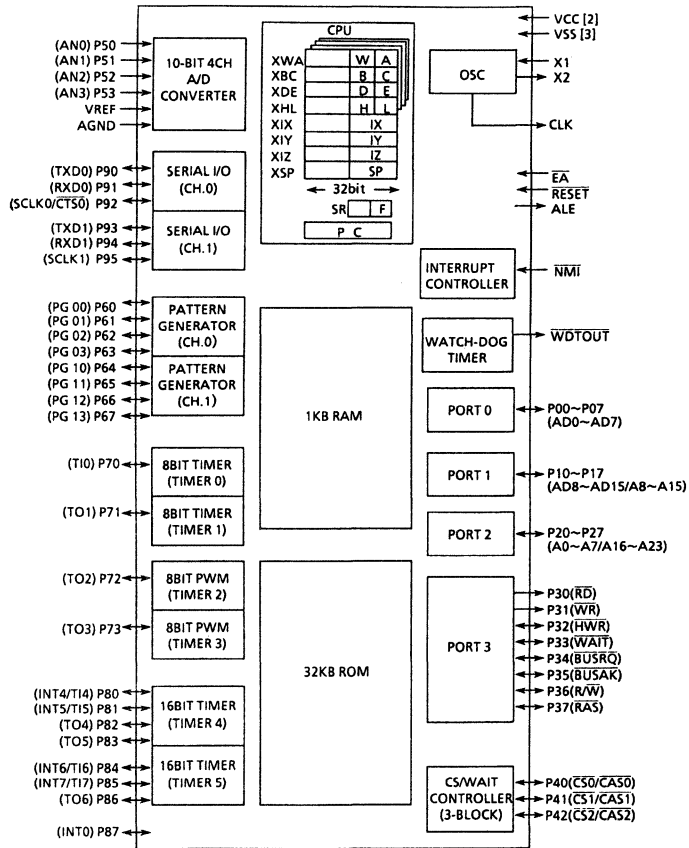


Figure 1. TMP96CM40F Block Diagram

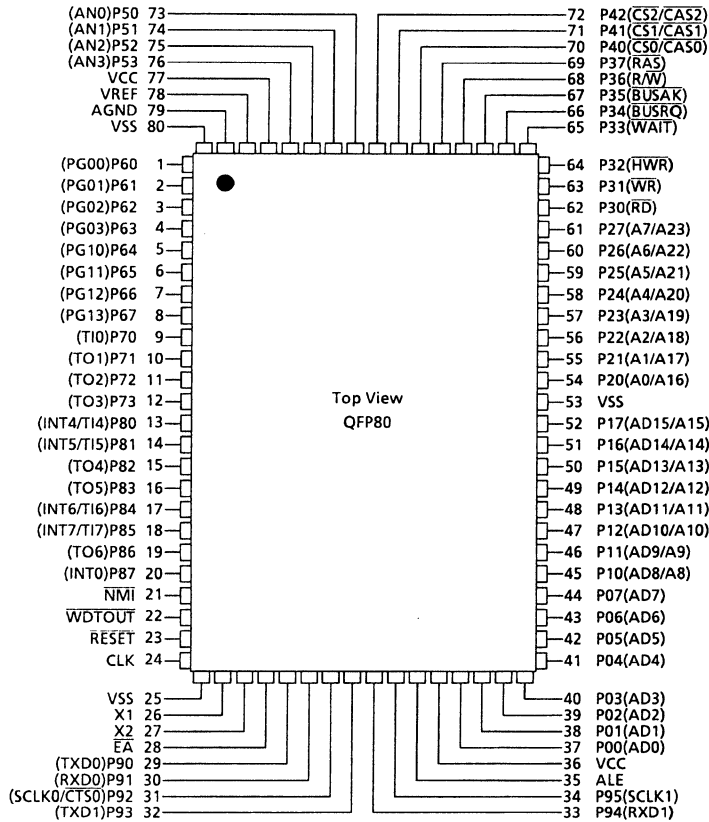


## 2. Pin Assignment and Functions

The assignment of input/output pins for TMP96CM40F, their name and outline functions are described below.

## 2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP96CM40F.



Note : Because the TMP96CM40 has an external ROM, P00 to P17 pins are fixed to AD0 to AD15; P30 to RD; and P31 to WR.

Figure 2.1. Pin Assignment (80-pin QFP)

## 2.2 Pin Names and Functions

described below.

The names of input/output pins and their functions are

**Table 2.2. Pin Names and Functions**

Pin Name	Number of Pins	I/O	Functions
P00 ~ P07 AD0 ~ AD7	8	I/O Tri-state	Port 0: I/O port that allows I/O to be selected on a bit basis Address / data (lower): 0 - 7 for address/data bus
P10 ~ P17 AD8 ~ AD15 A8 ~ A15	8	I/O Tri-state Output	Port 1: I/O port that allows I/O to be selected on a bit basis 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 RD	1	Output Output	Port 30: Output port Read: Strobe signal for reading external memory
P31 WR	1	Output Output	Port 31: Output port Write: Strobe signal for writing data on pins AD0 -7
P32 HWR	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 - 15
P33 WAIT	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P34 BUSRQ	1	I/O Input	Port 34: I/O port (with pull-up resistor) Bus request: Signal used to request high impedance for AD0 - 15, A0 - 23, RD, WR, HWR, R/W, RAS, CS0, CS1, and CS2 pins. (For external DMAC)
P35 BUSAK	1	I/O Output	Port 35: I/O (with pull-up resistor) Bus acknowledge: Signal indicating that AD0 - 15, A0 - 23, RD, WR, HWR, R/W, RAS, CS0, CS1, and CS2 pins are at high impedance after receiving BUSRQ. (For external DMAC)
P36 R/W	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0, write cycle.
P37 RAS	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs RAS strobe for DRAM.
P40 CS0 CAS0	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs CAS strobe for DRAM when address is within specified address area.

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

Pin Name	Number of Pins	I/O	Functions
P41 CS1 CAS1	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs CAS strobe for DRAM if address is within specified address area.
P42 CS2 CAS2	1	I/O Output Output	Port 42: I/O port (with pull-up resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs CAS strobe for DRAM if address is within specified address area.
P50 ~ P53 AN0 ~ AN3	4	Input Input	Port 5: Input port Analog input: Input to A/D converter
VREF	1	Input	Pin for reference voltage input to A/D converter
AGND	1	Input	Ground pin for A/D converter
P60 ~ P63 PG00 ~ PG03	4	I/O Output	Ports 60 - 63: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 00 - 03
P64 ~ P67 PG10 ~ PG13	4	I/O Output	Ports 64 - 67: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 10 - 13
P70 T10	1	I/O Input	Port 70: I/O port (with pull-up resistor) Timer input 0: Timer 0 input
P71 T01	1	I/O Output	Port 71: I/O port (with pull-up resistor) Timer output 1: Timer 0 or 1 output
P72 T02	1	I/O Output	Port 72: I/O port (with pull-up resistor) PWM output 2: 8-bit PWM timer 2 output
P73 T03	1	I/O Output	Port 73: I/O port (with pull-up resistor) PWM output 3: 8-bit PWM timer 3 output
P80 T14 INT4	1	I/O Input Input	Port 80: I/O port (with pull-up resistor) Timer input 4: Timer 4 count/capture trigger signal input Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge
P81 T15 INT5	1	I/O Input Input	Port 81: I/O port (with pull-up resistor) Timer input 5: Timer 4 count/capture trigger signal input Interrupt request pin 5: Interrupt request pin with rising edge
P82 T04	1	I/O Output	Port 82: I/O port (with pull-up resistor) Timer output 4: Timer 4 output pin
P83 T05	1	I/O Output	Port 83: I/O port (with pull-up resistor) Timer output 5: Timer 4 output pin

Note 1: Case of the settable CS2 and CAS2; when TMP96CM40F is bus release, this pin is not added the internal pull-down resistor but is added the internal pull-up resistor.

Pin Name	Number of Pins	I/O	Functions
P84 TI6 INT6	1	I/O Input Input	Port 84: I/O port (with pull-up resistor) Timer input 6: Timer 5 count/capture trigger signal input Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
P85 TI7 INT7	1	I/O Input Input	Port 85: I/O port (with pull-up resistor) Timer input 7: Timer 5 count/capture trigger signal input Interrupt request pin 7: Interrupt request pin with rising edge
P86 TO6	1	I/O Output	Port 86: I/O port (with pull-up resistor) Timer output 6: Timer 5 output pin
P87 INT0	1	I/O Input	Port 87: I/O port (with pull-up resistor) Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P90 TXD0	1	I/O Output	Port 90: I/O port (with pull-up resistor) Serial send data 0
P91 RXD0	1	I/O Input	Port 91: I/O port (with pull-up resistor) Serial receive data 0
P92 CTS0 SCLK0	1	I/O Input I/O	Port 92: I/O port (with pull-up resistor) Serial data send enable 0 (Clear to Send) Serial clock I/O 0
P93 TXD1	1	I/O Output	Port 93: I/O port (with pull-up resistor) Serial send data 1
P94 RXD1	1	I/O Input	Port 94: I/O port (with pull-up resistor) Serial receive data 1
P95 SCLK1	1	I/O I/O	Port 95: I/O port (with pull-up resistor) Serial clock I/O1
WD $\overline$ TOUT	1	Output	Watchdog timer output pin
NMI	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output: Outputs $\lceil X1 \div 4 \rceil$ clock. Pulled-up during reset.
EA	1	Input	External access: 0 should be inputted with TMP96C041AF. 1, with TMP96CM40F/TMP96PM40F.
ALE	1	Output	Address latch enable
RESET	1	Input	Reset: Initializes LSI. (With pull-up resistor)
X1/X2	2	I/O	Oscillator connecting pin
VCC	2		Power supply pin (+5V)
VSS	3		GND pin (0V)

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

**3. Operation**

This section describes in blocks the functions and basic operations of the TMP96CM40F device.

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

**3.1 CPU**

The TMP96CM40F device has a built-in high-performance 16-bit CPU. (For CPU operation, see TLCS-900 CPU in the book Core Architecture User Manual.)

**3.2 Memory Map**

The TMP96CM40F has two register modes. One is minimum mode; in this mode, the area of program memory is 64K bytes maximum. The other is maximum mode; in this mode, the area of the program memory is 16M bytes maximum.

Both minimum and maximum modes are the data memory area 16M bytes maximum.

That is, the program memory can locate 0H ~ FFFFFFFH in maximum mode.

(1) Internal ROM

The TMP96CM40 has the ROM of 32K bytes. This RAM is located to 80H ~ FFFFH. After the RESET operation, instruction execution starts from address 8000H. A part of the internal ROM area (8000H ~ 81FFFH) is interrupt entry area.

(2) Internal RAM

The TMP96CM40 has the RAM of 1K bytes. This RAM is located to 80H ~ 47FH. The CPU can access the part of RAM (80H ~ FFH, 128 bytes) with using the short instruction code of direct addressing mode.

(3) Internal I/O Devices

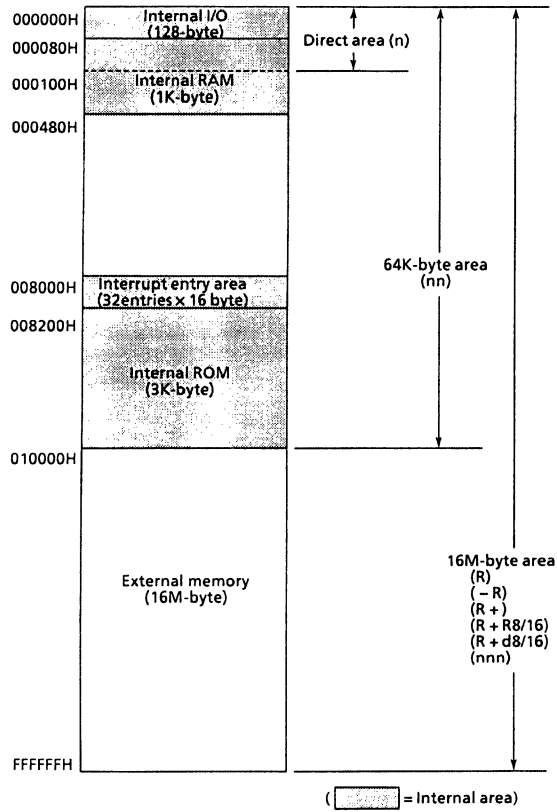
The TMP96CM40 uses the address space of 128 bytes for the internal I/O devices area. This area is located to 0H ~ 7FH.

The CPU can access the internal I/O devices area with using short instruction code of direct addressing mode.

Check the access area of each addressing mode and the memory map in Fig. 3.1.

**3.2 Memory Map**

Figure 3.2 is a memory map of the TMP96CM40F.



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**

## 4. Electrical Characteristics

### 4.1 Absolute Maximum

Symbol	Parameter	Rating	Unit
$V_{CC}$	Power Supply Voltage	-0.5 ~ 6.5	V
$V_{IN}$	Input Voltage	-0.5 ~ $V_{CC} + 0.5$	V
$\Sigma I_{OL}$	Output Current (total)	100	mA
$\Sigma I_{OH}$	Output Current (total)	-100	mA
PD	Power Dissipation ( $T_a = 85^\circ\text{C}$ )	500	mW
T SOLDER	Soldering Temperature (10s)	260	$^\circ\text{C}$
T STG	Storage Temperature	-65 ~ 150	$^\circ\text{C}$
T OPR	Operating Temperature	-40 ~ 85	$^\circ\text{C}$

## 4.2 DC Characteristics

$V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ C$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ C$  (4 ~ 20MHz)  
 (Typical values are for  $T_A = 25^\circ C$  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	$\overline{EA}$	-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	$I_{OH} = -400\mu A$
V OH1		$0.75V_{CC}$		V	$I_{OH} = -100\mu A$
V OH2		$0.9V_{CC}$		V	$I_{OH} = -20\mu A$
I DAR	Darlington Drive Current (8 Output Pins max.)	-1.0	-3.5	mA	$V_{EXT} = 1.5V$ $R_{EXT} = 1.1K\Omega$
I LI	Input Leakage Current	0.02 (Typ)	$\pm 5$	$\mu A$	$0.0 \leq V_{in} \leq V_{CC}$
I LO	Output Leakage Current	0.05 (Typ)	$\pm 10$	$\mu A$	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
$I_{CC}$	Operating Current (RUN)	30 (Typ)	50	mA	$f_c = 20MHz$
	IDLE	2.0 (Typ)	10	mA	
	STOP ( $T_A = -20 \sim 70^\circ C$ )	0.2 (Typ)	50	$\mu A$	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
	STOP ( $T_A = 0 \sim 50^\circ C$ )		10	$\mu A$	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
V STOP	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	$V_{IL2} = 0.2V_{CC}$ $V_{IH2} = 0.8V_{CC}$
R RST	RESET Pull Up Register	50	150	$K\Omega$	
C IO	Pin Capacitance		10	pF	$t_{osc} = 1MHz$
V TH	Schmitt Width $\overline{RESET}$ , $\overline{NMI}$ , INTO (P87)	0.4	1.0 (Typ)	V	
R K	Pull Down/Up Register	50	150	$K\Omega$	

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



4.3 AC Electrical Characteristics

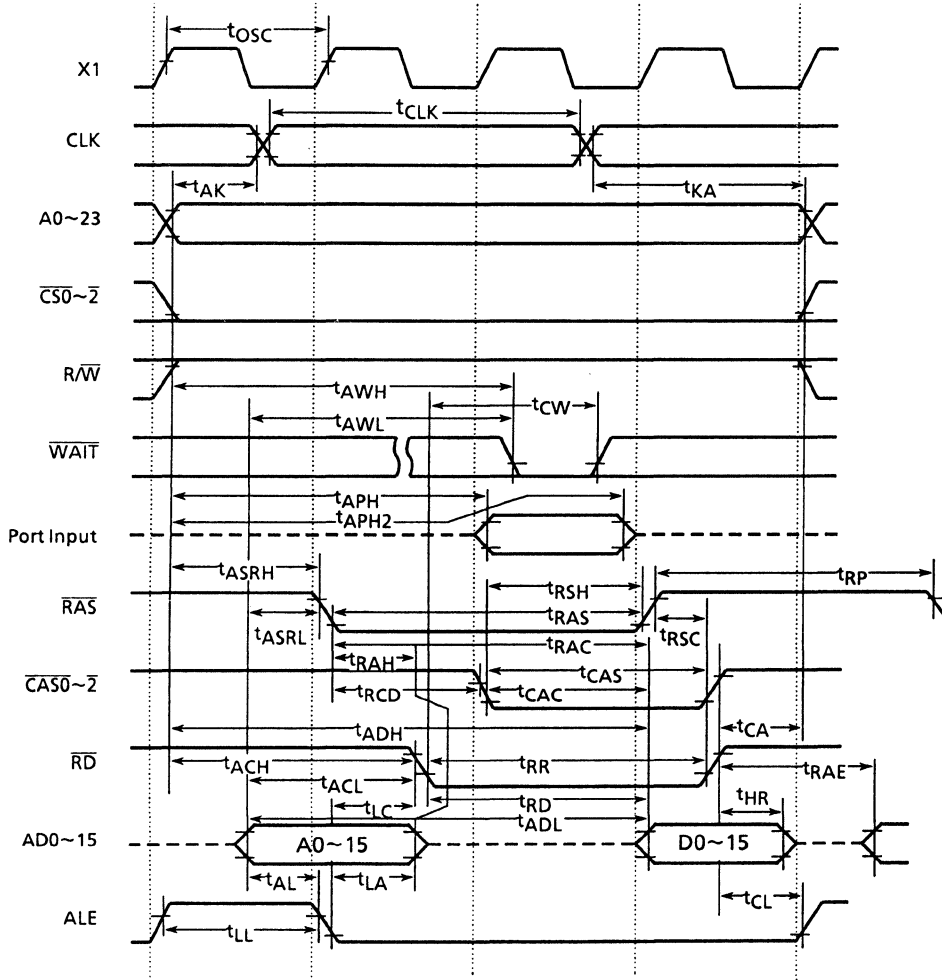
V<sub>CC</sub> = 5V±10% TA = -40 ~ 85°C (4 ~ 16MHz) TA = -20 ~ 70°C (4MHz ~ 20MHz)

No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t <sub>OSC</sub>	Osc. Period (= x)	50	250	62.5		50		ns
2	t <sub>CLK</sub>	CLK width	2x - 40		85		60		ns
3	t <sub>AK</sub>	A0 - 23 Valid→CLK Hold	0.5x - 20		11		5		ns
4	t <sub>KA</sub>	CLK Valid→A0 - 23 Hold	1.5x - 70		24		5		ns
5	t <sub>AL</sub>	A0-15 Valid→ALE fall	0.5x - 15		16		10		ns
6	t <sub>LA</sub>	ALE fall→A0 - 15 Hold	0.5x - 15		16		10		ns
7	t <sub>LL</sub>	ALE High width	x - 40		23		10		ns
8	t <sub>LC</sub>	ALE fall→RD/WR fall	0.5x - 30		1		-5		ns
9	t <sub>CL</sub>	RD/WR rise→ALE rise	0.5x - 20		11		5		ns
10	t <sub>ACL</sub>	A0 - 15 Valid→RD/WR fall	x - 25		38		25		ns
11	t <sub>ACH</sub>	A0 - 23 Valid→RD/WR fall	1.5x - 50		44		25		ns
12	t <sub>CA</sub>	RD/WR rise→A0 - 23 Hold	0.5x - 20		11		5		ns
13	t <sub>ADL</sub>	A0 - 15 Valid→D0 - 15 input		3.0x - 45		143		105	ns
14	t <sub>ADH</sub>	A0 - 23 Valid→D0 - 15 input		3.5x - 65		154		110	ns
15	t <sub>RD</sub>	RD fall→D0 - 15 input		2.0x - 50		75		50	ns
16	t <sub>RR</sub>	RD Low width	2.0x - 40		85		60		ns
17	t <sub>HR</sub>	RD rise→D0 - 15 Hold	0		0		0		ns
18	t <sub>RAE</sub>	RD rise→A0 - 15 output	x - 15		48		35		ns
19	t <sub>WW</sub>	WR Low width	2.0x - 40		85		60		ns
20	t <sub>DW</sub>	D0 - 15 Valid→WR rise	2.0x - 50		75		50		ns
21	t <sub>WD</sub>	WR rise→D0 - 15 Hold	0.5x - 10		21		15		ns
22	t <sub>AEH</sub>	A0 - 23 Valid→WAIT input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	t <sub>AWL</sub>	A0 - 15 Valid→WAIT input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	t <sub>CW</sub>	RD/WR fall→WAIT Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	t <sub>APH</sub>	A0 - 23 Valid→PORT input		2.5x - 120		80		36	ns
26	t <sub>APH2</sub>	A0 - 23 Valid→PORT Hold	2.5x + 50		206		175		ns
27	t <sub>CP</sub>	WR rise→PORT Valid		200		200		200	ns
28	t <sub>ASRH</sub>	A0 - 23 Valid→RAS fall	1.0x - 40		23		10		ns
29	t <sub>ASRL</sub>	A0 - 15 Valid→RAS fall	0.5x - 15		16		10		ns
30	t <sub>RAC</sub>	RAS fall→D0 - 15 input		2.5x - 70		130		86	ns
31	t <sub>RAH</sub>	RAS fall→A0 - 15 Hold	0.5x - 15		16		10		ns
32	t <sub>RAS</sub>	RAS Low width	2.0x - 40		85		60		ns
33	t <sub>RP</sub>	RAS High width	2.0x - 40		85		60		ns
34	t <sub>RSH</sub>	CAS fall→RAS rise	1.0x - 35		28		15		ns
35	t <sub>RSC</sub>	RAS rise→CAS rise	0.5x - 25		6		0		ns
36	t <sub>RCD</sub>	RAS fall→CAS fall	1.0x - 40		23		10		ns
37	t <sub>CAC</sub>	CAS fall→D0 - 15 input		1.5x - 65		29		10	ns
38	t <sub>CAS</sub>	CAS Low width	1.5x - 30		64		40		ns

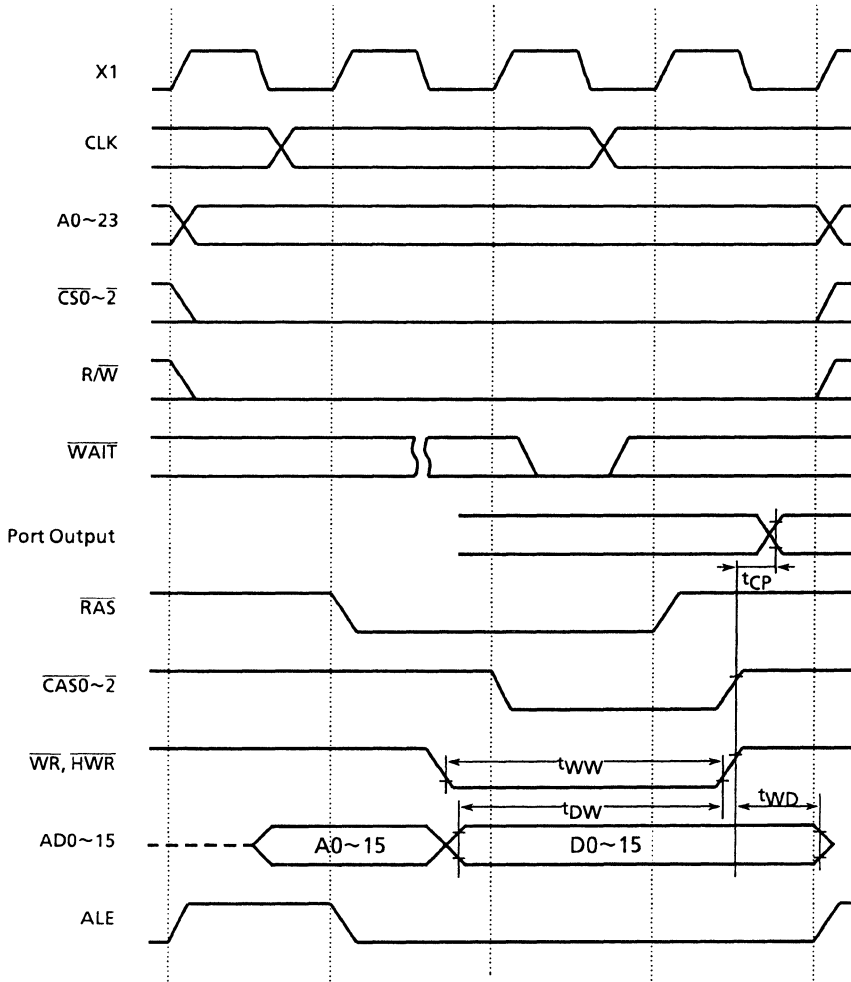
AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF  
(However CL = 100pF for AD0 ~ AD15, AD0 ~ AD23, ALE, RD, WR, FWR, R/W, CLK, RAS, CAS0 ~ CAS2)
- Input Level: High 2.4V /Low 0.45V (AD0 ~ AD15)  
High 0.8Vcc /Low 0.2Vcc (Except for AD0 ~ AD15)

(1) Read Cycle



(2) Write Cycle



## 4.4 A/D Conversion Characteristics

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

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 $\pm 0.5$ LSB not included)	$4 \leq f_c \leq 16$ MHz	Low change mode		$\pm 1.5$	$\pm 4.0$	LSB
		High change mode		$\pm 3.0$	$\pm 6.0$	
	$16 \leq f_c \leq 20$ MHz	Low change mode		$\pm 1.5$	$\pm 4.0$	
		High change mode		$\pm 4.0$	$\pm 6.0$	

## 4.5 Serial Channel Timing - I/O Interface Mode

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

## (1) SCLK Input Mode

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

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

## (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	$\mu s$
$t_{OSS}$	Output data $\rightarrow$ rising edge of SCLK	$t_{SCY} - 2x - 150$		725		550		ns
$t_{OHS}$	SCLK rising edge $\rightarrow$ output data hold	2x - 80		45		20		ns
$t_{HSR}$	SCLK rising edge $\rightarrow$ input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge $\rightarrow$ effective data input		$t_{SCY} - 2x - 150$		725		550	ns

## 4.6 Timer/Counter Input Clock (T10, T14, T15, T16, T17)

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

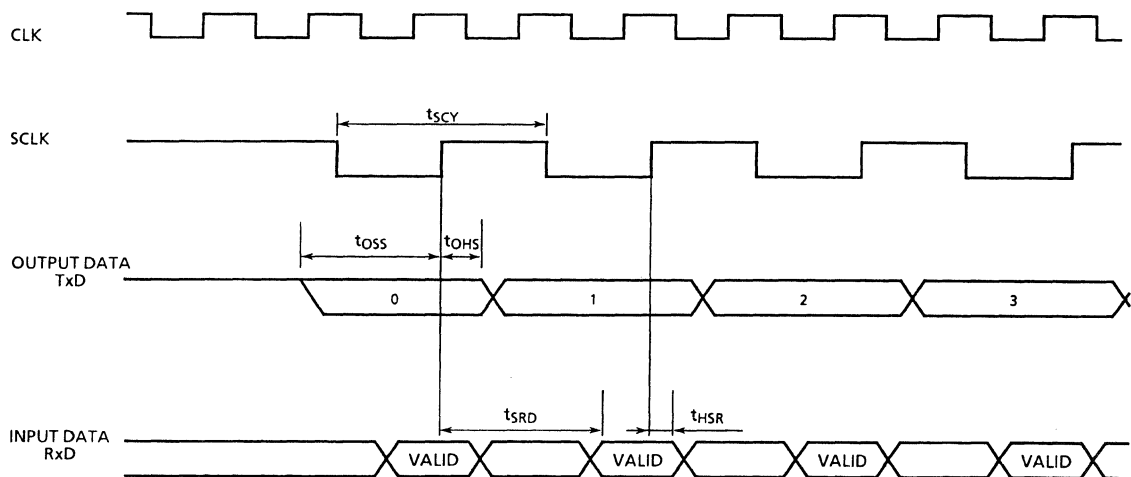
Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{VCK}$	Clock cycle	8x + 100		600		500		ns
$t_{VCKL}$	Low level clock pulse width	4x + 40		290		240		ns
$t_{VCKH}$	High level clock pulse width	4x + 40		290		240		ns

4.7 Interrupt Operation

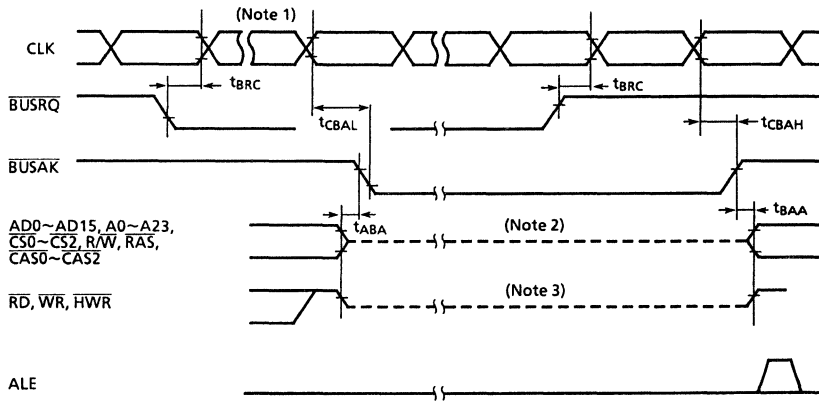
$V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ C$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ C$  (4 ~ 20MHz)

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{INTAL}$	$\overline{NMI}$ , INTO Low level pulse width	4x		250		200		ns
$t_{INTAH}$	$\overline{NMI}$ , INTO High level pulse width	4x		250		200		ns
$t_{INTBL}$	INT4 ~ INT7 Low level pulse width	$8x + 100$		600		500		ns
$t_{INTBH}$	INT4 ~ INT7 High level pulse width	$8x + 100$		600		500		ns

4.8 Timing Chart for I/O Interface Mode



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



Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{BRC}$	$\overline{\text{BUSRQ}}$ setup time for CLK	120		120		120		ns
$t_{CBAL}$	CLK→ $\overline{\text{BUSAK}}$ falling edge		$1.5x + 120$		214		195	ns
$t_{CBAH}$	CLK→ $\overline{\text{BUSAK}}$ rising edge		$0.5x + 40$		71		65	ns
$t_{ABA}$	Output buffer is off to $\overline{\text{BUSAK}}$	0	80	0	80	0	80	ns
$t_{BAA}$	$\overline{\text{BUSAK}}$ output buffer is on.	0	80	0	80	0	80	ns

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

Note 2: This line only shows the output buffer is off-state.

They don't indicate the signal level is fixed.

After the bus is released, the signal level is kept dynamically before the bus is released by the external capacitance.

Therefore, to fix the signal level by an external resistance under the bus is releasing, the design must be carefully because of the level fix will be delayed.

The internal programmable pull-up/pull-up resistance is switched active/non-active by the internal signal.

4.10 Typical Characteristics

$V_{CC} = 5V \pm 10\%$ ,  $T_a = 25^\circ C$  unless otherwise noted.

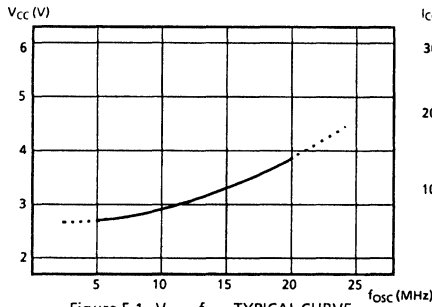


Figure 5.1  $V_{CC} - f_{OSC}$  TYPICAL CURVE

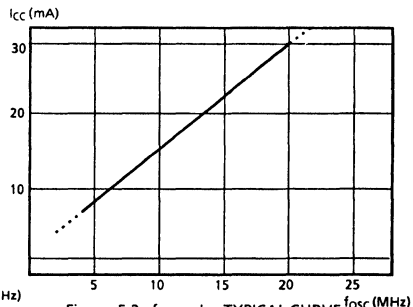


Figure 5.2  $f_{OSC} - I_{CC}$  TYPICAL CURVE

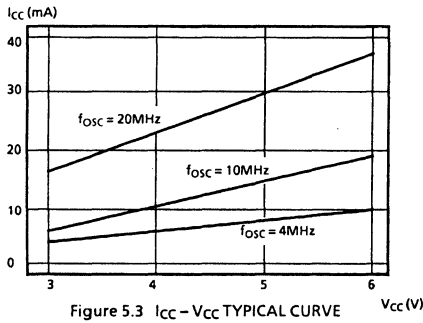


Figure 5.3  $I_{CC} - V_{CC}$  TYPICAL CURVE

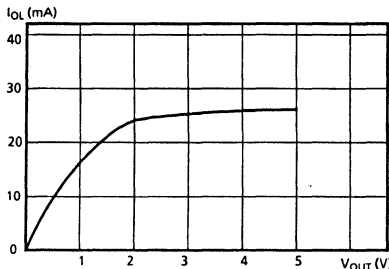


Figure 5.4  $V_{OUT} - I_{OL}$  TYPICAL CURVE

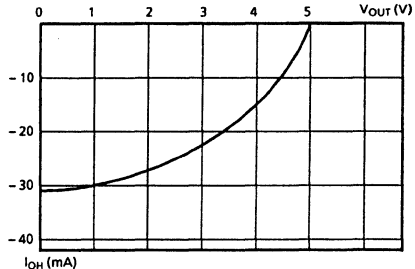


Figure 5.5  $V_{OUT} - I_{OH}$  TYPICAL CURVE

### 5. Differences Between TMP96C141AF, TMP96CM40F, and TMP96PM40F

The devices TMP96C141AF, TMP96CM40F and TMP96PM40F have much the same function but they are different from following points.

Parameter	TMP96C141AF	TMP96CM40F	TMP96PM40F
Internal ROM	Does not exist	Mask ROM 32K byte	PROM 32K byte
P00 to P07, AD0 to AD7	Only AD0 to AD7	After reset P00 ~ P07	
P10 to P17, AD8 to AD15, A8 to A15	Only AD8 to AD15	After reset P10 ~ P17	
P30, $\overline{RD}$	Only $\overline{RD}$	After reset P30	
P31, $\overline{WR}$	Only $\overline{WR}$	After reset P31	



### CMOS 16-bit Microcontrollers

#### TMP96PM40F

#### 1. Outline and Device Characteristics

The TMP96PM40F are high-speed advanced 16-bit microcontrollers developed for controlling medium to large-scale equipment. The TMP96C141AF does not have a ROM, the TMP96CM40F has a built-in ROM of 32K-byte, and the TMP96PM40F has a built-in OTP of 32K-byte.

It is possible to do write/verify of program data with using

an adapter socket and general purpose EPROM writer (TC571000 mode).

TMP96PM40F is pin compatible with TMP96CM40F (mask ROM type).

TMP96CM40F is housed in an 80-pin flat package.

Product Name	ROM	RAM	Package	Adapter Socket Name
TMP96PM40F	OTP type 32K x 8bit	1K x 8bit	80-FP	BM1139A

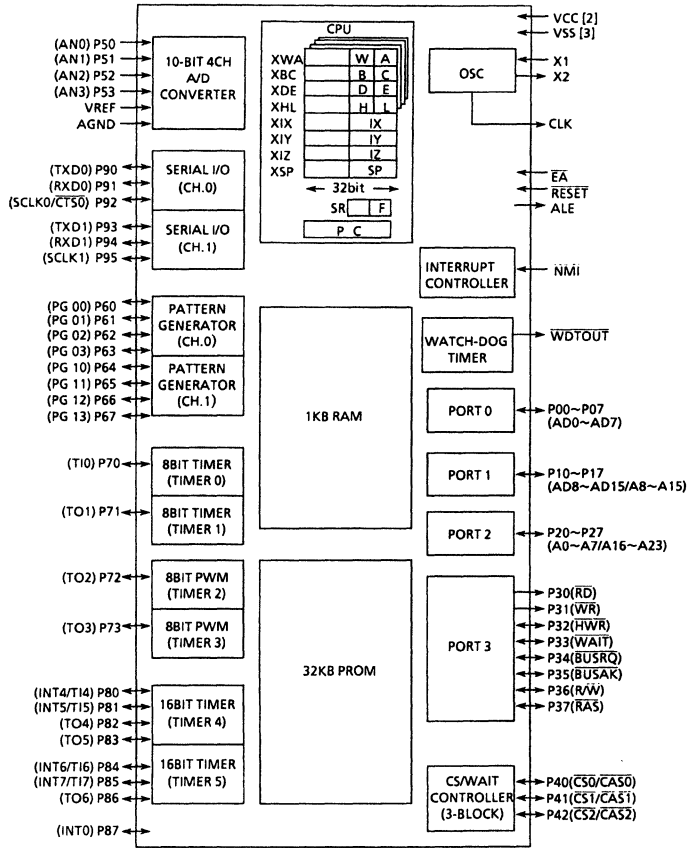


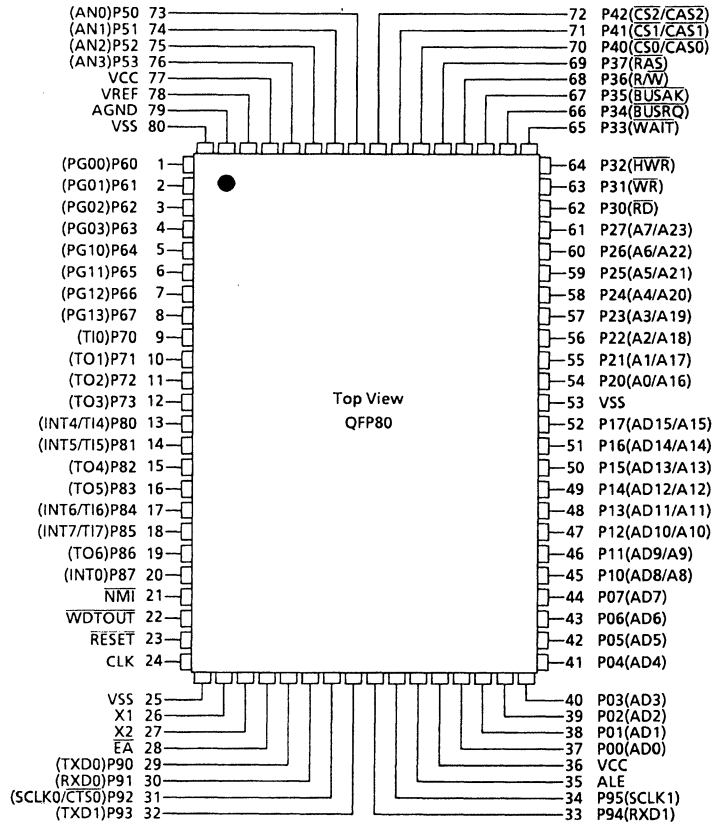
Figure 1. TMP96PM40 Block Diagram

## 2. Pin Assignment and Functions

The assignment of input/output pins for TMP96PM40F, their name and outline functions are described below.

### 2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP96PM40F.



Note: Because the TMP96PM40F has an external ROM, P00 to P17 pins are fixed to AD0 to AD15; P30 to RD; and P31 to WR.

Figure 2.1. Pin Assignment (80-pin QFP)

## 2.2 Pin Names and Functions

The TMP96PM40F has MCU mode and PROM mode.

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

### (1) MCU Mode

**Table 2.2. (1) Pin Names and Functions**

Pin Name	Number of Pins	I/O	Functions
P00 - P07 AD0 - AD7	8	I/O Tri-state	Port 0: I/O port that allows I/O to be selected on a bit basis Address / data (lower): 0 - 7 for address/data bus
P10 - P17 AD8 - AD15 A8 - A15	8	I/O Tri-state Output	Port 1: I/O port that allows I/O to be selected on a bit basis 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 $\overline{RD}$	1	Output Output	Port 30: Output port Read: Strobe signal for reading external memory
P31 $\overline{WR}$	1	Output Output	Port 31: Output port Write: Strobe signal for writing data on pins AD0 - 7
P32 $\overline{HWR}$	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 - 15
P33 $\overline{WAIT}$	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P34 $\overline{BUSRQ}$	1	I/O Input	Port 34: I/O port (with pull-up resistor) Bus request: Signal used to request high impedance for AD0 - 15, A0 - 23, $\overline{RD}$ , $\overline{WR}$ , $\overline{HWR}$ , $\overline{R/W}$ , $\overline{RAS}$ , $\overline{CS0}$ , $\overline{CS1}$ , and $\overline{CS2}$ pins. (For external DMAC)
P35 $\overline{BUSAK}$	1	I/O Output	Port 35: I/O (with pull-up resistor) Bus acknowledge: Signal indicating that AD0 - 15, A0 - 23, $\overline{RD}$ , $\overline{WR}$ , $\overline{HWR}$ , $\overline{R/W}$ , $\overline{RAS}$ , $\overline{CS0}$ , $\overline{CS1}$ , and $\overline{CS2}$ pins are at high impedance after receiving $\overline{BUSRQ}$ . (For external DMAC)
P36 $\overline{R/W}$	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0, write cycle.
P37 $\overline{RAS}$	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs RAS strobe for DRAM.
P40 $\overline{CS0}$ $\overline{CAS0}$	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs CAS strobe for DRAM when address is within specified address area.

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

Pin Name	Number of Pins	I/O	Functions
P41 CS1 CAS1	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs CAS strobe for DRAM if address is within specified address area.
P42 CS2 CAS2	1	I/O Output Output	Port 42: I/O port (with pull-up resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs CAS strobe for DRAM if address is within specified address area.
P50 - P53 AN0 - AN3	4	Input Input	Port 5: Input port Analog input: Input to A/D converter
VREF	1	Input	Pin for reference voltage input to A/D converter
AGND	1	Input	Ground pin for A/D converter
P60 - P63 PG00 - PG03	4	I/O Output	Ports 60 - 63: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 00 - 03
P64 - P67 PG10 - PG13	4	I/O Output	Ports 64 - 67: I/O ports that allow selection of I/O on a bit basis (with pull-up resistor) Pattern generator ports: 10 - 13
P70 T10	1	I/O Input	Port 70: I/O port (with pull-up resistor) Timer input 0: Timer 0 input
P71 T01	1	I/O Output	Port 71: I/O port (with pull-up resistor) Timer output 1: Timer 0 or 1 output
P72 T02	1	I/O Output	Port 72: I/O port (with pull-up resistor) PWM output 2: 8-bit PWM timer 2 output
P73 T03	1	I/O Output	Port 73: I/O port (with pull-up resistor) PWM output 3: 8-bit PWM timer 3 output
P80 TI4 INT4	1	I/O Input Input	Port 80: I/O port (with pull-up resistor) Timer input 4: Timer 4 count/capture trigger signal input Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge
P81 TI5 INT5	1	I/O Input Input	Port 81: I/O port (with pull-up resistor) Timer input 5: Timer 4 count/capture trigger signal input Interrupt request pin 5: Interrupt request pin with rising edge
P82 T04	1	I/O Output	Port 82: I/O port (with pull-up resistor) Timer output 4: Timer 4 output pin
P83 T05	1	I/O Output	Port 83: I/O port (with pull-up resistor) Timer output 5: Timer 4 output pin

Note 1: Case of the settable  $\overline{CS2}$  and  $\overline{CAS2}$ ; when TMP96PM40F is bus release, this pin is not added the internal pull-down resistor but is added the internal pull-up resistor.

Pin Name	Number of Pins	I/O	Functions
P84 TI6 INT6	1	I/O Input Input	Port 84: I/O port (with pull-up resistor) Timer input 6: Timer 5 count/capture trigger signal input Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
P85 TI7 INT7	1	I/O Input Input	Port 85: I/O port (with pull-up resistor) Timer input 7: Timer 5 count/capture trigger signal input Interrupt request pin 7: Interrupt request pin with rising edge
P86 TO6	1	I/O Output	Port 86: I/O port (with pull-up resistor) Timer output 6: Timer 5 output pin
P87 INT0	1	I/O Input	Port 87: I/O port (with pull-up resistor) Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P90 TXD0	1	I/O Output	Port 90: I/O port (with pull-up resistor) Serial send data 0
P91 RXD0	1	I/O Input	Port 91: I/O port (with pull-up resistor) Serial receive data 0
P92 CTS0 SCLK0	1	I/O Input I/O	Port 92: I/O port (with pull-up resistor) Serial data send enable 0 (Clear to Send) Serial clock I/O 0
P93 TXD1	1	I/O Output	Port 93: I/O port (with pull-up resistor) Serial send data 1
P94 RXD1	1	I/O Input	Port 94: I/O port (with pull-up resistor) Serial receive data 1
P95 SCLK1	1	I/O I/O	Port 95: I/O port (with pull-up resistor) Serial clock I/O1
WDTOUT	1	Output	Watchdog timer output pin
NMI	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output: Outputs [X1 + 4] clock. Pulled-up during reset.
EA	1	Input	External access: 0 should be inputted with TMP96C041AF. 1, with TMP96CM40F/TMP96PM40F.
ALE	1	Output	Address latch enable
RESET	1	Input	Reset: Initializes LSI. (With pull-up resistor)
X1/X2	2	I/O	Oscillator connecting pin
VCC	2		Power supply pin (+ 5V)
VSS	3		GND pin (0V)

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

(1) PROM Mode

Table 2.2. (2) Pin Name and Function of PROM Mode

Pin Name	Pin Number	Input/Output	Function	Pin Name (MCU Mode)
A7 ~ A0	8	Input	Memory address of program	P27 ~ P20
A15 ~ A8	8	Input		P17 ~ P10
A16	1	Input		P33
D7 ~ D0	8	I/O	Memory data of program	P07 ~ P00
CE	1	Input	Chip enable	P32
OE	1	Input	Output control	P30
PGM	1	Input	Program control	P31
VPP	1	Power Supply	12.75V/5V (Power supply of program)	EA
VCC	2	Power Supply	6.25V/5V	VCC
VSS	3	Power Supply	0V	VSS
Pin Name	Pin Number	Input/Output	Disposal of Pin	
P34	1	Input	Fix to low level (security pin)	
RESET	1	Input	Fix to low level (PROM mode)	
CLK	1	Input		
ALE	1	Output	Open	
X1	1	Input	Crystal	
X2	1	Output		
P95 ~ P94, VREF	3	Input	Fix to high level	
AGND	1	Input	0V	
P37 ~ P35 P42 ~ P40 P53 ~ P50 P67 ~ P60 P73 ~ P70 P87 ~ P80 NMI, WDTOUT P93 ~ P90	36	I/O	Open	

### 3. Operation

This section describes in blocks the functions and basic operations of TMP96PM40F device. TMP96PM40F is exchanged mask ROM of TMP96CM40F for PROM. The other specifications and functions are the same as TMP96CM40F.

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

#### 3.1 MCU Mode

- (1) Mode-setting and Function

The MCU mode is set by opening the CLK pin (Output status).

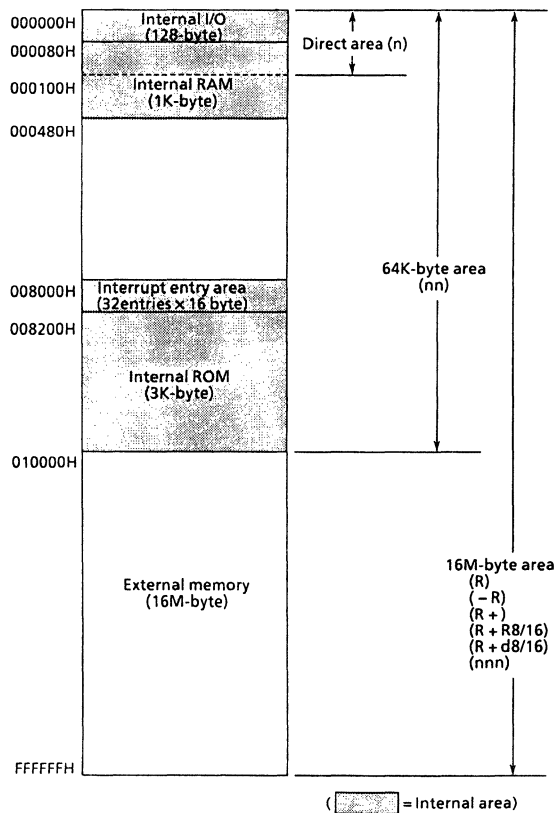
In the MCU mode, the operation is same as that of TMP96CM40F.

- (2) Memory Map

The memory map of TMP96PM40F is same as that of TMP96CM40F. Figure 3.1 shows the memory map of TMP96PM40F, and the accessing area by the respective addressing mode.

#### Memory Map

Figure 3.1 is a memory map of the TMP96PM40F.



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

Figure 3.1. Memory Map



3.2 PROM Mode

(1) Mode Setting and Function

PROM mode is set by setting the  $\overline{\text{RESET}}$  and CLK pins to the "L" level. The programming and verification for the internal PROM is achieved by using a general

EPROM programmer with the adapter socket. The device selection (ROM type) use following conditions (Set ROM type to TC571000D)

Size: 1Mbit (128K x 8bit) VPP: 12.75V TPW: 0.1ms

Figure 3.2 shows the setting of pins in PROM mode.

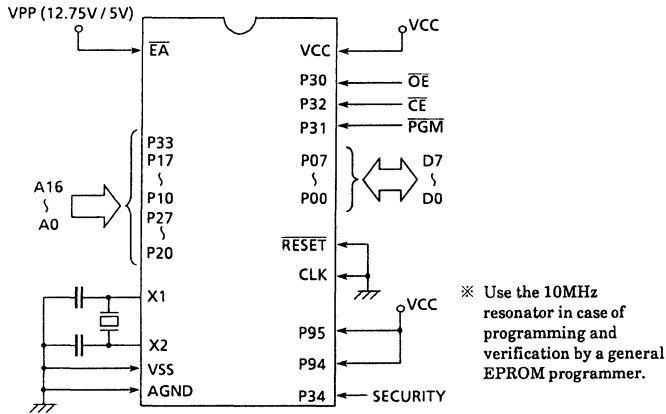


Figure 3.2. PROM Mode Pin Setting

(2) Programming Flow Chart

The programming mode is set by applying 12.5V (programming voltage) to the VPP pin when the following pins are set as follows,

(VCC: 6.25V,  $\overline{\text{RESET}}$ : "L" level, CLK: "L" level.) After the address and data have been fixed, the data on the Data Bus is programmed when the  $\overline{\text{CE}}$  pin is set to "L" level (0.1ms pulse is required). General programming procedure of an EPROM programmer is as follows,

- Write a data to a specified address for 0.1ms
- Verify the data. If the readout data does not match the expected data, another writing is performed until the correct data is written (Max. 25 times).

Then, verify the data and increment the address. The verification for all data is done under the condition of  $V_{PP} = V_{CC} = 5$  all data were written.

Figure 3.3 shows the programming flow chart.

High Speed Program Writing

Flow Chart

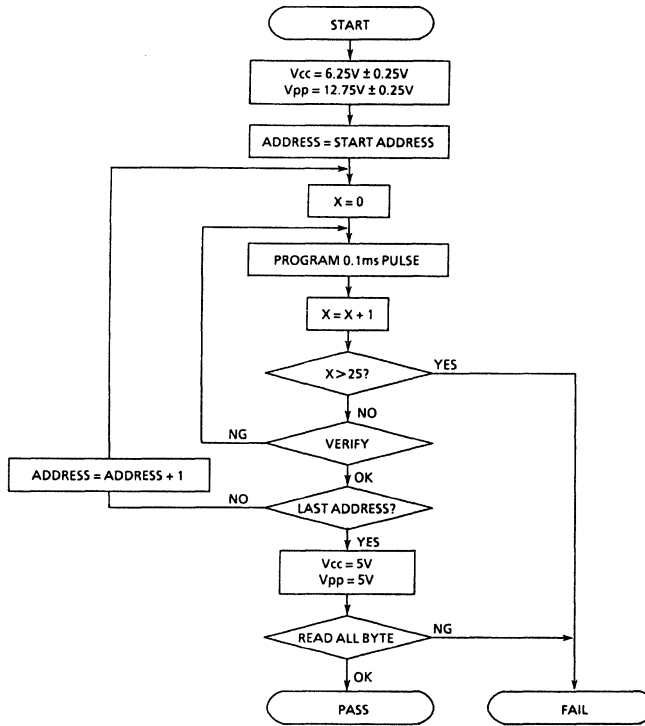


Figure 3.3. Flow Chart

(3) Security Bit

The TMP96PM 40 has a security bit in PROM cell. If the Security Bit is programmed to "0", the content of the PROM is disable to be read in PROM mode.

How to Program the Security Bit

- 1) Set the PROM mode.
- 2) Set the security pin (Port34) to "1".
- 3) Set the programming address to "000000H".
- 4) Set programming data to "FEH".

## 4. Electrical Characteristics

### 4.1 Absolute Maximum

Symbol	Parameter	Rating	Unit
$V_{CC}$	Power Supply Voltage	-0.5 - 6.5	V
$V_{IN}$	Input Voltage	-0.5 - $V_{CC} + 0.5$	V
$\Sigma I_{OL}$	Output Current (total)	100	mA
$\Sigma I_{OH}$	Output Current (total)	-100	mA
PD	Power Dissipation ( $T_a = 70^\circ\text{C}$ )	500	mW
T SOLDER	Soldering Temperature (10s)	260	$^\circ\text{C}$
T STG	Storage Temperature	-65 - 150	$^\circ\text{C}$
T OPR	Operating Temperature	-40 - 85	$^\circ\text{C}$

## 4.2 DC Characteristics

 $V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ C$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ C$  (4 ~ 20MHz)
(Typical values are for  $T_A = 25^\circ C$  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	RESET, NMI, INTO (P87)	-0.3	$0.25V_{CC}$	V	
V IL3	$\overline{EA}$	-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	RESET, 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 = -400 $\mu$ A
V OH1		$0.75V_{CC}$		V	I OH = -100 $\mu$ A
V OH2		$0.9V_{CC}$		V	I OH = -20 $\mu$ A
I DAR	Darlington Drive Current (8 Output Pins max.)	-1.0	-3.5	mA	V EXT = 1.5V R EXT = 1.1K $\Omega$
I LI	Input Leakage Current	0.02 (Typ)	$\pm 5$	$\mu$ A	$0.0 \leq V_{in} \leq V_{CC}$
I LO	Output Leakage Current	0.05 (Typ)	$\pm 10$	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
I <sub>CC</sub>	Operating Current (RUN)		60	mA	$f_c = 20MHz$
	IDLE	30 (Typ)	10	mA	
	STOP ( $T_A = -20 \sim 70^\circ C$ )	2.0 (Typ)	50	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
	STOP ( $T_A = 0 \sim 50^\circ C$ )	0.2 (Typ)	10	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
V STOP	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	V IL2 = $0.2V_{CC}$ , V IH2 = $0.8V_{CC}$
R RST	RESET Pull Up Register	50	150	K $\Omega$	
C IO	Pin Capacitance		10	pF	$f_c = 1MHz$
V TH	Schmitt Width RESET, NMI, INTO (P87)	0.4	1.0 (Typ)	V	
R K	Pull Down/Up Register	50	150	K $\Omega$	

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

4.3 AC Electrical Characteristics

V<sub>CC</sub> = 5V±10% TA = -40 ~ 85°C (4 ~ 16MHz) TA = -20 ~ 70°C (4MHz ~ 20MHz)

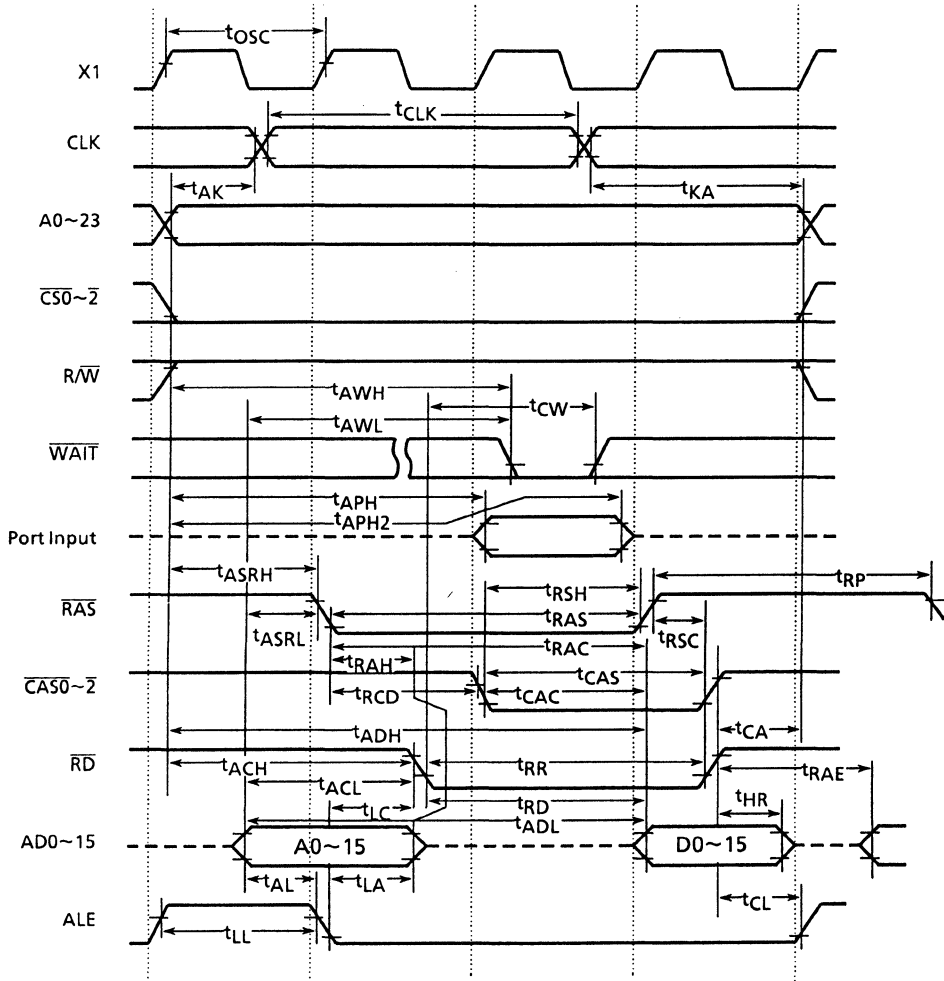
No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t <sub>OSC</sub>	Osc. Period (= x)	50	250	62.5		50		ns
2	t <sub>CLK</sub>	CLK width	2x - 40		85		60		ns
3	t <sub>AK</sub>	A0 - 23 Valid→CLK Hold	0.5x - 20		11		5		ns
4	t <sub>KA</sub>	CLK Valid→A0 - 23 Hold	1.5x - 70		24		5		ns
5	t <sub>AL</sub>	A0-15 Valid→ALE fall	0.5x - 15		16		10		ns
6	t <sub>LA</sub>	ALE fall→A0 - 15 Hold	0.5x - 15		16		10		ns
7	t <sub>LL</sub>	ALE High width	x - 40		23		10		ns
8	t <sub>LC</sub>	ALE fall→RD/WR fall	0.5x - 30		1		-5		ns
9	t <sub>CL</sub>	RD/WR rise→ALE rise	0.5x - 20		11		5		ns
10	t <sub>ACL</sub>	A0 - 15 Valid→RD/WR fall	x - 25		38		25		ns
11	t <sub>ACH</sub>	A0 - 23 Valid→RD/WR fall	1.5x - 50		44		25		ns
12	t <sub>CA</sub>	RD/WR rise→A0 - 23 Hold	0.5x - 20		11		5		ns
13	t <sub>ADL</sub>	A0 - 15 Valid→D0 - 15 input		3.0x - 55		133		95	ns
14	t <sub>ADH</sub>	A0 - 23 Valid→D0 - 15 input		3.5x - 65		154		110	ns
15	t <sub>RD</sub>	RD fall→D0 - 15 input		2.0x - 50		75		50	ns
16	t <sub>RR</sub>	RD Low width	2.0x - 40		85		60		ns
17	t <sub>HR</sub>	RD rise→D0 - 15 Hold	0		0		0		ns
18	t <sub>RAE</sub>	RD rise→A0 - 15 output	x - 15		48		35		ns
19	t <sub>WW</sub>	WR Low width	2.0x - 40		85		60		ns
20	t <sub>DW</sub>	D0 - 15 Valid→WR rise	2.0x - 50		75		50		ns
21	t <sub>WD</sub>	WR rise→D0 - 15 Hold	0.5x - 10		21		15		ns
22	t <sub>AEH</sub>	A0 - 23 Valid→WAIT input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	t <sub>AWL</sub>	A0 - 15 Valid→WAIT input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	t <sub>CW</sub>	RD/WR fall→WAIT Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	t <sub>APH</sub>	A0 - 23 Valid→PORT input		2.5x - 120		36		5	ns
26	t <sub>APH2</sub>	A0 - 23 Valid→PORT Hold	2.5x + 50		206		175		ns
27	t <sub>CP</sub>	WR rise→PORT Valid		200		200		200	ns
28	t <sub>ASRH</sub>	A0 - 23 Valid→RAS fall	1.0x - 40		23		10		ns
29	t <sub>ASRL</sub>	A0 - 15 Valid→RAS fall	0.5x - 15		16		10		ns
30	t <sub>RAC</sub>	RAS fall→D0 - 15 input		2.5x - 70		130		86	ns
31	t <sub>RAH</sub>	RAS fall→A0 - 15 Hold	0.5x - 15		16		10		ns
32	t <sub>RAS</sub>	RAS Low width	2.0x - 40		85		60		ns
33	t <sub>RP</sub>	RAS High width	2.0x - 40		85		60		ns
34	t <sub>RSH</sub>	CAS fall→RAS rise	1.0x - 35		28		15		ns
35	t <sub>RSC</sub>	RAS rise→CAS rise	0.5x - 25		6		0		ns
36	t <sub>RCD</sub>	RAS fall→CAS fall	1.0x - 40		23		10		ns
37	t <sub>CAC</sub>	CAS fall→D0 - 15 input		1.5x - 65		29		10	ns
38	t <sub>CAS</sub>	CAS Low width	1.5x - 30		64		40		ns

\*t<sub>ADL</sub> value is different from TMP96C141/TMP96CM40.

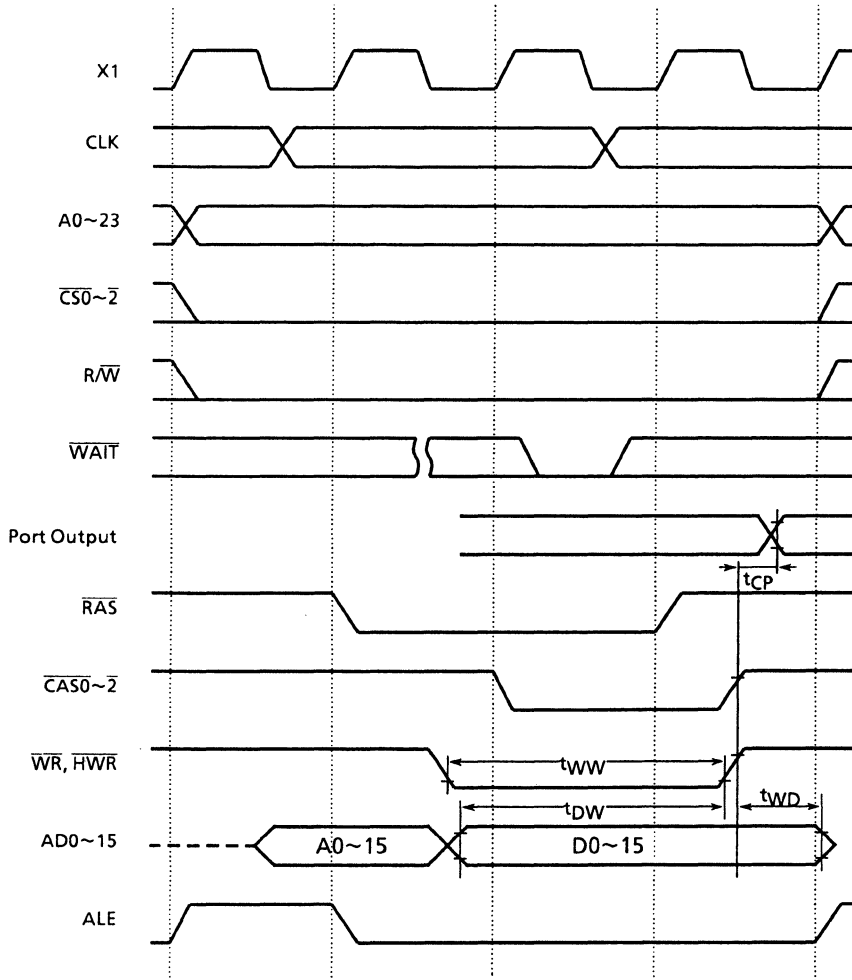
AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF  
(However CL = 100pF for AD0 ~ AD15, AD0 ~ AD23, ALE, RD, WR, HWR, R/W, CLK, RAS, CAS0 ~ CAS2)
- Input Level: High 2.4V /Low 0.45V (AD0 ~ AD15)  
High 0.8V<sub>CC</sub> /Low 0.2V<sub>CC</sub> (Except for AD0 ~ AD15)

(1) Read Cycle



(2) Write Cycle



## 4.4 A/D Conversion Characteristics

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

Symbol	Parameter		Min	Typ	Max	Unit
$V_{REF}$	Analog reference voltage		$V_{CC} - 1.5$	$V_{CC}$	$V_{CC}$	V
$A_{GND}$	Analog reference voltage		$V_{SS}$	$V_{SS}$	$V_{SS}$	
$V_{AIN}$	Analog input voltage range		$V_{SS}$		$V_{CC}$	
$I_{REF}$	Analog current for analog reference voltage			0.5	1.5	mA
Error (Quantize error of $\pm 0.5$ LSB not included)	$4 \leq f_c \leq 16\text{MHz}$	Low change mode		$\pm 1.5$	$\pm 4.0$	LSB
		High change mode		$\pm 3.0$	$\pm 6.0$	
	$16 \leq f_c \leq 20\text{MHz}$	Low change mode		$\pm 1.5$	$\pm 4.0$	
		High change mode		$\pm 4.0$	$\pm 8.0$	

## 4.5 Serial Channel Timing - I/O Interface Mode

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

## (1) SCLK Input Mode

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

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

## (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	$\mu s$
$t_{OSS}$	Output Data $\rightarrow$ rising edge of SCLK	$t_{SCY} - 2x - 150$		725		550		ns
$t_{OHS}$	SCLK rising edge $\rightarrow$ output data hold	$2x - 80$		45		20		ns
$t_{HSR}$	SCLK rising edge $\rightarrow$ input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge $\rightarrow$ effective data input			$t_{SCY} - 2x - 150$		725	550	ns

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

 $V_{CC} = 5V \pm 10\%$ ,  $TA = -40 \sim 85^{\circ}C$  (4 ~ 16MHz)  $TA = -20 \sim 70^{\circ}C$  (4 ~ 20MHz)

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

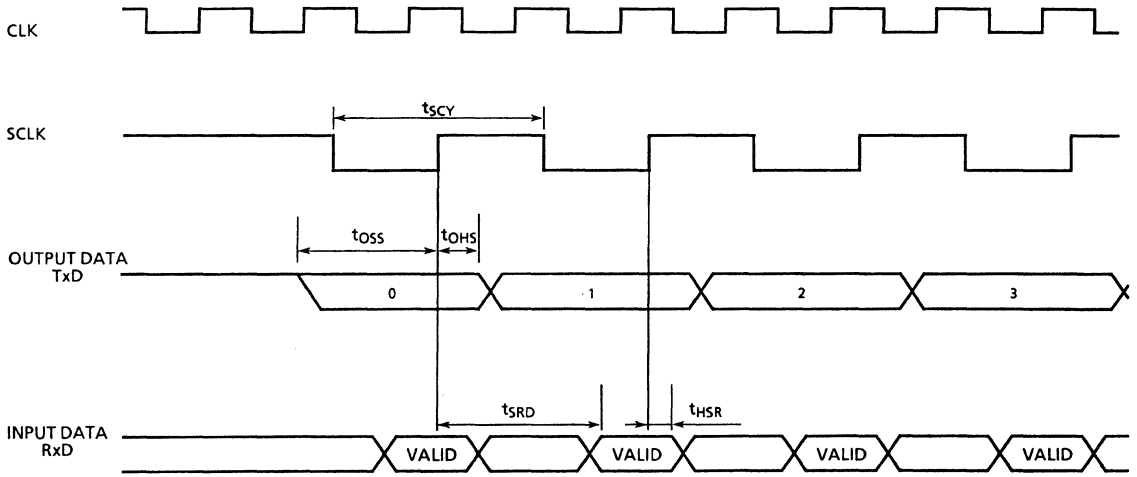


4.7 Interrupt Operation

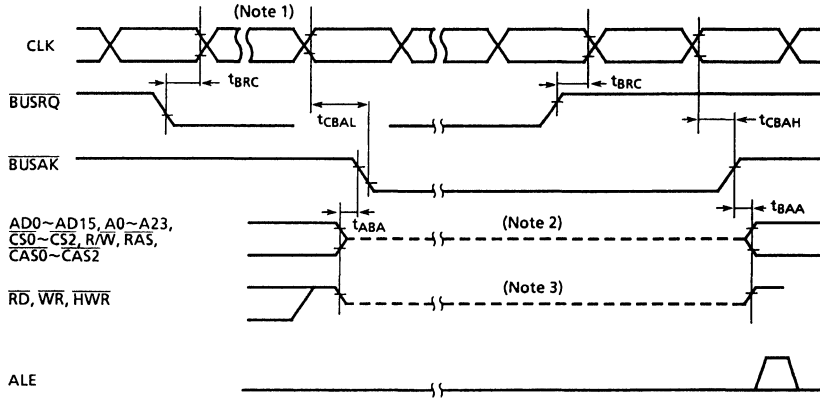
$V_{CC} = 5V \pm 10\%$ ,  $T_A = -40 \sim 85^\circ C$  (4 ~ 16MHz)  $T_A = -20 \sim 70^\circ C$  (4 ~ 20MHz)

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{INTAL}$	$\overline{NMI}$ , INTO Low level pulse width	4x		250		200		ns
$t_{INTAH}$	$\overline{NMI}$ , INTO High level pulse width	4x		250		200		ns
$t_{INTBL}$	INT4 ~ INT7 Low level pulse width	8x + 100		600		500		ns
$t_{INTBH}$	INT4 ~ INT7 High level pulse width	8x + 100		600		500		ns

4.8 Timing Chart for I/O Interface Mode



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



Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{\text{BRC}}$	$\overline{\text{BUSRQ}}$ setup time for CLK	120		120		120		ns
$t_{\text{CBAL}}$	CLK→ $\overline{\text{BUSAK}}$ falling edge		$1.5x + 120$		214		195	ns
$t_{\text{CBAH}}$	CLK→ $\overline{\text{BUSAK}}$ rising edge		$0.5x + 40$		71		65	ns
$t_{\text{ABA}}$	Output buffer is off to $\overline{\text{BUSAK}}$	0	80	0	80	0	80	ns
$t_{\text{BAA}}$	$\overline{\text{BUSAK}}$ output buffer is on.	0	80	0	80	0	80	ns

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

Note 2: This line only shows the output buffer is off-state.

They don't indicate the signal level is fixed.

After the bus is released, the signal level is kept dynamically before the bus is released by the external capacitance.

Therefore, to fix the signal level by an external resistance under the bus is releasing, the design must be carefully because of the level fix will be delayed.

The internal programmable pull-up/pull-down resistance is switched active/non-active by the internal signal.

## 4.10 Read Operation (PROM Mode)

DC Characteristic, AC Characteristic

TA = -40 ~ 85°C, V<sub>CC</sub> = 5V±10%

Symbol	Parameter	Condition	Min	Max	Unit
V <sub>PP</sub>	V <sub>PP</sub> Read Voltage	—	4.5	5.5	V
V <sub>H1</sub>	Input High Voltage (A0 ~ A16, $\overline{CE}$ , $\overline{OE}$ , $\overline{PGM}$ )	—	0.7 x V <sub>CC</sub>	V <sub>CC</sub> + 0.3	V
V <sub>IL1</sub>	Input Low Voltage (A0 ~ A16, $\overline{CE}$ , $\overline{OE}$ , $\overline{PGM}$ )	—	-3.0	0.3 x V <sub>CC</sub>	V
t <sub>ACC</sub>	Address to Output Delay	CL = 50pF	—	2.25TCYC + α	ns

TCYC = 400ns (10MHz Clock)

α = 200ns

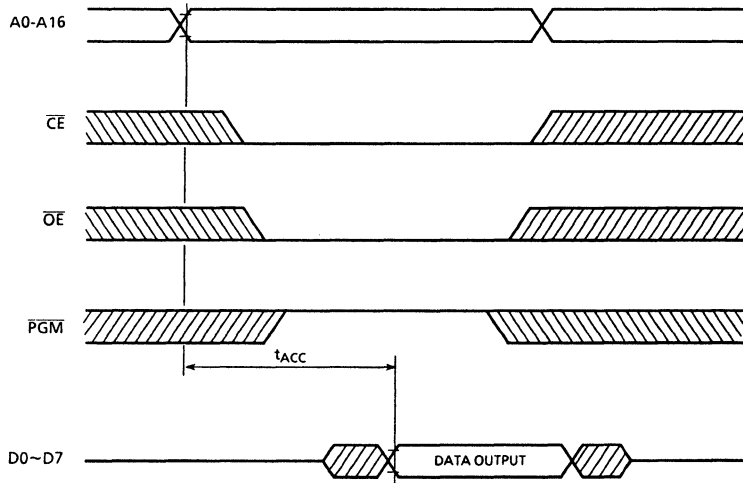
## 4.11 Programming Read Operation (PROM Mode)

DC Characteristic, AC Characteristic

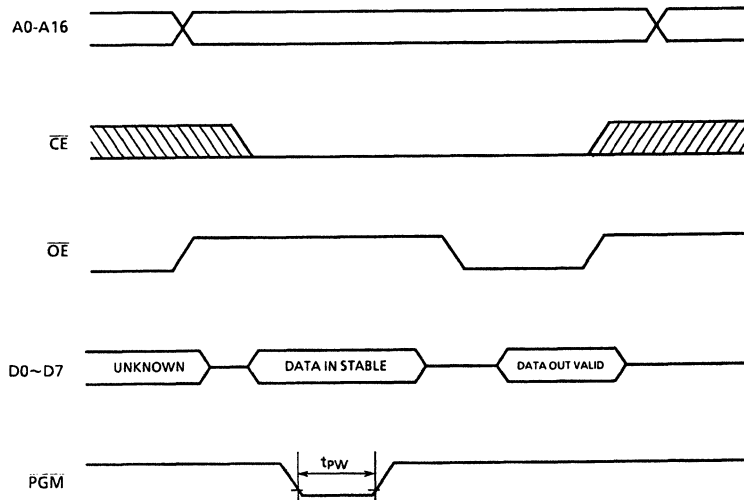
TA = 25±5°C, V<sub>CC</sub> = 6.25V±0.25V

Symbol	Parameter	Condition	Min	Typ	Max	Unit
V <sub>PP</sub>	Programming Supply Voltage	—	12.50	12.75	12.75	V
V <sub>H</sub>	Input High Voltage (D0 ~ D7)	—	0.2V <sub>CC</sub> + 1.1		V <sub>CC</sub> + 0.3	V
V <sub>IL</sub>	Input Low Voltage (D0 ~ D7)	—	-0.3		0.2V <sub>CC</sub> - 0.1	V
V <sub>H1</sub>	Input High Voltage (A0 ~ A16, $\overline{CE}$ , $\overline{OE}$ , $\overline{PGM}$ )	—	0.7V <sub>CC</sub>		V <sub>CC</sub> + 0.3	V
V <sub>IL1</sub>	Input Low Voltage (A0 ~ A16, $\overline{CE}$ , $\overline{OE}$ , $\overline{PGM}$ )	—	-3.0		0.3V <sub>CC</sub>	V
I <sub>CC</sub>	V <sub>CC</sub> Supply Current	f <sub>c</sub> = 10MHz	—		50	mA
I <sub>PP</sub>	V <sub>PP</sub> Supply Current	V <sub>PP</sub> = 13.00V	—		50	mA
t <sub>ACC</sub>	PGM Program Pulse Width	C <sub>L</sub> = 50pF	0.095	0.1	0.105	ms

4.12 Read Operation Timing Chart (PROM Mode)



4.13 Programming Operation Timing Chart (PROM Mode)



Note

1. The power supply of  $V_{PP}$  (12.75V) must be set power -on at the same time or the later time for a power supply of  $V_{CC}$  and must be clear power-on at the same time or early time for a power supply of  $V_{CC}$ .
2. The pulling up/down device on condition of  $V_{PP} = 12.75V$  suffer a damage for the device.
3. The maximum spec of  $V_{PP}$  pin is 14.0V. Be careful a overshoot at the program writing.

4.14 Typical Characteristics

$V_{CC} = 5V$ ,  $T_a = 25^\circ C$  unless otherwise noted.

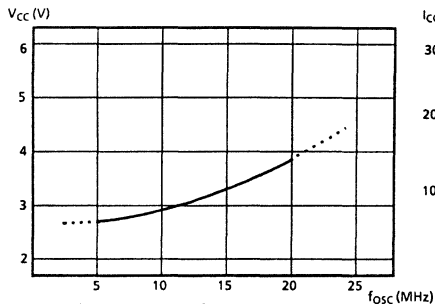


Figure 5.1  $V_{CC} - f_{OSC}$  TYPICAL CURVE

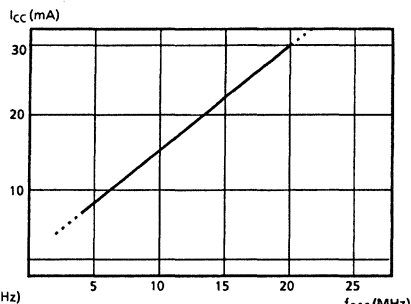


Figure 5.2  $f_{OSC} - I_{CC}$  TYPICAL CURVE

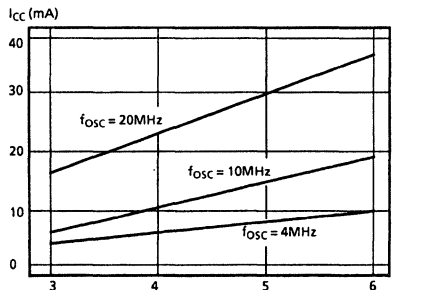


Figure 5.3  $I_{CC} - V_{CC}$  TYPICAL CURVE

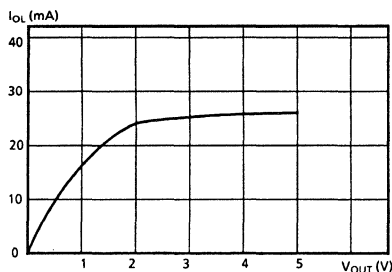


Figure 5.4  $V_{OUT} - I_{OL}$  TYPICAL CURVE

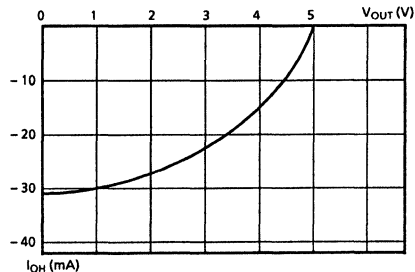


Figure 5.5  $V_{OUT} - I_{OH}$  TYPICAL CURVE

**5. Differences Between TMP96C141AF,  
TMP96CM40F, and TMP96PM40F**

The devices TMP96C141AF, TMP96CM40F, and TMP96PM40F have much the same function, but they are different from following points.

Parameter	TMP96C141AF	TMP96CM40F	TMP96PM40F
Internal ROM	Does not exist	Mask ROM 32K-byte	PROM 32K-byte
P00 to P07, AD0 to AD7	Does not exist	After reset P00 to P07	
P10 to P17, AD8 to AD15, A8 to A15	Only AD8 to AD15	After reset P10 to P17	
P30, $\overline{RD}$	Only $\overline{RD}$	After reset P30	
P31, $\overline{WR}$	Only $\overline{WR}$	After reset P31	

**Notes**



### 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 (1.6 $\mu$ s/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  $\square$  7-level priority can be set.

- 9 external interrupts

(13) I/O ports: 37 pins

(14) Standby function : 3 HALT modes (RUN, IDLE, STOP)

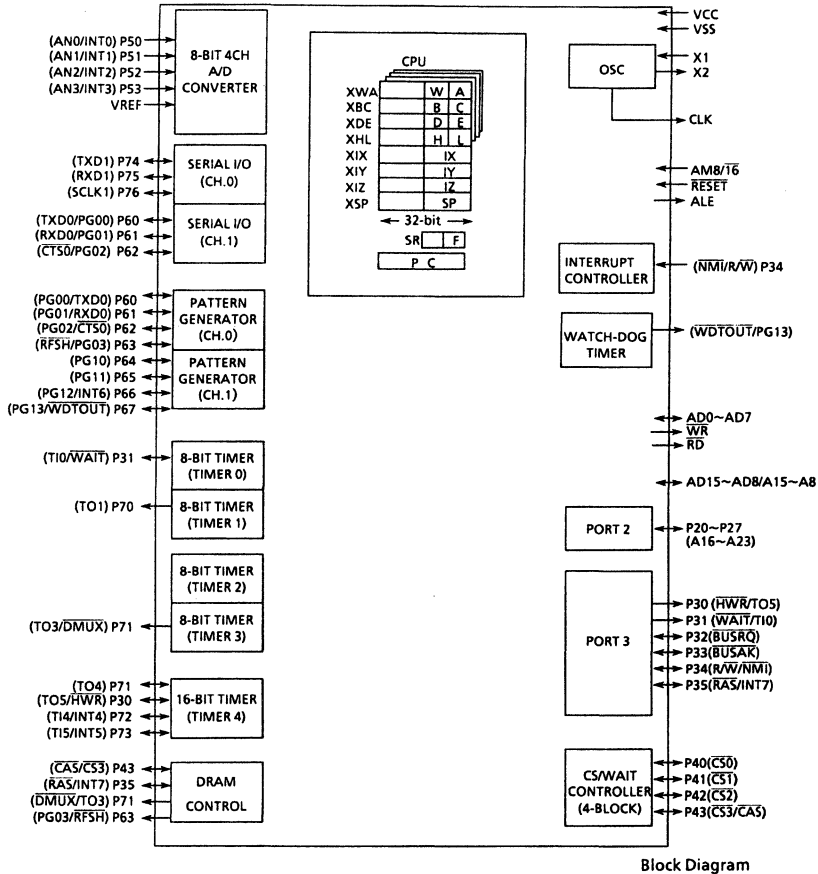


Figure 1. TMP96C031F Block Diagram

2. Pin Assignment and Functions

2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP96C031N.

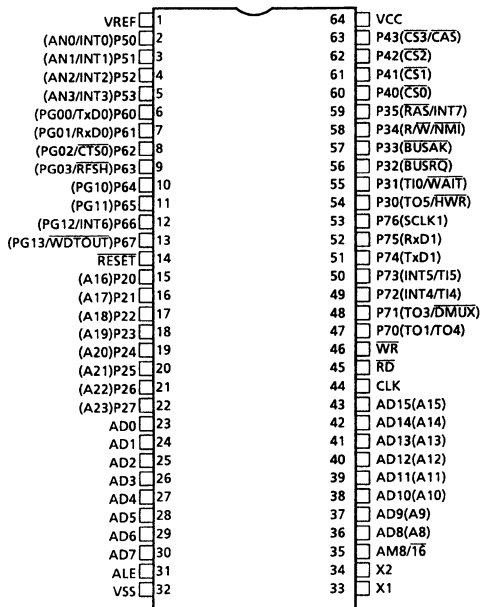


Figure 2.1 (1). Pin Assignment (64-SDIP)

Figure 2.1 (2) shows pin assignment of TMP96C031F.

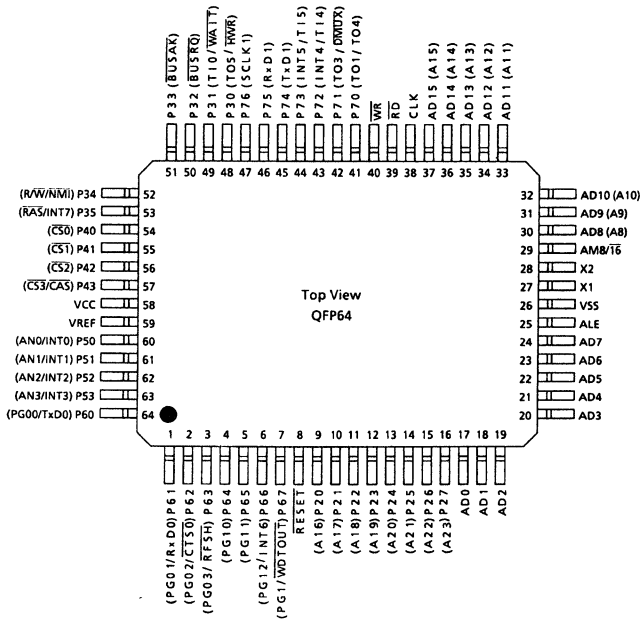


Figure 2.1 (2). Pin Assignment (64QFP)

## 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 T05 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 TIO 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 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, $\overline{RD}$ , $\overline{WR}$ , $\overline{HWR}$ , $\overline{R/W}$ , $\overline{RAS}$ , $\overline{CS0}$ , $\overline{CS1}$ , and $\overline{CS2}$ pins. (For external DMAC)
P33 BUSAK	1	I/O Input	Port 33: I/O port (with pull-up register) Bus acknowledge: Strobe indicating that AD0 - 15, A0 - 23, $\overline{RD}$ , $\overline{WR}$ , $\overline{HWR}$ , $\overline{R/W}$ , $\overline{RAS}$ , $\overline{CS0}$ , $\overline{CS1}$ , and $\overline{CS2}$ pins are at high impedance after receiving BUSRQ.
P34 $\overline{R/W}$ $\overline{NMI}$	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 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 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 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 WDTOUT PG13	1	I/O Output Output	Port 71: I/O port Watchdog timer output pin Pattern generator port 13
P70 T01 T04	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 T03 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 $[X1+4]_{\text{clock}}$ . Pulled-up during reset.
RD	1	Output	Read: Strobe signal for reading external memory.
WR	1	Output	Write: Strobe signal for writing data on pins ADO - 7.
AM8/16	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.
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 RESET input must be kept at 0 for at least 10 system clocks (10 states: 1 $\mu$ s 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 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 (AM8/16)

The TMP96C031F automatically operates in 8-bit bus/16-bit bus mode after reset depending on how the AM8/16 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.

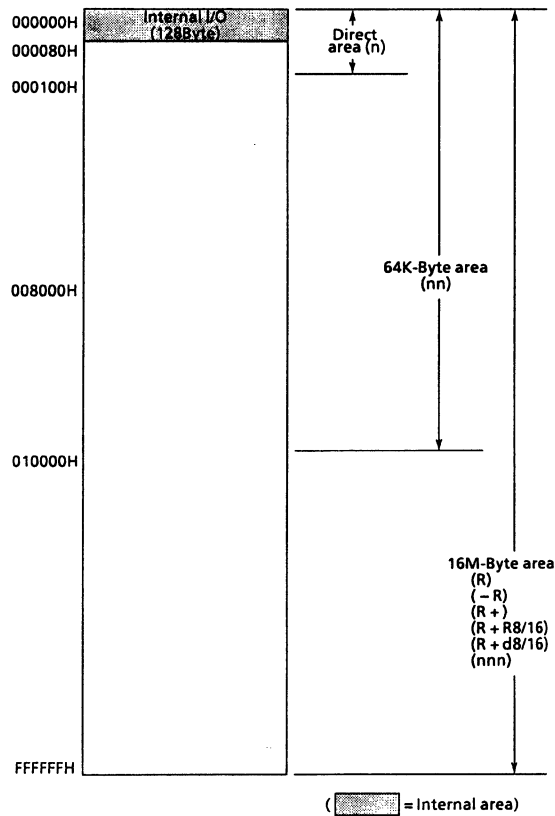
- For fixed external 8-bit data bus

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, INTO, and INTO 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  $\mu$ DMA processing mode. High-speed  $\mu$ DMA is a mode used by the CPU to automatically transfer byte or word data. It enables the CPU to process interrupts such as data saves to built-in I/Os at high speed.

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

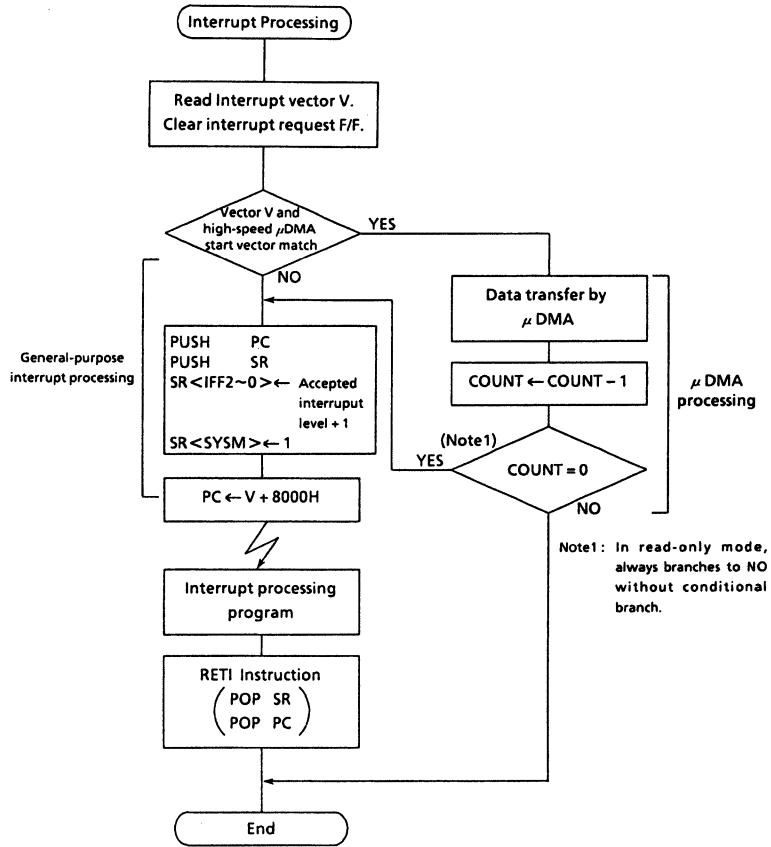


Figure 3.3 (1). Interrupt Processing Flowchart

### 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		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 $\mu$ DMA

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

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

**(1) High-Speed  $\mu$ DMA Operation**

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

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

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  $\mu$ DMA. 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  $\mu$ DMA processing.

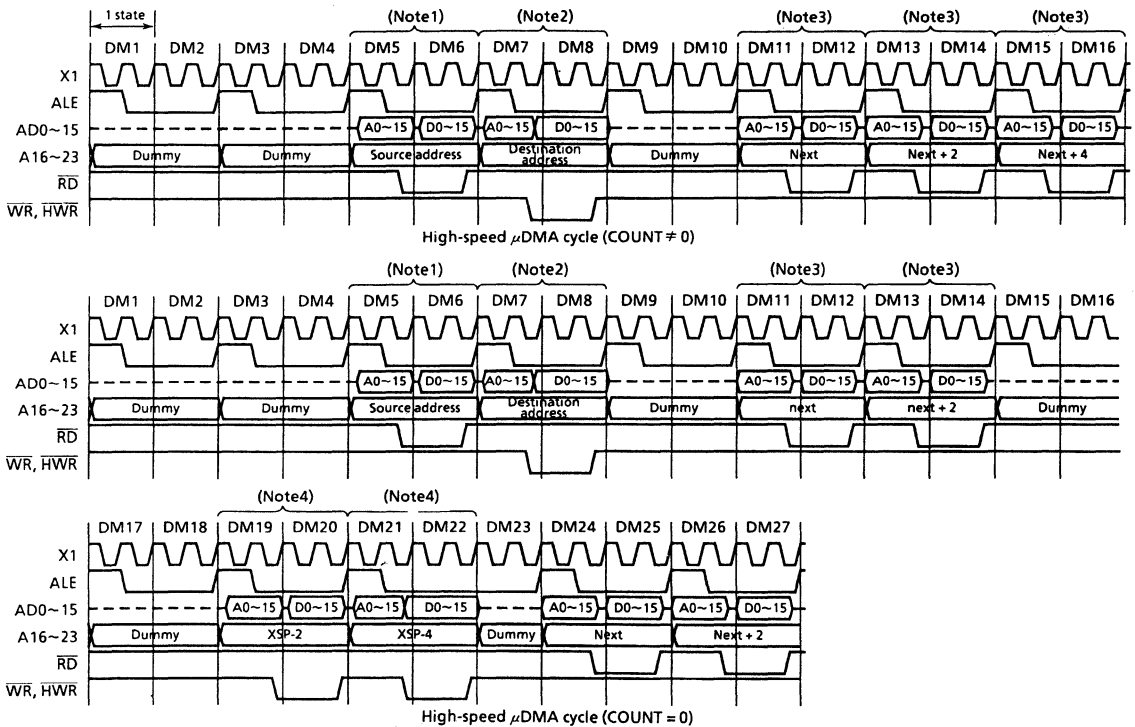
There are two data transfer modes: one-byte mode and one-word mode. Incrementing, decrementing, and fixing the transfer source/destination address after transfer can be done in both modes. Therefore data can easily be transferred between I/O and memory and between I/Os. For details of transfer modes,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

After transferring data using the high-speed  $\mu$ DMA 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  $\mu$ DMA cycle when the transfer destination address is in INC mode. (MIN mode, 16-bit bus for all address areas, 0 wait).

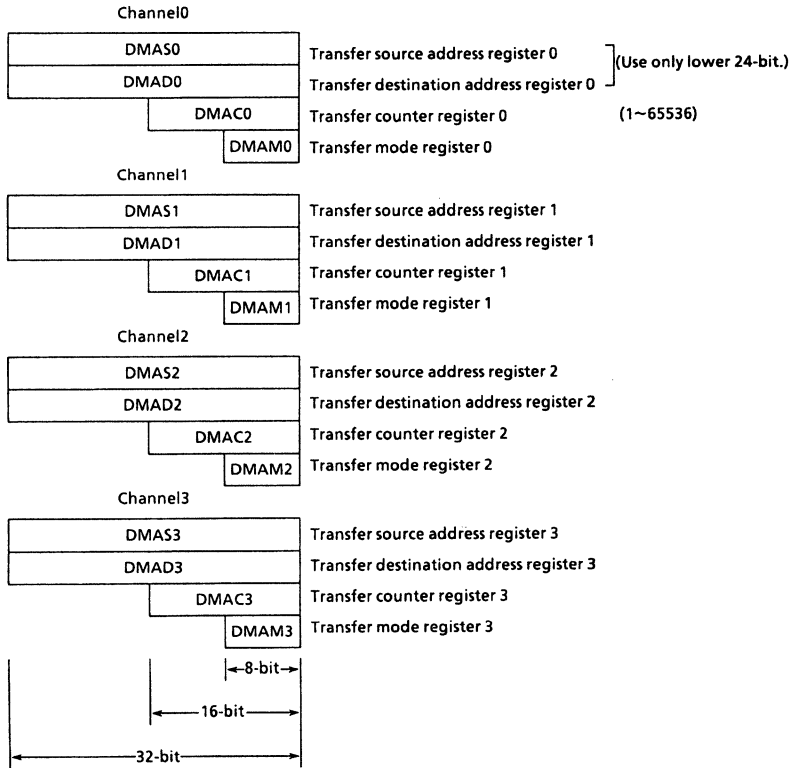
Interrupt sources processed by high-speed  $\mu$ DMA processing are those with the high-speed  $\mu$ DMA start vectors listed in Table 3.3 (1).



- 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	Z	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 accessed by high-speed μDMA. Do not use undefined codes for transfer mode control.

### 3.3.3 Interrupt Controller

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

Each interrupt channel (total of 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 dis-

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

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

The interrupt controller also has four registers used to store the high-speed other  $\mu$ DMA 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  $\mu$ DMA processing (see Table 3.3 (1)), enables the corresponding interrupt to be processed by  $\mu$ DMA processing. The values must be set in the  $\mu$ DMA parameter registers (e.g., DMAS and DMAD) prior to the  $\mu$ DMA processing.



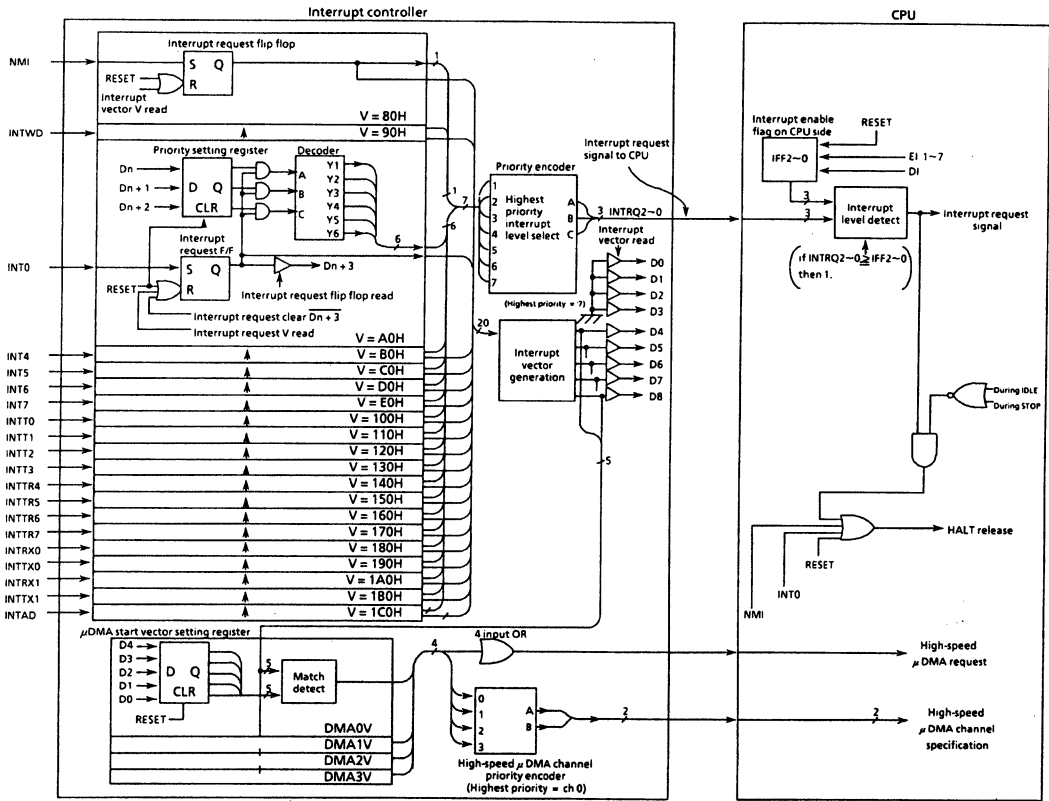


Figure 3.3.3 (1). Block Diagram of Interrupt Controller

(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
INTE10	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
INTE32	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
INTE54	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
INTE50	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
INTE51	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				

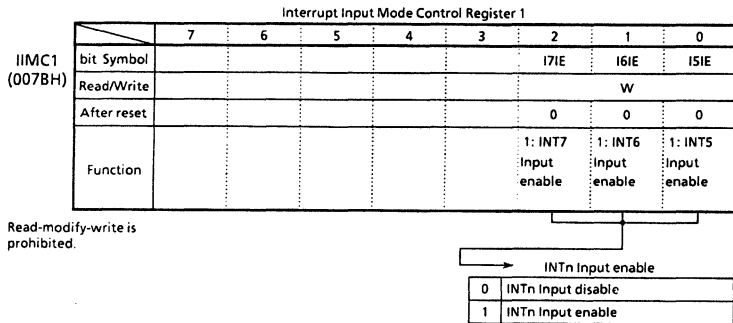
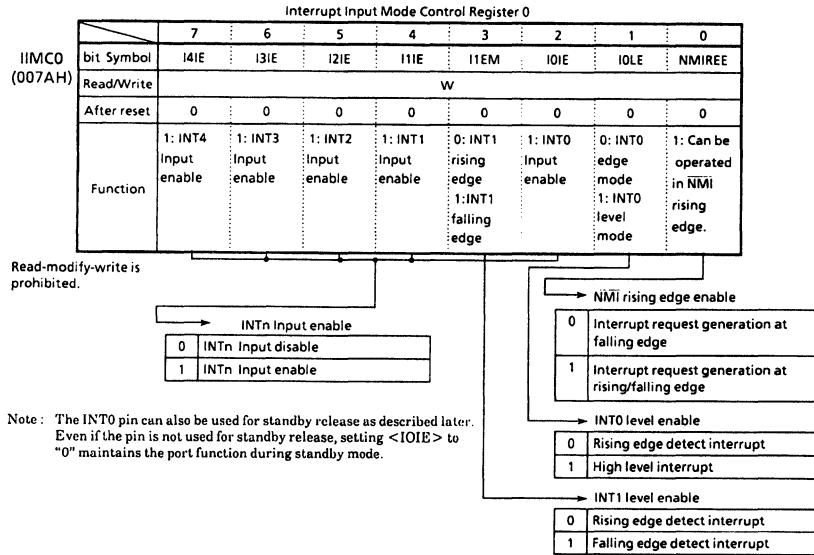
←Interrupt source  
 ←bit Symbol  
 ←Read/Write  
 ←After reset

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



**Setting of External Interrupt Pin Functions**

Interrupt	Pin Name	Mode	Setting Method
NMI	P34	Falling edge	IIMC <NMIREE> = 0
		Rising and falling edge	IIMC <NMIREE> = 1
INT0	P50	Rising edge	IIMC <IOLE> = 0, <IOIE> = 1
		Level	IIMC <IOLE> = 1, <IOIE> = 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 <I6IE> = 1
INT7	P35	Rising edge	IIMC <I7IE> = 1

**(3) High-Speed  $\mu$ DMA Start Vector**

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the interrupt vector with each channel's  $\mu$ DMA start vector (bits 4 to 8 of the interrupt vector). When both match, the interrupt is processed in  $\mu$ DMA

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.

		$\mu$ DMA0 Start Vector (read-modify-write is not possible.)							
		7	6	5	4	3	2	1	0
DMA0V (007CH)	bit Symbol				DMA0V8	DMA0V7	DMA0V6	DMA0V5	DMA0V4
	Read/Write				W				
	After reset	0							
	Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed $\mu$ DMA channel 0 is processed.							

		$\mu$ DMA1 Start Vector (read-modify-write is not possible.)							
		7	6	5	4	3	2	1	0
DMA1V (007DH)	bit Symbol				DMA1V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4
	Read/Write				W				
	After reset	0							
	Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed $\mu$ DMA channel 1 is processed.							

		$\mu$ DMA2 Start Vector (read-modify-write is not possible.)							
		7	6	5	4	3	2	1	0
DMA2V (007EH)	bit Symbol				DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4
	Read/Write				W				
	After reset	0							
	Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed $\mu$ DMA channel 2 is processed.							

		$\mu$ DMA3 Start Vector (read-modify-write is not possible.)							
		7	6	5	4	3	2	1	0
DMA3V (007FH)	bit Symbol				DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4
	Read/Write				W				
	After reset	0							
	Function	When bits 4 to 8 of the interrupt vector match bits 0 to 4 of DMA0V, high-speed $\mu$ DMA 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.

	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>	
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: $2^{16}/f_c$ 01: $2^{18}/f_c$ 10: $2^{20}/f_c$ 11: $2^{22}/f_c$ Detection time		Warming up time 0: $2^{16}/f_c$ 1: $2^{18}/f_c$	Standby mode 00: RUN 01: STOP 10: IDLE 11: Don't care		mode mode mode	1: Connects watchdog timer output to 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 for 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 NMI or INTO 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 INTO pin. Processing resumes from address next to HALT instruction.
IDLE	Can only be released by NMI or INTO pin. After standby mode is released, interrupt processing starts.	↑
STOP	↑	↑

Note 1: When releasing standby setting INTO 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/W/NMI)	Input mode Output mode NMI	PU* PU* Input	PU Output Input
P35 (RAS/INT7)	Input mode Output mode RAS	PU* PU* Output	PU Output Output
P40 - P42 (CS0 - CS2)	Output	PU*	Output
P43 (CS3/CAS)	Output 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/WDTOUT)	Input mode Output mode WDTOUT	- - Output	Input Output Output
P70 - P76	Input mode Output mode	- -	Input Output
ALE	Output	"0"	"0"
CLK	Output	-	"1"
RESET	Input	Input	Input
WR	Output	-	"1" Output
RD	Output	-	"1" Output
AM8/T16	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 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 func-

tions, 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.

**Table 3.5 (1) Port Function**

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

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	T05/HWR
	P31	1	Input/Output	↑	Bit	T10/WAIT
	P32	1	Input/Output	↑	Bit	BUSRQ
	P33	1	Input/Output	↑	Bit	BUSAK
	P34	1	Input/Output	↑	Bit	R/W/RAS
	P35	1	Input/Output	↑	Bit	RAS/INT7
Port4	P40	1	Output	↑	(Fixed)	CS0
	P41	1	Output	↑	(Fixed)	CS1
	P42	1	Output	↑	(Fixed)	CS2
	P43	1	Output	↑	(Fixed)	CS3/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/CTS0
	P63	1	Input/Output	—	Bit	PG03/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/WDTOUT
Port7	P70	1	Input/Output	—	Bit	T01/T04
	P71	1	Input/Output	—	Bit	T02/DMUX
	P72	1	Input/Output	—	Bit	INT4/T14
	P73	1	Input/Output	—	Bit	INT5/T15
	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 ( $\overline{\text{BUSAK}} = 0$ ).

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.
$\overline{\text{RD}}$ $\overline{\text{WR}}$	—	First sets all bits to high, then sets them to high impedance.
P30 ( $\overline{\text{HWR}}$ ) P34 ( $\overline{\text{R/W}}$ )	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 ( $\overline{\text{CS0}}$ ) P41 ( $\overline{\text{CS1}}$ ) P42 ( $\overline{\text{CS2}}$ ) P43 ( $\overline{\text{CS3}}$ )	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 ( $\overline{\text{DMUX}}$ ) P63 ( $\overline{\text{RFSH}}$ )	No status change. (Does not become high impedance.)	First sets all bits to high, then sets them to high impedance.
P35 ( $\overline{\text{RAS}}$ ) P43 ( $\overline{\text{CAS}}$ )	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.

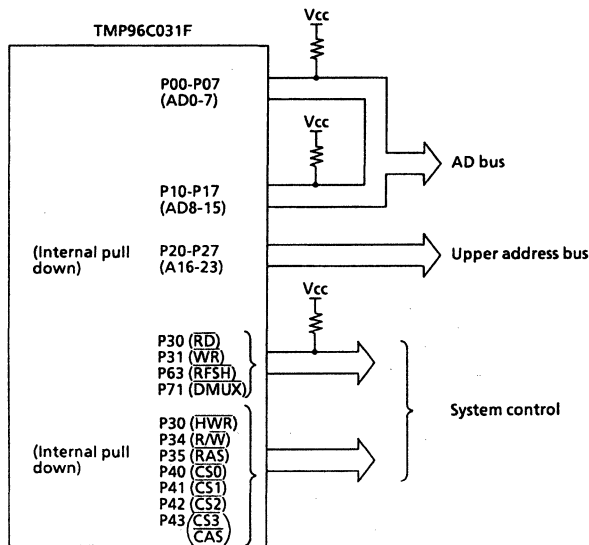


Figure 3.5 (1). External bus interface example when bus release function is in use



**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).

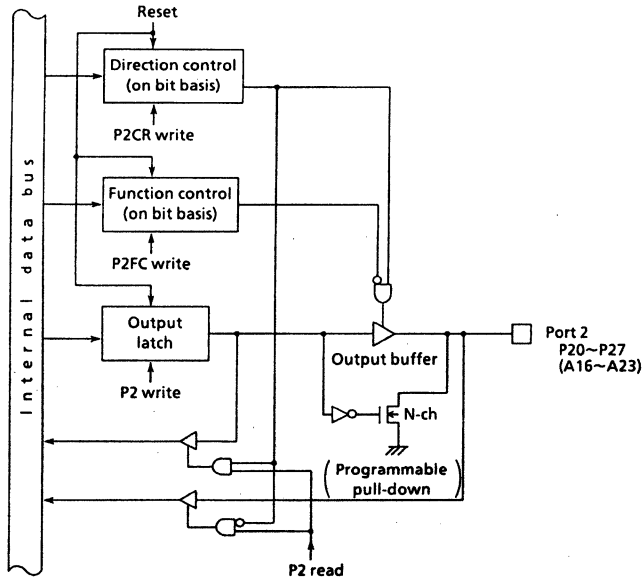


Figure 3.5 (2). Port 2

		Port 2 Register							
		7	6	5	4	3	2	1	0
P2 (0006H)	bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R/W							
	After reset	Input mode (Output latch register is cleared to "0".)							

		Port 2 Control Register							
		7	6	5	4	3	2	1	0
P2CR (0008H)	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							
		7	6	5	4	3	2	1	0
P2FC (0009H)	bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	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	0	Input	-	
1	1	Output	address bus (A23-16)	

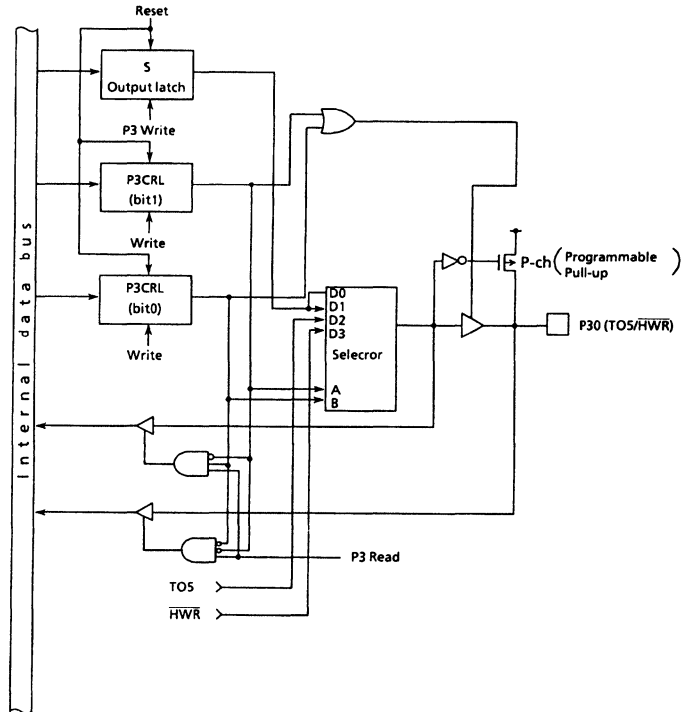
Note: <P2XF> is bit X in register P2FC; <P2XC> ; in register P2CR.

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)**

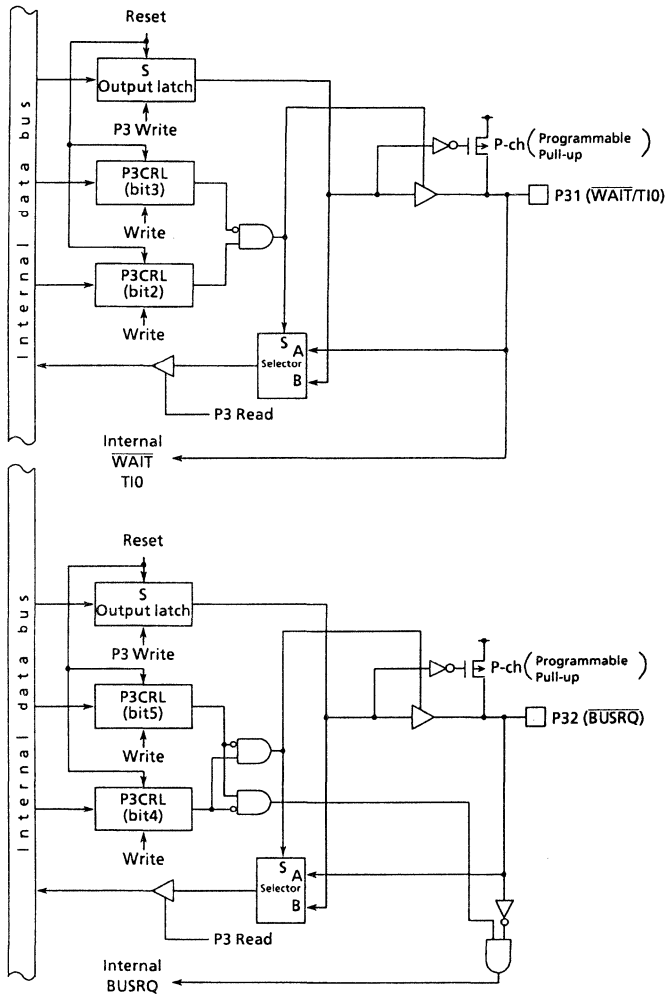


Figure 3.5 (5). Port 3 (P31, P32)

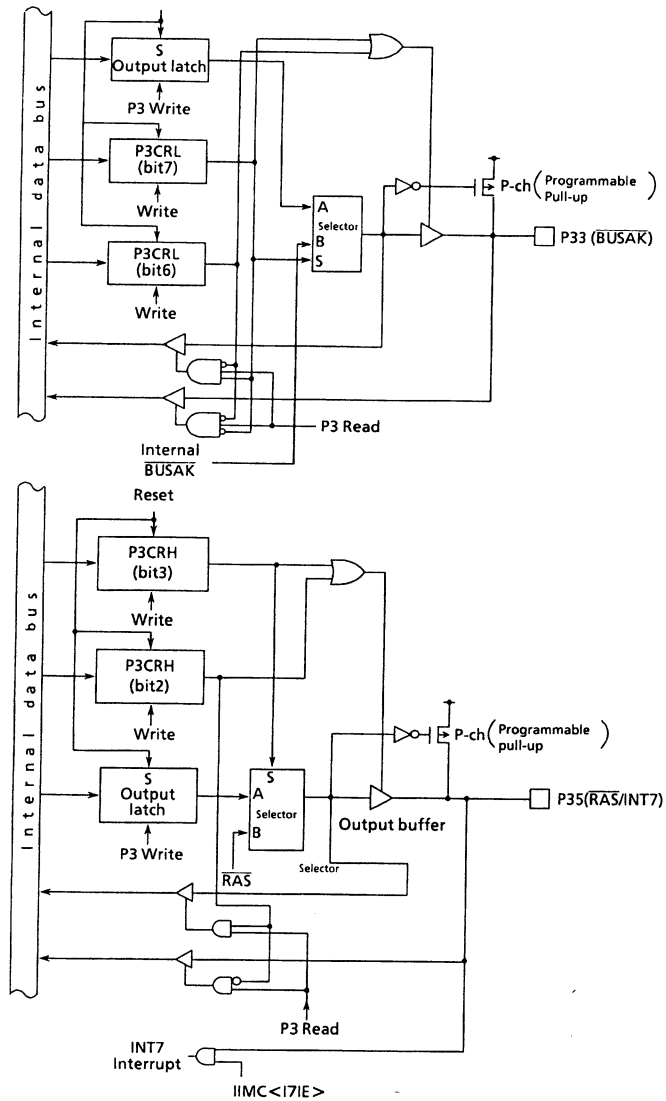


Figure 3.5 (6). Port 3 (P33, P35)

(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.

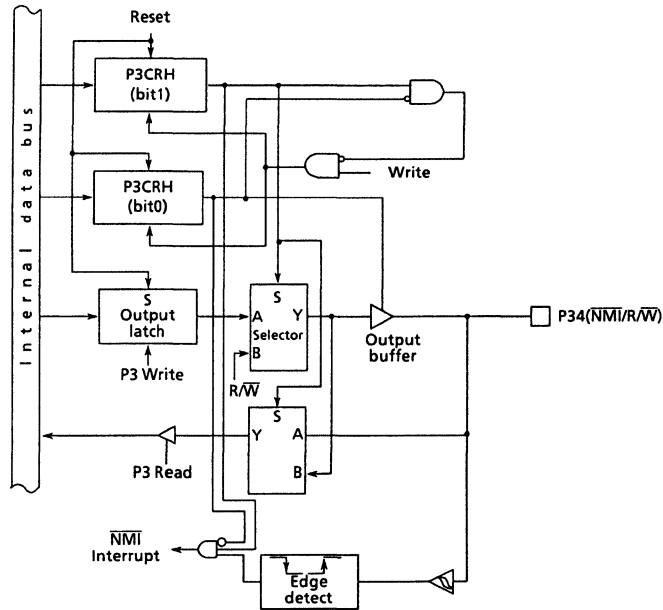


Figure 3.5 (6). Port 3 (P33, P34)

Port 3 Register

	7	6	5	4	3	2	1	0														
P3 (0007H)	bit Symbol		P35		P34		P33		P32		P31		P30									
	Read/Write												R/W									
	After reset												1		1		1		1		1	
	Function												Input mode (Pulled-up)									

Port 3 Control Register L

	7	6	5	4	3	2	1	0																				
P3CRL (000AH)	bit Symbol		P33C1		P33C0		P32C1		P32C0		P31C1		P31C0		P30C1		P30C0											
	Read/Write												W		W		W		W									
	After reset												0		0		0		0		0							
	Function												00: PORT input		01: PORT output		10: BUSAK		11: —		00: PORT input		01: PORT output		10: TOS		11: HW $\bar{R}$	

Port 3 Control Register H

	7	6	5	4	3	2	1	0																						
P3CRH (000BH)	bit Symbol		RDEN		P35C1		P35C0		P34C1		P34C0																			
	Read/Write												W		W		W													
	After reset												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.

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.

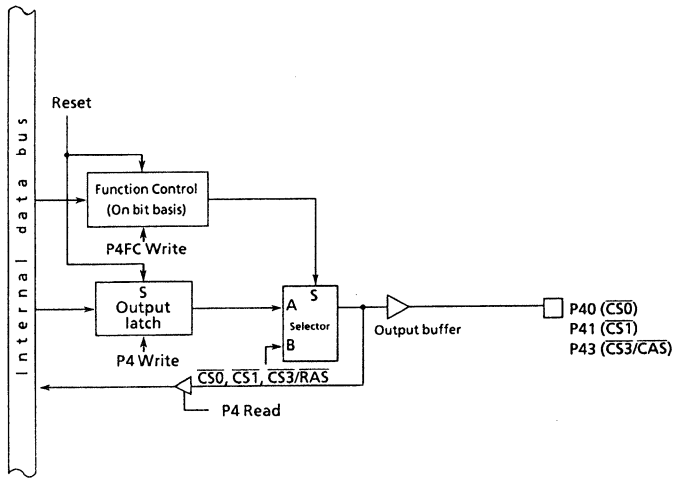
Figure 3.5 (8). Port 3 Registers

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)**



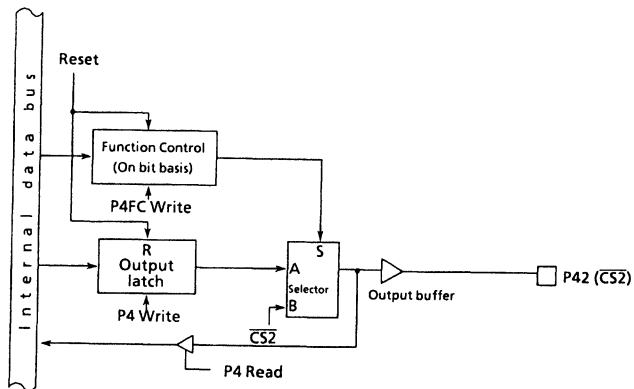


Figure 3.5 (10). Port 4 (P42)

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

		Port 4 Function Register										
		7	6	5	4	3	2	1	0			
P4FC (0010H)	bit Symbol	BUSWDT	/				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.

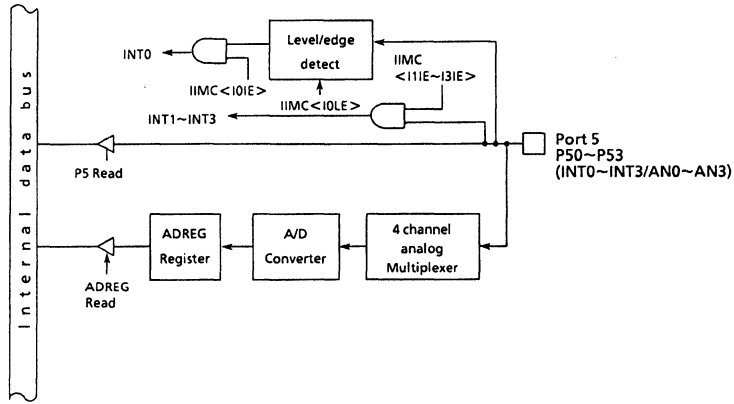


Figure 3.5 (11). Port 5 (P50, P51, P52, P53)

Port 5 Register

	7	6	5	4	3	2	1	0
P5 (000DH)	/				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 CTS0 output pin

P63: DRAM controller refresh signal pin

P66: external interrupt request input INT6 pin

P67: watchdog timer 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.

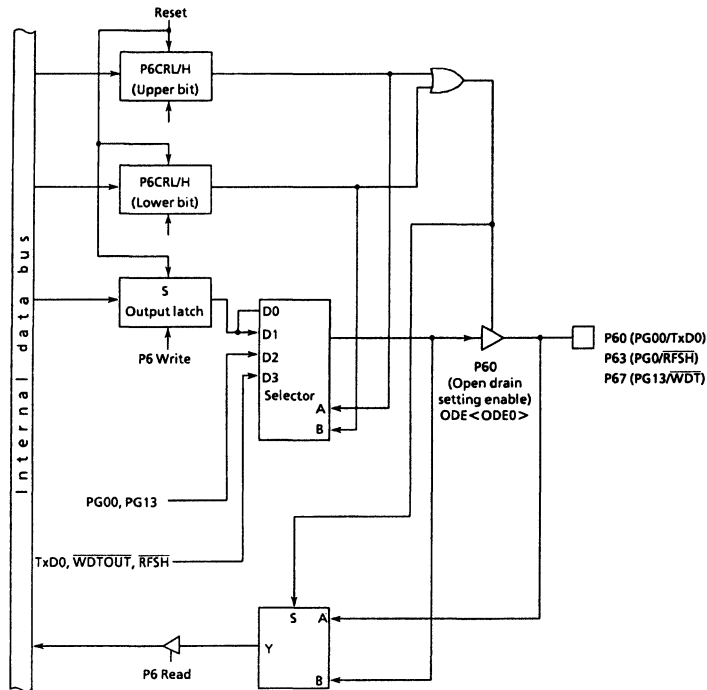


Figure 3.5 (13) Port 6 (P60, P67)

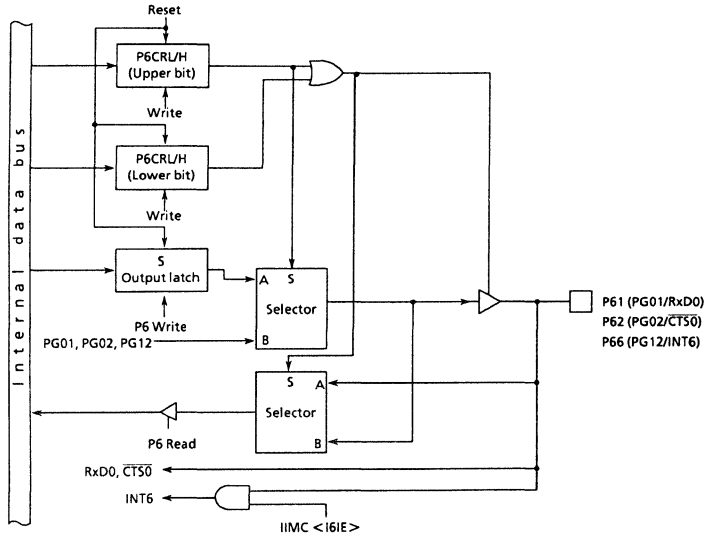


Figure 3.5 (14). Registers for Port 6 (P61, P62, P66)

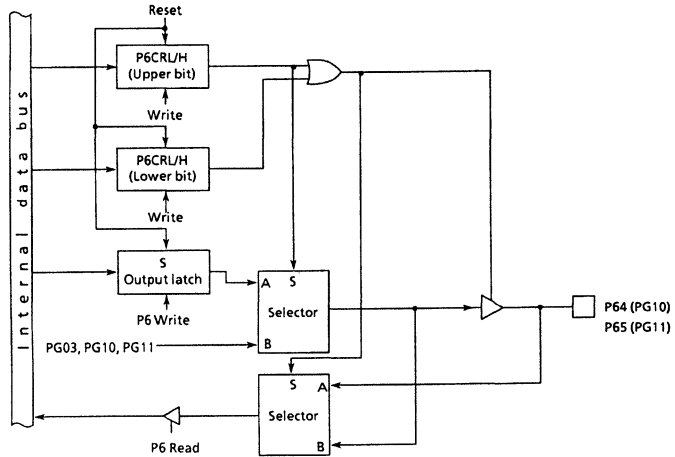


Figure 3.5 (15). Port 6 (P63, P64, P65)

		Port 6 Register							
		7	6	5	4	3	2	1	0
P6 (0012H)	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							
		7	6	5	4	3	2	1	0
P6CRL (0014H)	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: 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							
		7	6	5	4	3	2	1	0
P6CRH (0016H)	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: 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.

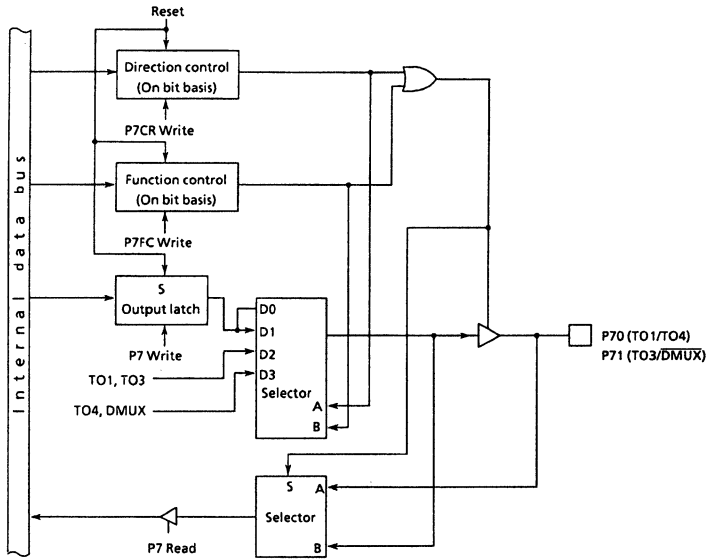
Note: To set the TXD0 pin to open drain output, write 1 in bit 0 <ODE0> in the ODE register. There is no port/function switch register for pin P61/RXD0. If pin P61 is used as an input port, data are input as serial receive data to SIO. When pin P66/PG12/ANT6 is used for INT6, set P6CRH <P66C1,0> to 00 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)**

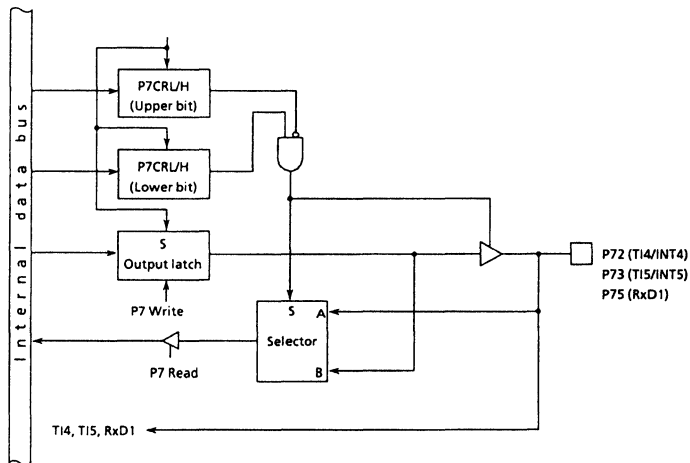


Figure 3.5 (18). Port 7 (P72, P73, P75)

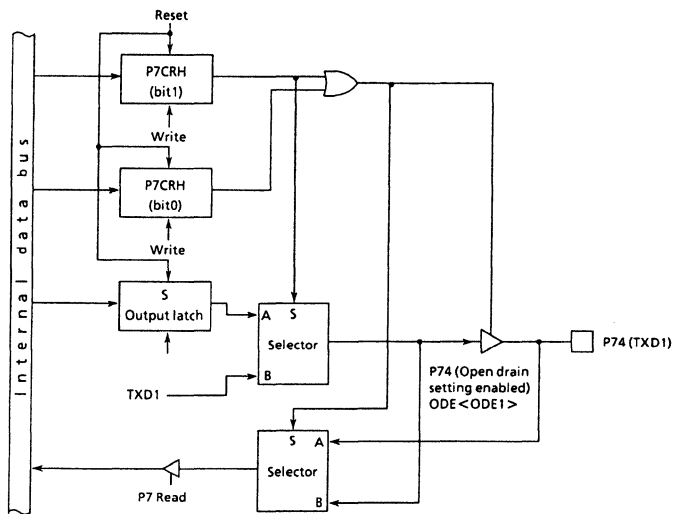


Figure 3.5 (19). Port 7 (P74)

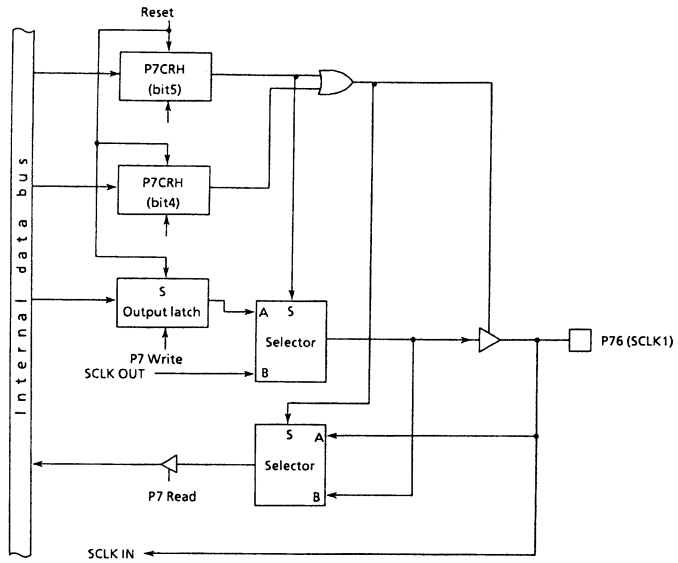


Figure 3.5 (20). Port 7 (P76)



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

		Port 7 Control Register L								
		7	6	5	4	3	2	1	0	
P7CRL (0015H)	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		PORT input		PORT input		PORT input	
		01:	PORT output		PORT output		PORT output		PORT output	
10:		—		—		TO3		TO1		
11:		—		—		DMUX		TO4		

		Port 7 Control Register H							
		7	6	5	4	3	2	1	0
P7CRH (0017H)	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		PORT input		PORT input
01:				PORT output		PORT output		PORT output	
10:				SCLK1		—		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 <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 11MC0<141E>/11MC1<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 (CS0 - CS3 pins), wait (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 (BOCS, 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: CS0 DIS 1: CS0 EN	1: SYSTEM ONLY	0: 7F00H ~7FFFH 1: address area specifi- cation	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: CS1 DIS 1: CS1 EN	↑	0: 80H ~7FFFH 1: address area specifi- cation	↑	↑			-
B2CS (006AH)	bit Symbol	B2E	B2SYS	B2ARE	B2BUS	B2W1	B2W0		
	Read/Write	W							
	After reset	1	0	0	Undefined	0	0		
	Function	0: CS2 DIS 1: CS2 EN	↑	0: 8000H ~3FFFFH 1: address area specifi- cation	↑	↑			-
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: CS3/ CAS DI 1: CS3/ CAS EN	↑	0: Un- defined 1: address area specifi- cation	↑	↑			0: CS3 output 1: CAS/ output

## (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 3FFFFFF.  $\overline{CS3}$  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 8-bit data bus mode. However, this bit is valid only in 16-bit bus mode ( $\overline{AM8}/\overline{T6}$  pin = "0"). In 8-bit bus mode ( $\overline{AM8}/\overline{T6}$  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 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	xxxxx	b7 - b0
			2n + 1	xxxxx	b15 - b8
	16-bit	2n + 2	xxxxx	b23 - b16	
		2n + 3	xxxxx	b31 - b24	
32-bit	2n + 1 (odd number)	8-bit	2n + 0	b15 - b8	b7 - b0
			2n + 2	b31 - b24	b23 - b16
	16-bit	2n + 1	xxxxx	b7 - b0	
		2n + 2	xxxxx	b15 - b8	
32-bit	16-bit	2n + 3	xxxxx	b23 - b16	
		2n + 4	xxxxx	b31 - b24	
32-bit	2n + 1 (odd number)	8-bit	2n + 3	xxxxx	b23 - b16
			2n + 4	xxxxx	b31 - b24
	16-bit	2n + 1	b7 - b0	xxxxx	
		2n + 2	b23 - b16	b15 - b8	
32-bit	16-bit	2n + 3	xxxxx	b23 - b16	
		2n + 4	xxxxx	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  $\overline{\text{WAIT}}$  pin status. Setting them to 01 inserts a 1-state wait regardless of the  $\overline{\text{WAIT}}$  status. Setting them to 10 inserts a 1-state wait and samples the  $\overline{\text{WAIT}}$  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  $\overline{\text{WAIT}}$  pin status. Resetting sets these bits to 00 (2-state wait mode).

Note: If there is 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 ( $\overline{\text{CS3/CAS}}$ ) pin. When this bit is set to "0",  $\overline{\text{CS3}}$  waveform is output. When it is set to "1",  $\overline{\text{CAS}}$  waveform is output. This bit is cleared to zero after reset.

(7) Self refresh control

(described in section 3.13.1 Refresh Controller.)

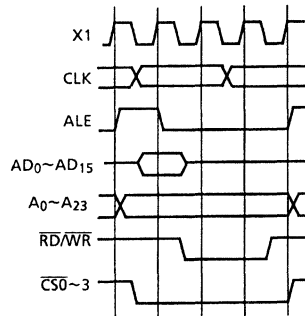
(8) Wait control outside space  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$

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

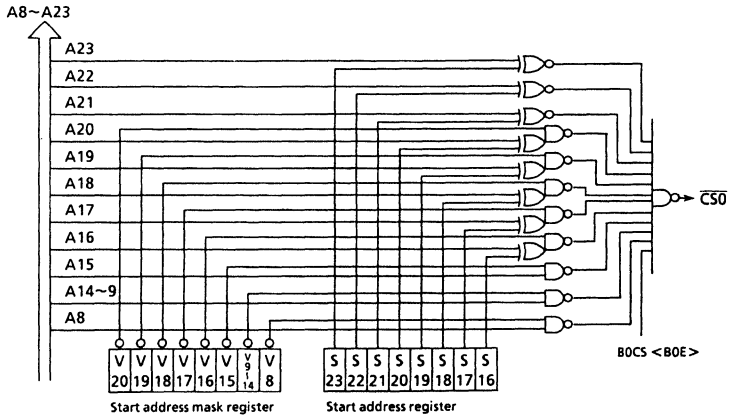


Figure 3.6 (2).  $\overline{CS0}$  Address Decode Block Diagram

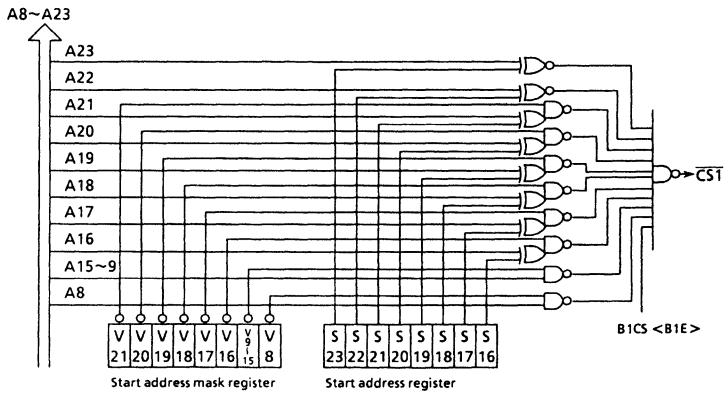


Figure 3.6 (3).  $\overline{CS1}$  Address Decode Block Diagram

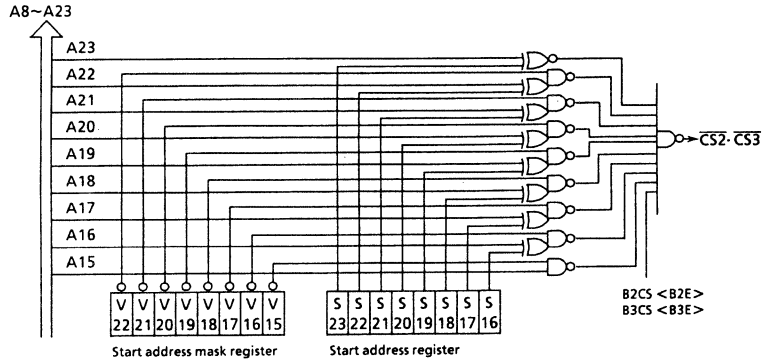


Figure 3.6 (4).  $\overline{CS2}$ ,  $\overline{CS3}$  Address Decode Block Diagram

(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	
		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 ( $\overline{CS0}$ )

	7	6	5	4	3	2	1	0
bit symbol	V20	V19	V18	V17	V16	V15	V14~9	V8
MAMR0 (0041H) 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 ( $\overline{CS1}$ )

	7	6	5	4	3	2	1	0
bit symbol	V21	V20	V19	V18	V17	V16	V15~9	V8
MAMR1 (0043H) 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 ( $\overline{CS2}$ ,  $\overline{CS3}$ )

	7	6	5	4	3	2	1	0
bit symbol	V22	V21	V20	V19	V18	V17	V16	V15
MAMR2 (0045H) / MAMR3 (0047H) 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  
(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 CS0 is enabled only when they match.

CS1, CS2, and 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 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.

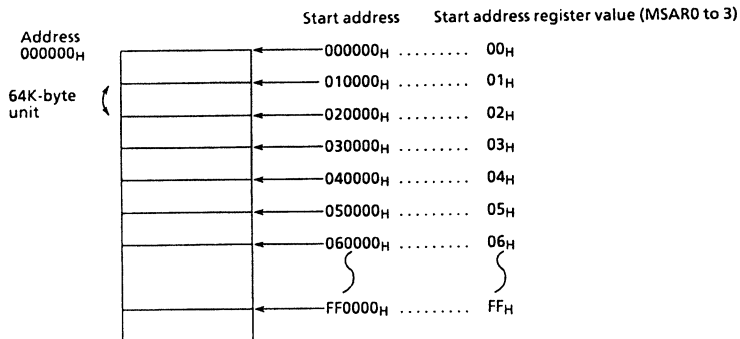


Figure 3.6 (5). Where to Set Start Address

**(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.

SIZE $\overline{CS}$	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.

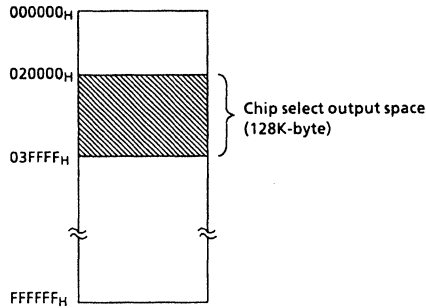
(Setting example 1)  $\overline{CS0}$

chip select output is as shown in the following memory map.

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

MSAR setting	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14-9	S8
	0	0	0	0	0	0	1	0	0	0	0
	default "0"										
MAMR setting	V23	V22	V21	V20	V19	V18	V17	V16	V15	V14-9	V8
	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) ( $\overline{CS0}$ )

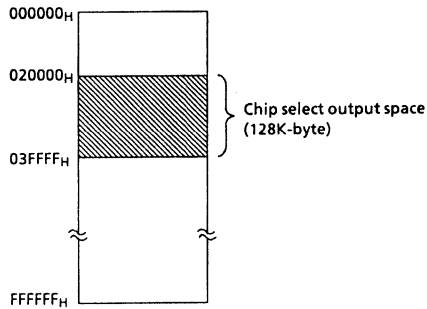
chip select output is as shown in the following memory map.

When MSAR is set to 01H and MSAR is set to 04H, the

MSAR setting	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14~9	S8
	0	0	0	0	0	0	0	1	0	0	0
	default "0"										
MAMR setting	V23	V22	V21	V20	V19	V18	V17	V16	V15	V14~9	V8
	0	0	0	0	0	0	0	0	1	0	0
	default "0"										

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) (CS0)

MAMR (excerpt).

Space where chip select is output by values set in MSAR and

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

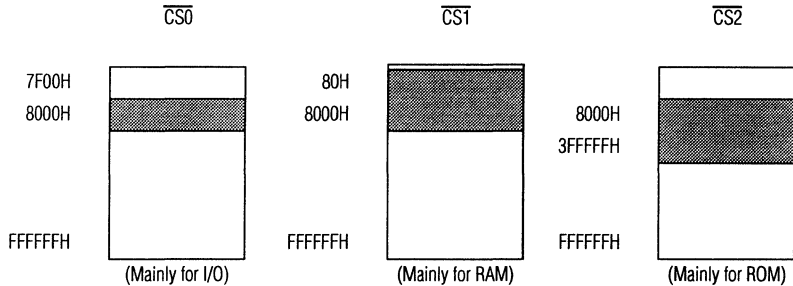
The following figures show the actual chip select image.  $\overline{CS0}$  can specify 7F00H to 7FFFH,  $\overline{CS1}$  can specify 80H to 7FFFH, and  $\overline{CS2}$  can specify 8000H to 3FFFFFFH. This is because external connection of devices (such as RAM or I/O) other than ROM is considered.

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 3FFFFFFH (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 AMB/ $\overline{16}$  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 7F00H 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.

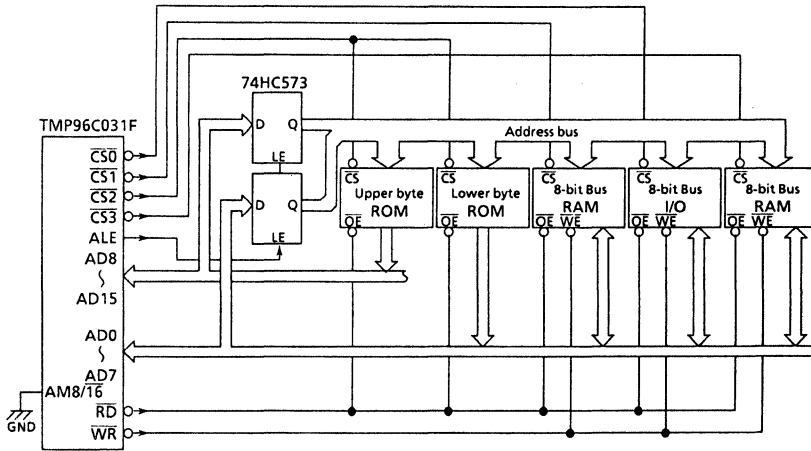


Figure 3.6 (7). Example of External Memory Connection (ROM = 16-bit, RAM and I/O = 8-bit)

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	(BOCS),	10010000B	; $\overline{CS0}$ = 8-bit, 2WAIT, 7F00H - 7FFFH, 2WAIT other than in $\overline{CS0}$ - $\overline{CS3}$ areas
LD	(B1CS),	100111XXB	; $\overline{CS1}$ = 8-bit, 0WAIT, 80H - 7FFFH
LD	(B2CS),	100001XXB	; $\overline{CS2}$ = 16-bit, 1WAIT, 8000H - 3FFFFFFH
LD	(B3CS),	10111100B	; $\overline{CS3}$ = 8-bit, 0WAIT, address area specification (400000H - 407FFFFH)
LD	(MSAR3),	01000000B	; $\overline{CS3}$ start address: 400000H
LD	(MAMR3),	00000000B	; $\overline{CS3}$ area = 32K-byte
LD	(P4FC),	XXXX1111B	; $\overline{CS0}$ - $\overline{CS3}$ output mode

Note: X: don't care

(2) Connection example 2

memory is connected to the TMP96C031. In this example, the ROM, RAM, and I/O are connected with 8-bit width.

Figure 3.6 (7) is an example (2) in which an external

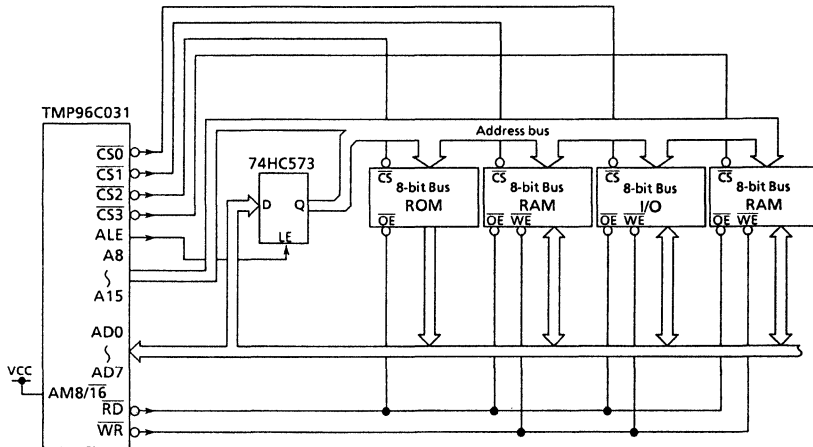


Figure 3.6 (8). Example of External Memory Connection (ROM = 16-bit, RAM and I/O = 8-bit)

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 ;  $\overline{CS0}$  = 8-bit, 2WAIT, 7F00H - 7FFFH, 2WAIT other than in  $\overline{CS0}$  -  $\overline{CS3}$  areas
LD (B1CS), 100111XXB ;  $\overline{CS1}$  = 8-bit, 0WAIT, 80H - 7FFFH
LD (B2CS), 100001XXB ;  $\overline{CS2}$  = 16-bit, 1WAIT, 8000H - 3FFFFFFH
LD (B3CS), 10111100B ;  $\overline{CS3}$  = 8-bit, 0WAIT, address area specification (400000H ~ 407FFFFH)
LD (MSAR3), 01000000B ;  $\overline{CS3}$  start address: 400000H
LD (MAMR3), 00000000B ;  $\overline{CS3}$  area = 32K-byte
LD (P4FC), XXXX1111B ;  $\overline{CS0}$  -  $\overline{CS3}$  output mode
Note: X: don't care

```

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

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:

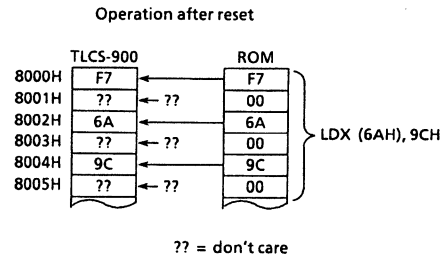
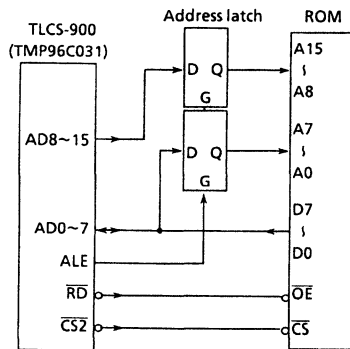
```

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 - 3FFFFFFH) 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)
  - 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)
- ) Either two 8-bit buses or  
one 16-bit bus can be used.

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 T10, 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  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$ , and  $\phi T256$  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.

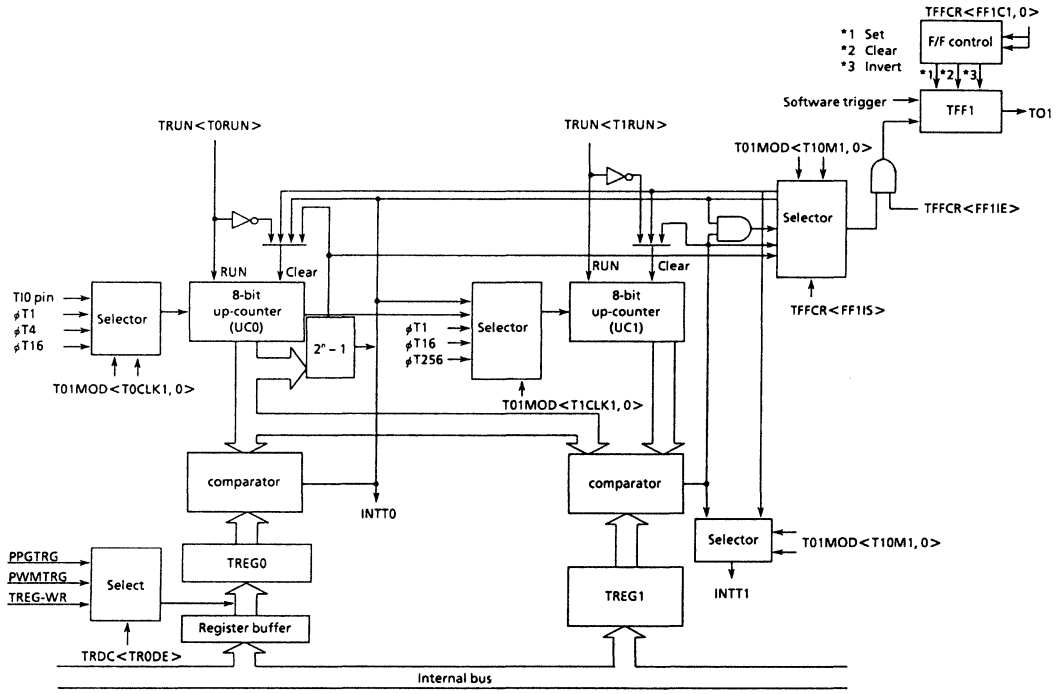


Figure 3.7 (1). Block Diagram of 8-Bit Timers (Timers 0 and 1)

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 ( $f_c$ ) after it has been divided by 4 ( $f_c/4$ ).

Among them, 8-bit timer uses 4 types of clock:  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$ , and  $\phi T256$ .

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	$f_c$
$\phi T1$ ( $8/f_c$ )	20MHz
$\phi T4$ ( $32/f_c$ )	400ns
$\phi T16$ ( $128/f_c$ )	$1.6\mu s$
$\phi T256$ ( $2048/f_c$ )	$6.4\mu s$
	$102\mu s$

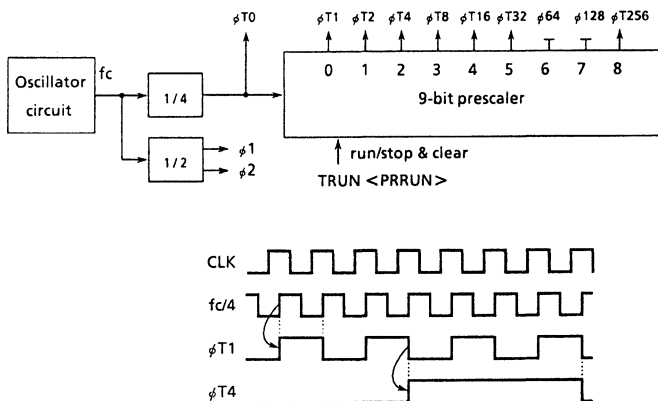


Figure 3.7 (2). Prescaler

## 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  $\phi$  T1,  $\phi$  T4, and  $\phi$  T16 depending to the value set in the T10 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  $\phi$  T1,  $\phi$  T16, and  $\phi$  T256 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 overflow output of timer 0 becomes the input clock of timer 1 (16-bit timer).

When T01MOD7, 6 = 00, T01MOD3, 2 = 01,  $\phi$  T1 (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

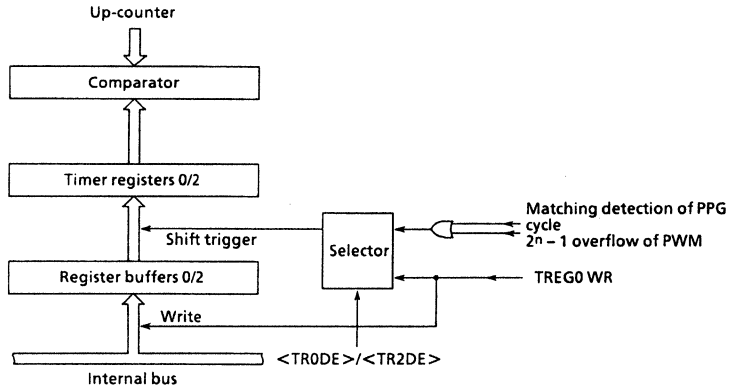
This is an 8-bit register for setting an interval time. When the set value of timer registers TREG0, TREG1, TREG2, TREG3 matches the value of up-counter, the comparator match detect signal becomes active. If the set value is 00H, this signal becomes active when the up-counter overflows.

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 <TRODE, TR2DE>. It is disabled when <TRODE>/<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 <TRODE>/<TR2DE> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <TRODE>/<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.



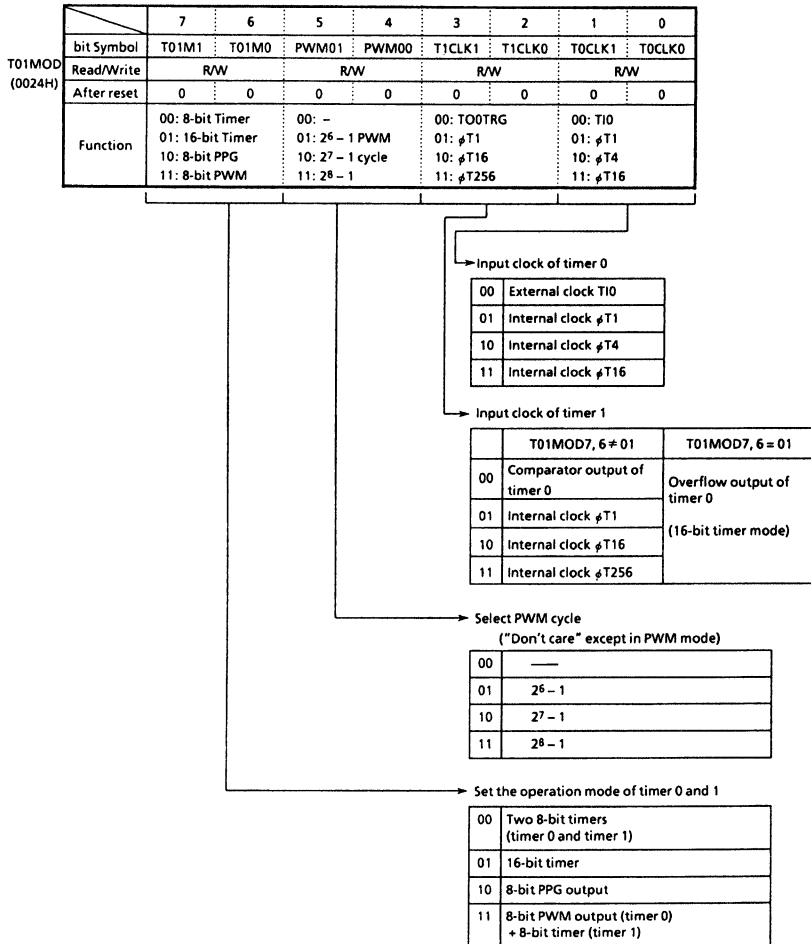


Figure 3.7 (4). Timer 0, 1 Mode Register (T01MOD)

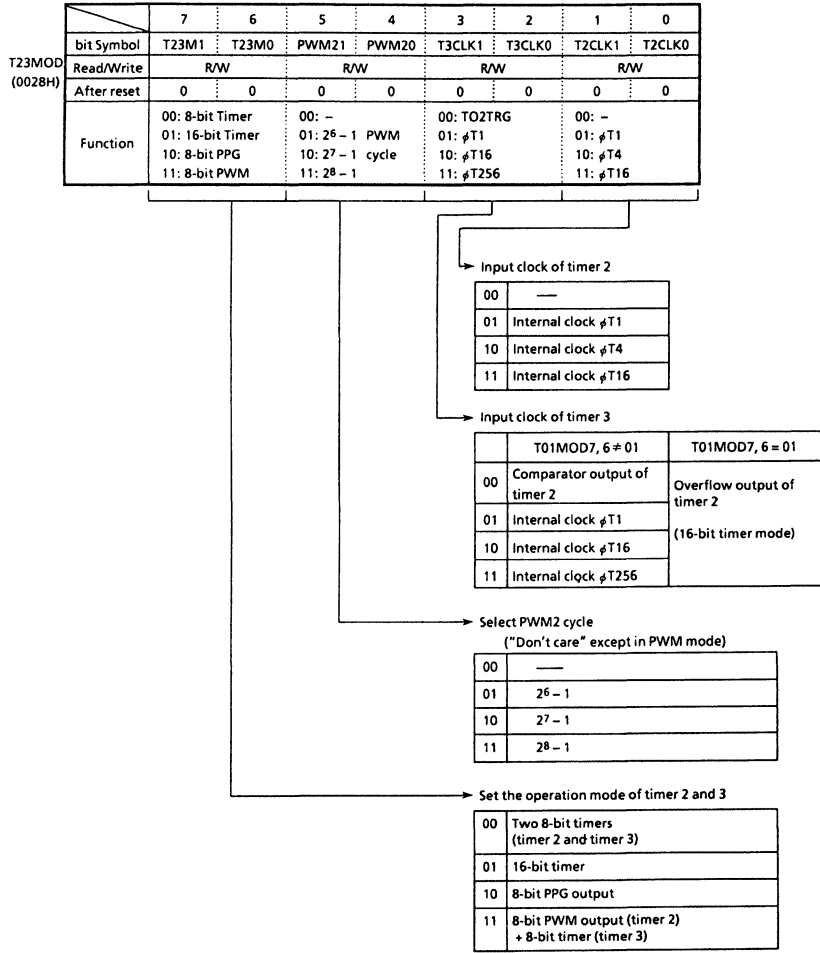


Figure 3.7 (5). Timer 2,3 Mode Register (T23MOD)

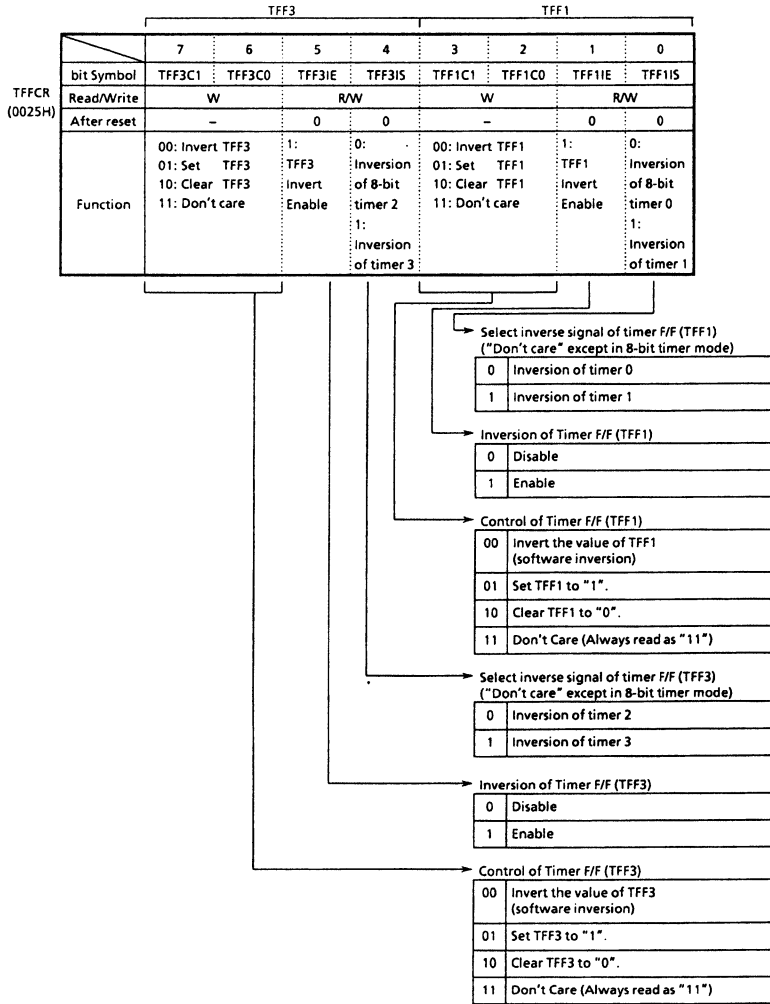


Figure 3.7 (6). 8-Bit Timer Flip-Flop Control Register (TFFCR)

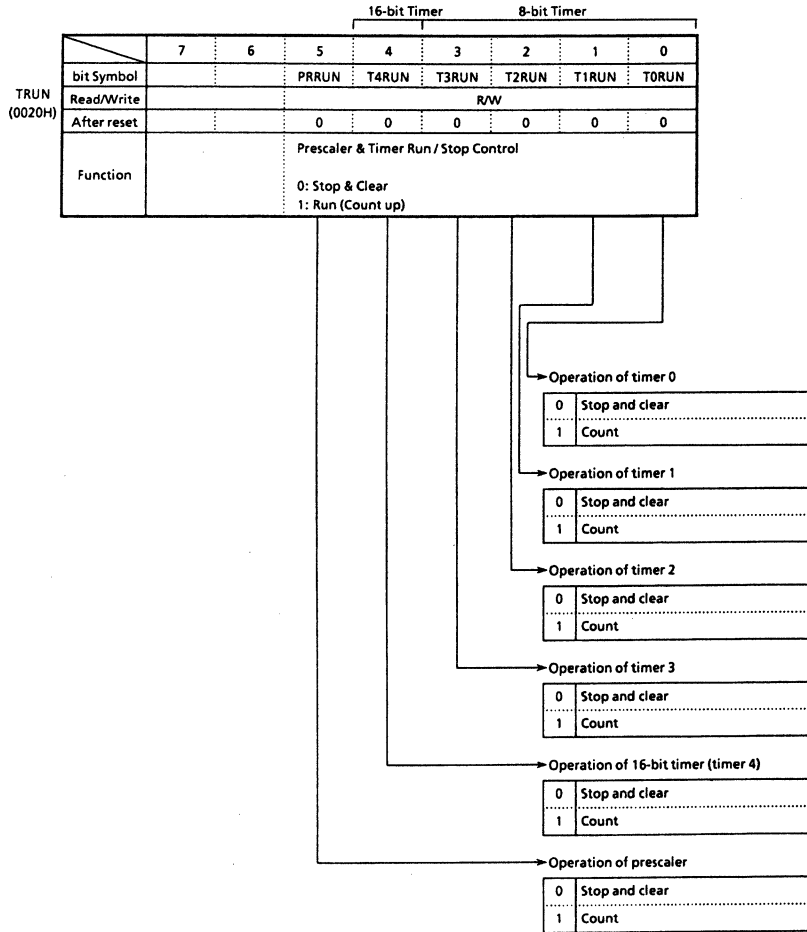


Figure 3.7 (7). Timer Operation Control Register (TRUN)

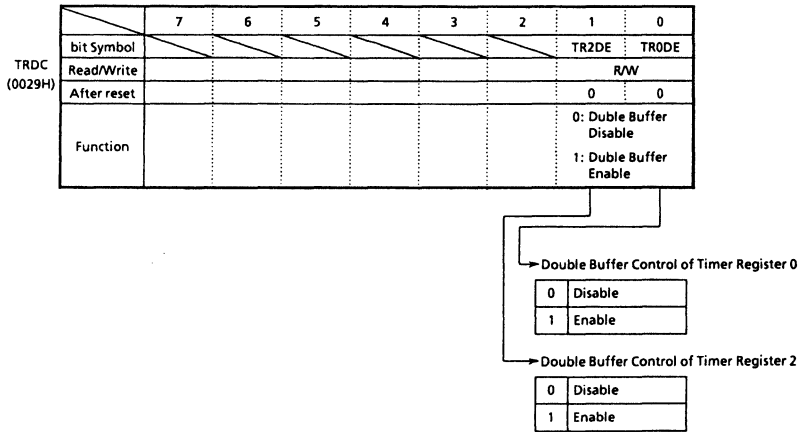


Figure 3.7 (8). Timer Register Double Buffer Control Register (TRDC)

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 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  $f_c = 16\text{MHz}$ , set each register in the following manner.

	MSB							LSB	
	7	6	5	4	3	2	1	0	
TRUN	← -	-	-	-	-	-	-	0	-
T01MOD	← 0	0	x	x	0	1	-	-	
TREG1	← 0	1	1	0	1	0	0	0	
INTT1	← 1	1	0	1	-	-	-	-	
TRUN	← x	x	1	-	-	-	1	-	

Note: x; don't care      -; no change

- Stop timer 1, and clear it to "0".
- Set the 8-bit timer mode, and select  $\phi T1$  ( $0.5\mu s @ f_c = 16\text{MHz}$ ) as the input clock.
- Set the timer register at  $40\mu s \phi T1 = 50H$ .
- Enable INTT1, and set it to "Level 5".
- Start timer 1 counting.

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 $f_c = 20\text{MHz}$ )	Resolution
$\phi T1$ (8/ $f_c$ )	$0.4\mu s \sim 102.4\mu s$	$0.4\mu s$
$\phi T4$ (32/ $f_c$ )	$1.6\mu s \sim 409.6\mu s$	$1.6\mu s$
$\phi T16$ (128/ $f_c$ )	$6.4\mu s \sim 1.638\text{ms}$	$6.4\mu s$
$\phi T256$ (2048/ $f_c$ )	$102.4\mu s \sim 2.621\text{ms}$	$102.4\mu s$

② Generating a 50% duty square wave pulse

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

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

	MSB				LSB			
	7	6	5	4	3	2	1	0
TRUN	← -	-	-	-	-	-	0	-
TO1MOD	← 0	0	x	x	0	1	-	-
TREG1	← 0	0	0	0	0	0	1	1
TFFCR	← -	-	-	-	1	0	1	1
P7CRL	← -	-	-	-	-	-	1	0
TRUN	← x	x	1	-	-	-	1	-

Stop timer 1, and clear it to "0".

Set the 8-bit timer mode, and select φT1 as the input clock.

Set the timer register at  $2.4\mu\text{s} + \phi T1 + 2 = 3$ .

Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.

Select P71 as TO1 pin.

Start timer 1 counting.

Note: x; don't care    -; no change

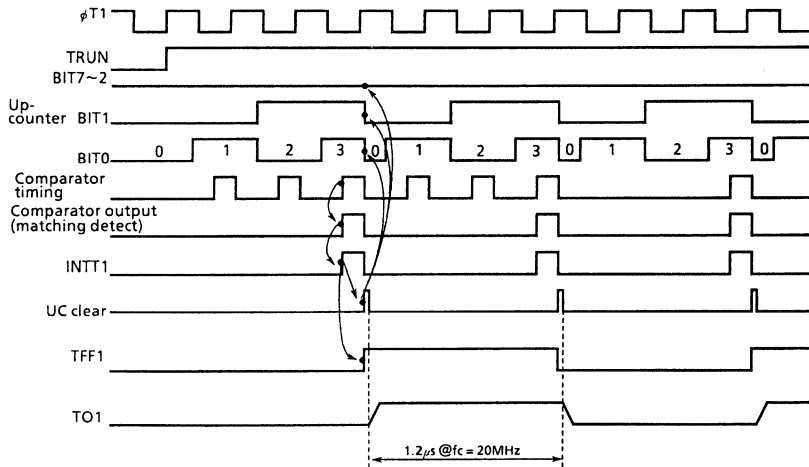


Figure 3.7 (9). Square Wave (50% Duty) Output Timing Chart

- 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.

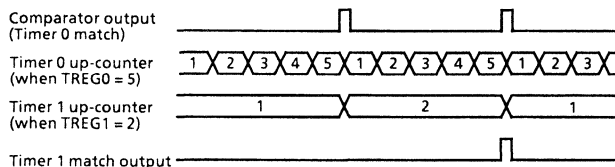


Figure 3.7 (10). Timer 1 Count Up by Timer 0

- 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 $f_c = 20\text{MHz}$ )	Resolution
$\phi T1$ (8/fc)	0.4 $\mu\text{s}$ - 26.214ms	0.4 $\mu\text{s}$
$\phi T4$ (32/fc)	1.6 $\mu\text{s}$ - 104.857ms	1.6 $\mu\text{s}$
$\phi T16$ (128/fc)	6.4 $\mu\text{s}$ - 419.430ms	6.4 $\mu\text{s}$

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  $f_c = 20\text{MHz}$ , set the following values for timer registers TREG0 and TREG1:

When counting with input clock of  $\phi T16$  (8 $\mu\text{s}$  @ 16MHz) 0.4 sec  $\div$  4 $\mu\text{s}$  = 62500 = F424H

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

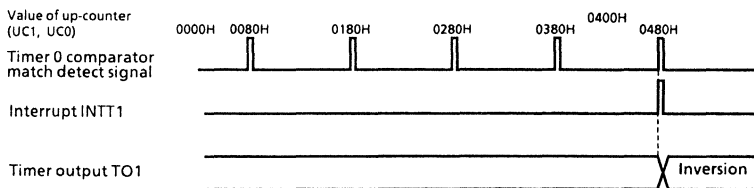
The comparator match signal is output from timer 0 each time the up-counter UC0 matches TREG0, where the up-counter UC0 is not to be cleared.

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 <sup>8</sup> + 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)

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



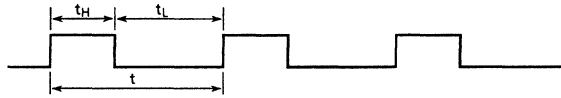
**Figure 3.7 (11). Output Timer by 16-Bit Timer Mode**

- (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.

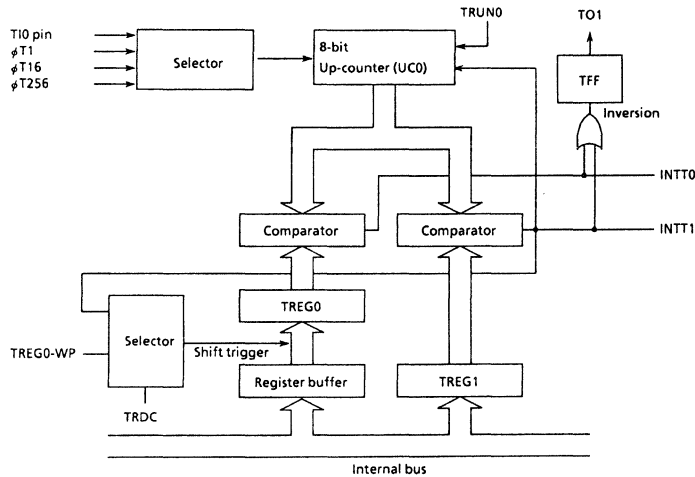
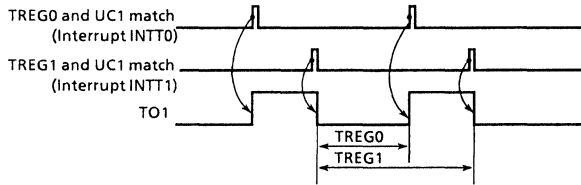
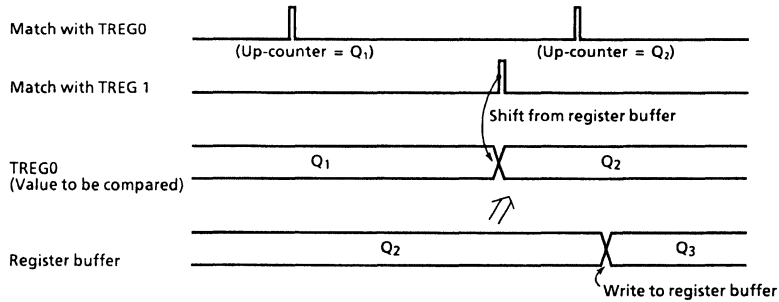


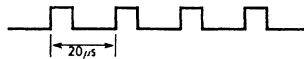
Figure 3.7 (12). Block Diagram of 8-Bit PPG Output Mode

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:  $t = 1/62.5\text{kHz} = 16\mu\text{s}$ .  
 Given  $\phi T1 = 0.4\mu\text{s @ } 20\text{MHz}$ ,  
 $16\mu\text{s} \div 0.4\mu\text{s} = 40$

Consequently, to set the timer register 1 (TREG1) to TREG1 = 40 = 28H and then duty to 1/4,  $t \times 1/4 = 16\mu\text{s} \times 1/4 = 4\mu\text{s}$   
 $4\mu\text{s} \div 0.4\mu\text{s} = 10$   
 Therefore, set timer register 0 (TREG0) to TREG0 = 10 = 0AH.

	MSB				LSB			
	7	6	5	4	3	2	1	0
TRUN	← x	x	-	-	-	-	0	0
T01MOD	← 1	0	x	x	x	x	0	1
TFFCR	← -	-	-	-	0	1	1	x
TREG0	← 0	0	0	0	1	0	1	0
TREG1	← 0	0	1	0	1	0	0	0
P7CRL	← -	-	-	-	-	-	1	0
TRUN	← x	x	1	-	-	-	1	1

Note: x; don't care    -; no change

Stop timer 0, and clear it to "0".

Set the 8-bit PPG mode, and select  $\phi T1$  as input clock.

Sets TFF1 and enables the inversion and double buffer enable.

Writing "10" provides negative logic pulse.

Write "0AH".

Write "28H".

Set P70 as the TO1 pin.

Start timer 0 and timer 1 counting.

(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  $2^n - 1$  ( $n = 6, 7, \text{ or } 8$ ; specified by T01MOD) counter overflow occurs. Up-counter UC1 is cleared when  $2^n - 1$  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  $\neq 0$ )

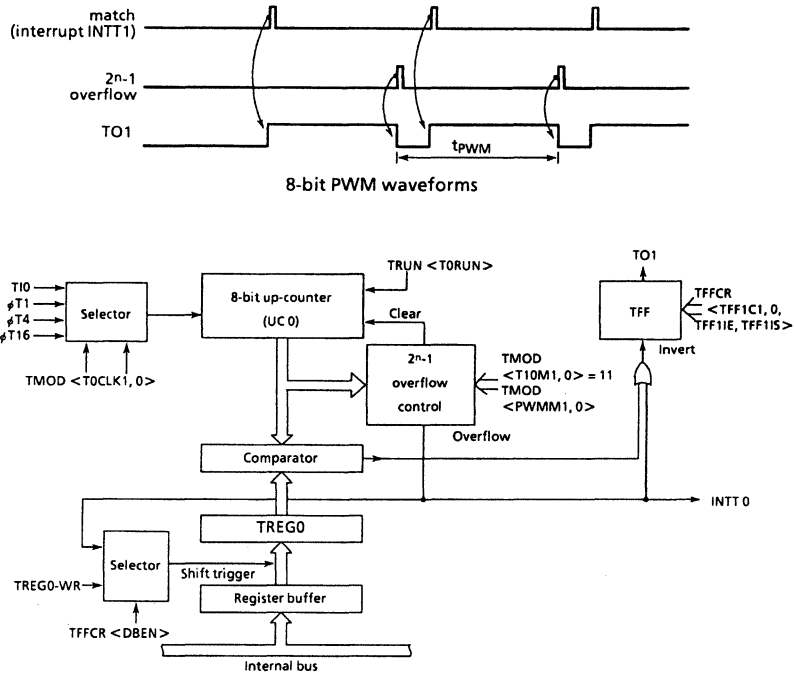
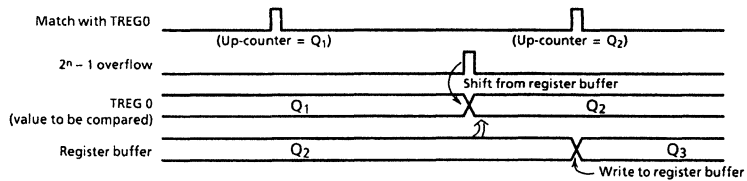


Figure 3.7 (13). Block Diagram of 8-Bit PWM Waveforms

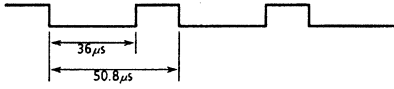
In this mode, the value of register buffer will be shifted in TREG0 if  $2^n - 1$  overflow is detected when the double buffer of TREG0 is enabled.

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  $f_c = 20\text{MHz}$ .



To realize  $50.8\mu\text{s}$  of PWM cycle by  $\phi T1 = 0.4\mu\text{s}$  ( $@ f_c = 20\text{MHz}$ ),

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

Consequently,  $n$  should be set to 7.

As the period of low level is  $36\mu\text{s}$ , for  $\phi T1 = 0.4\mu\text{s}$ , set the following value for TREG0:

$$36\mu\text{s} \div 0.4\mu\text{s} = 90 = 5\text{AH}$$

	MSB							LSB
	7	6	5	4	3	2	1	0
TRUN	← x	x	-	-	-	-	-	0
TO1MOD	← 1	1	1	0	-	-	0	1
TFFCR	← -	-	-	-	1	0	1	x
TREG0	← 0	1	0	1	1	0	1	0
P7CRL	← -	-	-	-	-	-	1	0
TRUN	← x	x	1	-	-	-	-	1

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

Stop timer 0, and clear it to "0".

Set 8-bit PWM mode (cycle :  $2^7 - 1$ )

and select  $\phi T1$  as the input clock.

Clears TFF1, enables the inversion and double buffer.

Write "5AH".

Set P70 as the TO1 pin.

Start timer 0 counting.

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

	PWM Cycle (@ $f_c = 20\text{MHz}$ )		
	$\phi T1$	$\phi T4$	$\phi T16$
$2^6 - 1$	25.2 $\mu\text{sec}$ (39.0kHz)	100 $\mu\text{sec}$ (10.0kHz)	4.03msec (2.4kHz)
$2^7 - 1$	50.8 $\mu\text{sec}$ (19.7kHz)	203 $\mu\text{sec}$ (4.9kHz)	812msec (1.2kHz)
$2^8 - 1$	102 $\mu\text{sec}$ (9.80kHz)	408 $\mu\text{sec}$ (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, $\phi$ T1, 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, $\phi$ T1, 4, 16)	0 : Lower timer 1 : Upper timer
8-bit timer x 2channel	00	-	( $\phi$ T1, T16, T256)	(External clock, $\phi$ T1, 4, 16)	0 : Lower timer 1 : Upper timer
8-bit PPG x 1channel	10	-	-	(External clock, $\phi$ T1, 4, 16)	-
8-bit PWM x 1channel (Lower) 8-bit timer x 1channel (Upper)	11	PWM cycle	( $\phi$ T1, T16, T256)	(External clock, $\phi$ T1, 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).



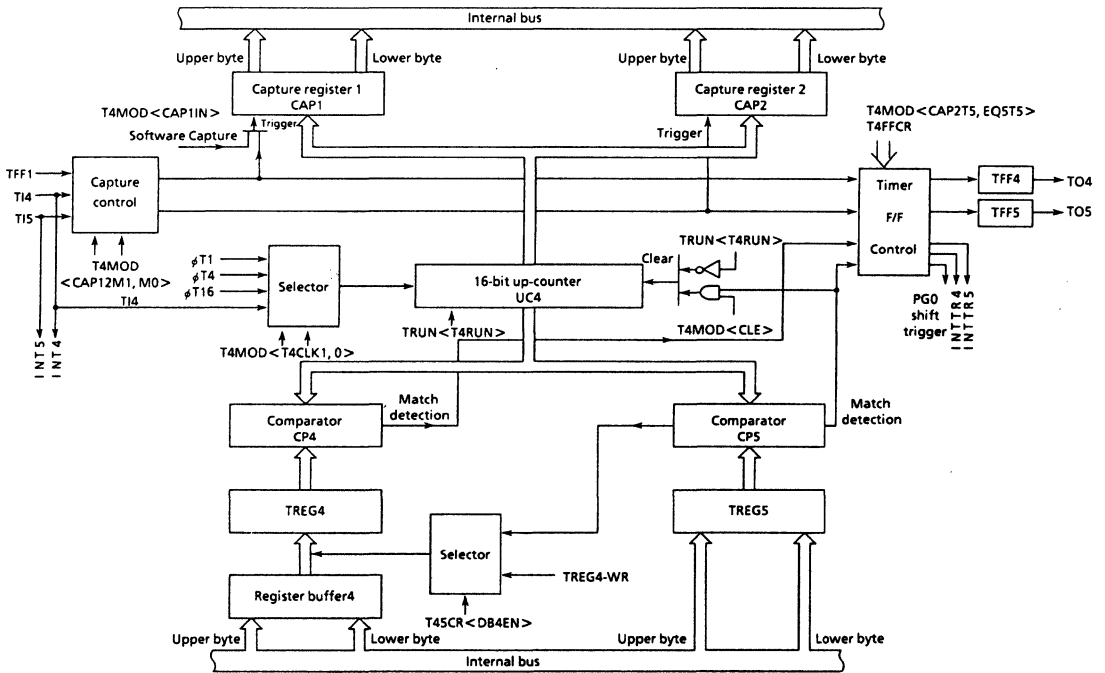


Figure 3.8 (1). Block Diagram of 16-Bit Timer (Timer 4)

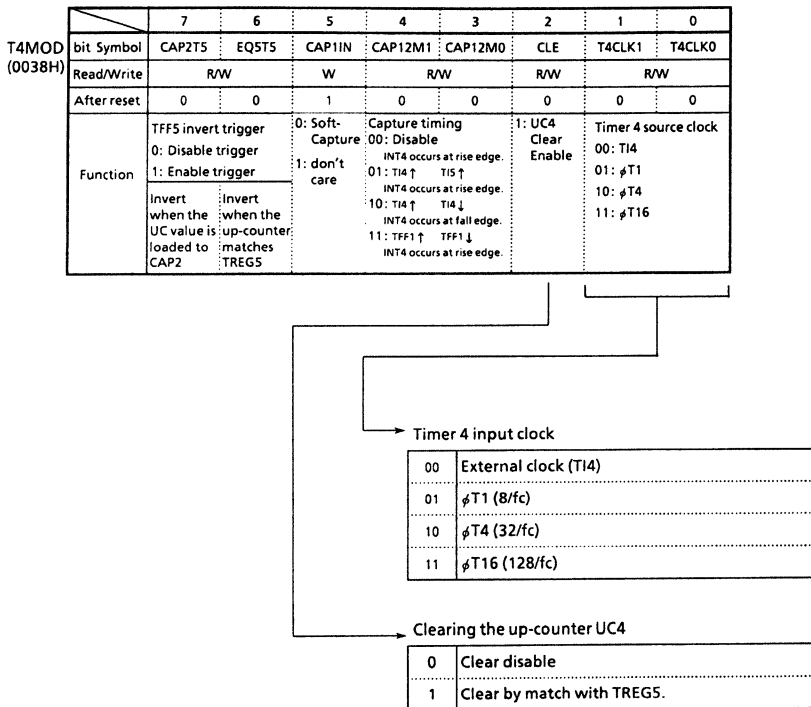


Figure 3.8 (2). 16-Bit Timer Mode Controller Register (T4MOD) (1/2)

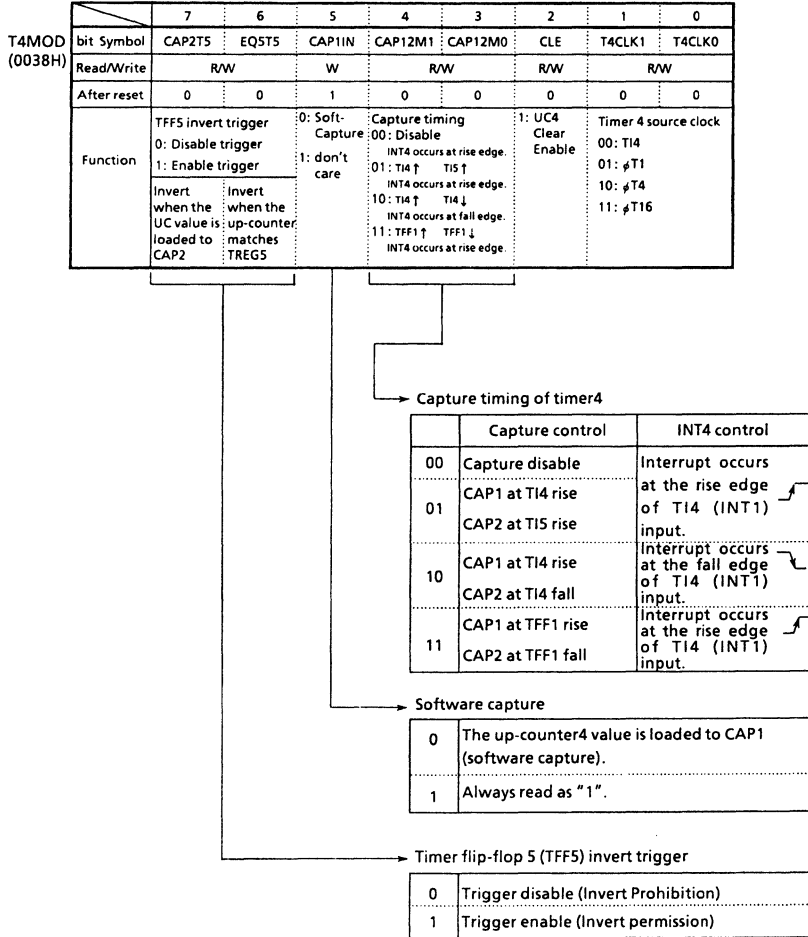


Figure 3.8 (3). 16-Bit Timer Mode Controller Register (T4MOD) (2/2)

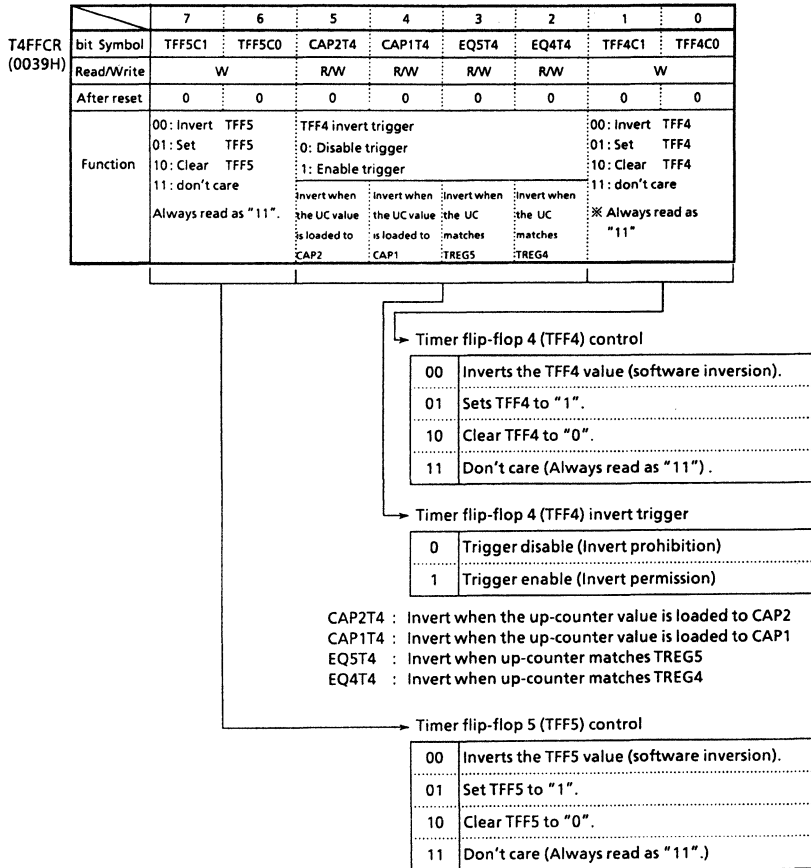


Figure 3.8 (4). 16-Bit Timer 4 F/F Control (T4FFCR)

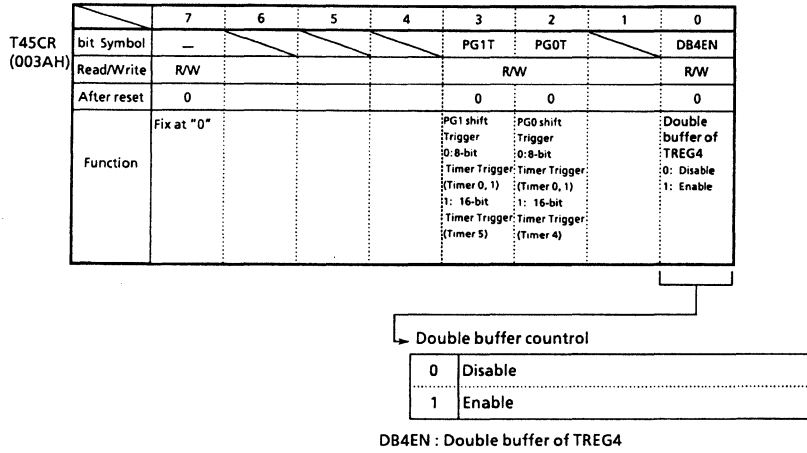


Figure 3.8 (5). 16-Bit Timer (Timer 4) Control Register (T45CR)

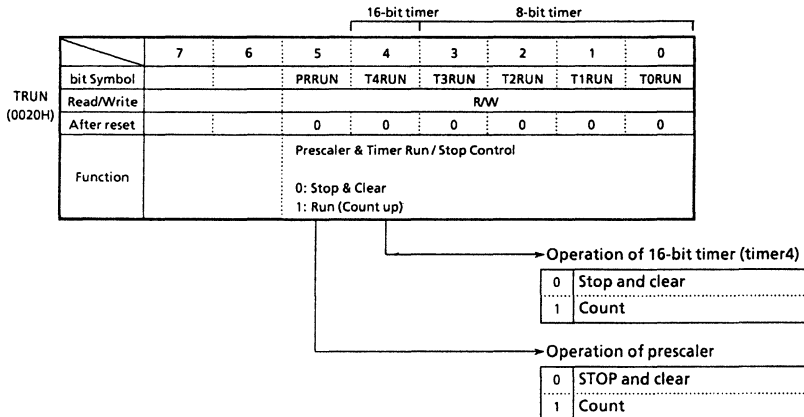


Figure 3.8 (6). Timer Operation Control Register (TRUN)

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  $\phi T1$  (8/fc),  $\phi T4$  (32/fc), and  $\phi T16$  (128/fc) from 9-bit prescaler (also used for 8-bit timer), and external clock from T14 pin (also used as P72/INT4 pin) can be selected. When reset, it will be initialized to <T4CLK1,0> = 00 to select T14 input pin mode. Counting or stop and clear of the counter is controlled by timer operation control register TRUN <T4RUN> .

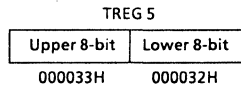
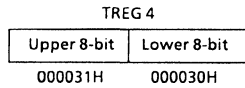
When clearing is enabled, up-counter UC4 will be cleared to zero each time it coincides matches the

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

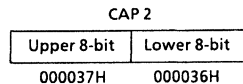
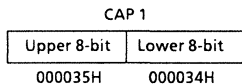
register buffer.

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 T14 pin (also used P80/INT4) input, while data is loaded to CAP2 at the rise edge of T15 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 T14 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

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

Generating interrupts at fixed intervals

		7	6	5	4	3	2	1	0
TRUN	←	x	x	-	0	-	-	-	-
INTET54	←	1	1	0	0	1	0	0	0
T4FFCR	←	1	1	0	0	0	0	1	1
T4MOD	←	0	0	1	0	0	1	*	*
									(** = 01, 10, 11)
TREG5	←	0	0	1	0	1	0	0	0
		*	*	*	*	*	*	*	*
TRUN	←	1	x	1	1	-	-	-	-

- Stop timer 4.
- Enable INTTR5 and sets interrupt level 4. Disable INTTR4.
- Disable trigger.
- Select internal clock for input and disable the capture function.
- Set the interval time (16-bit).

Note : x; don't care    -; no change

Start timer 4.

(2) 16-Bit Event Counter Mode

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.

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the

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	-	-	-	-
P7CR	←	-	-	0	0	-	-	-	-
INTET54	←	1	1	0	0	1	0	0	0
T4FFCR	←	1	1	0	0	0	0	1	1
T4MOD	←	0	0	1	0	0	1	0	0
TREG5	←	*	*	*	*	*	*	*	*
TRUN	←	x	x	1	1	-	-	-	-

- Stop timer 4.
- Set P72 to input mode.
- Enable INTTR5 and sets interrupt level 4, while disable INTTR4.
- Disable trigger.
- Select TI4 as the input clock.
- Set the number of counts (16-bit).
- Start timer 4.

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



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

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.

The PPG mode is obtained by inversion of the timer flip-flop TFF4 that is to be enabled by the match of the

$$(\text{Set value of TREG4}) < (\text{Set value of TREG5})$$

		7	6	5	4	3	2	1	0
TRUN	←	x	x	-	0	-	-	-	-
TREG4	←	*	*	*	*	*	*	*	*
TREG5	←	*	*	*	*	*	*	*	*
T45CR	←	*	*	*	*	*	*	*	*
T4FFCR	←	1	1	0	0	0	0	1	1
T4MOD	←	0	0	1	0	0	1	0	0
							(** = 01, 10, 11)		
P7CR	←	-	-	-	-	-	-	1	1
TRUN	←	x	x	1	1	-	-	-	-

- Stop timer 4.
- Set the duty. (16-bit).
- Set the cycle. (16-bit).
- Double Buffer of TREG4 enable  
(Change the duty and cycle at the interrupt INTTR5).
- Select the mode to invert TFF4 at the match with TREG4/TREG5, and also set the TFF4 to "0".
- Select the internal clock for the input, and disable the capture function.
- Assign P70 as TO4.
- Start timer 4.

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

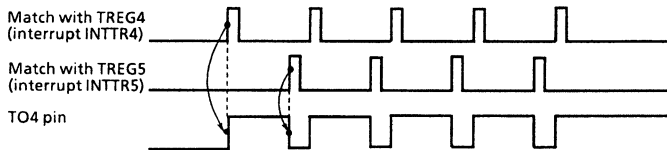


Figure 3.8 (7). Programmable Pulse Generation (PPG) Output Waveforms

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.

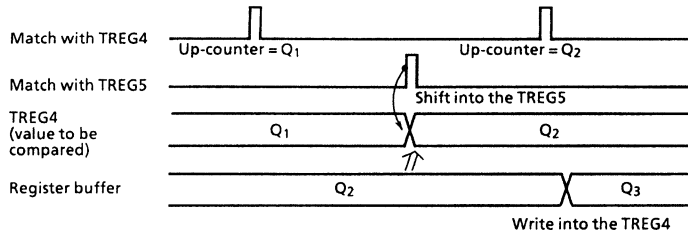


Figure 3.8 (8). Operation of Register Buffer

Shows the block diagram of this mode.

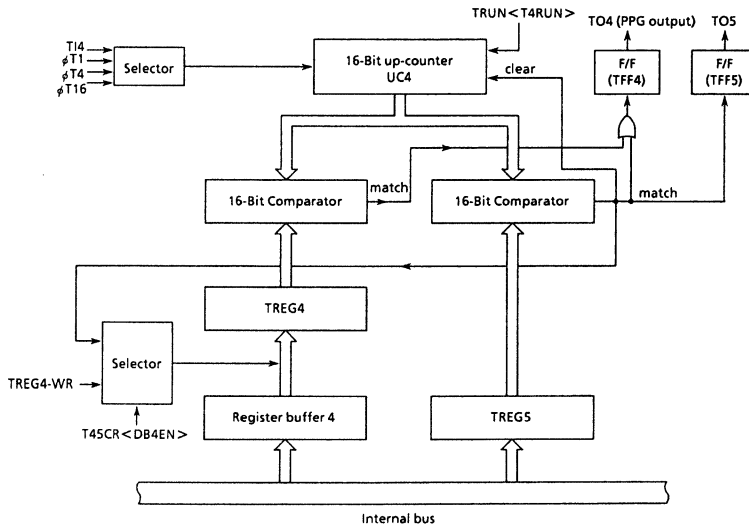


Figure 3.8 (9). Block Diagram of 16-Bit PPG Mode

(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  $T4MOD \langle CAP12M1, 0 \rangle = 01$ .

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  $\langle EQ5T4, EQ4T4 \rangle$  register should be set that the TFF4 inversion is enabled only when the up-counter value matches TREG4 or TREG5. When interrupt INTTR5 occurs, this inversion will be disabled.

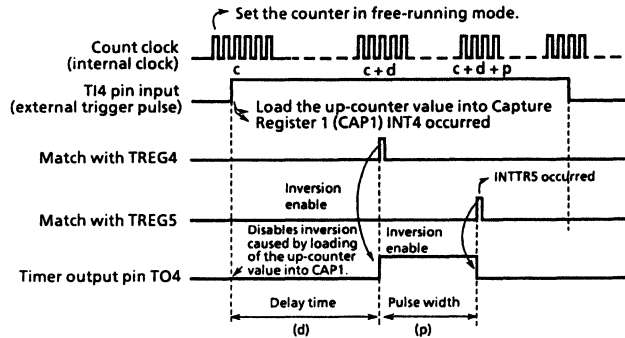
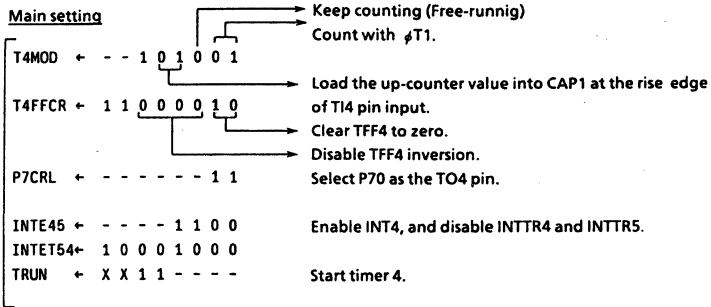
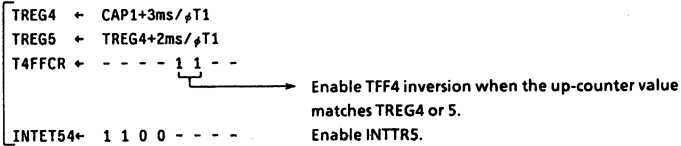


Figure 3.8 (10). One-Shot Pulse Output (with Delay)

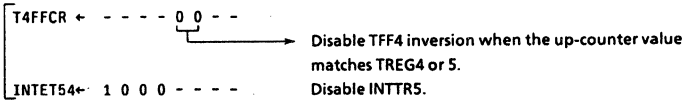
Setting example: To output 2ms one-shot with 3ms delay to the external trigger pulse to TI4 pin.



**Setting of INT4**



**Setting of INT5**



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

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

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

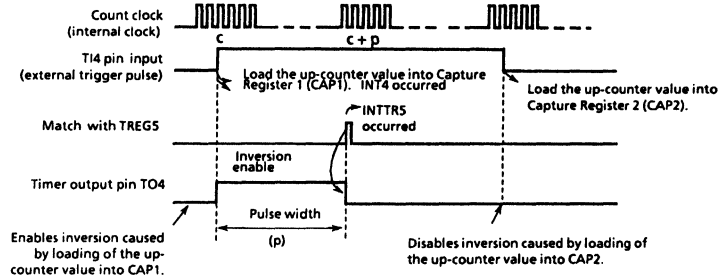


Figure 3.8 (11). One-Shot Pulse Output (without Delay)

② Frequency Measurement

The frequency of the external clock can be measured in this mode. The clock is input through the T14 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 T14 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.

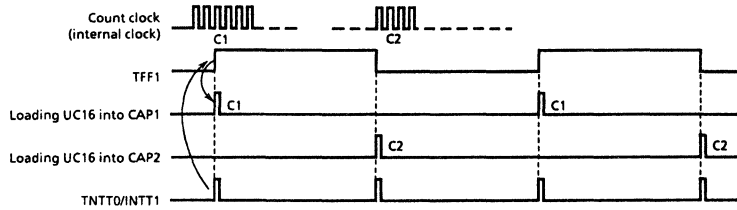


Figure 3.8 (12). Frequency Measurement

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 T14 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 T14.

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

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

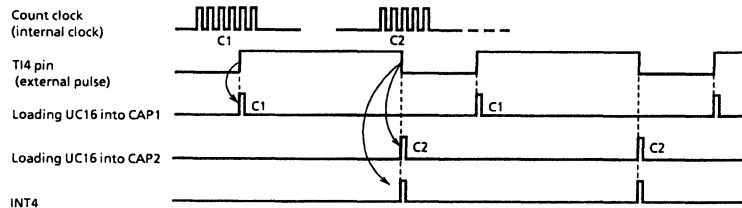


Figure 3.8 (13). Pulse Width Measurement

Note: Only in this pulse width measuring mode (T4MOD <CAP12M1, 0> = 10), external interrupt INT4 occurs at the falling edge of T14 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 T14 and T15.

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 T14. Then the interrupt INT4 is generated.

Similarly, the UC4 value is loaded into CAP2 at the rising edge of the input pulse to T15, 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.

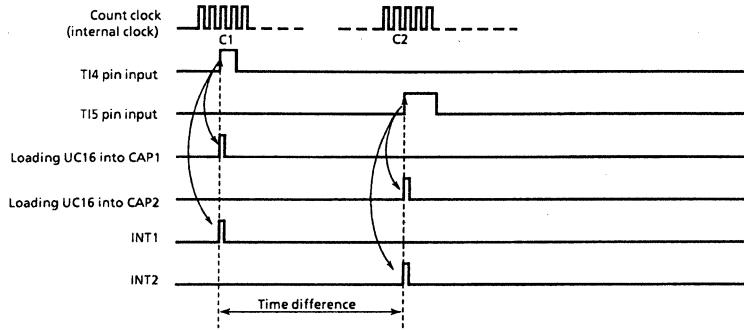


Figure 3.8 (14). Time Difference Measurement

(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.

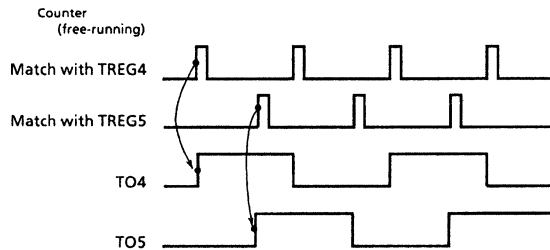


Figure 3.8 (15). Phase Output

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

	16MHz	20MHz
$\phi$ T1	32.768ms	26.214ms
$\phi$ T4	131.072ms	104.856ms
$\phi$ T16	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

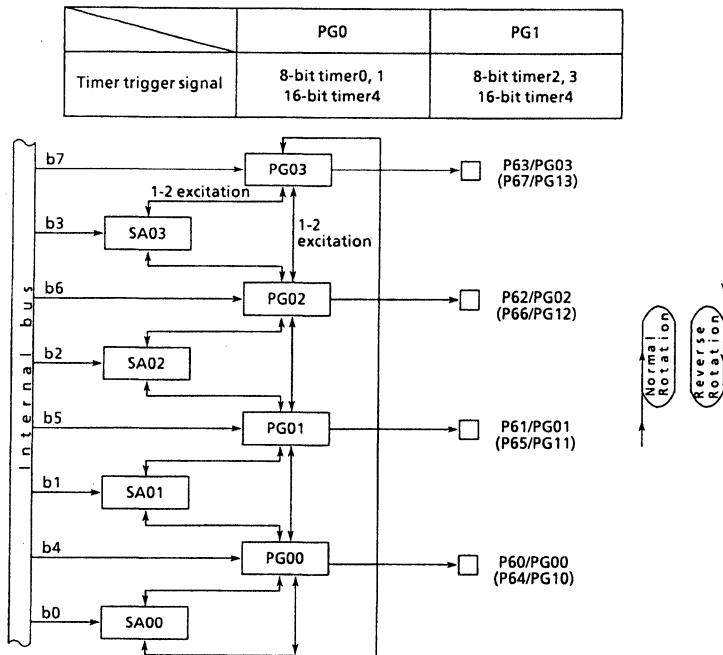


Figure 3.9 (1). Pattern Generator/Stepping Motor Control Block Diagram

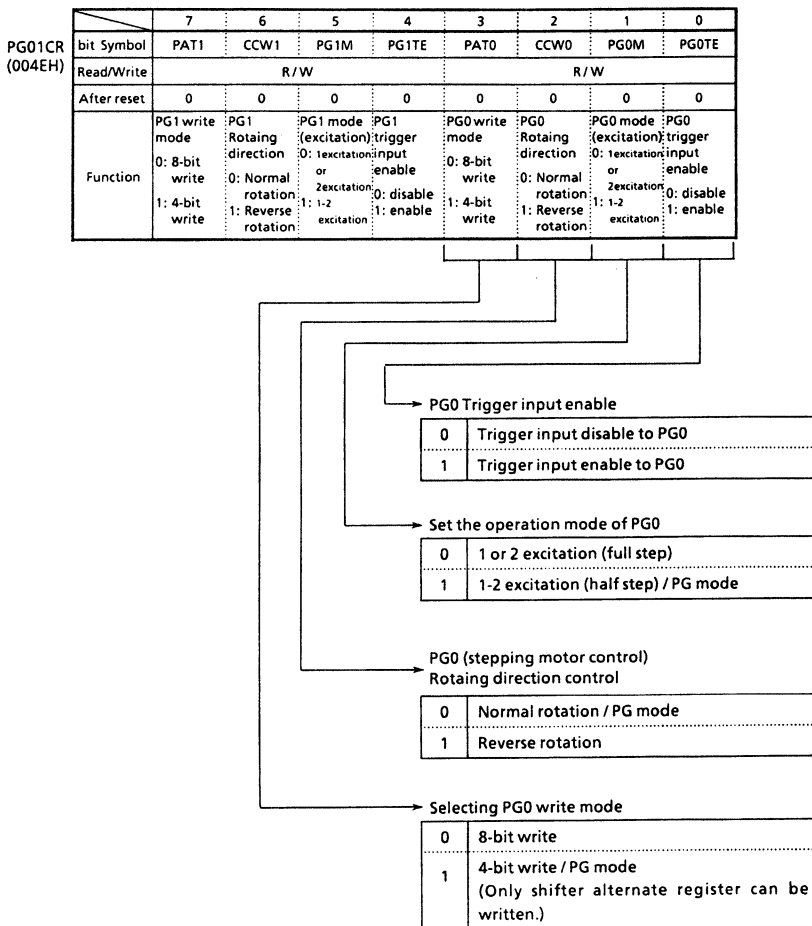


Figure 3.9 (2a). Pattern Generation Control Register (PG01CR)

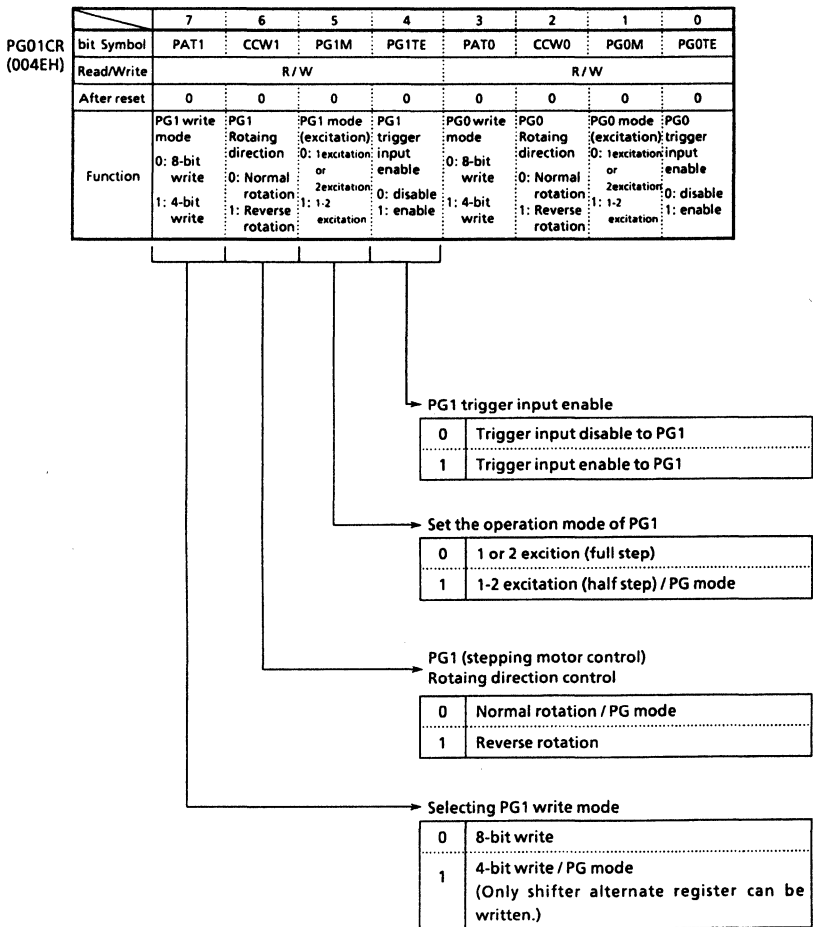


Figure 3.9 (2b). Pattern Generation Control Register (PG01CR)

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
PG0REG (004CH)	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)**

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
PG1REG (004DH)	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)**

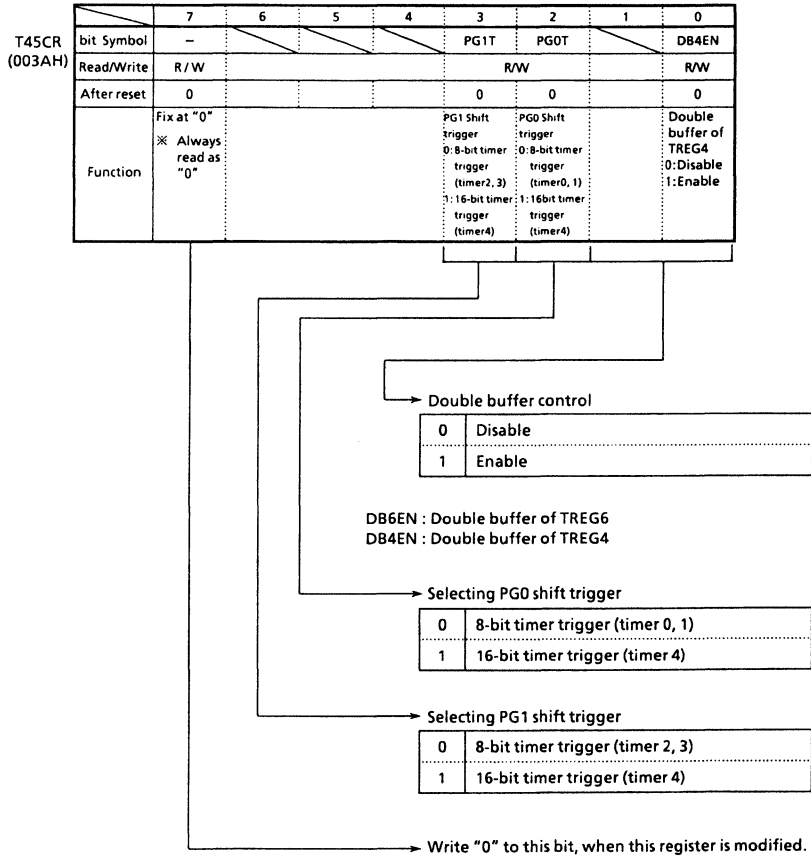


Figure 3.9 (5). 16-bit Timer Trigger Control Register (T45CR)

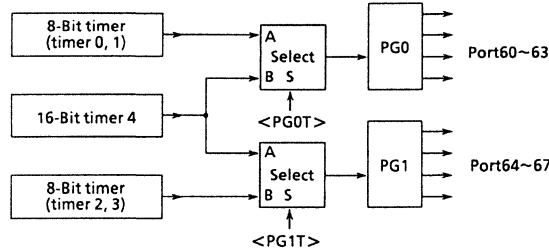


Figure 3.9 (6). Connection of Timer and Pattern Generator

(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.

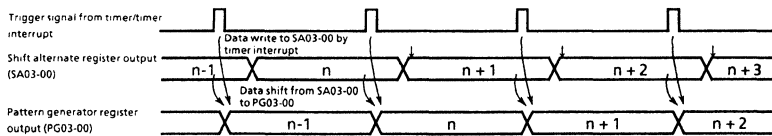


Figure 3.9 (7). Pattern Generation Mode Timing Example

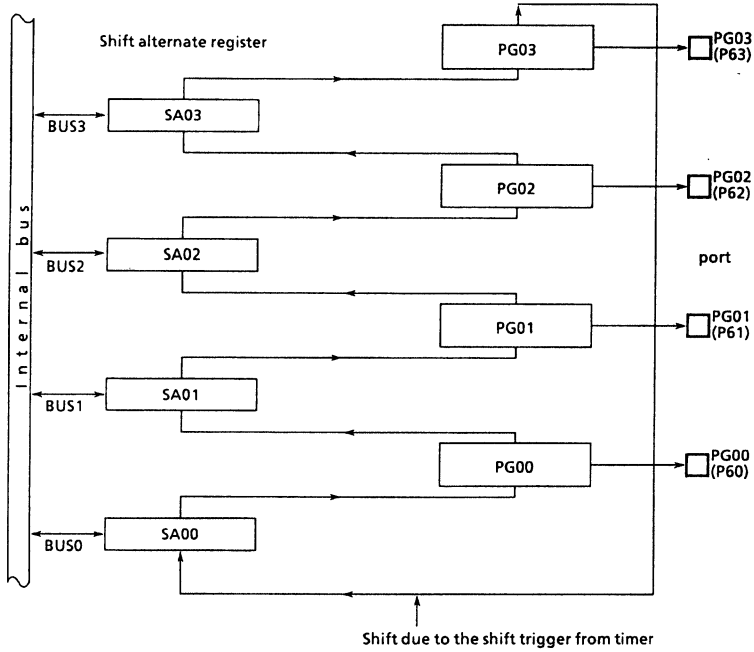


Figure 3.9 (7). Pattern Generation Mode Block Diagram (PG0)

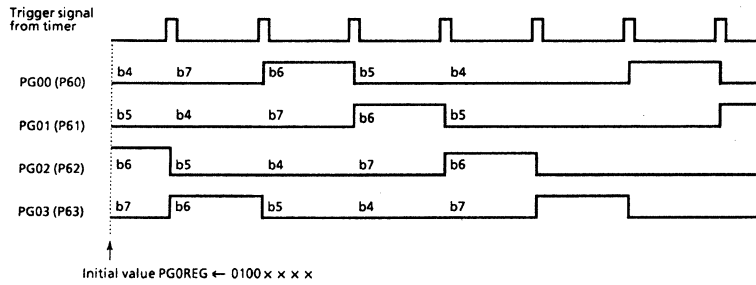
In this pattern generation mode, only writing the output latch is disabled by hardware, but other functions do the same operation as 1-2 excitation in stepping motor control port

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

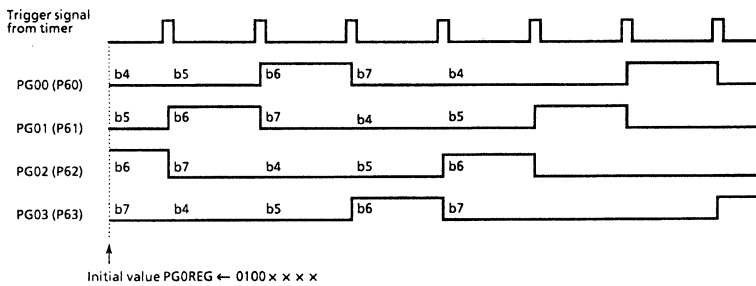
① 4-phase 1-Step/2-Step Excitation

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.



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

① Normal Rotation



② Reverse Rotation

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



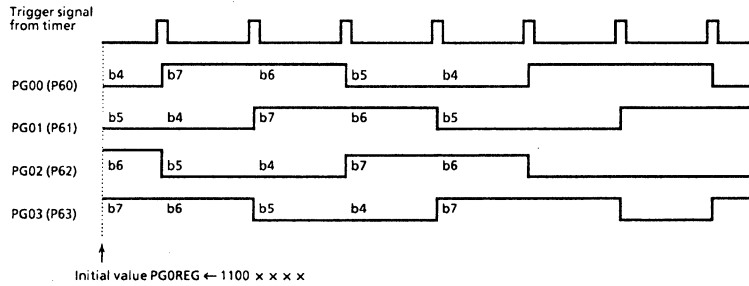


Figure 3.9 (9). Output Waveforms of 4-Phase 2-Step Excitation (Normal Rotation)

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.

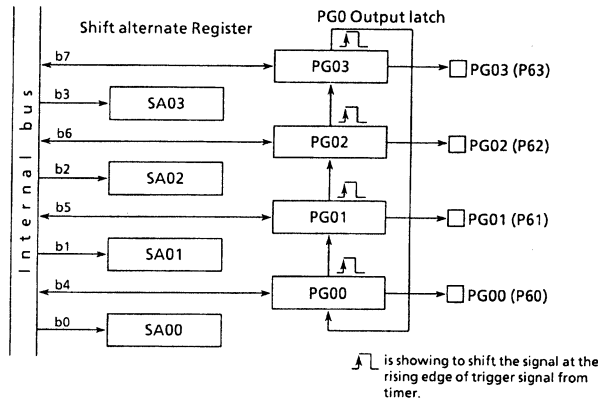
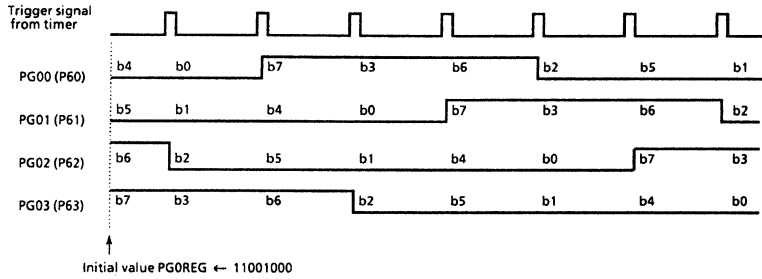


Figure 3.9 (10). Block Diagram of 4-Phase 1-Step Excitation/2-Step Excitation (Normal Rotation)

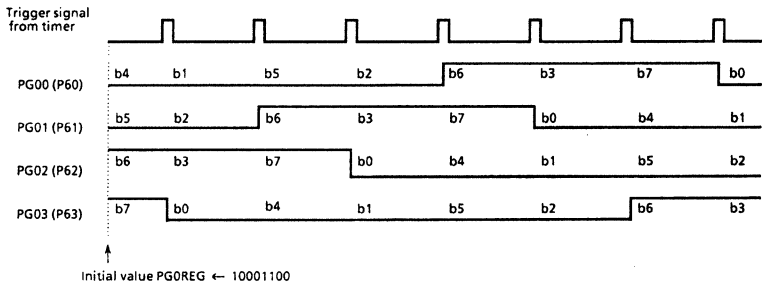
② 4-Phase 1-2 Step Excitation

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.

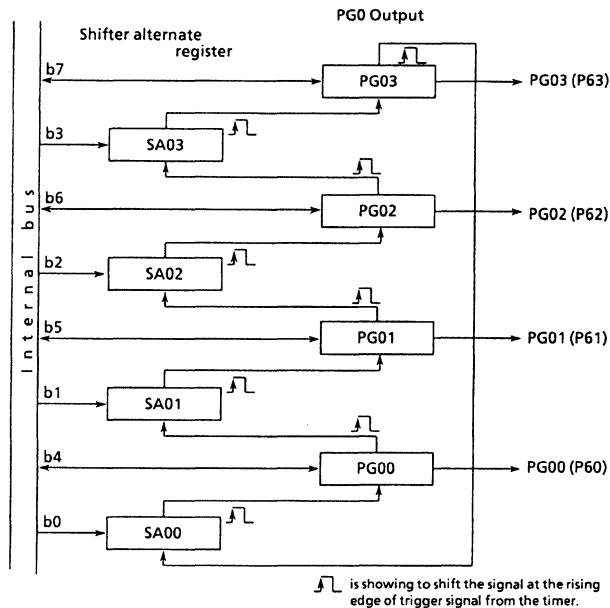


Figure 3.9 (12). Block Diagram of 4-Phase 1-2 Step Excitation (Normal Rotation)

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
TMOD	←	0	0	x	x	-	-	0	1
TFFCR	←	x	x	x	0	1	0	1	0
TREG0	←	*	*	*	*	*	*	*	*
P6CRL	←	1	0	1	0	1	0	1	0
PG01CR	←	-	-	-	-	0	0	1	1
PGOREG	←	1	1	0	0	1	0	0	0
TRUN	←	1	x	-	-	-	-	-	1

Note: x; don't care    -; no change

- Stop timer 0, and clears it to zero.
- Set 8-bit timer mode and selects  $\phi T1$  as the input clock of timer 0.
- Clear TFF1 to zero and enables the inversion trigger by timer 0.
- Set the cycle in timer register.
- Set P60 ~ P63 bits to PG output.
- Select PG0 4-phase 1 - 2 step excitation mode and normal rotation.
- Set an initial value.
- Start timer 0.

(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).

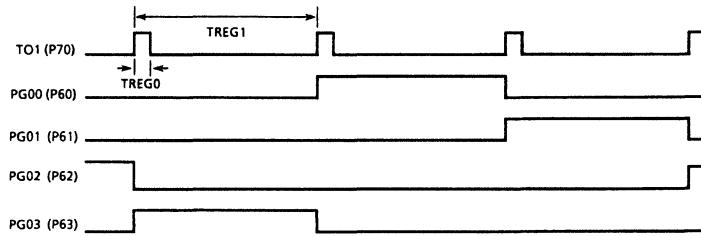


Figure 3.9 (13). Output Waveforms of 4-Phase 1-Step Excitation

Setting example:

	7	6	5	4	3	2	1	0
TRUN	← -	x	-	-	-	-	0	0
TMOD	← 1	0	x	x	x	x	0	1
TFFCR	← x	x	x	0	0	1	1	x
TREG0	← *	*	*	*	*	*	*	*
TREG1	← *	*	*	*	*	*	*	*
P7CR	← -	-	-	-	-	-	1	0
P6CRL	← 1	0	1	0	1	0	1	1
PG01CR	← -	-	-	-	0	0	0	1
PGOREG	← *	*	*	*	*	*	*	*
TRUN	← 1	x	-	-	-	-	1	1

- Stop timer 0, and clears it to zero.
- Set timer 0 and timer 1 in PPG output mode and selects  $\phi T1$  as the input clock.
- Enable TFF1 inversion and sets TFF1 to "1".
- Set the duty of TO1 to TREG0.
- Set the cycle of TO1 to TREG1.
- Assign P70 as TO1.
- Assign P60 - 63 as PG0.
- Set PG0 in 4-phase 1-step excitation mode.
- Set an initial value.
- Start timer 0 and timer 1.

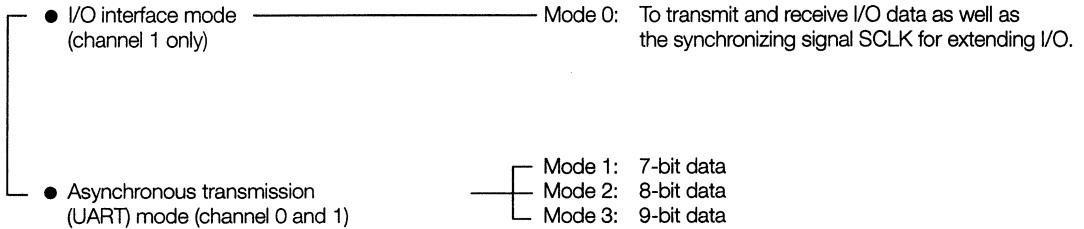
Note: x: don't care    -: no change

**3.10 Serial Channel**

The TMP96C031F contains 2 serial I/O channels for full duplex

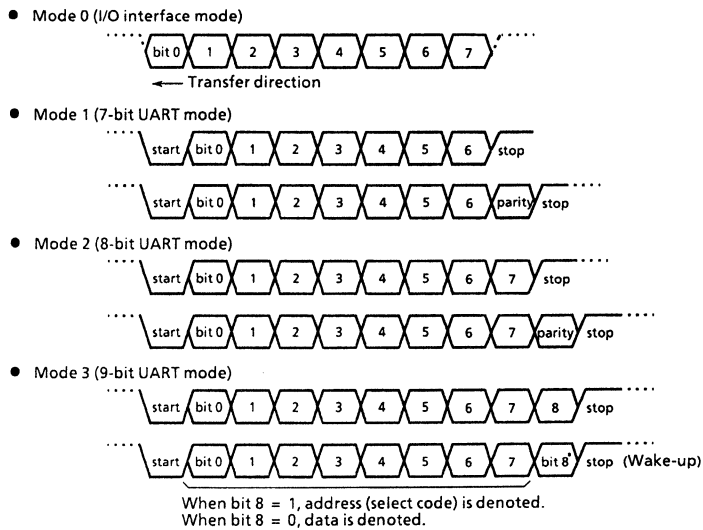
asynchronous transmission (UART) as well as for I/O extension.

The serial channel has the following operation modes.



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.

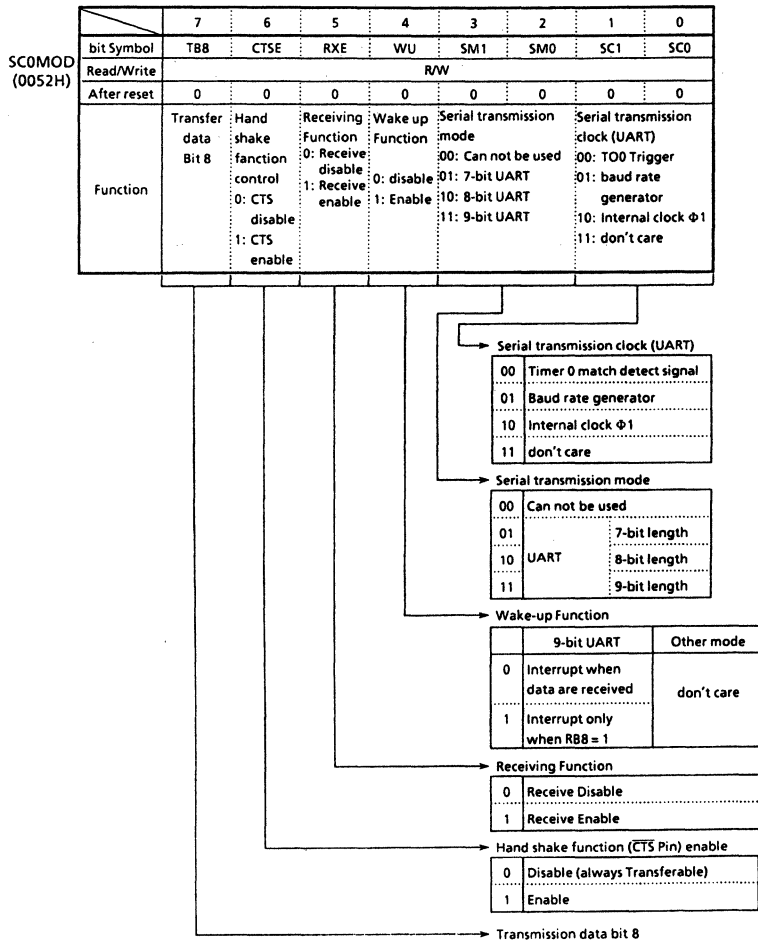
When the transmission buffer becomes empty and requests the CPU to send the next transmission data, or when data is stored in the receiving data register and the CPU is requested to read the data, INTTX or INTRX interrupt occurs. Besides, if an overrun error, parity error, or framing error occurs during 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 ( $\phi T0$ ,  $\phi T2$ ,  $\phi T8$ , and  $\phi T32$ ) 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

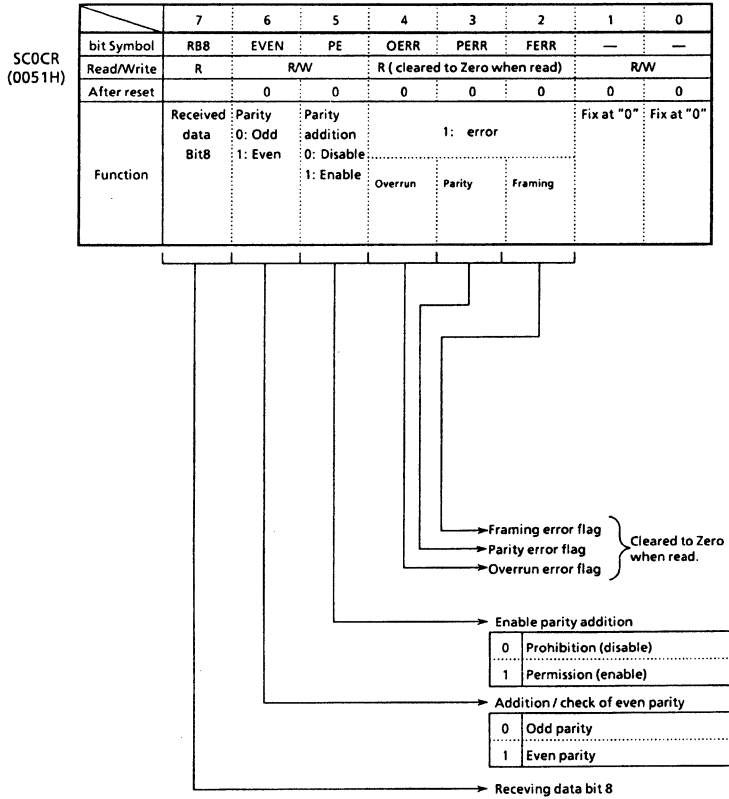
The serial channel is controlled by three control registers SC0CR, SC0MOD, and BR0CR. Transmitted and received data is stored in register SC0BUF.



Note : There is SC1MOD (56H) in Channel1

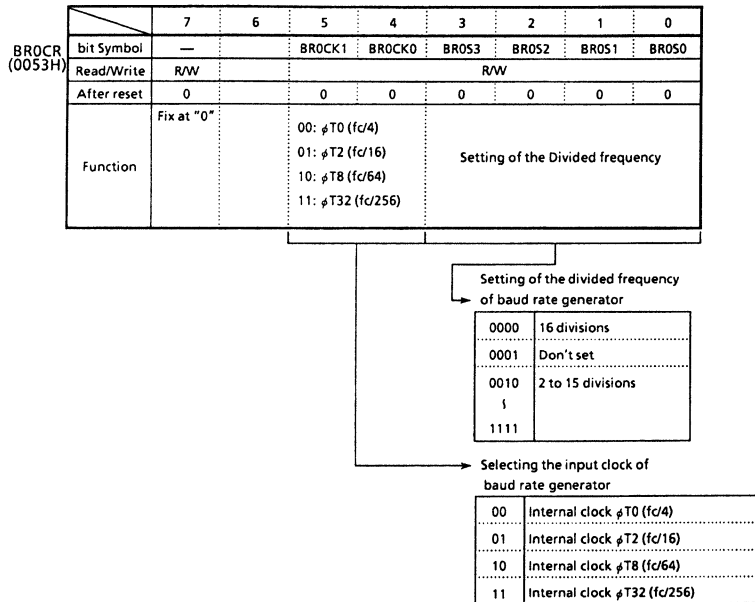
Figure 3.10 (2). Serial Mode Control Register (Channel 0, SC0MOD)





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

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



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)

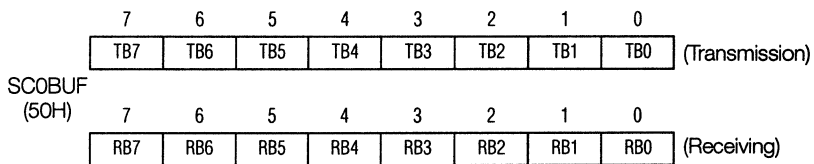


Figure 3.10 (5). Serial Transmission/Receiving Buffer Registers (Channel 0, SC0BUF)

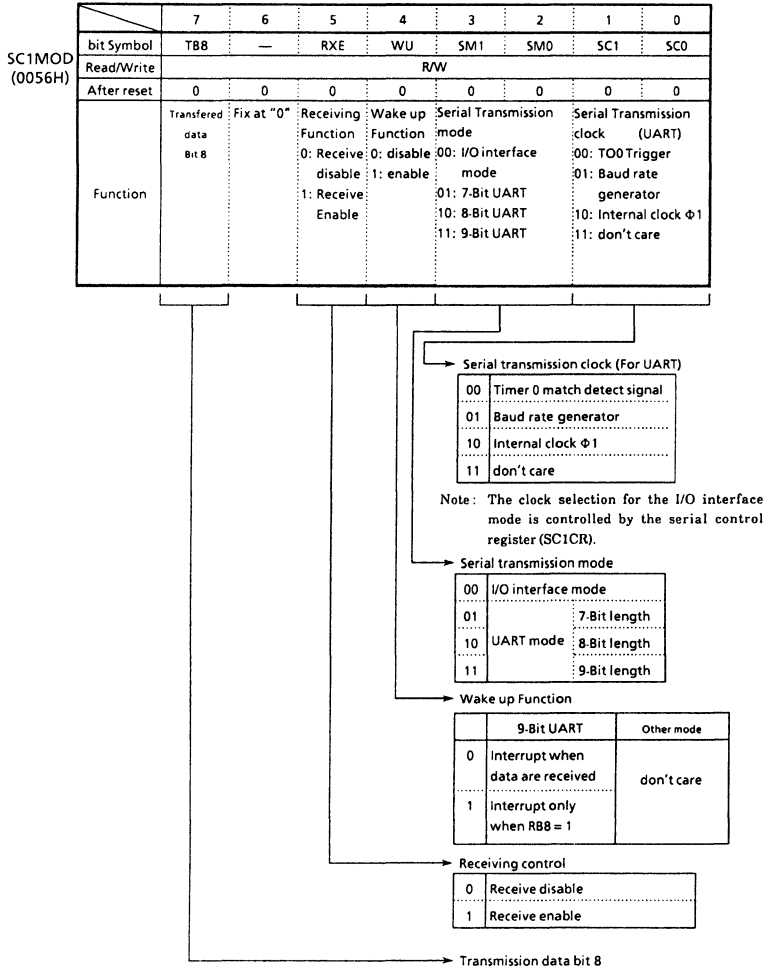
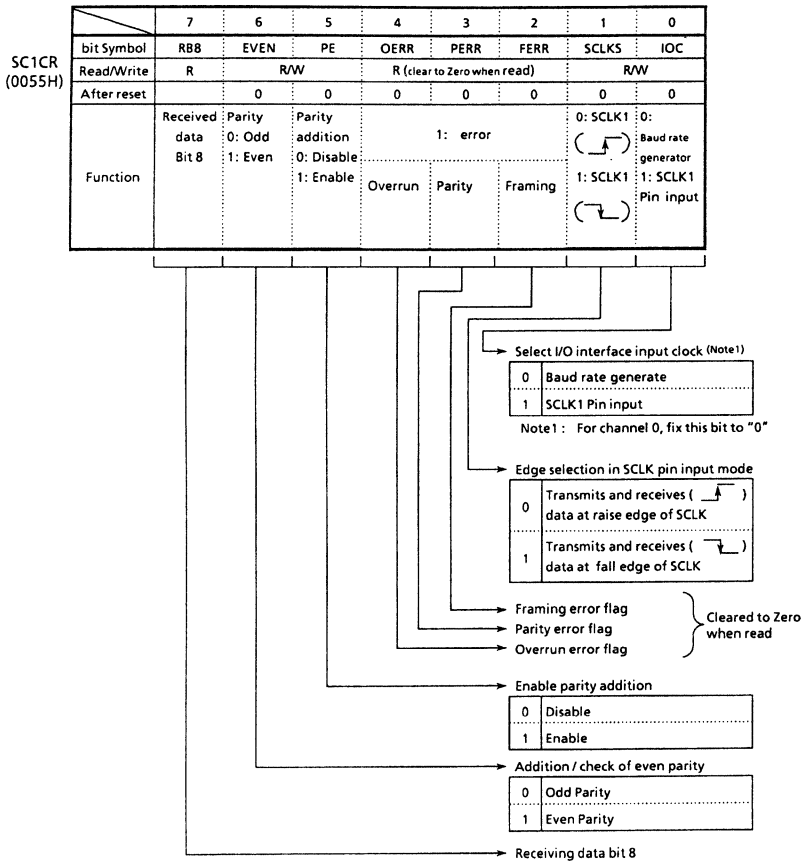
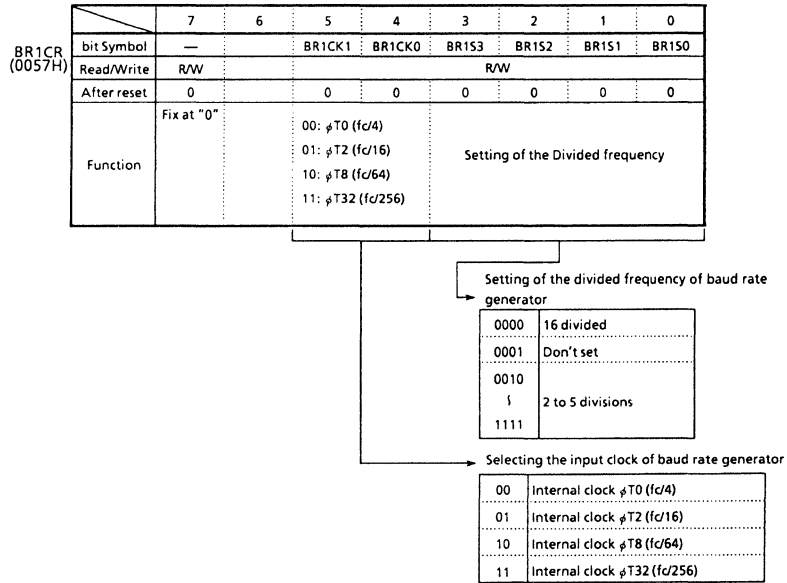


Figure 3.10 (6). Serial Mode Control Register (Channel 1, SC1MOD)



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

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



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

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

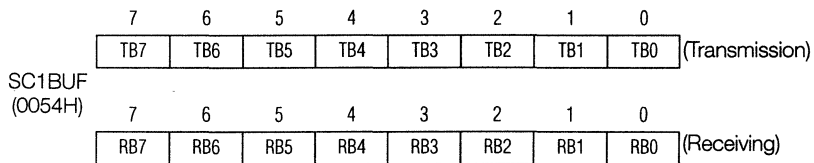


Figure 3.10 (9). Serial Transmission/Receiving Buffer Registers (Channel 1, SC1BUF)

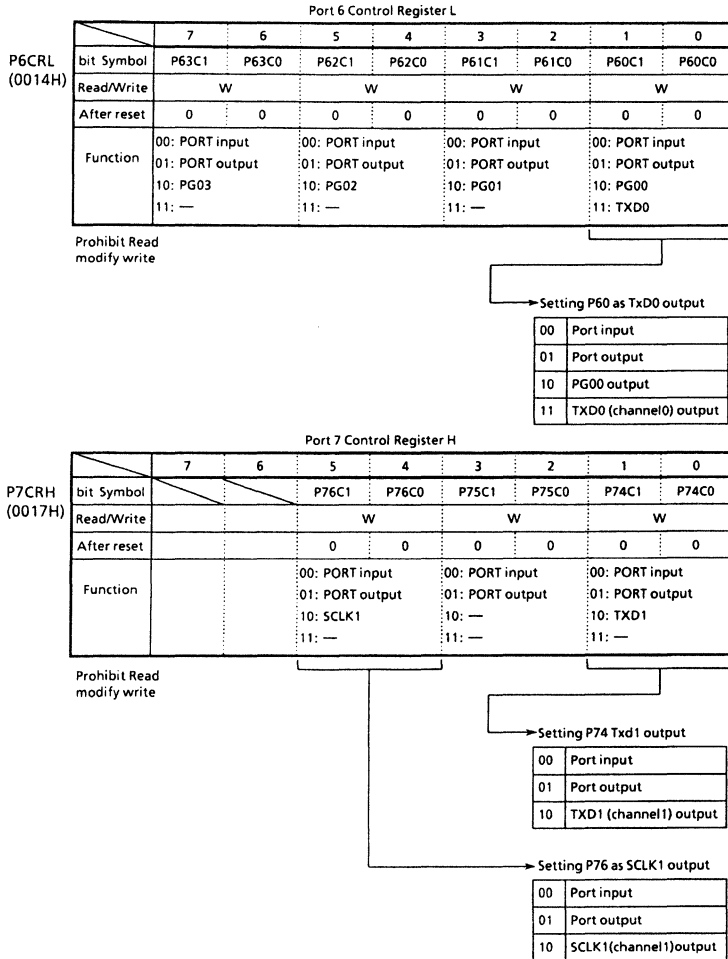
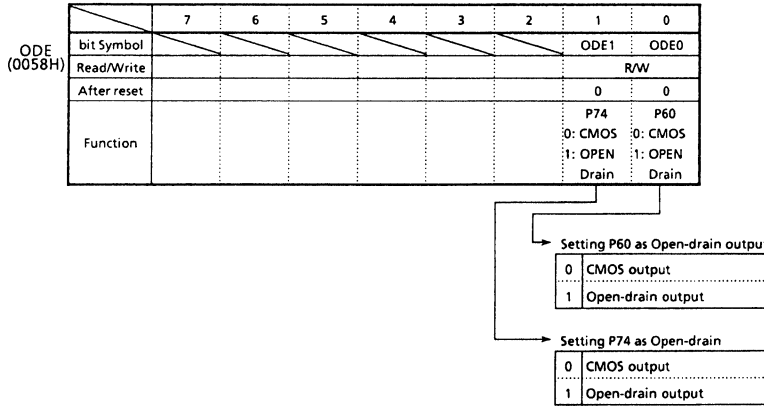


Figure 3.10 (10). Port 6, 7 Control Registers



**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.

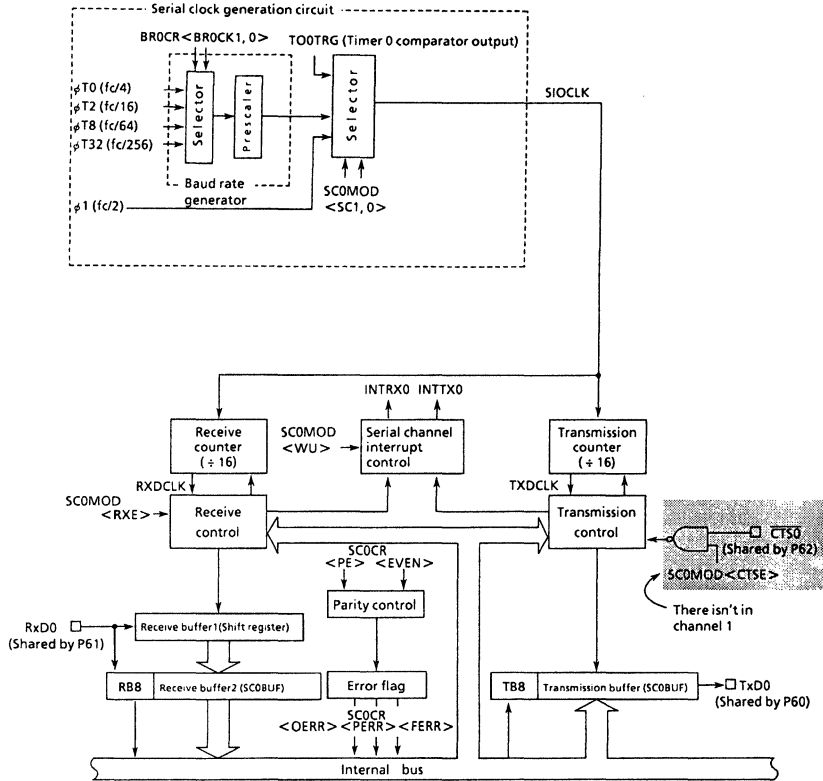


Figure 3.10 (12). Block Diagram of the Serial Channel 0



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

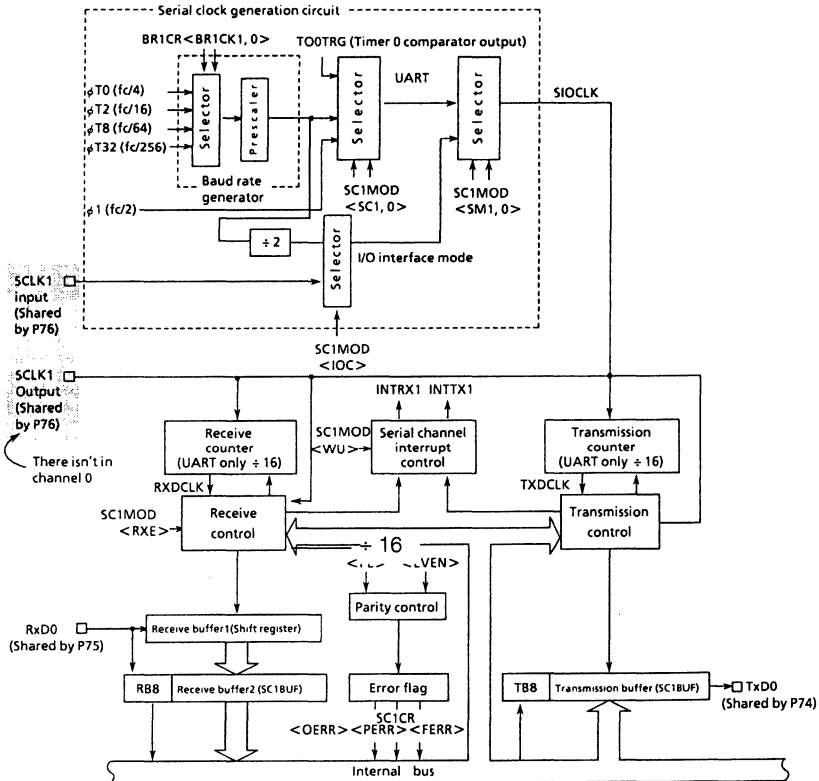


Figure 3.10 (13). Block Diagram of the Serial Channel 1

## 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,  $\phi T0$  ( $fc/4$ ),  $\phi T2$  ( $fc/16$ ),  $\phi T8$  ( $fc/64$ ), or  $\phi T32$  ( $fc/256$ ) is generated by the 9-bit prescaler which is shared by the timers.

One of these input clocks is selected by the baud rate generator control register BROCR/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:

$$\phi T0 = fc/4$$

$$\phi T2 = fc/16$$

$$\phi T8 = fc/64$$

$$\phi T32 = fc/256$$

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

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

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	$\phi T0$ (fc/4)	$\phi T2$ (fc/16)	$\phi T8$ (fc/64)	$\phi T32$ (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	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  $\phi T1$ ) is Used)

Unit (Kbps)

TREGO \ $f_c$	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):

$$\text{Transfer rate} = \frac{f_c}{\text{TREGO} \times 8 \times 16}$$

↑ (When timer 0 (input clock  $\phi T1$ ) is used)

Input clock of timer 0

$$\phi T1 = f_c / 8$$

$$\phi T4 = f_c / 32$$

$$\phi T16 = f_c / 128$$

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  $\phi 1$  (500Kbps @  $f_c = 16$  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.

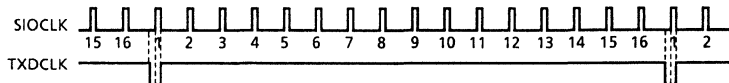


Figure 3.10 (14). Generation of Transmission Clock

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

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

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

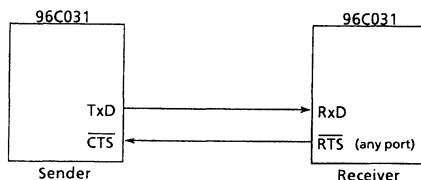
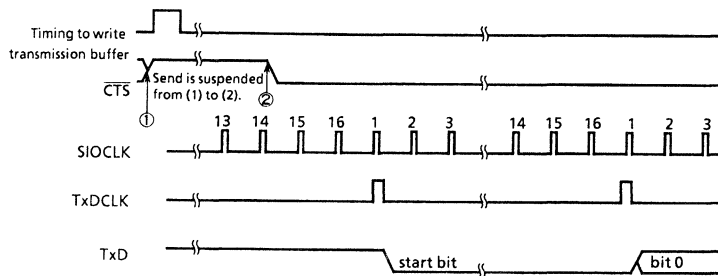


Figure 3.10 (15). Handshake Function



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

Figure 3.10 (16). Timing of  $\overline{\text{CTS}}$  (Clear to Send)

Ⓒ 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 TxDST which is generated by the transmission control. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX0/INTTX1 interrupt.

Ⓒ 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 com-

pared 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

Ⓒ 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.

① 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.

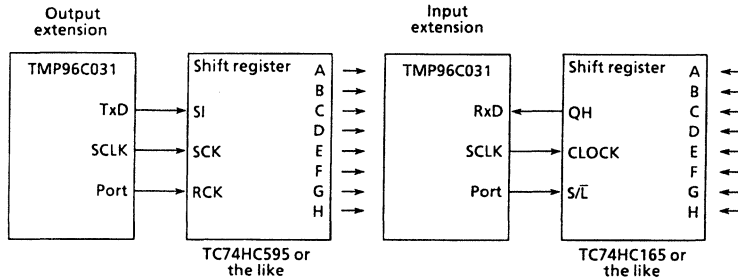


Figure 3.10 (17). Example of SCLK Output Mode Connection

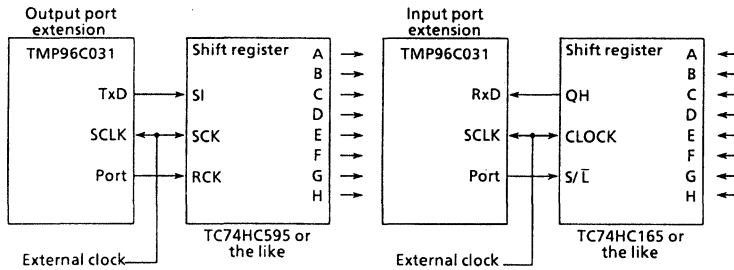


Figure 3.10 (18). Example of SCLK Input Mode Connection



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.

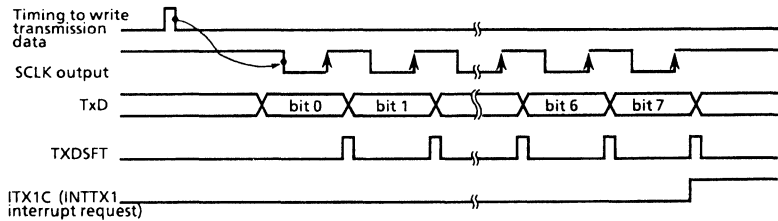


Figure 3.10 (19). Transmitting Operation in I/O Interface Mode (SCLK Output Mode)

In SCLK input 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 <ITX1C> will be set to generate INTTX1 interrupt.

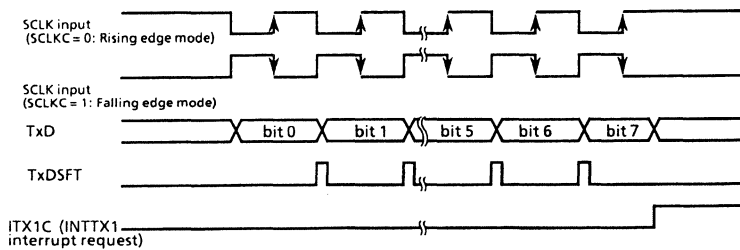


Figure 3.10 (20). Transmitting Operation in I/O Interface Mode (SCLK Input Mode)

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.

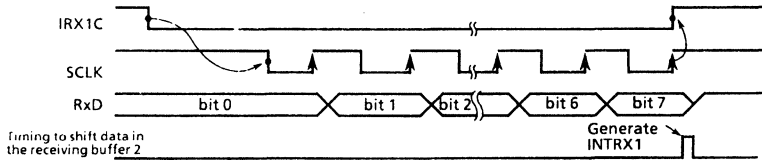


Figure 3.10 (21). Receiving Operation in I/O Interface Mode (SCLK Output Mode)

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 INTRX1 interrupt.

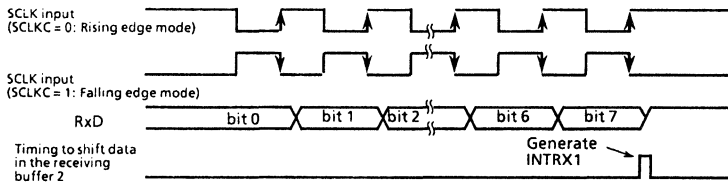


Figure 3.10 (22). Receiving Operation in I/O Interface Mode (SCLK Input Mode)

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 SCOMOD <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
SCOMOD	← x	0	-	x	0	1	0	1
SC0CR	← x	1	1	x	x	x	0	0
BROCR	← 0	x	1	0	0	1	0	1
TRUN	← x	x	1	-	-	-	-	-
INTES0	← 1	1	0	0	-	-	-	-
SC0BUF	← *	*	*	*	*	*	*	*

- Select P60 as the TxD pin.
- Set 7-bit UART mode.
- Add an even parity.
- Set transfer rate at 2400 bps.
- Start the prescaler for the baud rate generator.
- Enable INTTX0 interrupt and sets interrupt level 4.
- Set data for transmission.

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

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

The 8-bit UART mode can be specified by setting SCOMOD <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	-	-
SCOMOD	← -	0	1	x	1	0	0	1
SCOCR	← x	0	1	x	x	x	0	0
BROCR	← 0	x	0	1	0	1	0	1
TRUN	← x	x	1	-	-	-	-	-
INTESO	← -	-	-	-	1	1	0	0

- Select P61 (RxD) as the input pin.
- Enable receiving in 8-bit UART mode.
- Add an odd parity.
- Set transfer rate at 9600 bps.
- Start the prescaler for the baud rate generator.
- Enable INTTX0 interrupt and sets interrupt level 4.

Interrupt processing

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

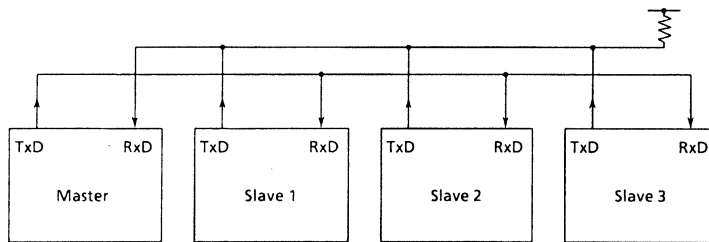
Note: x; don't care -; no change

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

9-bit UART mode can be specified by setting SC0MOD <SM1, 0> /SC1MOD <SM1, 0> to "11". In this mode, parity bit cannot be added. For transmission, the MSB (9th bit) is written in SCMOD <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.

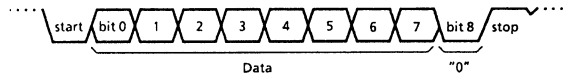
Figure 3.10 (23). Serial Link Using Wake-Up Function

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".

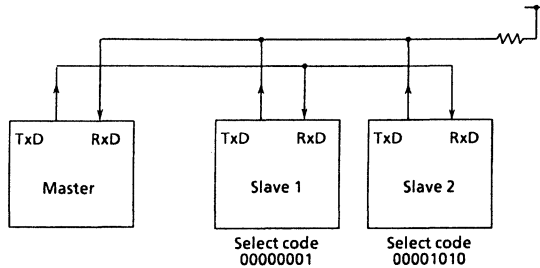


- ④ Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.
- ⑤ The master controller transmits data to the specified slave controller whose SC0MOD <WU> / SC1MOD <WU> bit is cleared to "0." The MSB (bit 8) <TB8> is cleared to "0".



- 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  $\phi 1$  ( $f_c/2$ ) as the transfer clock.



Since serial channels 0 and 1 operate in exactly the same

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

• Setting the master controller

Main 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 $\phi 1$ ( $f_c/2$ ) 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. Enable INTRX0 and INTTX0. Set <WU> to "1" in the 9-bit UART transmission mode with transfer clock $\phi 1$ ( $f_c/2$ ).
ODE	←	x	x	x	x	x	x	-	1	
INTES0	←	1	1	0	1	1	1	1	0	
SC0MOD	←	0	0	1	1	1	1	1	0	

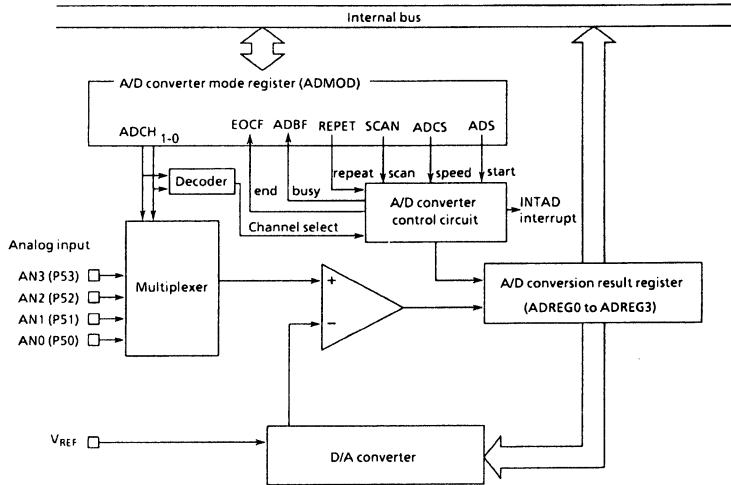
INTRX0 interrupt

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

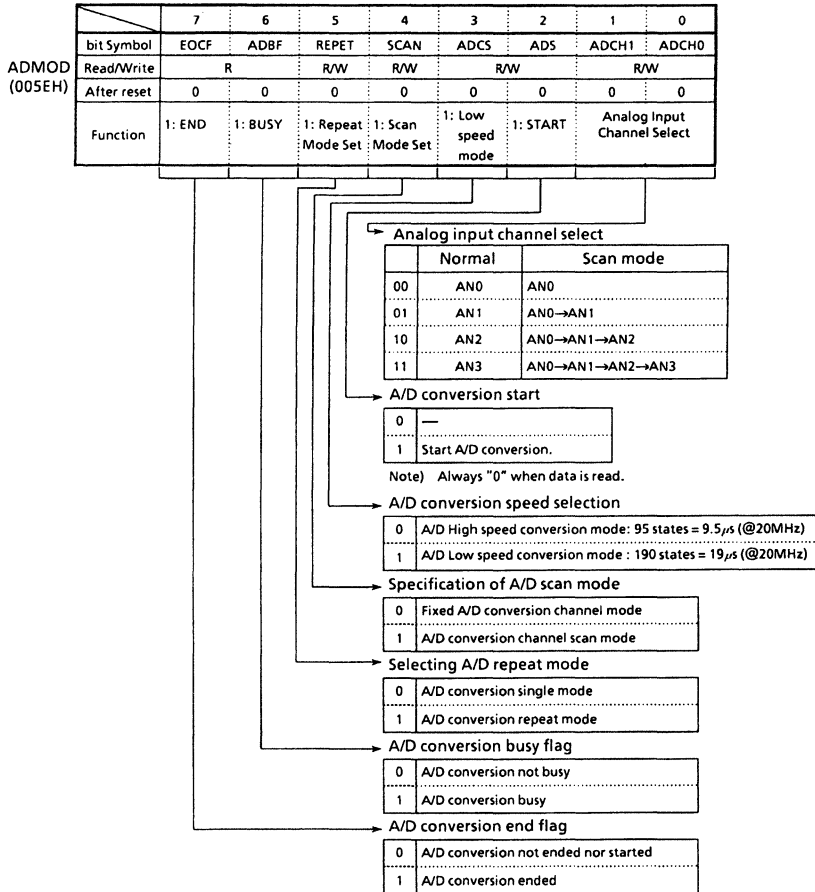


Figure 3.11 (2). A/D Converter Mode Register (ADMOD)



	7	6	5	4	3	2	1	0
ADREG0 (0060H)	bit Symbol							
	Read/Write							
	After reset							
	Function							
	Channel 0 of A/D conversion result register is stored.							
	Register for saving an A/D switch value for AN0							
	7	6	5	4	3	2	1	0
ADREG1 (0061H)	bit Symbol							
	Read/Write							
	After reset							
	Function							
	Channel 1 of A/D conversion result register is stored.							
	Register for saving an A/D switch value for AN1							
	7	6	5	4	3	2	1	0
ADREG2 (0062H)	bit Symbol							
	Read/Write							
	After reset							
	Function							
	Channel 2 of A/D conversion result register is stored.							
	Register for saving an A/D switch value for AN2							
	7	6	5	4	3	2	1	0
ADREG3 (0063H)	bit Symbol							
	Read/Write							
	After reset							
	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
ADMOD	←	x	x	0	0	0	1	1	1

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

INTAD routine

A	←	ADREG3
(FF10H)	←	A

The value of ADREG3 is read into the accumulator. Then the accumulator value is stored into memory at FF10H.

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
ADMOD	←	x	x	1	1	0	1	1	0

Disable INTAD.  
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).

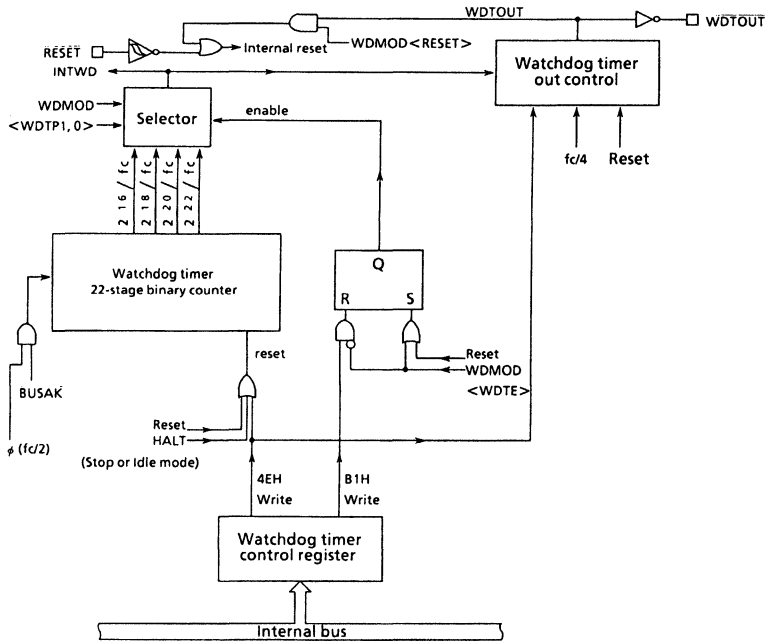


Figure 3.12 (1). Block Diagram of Watchdog Timer

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

Since the watchdog timer out pin ( $\overline{\text{WDTOUT}}$ ) 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 2.0 $\mu$ s @ 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).

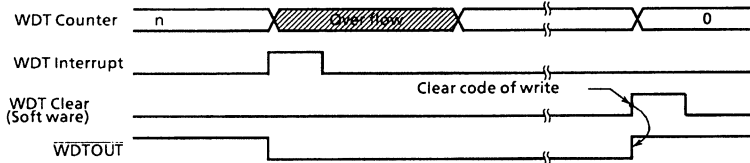


Figure 3.12 (2). Normal Mode

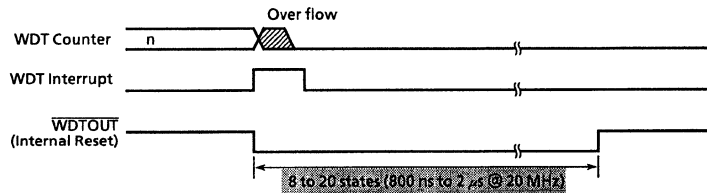


Figure 3.12 (3). Reset Mode

**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  $2^{16}/f_c$  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.

- 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".

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.

- 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	1	0	Write the clear code (4EH).
------	---	---	---	---	---	---	---	---	---	---	-----------------------------

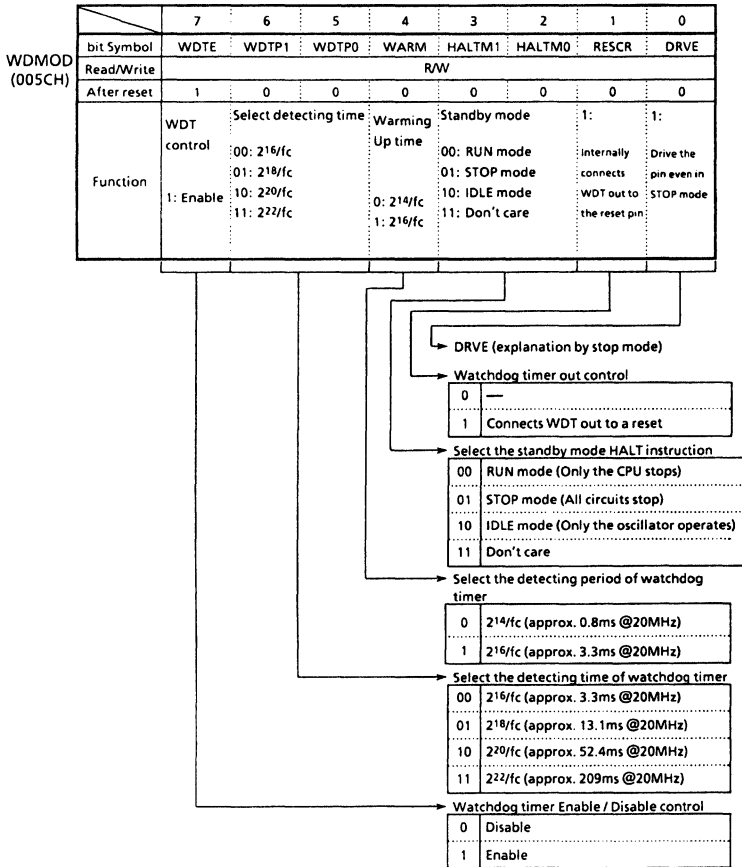


Figure 3.12 (4). Watchdog Timer Mode Register

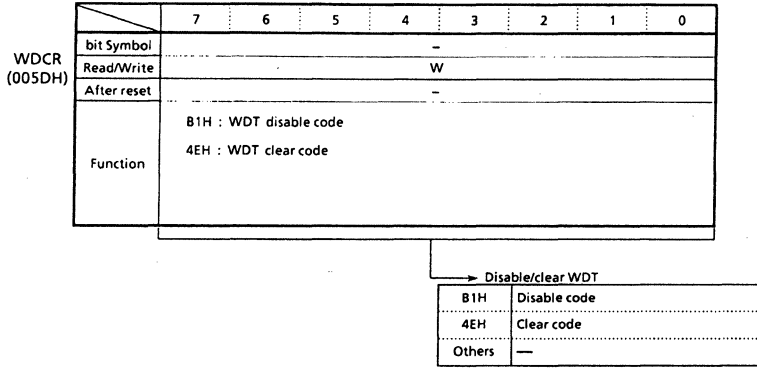


Figure 3.12 (5). Watchdog Timer Control Register

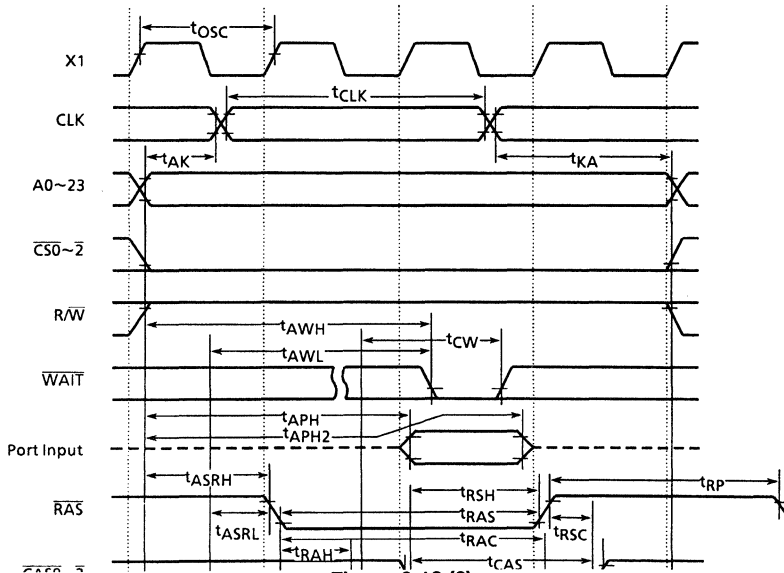


Figure 3.12 (6)



**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 oper-

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

The watchdog timer restarts operation immediately after resetting is released.

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

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

Example:

- ① Clear the binary counter

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

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

WDMOD ← 1 0 1 - - - x x

- ③ Disable the watchdog timer

WDMOD ← 0 - - - - - x x      Clear WDTE to "0".

WDCR ← 1 0 1 1 0 0 0 1      Write disable code (B1H).

- ④ Set 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.

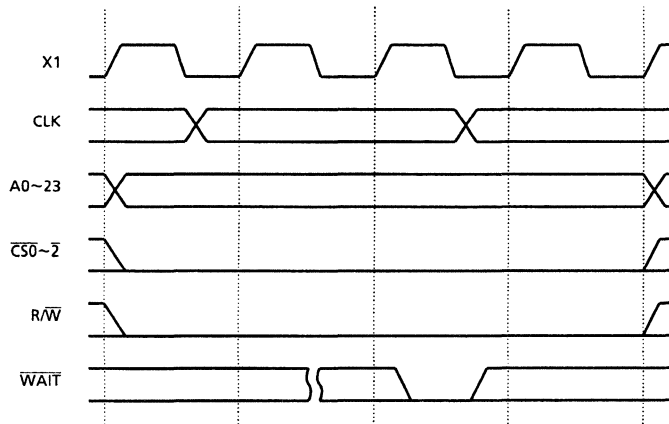


Figure 3.13 (1). DRAM Controller

**3.13.1 Control Register**

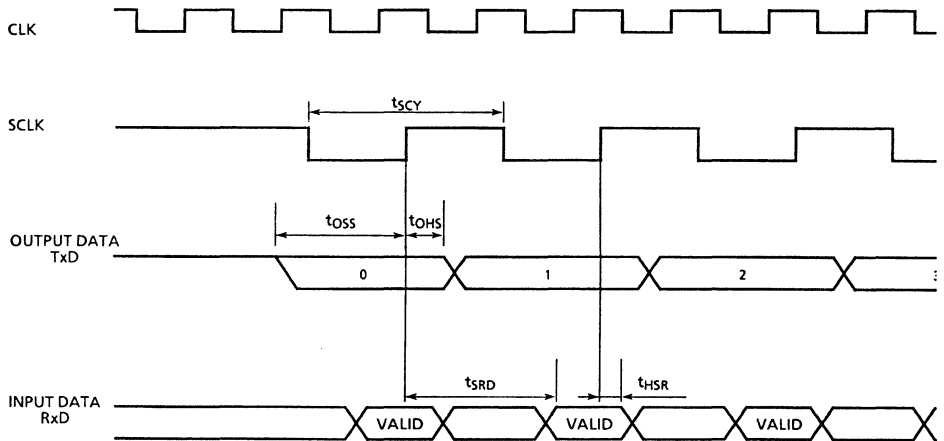


Figure 3.13 (2). Chip Select Wait Control Register (B3CS)

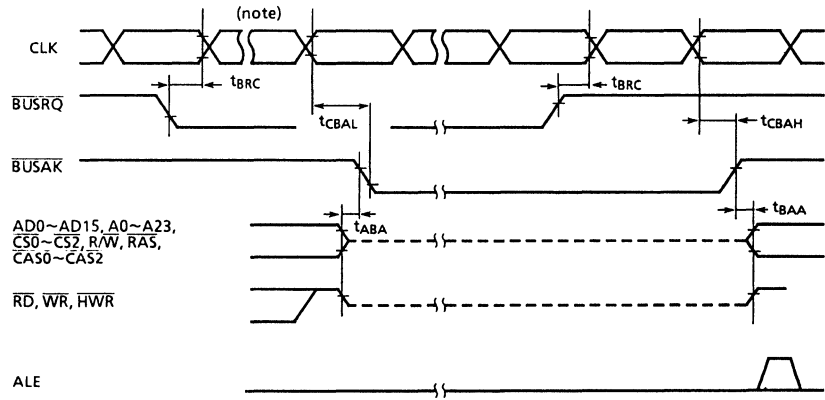


Figure 3.13 (3). Port 4 Function Register

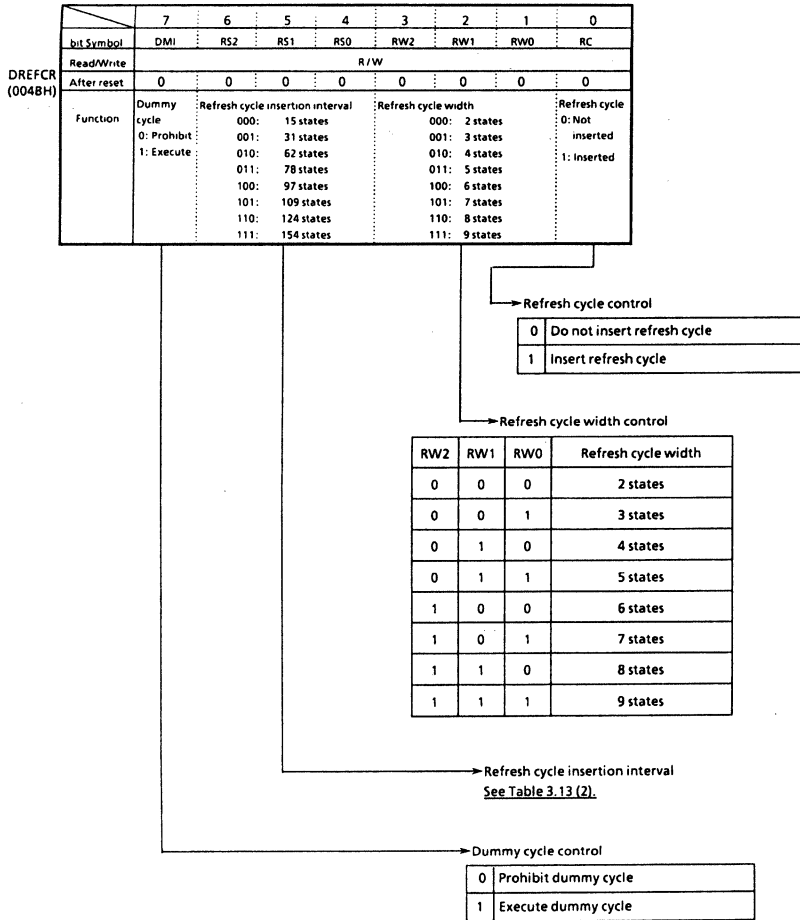


Figure 3.13 (4). Refresh Control Register

3.13.2 Operation Description

(1) Read/Write Control

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

accessed.

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

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

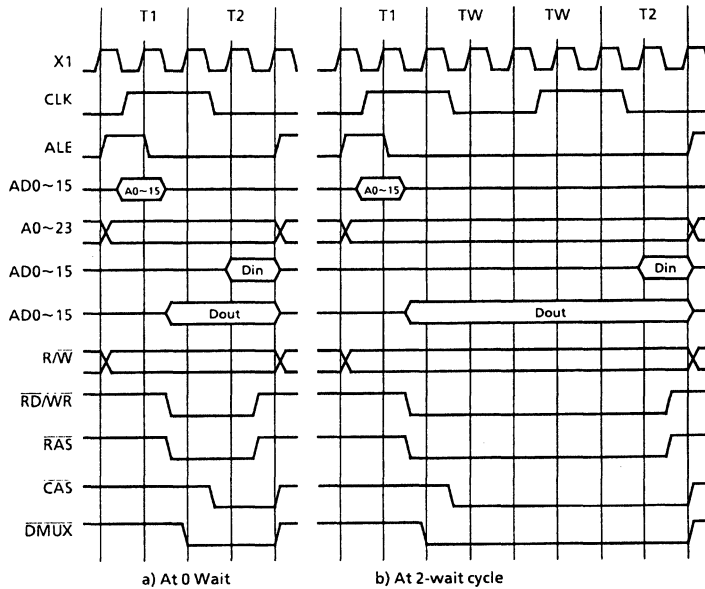


Figure 3.13 (5). Memory Access Cycle Timing

How to set the registers is described next.

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

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

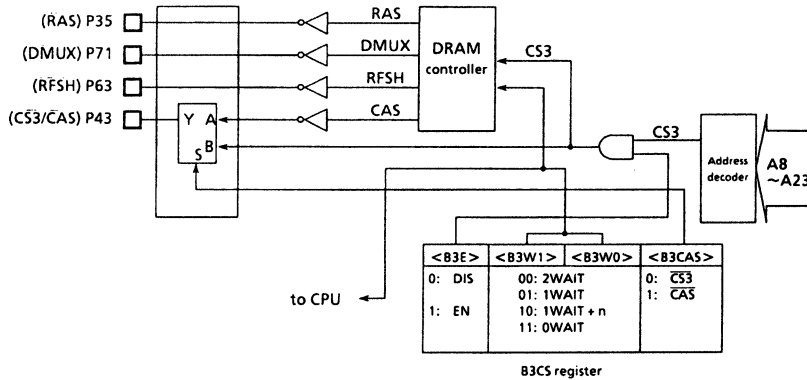


Figure 3.13 (6). Relationship Between Address Decoder and DRAM Controller

The  $\overline{\text{RAS}}$ ,  $\overline{\text{CAS}}$ ,  $\overline{\text{DMUX}}$ , and  $\overline{\text{RFSH}}$  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  $\overline{\text{RAS}}/\overline{\text{CAS}}$  used to refresh in DRAM. At the same time the state signal  $\overline{\text{RFSH}}$  which indicates a refresh cycle is output. DRAM can be refreshed easily because  $\overline{\text{RAS}}/\overline{\text{CAS}}$  output frequency and pulse width are programmable. The refresh controller has the following features.

- Refresh mode:  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  interval refresh mode  
 $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  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)  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  interval refresh mode

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

Therefore, TMP96C031F enables the  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  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  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  refresh cycle.

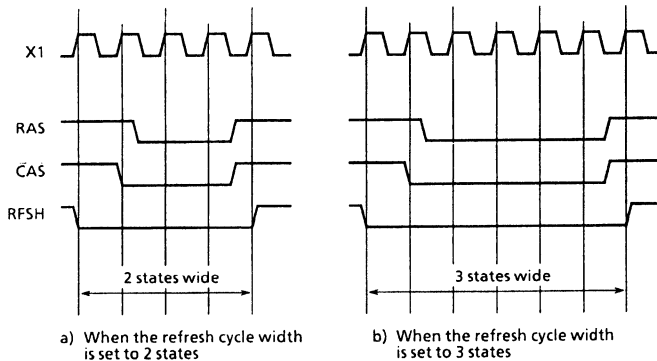


Figure 3.13 (7). Refresh Cycle Timing Example

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 (fosc)						
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 be used to change the refresh cycle width (RAS, CAS output). (2 to 9 states)

③ Refresh cycle control

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

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

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

However, RFSH is not output. ("1" is output.).

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

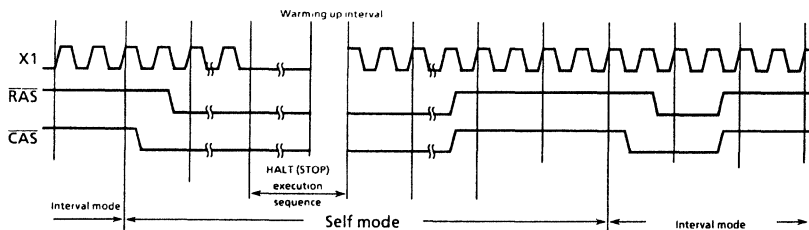


Figure 3.13 (8). Self Refresh Cycle Timing

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 CAS pin and 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 CAS before RAS refresh mode. (The first CAS before RAS refresh is performed immediately after cancellation because the refresh counter is cleared.)



(3) DRAM initialize

The DRAM controller can generate consecutive  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  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  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  dummy cycle timing.

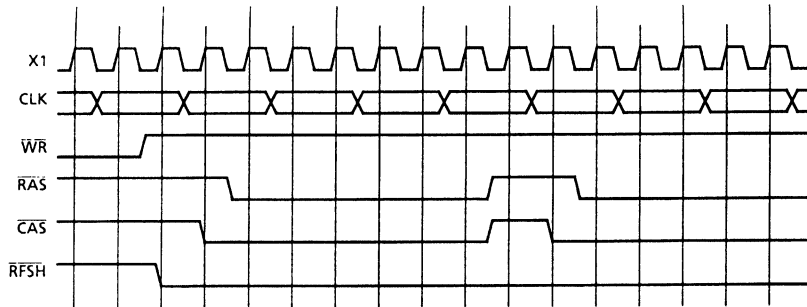


Figure 3.13 (9).  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  Dummy Cycle Timing (Fixed to 4 states)

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.

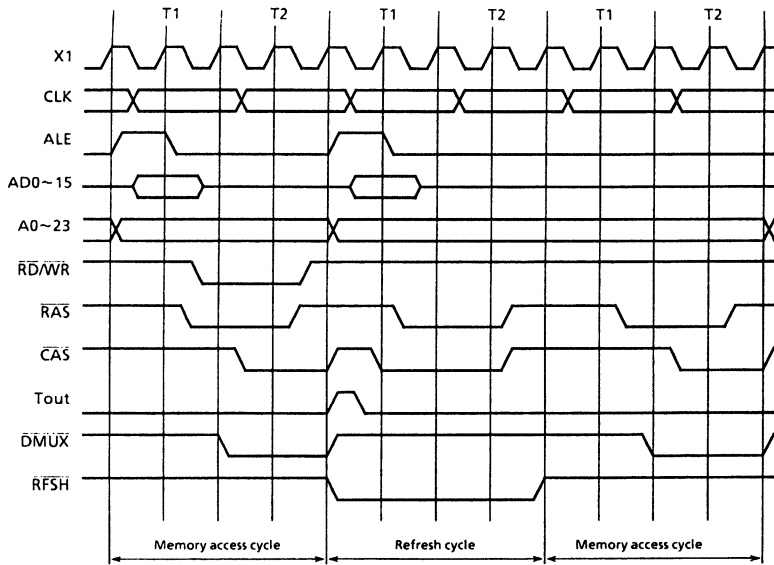
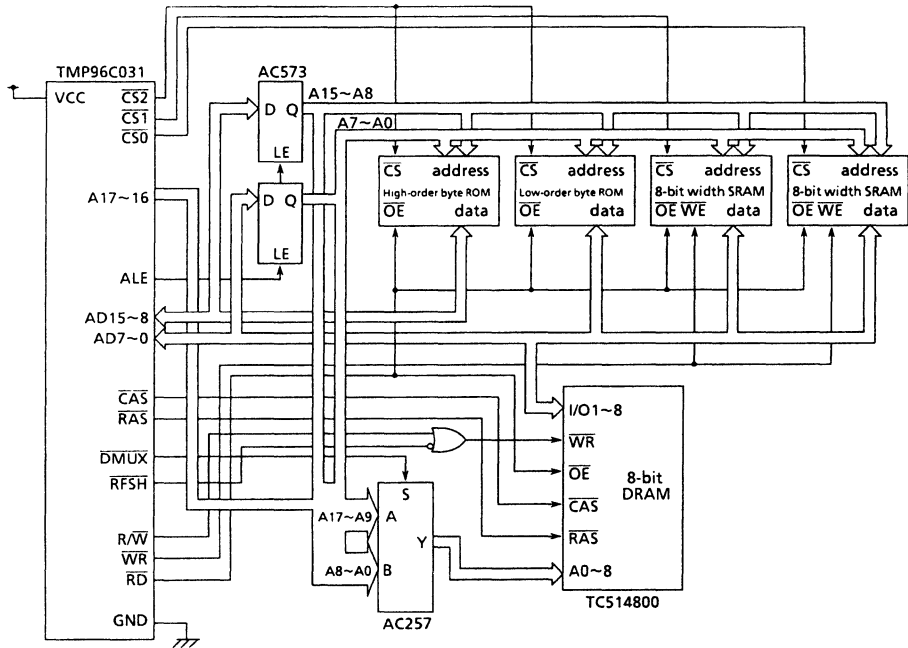


Figure 3.13 (101). Timing Diagram when Refresh Cycle is Inserted in Memory Access Cycle

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
$\Sigma I_{OL}$	Output Current (total)	100	mA
$\Sigma I_{OH}$	Output Current (total)	-100	mA
PD	Power Dissipation ( $T_a = 70^\circ\text{C}$ )	600	mW
T SOLDER	Soldering Temperature (10s)	260	$^\circ\text{C}$
T STG	Storage Temperature	-65 - 150	$^\circ\text{C}$
T OPR	Operating Temperature	-20 - 70	$^\circ\text{C}$

4.2 DC Characteristics (TMP96C031F)

$V_{CC} = 5V \pm 10\%$ ,  $T_a = -20 \sim 70^\circ C$  (Typical values are for  $T_a = 25^\circ C$  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	IOL = 1.6mA
V OH	Output High Voltage	2.4		V	IOH = -400 $\mu$ A
V OH1		$0.75V_{CC}$		V	IOH = -100 $\mu$ A
V OH2		$0.9V_{CC}$		V	IOH = -20 $\mu$ A
I DAR	Darlington Drive Current (8 Output Pins max.)	-1.0	-3.5	mA	V EXT = 1.5V R EXT = 1.1K $\Omega$
I LI	Input Leakage Current	0.02 (Typ)	$\pm 5$	$\mu$ A	$0.0 \leq V_{in} \leq V_{CC}$
I LO	Output Leakage Current	0.05 (Typ)	$\pm 10$	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
I <sub>CC</sub>	Operating Current (RUN)	30 (Typ)	60	mA	tosc = 20MHz
	IDLE	10	10	mA	
	STOP ( $T_a = -20 \sim 70^\circ C$ )	2.0 (Typ)	50	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
	STOP ( $T_a = 0 \sim 50^\circ C$ )	0.2 (Typ)	10	$\mu$ A	$0.2 \leq V_{in} \leq V_{CC} - 0.2$
V STOP	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	V IL2 = $0.2V_{CC}$ , V IH2 = $0.8V_{CC}$
R RST	RESET Pull Up Register	50	150	K $\Omega$	
C IO	Pin Capacitance		10	pF	tosc = 1MHz
V TH	Schmitt Width RESET, NMI, INTO (P50)	0.4	1.0 (Typ)	V	
R K	Pull Down/Up Register	50	150	K $\Omega$	

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

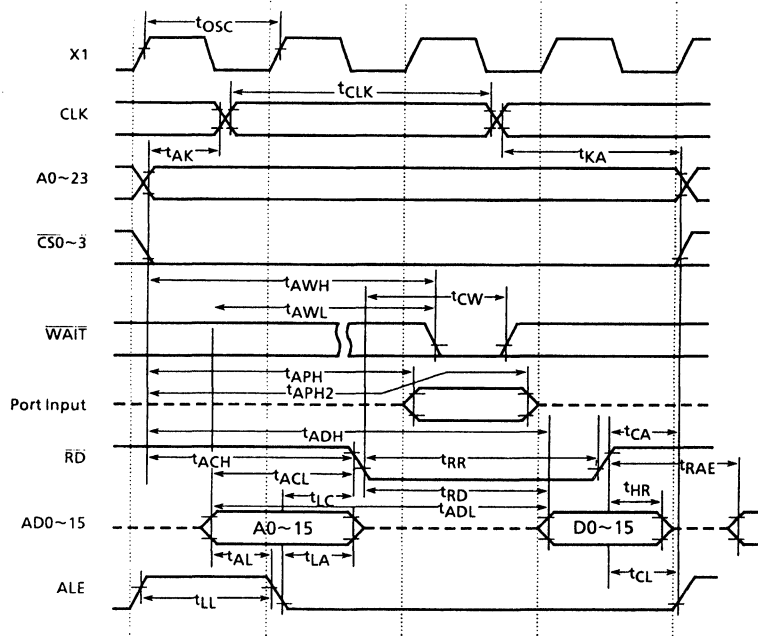
4.3 AC Electrical Characteristics (TMP96C031F)  $V_{CC} = 5V \pm 10\%$ ,  $T_a = -20 \sim 70^\circ\text{C}$  (4MHz ~ 20MHz)

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 $\rightarrow$ CLK Hold	0.5x - 20		11		5		ns
4	$t_{KA}$	CLK Valid $\rightarrow$ A0 - 23 Hold	1.5x - 70		24		5		ns
5	$t_{AL}$	A0-15 Valid $\rightarrow$ ALE fall	0.5x - 15		16		10		ns
6	$t_{LA}$	ALE fall $\rightarrow$ 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 $\rightarrow$ $\overline{RD}/\overline{WR}$ fall	0.5x - 30		1		-5		ns
9	$t_{CL}$	$\overline{RD}/\overline{WR}$ rise $\rightarrow$ ALE rise	0.5x - 20		11		5		ns
10	$t_{ACL}$	A0 - 15 Valid $\rightarrow$ $\overline{RD}/\overline{WR}$ fall	x - 25		38		25		ns
11	$t_{ACH}$	A0 - 23 Valid $\rightarrow$ $\overline{RD}/\overline{WR}$ fall	1.5x - 50		44		25		ns
12	$t_{CA}$	$\overline{RD}/\overline{WR}$ rise $\rightarrow$ A0 - 23 Hold	0.5x - 20		11		5		ns
13	$t_{ADL}$	A0 - 15 Valid $\rightarrow$ D0 - 15 input		3.0x - 45		143		105	ns
14	$t_{ADH}$	A0 - 23 Valid $\rightarrow$ D0 - 15 input		3.5x - 65		154		110	ns
15	$t_{RD}$	$\overline{RD}$ fall $\rightarrow$ 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 $\rightarrow$ D0 - 15 Hold	0		0		0		ns
18	$t_{RAE}$	$\overline{RD}$ rise $\rightarrow$ 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 $\rightarrow$ $\overline{WR}$ rise	2.0x - 50		75		50		ns
21	$t_{WD}$	$\overline{WR}$ rise $\rightarrow$ D0 - 15 Hold	0.5x - 10		21		15		ns
22	$t_{AEH}$	A0 - 23 Valid $\rightarrow$ WAIT input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	$t_{AWL}$	A0 - 15 Valid $\rightarrow$ WAIT input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	$t_{CW}$	$\overline{RD}/\overline{WR}$ fall $\rightarrow$ WAIT Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	$t_{APH}$	A0 - 23 Valid $\rightarrow$ PORT input		2.5x - 120		36		5	ns
26	$t_{APH2}$	A0 - 23 Valid $\rightarrow$ PORT Hold	2.5x + 50		206		175		ns
27	$t_{CP}$	$\overline{WR}$ rise $\rightarrow$ PORT Valid		200		200		200	ns

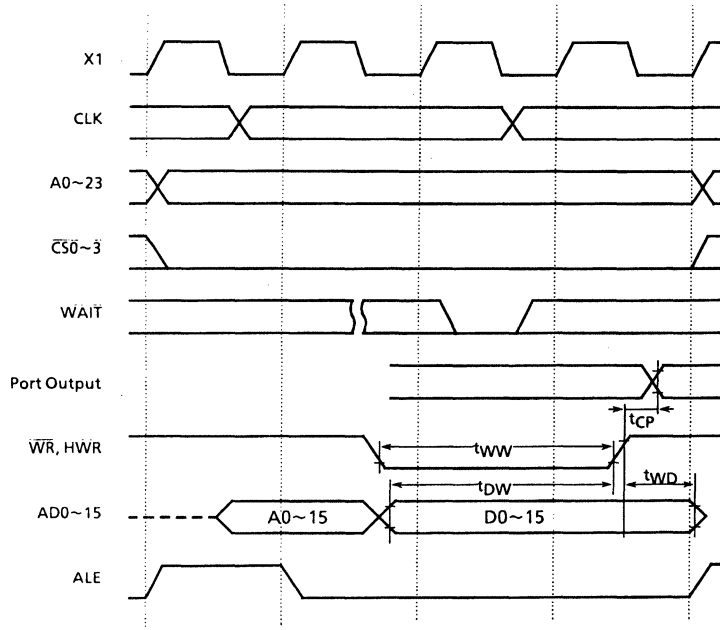
## AC Measuring Conditions

- 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 0.8V<sub>CC</sub> /Low 0.2V<sub>CC</sub> (Except for AD0 ~ AD15)

(1) Read Cycle



(2) Write Cycle





4.4 DRAM Control AC Characteristics (TMP96C031F)

V<sub>CC</sub> = 5V±10% TA = -20 ~ 70°C (4MHz ~ 20MHz)

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

\*1 CAS before RAS interval refresh mode

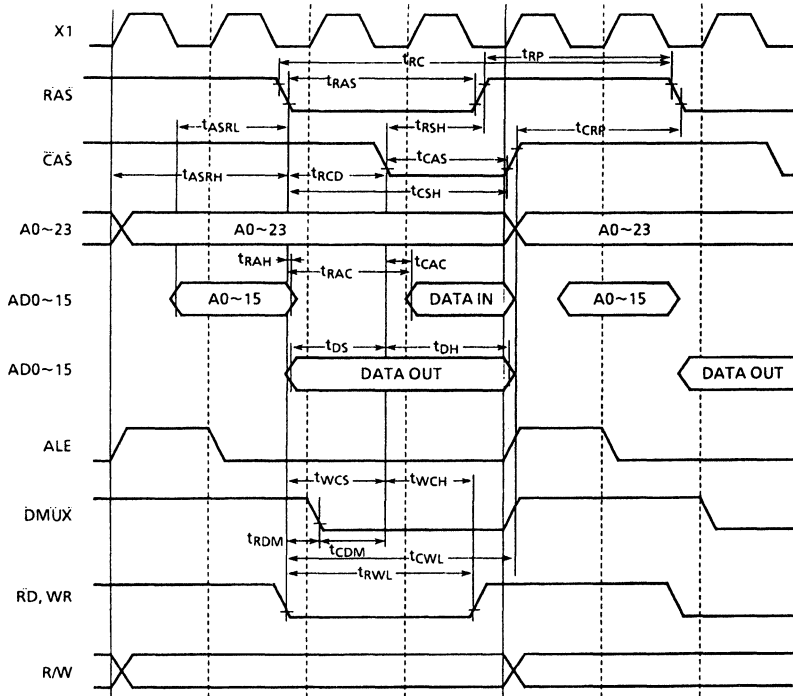
\*2 CAS before RAS self-refresh mode

\* Both refresh modes

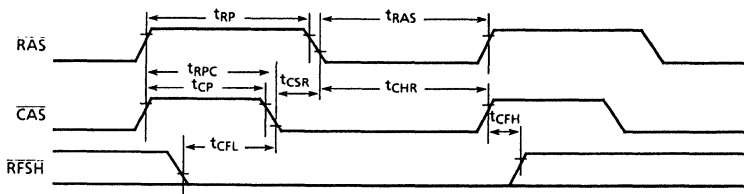
AC Measuring Conditions

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

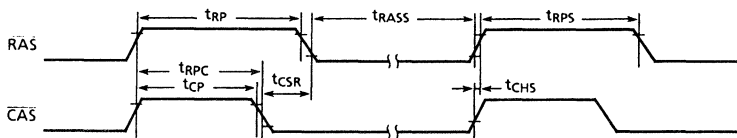
(1) Read/Write Access Cycle



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



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



**4.5 A/D Conversion Characteristics (TMP96C031F)**

$V_{CC} = 5V \pm 10\%$   $TA = -20 \sim 70^{\circ}C$

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 $\pm 0.5$ LSB not included)	Total error ( $TA = 25^{\circ}C, V_{CC} = V_{REF} = 5.0V$ )		1.0		LSB
	Total error			2.5	

**4.6 Serial Channel Timing - I/O Interface Mode**

$V_{CC} = 5V \pm 10\%$   $TA = -20 \sim 70^{\circ}C$

(1) SCLK Input Mode

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{SCY}$	SCLK cycle	16x		1		0.8		$\mu s$
$t_{OSS}$	Output Data $\rightarrow$ rising edge of SCLK	$t_{SCY}/2 - 5x - 50$		137		100		ns
$t_{OHS}$	SCLK rising edge $\rightarrow$ output data hold	$5x - 100$		212		150		ns
$t_{HSR}$	SCLK rising edge $\rightarrow$ input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge $\rightarrow$ 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	$\mu s$
$t_{OSS}$	Output Data $\rightarrow$ rising edge of SCLK	$t_{SCY} - 2x - 150$		725		550		ns
$t_{OHS}$	SCLK rising edge $\rightarrow$ output data hold	$2x - 80$		45		20		ns
$t_{HSR}$	SCLK rising edge $\rightarrow$ input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge $\rightarrow$ effective data input		$t_{SCY} - 2x - 150$		725		550	ns

**4.7 Timer/Counter Input Clock (T10, T14, T15)**

$V_{CC} = 5V \pm 10\%$   $TA = -20 \sim 70^{\circ}C$

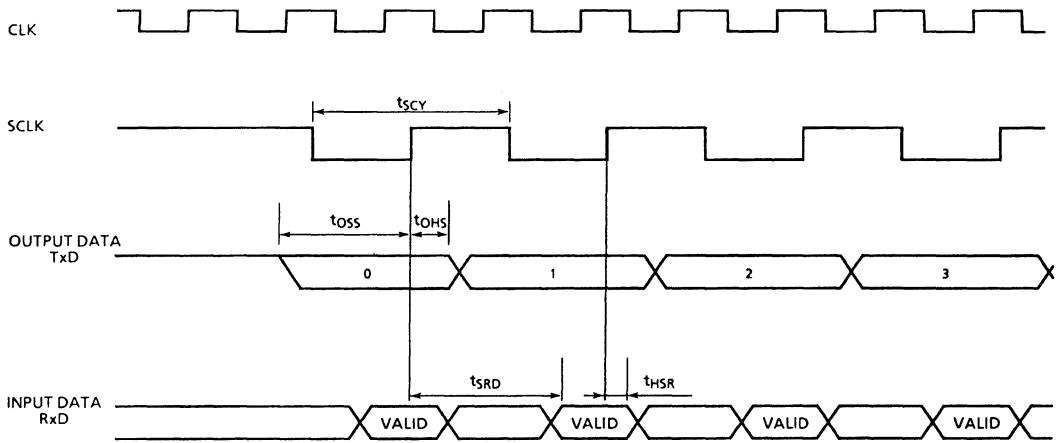
Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{VCK}$	Clock cycle	$8x + 100$		600		500		ns
$t_{VCKL}$	Low level clock pulse width	$4x + 40$		290		240		ns
$t_{VCKH}$	High level clock pulse width	$4x + 40$		290		240		ns

## 4.8 Interrupt Operation

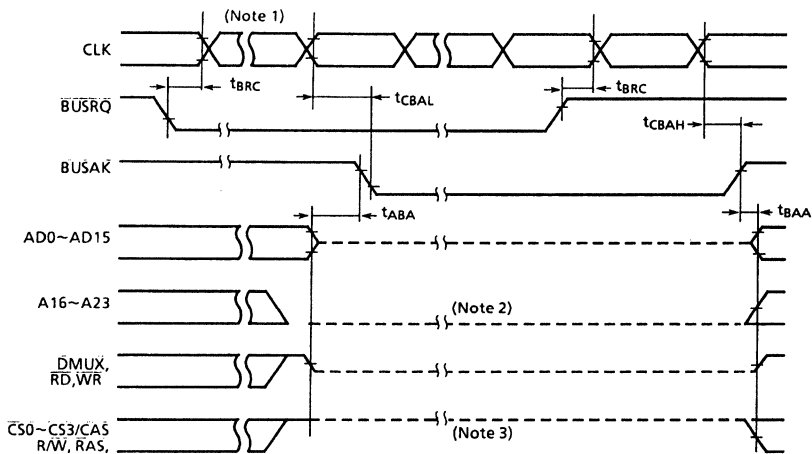
$$V_{CC} = 5V \pm 10\% \quad T_a = -20 \sim 70^\circ\text{C}$$

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{INTAL}$	$\overline{NMI}$ , INTO Low level pulse width	4x		250		200		ns
$t_{INTAH}$	$\overline{NMI}$ , INTO 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 <sub>BRC</sub>	BUSRQ setup time for CLK	120		120		120		ns
t <sub>CBAL</sub>	CLK→BUSAK falling edge		1.5x + 120		214		220	ns
t <sub>CBAH</sub>	CLK→BUSAK rising edge		0.5x + 40		71		65	ns
t <sub>ABA</sub>	Output buffer is off to BUSAK	0	80	0	80	0	80	ns
t <sub>BAA</sub>	BUSAK output buffer is on.	0	80	0	80	0	80	ns

Note 1: The Bus will be released after the WAIT request is inactive, when the 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

Symbol	Name	Address	7	6	5	4	3	2	1	0

→ bit Symbol

→ Read / Write

→ Initial value after reset

→ Remarks

Table 5 I/O Register Address Map

Address	Name	Address	Name	Address	Name	Address	Name
000000H		20H	TRUN	40H	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	PGOREG	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	ADM0D	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 (Prohibit RMW)	08H	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C		
			W									
			0	0	0	0	0	0	0	0	0	
			<<Refer to the "P2FC">>									
P2FC	PORT2 Function (Prohibit RMW)	09H	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F		
			W									
			0	0	0	0	0	0	0	0	0	
			P2FC/ P2CR = 00 : IN, 01 : OUT, 10 : -, 11 : A23 - 16									
P3CRL	PORT3 Control Low (Prohibit RMW)	0AH	P33C1	P33C0	P32C1	P32C0	P31C1	P31C0	P30C1	P30C0		
			W									
			0	0	0	0	0	0	0	0	0	
			00 : PORT input 01 : PORT output 10 : BUSAK 11 : -		00 : PORT input 01 : PORT output 10 : BUSRQ 11 : -		00 : PORT input 01 : PORT output 10 : - 11 : -		00 : PORT input 01 : PORT output 10 : TO5 11 : FWR			
P3CRH	PORT3 Control High (Prohibit RMW)	0BH	RDEN				P35C1	P35C0	P34C1	P34C0		
			W									
			0				0	0	0	0		
			1 : pseudo SRAM EN				00 : PORT input 01 : PORT output 10 : RAS 11 : -		00 : PORT input 01 : PORT output 10 : NMI 11 : R/W			
P4FC	PORT4 Function (Prohibit RMW)	10H	BUSWDT	BUDRM			P43F	P42F	P41F	P40F		
			W									
			0	0			0	0	0	0		
			00 : BUSSRQ DIS 01 : BUSRQ EN		0 : ON 1 : OFF				00 : PORT 01 : CS3/CAS	00 : PORT 01 : CS2	00 : PORT 01 : CS1	00 : PORT 01 : 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 : 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 : TxDO			
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 : 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 : T03 11 : DMUX		00 : PORT input 01 : PORT output 10 : T01 11 : T04			
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	PORUN	T1RUN	TORUN		
			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 : $2^6 - 1$ PWM 10 : $2^7 - 1$ Cycle 11 : $2^8 - 1$		00 : T0TRG 01 : $\phi T1$ 10 : $\phi T16$ 11 : $\phi T256$		00 : T10 Input 01 : $\phi T1$ 10 : $\phi T4$ 11 : $\phi T16$			
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 : $2^6 - 1$ PWM 10 : $2^7 - 1$ Cycle 11 : $2^8 - 1$		00 : T02TRG 01 : $\phi T1$ 10 : $\phi T16$ 11 : $\phi T256$		00 : T10 Input 01 : $\phi T1$ 10 : $\phi T4$ 11 : $\phi T16$			
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 ↑ T15 ↑ 10 : T14 ↑ T14 ↑ 11 : TFF1 ↑ TFF1 ↓		1 : UC4 Clear Enable	Source Clock 00 : T14 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 (Prohibit RMW)	4CH	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
			W				R/W			
			0	0	0	0	Undefined			
PG1REG	PG1 Register (Prohibit RMW)	4DH	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 <sup>16</sup> /f <sub>ic</sub> 01 : 2 <sup>18</sup> /f <sub>ic</sub> 10 : 2 <sup>20</sup> /f <sub>ic</sub> 11 : 2 <sup>22</sup> /f <sub>ic</sub>	Warming up Time 0 : 2 <sup>14</sup> /f <sub>ic</sub> 1 : 2 <sup>16</sup> /f <sub>ic</sub>	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	0	
			Receiving data bit 8	Parity 0 : Odd 1 : Even	1 : Parity Enable	1 : Error			Overrun	Parity	Framing	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	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 : T00 Trigger 01 : Baud rate generator 10 : Internal clock $\phi$ 1 11 : Don't care			
BR0CR	Baud Rate Control	53H	-		BROCK1	BROCK0	BR053	BR052	BR051	BR050		
			R/W	R/W								
			0		0	0	0	0	0	0		
			Fix at "0"		00 : $\phi$ 10 (fc/4) 01 : $\phi$ 12 (fc/16) 10 : $\phi$ 18 (fc/64) 11 : $\phi$ 32 (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: SCLK1
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 : T00 Trigger 01 : Baud rate generator 10 : Internal clock $\phi$ 1 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 : $\phi$ 10 (fc/4) 01 : $\phi$ 12 (fc/16) 10 : $\phi$ 18 (fc/64) 11 : $\phi$ 32 (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 (Prohibit RMW)	70H	INT1				INT0			
			I1C	I1M2	I1M1	I1M0	I0C	I0M2	I0M1	I0M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE23	INTerrupt Enable 2/3 (Prohibit RMW)	71H	INT3				INT2			
			I3C	I3M2	I3M1	I3M0	I2C	I2M2	I2M1	I2M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE45	INTerrupt Enable 4/5 (Prohibit RMW)	72H	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE67	INTerrupt Enable 6/7 (Prohibit RMW)	73H	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE10	INTerrupt Enable Timer 1/0 (Prohibit RMW)	74H	INTT1 (timer 1)				INTT0 (timer 0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE32	INTerrupt Enable Timer 2/3 (Prohibit RMW)	75H	INTT3 (timer 3)				INTT2 (timer 2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE54	INTerrupt Enable Treg 5/4 (Prohibit RMW)	76H	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE50	INTerrupt Enable Serial 0 (Prohibit RMW)	77H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTE51	INTerrupt Enable Serial 1 (Prohibit RMW)	78H	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W	W	W	R/W	W	W	W
			0	0	0	0	0	0	0	0
INTEAD	INTerrupt Enable A/D (Prohibit RMW)	79H	INTAD							
			IADC	IADM2	IADM1	IADM0				
			R/W	W	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	DMA0V5	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 : INTO rising edge 1 : INT1 falling edge	1 : INTO input enable	0 : INTO edge mode 1 : INTO level mode	1 : Operate even at 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	
			0: $\overline{\text{CS0}}$ DIS 1: 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: $\overline{\text{CS1}}$ DIS 1: 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: $\overline{\text{CS0}}$ DIS 1: CS0 EN	↑	0: 8000 ~ 3FFFFFF 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: $\overline{\text{CS3/CAS}}$ DIS 1: $\overline{\text{CS3/CAS}}$ EN	↑	0: Undefined 1: Address area specification	↑	↑		0: $\overline{\text{CS3/CAS}}$ DIS 1: $\overline{\text{CS3/CAS}}$ EN	0: Self refresh execution 1: Release		
MSAR0	Memory Start Address 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									
MAMRO	Memory Start Address 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 Address 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 Address Mask Reg. 1	43H	V21	V20	V19	V18	V17	V16	V15 ~ 9	V8		
			R/W									
			1	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 Address Reg. 2	44H	S23	S22	S21	S20	S19	S18	S17	S16		
			R/W									
			1	1	1	1	1	1	1	1	1	
			A23 ~ A16 Memory start address setting									
MAMR2	Memory Start Address Mask Reg. 2	45H	V22	V21	V20	V19	V18	V17	V16	V15		
			R/W									
			1	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 Address Reg. 3	46H	S23	S22	S21	S20	S19	S18	S17	S16		
			R/W									
			1	1	1	1	1	1	1	1	1	
			A23 ~ A16 Memory start address setting									
MAMR3	Memory Start Mask Address Reg. 3	46H	V22	V21	V20	V19	V18	V17	V16	V15		
			R/W									
			1	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

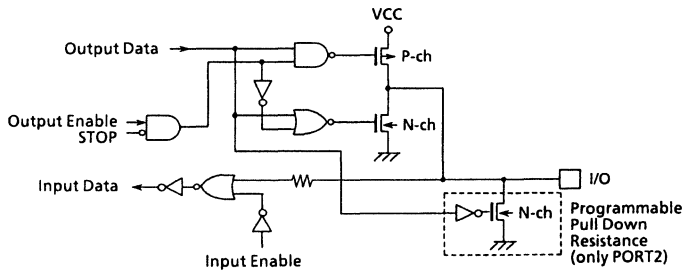
Basically, the gate singles written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The dedicated signal is described below.

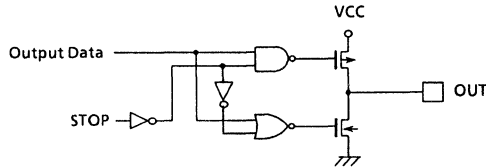
STOP: This signal becomes active "1" when the hold mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit [DRIVE] is set to "1", however, STP remains at "0".

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

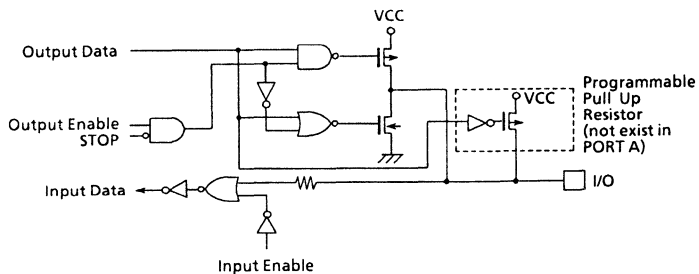
- PO (AD0 ~ AD7), P1 (AD8 ~ 15, A8 ~ 15), P2 (A16 ~ 23)



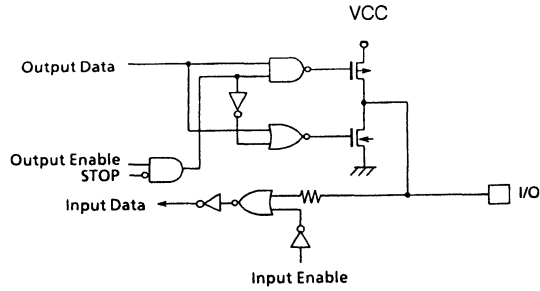
- $\overline{RD}$ ,  $\overline{WR}$



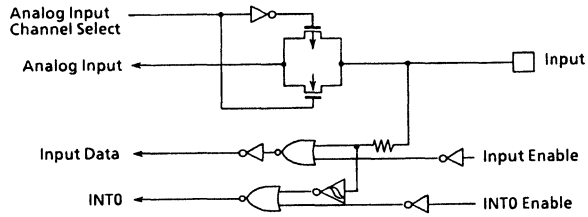
- P30 ~ 33, P35



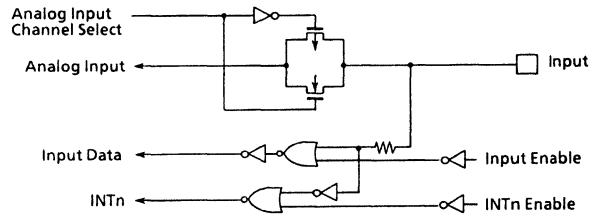
- P61 ~ P67, P70 ~ P73, P75 ~ P76



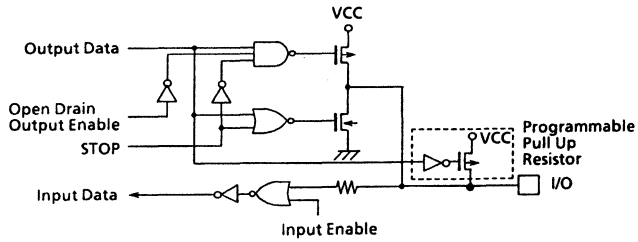
- P50 (AN0/INT0)



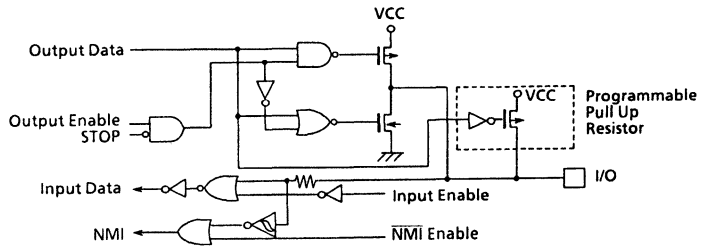
- P51 ~ P53 (AN1 ~ 3/INT1 ~ 3)



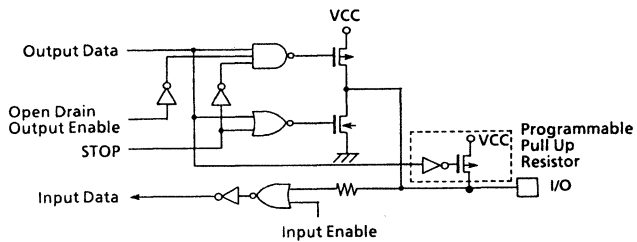
- P60 (TXD0), P74 (TXD1)



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

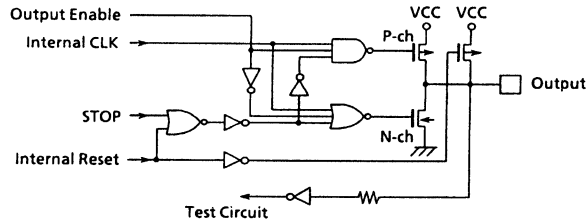


- P40- P43 ( $\overline{\text{CS0}} \sim \overline{\text{CS3}}/\overline{\text{CAS}}$ )

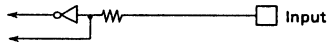




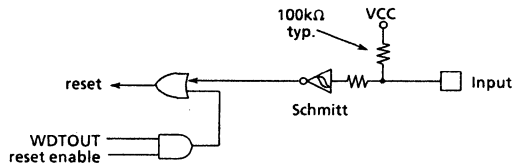
• CLK



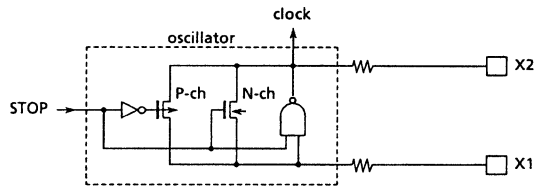
•  $\overline{\text{AM8/16}}$



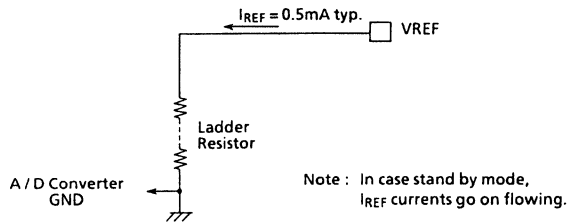
•  $\overline{\text{RESET}}$



• 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 TORUN 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

$$2/20\text{MHz} = 100\text{ns} = 1 \text{ state}$$

(2) Guidelines

① AM8/ $\overline{\text{T6}}$  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  $\mu$ DMA)

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.

### CMOS 16-bit Microcontrollers

#### TMP96C081F

#### 1. Outline and Device Characteristics

The TMP96C081F are high-speed advanced 16-bit microcontrollers developed for controlling medium to large-scale equipment.

The TMP96C081F are housed in a 100-pin flat package. Device characteristics are as follows:

- (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 (1.6 $\mu$ s/2 bytes @ 20MHz)
- (2) Minimum instruction execution time
  - 200ns @ 20MHz
- (3) Internal DMAC: 4 channels
- (4) External memory expansion
  - Can be expanded up to 16M bytes (for both programs and data).
  - AM8/ $\overline{16}$  pin (select the external data bus width)
  - Can mix 8- and 16-bit external data buses.
    - ...Dynamic data bus sizing
- (5) 8-bit timers: 2 channels
- (6) 8-bit PWM timers: 2 channels
- (7) 16-bit timers: 2 channels
- (8) Pattern generators: 4 bits, 2 channels
- (9) Serial interface: 2 channels
- (10) 10-bit A/D converter: 6 channels
- (11) Watchdog timer
- (12) Chip select/wait controller: 5 blocks
- (13) Interrupt functions
  - 3 CPU interrupts: ... SWI instruction, privileged violation, and illegal instruction
  - 18 internal interrupts  $\square$  7-level priority can be set.
  - 6 external interrupts  $\square$
- (14) I/O ports: Maximum 64 pins
- (15) Standby function : 3 halt modes (RUN, IDLE, STOP)

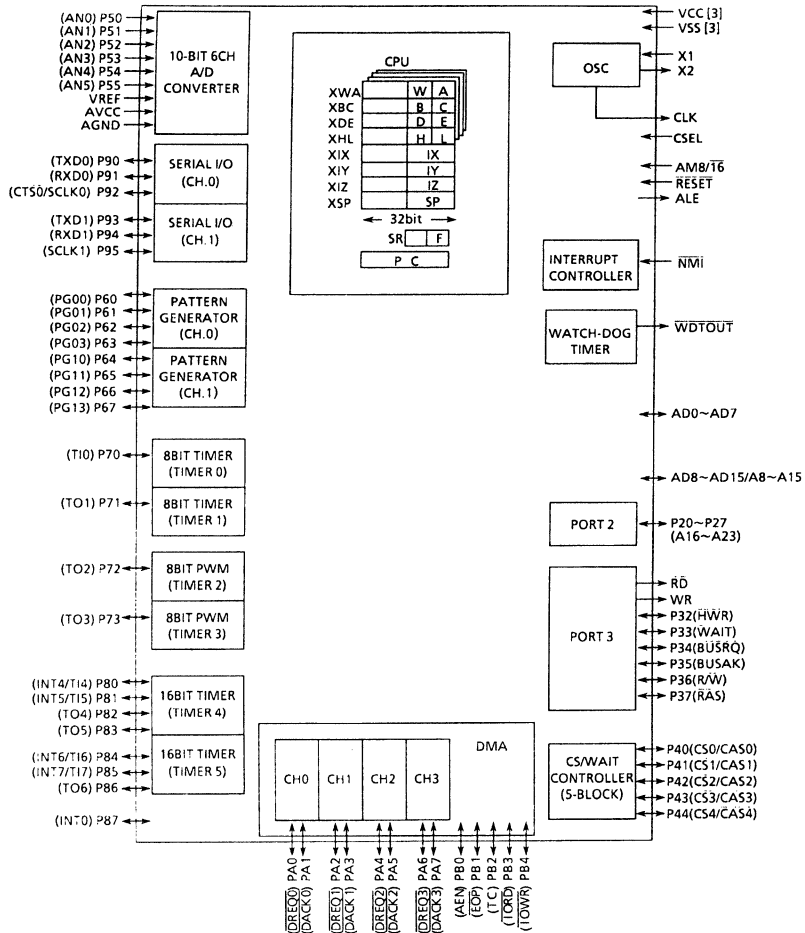


Figure 1. TMP96C081F Block Diagram

## 2. Pin Assignment and Functions

The assignment of input/output pins for TMP96C081F, their name and outline functions.

### 2.1 Pin Assignment

Figure 2.1 shows pin assignment of TMP96C081F.

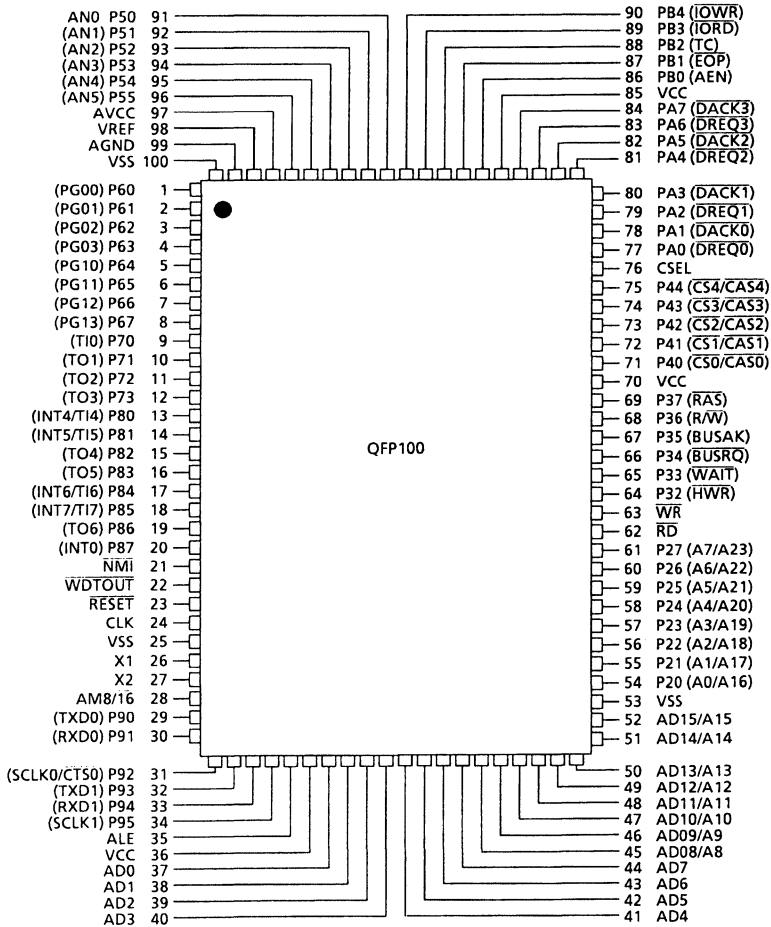


Figure 2.1. Pin Assignment (100-pin QFP)

## 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
$\overline{RD}$	1	Output	Read: Strobe signal for reading external memory
$\overline{WR}$	1	Output	Write: Strobe signal for writing data on pins AD0 -7
P32 $\overline{HWR}$	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 - 15
P33 $\overline{WAIT}$	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P34 $\overline{BUSRQ}$	1	I/O Output Input	Port 34: I/O port (with pull-up resistor) Bus request: Signal used to request high impedance for AD0 - 15, A0 - 23, $\overline{RD}$ , $\overline{WR}$ , $\overline{HWR}$ , $\overline{R/W}$ , $\overline{RAS}$ , $\overline{CS0}$ , $\overline{CS1}$ , $\overline{CS2}$ , $\overline{CS3}$ and $\overline{CS4}$ pins. (For external DMAC)
P35 $\overline{BUSAK}$	1	I/O Output	Port 35: I/O port (with pull-up resistor) Bus acknowledge: Strobe indicating that AD0 - 15, A0 - 23, $\overline{RD}$ , $\overline{WR}$ , $\overline{HWR}$ , $\overline{R/W}$ , $\overline{RAS}$ , $\overline{CS0}$ , $\overline{CS1}$ , $\overline{CS2}$ , $\overline{CS3}$ and $\overline{CS4}$ pins are at high impedance after receiving $\overline{BUSRQ}$ . (For external DMAC)
P36 $\overline{R/W}$	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle 0, write cycle.
P37 $\overline{RAS}$	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs $\overline{RAS}$ strobe for DRAM.
P40 $\overline{CS0}$ $\overline{CAS0}$	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs $\overline{CAS}$ strobe for DRAM when address is within specified address area.
P41 $\overline{CS1}$ $\overline{CAS1}$	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs $\overline{CAS}$ strobe for DRAM when address is within specified address area.
P42 $\overline{CS2}$ $\overline{CAS2}$	1	I/O Output Output	Port 42: I/O port (with pull-up resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs $\overline{CAS}$ strobe for DRAM when address is within specified address area.
P43 $\overline{CS3}$ $\overline{CAS3}$	1	I/O Output Output	Port 43: I/O port (with pull-up resistor) Chip select 3: Outputs 0 if address is within specified address area. Column address strobe 3: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P44 $\overline{CS4}$ $\overline{CAS4}$	1	I/O Output Output	Port 43: I/O port (with pull-up resistor) Chip select 3: Outputs 0 if address is within specified address area. Column address strobe 4: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.

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

Pin Name	Number of Pins	I/O	Functions
P50 – P53 AN0 – AN5	6	Input Input	Port 5: Input port Analog input: Input to A/D converter
VREF	1		Pin for reference voltage input to A/D converter
AVCC	1		Power supply pin for A/D converter
AGND	1		Ground
P60 – 63 PG00 – 03	4	I/O Output	Port s 60 – 63: I/O ports that allow selection of I/O on a bit basis Pattern generator ports: 00 - 03
P64 – 67 PG10 – 13	4	I/O Output	Port s 64 – 67: I/O ports that allow selection of I/O on a bit basis Pattern generator ports: 10 - 13
P70 TI0	1	I/O Input	Port 70: I/O port Timer input 0: Timer 0 or 1 output
P71 TO1	1	I/O Output	Port 71: I/O port Timer output 1: Timer 0 or 1 output
P72 TO2	1	I/O Output	Port 72: I/O port PWM output 2: 8-bit PWM timer 2 output
P73 TO3	1	I/O Output	Port 73: I/O port PWM output 3: 8-bit PWM timer 3 output
P80 TI4 INT4	1	I/O Input Input	Port 80: I/O port Timer output 4: Timer 4 count/capture trigger signal input Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge.
P81 TI5 INT5	1	I/O Input Input	Port 81: I/O port Timer output 5: Timer 4 count/capture trigger signal input Interrupt request pin 5: Interrupt request pin with programmable rising edge.
P82 TO4	1	I/O Output	Port 82: I/O port Timer output 4: Timer 4 output pin
P83 TO5	1	I/O Output	Port 83: I/O port Timer output 5: Timer 4 output pin

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

Pin Name	Number of Pins	I/O	Functions
P84 TI6 INT6	1	I/O Input Input	Port 84: I/O port Timer input 6: Timer 5 count/capture trigger signal input Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
P85 TI7 INT7	1	I/O Input Input	Port 85: I/O port Timer output 7: Timer 5 count/capture trigger signal input Interrupt request pin 7: Interrupt request pin with rising edge
P86 TO6	1	I/O Output	Port 86: I/O port Timer output 6: Timer 5 output pin
P87 INT0	1	I/O Input	Port 87: I/O port Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P90 TXD0	1	I/O Output	Port 90: I/O port Serial send data 0
P91 RXD0	1	I/O Input	Port 91: I/O port Serial receive data 0
P92 CTS0	1	I/O Input	Port 92: I/O port Serial data send enable 0 (Clear to Send)
P93 TXD1	1	I/O Output	Port 93/I/O port Serial send data 1
P94 RXD1	1	I/O Input	Port 94: I/O port Serial receive data 1
P95 SCLK1	1	I/O I/O	Port 95: I/O port Serial clock I/O 1
PA0 DREQ0	1	I/O Input	Port A0: I/O port DMA request 0: DMA channel 0 request pin
PA1 DACK0	1	I/O Output	Port A1: I/O port DMA acknowledge 0: DMA channel 0 acknowledge signal
PA2 DREQ1	1	I/O Input	Port A2: I/O port DMA request 1 : DMA channel 1 request pin
PA3 DACK1	1	I/O Output	Port A3: I/O port DMA acknowledge 1: DMA channel 1 acknowledge signal
PA4 DREQ2	1	I/O Input	Port A4: I/O port DMA request 2: DMA channel 2 request pin

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



Pin Name	Number of Pins	I/O	Functions
PA5 DACK2	1	I/O Output	Port A5: I/O port DMA acknowledge 2: DMA channel 2 acknowledge signal
PA6 DREQ3	1	I/O Input	Port A6: I/O port DMA request 3: DMA channel 3 request pin
PA7 DACK3	1	I/O Output	Port A7: I/O port DMA acknowledge 3: DMA channel 3 acknowledge signal
PB0 AEN	1	I/O Output	Port B0: I/O port Address enable: Enabled when internal DMA has bus mastership
PB1 EOP	1	I/O Input	Port B1: I/O port DMA acknowledge 3: DMA channel 3 acknowledge signal
PB2 TC	1	I/O Output	Port B2: I/O port Terminal count: Output signal to indicate transfer completion.
PB3 IORD	1	I/O Output	Port B3: I/O port I/O read signal: Strobe signal for reading external I/O when DMA is in signal address mode.
PB4 IOWR	1	I/O Output	Port B4: I/O port I/O write signal: Strobe signal for writing external I/O when DMA is in signal address mode.
WDTOUT	1	Output	Watchdog timer output pin
NMI	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output : Outputs $\lfloor X1 \div 4 \rfloor$ clock. Pulled-up during reset.
AM8/16	1	Input	Address mode: Selects external Data Bus width. "0" should be input with fixed 16 bit Bus width or 16 bit Bus interfaced width 8bit Bus. "1" should be input with fixed 8 bit Bus width.
CSEL	1	Input	CPU select: Signal used for emulation. Fix either "0" or "1".
ALE	1	Output	Address latch enable
RESET	1	Input	Reset: Initializes LSI. (With pull-up resistor)
X1/X2	2	I/O	Oscillator connecting pin
VCC	2		Power supply pin (+5V)pin
VSS	2		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 TMP96C081F device.

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

#### 3.1 CPU

The TMP96C081F 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 TMP96C081F that are not described in the previous section.

#### 3.1.1 Reset

To reset the TMP96C081F, the  $\overline{\text{RESET}}$  input must be kept at 0 for at least 10 system clocks (10 states: 1 $\mu$ s 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.)

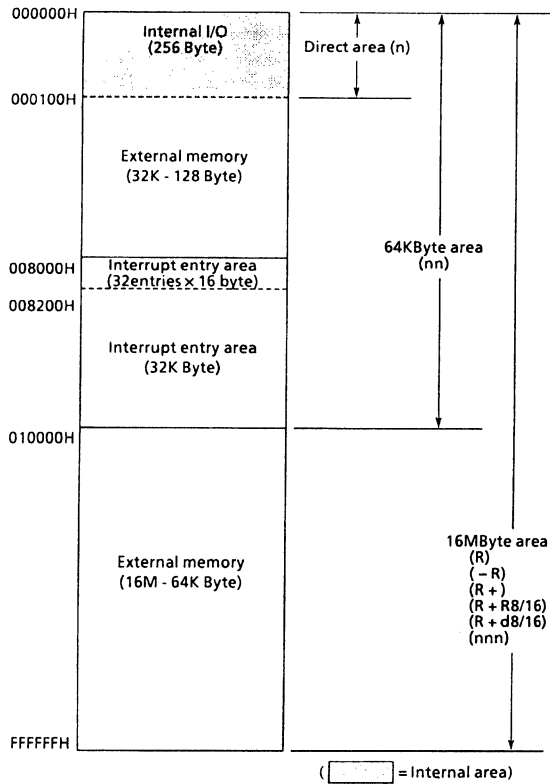
When reset is released, instruction execution starts from address 8000H. CPU internal registers other than the above are not changed.

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  $\overline{\text{WDTOU}}$  pin to 0. (Watchdog timer is set to enable after reset.)
- Pulls up the CLK pin to 1.
- Sets the ALE pin to 0.

### 3.2 Memory Map

Figure 3.2 is a memory map of the TMP96C081F.



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.

The TMP96C081F has altogether the following 27 interrupt sources:

- Interrupts from the CPU...3  
(Software interrupts, privileged violations, and Illegal (undefined) instruction execution)
- Interrupts from external pins (NMI, INT0, and INT4 to 7)...6
- Interrupts from built-in I/Os...18

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 micro DMA processing mode. High-speed micro DMA is a mode used by the CPU to automatically transfer byte or word data. It enables the CPU to process interrupts such as data saves to built-in I/Os at high speed.

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

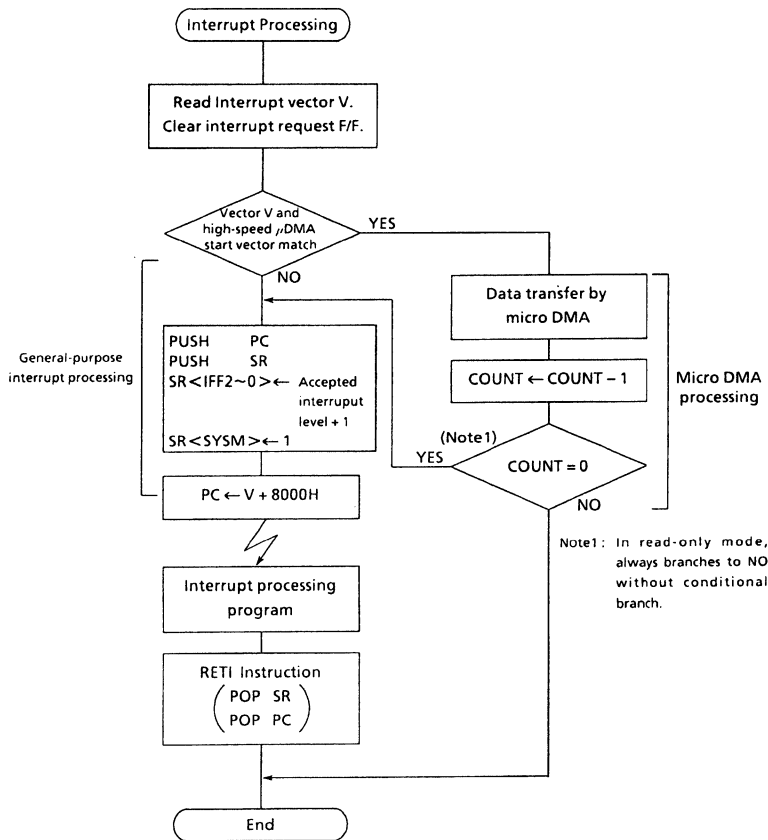


Figure 3.3 (1). Interrupt Processing Flowchart

### 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 interrupt request with a priority higher than the accepted now interrupt during the CPU is processed above (1) ~ (5) is accepted before the 1'st instruction in the interrupt processing routine, causing interrupt processing to nest. This is the same case of overlapped each Non-maskable interrupt (level "7"). The CPU does not accept an interrupt request of the same level as that of the interrupt being processed.

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

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

Table 3.3 (1) TMP96C081F Interrupt Table

Default Priority	Type	Interrupt Source	Vector Value "V"	Start Address	High-Speed Micro DMA Start Vector
1	Non-Maskable	Reset , or SW10 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	INTO 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
32		INTAD: A/D conversion completion	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		INTTR6: 16-bit timer 5 (TREG6)	0 1 6 0 H	8 1 6 0 H	16H
23		INTTR7: 16-bit timer 5 (TREG7)	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		DMA0: DMA channel 0	0 1 C 0 H	8 1 C 0 H	1CH
29		DMA1: DMA channel 1	0 1 D 0 H	8 1 D 0 H	1DH
30		DMA2: DMA channel 2	0 1 E 0 H	8 1 E 0 H	1EH
31	DMA3: DMA channel 3	0 1 F 0 H	8 1 F 0 H	1FH	

### 3.3.2 High-Speed Micro DMA

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

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

#### (1) High-Speed Micro DMA Operation

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

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

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

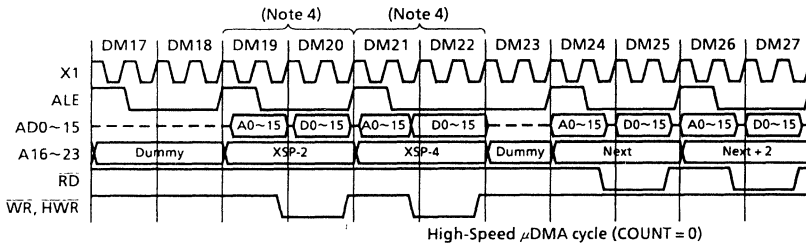
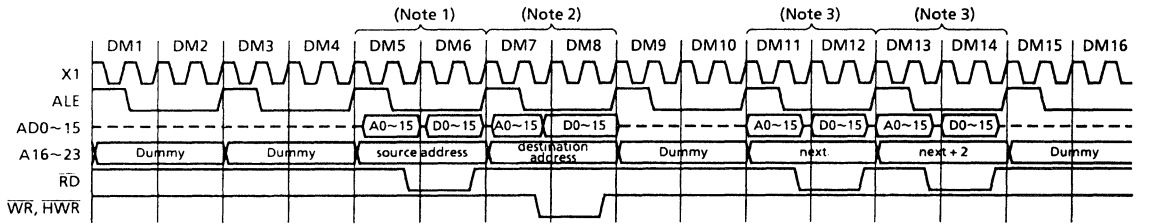
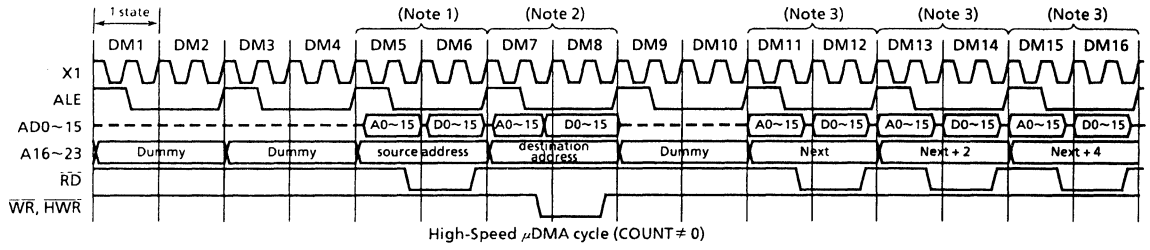
After transferring data using the high-speed micro DMA and the transfer counter has been decremented to 0, the program goes to a general-purpose interrupt processing. Note that after interrupt processing, when an interrupt for the same channel is generated, if the system requires resetting the transfer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

For the source and destination address registers, please reset the register number when specifying a particular register number. Otherwise, the register number is counted in order.

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

The following timing chart shows a high-speed micro DMA cycle of the transfer address in increment mode (all modes except the Read-only mode operate similarly). (Condition: MIN mode, 16-bit Bus width for 16M Byte, 0 wait)





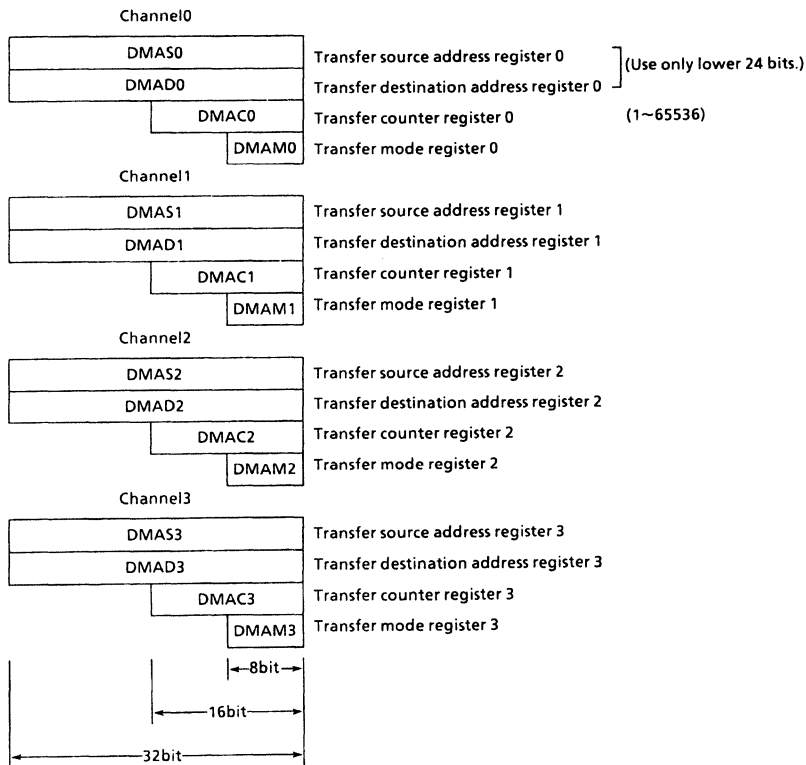
(Note1) This is added 2 states the case of the bus width of source address area is 8bit

(Note3) This may be a dummy cycle with instruction queue buffer.

(Note2) This is added 2 states the case of the bus width of destination address area is 8bit

(Note4) This is added 2 states the case of the bus width of stack address area is 8bit

(2) Register Configuration (CPU Control Register)



This Control Register cannot be set only "LDC cr, r" instruction.

(3) Transfer Mode Register Details

(DMAM0~3)

0	0	0	0	Mode	Note : When specifying values for this register, set the upper 4 bits to 0.
<p>Z: 0 = byte transfer, 1 = word transfer</p> <p>execution time (Min.)</p>					
0	0	0	Z	Transfer destination address INC mode ..... for I/O to memory (DMADn +) ← (DMASn) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	0	1	Z	Transfer destination address DEC mode ..... for I/O to memory (DMADn -) ← (DMASn) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	0	Z	Transfer source address INC mode ..... for I/O to memory (DMADn) ← (DMASn +) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	1	Z	Transfer source address DEC mode ..... for I/O to memory (DMADn) ← (DMASn -) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	0	Z	Fixed address mode ..... I/O to I/O (DMADn) ← (DMASn) DMACn ← DMACn - 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	1	0	Read-only mode ..... for DRAM refresh Dummy ← (DMASn) ; Reads 4 bytes. DMASn ← DMASn + 4 ; Increments lower word only. DMACn ← DMACn - 1	14 states (1.4 μs)
1	0	1	1	Counter mode ..... for interrupt counter DMASn ← DMASn + 1 DMACn ← DMACn - 1 if DMACn = 0 then INT.	11 states (1.1 μs)

(1 states = 100ns @ 20MHz)

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.)  
 Execution time : Indicates when the destination/source address space is 16-bit bus width and is set to 0 WAIT.

All address space which is can be accessed by high-speed μDMA is the space for system mode, specified chip

select/wait controller. Do not use undefined codes for transfer mode control.

#### (4) Example for High-Speed Micro DMA Operation

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

When the hardware configuration is as follows:

DRAM mapping size: = 1MB  
 DRAM data bus size: = 8 bits  
 DRAM mapping address range: = 1000000H to 1FFFFFFH

Set the following registers first; refresh is performed automatically.

##### ① Register initial value setting

```
LD   XIX, 1000000H
LDC  DMAS0, XIX    ... mapping start address
LD   A, 00001010B
LDC  DMAM0, A      ... read only mode (for DRAM
                   refresh
```

##### ② Timer setting

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

##### ③ Interrupt controller setting

Set the timer interrupt above level at the other desired interrupt request level. Write the above timer interrupt vector value in the micro DMA start vector register, DMA0V.

##### (Operation description)

The DRAM data bus is in an 8-bit bus and the high-speed μDMA is in read-only mode (4 bytes), so refresh is performed for four times per interrupt. When a 512 refresh/8ms DRAM is connected, DRAM refresh is performed sufficiently if the high-speed μDMA is started every  $15.625\mu\text{s} \times 4 = 62.4\mu\text{s}$  or less, since the timing is  $15.625\mu\text{s}/\text{refresh}$ .

##### (Overhead)

Each processing time by the high-speed μDMA is  $1.8\mu\text{s}$  (18 states) @ 20MHz with an 8-bit data bus. In the above example, the micro DMA is started every 62.5μs,  $1.8\mu\text{s}/62.5\mu\text{s} = 0.0288$ ; thus, the overhead is 2.88%.

##### (Note)

Please be aware that a refresh is ineffective at a bus release which an interrupt is in a wait state, because the high-speed micro DMA is started by an interrupt.

#### 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 INTO interrupt request, set the register after the DI instruction as follows.

```
INTE0AD←----- 0 ---    Zero-clears the INTO 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 (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 micro DMA 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 micro DMA processing (see Table 3.3 (1)), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter registers (e.g., DMAS and DMAD) prior to the micro DMA processing.

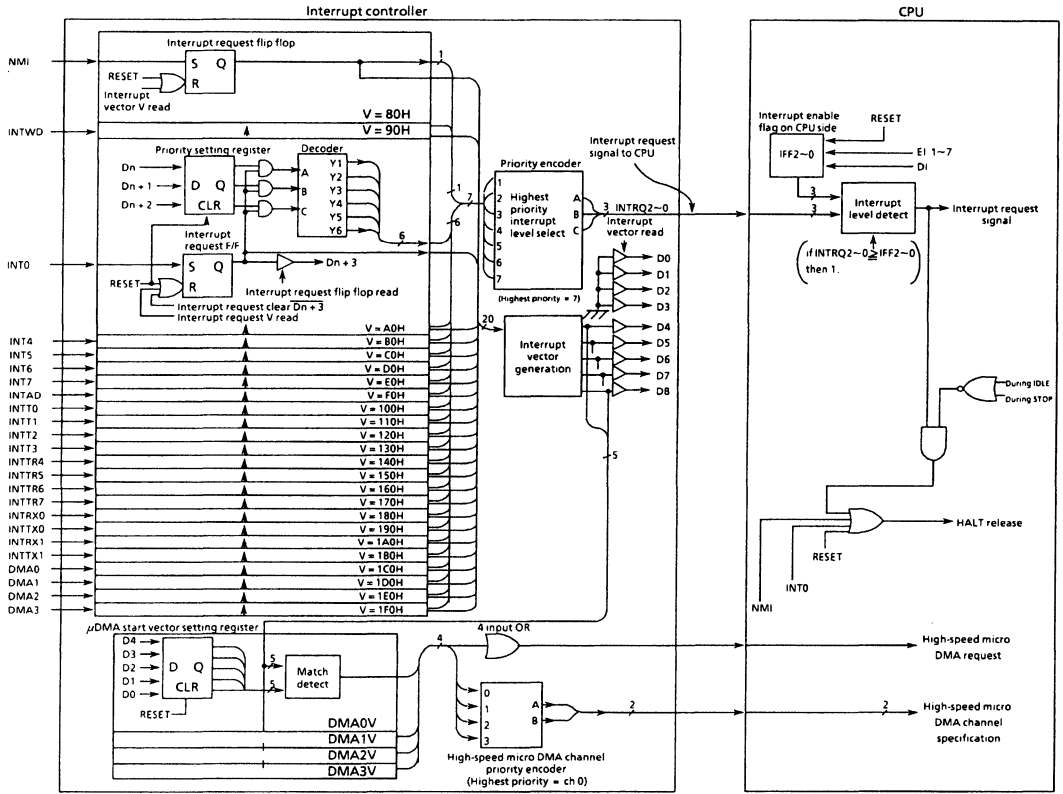


Figure 3.3.3 (1). Block Diagram of Interrupt Controller

(1) Interrupt Priority Setting Register

(Read-modify-write prohibited.)

Symbol	Address	7	6	5	4	3	2	1	0
INTE0AD	70H	INTAD				INT0			
		IADC	IADM2	IADM1	IADM0	IOC	IOM2	IOM1	IOM0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTE45	71H	INT5				INT4			
		ISC	ISM2	ISM1	ISM0	I4C	I4M2	I4M1	I4M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTE67	72H	INT7				INT6			
		I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTE710	73H	INTT1 (Timer1)				INTT0 (Timer0)			
		IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTEPW10	74H	INTT3 (Timer3/PWM1)				INTT2 (Timer2/PWM0)			
		IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTE754	75H	INTTR5 (TREG5)				INTTR4 (TREG4)			
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTE76	76H	INTTR7 (TREG7)				INTTR6 (TREG6)			
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTE50	77H	INTTX0				INTRX0			
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTE51	78H	INTTX1				INTRX1			
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTDMA0	79H	DMA1				DMA0			
		ID1C	ID1M2	ID1M1	ID1M0	ID0C	ID0M2	ID0M1	ID0M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0
INTDMA1	7AH	DMA3				DMA2			
		ID3C	ID3M2	ID3M1	ID3M0	ID2C	ID2M2	ID2M1	ID2M0
		R/W	W	W	W	R/W	W	W	W
		0	0	0	0	0	0	0	0

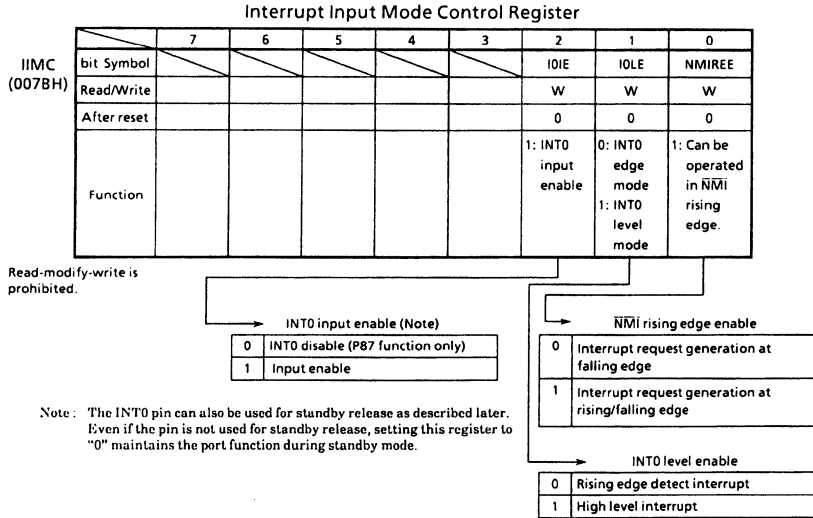
←Interrupt source  
←bit Symbol  
←Read/Write  
←After reset

ixxM2	ixxM1	ixxM0	Function (Write)
0	0	0	Prohibits interrupt request.
0	0	1	Sets interrupt request level to "1".
0	1	0	Sets interrupt request level to "2".
0	1	1	Sets interrupt request level to "3".
1	0	0	Sets interrupt request level to "4".
1	0	1	Sets interrupt request level to "5".
1	1	0	Sets interrupt request level to "6".
1	1	1	Prohibits interrupt request.

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

(2) External Interrupt Control



Setting of External Interrupt Pin Functions

Interrupt	Pin name	Mode	Setting method
$\bar{N}MI$	—	Falling edge	IIMC<NMIREE> = 0
		Rising and falling edges	IIMC<NMIREE> = 1
INT0	P87	Rising edge	IIMC<IOLE> = 0, <IOIE> = 1
		Level	IIMC<IOLE> = 1, <IOIE> = 1
INT4	P80	Rising edge	T4MOC<CAP12M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T4MOD<CAP12M1,0> = 1, 0
INT5	P81	Rising edge	—
INT6	P84	Rising edge	T5MOC<CAP34M1,0> = 0,0 or 0,1 or 1,1
		Falling edge	T5MOD<CAP34M1,0> = 1, 0
INT7	P85	Rising edge	—

**(3) High-Speed Micro DMA Start Vector**

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

micro DMA mode for the channel whose value matched. If the high-speed  $\mu$ DMA vector matches more than one channel, the channel with the lower channel number has a higher priority.

**Micro DMA0 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA0V (007CH)	bit Symbol			DMA0V8	DMA0V7	DMA0V6	DMA0V5	DMA0V4
				Read/Write				
	After reset			0	0	0	0	0

**Micro DMA1 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA1V (007DH)	bit Symbol			DMA1V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4
				Read/Write				
	After reset			0	0	0	0	0

**Micro DMA2 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA2V (007EH)	bit Symbol			DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4
				Read/Write				
	After reset			0	0	0	0	0

**Micro DMA3 Start Vector**

(read-modify-write is not possible.)

	7	6	5	4	3	2	1	0
DMA3V (007FH)	bit Symbol			DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4
				Read/Write				
	After reset			0	0	0	0	0

**(4) Notes**

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

while reading the interrupt vector after accepting the interrupt. If so, the CPU would read the default vector 00A0 and start the interrupt processing from the address 80A0.

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 TMP96C081F 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)		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
	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 <sup>16</sup> / f <sub>c</sub> 01 : 2 <sup>18</sup> / f <sub>c</sub> 10 : 2 <sup>20</sup> / f <sub>c</sub> 11 : 2 <sup>22</sup> / f <sub>c</sub> Detection time	Warming up time 0 : 2 <sup>16</sup> / f <sub>c</sub> 1 : 2 <sup>18</sup> / f <sub>c</sub>	Standby mode 00 : RUN mode 01 : STOP mode 10 : IDLE mode 11 : Don't care	1: Connects watchdog timer output to 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. (same the external oscillator) To release by a reset, it is necessary to allow a reset time long enough 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 NMI or INTO pin, or a reset can be used. The details are described below:

**Standby Release by Interrupt**


Interrupt Level Standby Mode	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. (Note 1)	Can only be released by INTO pin. Processing resumes from address next to HALT instruction.
IDLE	Can only be released by $\overline{\text{NMI}}$ or INTO pin. After standby mode is released, interrupt processing starts. (Note 1)	↑
STOP	↑ (Note 1)	↑

Note 1: When releasing standby setting INTO 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 (1/2)

Pin Name	I/O	DRVE = 0	DRVE = 1
P0	Input mode/AD0 ~ 7 Output mode	— —	— Output
P1	Input mode/AD8 ~ 15 Output mode /A8 ~ 15	— —	— Output
P2	Input mode Output mode/A0 ~ 7, A16 ~ 23	PD* PD*	PD* Output
$\overline{RD}$ , $\overline{WR}$	Output	—	Output
P32 ~ P37	Input mode Output mode	PU PU	PU Output
P40, P41, P43, P44	Input mode Output mode	PU* PU*	PU* Output
P42 ( $\overline{CS2/CAS2}$ )	Input mode Output mode	PD* PD*	PD* Output
P50 ~ P53	Input	—	—
P54	Input mode Output mode	Input —	Input Output
P60 ~ P63	Input	—	Input
P64	Input mode Output mode	Input —	Input Output
P7	Input mode Output mode	— —	Input Output
P80 ~ P83	Input mode Output mode	— —	Input Output
P86 ( $\overline{NMI}$ ) P87 ( $\overline{INT0}$ )	Input mode Output mode	Input —	Input Output
P90	Input mode Output mode	PU* PU*	PU* Output
P91 ~ P93	Input mode Output mode	— —	Input Output
PA	Input mode Output mode	— —	Input Output
PB	Input mode Output mode	— —	Input Output
$\overline{WDTOU}$	Output	Output	Output
ALE	Output	"0"	"0"

—: Input for input mode/input pin is invalid; output mode/output pin is at high impedance.

: 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.

Table 3. 4 (1) Pin States in STOP Mode (1/2)

Pin Name	I/O	DRVE = 0	DRVE = 1
CLK	Output	—	"1"
RESET	Input	Input	Input
AM8/16	Input	Input	Input
X1	Input	—	—
X2	Output	"1"	"1"
D - A0 D - A1	Output	0V	0V

—: Input for input mode/input pin is invalid; output mode/output pin is at high impedance.

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.

### 3.5 Functions of Ports

The TMP96C081F has 64 bits for I/O ports.

These port pins have I/O functions for the built-in CPU and

internal I/Os as well as general-purpose I/O port functions.

Table 3.5 (1) lists the function of each port pin.

**Table 3.5 (1) Functions of Ports**

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

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port2	P20 to P27	8	I/O	↓	Bit	A0 to A/A16 to A23
Port3	P32	1	I/O	↑	Bit	HWR
	P33	1	I/O	↑	Bit	WAIT
	P34	1	I/O	↑	Bit	BUSRQ
	P35	1	I/O	↑	Bit	BUSAK
	P36	1	I/O	↑	Bit	R/W
	P37	1	I/O	↑	Bit	RAS
Port4	P40	1	I/O	↑	Bit	CS0/CAS0
	P41	1	I/O	↑	Bit	CS1/CAST
	P42	1	I/O	↑	Bit	CS2/CAS2
	P43	1	I/O	↑	Bit	CS3/CAS3
	P44	1	I/O	↑	Bit	CS4/CAS4
Port5	P50 to P55	6	Input	—	(Fixed)	AN0 ~ AN5
Port6	P60 to P67	8	I/O	↑	Bit	PG00 ~ PG03, PG10 ~ PG13
Port7	P70	1	I/O	—	Bit	Ti0
	P71	1	I/O	—	Bit	T01
	P72	1	I/O	—	Bit	T02
	P73	1	I/O	—	Bit	T03
Port8	P80	1	I/O	—	Bit	T14/INT4
	P81	1	I/O	—	Bit	T15/INT5
	P82	1	I/O	—	Bit	T04
	P83	1	I/O	—	Bit	T05
	P84	1	I/O	—	Bit	T16/INT6
	P85	1	I/O	—	Bit	T17/INT7
	P86	1	I/O	—	Bit	T06
	P87	1	I/O	—	Bit	INT0
Port9	P90	1	I/O	↑	Bit	TxD0
	P91	1	I/O	—	Bit	RxD0
	P92	1	I/O	—	Bit	CTS0/SCLK0
	P93	1	I/O	↑	Bit	TxD1
	P94	1	I/O	—	Bit	RxD1
	P95	1	I/O	—	Bit	SCLK1
PortA	PA0 ~ PA7	8	I/O	—	Bit	DREQ0 ~ 3, DACK0 ~ 3
PortB	PA0	1	I/O	—	Bit	AEN
	PA1	1	I/O	—	Bit	EOP
	PA2	1	I/O	—	Bit	TC
	PA3	1	I/O	—	Bit	IORD
	PA4	1	I/O	—	Bit	IOWR

Resetting makes the port pins listed below function as general-purpose I/O ports.

I/O pins programmable for input or output are set to input ports.

To set port pins for built-in functions, a program is required.

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

When the bus is released ( $\overline{\text{BUSAK}} = "0"$ ), the output

buffer for AD0 - AD15, A0 - A23, control signal ( $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ ,  $\overline{\text{HWR}}$ , R/W, RAS, CS0/CAS0 - CS4/CAS4) is off and their state become high-impedance.

However, the output of built-in programmable pull up/down resistors are kept before the bus is released. These programmable pull up/down resistors can be selected ON/OFF by programmable when they are used as the input ports.

They are used as the output ports, they cannot be selected ON/OFF by programmable.

The following in the pin state when the bus is released ( $\overline{\text{BUSAK}} = "0"$ ).

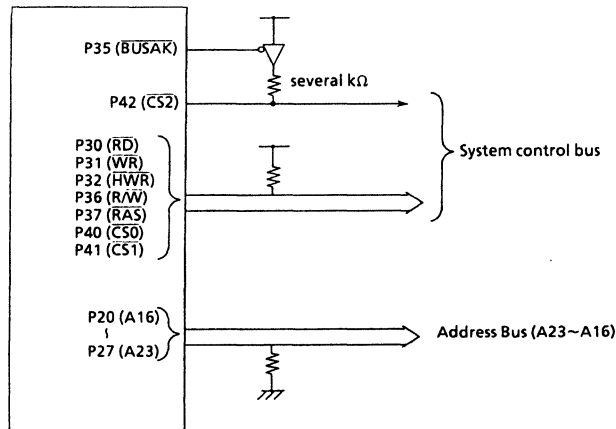
Pin Name	Pin state at bus release	
	Used as the port	Used as the function
P00 to P07 AD8 to 15/A8 to 15	—	becomes high-impedance (HZ).
$\overline{\text{RD}}$ $\overline{\text{WR}}$	—	becomes high-impedance (HZ); ("Hz" status after these pins are driven to high level.)
P32 ( $\overline{\text{HWR}}$ ) P37 (RAS)	The state is not changed. (does not become high-impedance (HZ).)	The output buffer is "OFF". These pins are added to the internal resistor of pull-up. It's no relation for the value of output latch.
P36 ( $\overline{\text{R/W}}$ ) P40 (CS0/CAS0) P41 (CS1/CAS1) P42 (CS3/CAS3) P42 (CS4/CAS4)	The state is not changed. (does not become high-impedance (HZ).)	The output buffer is "OFF". These pins are added to the internal resistor of pull-up. It's no relation for the value of output latch.
P42 (CS2/CAS2)	The state is not changed. (does not become high-impedance (HZ).)	The output buffer is "OFF". These pins are added to the internal resistor of pull-up. It's no relation for the value of output latch.
P20 - P27 (A16 - A23)	The state is not changed. (does not become high-impedance (HZ).)	The output buffer is "OFF". These pins are added to the internal resistor of pull-down. It's no relation for the value of output latch.

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

When bus is released, both internal memory and internal

I/O can be accessed. But the internal I/O continues to operate.

So, the watchdog timer also continues to run. Therefore, be careful about bus releasing time and set the detection time of WDT.



Example of the interface circuit (Using bus releasing function)

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

Resetting sets RD, WR to output, P40 P41, P43, P44 (CS0, CS1, CS3, CS4), P32 (HWR), P36 (R/W), P37 (RAS), and P35 (BUSAK) to input pull up resistor, P42 (CS2) and P20 ~ 27 (A16 ~ 23) to input with pull down resistor.

The above circuit is necessary to fix the signal level after reset because of the external pull up resistor collisions with the internal pull down resistor.

The value of this external pull up resistor must be several kΩ (The value of the internal pull down resistor is about 50 ~ 150kΩ).

P20 ~ P27 (A16 ~ 23) also needs circuit like P42 (CS2) to fix the signal level.

But for the P20 ~ P27 (A16 ~ 23) which does not have the means ("L" is active), add pull down directly like above circuit.

**3.5.1 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. To disconnect the pull-down resistor, write "1" in the output latch.

In addition to functioning as a general-purpose I/O port, Port 2 also functions as an address bus (A0 to 7 and A16 to 23).

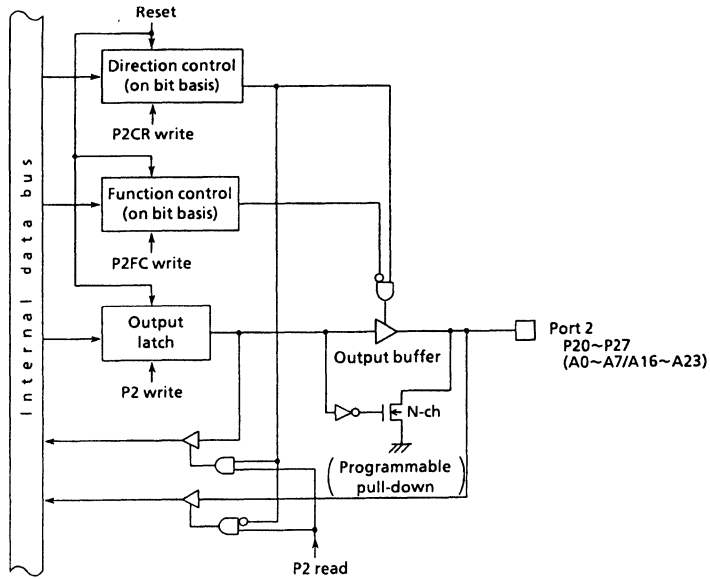


Figure 3.5 (1). Port 2

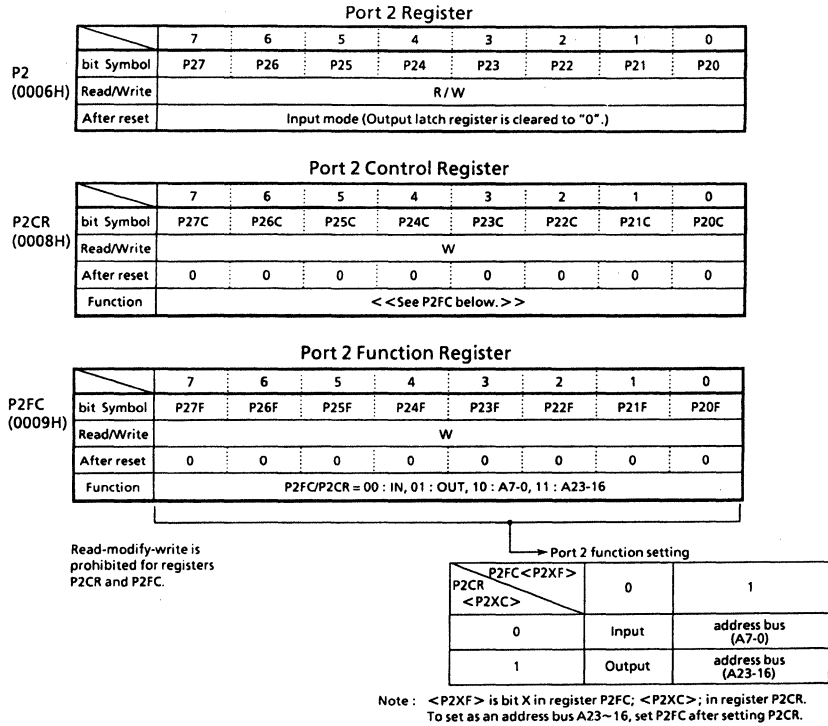


Figure 3.5 (2). Registers for Port 2



**3.5.2 Port 3 (P32 - P37)**

Port 3 is a 6-bit general-purpose I/O port.

I/O is set using control register P3CR and function register P3FC. Resetting resets all bits of output latch P3; control register P3CR (bits 0 and 1 are unused), and function register

P3FC to 0. Resetting sets P32 to P37 to input mode, and connects a pull-up resistor.

In addition to functioning as a general-purpose I/O port, Port 3 also functions as an I/O for the CPU's control/status signal.

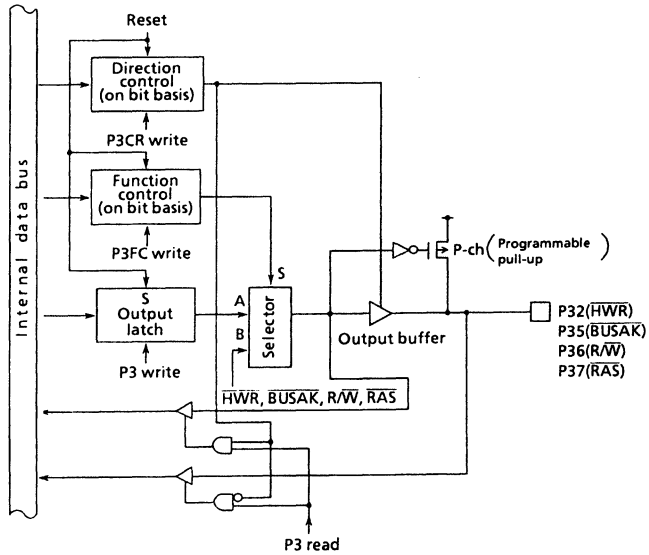


Figure 3.5 (3). Port 3 (P32, P35, P36, P37)

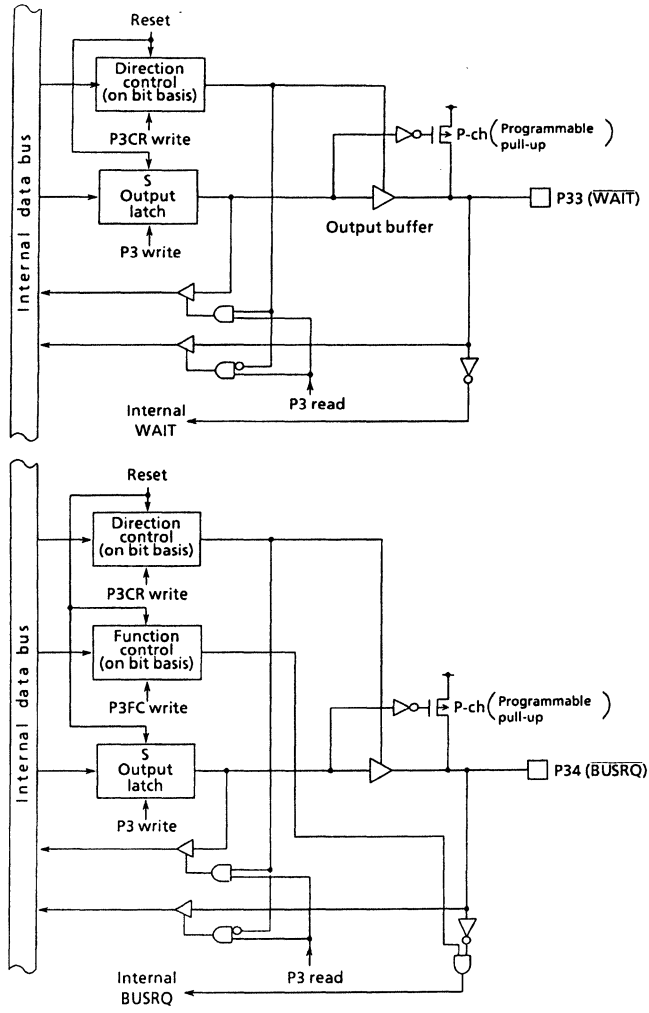
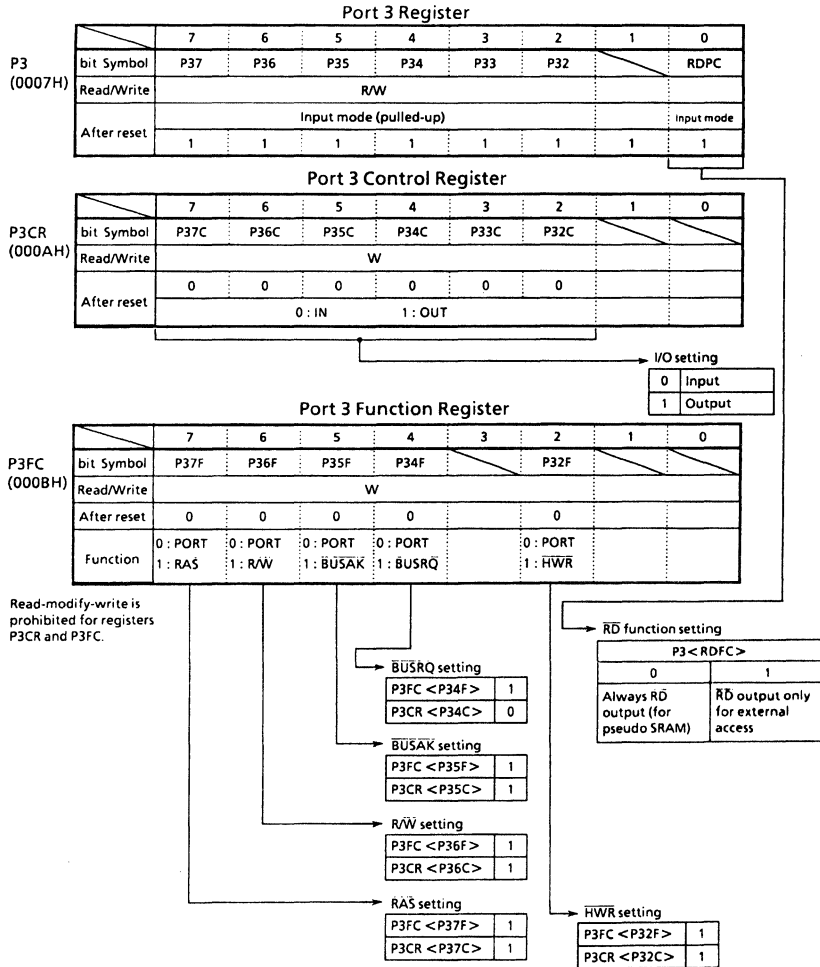


Figure 3.5 (4). Port 3 (P33, P34)



Note) When P33/WAIT pin is used as a WAIT pin, set P3CR < P33C > to '0' and bits 3 and 2 < Bnw1, 0 > of Chip Select/WAIT control register to '010', '100', '101'.

**Figure 3.5 (5). Registers for Port 3**

### 3.5.3 Port 4 (P40 - P42)

Port 4 is a 5-bit general-purpose I/O port. I/O can be set on a bit basis using control register P4CR and function register. Resetting does the following:

- Sets the P40, P41, P43 and P44 output latch registers to 1.
- Resets all bits of the P42 output latch register, the control register P4CR, and the function register P4FC to 0.
- Sets P40, P41, P43 and P44 to input mode and connects a pull-up resistor.
- Sets P42 to input mode and connects a pull-down resistor.

To disconnect the resistors, write 0 in the output latch (for pull-down).

In addition to functioning as a general-purpose I/O port, Port 4 also functions as a chip select output signal (CS0 to CS4 or CAS0 to CAS4).

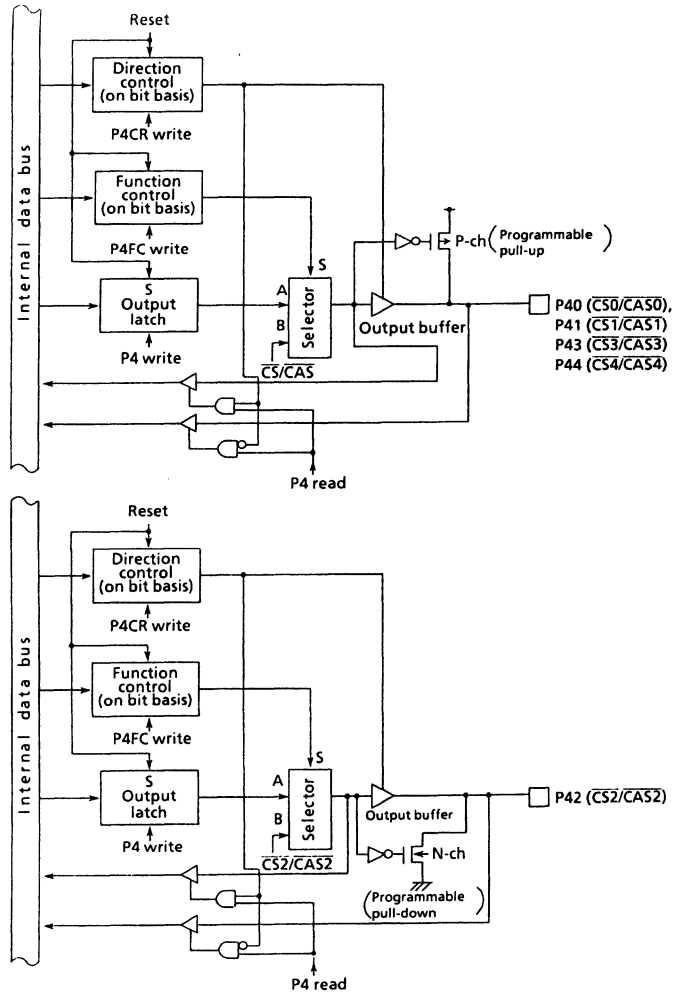
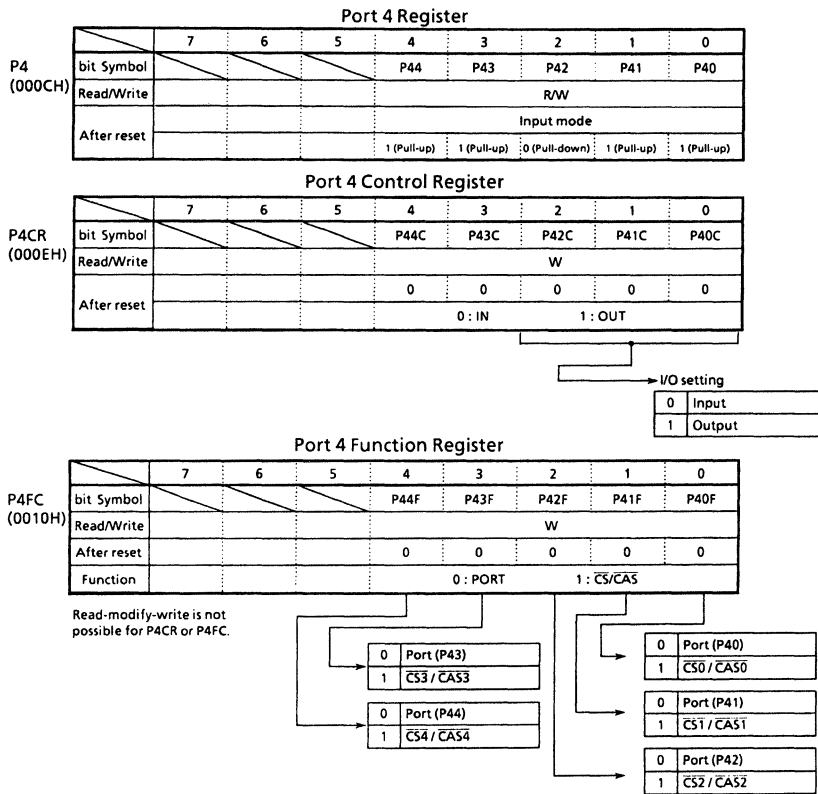


Figure 3.5 (6). Port 4

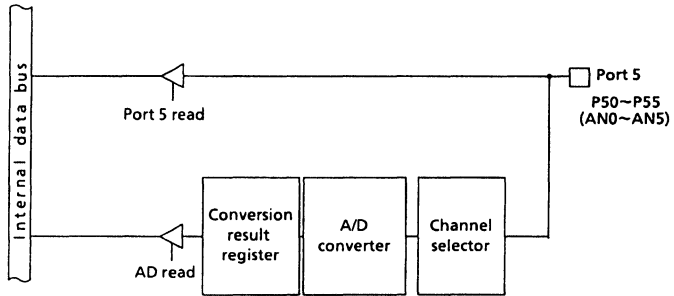


Note : To output chip select signal ( $\overline{CS0}/\overline{CAS0}$  to  $\overline{CS4}/\overline{CAS4}$ ), set the corresponding bits of the control register P4CR and the function register P4FC.  
 The B0CSL, B1CSL, B2CSL, B3CSL and B4CSL registers of the chip select / wait controller are used to select the  $\overline{CS}/\overline{CAS}$  function.

Figure 3.5 (7). Registers for Port 4

**3.5.6 Port 5 (P50 - P55)**

Port 5 is a 6-bit input port, also used as an analog input pin.



**Figure 3.5 (8). Port 5**

		Port 5 Register							
		7	6	5	4	3	2	1	0
P5 (000DH)	bit Symbol	/		P55	P54	P53	P52	P51	P50
	Read/Write	R							
	After reset	Input mode							

Note) The input channel selection of A/D Converter is set by A/D Converter mode register ADMOD2.

**Figure 3.5 (9). Registers for Port 5**

**3.5.5 Port 6 (P60 - P67)**

Port 6 is an 8-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 6 as an input port. It also sets all bits of the output latch to 1. In addition to functioning as a general-purpose I/O port, Port 6 also functions as a pattern gener-

ator PG0/PG1 output/ PG0 is assigned to P60 to P63; PG1, to P64 to P67. Writing 1 in the corresponding bit of the port 6 function register (P6FC) enables PG output. Resetting resets the function register P6FC value to 0, and sets all bits to ports.

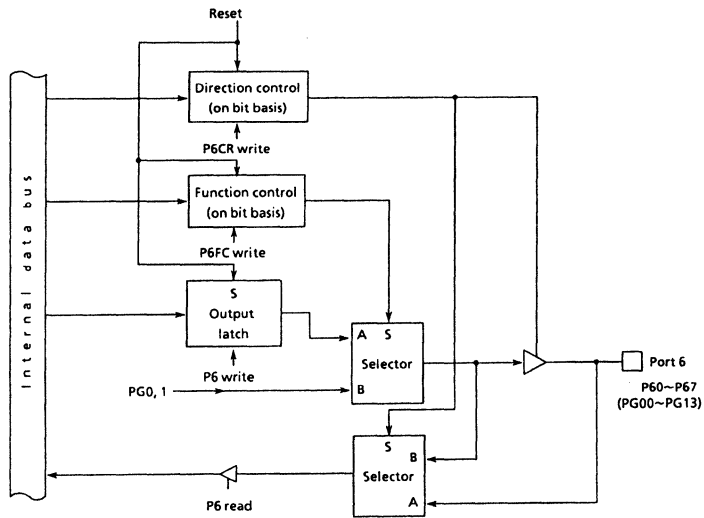


Figure 3.5 (10). Port 6



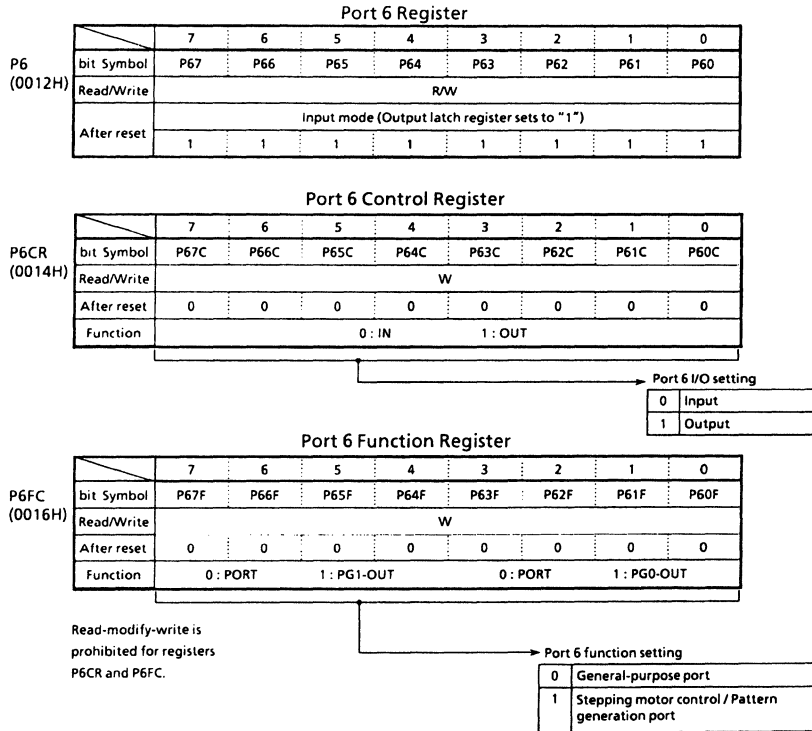


Figure 3.5 (11). Registers for Port 6

**3.5.6 Port 7 (P70 - P73)**

Port 7 is a 4-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 7 as an input port. In addition to functioning as a general-purpose I/O port, Port 70 also functions as an input clock pin T10; Port 71 as an 8-bit timer output

(TO1), Port 72 as a PWM0 output (TO2), and Port 73 as a PWM1 output (TO3) pin. Writing 1 in the corresponding bit of the Port 7 function register (P7FC) enables output of the timer. Resetting resets the function register P7FC value to 0, and sets all bits to ports.

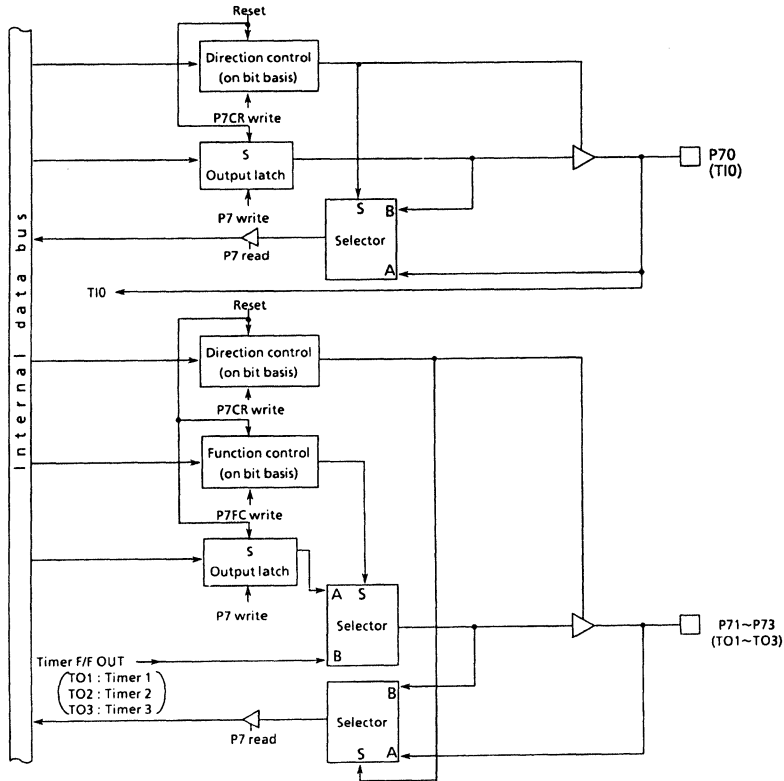
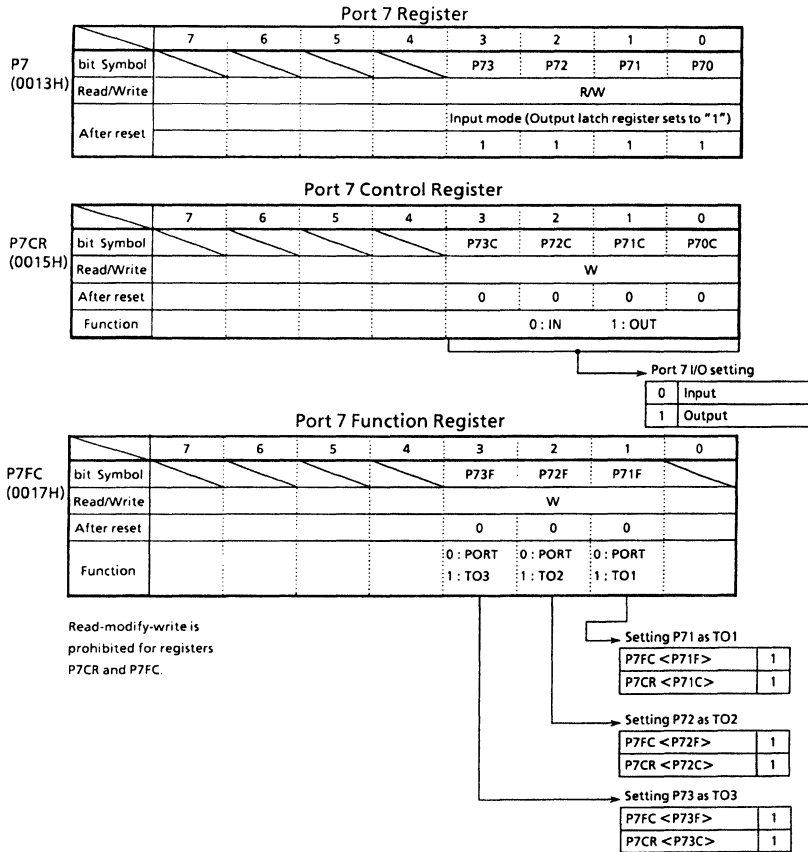


Figure 3.5 (12). Port 7



Note) P70/T10 pin does not have a register changing PORT/FUNCTION. For example, when it is used as an input port (P70), the input signal for P70 is inputted to 8 bit Timer 0 as a timer input 0(T10).

Figure 3.5 (13). Registers for Port 7

**3.5.7 Port 8 (P80 - P87)**

Port 8 is an 8-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 8 as an input port. It also sets all bits of the output latch register P8 to 1. In addition to functioning as a general-purpose I/O port, Port 8 also functions as an

input for 16-bit timer 4 and 5 clocks, an output for 16-bit timer F/F 4, 5, and 6 output, and an input for INTO. Writing 1 in the corresponding bit of the Port 8 function register (P8FC) enables those functions. Resetting resets the function register P8FC value to 0, and sets all bits to ports.

(1) P80 ~ P86

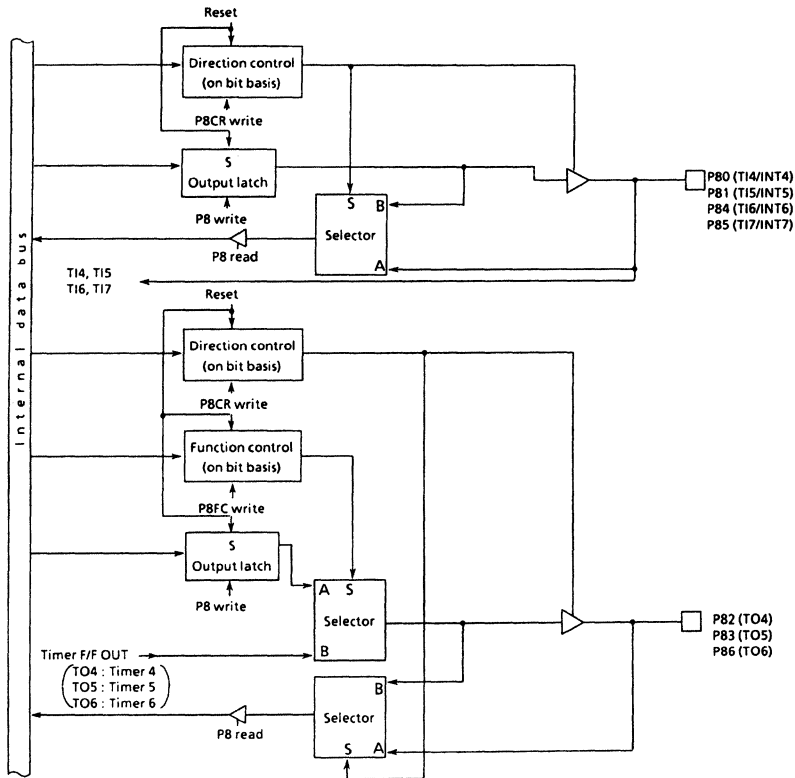


Figure 3.5 (14). Port 8 (P80 - P86)

(2) P87 (INT0)

Port 87 is a general-purpose I/O port, and also used as an INT0 pin for external interrupt request input.

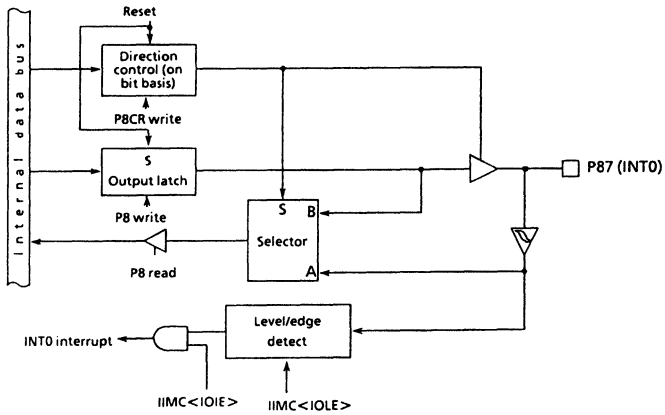
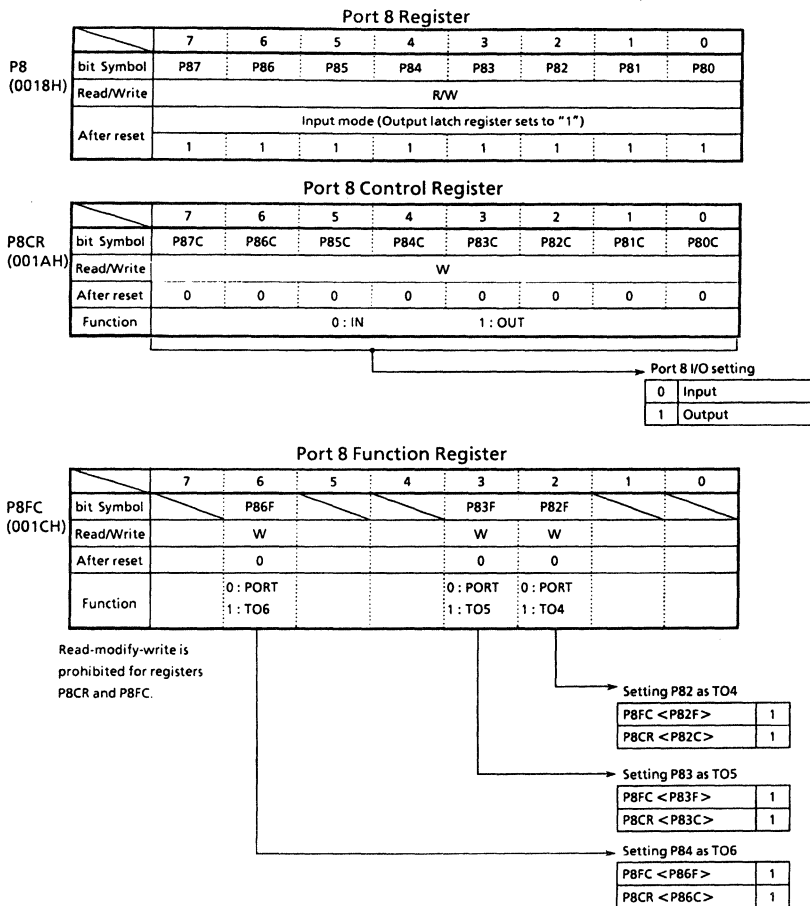


Figure 3.5 (15). Port 87



Note) P80/TI4, P81/TI5, P84/TI6, P85/TI7 pins do not have a register changing PORT/FUNCTION. Therefore this is the same as P70/TI0 pin.  
When P87/INT0 pin is used as an INT0 pin, set P8CR<P87C> to '0' and IIMC<IOIE> to '1'.

Figure 3.5 (16). Registers for Port 8

**3.5.8 Port 9 (P90 - P95)**

Port 9 is a 6-bit general-purpose I/O port. I/O can be set on bit basis.

Resetting sets Port 9 to an input port.

It also sets all bits of the output latch register to 1.

In addition to functioning as a general-purpose I/O port, Port 9 can also function as an I/O for serial channels 0 and 1. Writing 1 in the corresponding bit of the port 9 function register (P9FC) enables is function.

Resetting resets the function register value to 0, and sets all bits to ports.

(1) P80 ~ P86 (TXD0/TXD1)

Ports 90 and 93 also function as serial channel TXD output pins in addition to I/O ports.

They have a programmable open drain function.

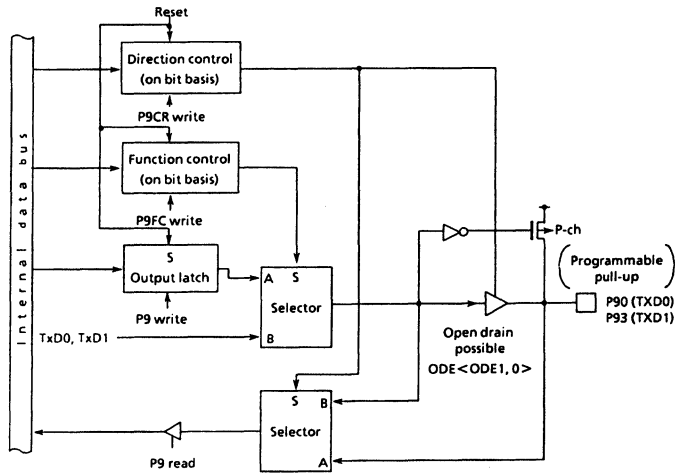


Figure 3.5 (17). Ports 90 and 93

(2) Ports 91 and 94 (RXD0, 1)

Ports 91 and 94 are I/O ports, and also used as RXD input pins for serial channels.

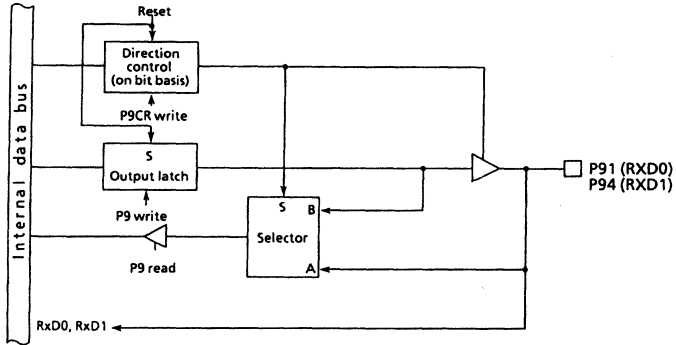


Figure 3.5 (18). Ports 91 and 94

(3) Port 92 (CTS/SCLK0)

Port 92 is an I/O port, and also used as CTS input pins for serial channels.

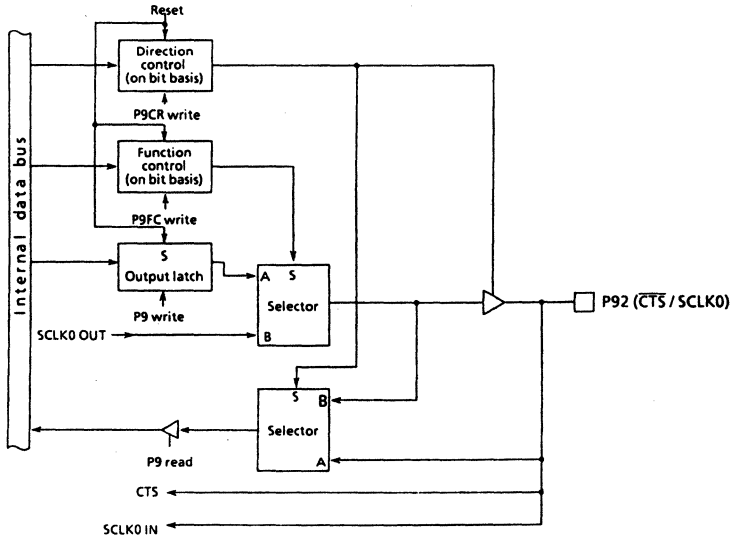


Figure 3.5 (19). Port 92



(4) Port 95 (SCLK1)

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

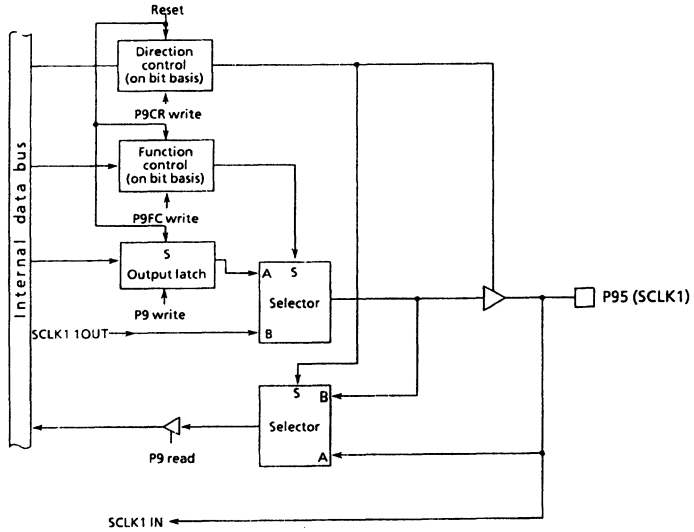


Figure 3.5 (20). Port 95

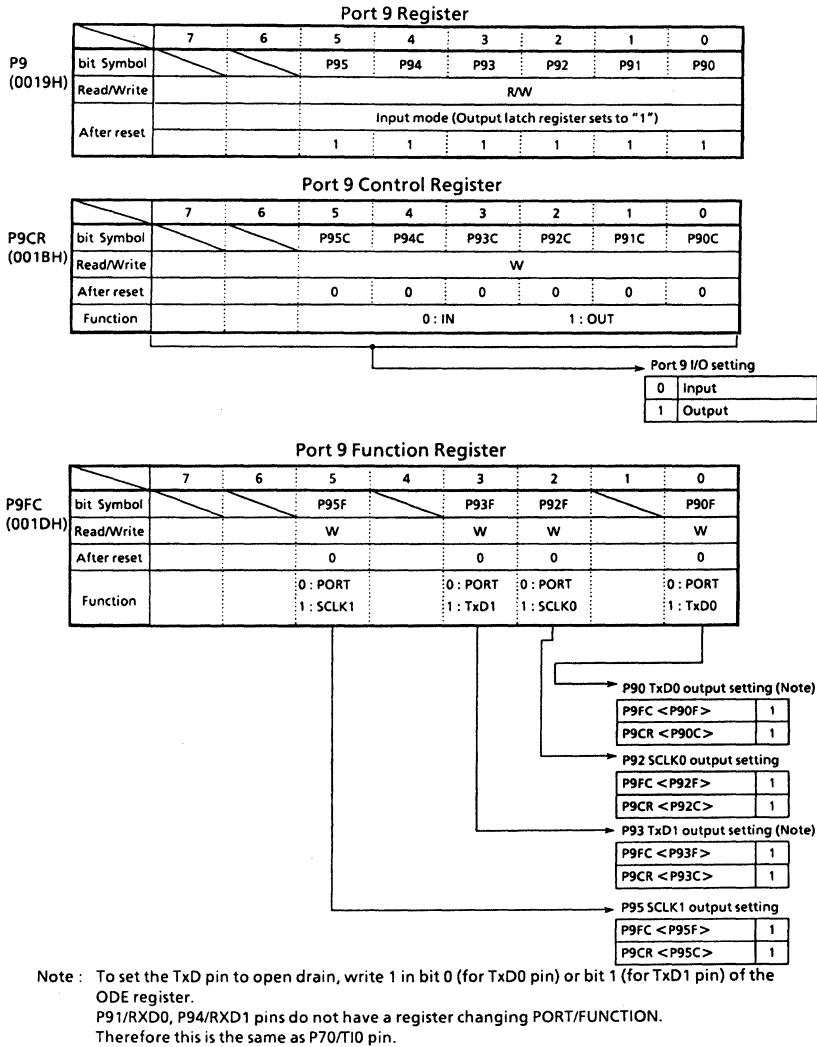


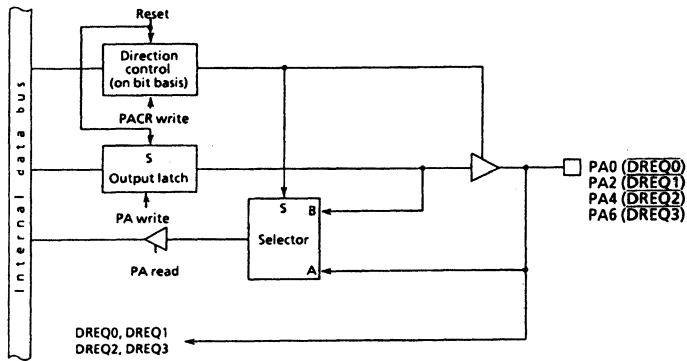
Figure 3.7 (21). Registers for Port 9

**3.5.9 Port A (PA0 - PA7)**

Port A is an 8-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port A as an input port. It also sets all bits of the output latch register PA to 1. In addition to functioning as a general-purpose I/O port, Port A also functions as an I/O for DREQ/DACK of DMAC. Writing "1" in the corresponding bit of the Port A function register (PAFC) enables those functions. Resetting resets the function register PAFC value to "0", and sets all bits to ports.

- (1) Port A0, A2, A4, A6 ( $\overline{\text{DREQ0}}$ ,  $\overline{\text{DREQ1}}$ ,  $\overline{\text{DREQ2}}$ ,  $\overline{\text{DREQ3}}$ )

Ports A0/A2/A4 and A6 also function as  $\overline{\text{DREQ}}$  input pins of DMAC in addition to I/O ports.



**Figure 3.5 (22). Ports A0, A2, A4, A6**

(2) Port A1, A3, A5, A7 ( $\overline{DACK0}$ ,  $\overline{DACK1}$ ,  $\overline{DACK2}$ ,  $\overline{DACK3}$ )

Ports A1, A3, A5 and A7 also function as  $\overline{DACK}$  output pins of DMAC in addition to I/O ports.

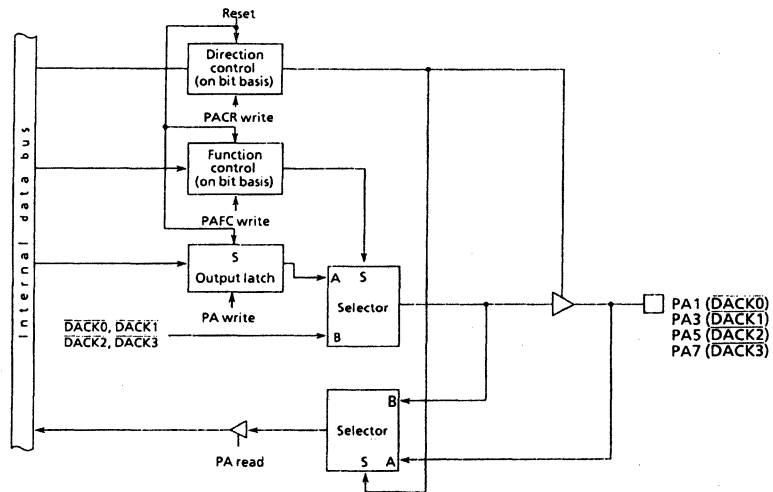
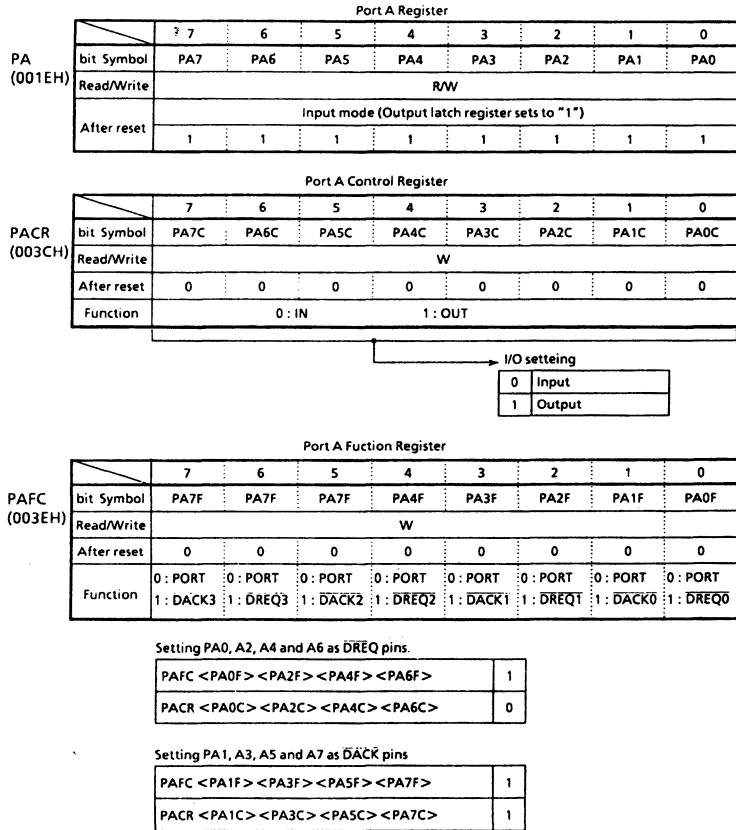


Figure 3.5 (23). Ports A1, A3, A5, A7



Note: Read-modify-write is prohibited for registers PACR and PAFC.

Figure 3.5 (24). Registers for Port A

**3.5.10 Port B (PB0 - PB4)**

Port B is a 5-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port B as an input port. It also sets all bits of the output latch register PB to 1. In addition to functioning as a general-purpose I/O port, Port B also functions as TC,  $\overline{EOP}$ , etc., of DMA. Writing "1" in the corresponding bit of the Port B function register (PBFC) enables those functions. Resetting resets the function register PBFC value to "0", and sets all bits to ports.

(1) Port B1 ( $\overline{EOP}$ )

Port B1 also functions as  $\overline{EOP}$  input pin of DMAC in addition to I/O port.

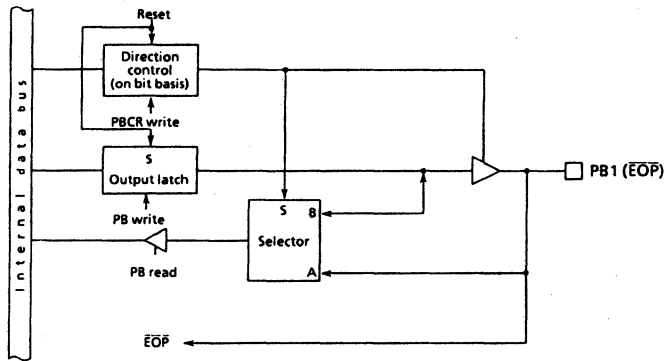


Figure 3.5 (25). Port B1

(2) Port B0, B2, B3, B4 (AEN, TC,  $\overline{\text{IORD}}$ ,  $\overline{\text{IOWR}}$ )

$\overline{\text{IORD}}$  and  $\overline{\text{IOWR}}$  pins of DMAC in addition to I/O ports.

Ports B0, B2, B3 and B4 also functions as AEN, TC,

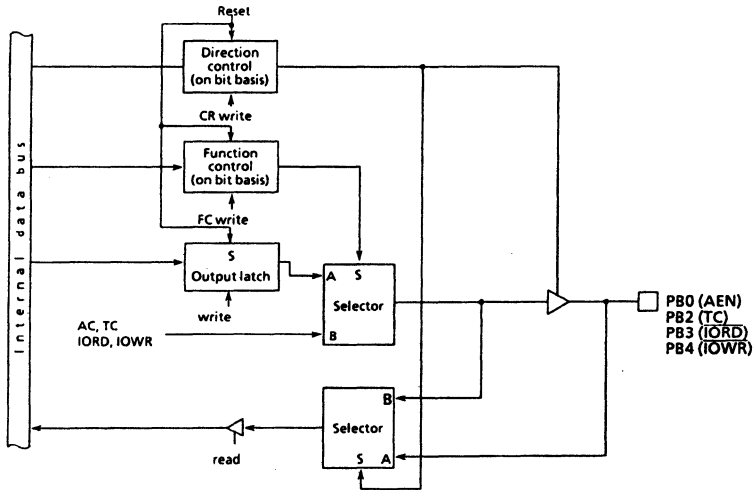


Figure 3.5 (26). Ports B0, B2, B3, B4

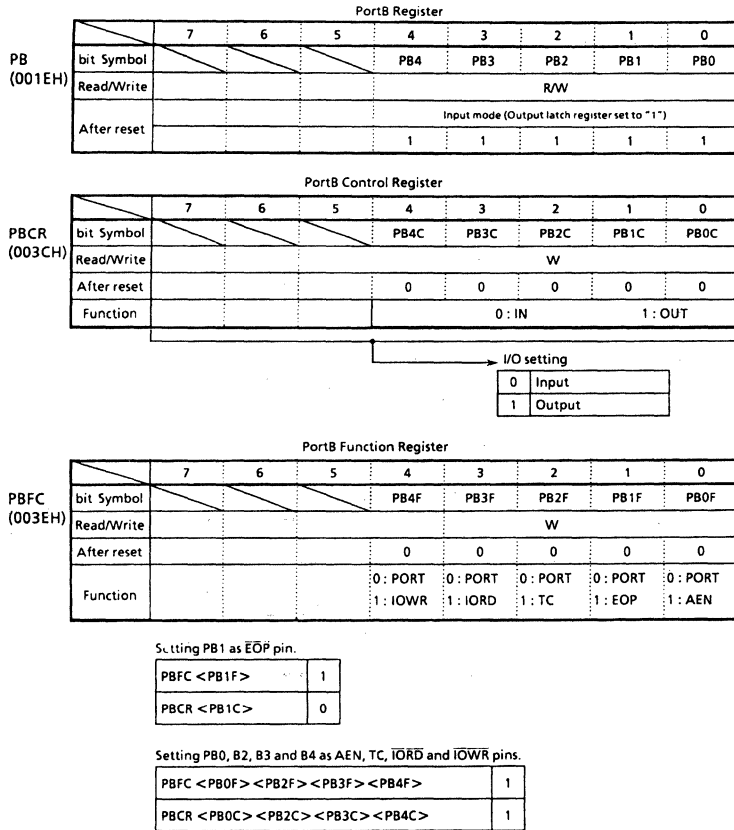


Figure 3.5 (27). Port B Register



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

The TMP96C081F has a built-in chip select/wait controller used to control chip select ( $\overline{CS0}$  -  $\overline{CS4}$  pins), wait ( $\overline{WAIT}$  pin), and data bus size (8 or 16 bits) for any of the five block address areas.

And there is an AM8/ $\overline{16}$  which selects external data width for TMP96C081.

#### 3.6.1 AM8/ $\overline{16}$ pin

- (1) (1-1) 16-bit bus width internalized with 8 bit width or fixed 16 bit bus

Set this pin to "0". Then AD8 - 15 or A8 - 15 are fixed to AD8 - 15 function compulsorily.

The bus width when the CPU accesses an external area is set by Chip Select/Wait Control Register described at 3.6.2.

However, the bus width of program memory only after reset must be 16 bit bus width in this case.

- (1-2) Fixed 8-bit bus width

Set this pin to "1". Then AD8 - 15 or A8 - 15 function compulsorily.

The value of bit 4: <B0BUS>, <B1BUS>, <B2BUS>, <B3BUS> or <B4BUS> described at 3.6.2 are ignored and the bus width is fixed to 8 bit.

#### 3.6.2 Control Registers

Table 3.6 (1) shows control registers

One block address areas are controlled by 1-byte CS/WAIT control register (B0CSL to B4CSL and B0CSH and B4CSH). These registers can be written to only when the CPU is in system mode.

- (1) Enable

Control register bit 7 <B0E to B3E> is a master bit used to specify enable ("1")/disable ("0") of the setting. Resetting sets B0E, B1E, B3E and B4E to disable ("0") and B2E to enable ("1").

- (2) System only specification

Control register bit 7 (B0SYS, B1SYS, B2SYS, and B4E) 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)  $\overline{CS}/\overline{CAS}$  Waveform select

Control register bit 5 (B0CAS to B4CAS) is used to specify waveform mode output from the chip select pin ( $\overline{CS0}/\overline{CAS0}$ - $\overline{CS4}/\overline{CAS4}$ ). Setting this bit to 0 specifies  $\overline{CS0}$  to  $\overline{CS4}$  waveforms; setting it to 1 specifies  $\overline{CAS0}$  to  $\overline{CAS4}$  waveforms.

Resetting clears bit 5 to 0.

- (4) Data bus width select

Bit 4 (B0BUS to B4BUS) of the control register is used to specify data bus size. Setting this bit to 0 accesses in the memory in 16-bit data bus mode; setting it to "1", memory is accessed in 8-bit data bus mode.

Changing data bus depending on the access address is called dynamic bus sizing. Table 3.6 (2) shows the details of the bus operation.

This bit is changed by the state of AM8/ $\overline{16}$  pin.

- (5) Wait control

Control register bits 2 to 0 (B0W2, 1, 0 to B4W2, 1, 0) are used to specify the number of waits. Setting these bits to 010 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 these bits to 100 or 101 inserts a 0-2-state wait 0-1-state wait according to the  $\overline{WAIT}$  pin status.

Resetting sets these bits to 000 (2-state wait mode).

- (6) Address area specification

Bits 3 to 0 in the B0CSH to B4CSH control register are used to specify the corresponding address areas. If 0000 is set in those four bits (same value after reset), the setting is enable and low strobe signal is output from the chip select pins ( $\overline{CS0}/\overline{CAS0}$  to  $\overline{CS4}/\overline{CAS4}$ ) when the following address are accessed; 7F00H to 7FFFH by  $\overline{CS0}$ , 0080H to 7F00H to 7FFFH by  $\overline{CS1}$  and 8000H to 3FFFFFFH by  $\overline{CS2}$ . If other than 0000 is set in those four bits, the chip select pin compares the setting with that in A23 to A20. If the setting match, the chip select pin outputs a low strobe signal. Bit 7 to 4 are used to mask bits 3 to 0. Setting 1 in the four bits compares address A23 to A20 with the masked address bits and detects match. Setting 1 in the four bits determines match regardless of setting.

Unlike  $\overline{CS0}$  to  $\overline{CS2}$ ,  $\overline{CS3}$  and  $\overline{CS4}$  do not have an address area assigned in advance after reset. Thus address area of 1M byte can be freely set using B3CSH or B4CSH.

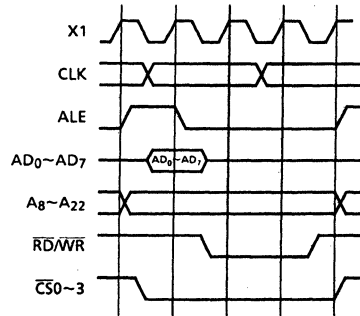


Figure 3.6 (1). Chip select ( $\overline{CS0} \sim \overline{CS4}$ )

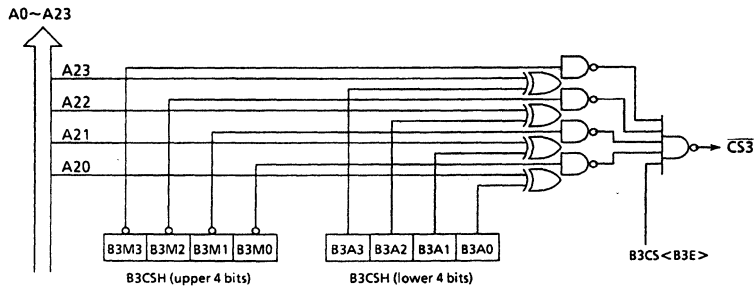


Figure 3.6 (2).  $\overline{CS3}$  Address Decoder Block Diagram

Table 3.6 (3) Chip Select/Wait Control Register (1/2)

Code	Name	Address	B7	B6	B5	B4	B3	B2	B1	B0
B0CSL	Block0 CS/WAIT Control Register Low	68H	B0E	B0SY5	B0CAS	B0BUS	—	B0W2	B0W1	B0W0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System only	0:CS0 1:CA50	0:16bit Bus 1:8bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B0CSH	Block0 CS/WAIT Control Register High	69H	B0M3	B0M2	B0M1	B0M0	B0A3	B0A2	B0A1	B0A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparison 0000 : 7F00H~7FFFH other : Compare A23~A20			
B1CSL	Block1 CS/WAIT Control Register Low	6AH	B1E	B1SY5	B1CAS	B1BUS	—	B1W2	B1W1	B1W0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System only	0:CS1 1:CA51	0:16bit Bus 1:8bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B1CSH	Block1 CS/WAIT Control Register High	6BH	B1M3	B1M2	B1M1	B1M0	B1A3	B1A2	B1A1	B1A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparison 0000 : 0080H~7FFFH other : Compare A23~A20			
B2CSL	Block2 CS/WAIT Control Register Low	6CH	B2E	B2SY5	B2CAS	B2BUS	—	B2W2	B2W1	B2W0
			W	W	W	W		W	W	W
			1	0	0	0		0	0	0
			1: CS/CAS Enable	1: System only	0:CS2 1:CA52	0:16bit Bus 1:8bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B2CSH	Block2 CS/WAIT Control Register High	6DH	B2M3	B2M2	B2M1	B2M0	B2A3	B2A2	B2A1	B2A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparison 0000 : 8000H~3FFFFH other : Compare A23~A20			

Table 3.6 (3) Chip Select/Wait Control Register (2/2)

Code	Name	Address	B7	B6	B5	B4	B3	B2	B1	B0
B3CSL	Block3 CS/WAIT Control Register Low	6EH	B3E	B3SYS	B3CAS	B3BUS	—	B3W2	B3W1	B3W0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System only	0:CS4 1:CA53	0:16bit Bus 1:8bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B3CSH	Block3 CS/WAIT Control Register High	6FH	B3M3	B3M2	B3M1	B3M0	B3A3	B3A2	B3A1	B3A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparison 0000 : Address areas are not specified other : Compare A23~A20			
B4CSL	Block4 CS/WAIT Control Register Low	4AH	B4E	B4SYS	B4CAS	B4BUS	—	B4W2	B4W1	B4W0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System only	0:CS4 1:CA54	0:16bit Bus 1:8bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B4CSH	Block4 CS/WAIT Control Register High	4BH	B4M3	B4M2	B4M1	B4M0	B4A3	B4A2	B4A1	B4A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparison 0000 : Address areas are not specified other : Compare A23~A20			

Note : Only block 2 is enable the CS/CAS after reset.

Table 3.6 (4) Dynamic Bus Sizing

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

### 3.6.3 Chip Select Image

Address areas from 0100H to FFFFFFFH can be specified for 1M, 2M, 4M, 8M, 16M bytes.

The chip select controller compares an address on a bus and values of bit 3 to 0 of B0CSH to B4CSH at every cycle. When the values of bit 3 to 0 are "0000", the chip select controller compares the address and the address value which has already been defined to each block.

If the result of the comparison matches, the specified address area is assumed to be accessed. If channels B0E to B4E are set to enable, a chip select pin ( $\overline{CS0}$  to  $\overline{CS4}$ ) corresponding to these channels outputs a low strobe signal.

When comparing addresses, a compared result of a particular address can be ignored by setting the values of bit 7 to 4 of B0CSH to B4CSH. Consequently, an address area size can be specified.

An image of the actual chip select is shown below. After reset (when bits 3 to 0 of B0CSH to B4CSH set 0000), 7F00H to 7FFFH is specified for  $\overline{CS0}$ ; 0080H to 7FFFH, for  $\overline{CS1}$ ; 8000H to 3FFFFFFFH, for  $\overline{CS2}$ .

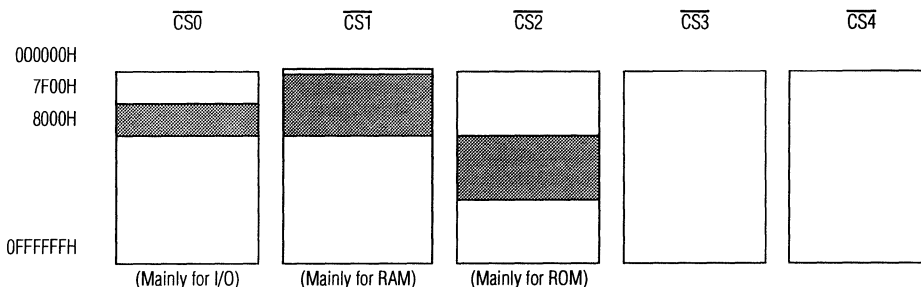
The reason is that a device other than ROM (i.e., RAM or I/O) might be connected externally.

The addresses 7F00H to 7FFFH (256 bytes) for  $\overline{CS0}$  are mapped mainly for possible expansions to external I/O.

The addresses 0100H to 7FFFH (approx. 32K bytes) for  $\overline{CS1}$  0 are mapped there mainly for possible extensions to external RAM.

The addresses 8000H to 3FFFFFFFH (approx. 4M bytes) for  $\overline{CS2}$  are mapped mainly for possible extensions to external ROM. After reset,  $\overline{CS2}$  is enable in 16-bit bus and 2-wait. The program is externally read at address.

The address 8000H in this setting (16-bit bus, 2-wait). (If AM8/16 pin is "1",  $\overline{CS2}$  is enable in 8-bit bus.)



- Note 1: Access priority is built-in I/O then the chip select controller.
- Note 2: External areas other than  $\overline{CS0}$  to  $\overline{CS4}$  are accessed in 16-bit data bus (0 wait) mode.

When using the chip select/wait controller and specifying the same address area more than once, the priority is  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS2}$ ,  $\overline{CS3}$ ,  $\overline{CS3}$ , and  $\overline{CS4}$ . (However, when address 7F00H to 7FFFH for  $\overline{CS0}$  and 100H to 7FFFH for  $\overline{CS1}$  are specified, in other words, specification overlap, only the  $\overline{CS0}$  setting pin is active.)

- Note 3: When the bus is released ( $\overline{BUSAK} = "0"$ ),  $\overline{CS0}$  -  $\overline{CS4}$  pins are also (the output buffer is OFF). Refer to [ Note about the bus release ] in 3.5 Function of Ports about the state of pins.

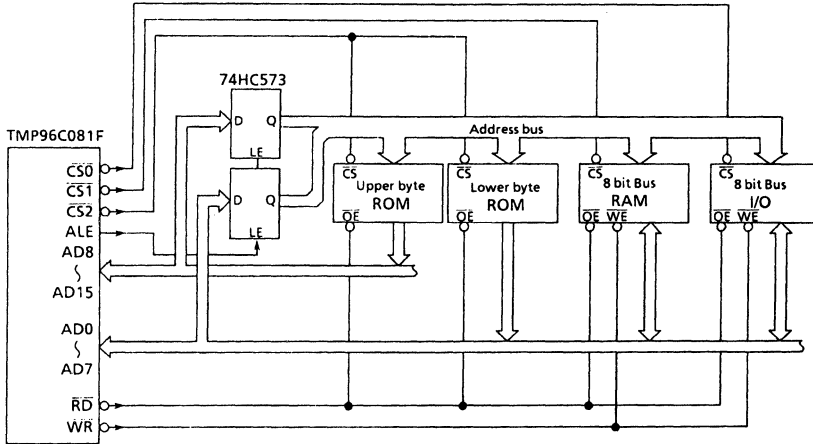
Table 3.6 (5) Specified the address areas of B3CSH

B3CSH		Specified the Address Areas	Area Size
0000	0001	100000H ~ 1FFFFFFH	1M Bytes
0001	0011	200000H ~ 3FFFFFFH	2M Bytes
0011	0111	400000H ~ 7FFFFFFH	4M Bytes
0111	1000	800000H ~ FFFFFFFH	8M Bytes
1111	1000	100H ~ 1FFFFFFH	16M Bytes

**3.6.4 Example of Usage**

Figure 3.6 (6) is an example in which an external memory is connected to the TMP96C081F. In this example, a ROM is

connected using 16 bit Bus; a RAM and an I/O are connected using 8 bit Bus.



**Figure 3.6 (6). Example of External Memory Connection (ROM = 16 bits, RAM and I/O = 8 bits)**

Resetting sets pins  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS2}$ ,  $\overline{CS3}$  to  $\overline{CS4}$  to input port mode.  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS3}$ , and  $\overline{CS4}$  are set high due to an internal pull-up resistor;  $\overline{CS2}$ , low due to an internal

pull-down resistor. The program used to set these pins is as follows:

```

P4CR EQU 0EH
P4FC EQU 10H
B0CSL EQU 68H
B1CSL EQU 6AH
B2CSL EQU 6CH
LD (BOCS), 0090H ; CS0 = 8 bits, 2WAIT, 7F00H ~ 7FFFH
LD (B1CS), 0093H ; CS1 = 8 bits, 0WAIT, 0080H ~ 7FFFH
LD (B2CS), 0081H ; CS2 = 16 bits, 1WAIT, 8000H ~ 3FFFFFFH
LD (P4CR), 07H
LD (P4FC), 07H
    
```

)  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS2}$  output mode setting

(2) Example Usage-2

Figure 3.6 (7) is an example in which an external memory is

connected to the TMP96C081F. In this example, a ROM, a RAM, and I/O are connected using 8 bit Bus.

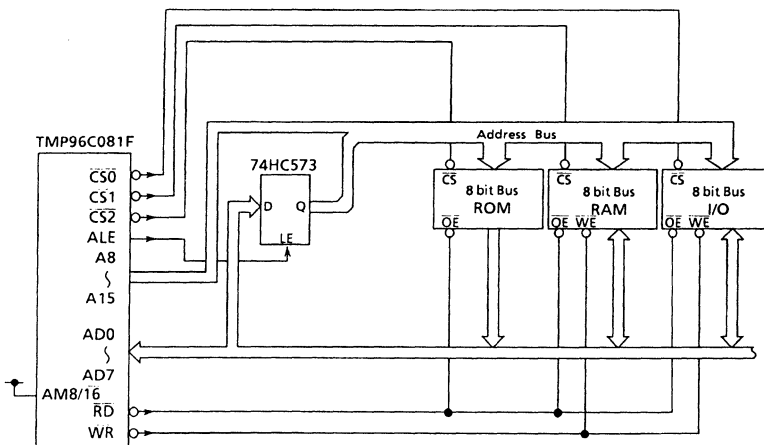


Figure 3.6 (7). Example of External Memory Connection (ROM and RAM and I/O = 8bits)

Resetting sets pins  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS2}$ ,  $\overline{CS3}$  and  $\overline{CS4}$  to input port mode.  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS3}$ , and  $\overline{CS4}$  are set high due to an internal pull-up resistor;  $\overline{CS2}$ , low due to an internal

pull-down resistor. The program used to set these pins is as follows:

```

P4CR EQU    0EH
P4FC EQU    10H
B0CSL EQU   68H
B0CSH EQU   69H
B1CSL EQU   6AH
B1CSH EQU   6BH
B1CSL EQU   6CH
B2CSH EQU   6DH

LD  (B0CSL), 1X010000B ) CS0 = 8bits, 2WAIT, 7F00H~7FFFH
LD  (B0CSH), 00000000B
LD  (B1CSL), 1X010011B ) CS1 = 8bits, 0WAIT, 80H~7FFFH
LD  (B1CSH), 00000000B
LD  (B2CSL), 1X000001B ) CS2 = 16bits, 1WAIT, 8000H~3FFFFFFH
LD  (B2CSH), 00000000B

LD  (P4CR),  XXXXX111B }  $\overline{CS0}$ ,  $\overline{CS1}$ ,  $\overline{CS2}$  output mode setting
LD  (P4FC),  XXXXX111B

```

Note) X : don't care

When  $\overline{AM8/16}$  Pin is set to high level, all address area are fixed to 8-bit data bus regardless of CS area statuses.



**3.6.5 How to Start with an 8-Bit Data Bus**  
 (with AM8/16 pin sets "0")

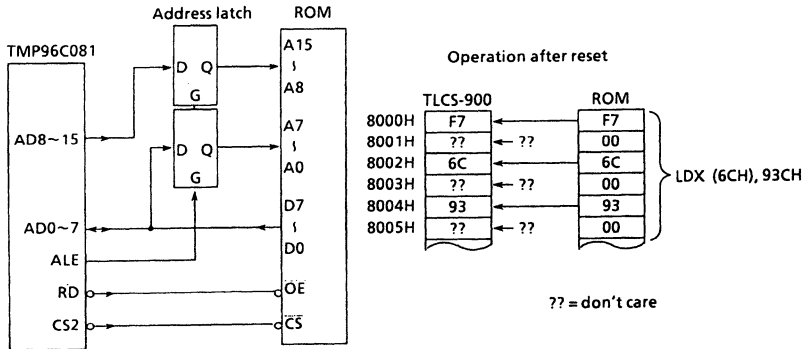
Resetting sets the  $\overline{CS2}$  pin low due to an internal pull-down resistor; memory access starts in 16-bit data bus (2-wait) mode. To start in 8-bit data bus mode, a special operation is required. Operation is as described in the example below:

```

B2CS EQU 6AH ; CS2 register address
ORG 8000H ; RESET address
LDX (B2CSL), 9CH ; CS2 8bit, 0WAIT, 8000H ~
    
```

After reset, the program reads the LDX (B2CSL), 93H 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 - 3FFFFFFH) is accessed in 8-bit data bus mode without any problem. The above program does not include setting the P4/ $\overline{CS2}$  pin to output; add a program to set the P4CR and P4FC registers as required.



### 3.7 8-bit Timers

The TMP96C081F contains two 8-bit timers (timers 0 and 1), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timer. The following four operating modes are provided for the 8-bit timers.

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

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

Each interval timer consists of an 8-bit up counter, 8-bit comparator, and 8-bit timer register. Besides, one timer flip-flop (TFF1) is provided for timer 0 and timer 1

Among the input clock sources for the interval timers, the internal clocks of  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$ , and  $\phi T256$  are obtained from the 9-bit prescaler shown in Figure 3.7 (2).

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

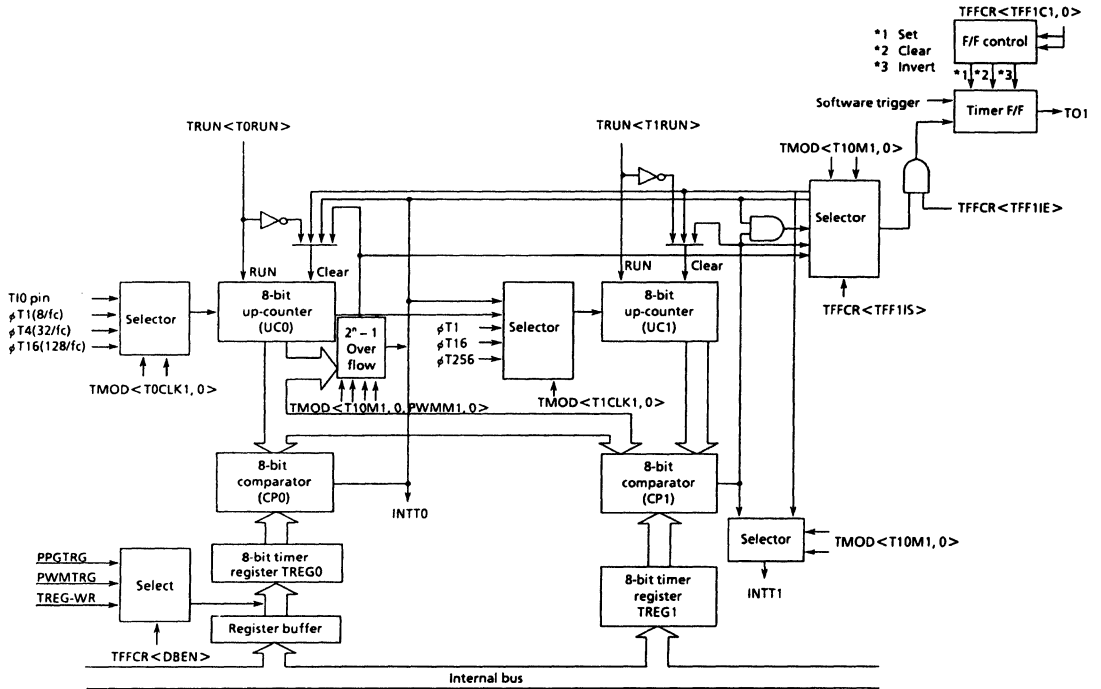


Figure 3.7 (1). Block Diagram of 8-Bit Timers (Timers 0 and 1)

① Prescaler

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 ( $f_c$ ) after it has been divided by 4 ( $f_c/4$ ).

Among them, 8-bit timer uses 4 types of clock:

$\phi T1$ ,  $\phi T4$ ,  $\phi T16$ , and  $\phi T256$ .

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	$f_c$
$\phi T1$ ( $8/f_c$ )	20MHz
$\phi T4$ ( $32/f_c$ )	$0.4\mu s$
$\phi T16$ ( $128/f_c$ )	$1.6\mu s$
$\phi T256$ ( $2048/f_c$ )	$6.4\mu s$
	$102\mu s$

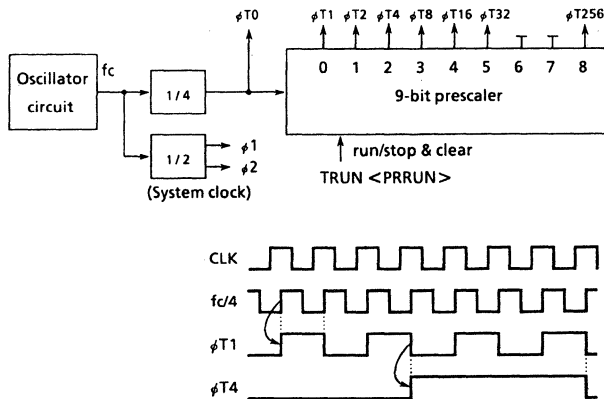


Figure 3.7 (2). Prescaler

## ② Up-counter

This is an 8-bit binary counter which counts up by the input clock pulse specified by TMOD.

The input clock of timer 0 is selected from the external clock T10 pin and three internal clocks  $\phi T1$  (8/fc),  $\phi T4$  (32/fc), and  $\phi T16$  (128/fc), according to the set value of TMOD register.

The input clock of timer 1 differs depending on the operation mode. When set in 16-bit timer mode, the overflow output of timer 0 is used as the input clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks  $\phi T1$  (8/fc),  $\phi T16$  (128/fc), and  $\phi T256$  (2048/fc) as well as the comparator output (match detection signal) of timer 0 according to the set value of TMOD register.

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

When TMOD <T10M1,0> = 00 and TMOD <T1CLK1,0> = 01,  $\phi T1$  (8/fc) becomes the input of timer 1 (8bit timer mode).

Operation mode is also set by TMOD register. When reset, it is initialized to TMOD <T01M1, 0> = 00, whereby the up-counter is placed in the 8-bit timer mode.

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

## ③ Timer register

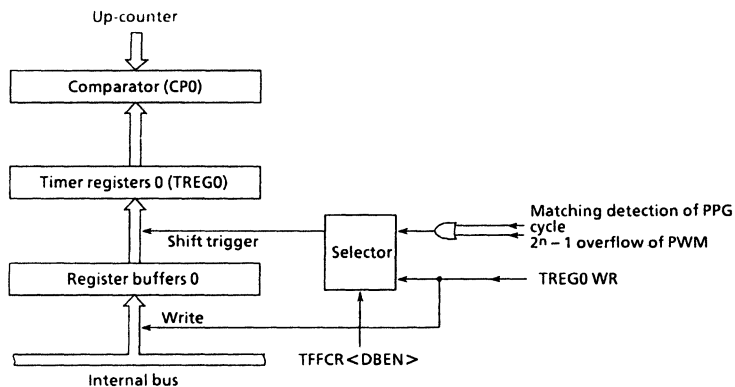
This is an 8-bit register for setting an interval time. When the set value of timer registers TREG0, TREG1, matches the value of up-counter, the comparator match detect signal becomes active. If the set value is 00H, this signal becomes active when the up-counter overflows.

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

The timer flip-flop control register TFFCR <DBEN> bit controls whether the double buffer structure in the TREG0 should be enabled or disabled. It is disabled when <DBEN> = 0 and enabled when they are set to 1.

In the condition of double buffer enable state, the data is 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 <DBEN> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <DBEN> to 1, and write the following data in the register buffer.



**Figure 3.7 (3). Configuration of Timer Register 0**

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

The memory address of each timer register is as follows.

TREG0: 000022H  
TREG1: 000023H

All registers are write-only and cannot be read.

#### ④ Comparator

A comparator compares the value in the up-counter with the values to which the timer register is set. When they match, the up-counter is cleared to zero and an

interrupt signal (INTT0, INTT1) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

#### ⑤ Timer flip-flop (timer F/F: TFF1)

The status of the timer flip-flop is inverted by the match detect signal (comparator output) of each interval timer and the value can be output to the timer output pins TO1 (also used as P71).

A timer F/F is provided for a pair of timer 0 and timer 1 and is called TFF1. TFF1 is output to TO1 pin.

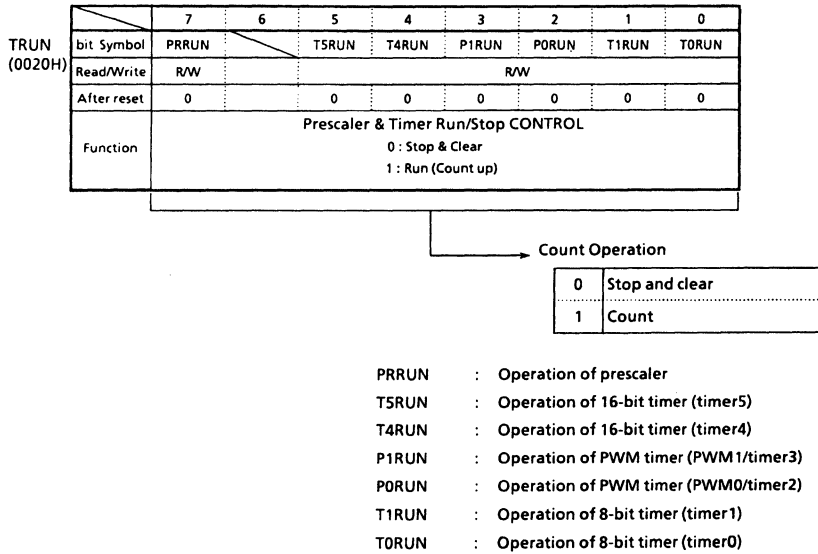


Figure 3.7 (4). Timer Operation Control Register (TRUN)

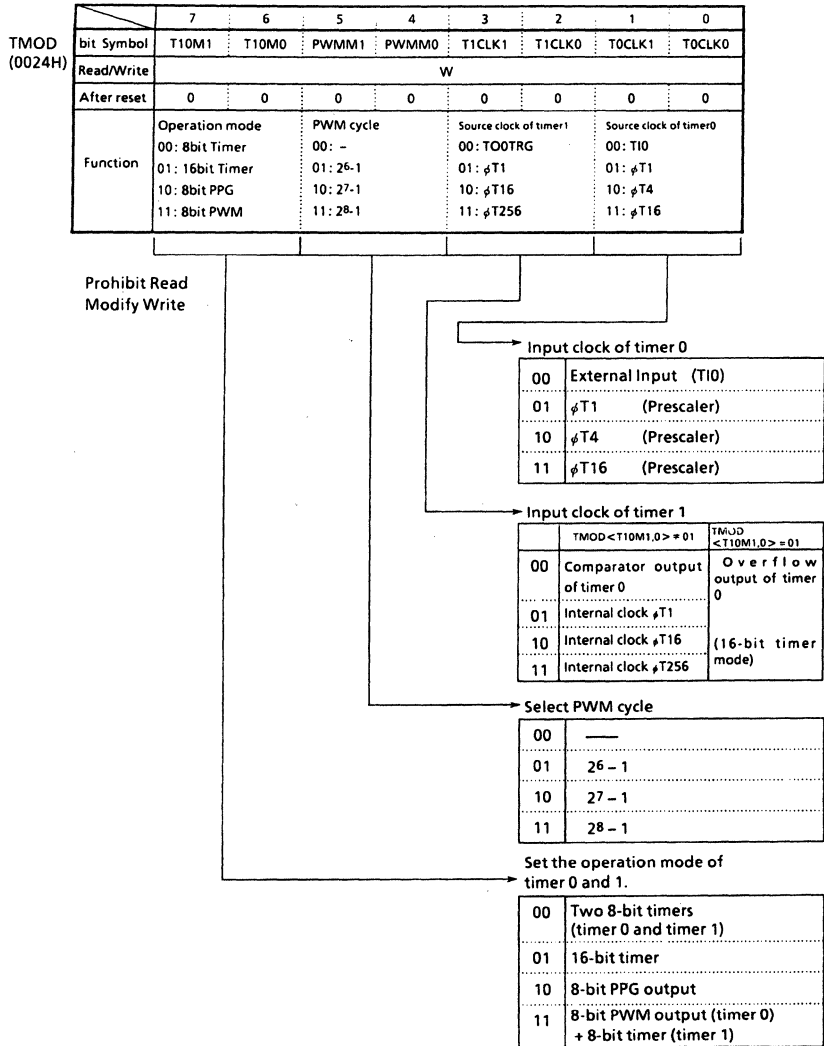


Figure 3.7 (5). Timer Mode Control Register (TMOD)



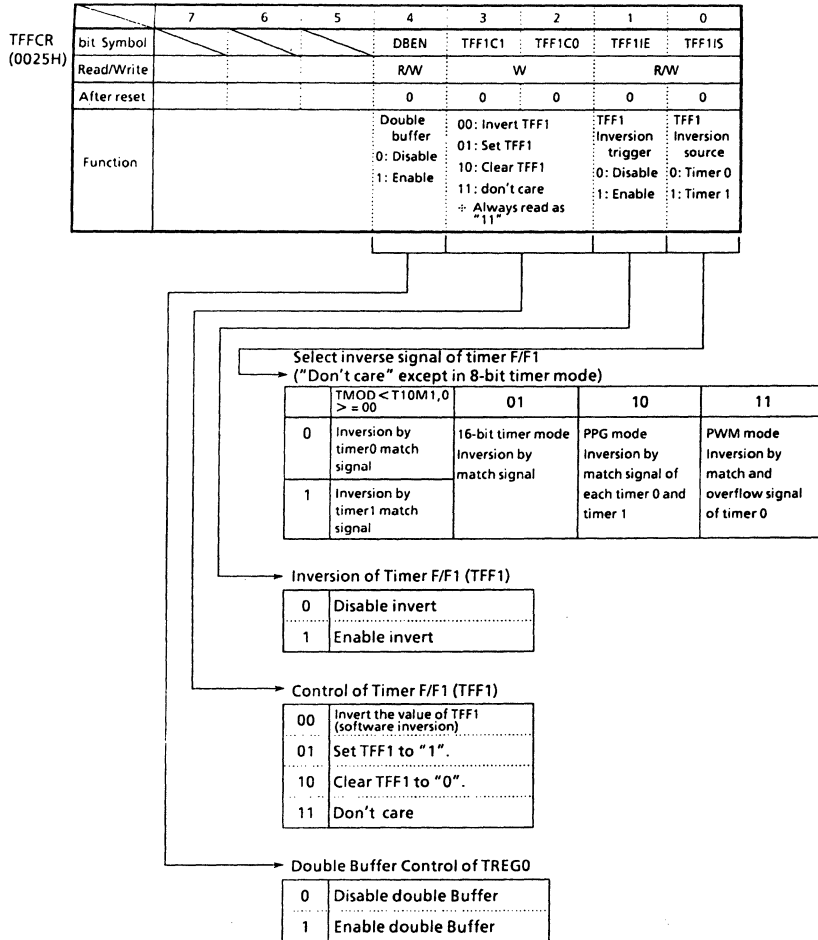


Figure 3.7 (6). Timer Flip-Flop Control Register (TFFCR)

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

(1) 8-bit timer mode

Two interval timers 0, 1, 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 TMOD 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  $f_c = 20\text{MHz}$ , set each register in the following manner.

		MSB							LSB	
		7	6	5	4	3	2	1	0	
TRUN	←	-	x	-	-	-	-	0	-	
TMOD	←	0	0	x	x	0	1	-	-	
TREG1	←	0	1	0	1	0	0	0	0	
INTET10	←	1	1	0	1	-	-	-	-	
TRUN	←	1	x	1	-	-	-	1	-	

Note: x; don't care      -; no change

- Stop timer 1, and clear it to "0".
- Set the 8-bit timer mode, and select  $\phi T1$ . ( $0.4\mu\text{s}$  @  $f_c = 20\text{MHz}$ ) as the input clock
- Set the timer register at  $40\mu\text{s}$   $\phi T1 = 100\text{H}$  (64H).
- Enable INTT1, and set it to "Level 5".
- Start timer 1 counting.

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 $f_c = 20\text{MHz}$ )	Resolution
$\phi T1$ (8/ $f_c$ )	0.4 $\mu\text{s}$ ~ 102.4 $\mu\text{s}$	0.4 $\mu\text{s}$
$\phi T4$ (32/ $f_c$ )	1.6 $\mu\text{s}$ ~ 409.6 $\mu\text{s}$	1.6 $\mu\text{s}$
$\phi T16$ (128/ $f_c$ )	6.4 $\mu\text{s}$ ~ 1.638ms	6.4 $\mu\text{s}$
$\phi T256$ (2048/ $f_c$ )	102.4 $\mu\text{s}$ ~ 2.621ms	102.4 $\mu\text{s}$

Note: The input of timer 0 and timer 1 are different from as follows:

Timer 0: T10 input,  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$

Timer 1: match output of Timer 0,  $\phi T1$ ,  $\phi T16$ ,  $\phi T256$

② Generating a 50% duty square wave pulse

The timer flip-flop (TFF1) is inverted at constant intervals, and its status is output to timer output pin (TO1).

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

	MSB				LSB			
	7	6	5	4	3	2	1	0
TRUN	← -	x	-	-	-	-	0	-
TMOD	← 0	0	x	x	0	1	-	-
TREG1	← 0	0	0	0	0	0	1	1
TFFCR	← -	-	-	-	1	0	1	1
P7CR	← x	x	x	x	-	-	1	-
P7FC	← x	x	x	x	-	-	1	x
TRUN	← 1	x	-	-	-	-	1	-

Note: x; don't care      -: no change

- Stop timer 1, and clear it to "0".
- Set the 8-bit timer mode, and select  $\phi T1$  as the input clock.
- Set the timer register at  $3.0\mu\text{s} + \phi T1 + 2 = 3$ .
- Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.
- Select P71 as TO1 pin.
- Start timer 1 counting.

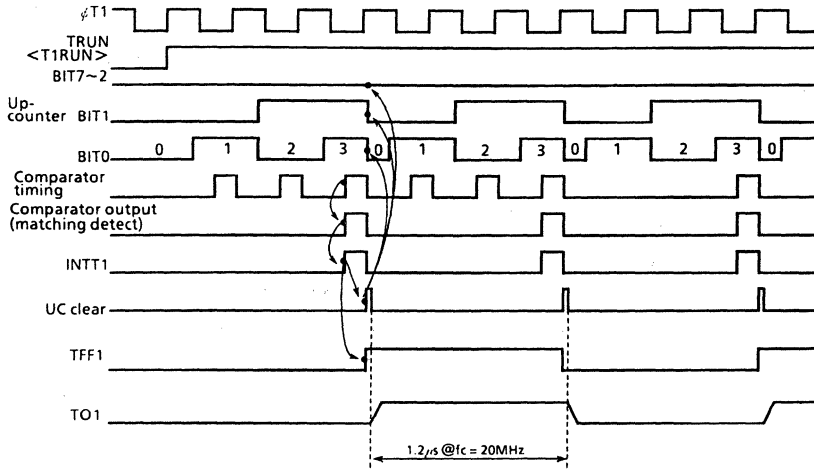


Figure 3.7 (7). Square Wave (50% Duty) Output Timing Chart

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

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

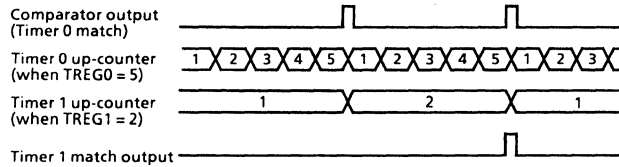


Figure 3.7 (8). Timer 1 Count Up by Timer 0

- ④ Output inversion with software

The value of timer flip-flop (TFF1) can be inverted, independent of timer operation.

Writing "00" into TFFCR <TFF1C1,0> (memory address: 000025H of bit 3 and bit 2) inverts the value of TFF1.

- 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 using the pair of timer 0 and 1.

To make a 16-bit interval timer by cascade connecting timer 0 and timer 1, set timer 0/timer 1 mode register, TMOD <T10M1,0> to "0,1".

When set in 16-bit timer mode, the overflow output of timer 0 will become the input clock of timer 1, regardless of the set value TMOD <T1CLK1,0>. Table 3.7 (2) shows the relation between the cycle of timer (interrupt) and the selection of input clock.

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

Input Clock	Interrupt Cycle (at $f_c = 20\text{MHz}$ )	Resolution
$\phi T1 (8/f_c)$	0.4 $\mu\text{s}$ ~ 26.214ms	0.4 $\mu\text{s}$
$\phi T4 (32/f_c)$	1.6 $\mu\text{s}$ ~ 104.857ms	1.6 $\mu\text{s}$
$\phi T16 (128/f_c)$	6.4 $\mu\text{s}$ ~ 419.430ms	6.4 $\mu\text{s}$

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

Setting example: To generate an interrupt INTT1 every 0.4 seconds at  $f_c = 20\text{MHz}$ , set the following values for timer registers TREG0 and TREG1:

When counting with input clock of  $\phi T16 (6.4\mu\text{s} @ 20\text{MHz}) 0.4 \text{ sec} \div 6.4\mu\text{s} = 62500 = \text{F424H}$

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

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

With the timer 1 comparator, the match detect signal is output at each comparator timing when up-counter UC1 and TREG1 values match. When the match detect signal is output simultaneously from both comparators of timer 0 and timer 1, the up-counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If inversion is enabled, the value of the timer flip-flop TFF1 is inverted.

	Timer 0			Timer 1		
	INTT0	TO1	Value of comparison	INTT1	TO1	Value of comparison
16-Bit Timer Mode (Counts Timer 1 upward when Timer 0 is overflowed)	Generation of an interrupt	Disable to Output	TREG0 (Counts upward even when matched)	Generation of an interrupt	Disable to output	TREG1 * 2 <sup>8</sup> + TREG0 (16 bit full)
8-Bit Timer Mode (Counts Timer 1 upward when Timer 0 is matched)	Generation of an interrupt	Enable to output (Timer 0 or Timer 1)	TREG0 (Cleared when matched)	Generation of an interrupt	Enable to output (Timer 0 or Timer 1)	TREG1 * TREG0 (Value of Multiplier)

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

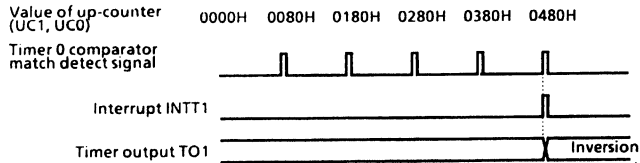


Figure 3.7 (9). Output Timer by 16-Bit Timer Mode

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

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

Timer 0 outputs pulse to TO1 pin (also used P70). In this mode, a programmable square wave is gener-

ated by inverting timer output each time the 8-bit up-counter (UC0) matches the timer registers TREG0 and TREG1.

However, it is required that the set value of TREG0 is smaller than that of TREG1.

Though the up-counter (UC1) of timer 1 is not used in this mode, UC1 should be set for counting by setting TRUN <T1RUN> to 1.

Figure 3.7 (1) shows the block diagram for this mode.

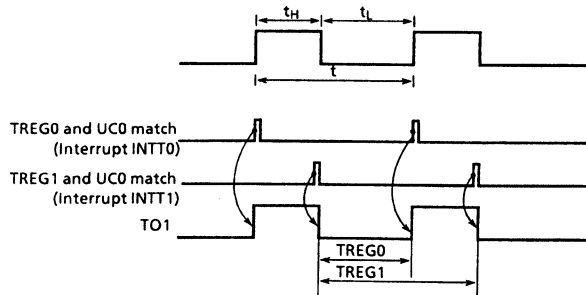


Figure 3.7 (10). Block Diagram of 8-Bit PPG Output Mode

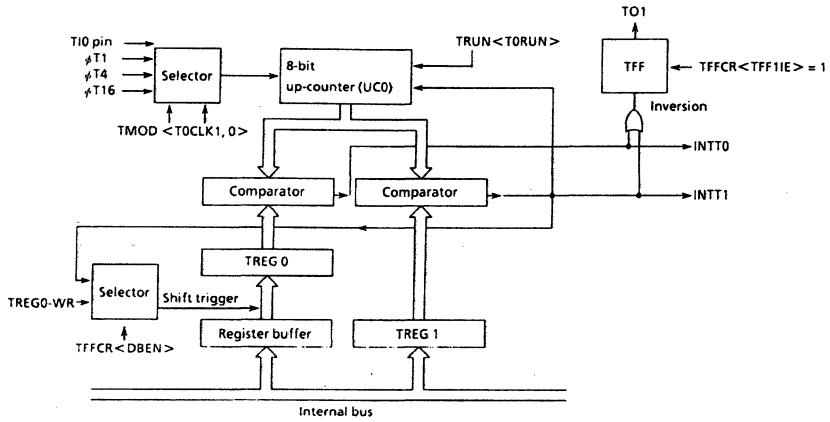


Figure 3.7 (11). Block Diagram of 8-Bit PPG Output Mode

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

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

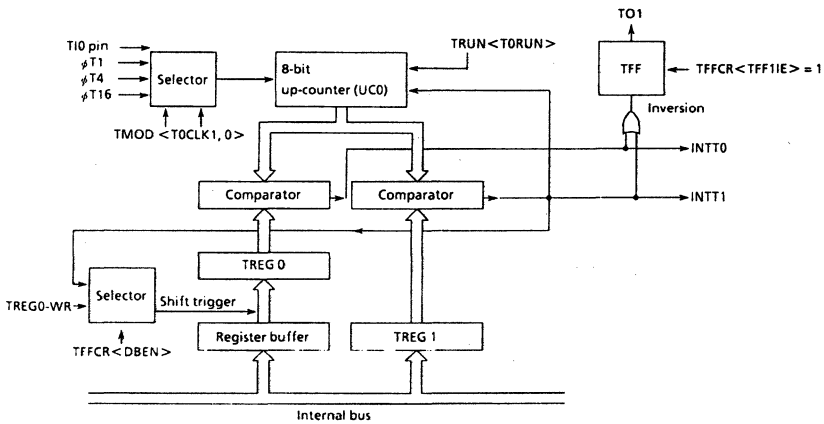


Figure 3.7 (12). Operation of Register Buffer



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:  $t = 1/62.5\text{kHz} = 16\mu\text{s}$ .

Given  $\phi T1 = 0.4\mu\text{s}$  @ 20MHz),

$$16\mu\text{s} \div 0.4\mu\text{s} = 40$$

Consequently, to set the timer register 1 (TREG1) to TREG1 = 40 = 28H and then duty to 1/4,  $t \times 1/4 =$

$$16\mu\text{s} \times 1/4 = 4\mu\text{s}$$

$$4\mu\text{s} \div 0.4\mu\text{s} = 10$$

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

		7	6	5	4	3	2	1	0
TRUN	←	-	x	-	-	-	-	0	0
TMOD	←	1	0	x	x	x	x	0	1
TREG0	←	0	0	0	0	1	0	1	0
TREG1	←	0	0	0	0	1	0	0	0
TFFCR	←	-	-	-	1	0	1	1	x
P7CR	←	x	x	x	x	-	-	1	-
P7FC	←	x	x	x	x	-	-	1	x
TRUN	←	1	x	-	-	-	-	1	1

Note: x; don't care    -; no change

Stop timer 0, and clear it to "0".

Set the 8-bit PPG mode, and select  $\phi T1$  as input clock.

Write "0AH".

Write "28H".

Sets TFF1 and enables the inversion and double buffer enable.

Writing "10" provides negative logic pulse.

Select P71 as T01 pin.

Start timer 0 and timer 1 counting.

(4) 8-bit PWM Output mode

This mode is valid only for timer 0. In this mode, maximum 8-bit resolution PWM pulse can be output. PWM pulse is output to T01 pin (also used as P71) when using timer 0. Timer 1 can also be used as 8-bit timer.

Timer output is inverted when up-counter (UC1) matches the set value of timer register TREG or when  $2^n - 1$  ( $n = 6, 7, \text{ or } 8$ ; specified by T01MOD

<PWM01,07) counter overflow occurs. Up-counter UC1 is cleared when  $2^n - 1$  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)  $\neq 0$

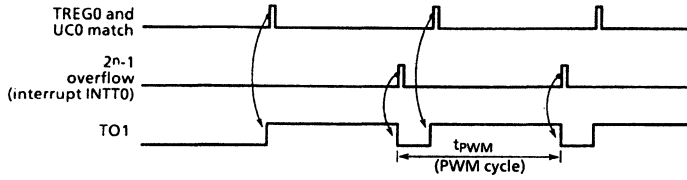


Figure 3.7 (13). 8-Bit PWM Waveforms

Figure 3.7 (14) shows the block diagram of this mode.

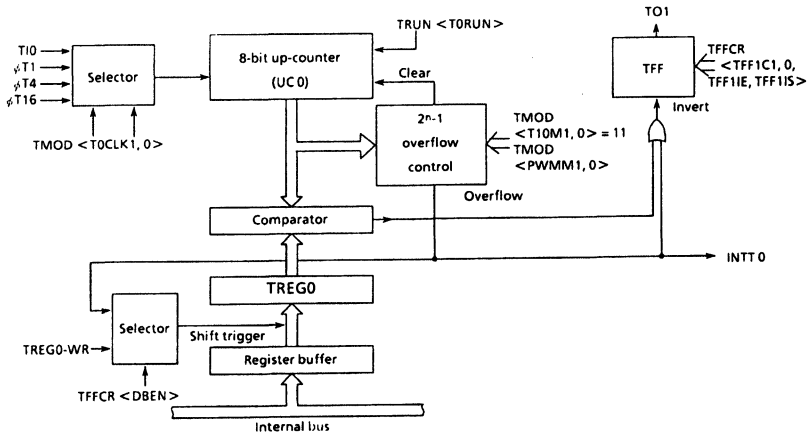


Figure 3.7 (14). Block Diagram of 8-Bit PWM Waveforms

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.

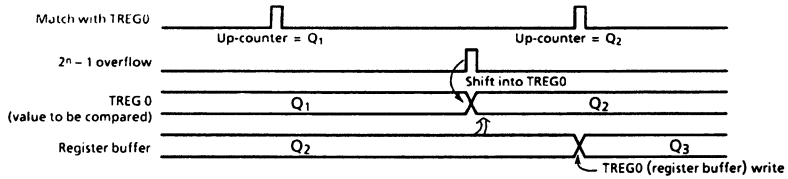
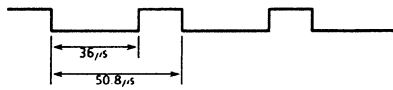


Figure 3.7 (15). Operator of Register Buffer

Example: To output the following PWM waves to TO1 pin at  $f_c = 20\text{MHz}$ .



To realize  $50.8\mu\text{s}$  of PWM cycle by  $\phi T1 = 0.4\mu\text{s}$  (@  $f_c = 20\text{MHz}$ ),

$$50.8\mu\text{s} \div 0.4\mu\text{s} = 127 = 2^7 - 1$$

Consequently,  $n$  should be set to 7. As the period of low level is  $36\mu\text{s}$ , for  $\phi T1 = 0.4\mu\text{s}$ , set the following value for TREG0:

$$36\mu\text{s} \div 0.4\mu\text{s} = 90 = 5\text{AH}$$

	MSB							LSB
	7	6	5	4	3	2	1	0
TRUN	← -	x	-	-	-	-	-	0
TMOD	← 1	1	1	0	-	-	0	1
TREG0	← 0	1	0	1	1	0	1	0
TFFCR	← x	x	x	x	1	0	1	x
P7CR	← x	x	x	x	-	-	1	-
P7FC	← x	x	x	x	-	-	1	x
TRUN	← 1	x	-	-	-	-	-	1

Note: x; don't care    -; no change

- Stop timer 0, and clear it to "0".
- Set 8-bit PWM mode (cycle:  $2^7 - 1$ ) and select  $\phi T1$  as the input clock.
- Write "5AH".
- Clears TFF1, enables the inversion and double buffer.
- Set P71 as T01 pin.
- Start timer 0 counting.

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

	PWM Cycle (@ $f_c = 20$ MHz)		
	$\phi T1$	$\phi T4$	$\phi T16$
$2^6 - 1$	25.2 $\mu$ sec (39.0kHz)	100 $\mu$ sec (10.0kHz)	0.40msec (2.4kHz)
$2^7 - 1$	50.8 $\mu$ sec (19.7kHz)	203 $\mu$ sec (4.9kHz)	0.81msec (1.2kHz)
$2^8 - 1$	102 $\mu$ sec (9.80kHz)	408 $\mu$ sec (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

Register Name	TMOD				TFFCR
Name of Function in Register	T10M	PWMM	T1CLK	TOCLK	TFF1S
Function	Timer Mode	PWM0 Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select
16-bit timer mode	01	—	—	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	—
8-bit timer x 2 channels	00	—	Lower timer match, $\phi T1, 16, 256$ (00, 01, 10, 11)	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	0 : Lower timer output 1 : Upper timer output
8-bit PPG x 1 channel	10	—	—	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	—
8-bit PWM x 1channel	11	$2^6 - 1, 2^7 - 1, 2^8 - 1$ (01, 10, 11)	—	External clock, $\phi T1, \phi T4, \phi T16$ (00, 01, 10, 11)	—
8-bit timer x 1channel	11	—	( $\phi T1, T16, T256$ ) (01, 10, 11)	—	Output disabled

Note: —: Don't care

### 3.8 8-Bit PWM Timer

The TMP96C081F has two built-in 8-bit PWM timers (timers 2 and 3).

They have two operating modes.

- 8-bit PWM (pulse width modulation: variable duty fixed interval output mode)
- 8-bit interval timer mode

Figure 3.8 (1) is a block diagram of 8-bit PWM timer (timers 2 and 3).

PWM timers consist of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Two timer flip-flops (TFF2 for timer 2 and TFF3 for timer 3) are provided.

Input clocks  $\phi P1$ ,  $\phi P4$ , and  $\phi P16$  for the PWM timers can be obtained using the built-in prescaler.

PWM timer operating mode and timer flip-flops are controlled by four control registers (P0MOD, P1MOD, PFFCR, and TRUN).

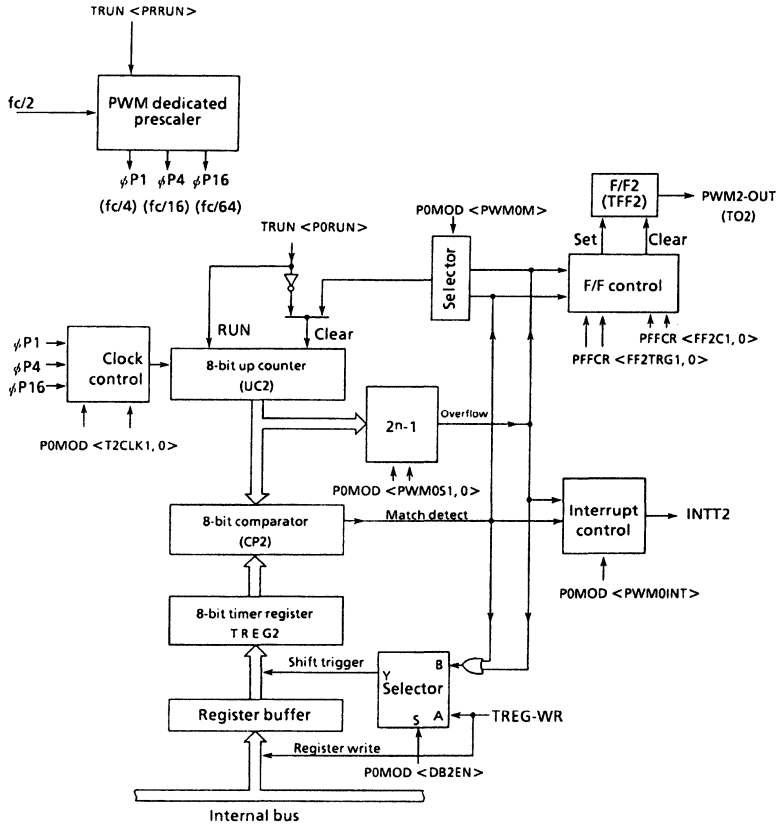


Figure 3.8 (1). Block Diagram of 8-Bit Timer 0 (Timer 2)

Note: Block diagram for 8-bit PWM timer 1 (timer 3) is the same as the above diagram.

① Prescaler

Generates input clocks dedicated to PWM timers by further dividing the fundamental clock ( $f_c$ ) after it has been divided by 2 ( $f_c/2$ ). Since the register used to control the prescaler is the same as the one for other timers, the prescaler cannot be operated independently.

The PWM timer uses three input clocks:  $\phi/P1$ ,  $\phi/P4$ , and  $\phi/P16$ .

Like the 9-bit prescaler described in the 8-bit timer section, this prescaler can be counted/stopped using bit 7 <PRRUN> of the timer operation control register TRUN. Setting <PRRUN> to 1 starts counting; setting it to 0 zero-clears and stops counting. Resetting clears <PRRUN> to 0, which clears and stops the prescaler.

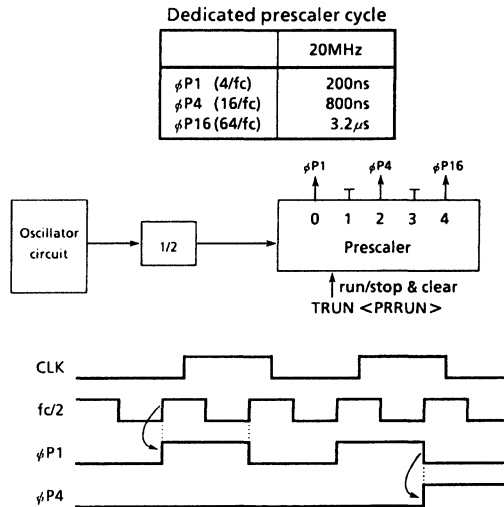


Figure 3.8 (2). Prescaler

② Up-counter

An 8-bit binary counter which counts up using the input clock specified by PWM mode register (P0MOD or P1MOD).

The input clock for the PWM0/PWM1 is selected from the internal clocks  $\phi P1$ ,  $\phi P4$ , and  $\phi P16$  (PWM dedicated prescaler output) depending on the value set in the P0MOD/P1MOD register.

Operating mode is also set by P0MOD and P1MOD registers. At reset, they are initialized to P0MOD <PWM0M> = 0 and P1MOD <PWM1M> = 0, thus, the up-counter is in PWM mode. In PWM mode, the up-counter is cleared when a  $2^n - 1$  overflow occurs; in timer mode, the up-counter is cleared at compare and match.

Count/stop and clear of the up-counter can be controlled for each PWM timer using the timer operation control register TRUN. Resetting clears all up-counters and stops timers.

③ Timer registers

Two 8-bit registers used for setting an interval time. When the value set in the timer registers (TREG 2 and 3) matches the value in the up-counter, the match detect signal of the comparator becomes active.

Timer registers TREG2 and TREG3 are each paired with register buffer to make a double buffer structure.

TREG2 and TREG3 are controlled double buffer enable/disable by P0MOD <DB2EN> and P1MOD <DB3EN>: disabled when <DB2EN>/<DB3EN> = 0, enabled when <DB2EN>/<DB3EN> = 1.

Data is transferred from register buffer to timer when a  $2^n - 1$  overflow occurs in the PWM mode, or when compare and match occurs in 8-bit timer mode. That is, with a PWM timer, the timer mode can be operated in double buffer enable state, unlike timer mode for timers 0 and 1.

At reset, <DB2EN>/<DB3EN> is initialized to 0 to disable double buffer. To use double buffer, write the data in the timer register at first, then set <DB2EN>/<DB3EN> to 1, and write the following data in the register buffer.

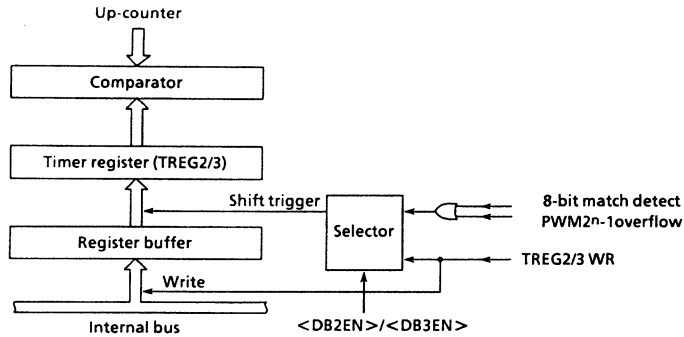


Figure 3.8 (3). Structure of Timer Registers 2 and 3

Note: The timer register and register buffer are allocated to the same memory address. When  $\langle DB2EN \rangle / \langle DB3EN \rangle = 0$ , the same value is written to both register buffer and timer register. When  $\langle DB2EN \rangle / \langle DB3EN \rangle = 1$ , the value is written to the register buffer only.

Memory addresses of the timer registers are as follows:

TREG2 : 000026H

TREG3 : 000027H

Both timer registers are write only; however, register buffer values can be read when reading the above addresses.

#### ④ Comparator

Compares the value in the up-counter with the value in the timer register (TREG2/TREG3). When they match, the comparator outputs the match detect signal. A timer interrupt (INTT2/INTT3) is generated at compare and match if the interrupt select bit  $\langle PWM01NT \rangle / \langle PWM1NT \rangle$  of the mode register (P0MOD/P1MOD) is set to 1. In timer mode, the comparator clears the up-counter to 0 at compare and match. It also inverts the value of the timer flip-flop if timer flip-flop invert is enabled.

#### ⑤ Timer flip-flop

The value of the timer flip-flop is inverted by the match detect signal (comparator output) of each interval timer or  $2^n - 1$  overflow. The value can be output to the timer output pin TO2/TO3 (also used as P72/P73).



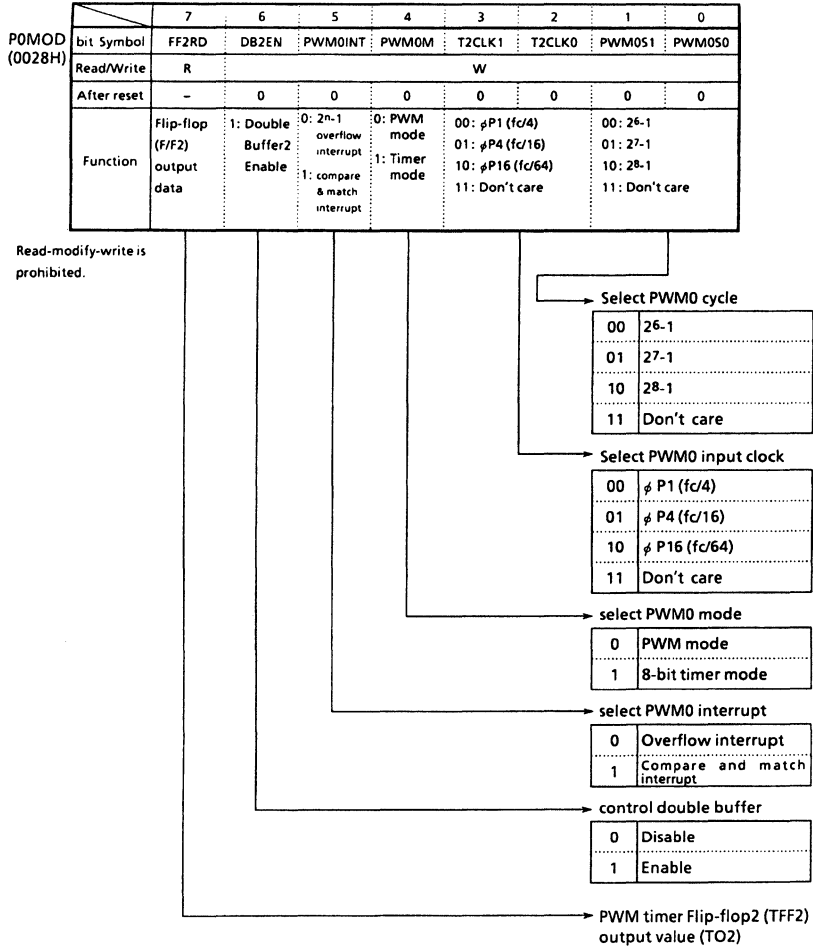


Figure 3.8 (4). 8-Bit PWM0 Mode Control Register

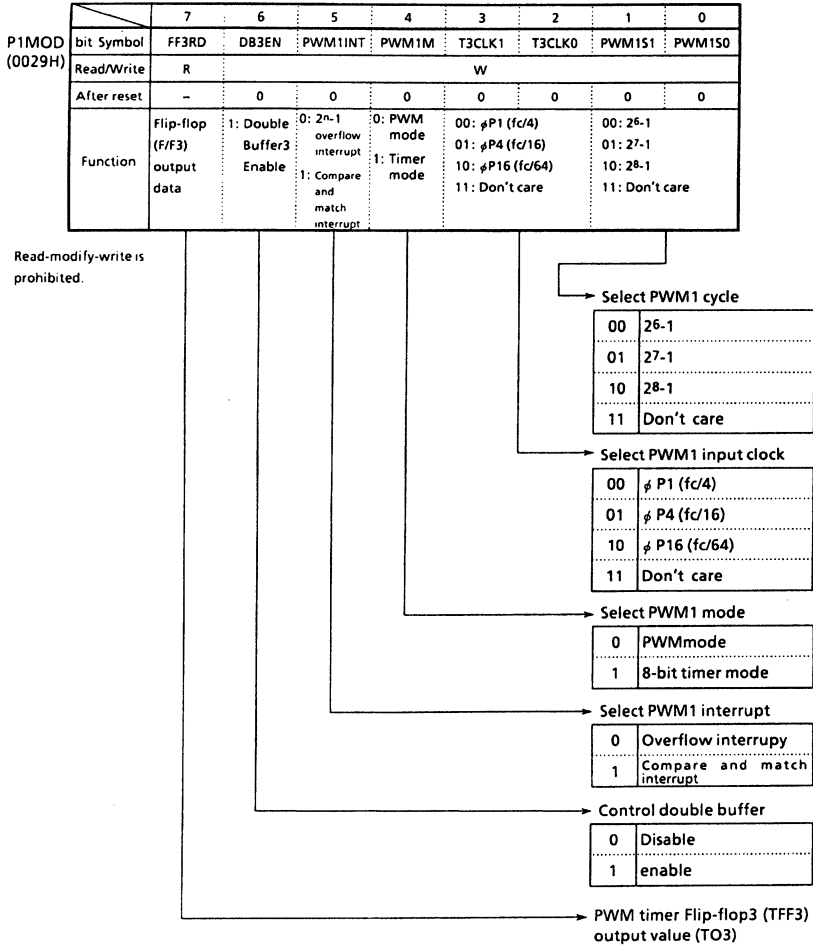


Figure 3.8 (5). 8-Bit PWM1 Mode Control Register

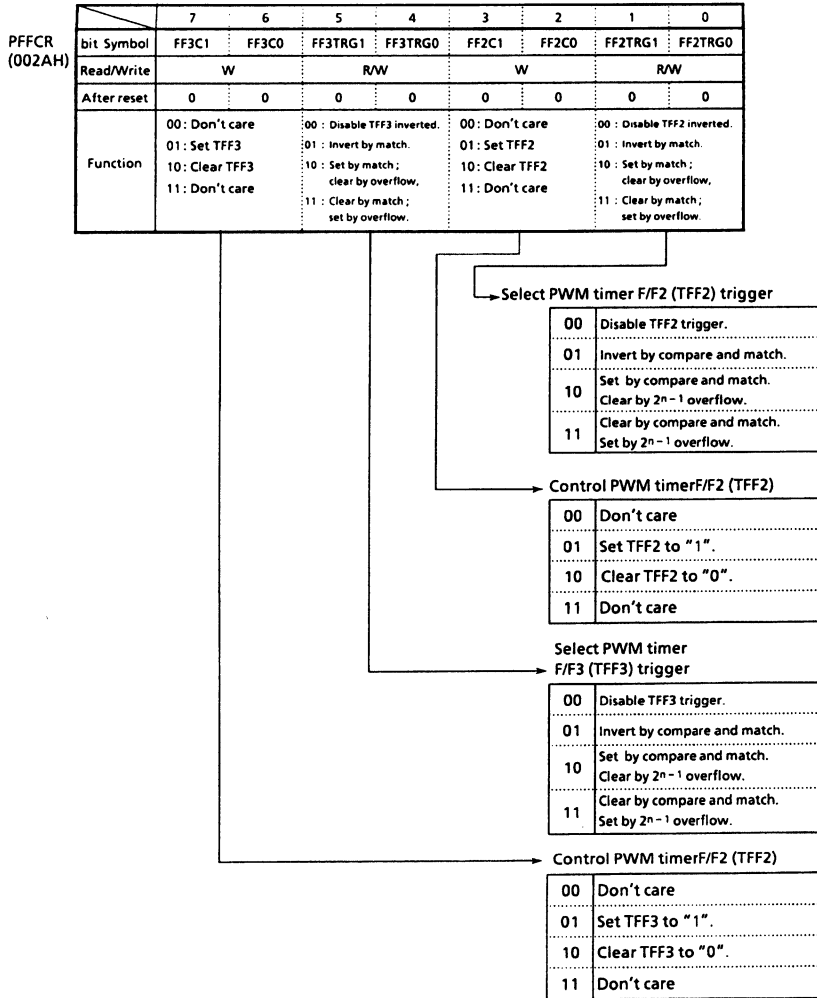


Figure 3.8 (6). 8-Bit PWM F/F Control Register

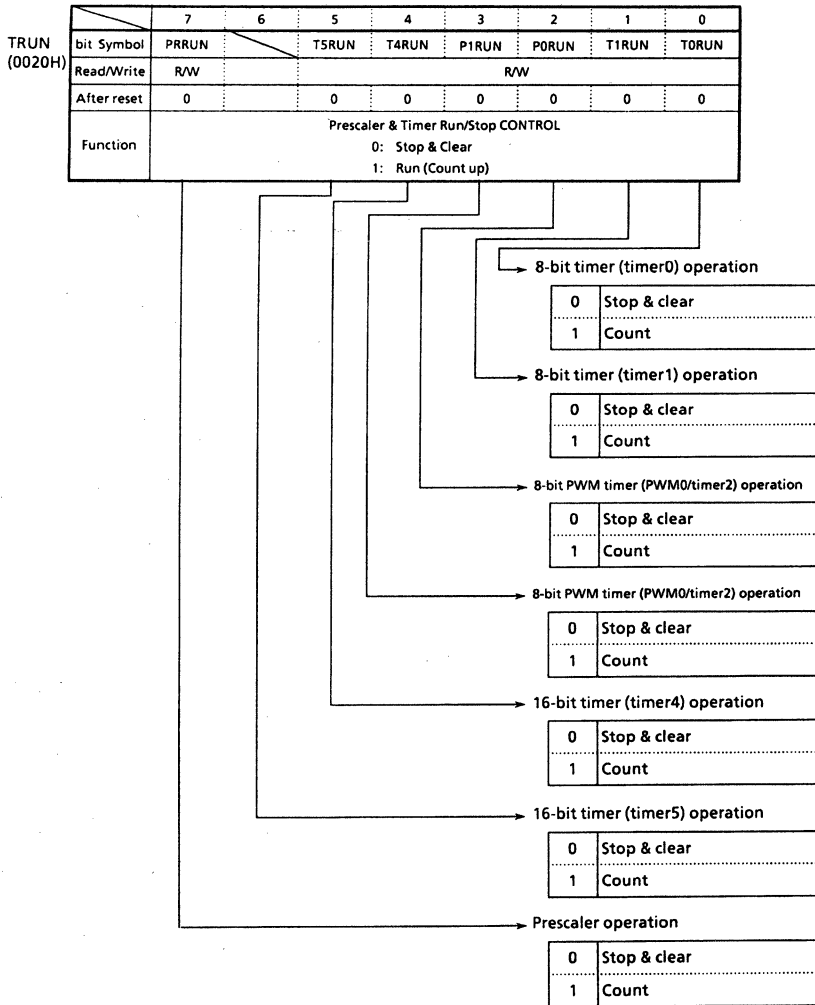


Figure 3.8 (7). Timer Operation Control Register (TRUN)

The following explains PWM timer operations.

(1) PWM timer mode

Both PWM timers can output 8-bit resolution PWM independently. Since both timers operate in exactly the same way, PWM0 is used for purposes of explanation. PWM output changes under the following two conditions.

Condition 1:

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

Condition 2:

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

The up-counter (UC2) is cleared by a  $2^n - 1$  counter overflow.

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

PWM timing

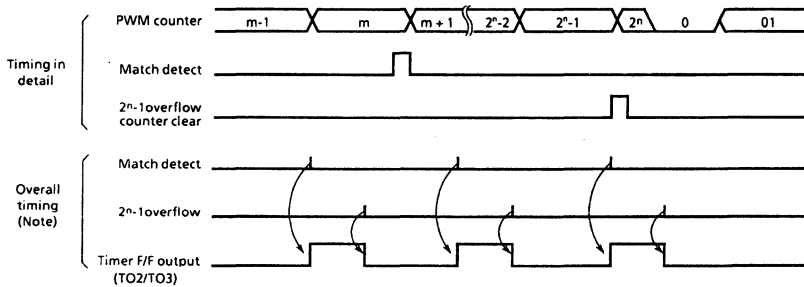


Figure 3.8 (8). Output Waves in PWM Timer Mode

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

Figure 3.8 (9) is a block diagram of this mode.

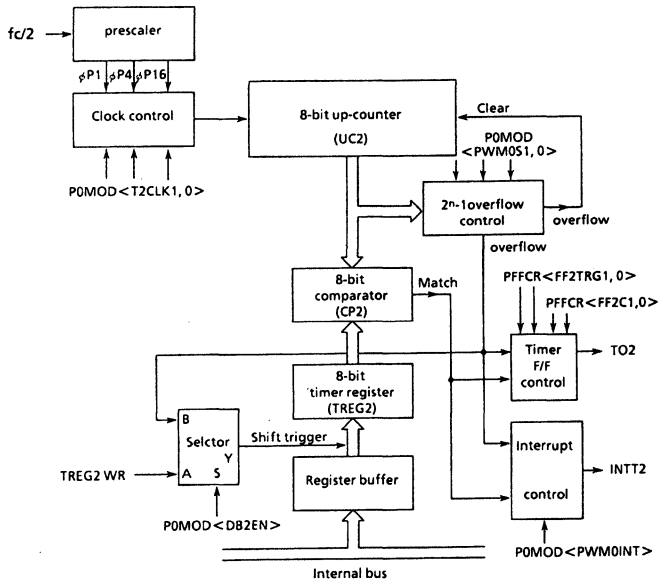


Figure 3.8 (9). Block Diagram of PWM Timer Mode (PWM0)

In this mode, enabling double buffer is very useful. The register buffer value shifts into TREG2 when a  $2^n - 1$  overflow is detected, when double buffer is enabled.

Using double buffer makes handling small duty waves easy.

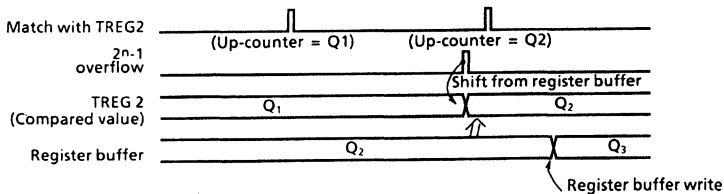
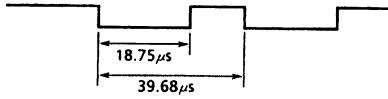


Figure 3.8 (10). Register Buffer Operation

Example: To output the following PWM waves to TO2 pin using PWM0 at  $f_c = 20\text{MHz}$ .



$$39.68\mu\text{s} + 0.25\mu\text{s} = 127 = 2^7 - 1.$$

Consequently, set n to 7.

Since the low level cycle =  $18.75\mu\text{s}$ ; for  $\phi P1 = 0.312\mu\text{s}$

$$18.75\mu\text{s} + 0.312 = 60 = 3\text{CH}$$

set the 3CH in TREG2.

To implement  $39.68\mu\text{s}$  PWM cycle by  $\phi P1 = 0.312\mu\text{s}$  (@  $f_c = 20\text{MHz}$ )

		7	6	5	4	3	2	1	0	
TRUN	←	-	x	-	-	-	0	-	-	Stops PWM0 and clears it to 0.
P0MOD	←	-	0	0	0	0	0	0	1	Sets PWM ( $2^7 - 1$ ) mode, input clock $\phi P1$ , overflow interrupt, and disables double buffer.
TREG2	←	0	0	1	1	1	1	0	0	Writes 3CH.
P0MOD	←	-	1	0	0	0	0	0	1	Enables double buffer.
PFFCR	←	-	-	-	-	0	1	1	1	Sets TFF2 and a mode where TFF2 is set by compare and match, and cleared by overflow.
P7CR	←	x	x	x	x	-	1	-	-	) Sets P72 as the TO2 pin.
P7FC	←	x	x	x	x	-	1	-	x	
TRUN	←	1	x	-	-	-	1	-	-	Starts PWM0 counting.

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

Table 3.8 (1) PWM Cycle and  $2^n - 1$  Counter Setting

	Formula	20MHz		
		$\phi P1$	$\phi P4$	$\phi P16$
$2^6 - 1$	$2^6 - 1 - \phi Pn$	12.6μsec (79kHz)	50.4μsec (20kHz)	201μsec (4.9kHz)
$2^7 - 1$	$2^7 - 1 - \phi Pn$	25.4μsec (39kHz)	101.6μsec (9.8kHz)	406μsec (2.5kHz)
$2^8 - 1$	$2^8 - 1 - \phi Pn$	51.0μsec (20kHz)	204.0μsec (4.9kHz)	816μsec (1.2kHz)

(2) 8-bit timer mode

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

① Generating interrupts at a fixed interval

To generate timer 2 interrupt (INTT2) at a fixed interval using PWM0 timer, first stop PWM0, then set the operating mode, input clock, and interval in the POMOD and TREG2 registers. Next, enable INTT2 and start counting PWM0.

Example: To generate a timer 2 interrupt every 32μs at fc = 20MHz, set registers as follows:

		7	6	5	4	3	2	1	0
TRUN	←	-	x	-	-	-	0	-	-
POMOD	←	x	0	1	1	0	0	x	x
TREG2	←	1	0	1	0	0	0	0	0
INTEPW10	←	-	-	-	-	1	1	0	0
TRUN	←	1	x	-	-	-	1	-	-

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

Stops PWM0 and clears it to 0.

Sets 8-bit timer mode and selects φP1 (0.2μs) and compare interrupt.

Sets 32μs/0.2μs = A0H in timer register.

Enables INTT2 and sets interrupt level 4.

Starts PWM0 counting.

Select an input clock using the table below.

**Table 3.8 (2) Interrupt Cycle and Input Clock Selection using 8-Bit Timer Mode**

Input Clock	Interrupt Cycle (at fc = 20MHz)	Resolution
φP1 (4/fc)	0.2μs - 51.2μs	0.2μs
φP4 (16/fc)	0.8μs - 204.8μs	0.8μs
φP16 (64/fc)	3.2μs - 819.2μs	3.2μs

Note: To generate interrupts in 8-bit timer mode, bit 5 (interrupt control bit <PWM01NT>/<PWM1NT> of POMOD/P1MOD) must be set to 1.



② Generating a 50% square wave

value to the timer output pin (TO2).

To generate a 50% square wave, invert the timer flip-flop at a fixed interval and output the timer flip-flop

Example: To output a 2.4µs square wave at  $f_c = 20\text{MHz}$  from TO2 pin, set register as follows:

	7	6	5	4	3	2	1	0
TRUN	← -	x	-	-	-	0	-	-
POMOD	← x	0	1	1	0	0	x	x
TREG2	← 0	0	0	0	0	1	1	0
PFFCR	← -	-	-	-	1	0	0	1
P7CR	← x	x	x	x	-	1	-	-
P7FC	← x	x	x	x	-	1	-	x
TRUN	← 1	x	-	-	-	1	-	-

Stops PWM0 and clears it to 0.

Sets 8-bit timer mode and selects  $\phi P1$  (0.2µs) as the input clock.

Sets  $2.4\mu\text{s}/0.2\mu\text{s}/2 = 6$  in the timer register.

Clears TFF2 to 0 and inverts using comparator output.

) Sets P72 as the TO2 pin.

Starts counting PWM0.

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

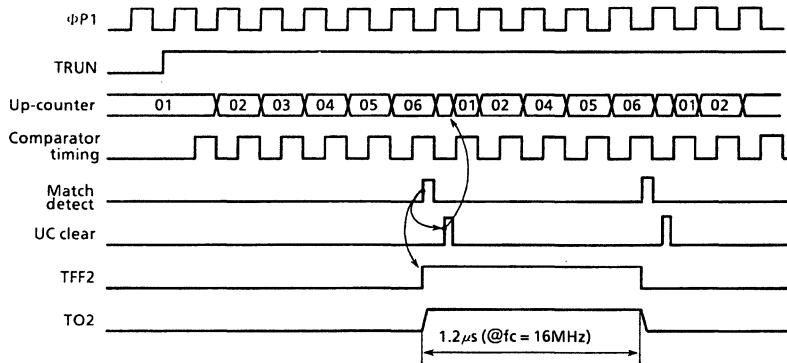


Figure 3.8 (11). Square Wave (50% Duty) Output Timing Chart

This mode is as shown in Figure 3.8 (12) below.

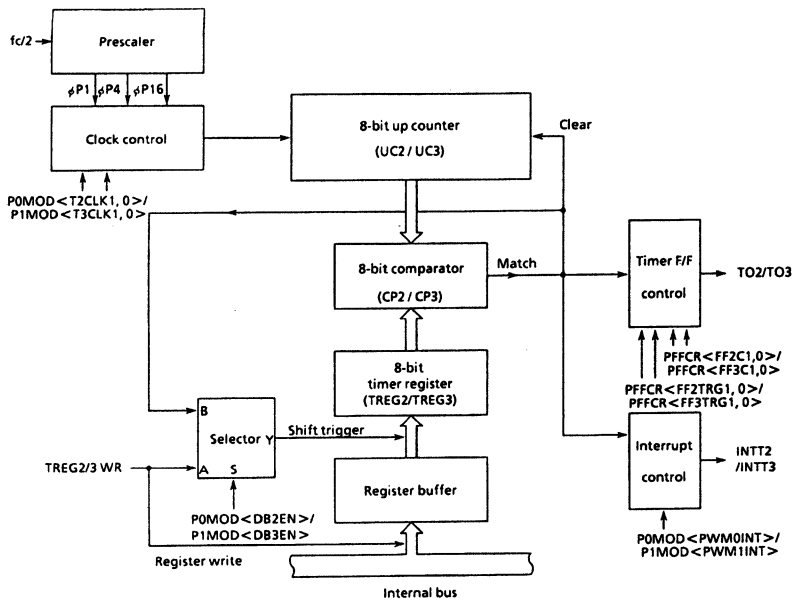


Figure 3.8 (12). Block Diagram of 8-Bit Timer Mode

### 3.9 16-Bit Timer

The TMP96C081F contains two (timer 4 and timer 5) multi-functional 16-bit timer/event counter with the following operation modes.

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

- Time differential measurement mode

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

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

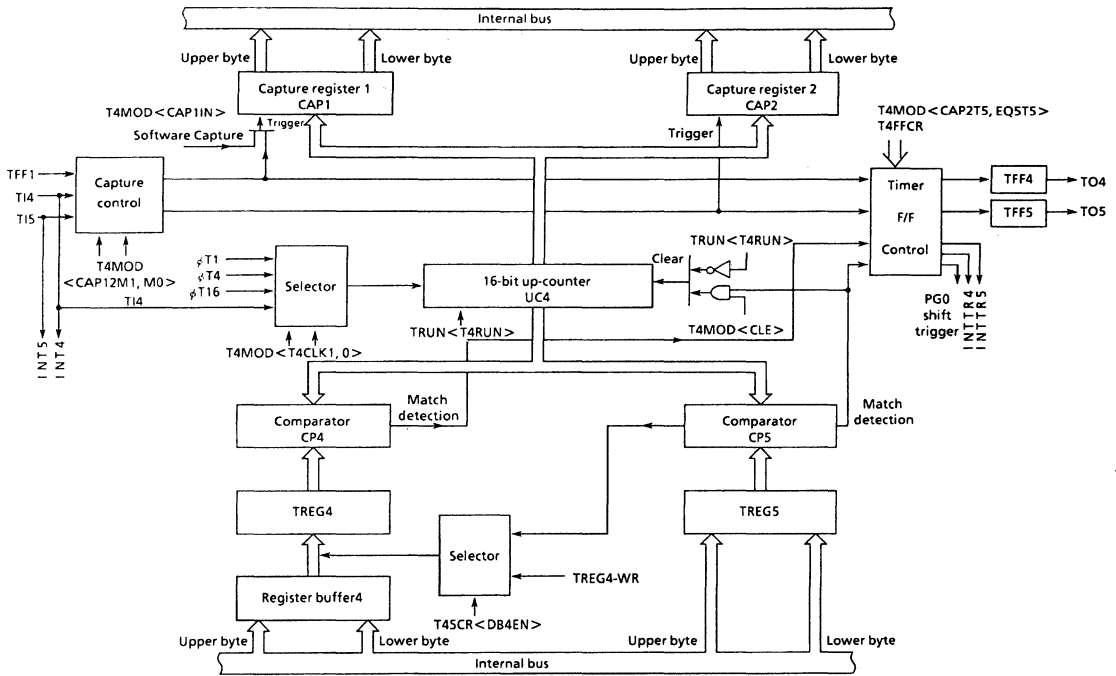


Figure 3.9 (1). Block Diagram of 16-Bit Timer (Timer 4)

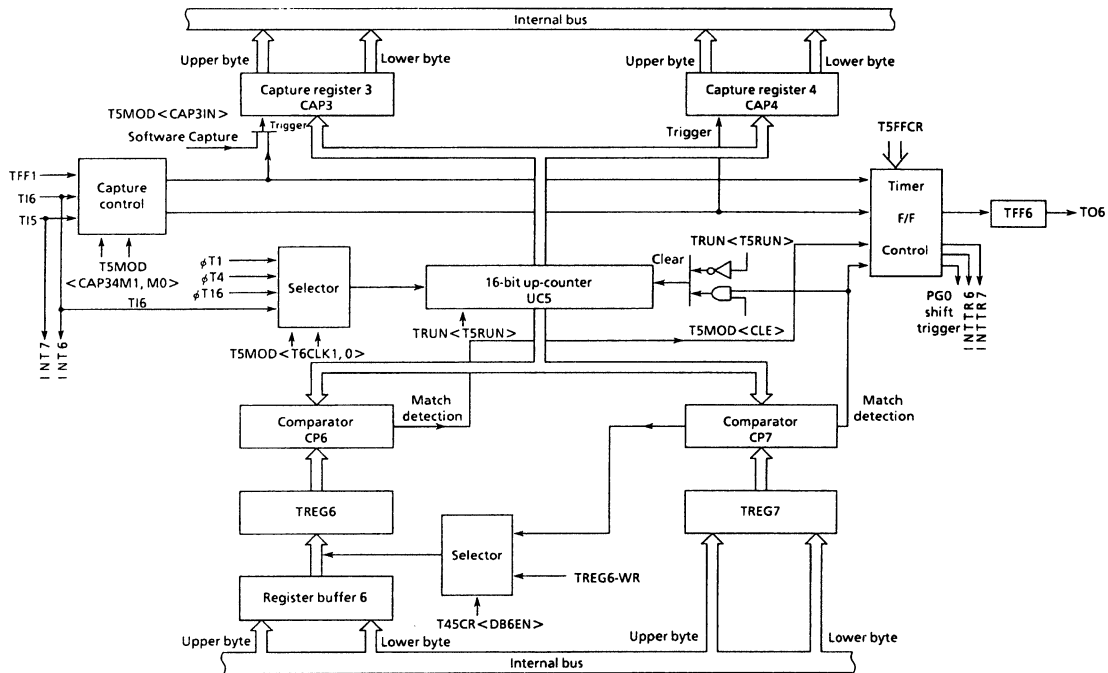


Figure 3.9 (2). Block Diagram of 16-Bit Timer (Timer 5)

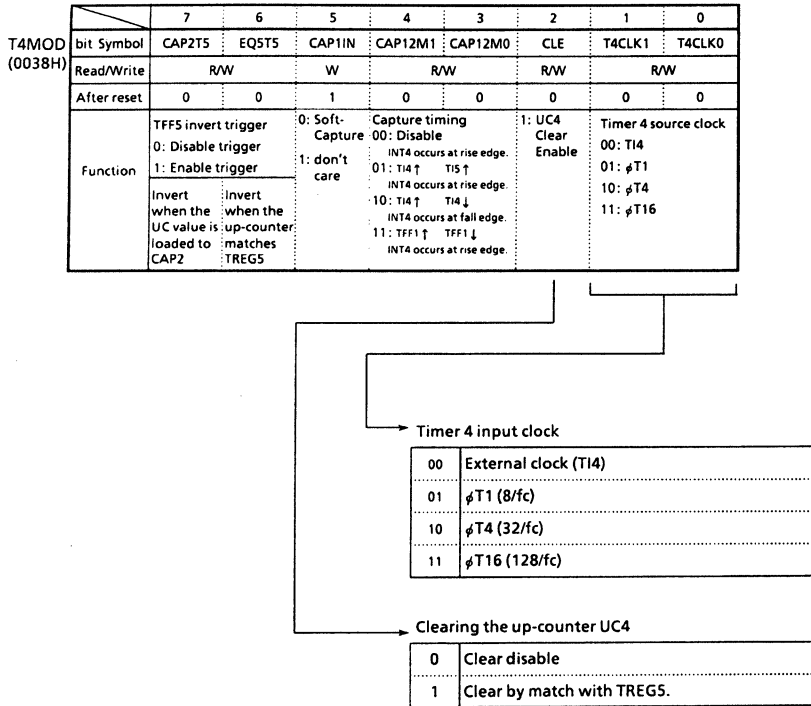
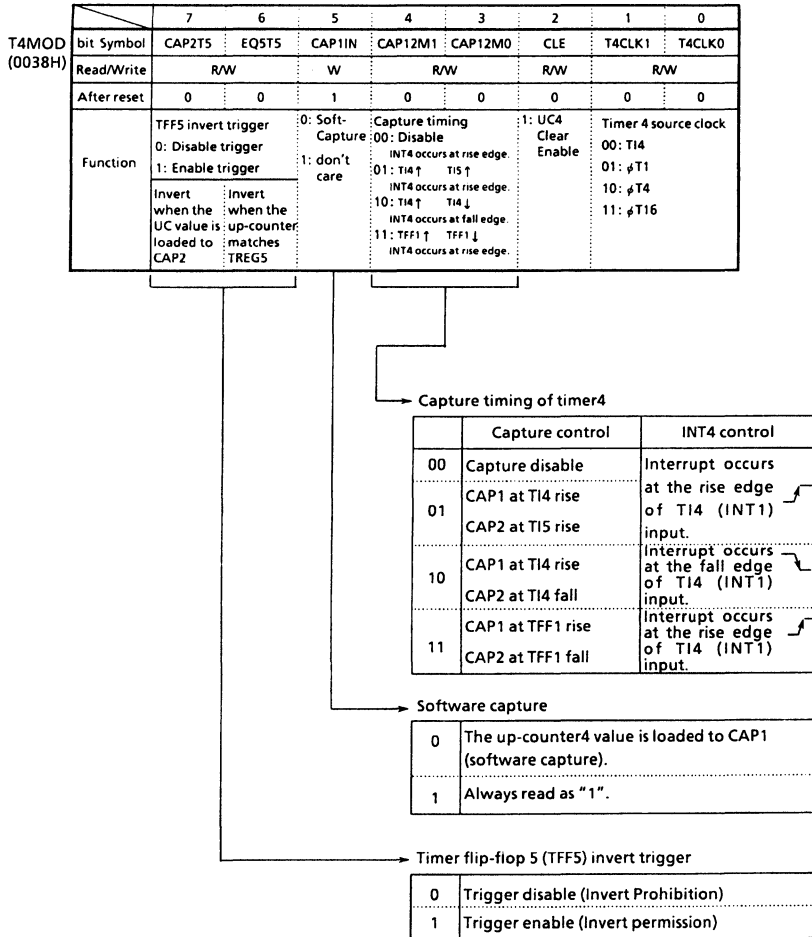


Figure 3.9 (3). 16-Bit Timer Mode Controller Register (T4MOD) (1/2)



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

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

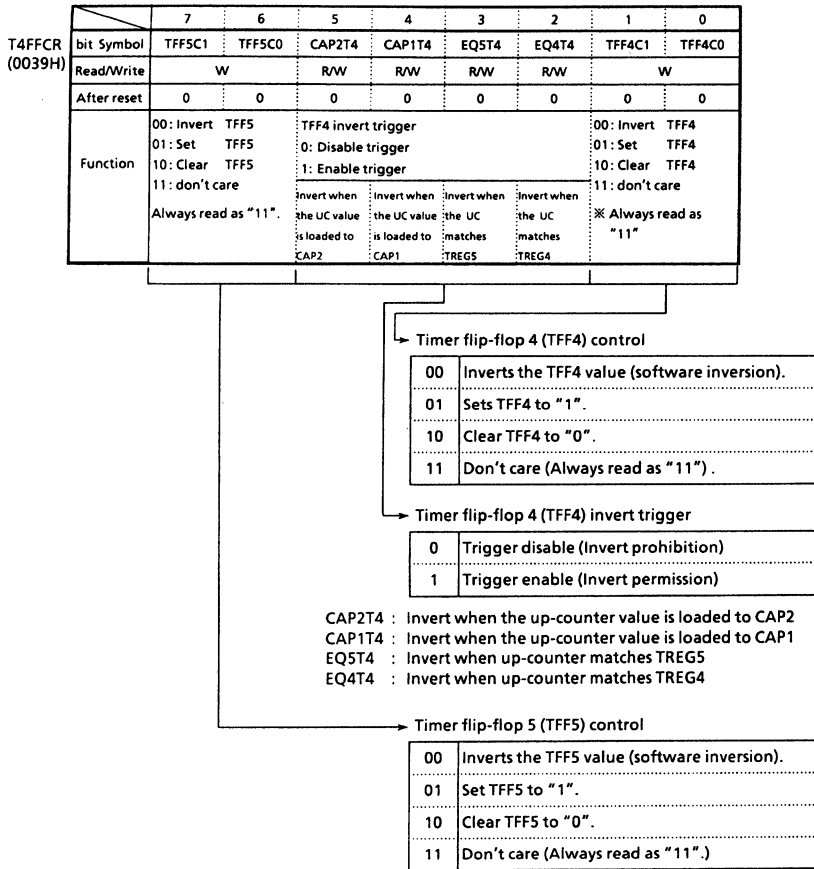


Figure 3.9 (5). 16-Bit Timer 4 F/F Control (T4FFCR)



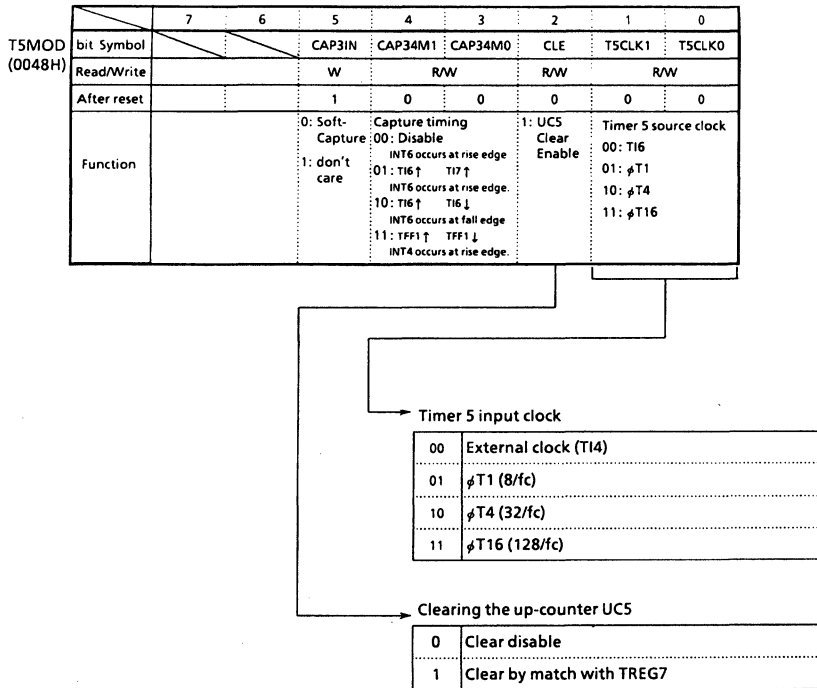


Figure 3.9 (6). 16-Bit Timer Mode Control Register (T5MOD) (1/2)

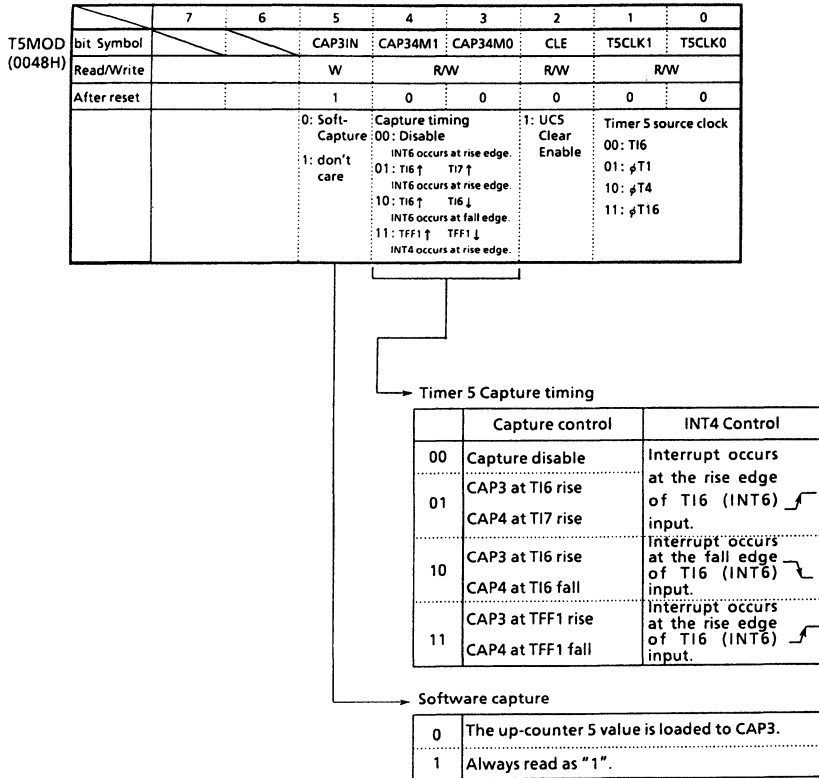
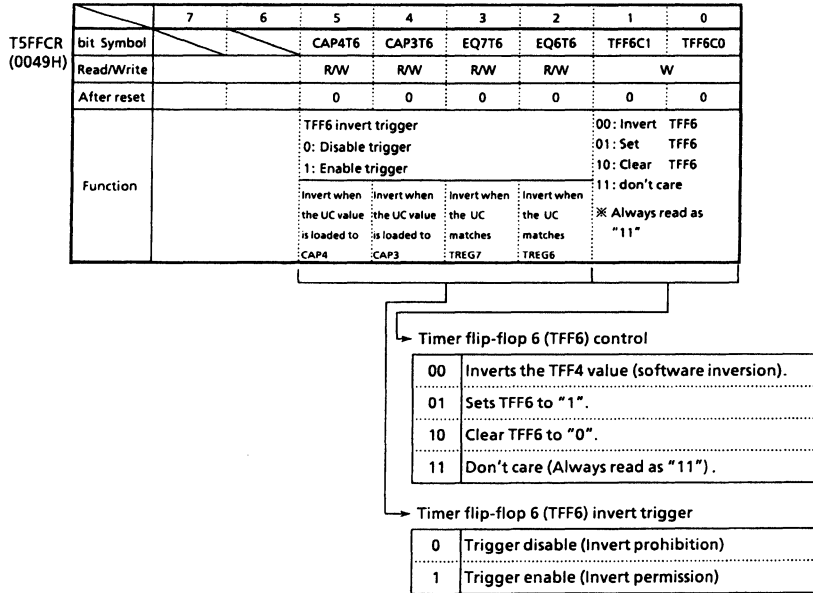
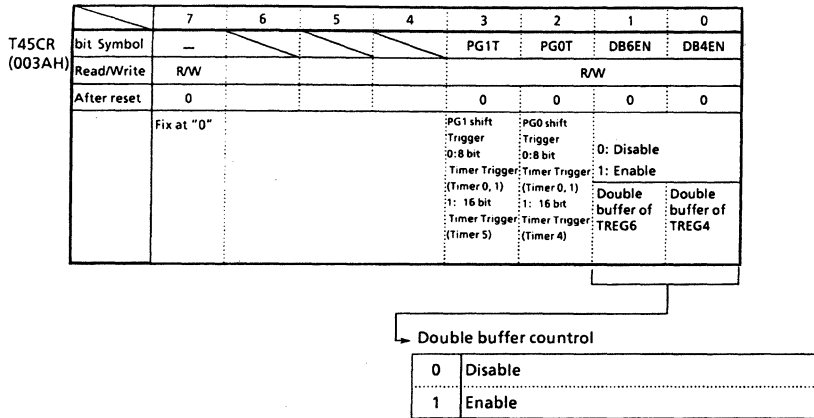


Figure 3.9 (7). 16-Bit Timer Control Register (T5MOD) (2/2)



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

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



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

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

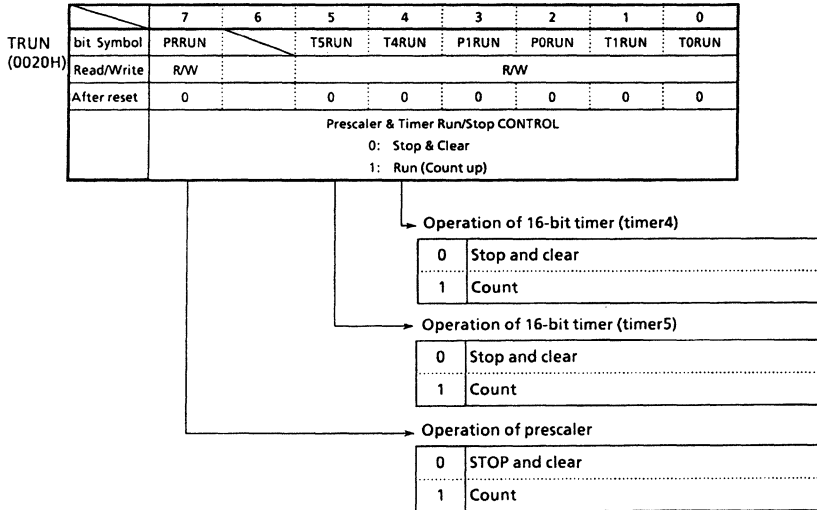


Figure 3.9 (10). Timer Operation Control Register (TRUN)

① Up-counter (UC4/UC5)

UC4/UC5 is a 16-bit binary counter which counts up according to the input clock specified by T4MOD <T4CLK1, 0> or T5MOD <T5CLK1, 0> register.

As the input clock, one of the internal clocks  $\phi$  T1 (8/fc),  $\phi$  T4 (32/fc), and  $\phi$  T16 (128/fc) from 9-bit prescaler (also used for 8-bit timer), and external clock from TI4 pin (also used as P80/INT4 pin) or TI6 (also used as P84/INT6 pin) can be selected. When reset, it will be initialized to <T4CLK1, 0>/<T5CLK1, 0> = 00 to select TI4/TI6 input mode. Counting or stop and clear of the counter is controlled by timer operation control register TRUN <T4RUN, T5RUN>.

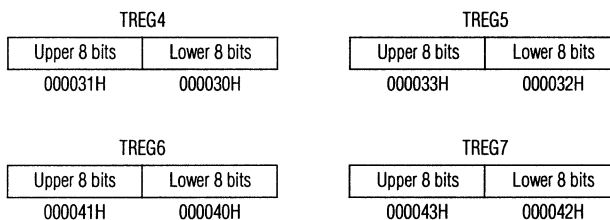
When clearing is enabled, up-counter UC4/UC5 will be cleared to zero each time it coincides matches the

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

② Timer Registers

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

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



TREG4 and TREG6 timer register is of double buffer structure, which is paired with register buffer. The timer control register T45CR <DB4EN, DB6EN> controls whether the double buffer structure should be enabled or disabled. : disabled when <DB4EN, DB6EN> = 0, while enabled when <DB4EN, DB6EN> = 1.

When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up-counter (UC4/UC5) and timer register TREG5/TREG7.

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

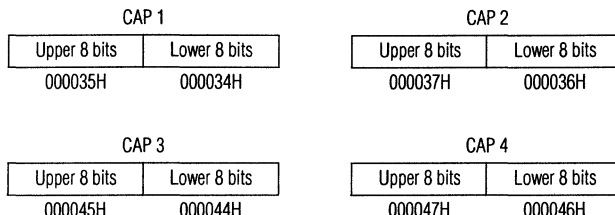
TREG4, TREG6 and register buffer are allocated to the

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

③ Capture Register

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

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



#### ④ Capture Input Control

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

- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 00  
Capture function is disabled. Disable is the default on reset.
- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 01  
Data is loaded to CAP1, CAP3 at the rise edge of TI4 pin (also used as P80/INT4) and TI6 pin (also used as P84/INT6) input, while data is loaded to CAP2, CAP4 at the rise edge of TI5 pin (also used as P81/INT5 and TI7 pin (also used as P85/INT7) input. (Time difference measurement)
- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 10  
Data is loaded to CAP1 at the rise edge of TI4 pin input and to CAP3 at the rise edge of TI6, while to CAP2, CAP4 at the fall edge. Only in this setting, interrupt INT4/INT6 occurs at fall edge. (Pulse width measurement)
- When T4MOD <CAP12M1, 0>/T5MOD <CAP34M1, 0> = 11

Data is loaded to CAP1, CAP3 at the rise edge of timer flip-flop TFF1, while to CAP2, CAP4 at the fall edge. Besides, the value of up-counter can be loaded to capture registers by software. Whenever "0" is written in T4MOD <CAPIN>, T5MOD <CAP3IN> the current value of up-counter will be loaded to capture register CAP1/CAP3. It is necessary to keep the prescaler in RUN mode (TRUN <PRRUN> to be "1").

#### ⑤ Comparator

These are 16-bit comparators which compare the up-counter UC4/UC5 value with the set value of (TREG4, TREG5)/(TREG6, TREG7) to detect the match. When a match is detected, the comparators generate an interrupt (INTT4, INTT5)/(INTT6, INTT7) respectively. The up-counter UC4/UC5 is cleared only when UC4/UC5 matches TREG5/TREG7. (The clearing of up-counter UC4/UC5 can be disabled by setting T4MOD <CLE>/T5MOD <CLE> = 0.)

#### ⑥ Timer Flip-Flop (TFF4/TFF6)

This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable/enable of inversion can be set for each element by T4FFCR <CAP2T4, CAP1T4, EQ5T4, EQ4T4>/T6FFCR <CAP4T6, CAP3T6, EQ7T6, EQ6T6>. TFF4/TFF6 will be inverted when "00" is written in T4FFCR <TFF4C1, 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 TFF4/TFF6 can be output to the timer output pin TO4 (also used as P82) and TO6 (also used as P86).

#### 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).

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

(1) 16-bit Timer Mode

Timers 4 and 5 operate independently.  
 Since both timers operate in exactly the same way,

timer 4 is used for the purposes of explanation.  
 Generating interrupts at fixed intervals:  
 In this example, the interval time is set in the timer register TREG5 to generate the interrupt INTTR5.

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

Note: x; don't care      -: no change

(2) 16-bit Event Counter Mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TI4/TI6 pin input) as the input clock. To read the value of the counter, first perform "software capture" once and read the captured value.

The counter counts at the rise edge of TI4/TI6 pin input.  
 TI4/TI6 pin can also be used as P80/INT4 and P84/INT6.  
 Since both timers operate in exactly the same way, timer 4 is used for the purposes of explanation.

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

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

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

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

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

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

		7	6	5	4	3	2	1	0
TRUN	←	-	x	-	0	-	-	-	-
TREG4	←	*	*	*	*	*	*	*	*
TREG5	←	*	*	*	*	*	*	*	*
T45CR	←	0	x	x	x	-	-	-	1
T4FFCR	←	1	1	0	0	1	1	0	0
T4MOD	←	0	0	1	0	0	1	*	*
					(** = 01, 10, 11)				
P8CR	←	-	-	-	-	-	1	-	-
P8FC	←	x	-	x	x	-	1	x	x
TRUN	←	1	x	-	1	-	-	-	-

- Stop timer 4.
  - Set the duty (16 bits).
  - Set the cycle (16 bits).
  - Double buffer of TREG4 enable.
  - (Changes the duty and cycle at the interrupt INTTR5)
  - Set the mode to invert TFF4 at the match with TREG4/TREG5, and also sets TFF4 to "0".
  - Select internal clock for input and disables the capture function.
- ) Assign P82 as TO4.
- Start timer 4.

Note: x; don't care    -; no change

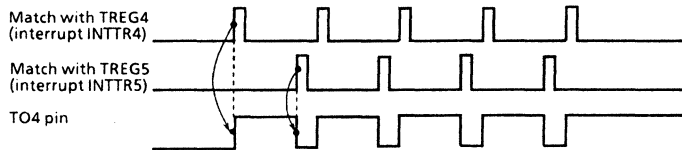


Figure 3.9 (11). Programmable Pulse Generation (PPG) Output Waveforms



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

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

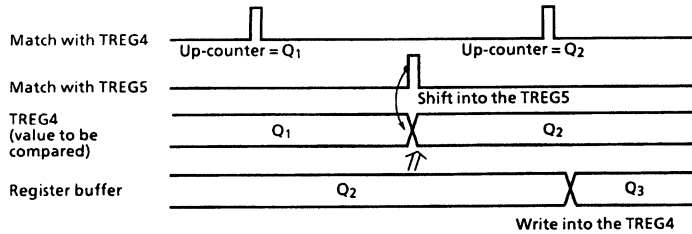


Figure 3.9 (12). Operation of Register Buffer

Shows the block diagram of this mode.

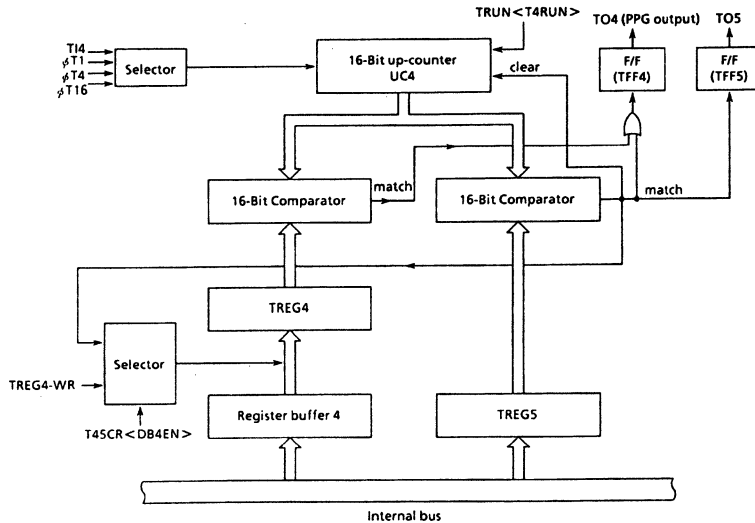


Figure 3.9 (13). Block Diagram of 16-Bit PPG Mode

(4) Application Examples of Capture Function

① One-Shot Pulse Output from External Trigger Pulse

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 TFF4 status to TO4 pin can be enabled or disabled. Combined with interrupt function, they can be applied in many ways, for example:

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

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

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

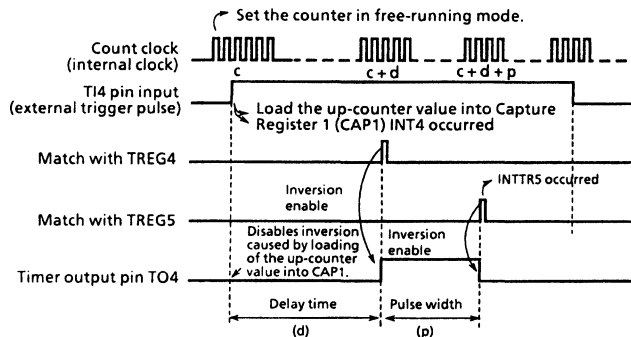
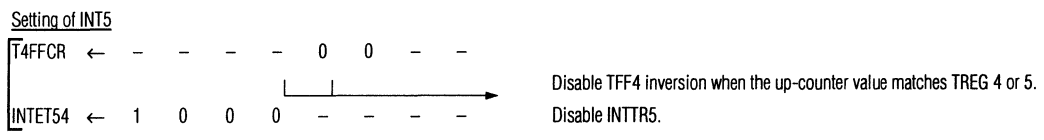
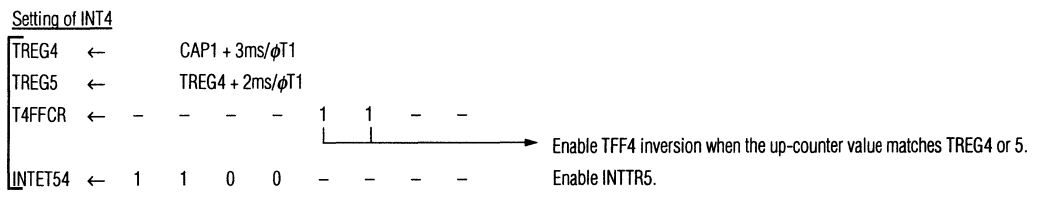
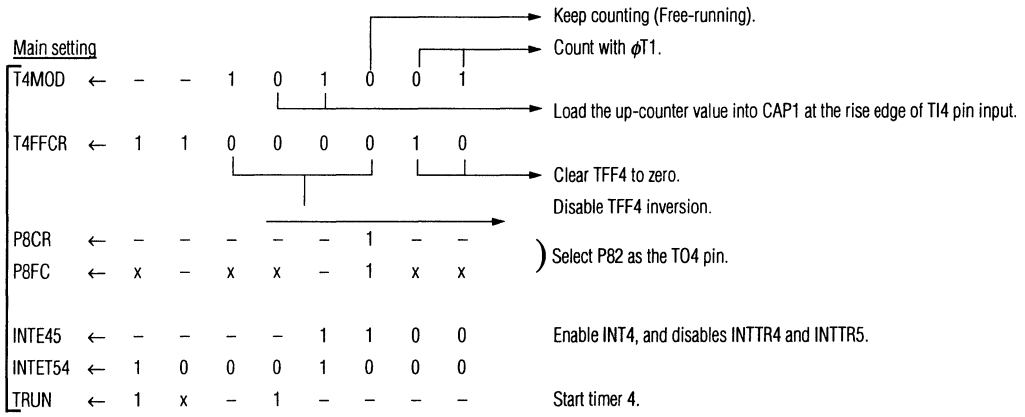


Figure 3.9 (14). One-Shot Pulse Output (with Delay)

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



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

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

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

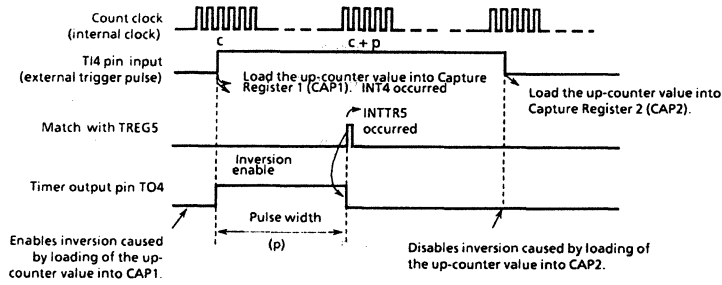


Figure 3.9 (15). One-Shot Pulse Output (without Delay)

② Frequency Measurement

The frequency of the external clock can be measured in this mode. The clock is input through the T14 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 T14 pin input should be selected for the input clock

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

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

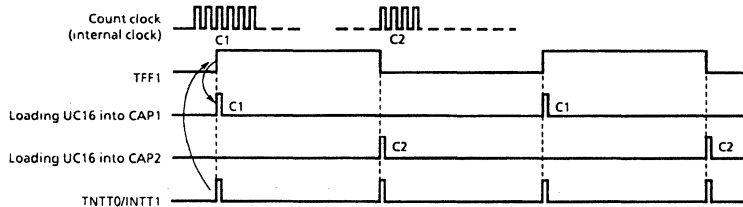


Figure 3.9 (16). Frequency Measurement

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

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

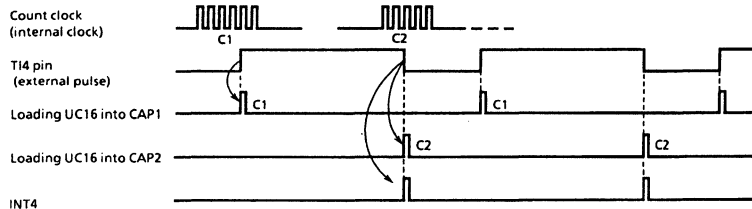
③ Pulse Width Measurement

This mode allows 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 T14 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 T14.

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

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



**Figure 3.9 (17). Pulse Width Measurement**

Note: Only in this pulse width measuring mode (T4MOD <CAP12M1, 0> = 10), external interrupt INT4 occurs at the falling edge of T14 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 T14 and T15.

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 T14. Then the interrupt INT4 is generated.

Similarly, the UC4 value is loaded into CAP2 at the rising edge of the input pulse to T15, generating the interrupt INT5.

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

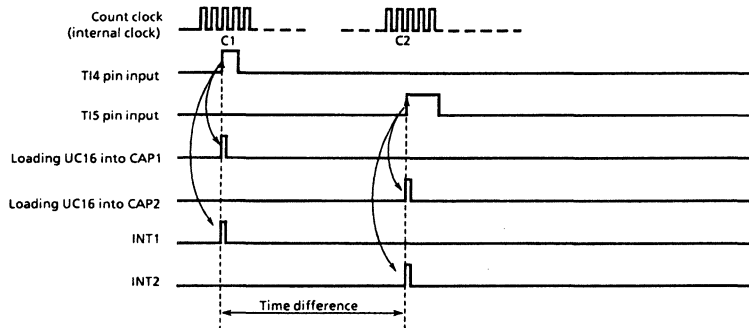


Figure 3.9 (18). Time Difference Measurement

(5) Different Phased Pulses Output Mode

In this 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.

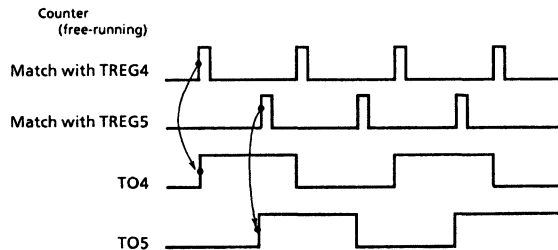


Figure 3.9 (19). Phase Output

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

	20MHz
$\phi T1$	0.819ms
$\phi T4$	3.277ms
$\phi T16$	13.11ms

**3.10 Stepping Motor Control/Pattern Generation Port**

TMP96C141/TMP96CM40/TMP96PM40 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 5, to update the output.

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

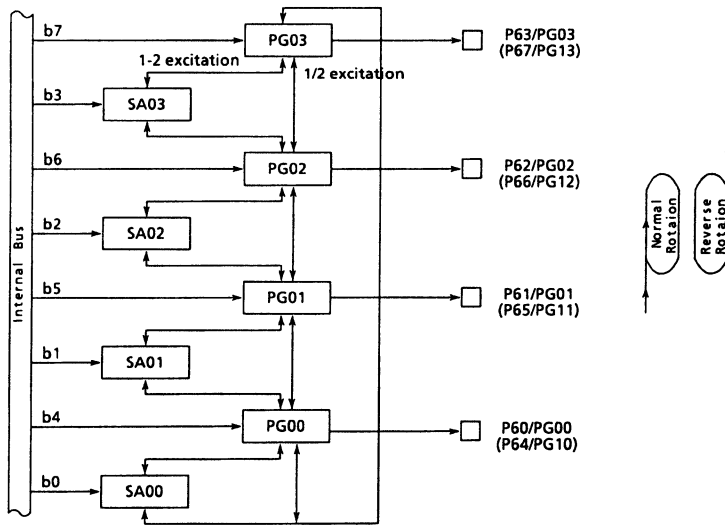


Figure 3.10 (1). PG Block Diagram

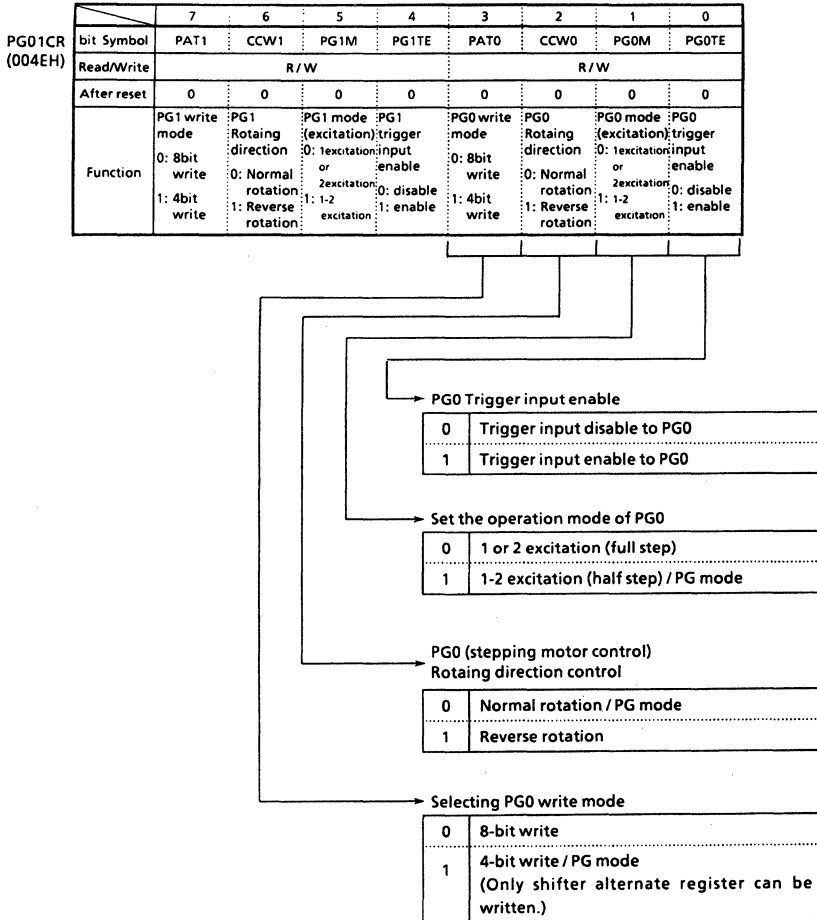


Figure 3.10 (2a). Pattern Generation Control Register (PG01CR)



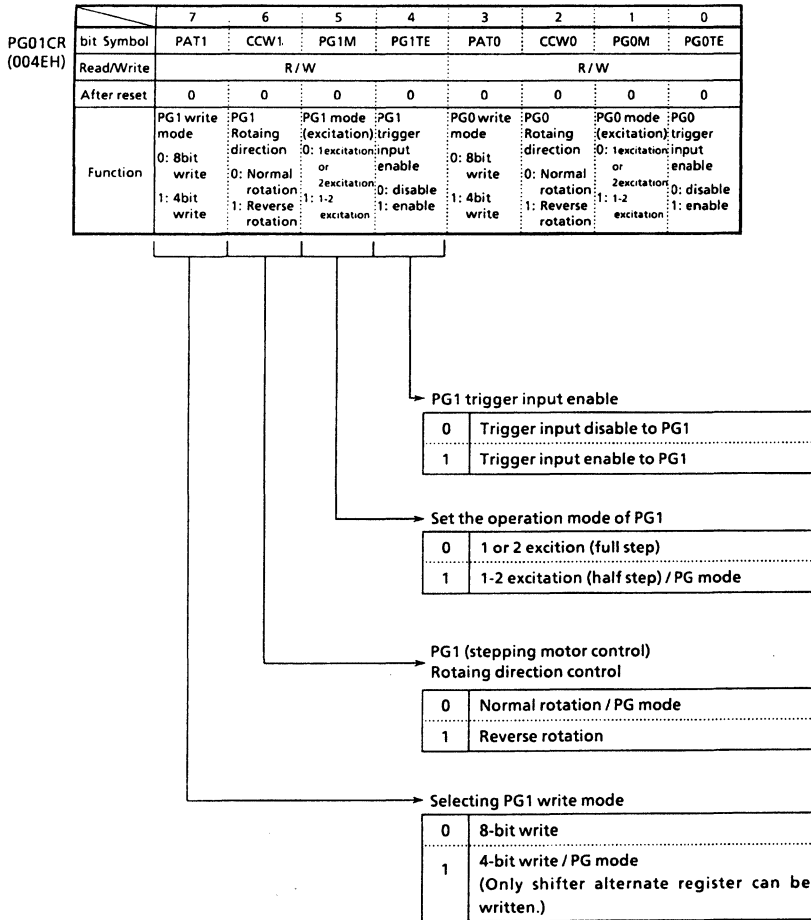


Figure 3.10 (2b). Pattern Generation Control Register (PG01CR)

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
PG0REG (004CH)	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 3.10 (3). Pattern Generation 0 Register (PG0REG)**

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
PG1REG (004DH)	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 3.10 (4). Pattern Generation 1 Register (PG1REG)**

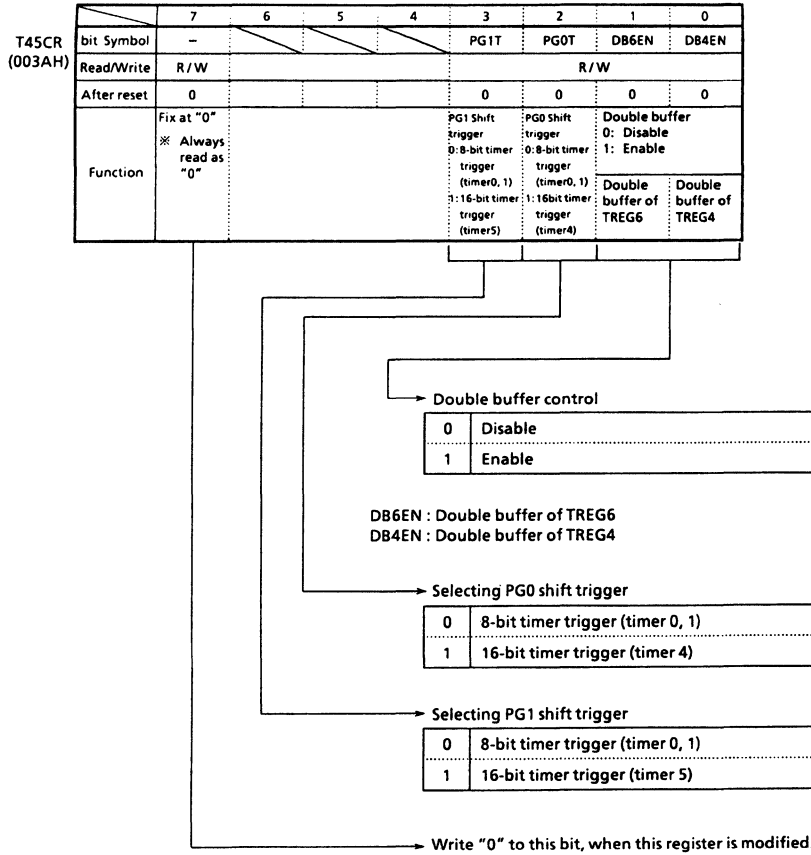


Figure 3.10 (5). 16-bit Timer Trigger Control Register (T45CR)

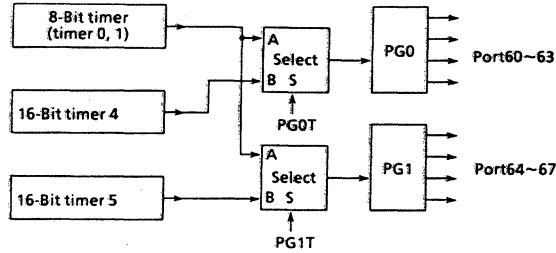


Figure 3.10 (6). Connection of Timer and Pattern Generator

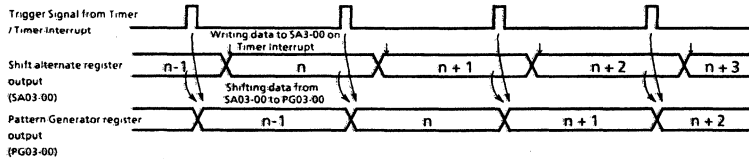
(1) Pattern Generation Mode

PG functions as a pattern generation according to the setting of PG01CR <PAT1>/<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 P6FC, any port pin can be assigned to pattern generator output.

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



Example of pattern generation mode

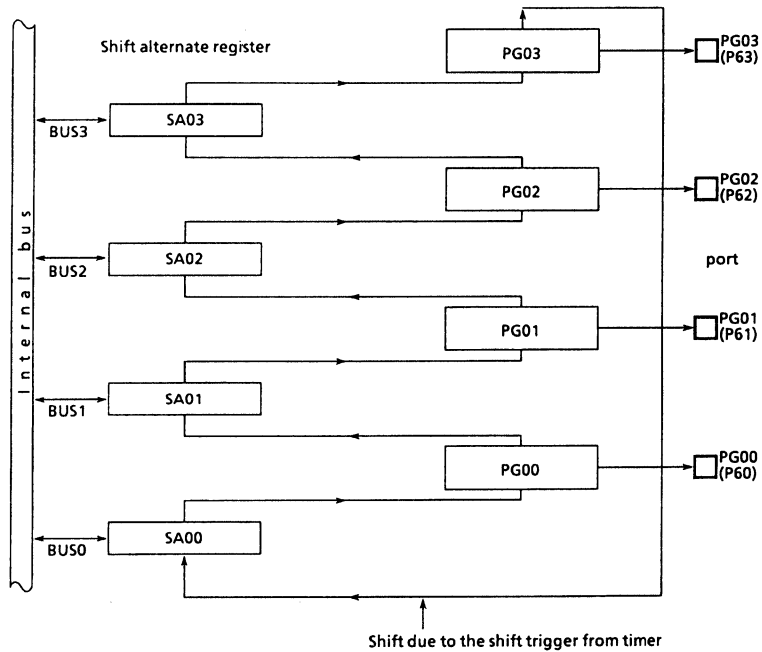


Figure 3.10 (7). Pattern Generation Mode Block Diagram (PG0)

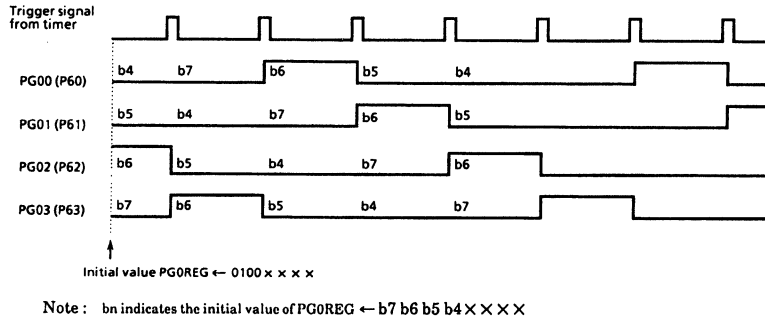
In this pattern generation mode, only writing the output latch is disabled by hardware, but other functions do the same operation as 1-2 excitation in stepping motor control port

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

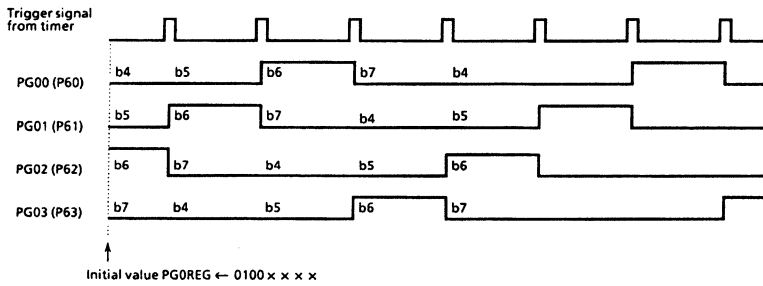
(2) Stepping Motor Control Mode

① 4-phase 1-Step/2-Step Excitation

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



① Normal Rotation



② Reverse Rotation

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

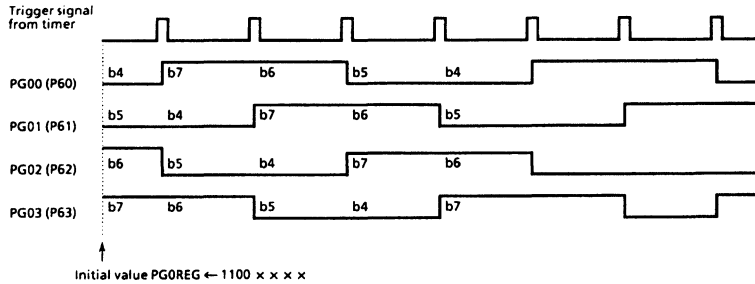


Figure 3.10 (9). Output Waveforms of 4-Phase 2-Step Excitation (Normal Rotation)

The operation when channel 0 is selected is explained below.

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

The direction of shift is specified by PG01CR <CCW0>: Normal rotation (PG00 → PG01 → PG02 → PG03) when <CCW0> is set to "0"; reverse rotation (PG00 ← PG01 ← PG02 ← PG03) when "1". 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 3.10 (10) shows the block diagram.

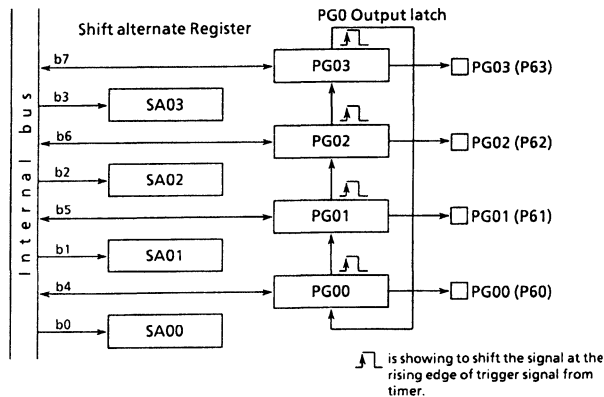
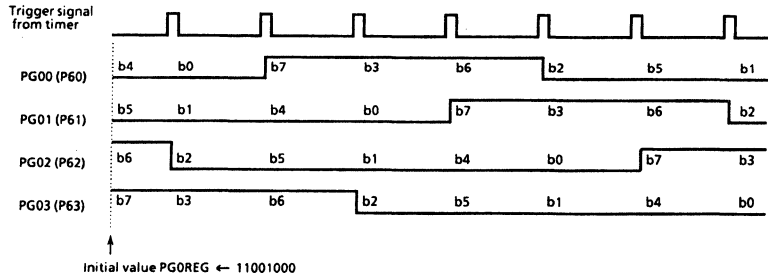


Figure 3.10 (10). Block Diagram of 4-Phase 1-Step Excitation/2-Step Excitation (Normal Rotation)

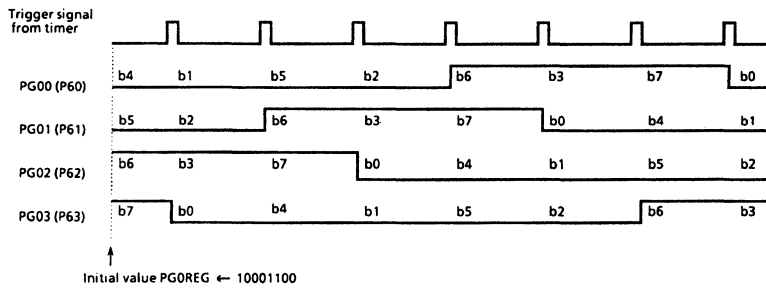
② 4-Phase 1-2 Step Excitation

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



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

① Normal Rotation



② Reverse Rotation

Figure 3.10 (11). Output Waveforms of 4-Phase 1-2 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 3.10 (12) shows the block diagram.

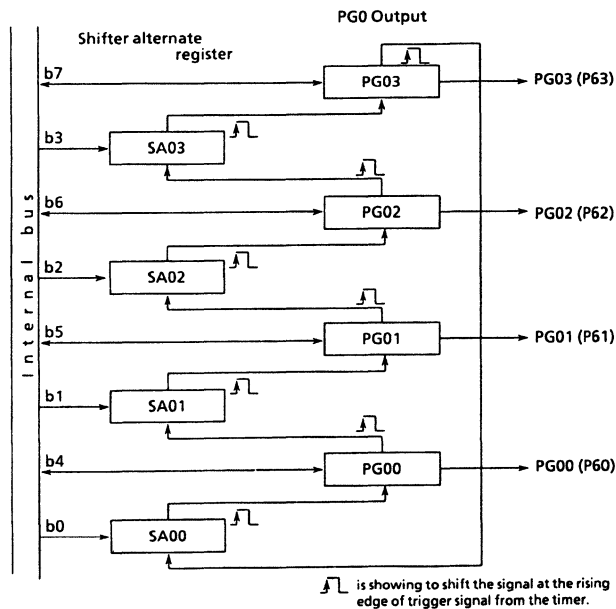


Figure 3.10 (12). Block Diagram of 4-Phase 1-2 Step Excitation (Normal Rotation)

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

timer 0 is selected, set each register as follows:

		7	6	5	4	3	2	1	0
TRUN	←	-	x	-	-	-	-	-	0
TMOD	←	0	0	x	x	-	-	0	1
TFFCR	←	x	x	x	0	1	0	1	0
TREG0	←	*	*	*	*	*	*	*	*
P6CR	←	-	-	-	-	1	1	1	1
P6FC	←	-	-	-	-	1	1	1	1
PG01CR	←	-	-	-	-	0	0	1	1
PG0REG	←	1	1	0	0	1	0	0	0
TRUN	←	1	x	-	-	-	-	-	1

Note: x; don't care    -; no change

Stop timer 0, and clears it to zero.

Set 8-bit timer mode and selects φT1 as the input clock of timer 0.

Clear TFF1 to zero and enables the inversion trigger by timer 0.

Set the cycle in timer register.

Set P60 – P63 bits to the output mode.

Set P60 – P63 bits to the PG output.

Select PG0 4-phase 1-2 step excitation mode and normal rotation.

Set an initial value.

Start timer 0.

(3) Trigger Signal From Timer

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

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

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

	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/Timer 5. In this case, the PG shift trigger signal from the 16-bit timer is output only when the up-counter UC4/UC5 value matches TREG5/TREG7.

When using a trigger signal from Timer 4, set either

T4FFCR <EQ5T4> or T4MOD <EQ5T5> to "1" and a trigger is generated when the value in UC4 and the value in TREG5 match. When using a trigger signal from Timer 5, set T5FFCR <EQ7T6> to 1. Generates a trigger when the value in UC5 and the value in TREG7 match.

(4) Application of PG and Timer Output

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

To drive a stepping motor, in addition to the value of each phase (PG output), synchronizing signal is often required at the timing when excitation is changed over. In this application, port 6 is used as a stepping motor control port to output a synchronizing signal to the TO1 pin (shared by P71).

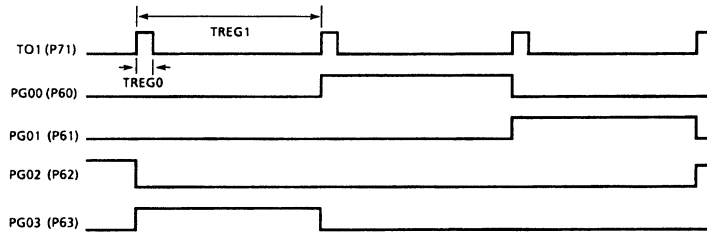


Figure 3.10 (13). Output Waveforms of 4-Phase 1-Step Excitation

Setting example:

	7	6	5	4	3	2	1	0
TRUN	← -	x	-	-	-	-	0	0
TMOD	← 1	0	x	x	x	x	0	1
TFFCR	← x	x	x	0	0	1	1	x
TREG0	← *	*	*	*	*	*	*	*
TREG1	← *	*	*	*	*	*	*	*
P7CR	← x	x	x	x	-	-	1	-
P7FC	← x	x	x	x	-	-	1	x
P6CR	← -	-	-	-	1	1	1	1
P6FC	← -	-	-	-	1	1	1	1
PG01CR	← -	-	-	-	0	0	0	1
PGOREG	← *	*	*	*	*	*	*	*
TRUN	← 1	x	-	-	-	-	1	1

- Stop timer 0, and clears it to zero.
- Set timer 0 and timer 1 in PPG output mode and selects  $\phi T1$  as the input clock.
- Enable TFF1 inversion and sets TFF1 to "1".
- Set the duty of TO1 to TREG0.
- Set the cycle of TO1 to TREG1.
- ) Assign P71 as TO1.
- ) Assign P60 - 63 as PG0.
- Set PG0 in 4-phase 1-step excitation mode.
- Set an initial value.
- Start timer 0 and timer 1.

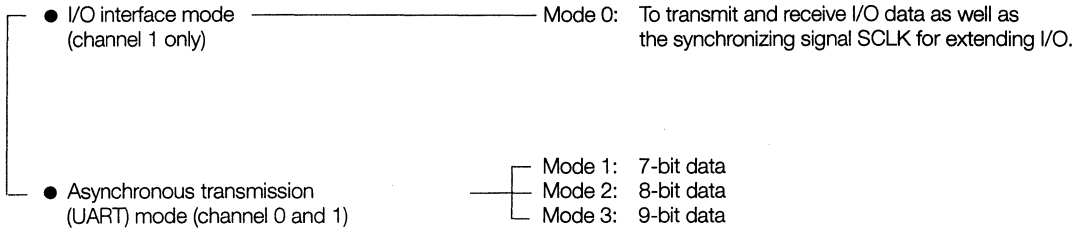
Note: x; don't care    -; no change

**3.11 Serial Channel**

TMP96C141/TMP96CM40/TMP96PM40 contains two serial I/O channels for full duplex asynchronous transmission (UART)

as well as for I/O extension.

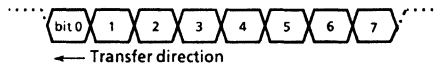
The serial channel has the following operation modes:



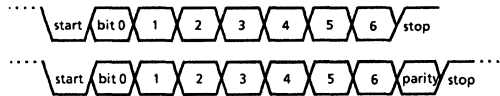
In mode 1 and mode 2, a parity bit can be added. Mode 3 has wake-up function for making the master controller start slave controllers in serial link (multi-controller system).

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

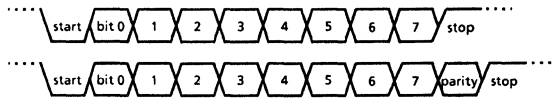
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)



When bit 8 = 1, address (select code) is denoted.  
When bit 8 = 0, data is denoted.

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

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

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

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

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

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

detected to be normal at least twice in three samplings.

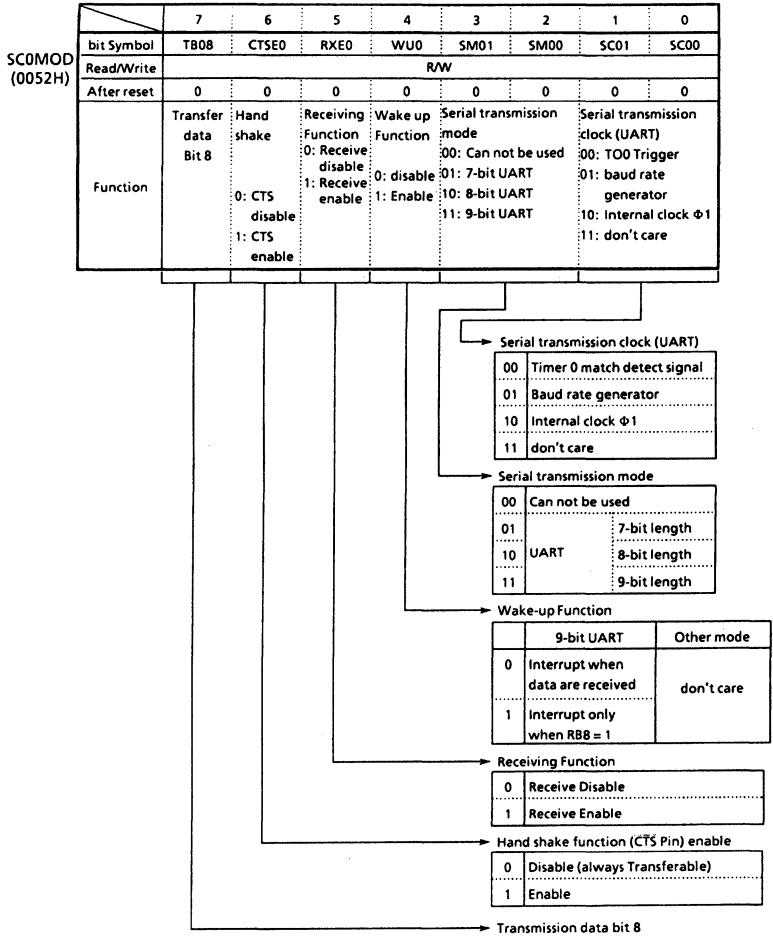
When the transmission buffer becomes empty and requests the CPU to send the next transmission data, or when data is stored in the receiving data register and the CPU is requested to read the data, INTTX or INTRX interrupt occurs. Besides, if an overrun error, parity error, or framing error occurs during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

The serial channel 0/1 includes a special baud rate generator, which can set any baud rate by dividing the frequency of four clocks ( $\phi T0$ ,  $\phi T2$ ,  $\phi T8$ , and  $\phi T32$ ) from the internal prescaler (shared by 8-bit/16-bit timer) by the value 2 to 16.

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

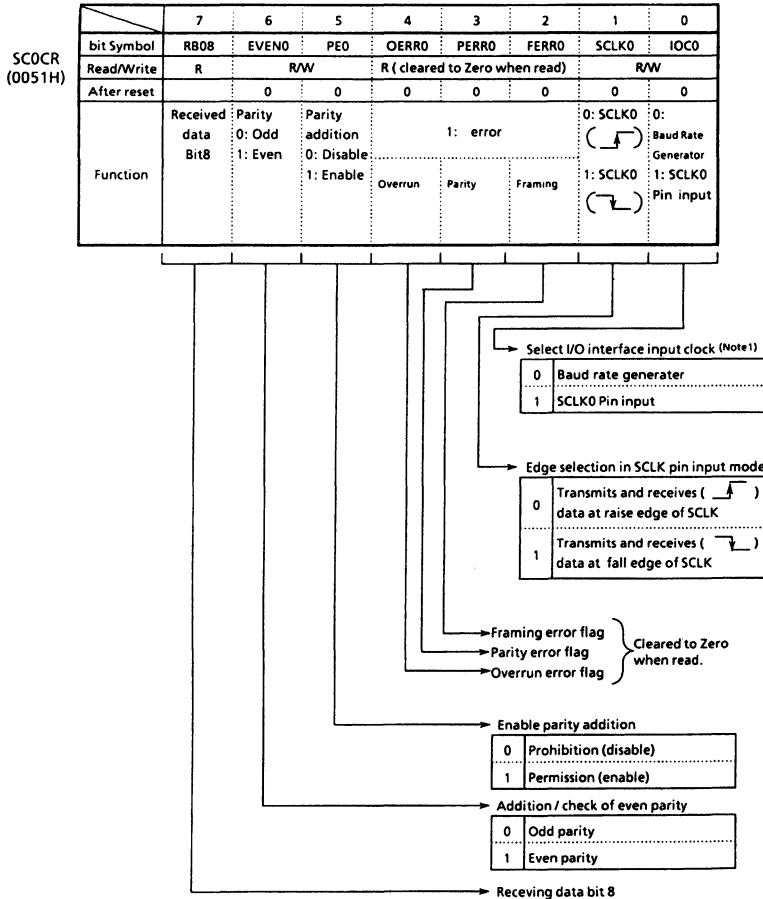
### 3.11.1 Control Registers

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



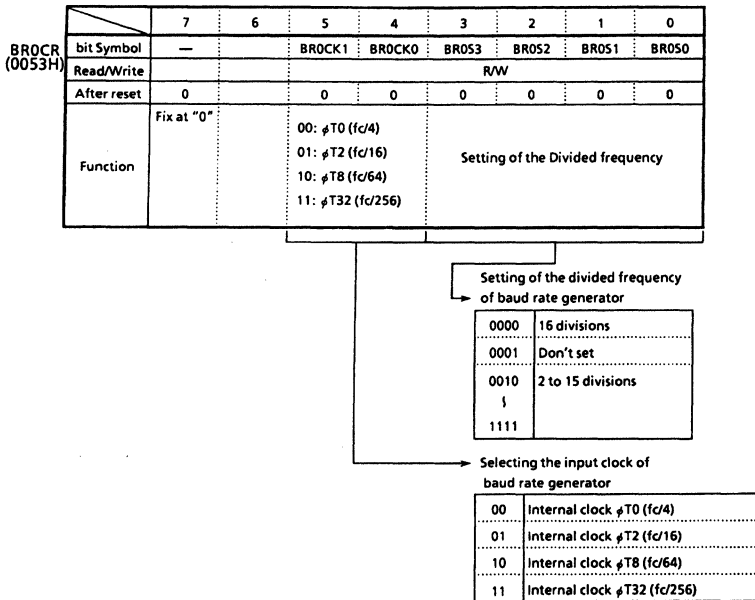
Note : There is SC1MOD (56H) in Channel1

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



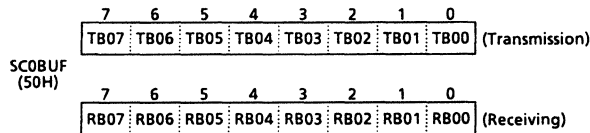
Note : As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

Figure 3.11 (3). Serial Control Register (Channel, SCOCR)



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

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



Note: When setting the serial transfer mode to "9-bit length UART mode", transfer data "TB08" is written to bit7 of SC0MOD register, and received data "RB08" is read from bit7 of SC0CR register.

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



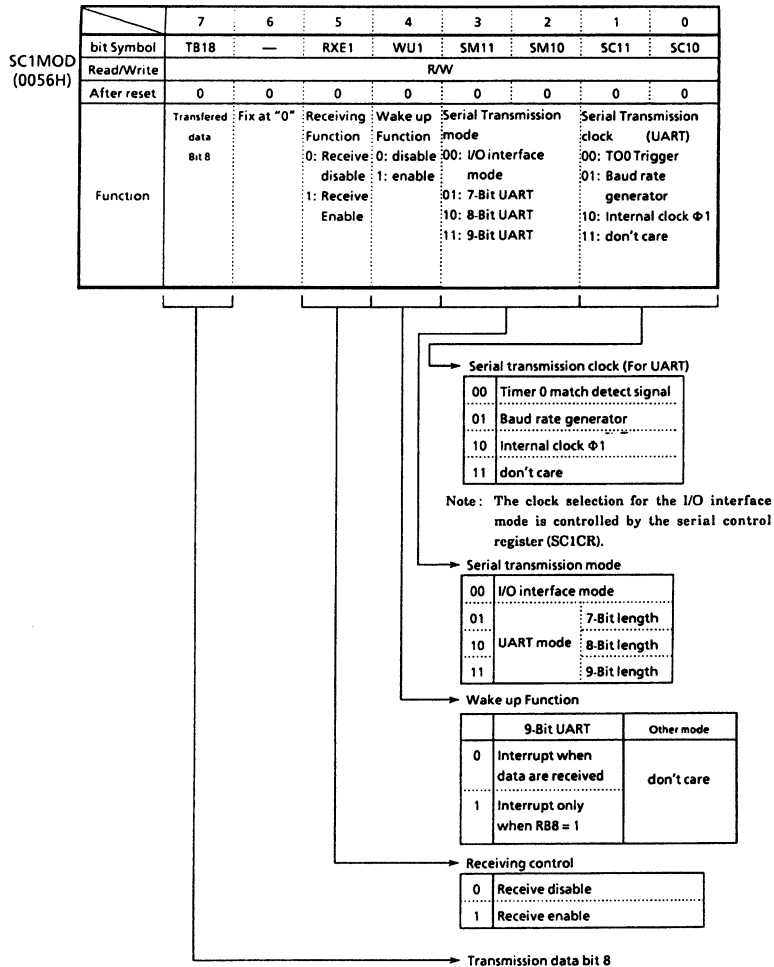
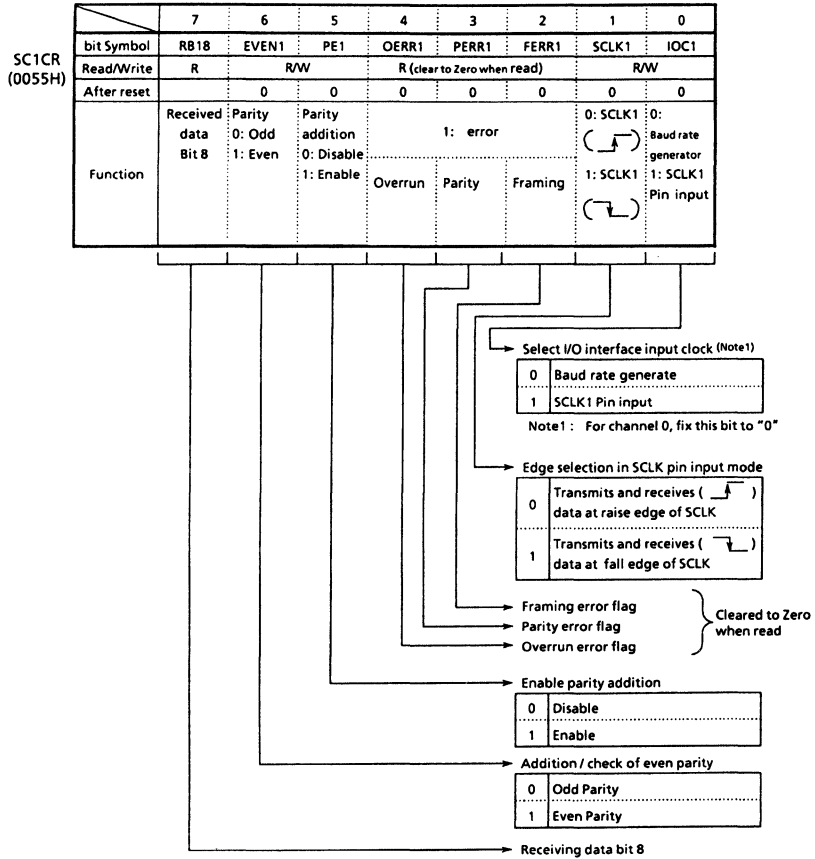
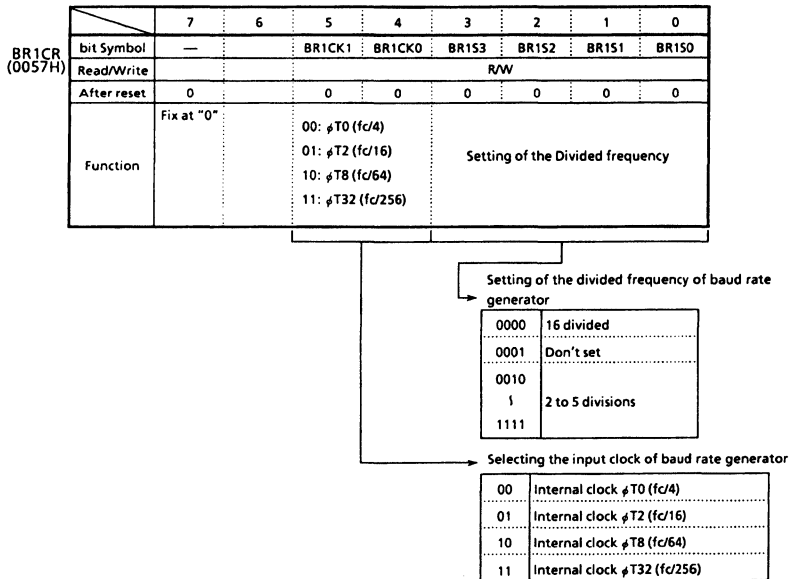


Figure 3.11 (6). Serial Mode Control Register (Channel 1, SC1MOD)



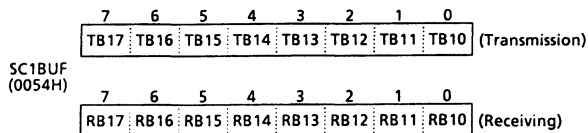
Note : As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

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



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

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



Note : When setting the serial transfer mode to "9-bit length UART mode", transfer data "TB18" is written to bit7 of SC1MOD register, and received data "RB18" is read from bit7 of SC1CR register.

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

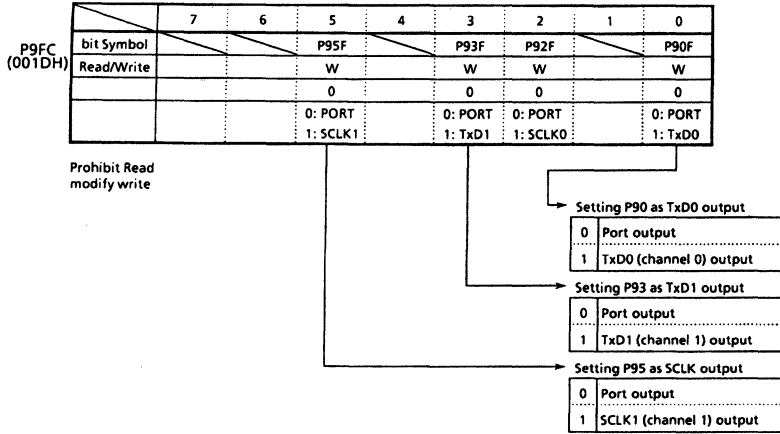
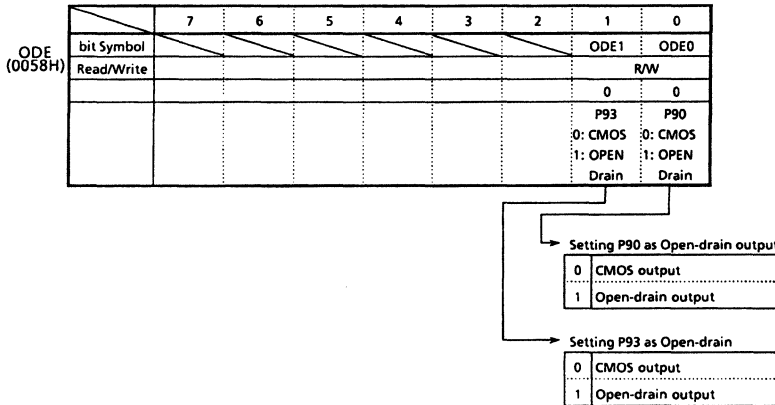


Figure 3.11 (10). Port 9 Function Register (P9FC)



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

3.11.2 Configuration

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

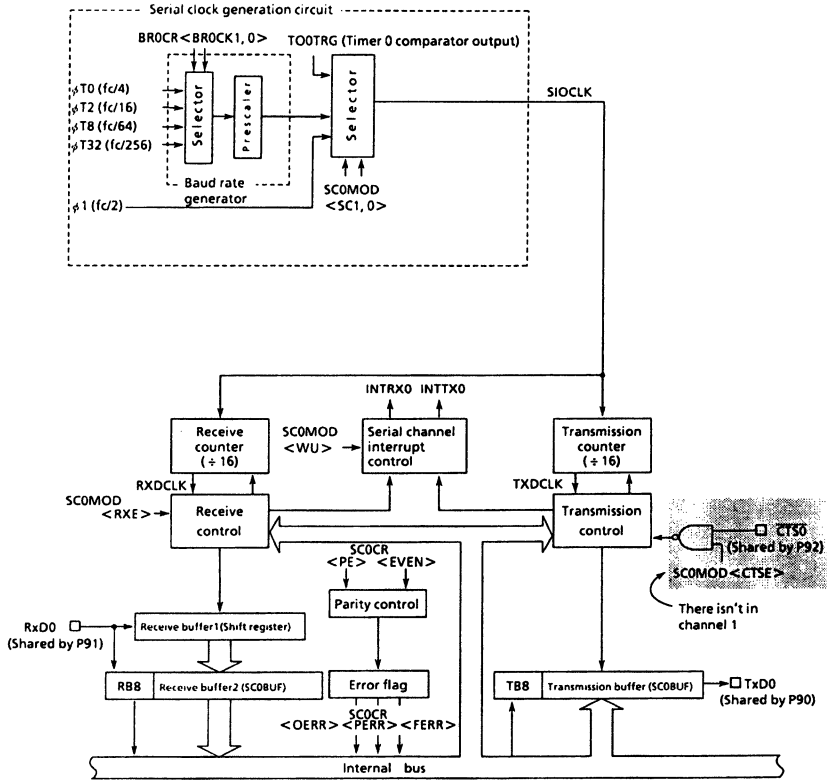


Figure 3.11 (12). Block Diagram of the Serial Channel 0

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

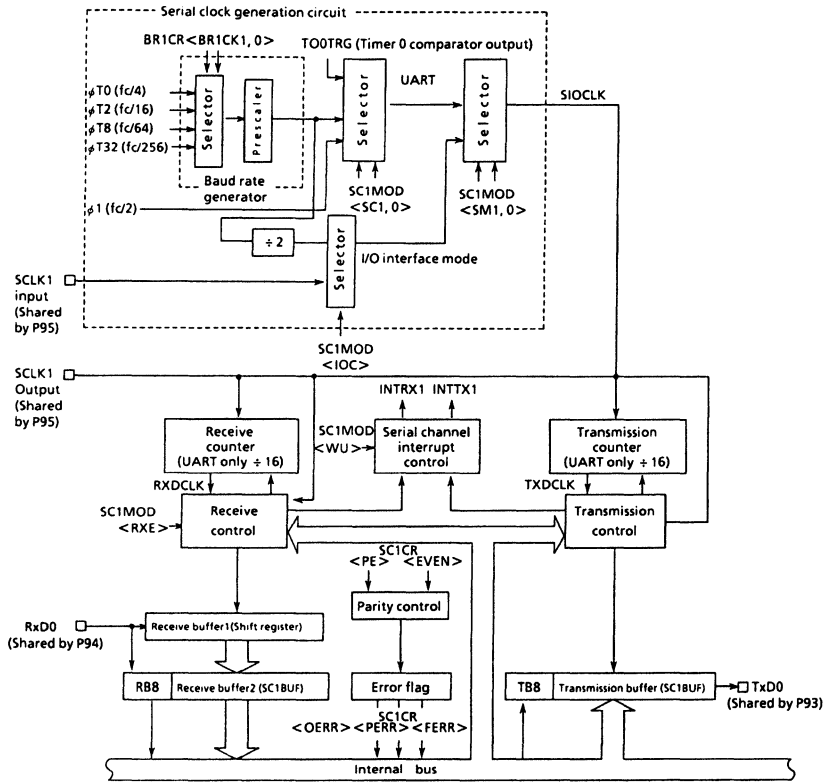


Figure 3.11 (13). Block Diagram of the Serial Channel 1

① Baud Rate Generator

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

The input clock to the baud rate generator,  $\phi T0$  ( $fc/4$ ),  $\phi T2$  ( $fc/16$ ),  $\phi T8$  ( $fc/64$ ), or  $\phi T32$  ( $fc/256$ ) is generated by the 9-bit prescaler which is shared by the timers.

One of these input clocks is selected by the baud rate generator control register 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

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

● I/O interface mode

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

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

$$\phi T0 = fc/4$$

$$\phi T2 = fc/16$$

$$\phi T8 = fc/64$$

$$\phi T32 = fc/256$$

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

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

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

Also with 8-bit timer 0, the serial channel can get a transfer rate. Table 3.11 (2) shows an example of baud rate using timer 0.

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

fc [MHz]	Input Clock					Unit (kbps)
	Frequency Divisor	$\phi T0$ (fc/4)	$\phi T2$ (fc/16)	$\phi T8$ (fc/64)	$\phi T32$ (fc/256)	
9.830400	2	76.800	19.200	4.800	1.200	
↑	4	38.400	9.600	2.400	0.600	
↑	8	19.200	4.800	1.200	0.300	
↑	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 3.11 (2) Selection of Transfer Rate (1) (When Timer 0 (Input Clock  $\phi T1$ ) is Used)**

Unit (Kbps)

TREGO \ 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):

$$\text{Transfer rate} = \frac{f_c}{\text{TREGO} \times 8 \times 16}$$

↑ (When timer 0 (input clock  $\phi T1$ ) is used)

Input clock of timer 0

$$\phi T1 = f_c / 8$$

$$\phi T4 = f_c / 32$$

$$\phi T16 = f_c / 128$$

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.

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

When in SCLK output mode with the setting of SC1CR <IOC> = "0", the basic clock will be generated by dividing by 2 the output of the baud rate generator as described before. When in SCLK input mode with the setting of 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.

2) Asynchronous Communication (UART) mode

According to the setting of SC0CR and SC1CR <SC1, 0>, the above baud rate generator clock, internal clock  $\phi 1$  (500 Kbps @  $f_c = 16$  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. Sixteen pulses of SIOCLK are used for receiving one bit of data, and the data bit is sampled three times at 7th, 8th and 9th clock.

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

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

④ Receiving Control

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

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

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



2) Asynchronous Communication (UART) mode

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

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

⑤ Receiving Buffer

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

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

unless the receiving buffer 2 (SC0BUF/SC1BUF) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of the receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC0CR <RB8> SC1CR <RB8> are still preserved.

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 TxDCCLK every 16 clock pulses.

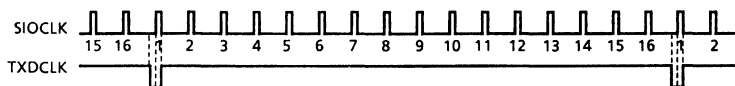


Figure 3.11 (14). Generation of Transmission Clock

⑦ Transmission Controller

1) 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.

2) Asynchronous Communication (UART) mode

When transmission data is written in the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TxDCCLK, 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  $\text{SCOMOD} \langle \text{CTSE} \rangle$ .

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  $\text{INTTX0}$  Interrupts are gener-

ated, 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  $\text{RXD}$  interrupt routine.

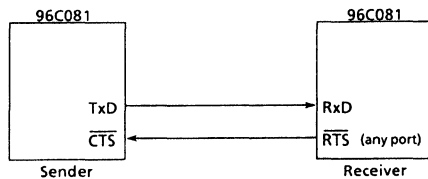
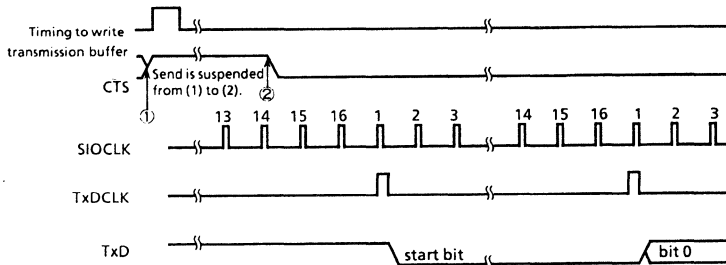


Figure 3.11 (15). Handshake Function



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

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

Figure 3.11 (16). Timing of  $\overline{\text{CTS}}$  (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

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

**Transmitting**

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

2) I/O Interface mode

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

3.11.3 Operational Description

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

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

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

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

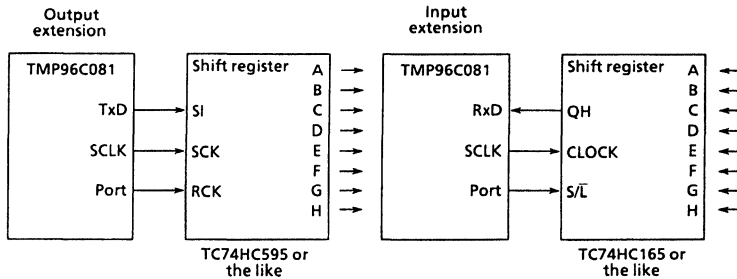


Figure 3.11 (17). Example of SCLK Output Mode Connection

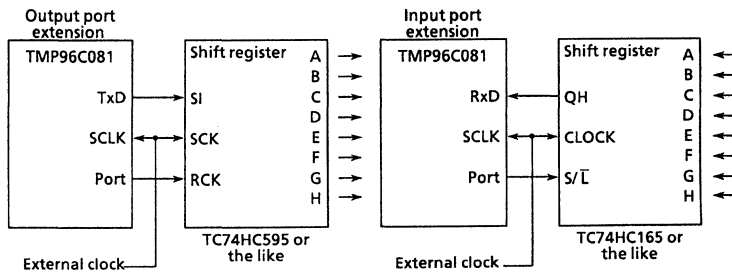


Figure 3.11 (18). Example of SCLK Input Mode Connection

① Transmission

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

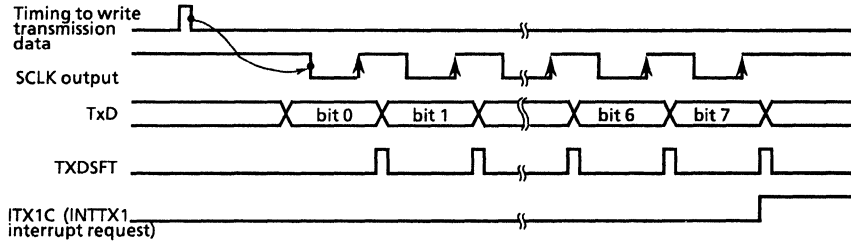


Figure 3.11 (19). Transmitting Operation in I/O Interface Mode (SCLK Output Mode)

In SCLK input 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.

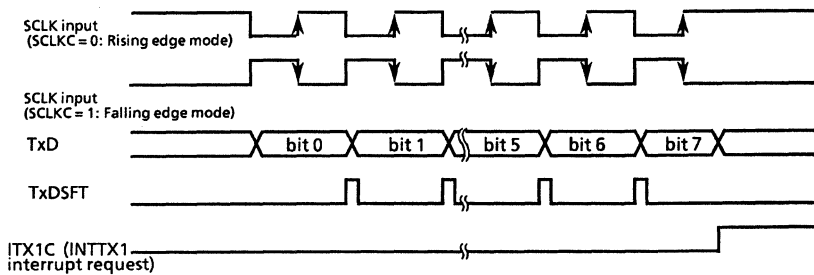


Figure 3.11 (20). Transmitting Operation in I/O Interface Mode (SCLK Input Mode)

② Receiving

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

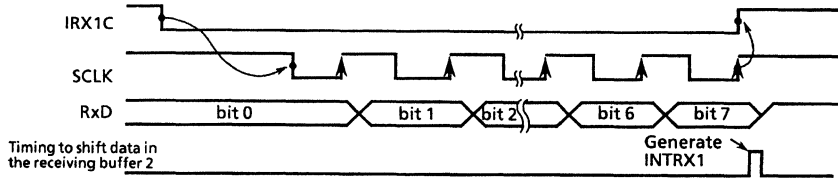


Figure 3.11 (21). Receiving Operation in I/O Interface Mode (SCLK Output Mode)

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

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

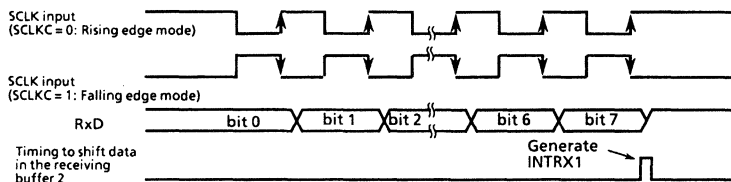


Figure 3.11 (22). Receiving Operation in I/O Interface Mode (SCLK Input Mode)

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

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

The 7-bit mode can be set by setting serial channel mode register SCOMOD <SM01,00>/SC1MOD <SM11,10> 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
P9CR	←	x	x	-	-	-	-	-	1
P9FC	←	x	x	-	x	-	x	x	1
SCOMOD	←	x	0	-	x	0	1	0	1
SC0CR	←	x	1	1	x	x	x	0	0
BROCR	←	0	x	1	0	0	1	0	1
TRUN	←	1	x	-	-	-	-	-	-
INTES0	←	1	1	0	0	-	-	-	-
SC0BUF	←	*	*	*	*	*	*	*	*

- ) Select P90 as the TxD pin.
- Set 7-bit UART mode.
- Add an even parity.
- Set transfer rate at 2400 bps.
- Start the prescaler for the baud rate generator.
- Enable INTTX0 interrupt and sets interrupt level 4.
- Set data for transmission.

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

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

The 8-bit UART mode can be specified by setting SCOMOD <SM01,00>/SC1MOD <SM11,10> 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	
P9CR	← x	x	-	-	-	-	0	-	Select P91 (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.
BROCR	← 0	x	0	1	0	1	0	1	Set transfer rate at 9600 bps.
TRUN	← 1	x	-	-	-	-	-	-	Start the prescaler for the baud rate generator.
INTES0	← -	-	-	-	1	1	0	0	Enable INTTX0 interrupt and sets interrupt level 4.

Interrupt processing

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

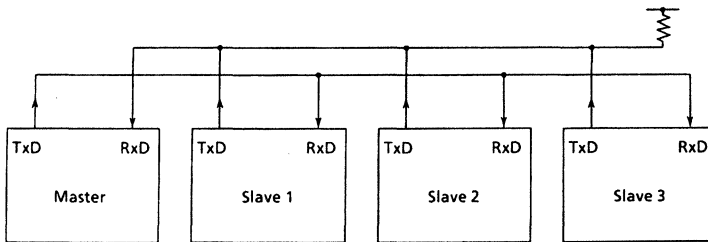
Note: x; don't care -; no change

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

The 9-bit UART mode can be specified by setting SC0MOD <SM01,00>/SC1MOD <SM11,10> to "11". In this mode, parity bit cannot be added. For transmission, the MSB (9th bit) is written in SCMOD <TB8>, while in receiving it is stored in SCCR <RB8>. For writing and reading the buffer, the MSB is read or written first, then SC0BUF/SC1BUF.

Wake-up function

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



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

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

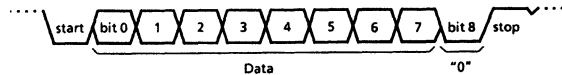


Protocol

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



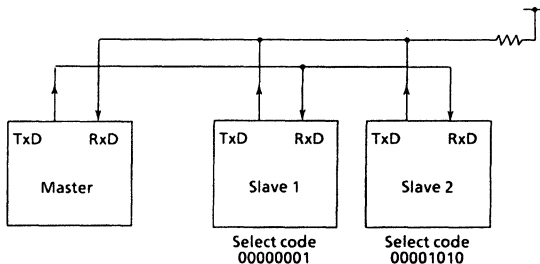
- ④ Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.
- ⑤ The master controller transmits data to the specified slave controller whose SC0MOD <WU>/SC1MOD <WU> bit is cleared to "0." The MSB (bit 8) <TB8> is cleared to "0".



- 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  $\phi 1$  ( $f_c/2$ ) as the transfer clock.



Since serial channels 0 and 1 operate in exactly the

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

• Setting the master controller

Main setting

P9CR	←	x	x	-	-	-	-	0	1
P9FC	←	x	x	-	x	-	x	x	1
INTES0	←	1	1	0	0	1	1	0	1
SCOMOD	←	1	0	1	0	1	1	1	0
SCOBUF	←	0	0	0	0	0	0	0	1

- ) Select P90 as TxD pin and P91 as RxD pin.
- Enable INTTX0 and sets the interrupt level 4.
- Enable INTRX0 and sets the interrupt level 5.
- Set  $\phi 1$  ( $f_c/2$ ) as the transmission clock in 9-bit UART mode.
- Set the select code for slave controller 1.

INTTX0 interrupt

SCOMOD	←	-	0	-	-	-	-	-	-
SCOBUF	←	*	*	*	*	*	*	*	*

- Set TB8 to "0".
- Set data for transmission.

• Setting the slave controller 2

Main setting

P9CR	←	x	x	-	-	-	-	0	1
P9FC	←	x	x	-	x	-	x	x	1
ODE	←	x	x	x	x	x	x	-	1
INTES0	←	1	1	0	1	1	1	1	0
SCOMOD	←	0	0	1	1	1	1	1	0

- ) Select P91 as RxD pin and P90 as TxD pin (open drain output).
- Enable INTRX0 and INTTX0.
- Set <WU> to "1" in the 9-bit UART transmission mode with transfer clock  $\phi 1$  ( $f_c/2$ ).

INTRX0 interrupt

Acc ← SCOBUF  
 If Acc = Select Code  
 Then SCOMOD4

←	-	-	-	-	0	-	-	-	-	Clear <WU> to "0".
---	---	---	---	---	---	---	---	---	---	--------------------

### 3.12 Analog/Digital Converter

The TMP96C081F contains a high-speed analog/digital converter (A/D converter) with 6-channel analog input that features 10-bit successive approximation.

Figure 3.12 (1) shows the block diagram of the A/D converter. 6-channel analog input pins (AN5 to AN0) are shared by input-only P5 and so can be used as input port.

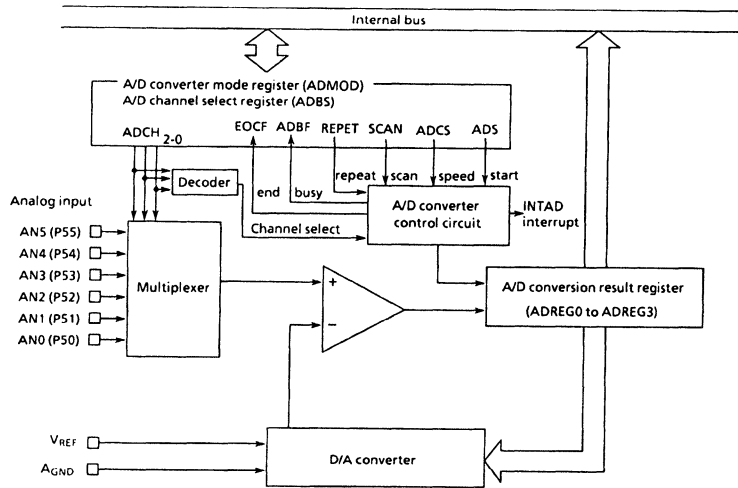


Figure 3.12 (1). Block Diagram of A/D Converter

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

Note 2: In order to reduce power supply current in IDLE or STOP mode, the A/D converter enters standby mode even if the internal comparator remains in the enable state depending on the timing. Stop the A/D converter before executing the HALT command. Since the ladder resistor between V<sub>REF</sub> - A<sub>GND</sub> cannot be stopped internally, I<sub>REF</sub> is transmitted regardless of a mode.

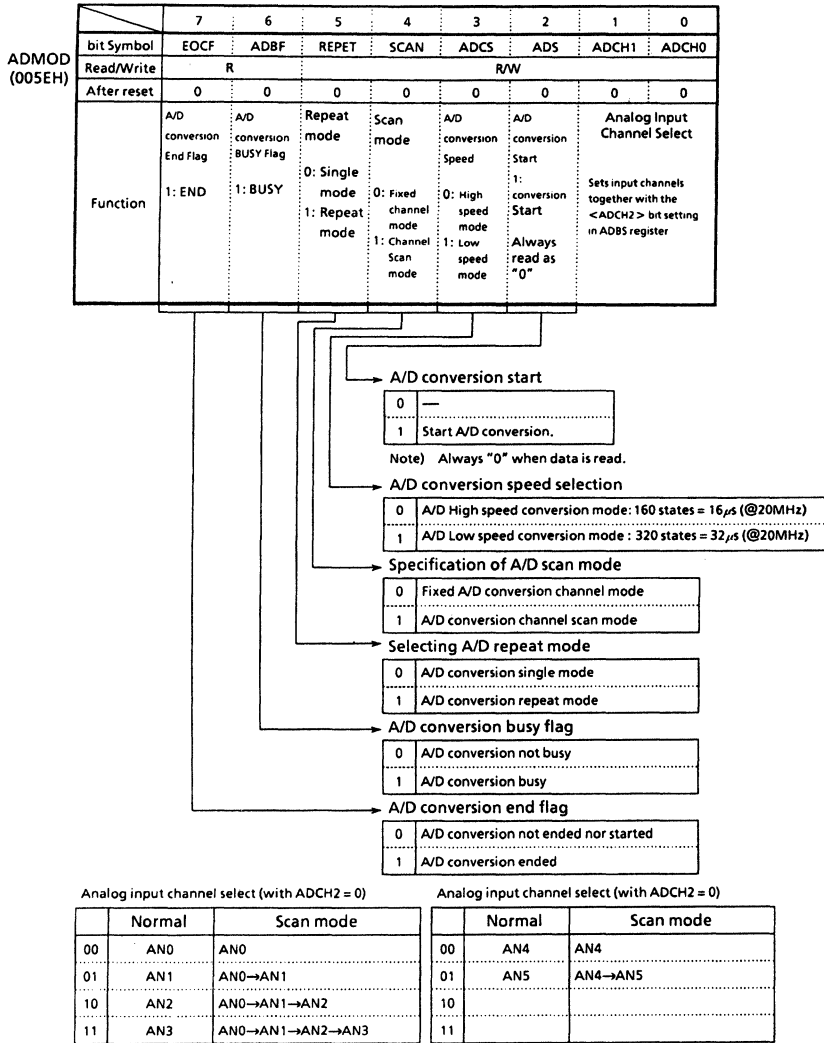


Figure 3.12 (2). A/D Control Register

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
ADREG04L (0060H)	bit Symbol	ADR041	ADR40						
	Read/Write	R							
	After reset	Undefined		1	1	1	1	1	1
	Function	Lower 2 bits of A/D result for AN0 or AN4 are stored.							

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
ADREG04H (0061H)	bit Symbol	ADR049	ADR048	ADR047	ADR046	ADR045	ADR044	ADR043	ADR042
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN0 or AN4 are stored.							

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
ADREG15L (0062H)	bit Symbol	ADR151	ADR150						
	Read/Write	R							
	After reset	Undefined		1	1	1	1	1	1
	Function	Lower 2 bits of A/D result for AN1 or AN5 are stored.							

		<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
ADREG15H (0063H)	bit Symbol	ADR159	ADR158	ADR157	ADR156	ADR155	ADR154	ADR153	ADR152
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN1 or AN5 are stored.							

Figure 3.12 (3-1). A/D Conversion Result Register (ADREG04, 15)

	7	6	5	4	3	2	1	0	
ADREG2L (0064H)	bit Symbol	ADR21	ADR20	/				/	/
	Read/Write	R							
	After reset	Undefined							
	Function	Lower 2 bits of A/D result for AN2 are stored.							

	7	6	5	4	3	2	1	0	
ADREG2H (0065H)	bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN2 are stored.							

	7	6	5	4	3	2	1	0	
ADREG3L (0066H)	bit Symbol	ADR31	ADR30	/				/	/
	Read/Write	R							
	After reset	Undefined	1	1	1	1	1	1	
	Function	Lower 2 bits of A/D result for AN3 are stored.							

	7	6	5	4	3	2	1	0	
ADREG3H (0067H)	bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
	Read/Write	R							
	After reset	Undefined							
	Function	Upper 8 bits of A/D result for AN3 are stored.							

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

	7	6	5	4	3	2	1	0	
ADBS (005FH)	bit Symbol	/							ADCH2
	Read/Write								R/W
	After reset								0
	Function								0: AN0~3 1: AN4~5

→ Analog input channel select

	Normal	Scan mode
0	AN0~3	AN0→AN1→AN2→AN3
1	AN4~5	AN4→AN5

Figure 3.12 (3-3). A/D Channel Select Register

**3.12.1 Operation**

(1) Analog Reference Voltage

High analog reference voltage is applied to the VREF pin, and low analog reference voltage is applied to AGND pin.

The reference voltage between VREG and AGND is divided by 1024 using ladder resistance, and compared with the analog input voltage for A/D conversion.

(2) Analog Input Channels

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

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

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

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

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

- A/D conversion repeat mode

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

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

(7) Storing the A/D Conversion Result

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

(8) Reading the A/D Conversion Result

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

Setting example: 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	-	-	-	-	Enable INTAD and sets interrupt level 4.
ADMOD	←	x	x	0	0	0	1	1	1	Specify AN3 pin as an analog input channel and starts A/D conversion in high speed mode.
INTAD routine										
WA	←	ADREG3								Read ADREG3L and ADREG3H values and writes to WA (16 bit).
WA	>	>	6							Right-shifts WA six times and writes 0 in upper bits.
(00FF10H)	←	WA								Writes contents of WA in memory at FF10H.
When the analog input voltage of AN0 ~ AN2 pins is A/D converted in high speed conversion channel scan repeat mode.										
INTE0AD	←	1	0	0	-	-	-	-	-	Disable INTAD.
ADMOD	←	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

### 3.13 Watchdog Timer (Runaway Detecting Timer)

The TMP96C081F is containing watchdog timer of Runaway detecting.

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

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

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



3.13.1 Configuration

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

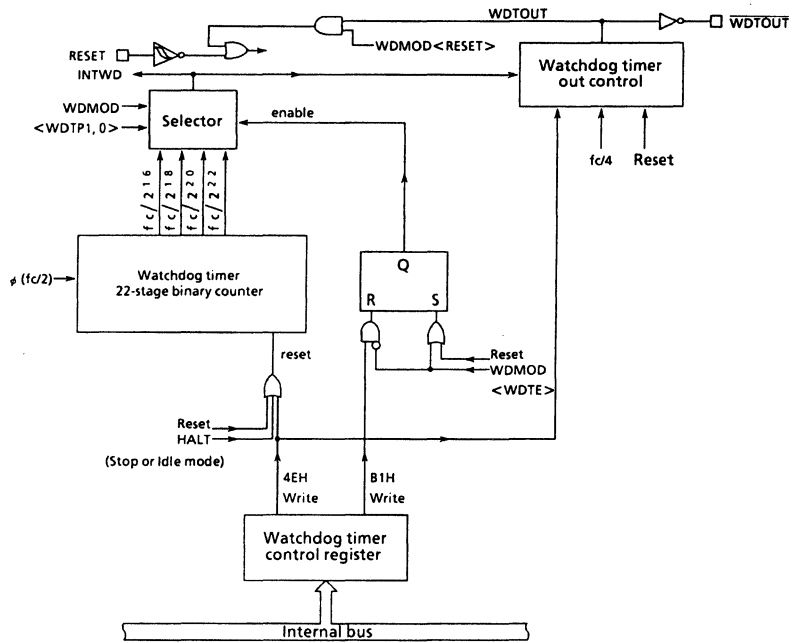


Figure 3.13 (1). Block Diagram of Watchdog Timer

The watchdog timer is a 22-stage binary counter which uses  $\phi$  ( $f_c/2$ ) as the input clock. There are four outputs from the binary counter:  $2^{16}/f_c$ ,  $2^{18}/f_c$ ,  $2^{20}/f_c$ , and  $2^{22}/f_c$ . 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. The watchdog timer out pin is set to 1 by clearing the watchdog timer (by writing a clear code 4EH in the WDCR register). 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 2.0 $\mu$ s @ 20MHz) and resets itself.

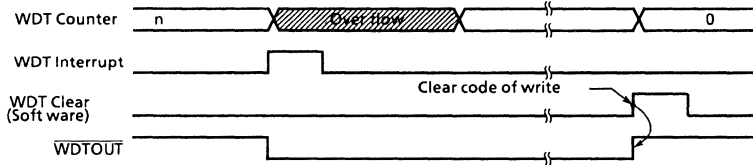


Figure 3.13 (2). Normal Mode

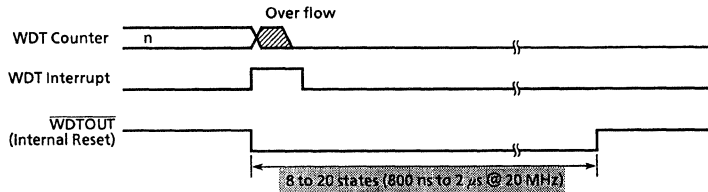


Figure 3.13 (3). Reset Mode

### 3.13.2 Control Registers

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

(1) Watchdog Timer Mode Register (WDMOD)

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

This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD <WDTP1, 0> = 00 when reset, and therefore  $2^{16}/f_c$  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.

- Disable control

WDMOD	←	0	-	-	-	-	-	x	x
WDCR	←	1	0	1	1	0	0	0	1

Clear WDMOD <WDTE> to "0".

Write the disable code (B1H).

- Enable control  
Set WDMOD <WDTE> to "1".

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.

- 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).

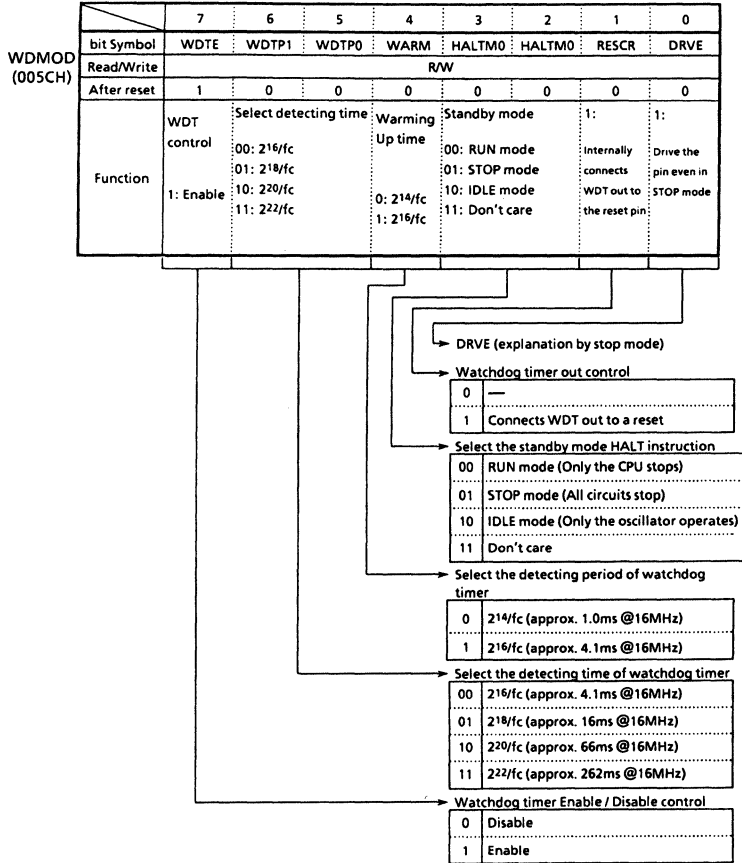


Figure 3.13 (4). Watchdog Timer Mode Register

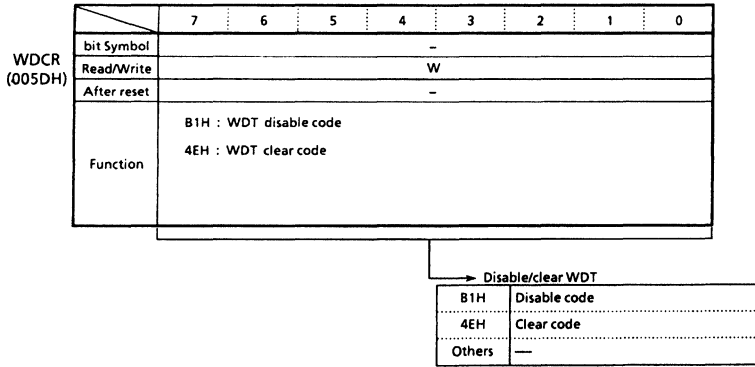


Figure 3.13 (5). Watchdog Timer Control Register

### 3.13.3 Operation

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

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

The watchdog timer restarts operation immediately after resetting is released.

The watchdog timer stops its operation in the IDLE and STOP modes.

The watchdog timer stops when a bus is released or internal DMAC is operating.

Example:

- ① Clear the binary counter

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

- ② Set the watchdog timer detecting time to  $2^{18}/f_c$

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:  $2^{16}/f_c$ )

WDMOD ← - - - 1 0 1 x x      Set the STOP mode.

Executes HALT command

Execute HALT instruction. Set the standby mode.

### 3.14 Direct Memory Access Controller (DMAC)

#### 3.14.1 Outline

The Direct Memory Access Controller (DMAC) is a peripheral circuit used to directly access memory in more than one mode. The DMAC enables direct data transfer between the internal and the external I/Os of the TMP96C081 without interfering with the CPU operation, thus contributing to system efficiency.

The DMAC has four independent built-in channels. Programmable channel control registers support three transfer modes and two address specification methods. Auto-initialization is also supported. This enables repeated DMA transfers, by automatically returning the values previously set in the registers at the completion of one data transfer.

Also supported are:

- increment/decrement of transfer source and destination address
- accesses necessary for the DMAC built into the MCU, such as access to fixed addresses and access in I/O mode (see 3.14.4 (4) Address specification).

Channels have the following capabilities:

maximum number of transfer blocks: 64K words  
 maximum address area: 16M bytes

#### 3.14.2 Features

- Four independent DMA channels
- Transfer speed at 20MHz:
  - Single address transfer:
    - external memory → external I/O, 5M bytes (words)/s
    - external memory → internal I/O, 5M bytes/s
    - internal memory → external I/O, 3.3M bytes/s
  - Dual address transfer:
    - 8 → 8 or 16 → 16 bit transfer, 2.5M bytes (word)/s
    - 8 → 16 or 16 → 8 bit transfer, 3.3 M bytes (word)/s
- Three transfer modes; byte, demand, and continuous
- Transfer between memories or between memories and I/Os
- Independent auto-initialization for each channel
- Programmable address increment/decrement/fixd I/O mode
- Control of enable/disable for each DMA request
- Three methods of ending transfers: completion of specified number of transfers, end of process by external circuit, and data match detect
- DMA request by software
- Report of transfer end to CPU by interrupt
- Channel priority can be rotated
- Transfer between ROM and RAM in different bus sizes

#### 3.14.3 Circuit Configuration and Related Pins

The circuit configuration

Note: An asterisk\* next to the pin name represents a channel number.

- (1)  $\overline{\text{DREQ}}^*$   
Input pin used for DMA start requested by an external circuit.
- (2)  $\overline{\text{DACK}}^*$   
Output pin used respond to DMA start requested by an external circuit.
- (3) TC\*  
Output signal pin used to indicate DMA completion. The specified number of DMA transfers in response to a DMA request is ended.
- (4)  $\overline{\text{EOP}}^*$   
Input pin used to request DMA transfer ended by an external circuit.
- (5) AEN\*  
Output signal used to externally notify DMA in operation.
- (6)  $\overline{\text{IORD}}^*$   
Read signal used for external I/Os.
- (7)  $\overline{\text{IOWR}}^*$   
Write signal used for external I/Os.

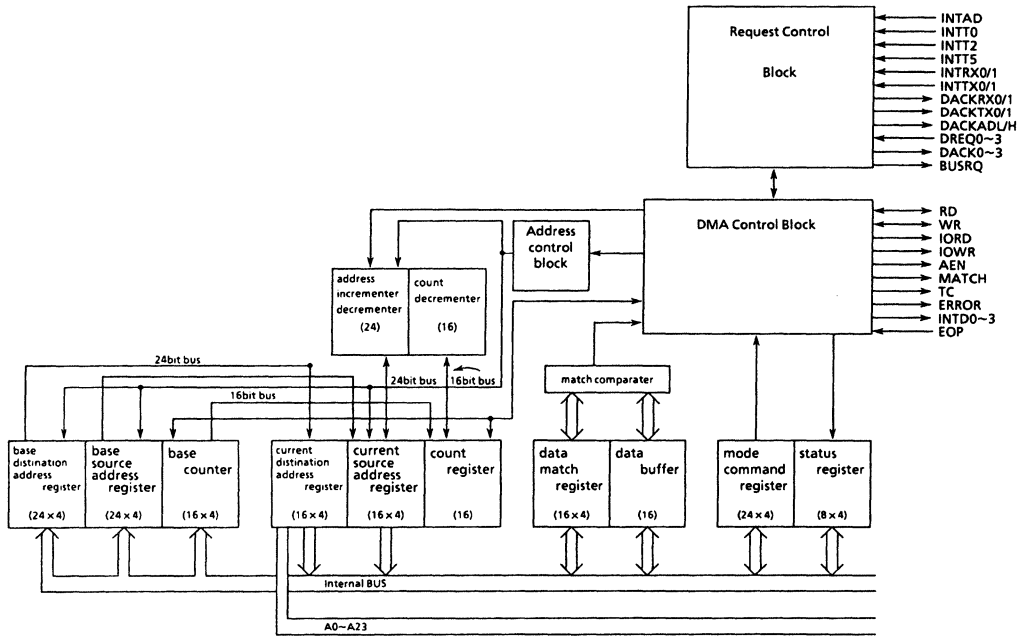


Figure 3.14 (1). DMAC Block



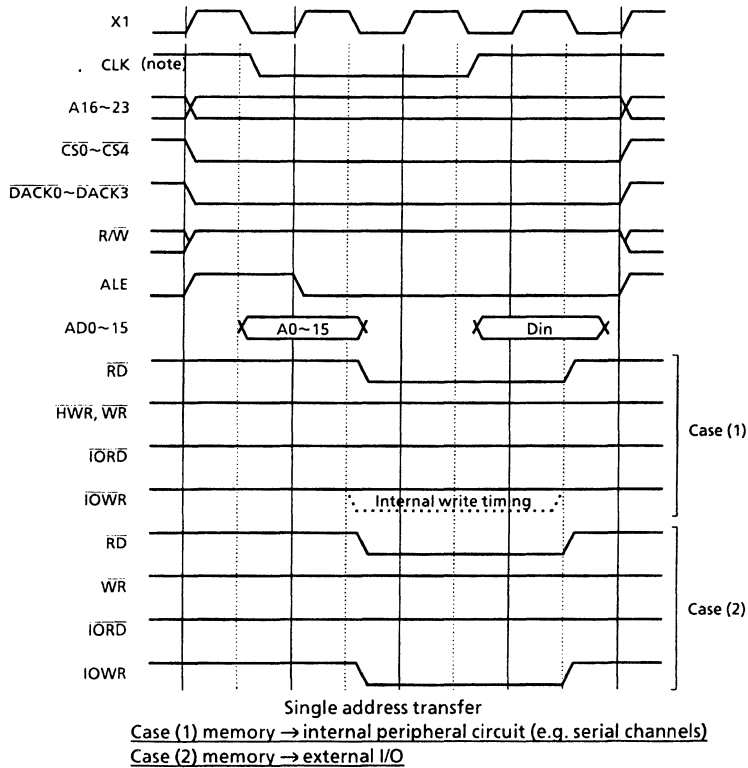
**3.14.4 DMA Operation**

This section describes address specification methods, transfer modes, start/stop methods, and interrupt generation.

(1) Address Specification Methods

- Single address transfer

1-byte or 1-word transfer in 1-bus cycle. Simultaneous read and write from/to memories and internal/external peripheral circuits. When an internal peripheral circuit is accessed, a wait may automatically be inserted to adjust the clock phase. Note that single address transfer is enabled only for SIO0, SIO1, and A/D among internal peripheral circuits.



**Figure 3.14 (1). Single Address Transfer Timing**

Note: CLK output is not necessarily the same as the above phase.

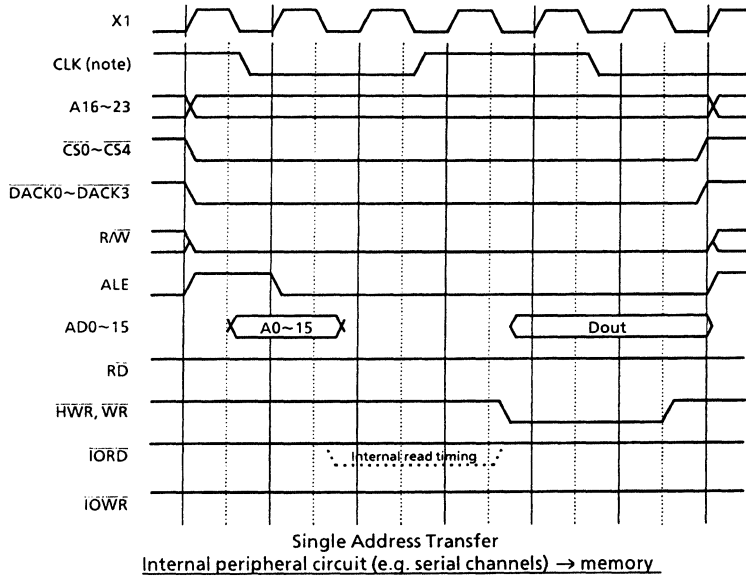
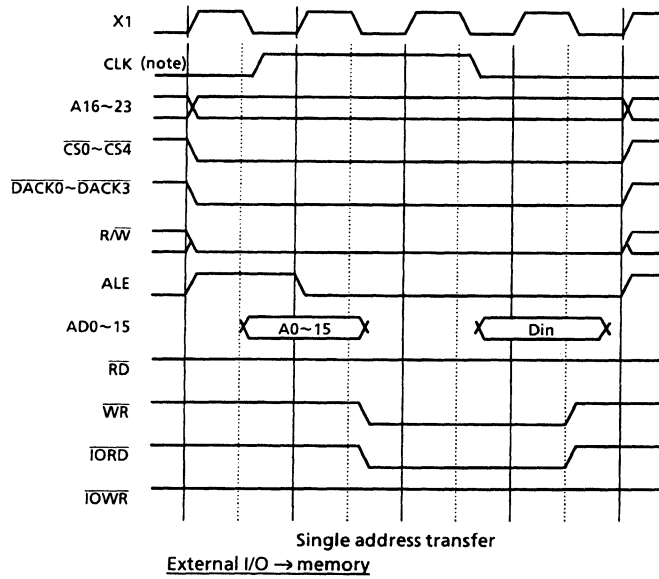


Figure 3.14 (2). Single Address Transfer Timing

Note: CLK output is not necessarily the same as the above phase.



**Figure 3.14 (3). Single Address Transfer Timing**

Note: CLK output is not necessarily the same as the above phase.

- Dual address transfer

1-byte or 1-word transfer in 2-bus cycle consisting of read and write cycles. When an internal peripheral circuit is accessed, a wait may automatically be inserted to adjust the clock phase.

Note that  $\overline{\text{DACK0}}$  to  $\overline{\text{DACK3}}$  signals are output only when  $\overline{\text{DACK}}$  output is enabled. For details, see (10)  $\overline{\text{DACK}}$  output enable bit.

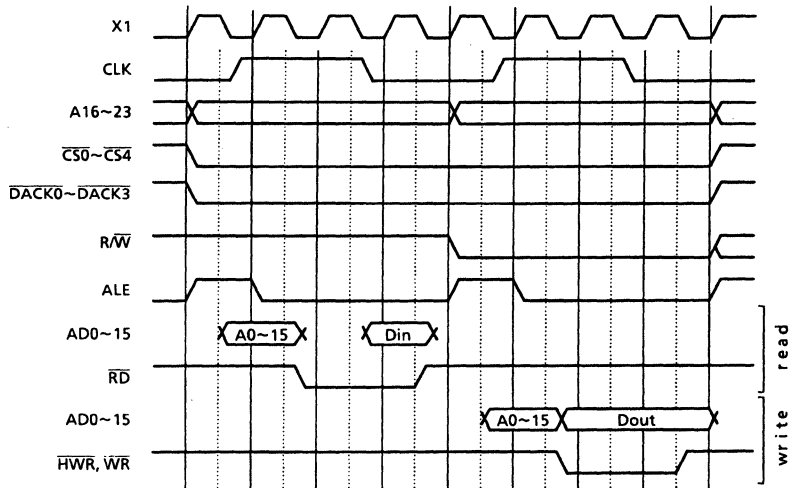


Figure 3.14 (4). Dual Address Transfer Timing

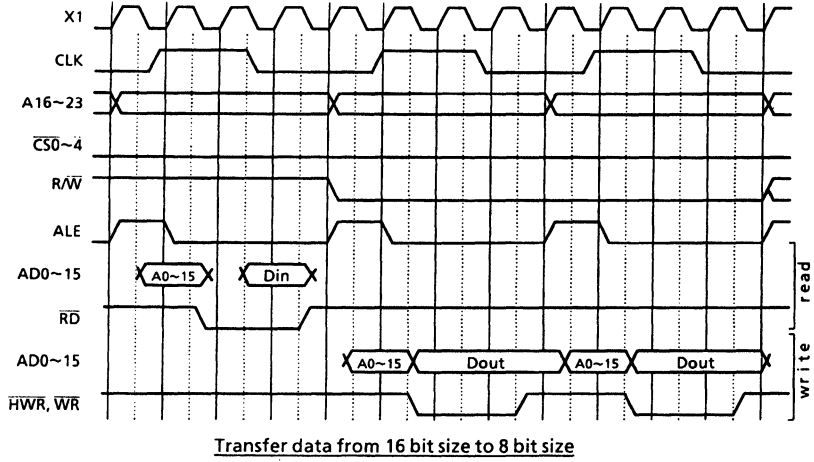


Figure 3.14 (5). Dual Address Transfer Timing

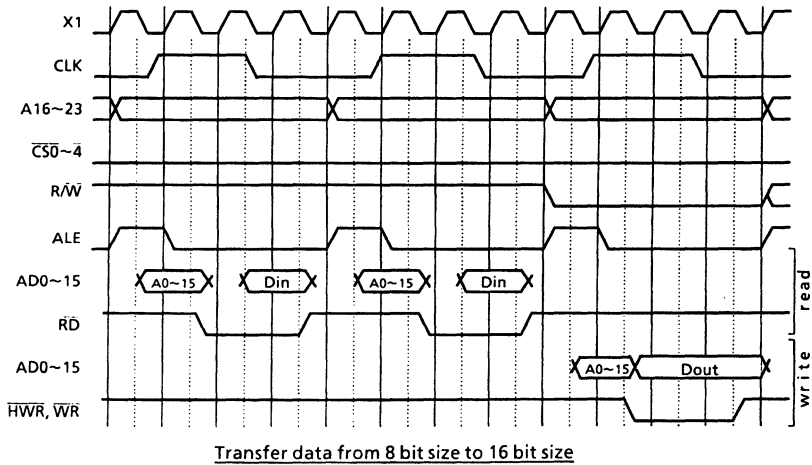


Figure 3.14 (6). Dual Address Transfer Timing

## (2) Transfer Modes

The DMAC supports the following three transfer modes.

- Byte mode

In this mode, bus mastership is released after each transfer in response to a DMA request and when  $\overline{\text{DREQ}}$  (DMA request input) is sensed. If  $\overline{\text{DREQ}}$  is active, bus mastership is re-captured to resume DMA transfer; if  $\overline{\text{DREQ}}$  is inactive, DMA transfer ends.

- Continue mode

In this mode, the specified number of DMA transfers is executed by one DMA request.

- Demand mode

In this mode, DMA transfers continue as long as  $\overline{\text{DREQ}}$  by one DMA request. Transfer ends when  $\overline{\text{DREQ}}$  becomes inactive.

(Note) When starting DMA transfer using a built-in peripheral circuit which can be connected internally, the  $\overline{\text{DREQ}}$  detect method (pulse or level) must be set depending on the output waveform of the interrupt signal to be connected.

## (3) DMA Transfer Start and Stop Methods

- Start

The DMAC can be started using any of the following three methods:

1.  $\overline{\text{DREQ}}$  input by external peripheral circuit  
Inputting  $\overline{\text{DREQ}}$  to the external  $\overline{\text{DREQ}}$  pin corresponding to a channel starts DMA transfer.
2.  $\overline{\text{DREQ}}$  input by internal peripheral circuit  
Selecting the internal  $\overline{\text{DREQ}}$  input corresponding to a channel and ending the anticipated operation starts DMA transfer.
3. Software request  
Setting the DSREQ\* DMA start request bit in the DMACR\* command register for a channel starts transfer.

- Stop

The DMAC can be stopped in any of the following three methods:

1. Completion of a set number of DMA transfers (terminal count: TC)  
When the number of data transfers set in the count register for a channel is completed, the DMAC releases bus mastership and outputs the TC signal. Interrupt generation and setting the TC\* (\* = channel number) bit in the status register notify the CPU of DMAC stop.
2. End of process signal input by external circuit (End of process: EOP)  
Inputting the EOP signal to a channel in DMA transfer ends transfer and releases bus mastership. Interrupt generation and setting the EOP\* (\* = channel number) bit in the status register notify the CPU of DMAC stop.
3. Data match detected (match: MATCH)  
Detecting a match between the data set in the match register for a channel and the data to be transferred ends transfer. Interrupt generation and setting the MAT\* (\* = channel number) bit in the status register notify the CPU of DMAC stop.

## (4) DMA Transfer Start and Stop Methods

The DMAC supports four methods for updating the transfer source and destination address:

1. Fixed mode:  
Used to access address set in the transfer source and destination address registers.
2. Increment mode:  
Used to sequentially increment addresses set in the transfer source and destination address registers: DMASAA/B/C\* and DMADAA/B/C (\* = channel number).
3. Decrement mode:  
Used to sequentially decrement addresses set in the transfer source and destination address registers: DMASAA/B/C\* and DMADAA/B/C (\* = channel number).
4. I/O mode:  
Used to access 2-byte data repeatedly. Useful for accessing internal peripheral circuits

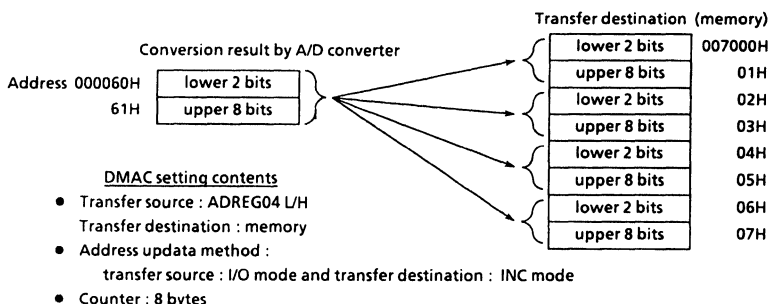


Figure 3.14 (5). Example: Setting Transfer Source Address Update Method to I/O Mode

(5) Transfer Error Generation

The DMAC can notify the CPU that a transfer is accepted by setting 0 in the count register. When 0 is set in the count register, accepting a transfer request causes an error to occur in the DMAC. In this case, the DMAC does not perform DMA transfer. It generates an interrupt and sets the ERR\* bit (\* = channel) in the status register. After an error occurred in the DMAC, DMA transfer can be enabled by the following steps:

- (1) set a value other than 0 in the count register.
- (2) reset the error bit in the status register.
- (3) set the DMA enable bit.

However, if the DMA request source disappears during step (1) to (3), DMA transfer will not be performed.

(6) Auto-initialization

When DMA transfer stops (explained in (3), DMA transfer start and stop methods), and RLD\*, the auto-initialization bit in the DMACR\* command register for a channel is set, the pre-transfer data is returned to the registers. One of the advantages of auto-initialization is that it eliminates the need to set the DMA control registers when the same address is repeatedly accessed. Auto-initialization is used for count registers, and trans-

fer source and destination address registers.

Usually, the DMAEN\* bit in the DMACR\* command register is cleared after a channel ends transfer. However, if RLD is set and DMA transfer stops due to interrupt disable (set in the DMACR command register), the DMAEN\* bit will not be cleared. In this case, when a DMA request is generated, DMA transfer restarts without setting DMA enable again.

(7) Channel Priority Setting

The DMAC supports the following two methods for setting channel priority. Priority is set in the PRI bit of the DMAMODA3 mode A register.

1. Fixed method  
Used to assign fixed priorities to four channels. Channel 0 has the highest priority: next highest is channel 1, then channel 2, and channel 3 has the lowest priority.
2. Rotate mode  
Used to change the priorities of four channels. In this mode, when DMA transfer ends for a channel, the priority of that channel becomes the lowest, then the next channel has the highest priority.

## (8) Interrupt Generation

The DMAC can notify the CPU of an event such as transfer end by generating an interrupt. The DMAC supports the following four interrupt sources. It can set interrupt generation enable/disable using the DMACR\* (\* = channel number) command register.

## 1. Match (MATCH)

Data set in the match register matched the data being transferred.

## 2. Terminal count (TC)

Number of transfers set in the count register has ended.

## 3. End of process (EOP)

Transfer ended by end of process input externally.

## 4. Error (ERROR)

DMA start request is generated with 0s in the count register and an error has occurred in the corresponding channel.

When an interrupt is generated, this is reflected in the DMAST\* status register. Thus, writing 1 in DMAST\* clears the status.

(9) DACK Output Enable Bit

Bit 7 in the DMACR\* command register for channels 0 to 2 is used to set whether to send a response (DACK) when a DMA request (DREQ) is input to a channel. DACK signal output can be classified as shown in the table below, depending on the combination of address specification method and transfer mode.

Bit 7 in the DMAC3\* command register for channel 3 is used to set DMAC channel priority. Thus, DACK signal output cannot be set for channel 3. Instead, channel 3 supports DACK output using 0 set in bit 7 of other channels.



Table 3.14 (7) DAMC Register Setting and Address and DACK Output

DMA start source	DACK output select MODA bit7	MIO/IOM MODA bit3	Sng/Dua1 MODA bit2	Internal connection /no connection MODB bit4	Output address	Internal DACK address	External DACK output
	0 : no output 1 : output	0 : IO → M 1 : M → IO	0 : Dual 1 : Sngl	0 : no connection 1 : connection	SA : Transfer source DA : destination	○ : output	○ : output
External DREQ	0	0	0	0	SA/DA	x	x
	1	0	0	0	SA/DA	x	○
	0	0	1	0	DA	x	○
	1	0	1	0	DA	x	○
	0	1	0	0	SA/DA	x	x
	1	1	0	0	SA/DA	x	○
	0	1	1	0	SA	x	○
Internal DREQ	0	0	0	1	SA/DA	x	x
	1	0	0	1	SA/DA	x	○
	0	0	1	1	DA	○	x
	1	0	1	1	DA	○	○
	0	1	0	1	SA/DA	x	x
	1	1	0	1	SA/DA	x	○
	0	1	1	1	SA	○	x
Software request	0	0	0	0	SA/DA	x	x
	1	0	0	0	SA/DA	x	○
	0	0	1	0	DA	x	○
	1	0	1	0	DA	x	○
	0	1	0	0	SA/DA	x	x
	1	1	0	0	SA/DA	x	○
	0	1	1	0	SA	x	○
	1	1	1	0	SA	x	○

- Notes :
- Internal DREQ signal is used by the peripheral circuit specified by bits 7 to 5 in the mode B register. Internal DACK signal is only output at SIO0 send request, SIO0 receive request, SIO1 send request, SIO1 receive request, and A/D conversion end.
  - To start the DMA by an external  $\overline{DREQ}$  or software request, set the internal connect bit (bit 4 in the mode B register) to 0.
  - Do not set <PAFC> for  $\overline{DREQ}$  function with software request start.

**3.14.5 DMAC Control Registers**

The TMP96C081F has four built-in independent channels, each of which has 14 registers belonging to eight register groups. Channel functions are almost the same. Channel-specific functions specific to channels are also explained below.

When setting the following (1) to (8) registers, the TMP96C081F must be in DMA disable state (bit DMAEN\* in the DMACR\* command register = 0).

- (1) Source Address Register, Common to All Channels  
(\*: channel number)

Used to specify the start address of the data transferred by the DMAC, and memory-mapped I/O register addresses. To transfer 16-bit data, specify even-numbered addresses.

DMASAA\* (Transfer source address : lower 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMASAA*	SA07*	SA06*	SA05*	SA04*	SA03*	SA02*	SA01*	SA00*

DMASAB\* (Transfer source address : middle 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMASAB*	SA15*	SA14*	SA13*	SA12*	SA11*	SA10*	SA09*	SA08*

DMASAC\* (Transfer source address : upper 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMASAC*	SA23*	SA22*	SA21*	SA20*	SA19*	SA18*	SA17*	SA16*

- (2) Destination Address Register, Common to All Channels  
(\* = Channel Number)

Used to specify the transfer destination address. To transfer 16-bit data, specify even-numbered addresses.

DMADAA\* (Transfer destination address : lower 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMADAA*	DA07*	DA06*	DA05*	DA04*	DA03*	DA02*	DA01*	DA00*

DMADAB\* (Transfer destination address : middle 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMADAB*	DA15*	DA14*	DA13*	DA12*	DA11*	DA10*	DA09*	DA08*

DMADAC\* (Transfer destination address : upper 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMADAC*	DA23*	DA22*	DA21*	DA20*	DA19*	DA18*	DA17*	DA16*

- (3) Count Register, Common to All Channels (\* = Channel Number)

Used to specify the number of bytes for 8-bit DMA

transfer or the number of words for 16-bit DMA transfer. If the data width of the transfer source differs from the transfer destination, the data width of the transfer source is used for counting data.

DMACTA\* (Count register : lower 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMACTA*	CT07*	CT06*	CT05*	CT04*	CT03*	CT02*	CT01*	CT00*

DMACTB\* (Count register : upper 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMACTB*	CT15*	CT14*	CT13*	CT12*	CT11*	CT10*	CT09*	CT08*

- (4) Match Register, Common to All Channels (\* = Channel Number)

Used to specify data for stopping DMA transfer if the specified data is accessed during DMA transfer. For 8-

bit to 8-bit transfer, the upper 8-bits are not compared but ignored.

If the data width of the transfer source differs from the transfer destination, the data width of the transfer source is used for counting data.

DMAMAA\* (Match data : lower 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMAMAA*	MA07*	MA06*	MA05*	MA04*	MA03*	MA02*	MA01*	MA00*

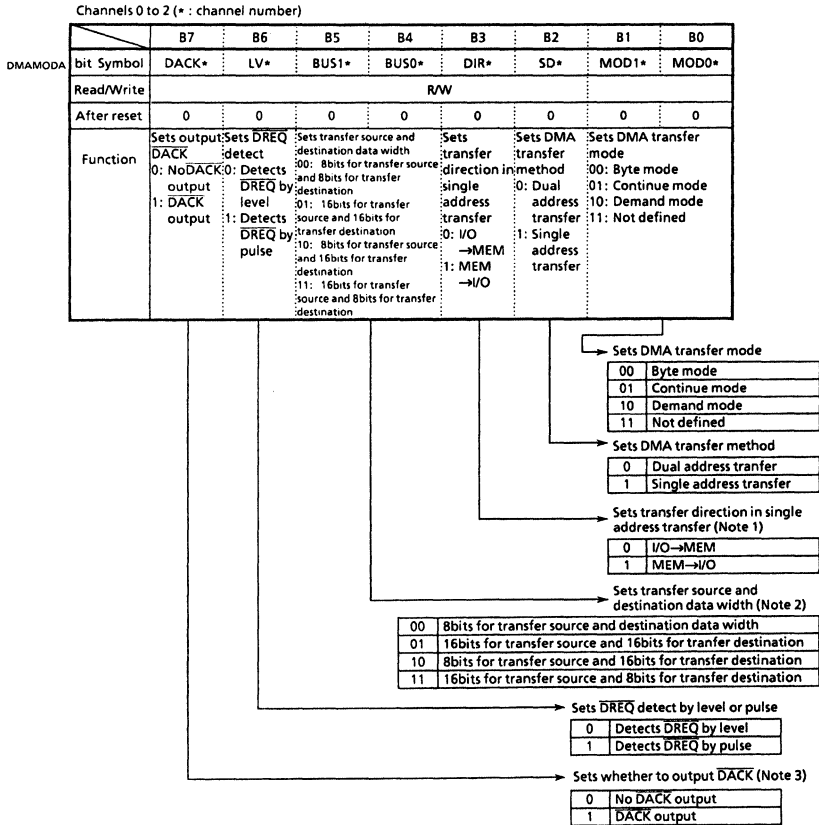
DMAMAB\* (Match data : upper 8 bits)

Register name	B7	B6	B5	B4	B3	B2	B1	B0
DMAMAB*	MA15*	MA14*	MA13*	MA12*	MA11*	MA10*	MA09*	MA08*

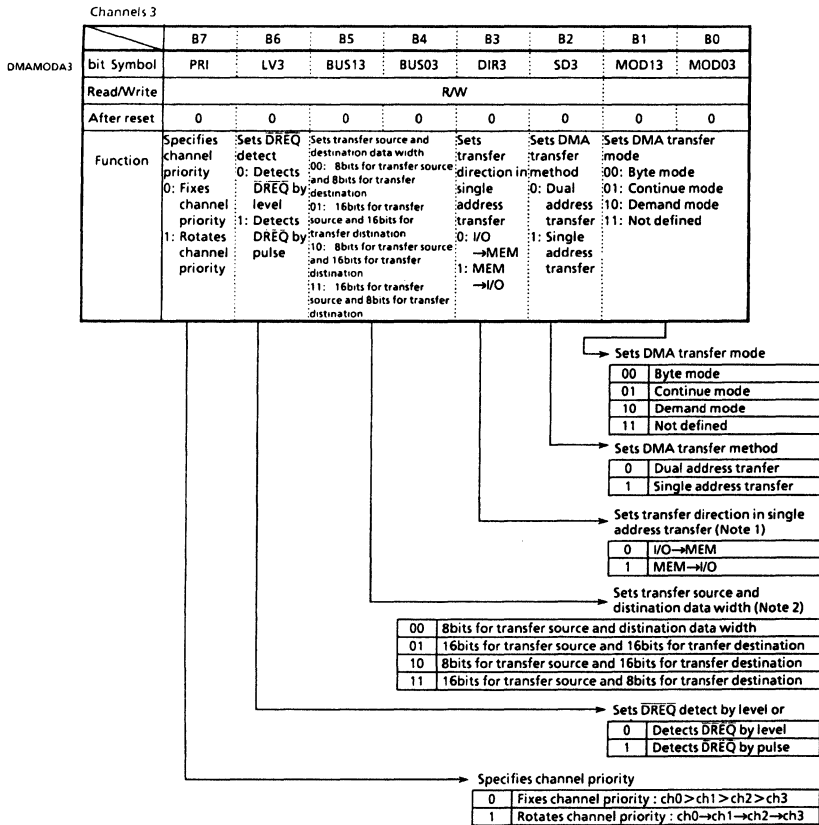
- (5) Mode Register A, Common to Channels 0 to 2; Excluding Channel 3

Used to specify transfer mode, data width, and transfer direction. With channels 0 to 2, bit 7 is used to specify

whether to output DACK to request source at DMA transfer by an external DREQ. With channel 3, bit 7 is used to specify channel priority specification method for the DMAC; that is, DACK output cannot be set.



(Note1) The DIR\* bit is significant only when single address transfer is specified in the SD\* bit below.  
 (Note2) Operations depend on transfer mode. For details, see Table 3.14 (7), DMAC register setting and address/DACK output.  
 (Note3) Set the transfer source and destination data widths to same as those in the CS/w controller. When transferring data to/from internal memory mapped I/O (including internal I/O in single address transfer). Set the transfer data width to 8bits.  
 (Note4) Set the transfer source and destination data widths to same as those when single address transfer is specified.



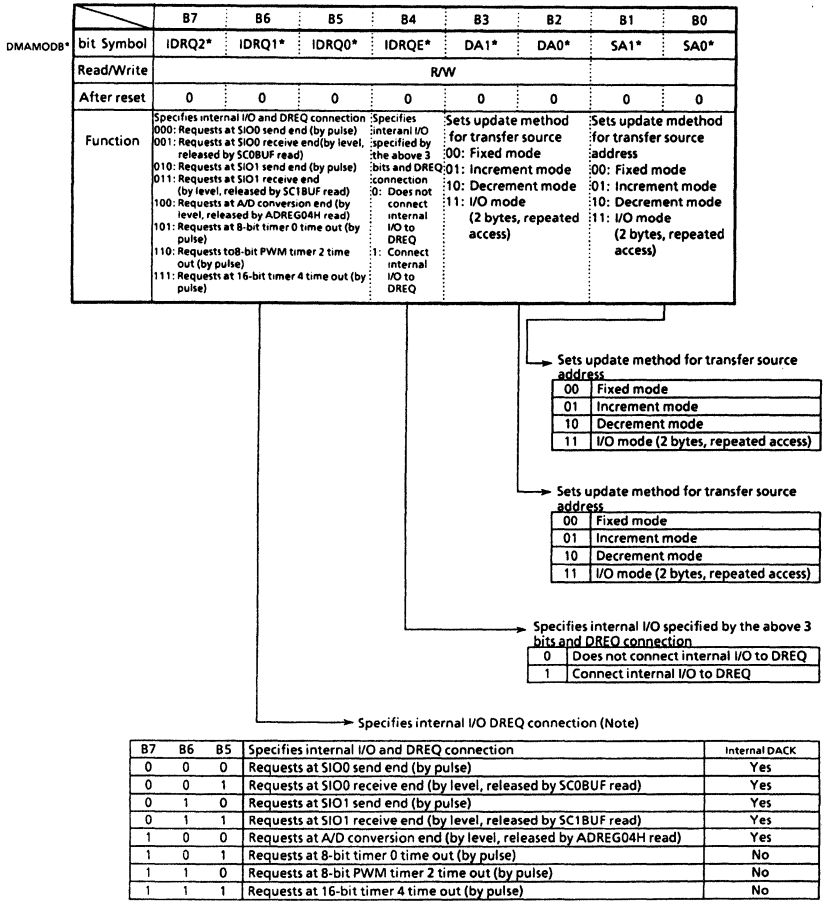
(Note1) The DIR\* bit is significant only when single address transfer is specified in the SD bit below.

(Note2) Set the transfer source and destination data widths to same as those in the CS/W controller. When transferring data to/from internal memory mapped I/O (including internal I/O in single address transfer), set the transfer data width to 8bits.

(6) Mode Register B, Common to All Channels

internal I/O and DMA start request; also, the method for updating transfer source and destination address.

Used to control connection between the TMP96C081

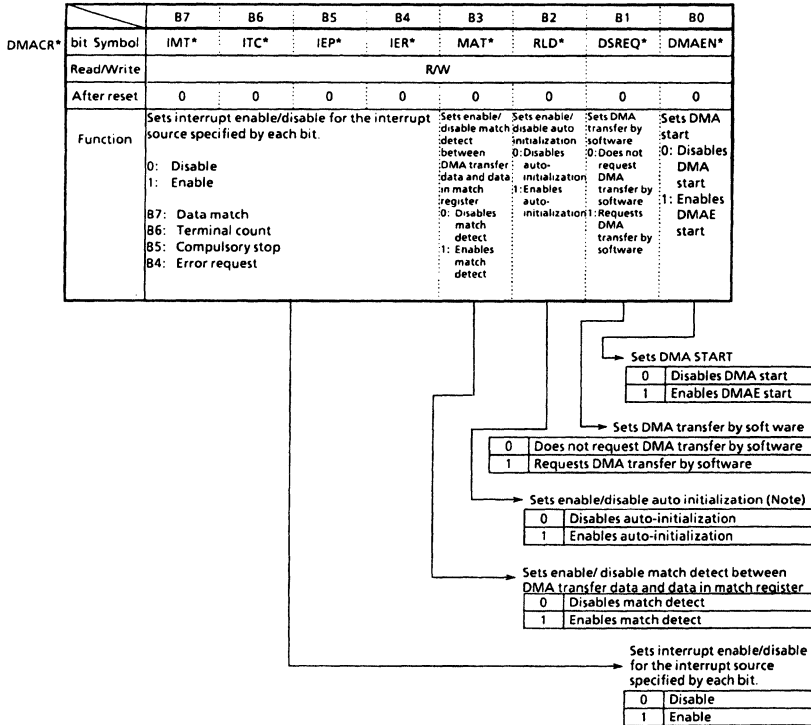


(Note) For a request by pulse, set the LV\* bit in the DMAMODA\* register to 1; for a request by level, to 0. Since DACK for A/D conversion end is connected to CH0, use CH0 for DMA transfer. AN0 and AN4 of A/D converter, use AN0 or AN4 for DMA transfer. (selectable using AD85 register)

(7) Command Register, Common to All Channels

tion, or software request.

Used to control interrupt enable/disable, auto-initializa-



(Note) When auto-initialization is set, start of auto-initialization can be selected depending on the interrupt source. (More than one interrupt source can be specified)  
For interrupt source for which auto-initialization is enabled, to disable interrupts, sets 0s in bits 7 to 4 in the DMACR\* ; for interrupt source for which auto-initialization is disabled, to enable interrupts, set 1s.

- (8) Status Register B, Common to All Channels  
(\* = Channel Number)

Used to display DMA stop source or error generation, or whether there are any sources held for internal/external DREQs.

Writing 1 in the EOP\*, MAT\*, TC\*, or ERR\* bit clears the DMA transfer stop source. Once EN\* bit is latched, it is held until the status is cleared or reset. When  $\overline{DREQ}$  is detected by level (by setting the LV bit in the mode A\* register), the EN\* bit must be cleared in order to determine the current DMA request before the status register.

	B7	B6	B5	B4	B3	B2	B1	B0
DMACR*				EN*	EOP*	MAT*	TC*	ERR*
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function				Displays whether DMA start source is held 0: Start source not held 1: Start source held	Displays whether DMA transfer stop source is generated 0: Stop source not generated 1: Stop source generated			Displays whether error is generated 0: Error not generated 1: Error generated

\* (Write function is available only in bit clear mode)

Displays whether error is generated

0	Disables DMAE start
1	Enable DMAE start

Displays whether DMA transfer stop source is generated

0	Stop source not generated
1	Stop source generated

Displays whether DMA start source is held

0	Error not generated
1	Error generated

(Note) This bit does not influence software requests.



Examples of program settings using DMAC

(a) Initial setting

LD	SP	, 0200H	Sets stack pointer to 200H
MAX			Sets maximum mode
LD	(B1CSL)	, 93H	$\overline{CS}/\overline{CAS}$ ENABLE, 8bit Bus & 0 wait
LD	(B1CSH)	, 00H	Addresses 100H to 7FFFH
LD	(P4CR)	, 1FH	output set
LD	(P4FC)	, 1FH	$\overline{CS}/\overline{CAS}$ set
LD	(P10CR)	, 02H	output set
LD	(P10FC)	, 03H	$\overline{DACK0}$ , $\overline{DREQ0}$ set
LD	(P11CR)	, 10H	output set
LD	(P11FC)	, 1FH	$\overline{IORD}$ , $\overline{IOWR}$ , $\overline{TC EOP}$ , AEN set
LD	(INTDMA0)	, 06H	Sets DMAC ch0 interrupt request level to 6
EI		6	Sets CPU interrupt receive level to 6

(b) Single address transfer start at completion of SIOOTx

	MSB		LSB		
	7	6	5	4 3 2 1 0	
DMASAA0	+ X	X	X	X X X X X X	} Sets transfer source address
DMASAB0	+ X	X	X	X X X X X X	
DMASAC0	+ X	X	X	X X X X X X	
DMACTA0	+ X	X	X	X X X X X X	} Sets number of transfers
DMACTB0	+ X	X	X	X X X X X X	
DMAMAA0	+ X	X	X	X X X X X X	
DMAMAB0	+ X	X	X	X X X X X X	} Sets data to be mached
DMAMODEA0	+ X	1	0	0 1 1 0 0	
DMAMODEB0	+ 0	0	0	1 0 0 1 0	Pulse detect, transfer source : 8 bits → transfer destination : 8 bits, memory → I/O, single address transfer, byte mode
DMACR0	+ X	X	X	X X X 0 X	Requests at completion of SIOOTx Does not set software request

(c) Single address transfer at completion of A/D conversion and continue mode start

	MSB	7	6	5	4	3	2	1	0	LSB	
DMADAA0	←	X	X	X	X	X	X	X	X	X	} Sets transfer source address
DMADAB0	←	X	X	X	X	X	X	X	X		
DMADAC0	←	X	X	X	X	X	X	X	X		
DMACTA0	←	0	0	0	0	0	0	1	0	} Sets number of transfers to 2	
DMACTB0	←	0	0	0	0	0	0	0	0		
DMAMAA0	←	X	X	X	X	X	X	X	X	} Sets data to be matched	
DMAMAB0	←	X	X	X	X	X	X	X	X		
DMAMODEA0	←	X	0	0	0	0	1	0	1	Level detect, transfer source : 8 bits → transfer destination : 8 bits, I/O → memory, single address transfer, continue mode	
DMAMODEB0	←	1	0	0	1	0	0	0	1	Requests at completion of A/D conversion	
DMACRO	←	X	X	X	X	X	0	0	X	Does not set software request	

(d) Dual address transfer at completion of A/D conversion and continue mode start (update method : I/O mode)

	MSB	7	6	5	4	3	2	1	0	LSB	
DMASAA0	←	0	1	1	0	0	0	0	0	} Sets A/D register address	
DMASAB0	←	0	0	0	0	0	0	0	0		
DMASAC0	←	0	0	0	0	0	0	0	0		
DMADAA0	←	X	X	X	X	X	X	X	X	} Sets transfer destination address	
DMADAB0	←	X	X	X	X	X	X	X	X		
DMADAC0	←	X	X	X	X	X	X	X	X		
DMACTA0	←	0	0	0	0	0	0	1	0	} Sets number of transfers to 2. (Even number must be set because ADREG04L and ADREG04H must be read in two bytes.)	
DMACTB0	←	0	0	0	0	0	0	0	0		
DMAMAA0	←	X	X	X	X	X	X	X	X	} Sets data to be matched	
DMAMAB0	←	X	X	X	X	X	X	X	X		
DMAMODEA0	←	X	0	0	0	X	0	0	1	Level detect, transfer source : 8 bits → transfer destination : 8 bits, dual address transfer, continue mode	
DMAMODEB0	←	1	0	0	1	X	X	1	1	Requests at completion of A/D conversion. Transfer source : I/O mode	
DMACRO	←	X	X	X	X	X	0	0	X	Does not set software request	

(e) DMA start at 16-bit timer 4 time out, and overwrite of timer register

Dual address transfer, transfer source : 16 bits → transfer destination : 8 bits

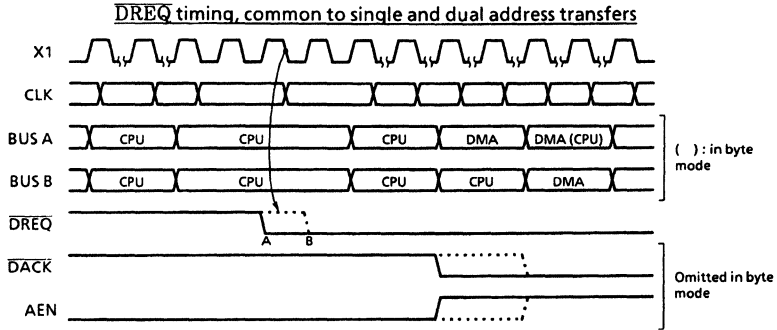
Note : In this case, CS at the transfer source must be set to 16 bits.

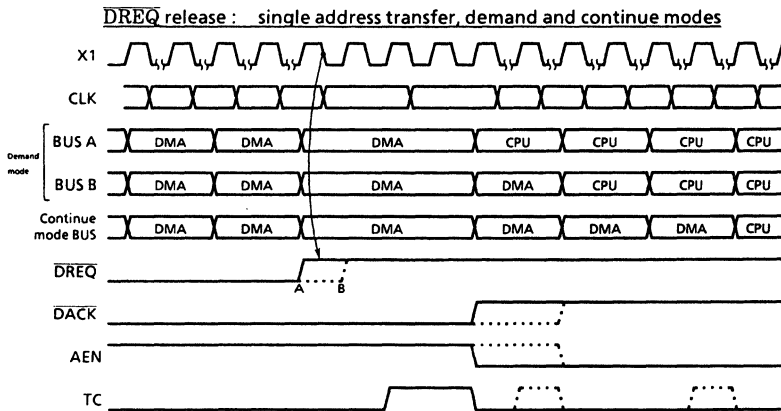
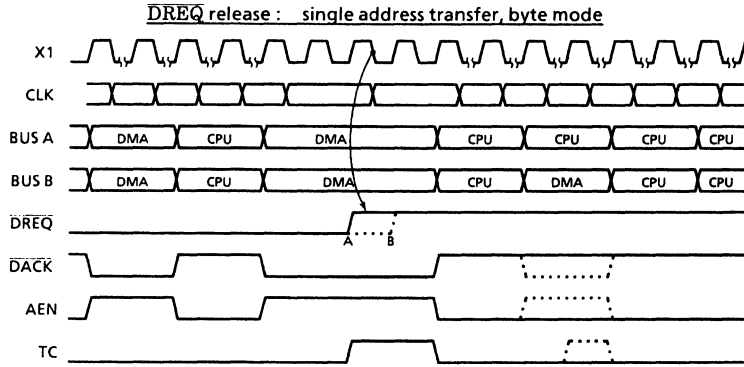
	MSB	7	6	5	4	3	2	1	0	LSB	
DMASAA0	←	X	X	X	X	X	X	X	X	X	} Sets transfer source address
DMASAB0	←	X	X	X	X	X	X	X	X		
DMASAC0	←	X	X	X	X	X	X	X	X		
DMADAA0	←	0	0	1	1	0	0	0	0	0	} Sets timer register address
DMADAB0	←	0	0	0	0	0	0	0	0		
DMADAC0	←	0	0	0	0	0	0	0	0		
DMACTA0	←	X	X	X	X	X	X	X	X	X	} Sets number of transfers
DMACTB0	←	X	X	X	X	X	X	X	X		
DMAMAA0	←	X	X	X	X	X	X	X	X		
DMAMAB0	←	X	X	X	X	X	X	X	X	X	} Sets data to be matched
DMAMODEA0	←	X	1	1	1	X	0	0	0		
DMAMODEB0	←	1	1	1	1	1	1	0	1		} Starts at 16-bit timer 4 time out. Transfer source : I/O mode Transfer destination : increment mode
DMACRO	←	X	X	X	X	X	0	X	X		

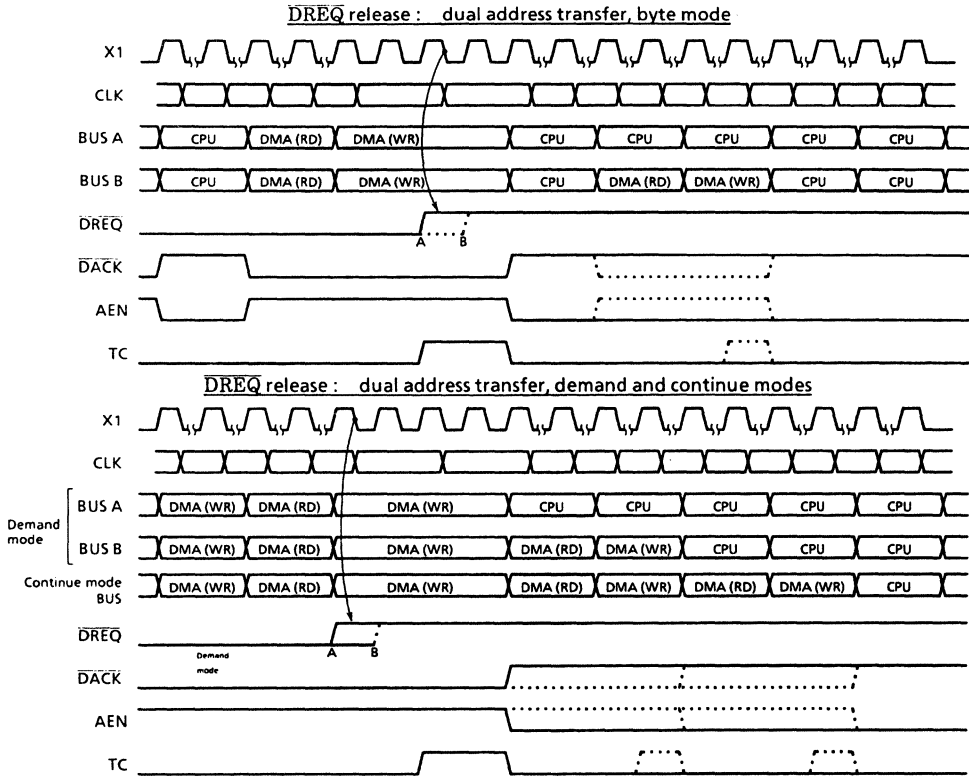
DMA Cycle Timing

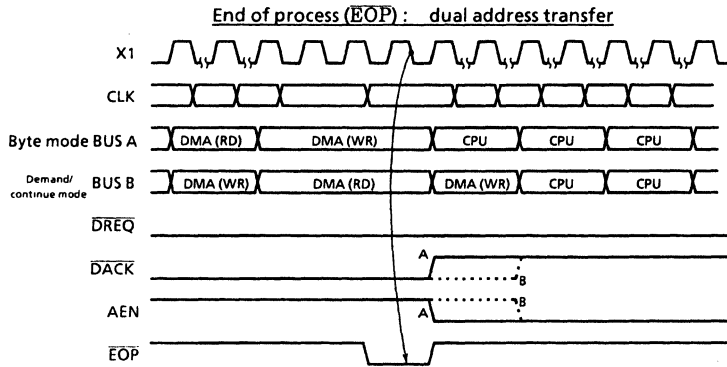
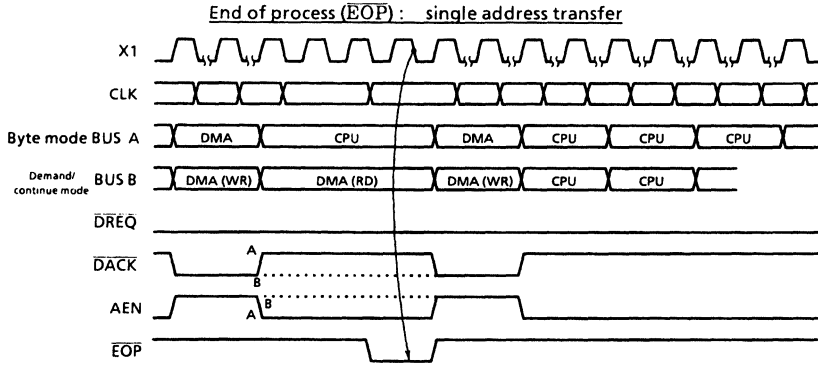
When bus cycle return to CPU cycle from DMA cycle, 2

state dummy cycle may insert to bus cycle.

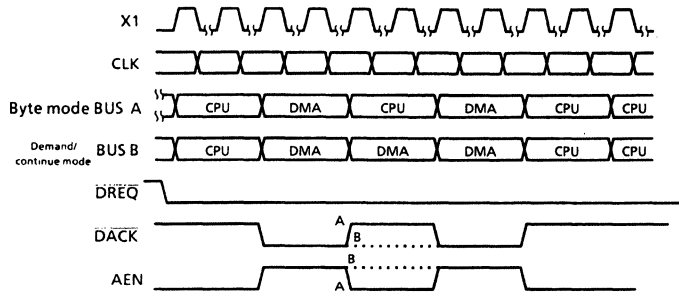




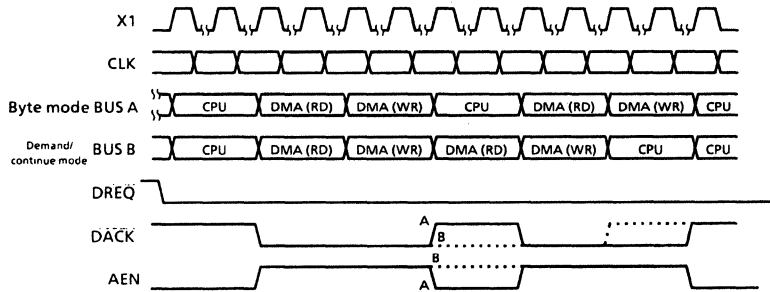




MATCH (second byte is used for match) : single address transfer

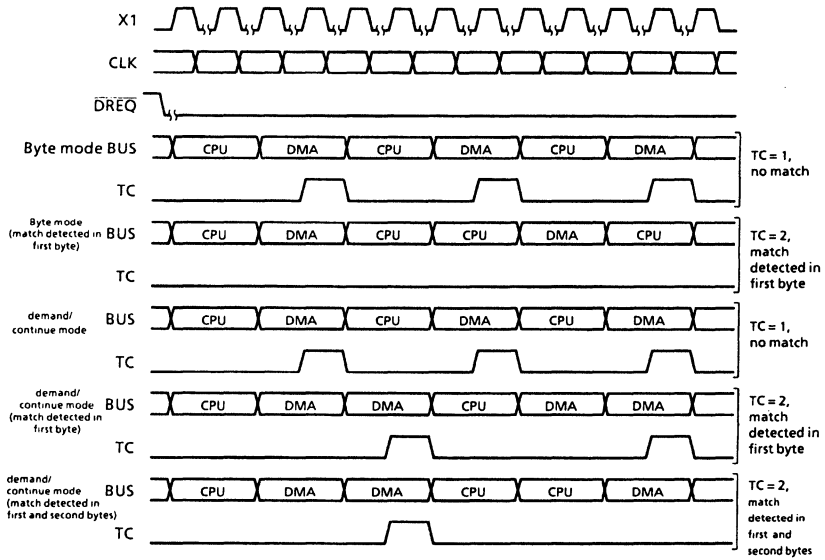


MATCH (second byte is used for match) : dual address transfer

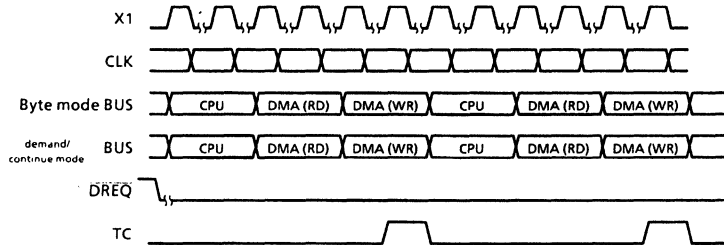


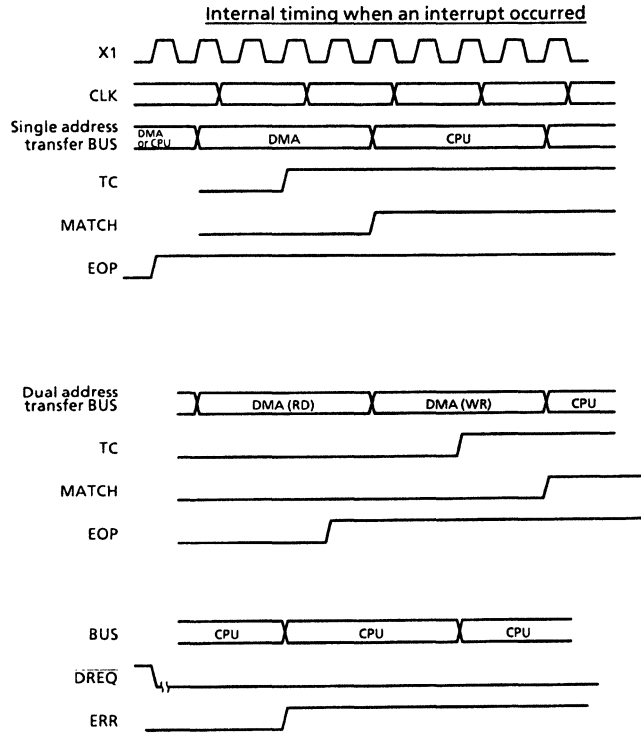


Auto initialization (TC = 1 or 2, match or no match): single address transfer

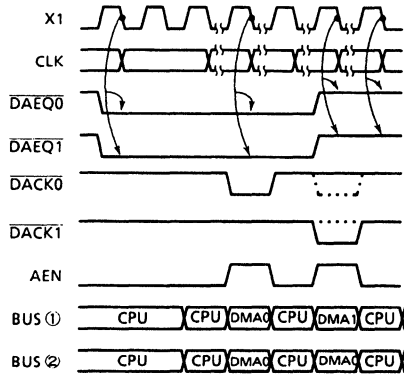


Auto initialization (TC = 1): dual address transfer

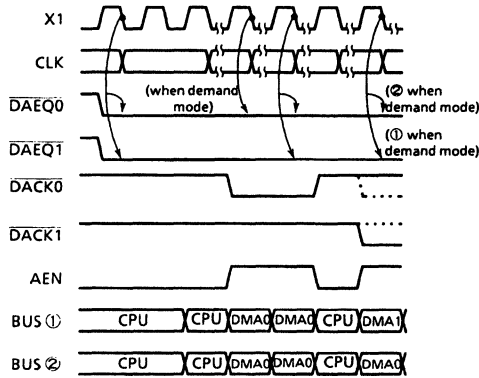




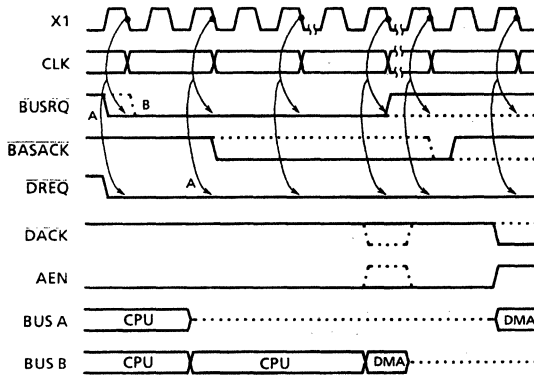
Priority order when transferring in byte mode (① Rotate mode, ② Fixed mode)



Priority order when transferring in demand and continue mode (Auto initialization ① Rotate, ② Fixed mode, when TC = 1)

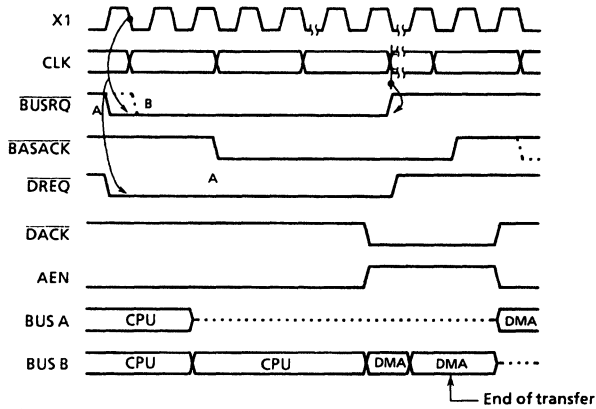


Priority order between  $\overline{\text{BUSRQ}}$  and  $\overline{\text{DREQ}}$  when transferring in byte mode

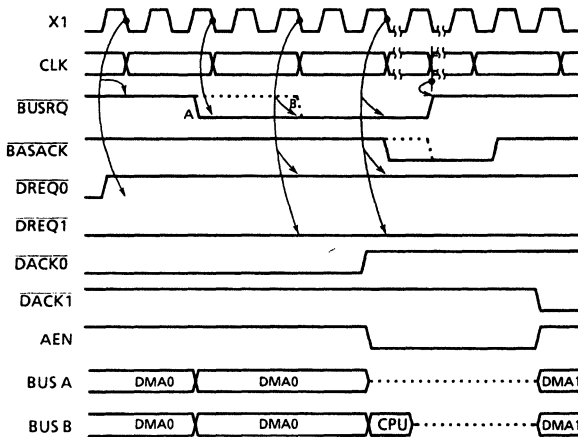


If  $\overline{\text{BUSRQ}}$  and  $\overline{\text{DREQ}}$  are generated at sampling,  $\overline{\text{BUSRQ}}$  has the priority over  $\overline{\text{DREQ}}$

Priority order between  $\overline{\text{BUSRQ}}$  and  $\overline{\text{DREQ}}$  when transferring in demand or continue mode



Priority order between  $\overline{\text{BUSRQ}}$  and other channel's  $\overline{\text{DREQ}}$  when transferring DMA



Because other channel's  $\overline{\text{DREQ}}$  is sampled at the DMA last transfer cycle,  $\overline{\text{BUSRQ}}$  has the priority even if  $\overline{\text{BUSRQ}}$  is generated after other channel's  $\overline{\text{DREQ}}$  is generated.

## 4. Electrical Characteristics

### 4.1 Absolute Maximum (TMP96C081F)

Symbol	Parameter	Rating	Unit
$V_{CC}$	Power Supply Voltage	-0.5 ~ 6.5	V
$V_{IN}$	Input Voltage	-0.5 ~ $V_{CC} + 0.5$	V
$\Sigma I_{OL}$	Output Current (total)	100	mA
$\Sigma I_{OH}$	Output Current (total)	-100	mA
PD	Power Dissipation ( $T_a = 70^\circ\text{C}$ )	600	mW
T SOLDER	Soldering Temperature (10s)	260	$^\circ\text{C}$
T STG	Storage Temperature	-65 ~ 150	$^\circ\text{C}$
T OPR	Operating Temperature	-20 ~ 70	$^\circ\text{C}$

4.2 DC Characteristics (TMP96C081F)

$V_{cc} = 5V \pm 10\%$ ,  $T_a = -20 \sim 70^\circ C$  (Typical values are for  $T_a = 25^\circ C$  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, PA, PB	-0.3	$0.3V_{cc}$	V	
V IL2	RESET, NMI, INTO (P87)	-0.3	$0.25V_{cc}$	V	
V IL3	AM8/T6	-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, PA, PB	$0.7V_{cc}$	$V_{cc} + 0.3$	V	
V IH2	RESET, NMI, INTO (P87)	$0.75V_{cc}$	$V_{cc} + 0.3$	V	
V IH3	AM8/T6	$V_{cc} - 0.3$	$V_{cc} + 0.3$	V	
V IH4	X1	$0.8V_{cc}$	$V_{cc} + 0.3$	V	
V OL	Output Low Voltage		0.45	V	IOL = 1.6mA
V OH	Output High Voltage	2.4		V	IOH = -400µA
V OH1		$0.75V_{cc}$		V	IOH = -100µA
V OH2		$0.9V_{cc}$		V	IOH = -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	$0.0 \leq V_{in} \leq V_{cc}$
I LO	Output Leakage Current	0.05 (Typ)	±10	µA	$0.2 \leq V_{in} \leq V_{cc} - 0.2$
I <sub>cc</sub>	Operating Current (RUN)	26 (Typ)	50	mA	f <sub>osc</sub> = 16MHz
	IDLE	1.7 (Typ)	10	mA	
	STOP (Ta = -20 ~ 70°C)	0.2 (Typ)	50	µA	$0.2 \leq V_{in} \leq V_{cc} - 0.2$
	STOP (Ta = 0 ~ 50°C)		10	µA	$0.2 \leq V_{in} \leq V_{cc} - 0.2$
V STOP	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	V IL2 = $0.2V_{cc}$ , V IH2 = $0.8V_{cc}$
R RST	RESET Pull Up Register	50	150	KΩ	
C IO	Pin Capacitance		10	pF	tosc = 1MHz
V TH	Schmitt Width RESET, NMI, INTO (P87)	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 (TMP96C081F)  $V_{CC} = 5V \pm 10\%$ ,  $T_a = -20 \sim 70^\circ C$  (4MHz ~ 20MHz)

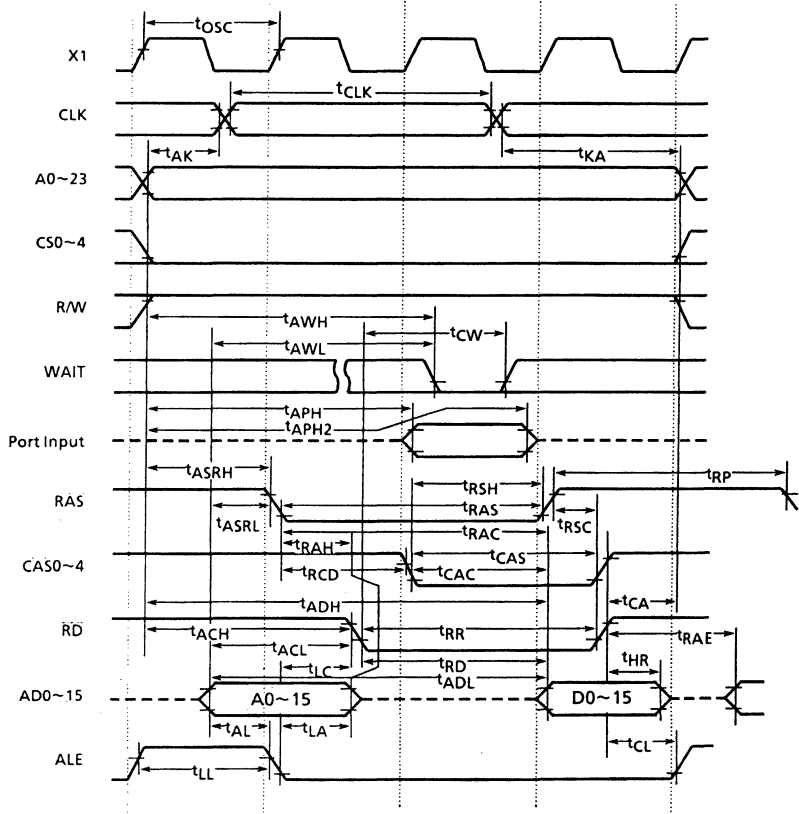
No.	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	$t_{OSC}$	Osc. Period (= x)	62.5	250	62.5		50		ns
2	$t_{CLK}$	CLK width	2x - 40		85		60		ns
3	$t_{AK}$	A0 - 23 Valid $\rightarrow$ CLK Hold	0.5x - 20		11		5		ns
4	$t_{KA}$	CLK Valid $\rightarrow$ A0 - 23 Hold	1.5x - 70		24		5		ns
5	$t_{AL}$	A0-15 Valid $\rightarrow$ ALE fall	0.5x - 15		16		10		ns
6	$t_{LA}$	ALE fall $\rightarrow$ 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 $\rightarrow$ RD/WR fall	0.5x - 30		1		-5		ns
9	$t_{CL}$	RD/WR rise $\rightarrow$ ALE rise	0.5x - 20		11		5		ns
10	$t_{ACL}$	A0 - 15 Valid $\rightarrow$ RD/WR fall	x - 25		38		25		ns
11	$t_{ACH}$	A0 - 23 Valid $\rightarrow$ RD/WR fall	1.5x - 50		44		25		ns
12	$t_{CA}$	RD/WR rise $\rightarrow$ A0 - 23 Hold	0.5x - 20		11		5		ns
13	$t_{ADL}$	A0 - 15 Valid $\rightarrow$ D0 - 15 input		3.0x - 45		143		105	ns
14	$t_{ADH}$	A0 - 23 Valid $\rightarrow$ D0 - 15 input		3.5x - 65		154		110	ns
15	$t_{RD}$	RD fall $\rightarrow$ D0 - 15 input		2.0x - 50		75		50	ns
16	$t_{RR}$	RD Low width	2.0x - 40		85		60		ns
17	$t_{HR}$	RD rise $\rightarrow$ D0 - 15 Hold	0		0		0		ns
18	$t_{RAE}$	RD rise $\rightarrow$ A0 - 15 output	x - 15		48		35		ns
19	$t_{WW}$	WR Low width	2.0x - 40		85		60		ns
20	$t_{DW}$	D0 - 15 Valid $\rightarrow$ WR rise	2.0x - 50		75		50		ns
21	$t_{WD}$	WR rise $\rightarrow$ D0 - 15 Hold	0.5x - 10		21		15		ns
22	$t_{AEH}$	A0 - 23 Valid $\rightarrow$ WAIT input (1WAIT + n mode)		3.5x - 90		129		85	ns
23	$t_{AWL}$	A0 - 15 Valid $\rightarrow$ WAIT input (1WAIT + n mode)		3.0x - 80		108		70	ns
24	$t_{CW}$	RD/WR fall $\rightarrow$ WAIT Hold (1WAIT + n mode)	2.0x + 0		125		100		ns
25	$t_{APH}$	A0 - 23 Valid $\rightarrow$ PORT input		2.5x - 120		80		36	ns
26	$t_{APH2}$	A0 - 23 Valid $\rightarrow$ PORT Hold	2.5x + 50		206		175		ns
27	$t_{CP}$	WR rise $\rightarrow$ PORT Valid		200		200		200	ns
28	$t_{ASRH}$	A0 - 23 Valid $\rightarrow$ RAS fall	1.0x - 40		23		10		ns
29	$t_{ASRL}$	A0 - 15 Valid $\rightarrow$ RAS fall	0.5x - 15		16		10		ns
30	$t_{RAC}$	RAS fall $\rightarrow$ D0 - 15 input		2.5x - 70		130		86	ns
31	$t_{RAH}$	RAS fall $\rightarrow$ A0 - 15 Hold	0.5x - 15		16		10		ns
32	$t_{RAS}$	RAS Low width	2.0x - 40		85		60		ns
33	$t_{RP}$	RAS High width	2.0x - 40		85		60		ns
34	$t_{RSH}$	CAS fall $\rightarrow$ RAS rise	1.0x - 35		28		15		ns
35	$t_{RSC}$	RAS rise $\rightarrow$ CAS rise	0.5x - 25		6		0		ns
36	$t_{RCD}$	RAS fall $\rightarrow$ CAS fall	1.0x - 40		23		10		ns
37	$t_{CAC}$	CAS fall $\rightarrow$ D0 - 15 input		1.5x - 65		29		10	ns
38	$t_{CAS}$	CAS Low width	1.5x - 30		64		45		ns

## AC Measuring Conditions

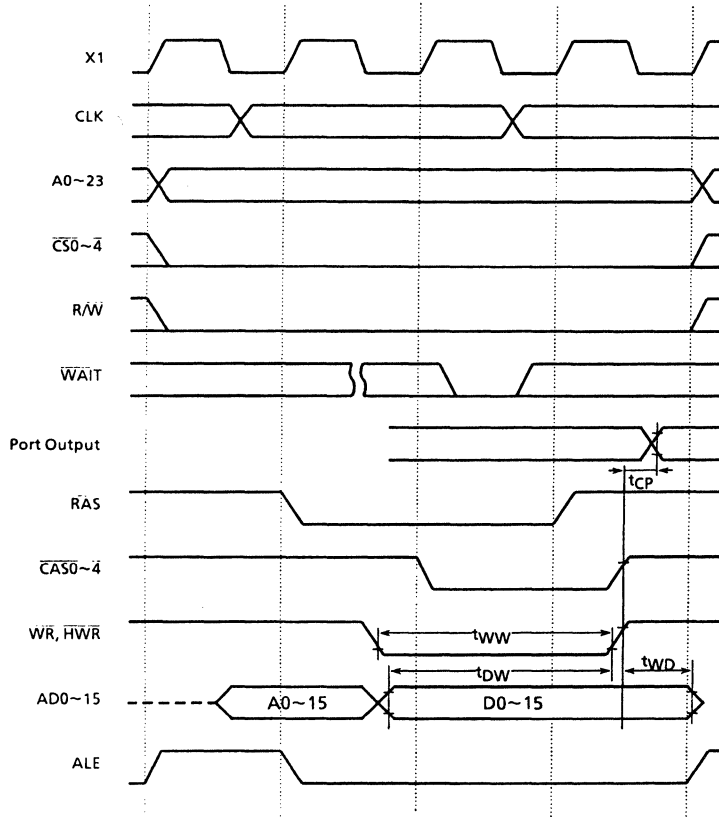
- Output Level: High 2.2V / Low 0.8V, CL50pF  
(However CL = 100pF for AD0 ~ AD15, AD0 ~ AD23, ALE, RD, WR, HWR, RW, CLK, RAS, CAS0 ~ CAS2)
- Input Level: High 2.4V / Low 0.45V (AD0 ~ AD15)  
High 0.8Vcc / Low 0.2Vcc (Except for AD0 ~ AD15)



(1) Read Cycle



(2) Write Cycle



4.3.2 Serial Channel Timing - I/O Interface Mode

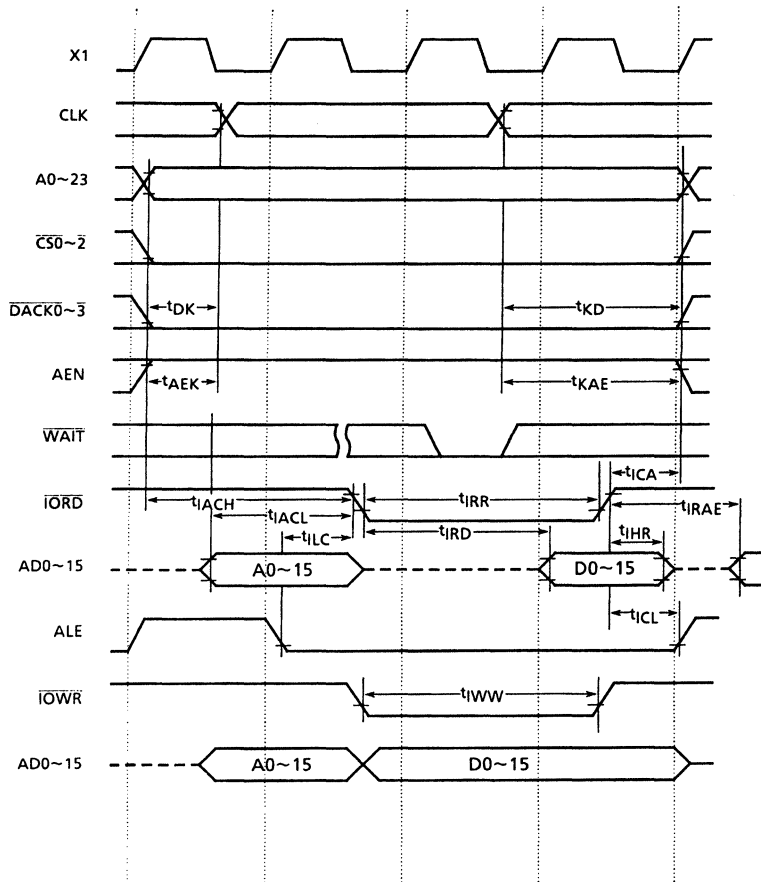
V<sub>cc</sub> = 5V±10% TA = -20 ~ 70°C (4MHz ~ 20MHz)

Number	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t <sub>DK</sub>	DACK valid→CLK hold	0.5x - 20		11		5		ns
2	t <sub>KD</sub>	CLK valid→DACK hold	1.5x - 70		24		5		ns
3	t <sub>AEK</sub>	AEN valid→CLK hold	0.5x - 20		11		5		ns
4	t <sub>KAE</sub>	CLK valid→AEN hold	1.5x - 70		24		5		ns
5	t <sub>ILC</sub>	ALE fall→ $\overline{\text{IORD}}/\overline{\text{IOWR}}$ fall	0.5x - 30		1		-5		ns
6	t <sub>DK</sub>	$\overline{\text{IORD}}/\overline{\text{IOWR}}$ rise→ALE rise	0.5x - 20		11		5		ns
7	t <sub>KD</sub>	A0 - 15 valid→ $\overline{\text{IORD}}/\overline{\text{IOWR}}$ fall	x - 25		38		25		ns
8	t <sub>AEK</sub>	A0 - 23 valid→ $\overline{\text{IORD}}/\overline{\text{IOWR}}$ fall	1.5x - 20		44		25		ns
9	t <sub>KAE</sub>	$\overline{\text{IORD}}/\overline{\text{IOWR}}$ rise→A0 - 23 hold	0.5x - 20		11		5		ns
10	t <sub>ILC</sub>	$\overline{\text{IORD}}$ fall→D0 - 15 input		2.0x - 50		75		50	ns
11	t <sub>IACL</sub>	$\overline{\text{IOWR}}$ low→pulse width	2.0x - 40		85		60		ns
12	t <sub>IACH</sub>	$\overline{\text{IORD}}$ rise→D0 - 15 hold	0		0		0		ns
13	t <sub>IRAE</sub>	$\overline{\text{IORD}}$ fall→A0 - 15 output	x - 15		48		35		ns
14	t <sub>IWW</sub>	$\overline{\text{IOWR}}$ low→pulse width	2.0x - 40		85		60		ns

AC Measuring Conditions

- Output Level: High 2.2V /Low 0.8V, CL50pF  
(However CL = 100pF for ADO ~ AD15, ADO ~ AD23, ALE,  $\overline{\text{IORD}}$ ,  $\overline{\text{IOWR}}$ , CLK, AEN,  $\overline{\text{DACK0}}$  ~  $\overline{\text{DACK3}}$ )
- Input Level: High 2.4V /Low 0.45V (ADO ~ AD15)  
High 0.8V<sub>cc</sub> /Low 0.2V<sub>cc</sub> (Except for ADO ~ AD15)

(1) DMA Cycle



4.4 A/D Conversion Characteristics (TMP96C081F)

$V_{CC} = 5V \pm 10\%$  TA = -20 ~ 70°C

Symbol	Parameter	Min	Typ	Max	Unit
$V_{REF}$	Analog reference voltage	$V_{CC} - 1.5$	$V_{CC}$	$V_{CC}$	V
$A_{GND}$	Analog reference voltage	$V_{SS}$	$V_{SS}$	$V_{SS}$	
$V_{AIN}$	Analog input voltage range	$V_{SS}$		$V_{CC}$	
$I_{REF}$	Analog current for analog reference voltage		0.5	1.5	mA
Error (Quantize error of $\pm 0.5$ LSB not included)	Total error (TA = 25°C, $V_{CC} = V_{REF} = 5.0V$ )			$\pm 4.0$	LSB
	Total error			$\pm 6.0$	

4.5 Serial Channel Timing - I/O Interface Mode

$V_{CC} = 5V \pm 10\%$  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		$\mu s$
$t_{OSS}$	Output Data $\rightarrow$ rising edge of SCLK	$t_{SCY}/2 - 5x - 50$		137		100		ns
$t_{OHS}$	SCLK rising edge $\rightarrow$ output data hold	5x - 100		212		150		ns
$t_{HSR}$	SCLK rising edge $\rightarrow$ input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge $\rightarrow$ 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	$\mu s$
$t_{OSS}$	Output Data $\rightarrow$ rising edge of SCLK	$t_{SCY} - 2x - 150$		725		550		ns
$t_{OHS}$	SCLK rising edge $\rightarrow$ output data hold	2x - 80		45		20		ns
$t_{HSR}$	SCLK rising edge $\rightarrow$ input data hold	0		0		0		ns
$t_{SRD}$	SCLK rising edge $\rightarrow$ effective data input			$t_{SCY} - 2x - 150$		725	550	ns

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

$V_{CC} = 5V \pm 10\%$  TA = -20 ~ 70°C

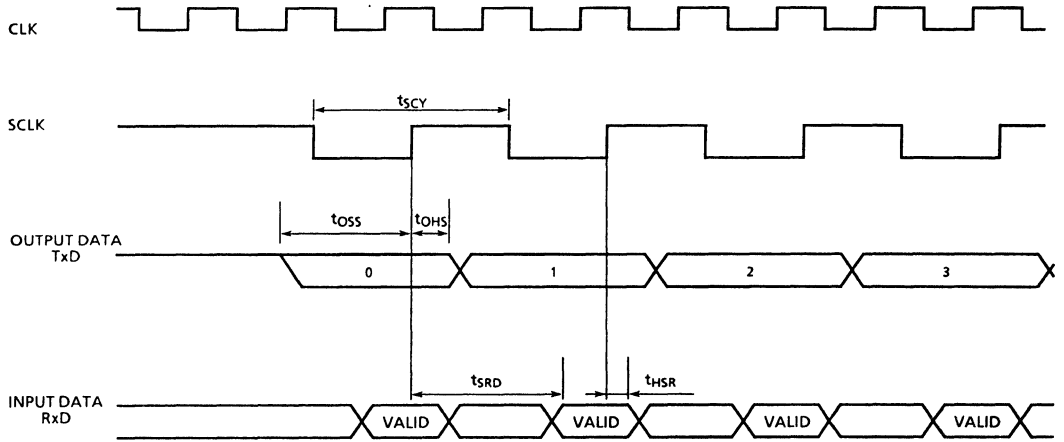
Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{VCK}$	Clock cycle	8x + 100		600		500		ns
$t_{VCKL}$	Low level clock pulse width	4x + 40		290		240		ns
$t_{VCKH}$	High level clock pulse width	4x + 40		290		240		ns

## 4.7 Interrupt Operation

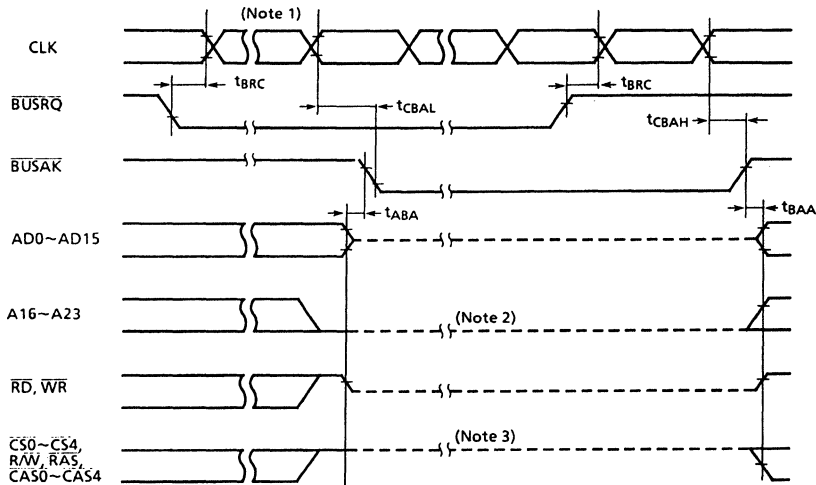
$$V_{CC} = 5V \pm 10\% \quad T_a = -20 \sim 70^{\circ}\text{C}$$

Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{INTAL}$	$\overline{NMI}$ , INTO Low level pulse width	4x		250		200		ns
$t_{INTAH}$	$\overline{NMI}$ , INTO High level pulse width	4x		250		200		ns
$t_{INTBL}$	INT4 ~ INT7 Low level pulse width	8x + 100		600		500		ns
$t_{INTBH}$	INT4 ~ INT7 High level pulse width	8x + 100		600		500		ns

4.8 Timing Chart for I/O Interface Mode



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



Symbol	Parameter	Variable		16MHz		20MHz		Unit
		Min	Max	Min	Max	Min	Max	
$t_{BRC}$	$\overline{\text{BUSRQ}}$ setup time for CLK	120		120		120		ns
$t_{CBAL}$	CLK $\rightarrow$ $\overline{\text{BUSAK}}$ falling edge		$1.5x + 120$		245		220	ns
$t_{CBAH}$	CLK $\rightarrow$ $\overline{\text{BUSAK}}$ rising edge		$0.5x + 40$		71		65	ns
$t_{ABA}$	Floating time to $\overline{\text{BUSAK}}$ fall	0	80	0	80	0	80	ns
$t_{BAA}$	Floating time to $\overline{\text{BUSAK}}$ rise	0	80	0	80	0	80	ns

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

Note 2: The built-in programmable pull-down register is always provided.

Note 3: The built-in programmable pull-up register is always provided.

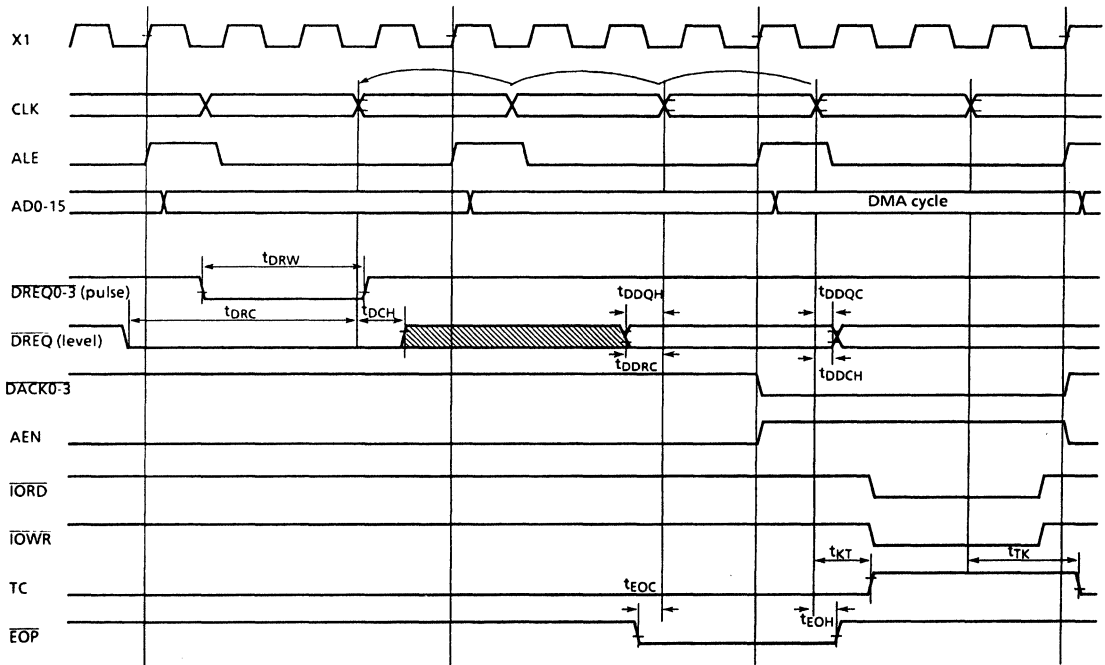
Although CS2/CAS2 pin does not have programmable pull-up register, the built-in pull-up register is always supported with a bus release.



## 4.10 DMAC

Number	Symbol	Parameter	Variable		16MHz		20MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	$t_{DRW}$	$\overline{DREQ}$ low $\rightarrow$ Pulse width (input)	2x		250		100		ns
2	$t_{DRC}$	CLK valid $\rightarrow$ $\overline{DREQ}$ fall (input)					80		ns
3	$t_{DCH}$	CLK valid $\rightarrow$ $\overline{DREQ}$ rise (input)			0		0		ns
4	$t_{DDRC}$	$\overline{DREQ}$ fall (input) $\rightarrow$ CLK valid (Receiving $\overline{DREQ}$ in demand mode)			0		0		ns
5	$t_{DDCH}$	CLK valid $\rightarrow$ $\overline{DREQ}$ rise (input) (Receiving $\overline{DREQ}$ in demand mode)			0		0		ns
6	$t_{DDQH}$	$\overline{DREQ}$ rise (input) $\rightarrow$ CLK valid (Canceling $\overline{DREQ}$ in demand mode)			0		0		ns
7	$t_{DDQC}$	CLK valid $\rightarrow$ $\overline{DREQ}$ high hold (input) (Canceling $\overline{DREQ}$ in demand mode)			0		0		ns
8	$t_{KT}$	CLK valid $\rightarrow$ TC rise (output)	x - 25		38		25		ns
9	$t_{TK}$	CLK valid $\rightarrow$ TC fall (output)	1.5x - 70		24		5		ns
10	$t_{EOC}$	CLK valid $\rightarrow$ $\overline{EOP}$ fall (input)			40		40		ns
11	$t_{EOH}$	CLK valid $\rightarrow$ $\overline{EOP}$ rise (input)			0		0		ns

4.10 DMA Timing Chart



**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

Configuration of the table

Symbol	Name	Address	7	6	5	4	3	2	1	0	
											→bit Symbol
											→Read / Write
											→Initial value after reset
											→Remarks

Table 5 I/O Register Address Map (1/2)

Address	Name	Address	Name	Address	Name	Address	Name
000000H		20H	TRUN	40H	TREG6L	60H	ADREG0L
1H		21H		41H	TREG6H	61H	ADREG0H
2H		22H	TREG0	42H	TREG7L	62H	ADREG1L
3H		23H	TREG1	43H	TREG7H	63H	ADREG1H
4H		24H	TMOD	44H	CAP3L	64H	ADREG2L
5H		25H	TFFCR	45H	CAP3H	65H	ADREG2H
6H	P2	26H	TREG2	46H	CAP4L	66H	ADREG3L
7H	P3	27H	TREG3	47H	CAP4H	67H	ADREG3H
8H	P2CR	28H	P0MOD	48H	T5MOD	68H	B0CSL
9H	P2FC	29H	P1MOD	49H	T5FFCR	69H	B1CSH
AH	P3CR	2AH	PFFCR	4AH	B4CSL	6AH	B1CSL
BH	P3FC	2BH		4BH	B4CSH	6BH	B1CSH
CH	P4	2CH		4CH	PGOREG	6CH	B2CSL
DH	P5	2DH		4DH	PG1REG	6DH	B2CSH
EH	P4CR	2EH		4EH	PG01CR	6EH	B3CSL
FH		2FH		4FH		6FH	B3CSH
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H		31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BR0CR	73H	INTET10
14H	P6CR	34H	CAP1L	54H	SC1BUF	74H	INTEPW10
15H	P7CR	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P6FC	36H	CAP2L	56H	SC1MOD	76H	INTET76
17H	P7FC	37H	CAP2H	57H	BR1CR	77H	INTES0
18H	P8	38H	T4MOD	58H	ODE	78H	INTES1
19H	P9	39H	TFF4CR	59H		79H	INTDMA0
1AH	P8CR	3AH	T45CR	5AH		7AH	INTDMA1
1BH	P9CR	3BH		5BH		7BH	IIMC
1CH	P8FC	3CH	PACR	5CH	WDMOD	7CH	DMA0V
1DH	P9FC	3DH	PBCR	5DH	WDCR	7DH	DMA1V
1EH	PA	3EH	PAFC	5EH	ADMOD	7EH	DMA2V
1FH	PA	3FH	PBFC	5FH	ADBS	7FH	DMA3V

Table 5 I/O Register Address Map (2/2)

Address	Name	Address	Name	Address	Name	Address	Name
000080H	DMASAA0	A0H	DMASAA2	C0H		E0H	
81H	DMASAB0	A1H	DMASAB2	C1H		E1H	
82H	DMASAC0	A2H	DMASAC2	C2H		E2H	
83H		A3H		C3H		E3H	
84H	DMADAA0	A4H	DMADAA2	C4H		E4H	
85H	DMADAB0	A5H	DMADAB2	C5H		E5H	
86H	DMADAC0	A6H	DMADAC2	C6H		E6H	
87H		A7H		C7H		E7H	
88H	DMACTA0	A8H	DMACTA2	C8H		E8H	
89H	DMACTB0	A9H	DMACTB2	C9H		E9H	
8AH	DMAMAA0	AAH	DMAMAA2	CAH		EAH	
8BH	DMAMAB0	ABH	DMAMAB2	CBH		EBH	
8CH	DMAMODEA0	ACH	DMAMODEA2	CCH		ECH	
8DH	DMAMODEB0	ADH	DMAMODEB2	CDH		EDH	
8EH	DMACR0	AEH	DMACR2	CEH		EEH	
8FH	DMAST0	AFH	DMAST2	CFH		EFH	
90H	DMASAA1	B0H	DMASAA3	D0H		FOH	
91H	DMASAB1	B1H	DMASAB3	D1H		F1H	
92H	DMASAC1	B2H	DMASAC3	D2H		F2H	
93H		B3H		D3H		F3H	
94H	DMADAA1	B4H	DMADAA3	D4H		F4H	
95H	DMADAB1	B5H	DMADAB3	D5H		F5H	
96H	DMADAC1	B6H	DMADAC3	D6H		F6H	
97H		B7H		D7H		F7H	
98H	DMACTA1	B8H	DMACTA3	D8H		F8H	
99H	DMACTB1	B9H	DMACTB3	D9H		F9H	
9AH	DMAMAA1	BAH	DMAMAA3	DAH		FAH	
9BH	DMAMAB1	BBH	DMAMAB3	DBH		FBH	
9CH	DMAMODEA1	BCH	DMAMODEA3	DCH		FCH	
9DH	DMAMODEB1	BDH	DMAMODEB3	DDH		FDH	
9EH	DMACR1	BEH	DMACR3	DEH		FEH	
9FH	DMAST1	BFH	DMAST3	DFH		FFH	

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

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	P37	P36	P35	P34	P33	P32	P31	P30
			R/W							
			Input mode							
			1	1	1	1	1	1	1	1
P4	PORT4	0CH				P44	P43	P42	P41	P40
			R/W							
			Input mode							
							0	1	1	
P5	PORT5	0DH			P55	P54	P53	P52	P51	P50
			R							
			Input mode							
P6	PORT6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			Input mode							
			1	1	1	1	1	1	1	1
P7	PORT7	13H					P73	P72	P71	P70
			R/W							
			Input mode							
							1	1	1	1
P8	PORT8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			R/W							
			Input mode							
			1	1	1	1	1	1	1	1
P9	PORT9	19H			P95	P94	P93	P92	P91	P90
			R/W							
			Input mode							
					1	1	1	1	1	1

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

Symbol	Name	Address	7	6	5	4	3	2	1	0
PA	PORT8	18H	P107	P106	P105	P104	P103	P102	P101	P100
			R/W							
			Input mode							
			1	1	1	1	1	1	1	1
PB	PORT9	19H	P117	P116	P115	P114	P113	P112	P111	P110
			R/W							
			Input mode							
					1	1	1	1	1	1

Note: Clearing the output latch register RDFC to "0" outputs the  $\overline{RD}$  strobe from  $\overline{RD}$  pin for PSRAM, even when the internal address is accessed. If the output latch register RDFC remains "1", the  $\overline{RD}$  strobe is output only when the external address is accessed.

- 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 (1/3)

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 : A7 - 0, 11 : A23 - 16							
P3CR	PORT3 Control	0AH (Prohibit RMW)	P37C	P36C	P35C	P34C	P33C	P32C		
			W							
			0	0	0	0	0	0	0	
			0 : IN 1 : OUT							
P3FC	PORT3 Function	0BH (Prohibit RMW)	P37F	P36F	P35F	P34F		P32F	P31F	P30F
			W							
			0	0	0	0		0	0	0
			0 : PORT 1 : $\overline{RAS}$	0 : PORT 1 : R/W	0 : PORT 1 : $\overline{BUSAK}$	0 : PORT 1 : $\overline{BUSRQ}$		0 : PORT 1 : $\overline{HWR}$	0 : PORT 1 : $\overline{WR}$	0 : PORT 1 : $\overline{RD}$
P4CR	PORT4 Control	0EH (Prohibit RMW)				P44C	P43C	P42C	P41C	P40C
			W							
						0	0	0	0	0
			0 : IN 1 : OUT							
P4FC	PORT4 Function	10H (Prohibit RMW)				P44F	P43F	P42F	P41F	P40F
			W							
						0	0	0	0	0
			0 : PORT 1 : $\overline{CS/CAS}$							

Note: With the TMP96C141/TMP96C141A/TMP96C041A, which requires an external ROM. PORT0 functions as AD0 to AD7; PORT1, AD8 to AD15; P30, the  $\overline{RD}$  signal; P31, the  $\overline{WR}$  signal, regardless of the values set in P0CR, P1CR, P1FC, P30F and P31F.

(2) I/O Port Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0		
P6CR	PORT6 Control	14H (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C		
			W									
			0	0	0	0	0	0	0	0	0	
			0 : IN				1 : OUT					
P7CR	PORT7 Control	15H (Prohibit RMW)					P73C	P72C	P71C	P70C		
			W									
							0	0	0	0		
			0 : IN				1 : OUT					
P6FC	PORT6 Function	16H (Prohibit RMW)	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F		
			W									
			0	0	0	0	0	0	0	0	0	
			0 : IN				: OUT				0 : PORT	
P7FC	PORT7 Function	17H (Prohibit RMW)					P73F	P72F	P71F			
			W									
							0	0	0			
							0 : PORT		0 : PORT		0 : PORT	
				1 : T03		1 : T02		1 : T01				
P8CR	PORT8 Control	1AH (Prohibit RMW)	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C		
			W									
			0	0	0	0	0	0	0	0	0	
			0 : IN				1 : OUT					
P9CR	PORT9 Control	1BH (Prohibit RMW)			P95C	P94C	P93C	P92C	P91C	P90C		
			W									
					0	0	0	0	0	0	0	
			0 : IN				1 : OUT					
P8FC	PORT8 Function	1CH (Prohibit RMW)		P86F			P83F	P82F				
				W			W	W				
				0			0	0				
			0 : PORT				0 : PORT		0 : PORT			
1 : T06				1 : T05		1 : T04						
P9FC	PORT9 Function	1DH (Prohibit RMW)			P95F		P93F	P92F		P90F		
					W		W	W		W		
					0		0	0		0		
			0 : PORT				0 : PORT		0 : PORT		0 : PORT	
1 : SCLK1				1 : TxD1		1 : SCLK0		1 : TxD0				
PACR	PORTA Control	3CH (Prohibit RMW)	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C		
			W									
			0	0	0	0	0	0	0	0	0	
			0 : IN				1 : OUT					
PBCR	PORT8 Control	3DH (Prohibit RMW)				P84F	P83F	P82F	P81F	P80F		
			W									
						0	0	0	0	0	0	
			0 : IN				1 : OUT					



(2) I/O Port Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PAFC	PORTA Function	3EH  (Prohibit RMW)	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
			W							
			0	0	0	0		0	0	0
			0 : PORT 1 : DACK3	0 : PORT 1 : DREQ3	0 : PORT 1 : DACK2	0 : PORT 1 : DREQ2	0 : PORT 1 : DACK1	0 : PORT 1 : DREQ1	0 : PORT 1 : DACK0	0 : PORT 1 : DREQ0
PBFC	PORTB Function	3FH  (Prohibit RMW)				PB4F	PB3F	PB2F	PB1F	PB0F
			W							
			0	0	0	0		0	0	0
					0 : PORT 1 : IOWR	0 : PORT 1 : IORD	0 : PORT 1 : TC	0 : PORT 1 : EOP	0 : PORT 1 : AEN	

(3) Timer Control (1/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer Control	20H	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
			R/W		R/W					
			0		0	0	0	0	0	0
Prescaler and Timer Run/Stop CONTROL 0 : Stop and Clear 1 : Run (Count up)										
TREG0	8bit Timer Register 0	22H  (Prohibit RMW)	-							
			W							
			Undefined							
TREG1	8bit Timer Register 1	23H  (Prohibit RMW)	-							
			W							
			Undefined							
TMOD	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 : 8bit Timer 01 : 16bit Timer 10 : 8bit PPG 11 : 8bit PWM	00 : - 01 : $2^6 - 1$ PWM 10 : $2^7 - 1$ 11 : $2^8 - 1$		00 : T00TRG 01 : $\phi T1$ 10 : $\phi T16$ 11 : $\phi T256$		00 : T10 Input 01 : $\phi T1$ 10 : $\phi T4$ 11 : $\phi T16$		
TFFCR	8bit Timer Flip-flop Control	25H				DBEN	TFF1C1	TFF1C0	TFF11E	TFF11S
						R/W	W		R/W	
						0	0	0	0	0
					1 : Double Buffer Enable	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care		1 : TFF1 Invert Enable	0 : Inverted by Timer 0	
TREG2	PWM Timer Register 2	26H	-							
			(R)/W (Can read double buffer values.)							
			Undefined							

(3) Timer Control (2/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0	
TREG3	PWM Timer Register 3	27H	-								
			(R)/W (Can read double buffer values.)								
			Undefined								
P0MOD	PWM0 MODE	28H  (Prohibit RMW)	FF2RD	DB2EN	PWM0INT	PWM0M	T2CLK1	T2CLK0	PWM0S1	PWM0S0	
			R	W							
			-	0	0	0	0	0	0	0	
			TFF2 output value	1: Double Buffer Enable	0: Overflow Interrupt 1: Compare/Match Interrupt	0: PWM Mode 1: Timer Mode	00: $\phi$ P1(tc/4) 01: $\phi$ P4(tc/16) 10: $\phi$ P16(tc/64) 11: Don't care	00: $2^6 - 1$ 01: $2^7 - 1$ 10: $2^8 - 1$ 11: Don't care			
P1MOD	PWM1 MODE	29H  (Prohibit RMW)	FF3RD	DB3EN	PWM1INT	PWM1M	T3CLK1	T3CLK0	PWM1S1	PWM1S0	
			R	W							
			-	0	0	0	0	0	0	0	
			TFF3 output value	1: Double Buffer Enable	0: Overflow Interrupt 1: Compare/Match Interrupt	0: PWM Mode 1: Timer Mode	00: $\phi$ P1(tc/4) 01: $\phi$ P4(tc/16) 10: $\phi$ P16(tc/64) 11: Don't care	00: $2^6 - 1$ 01: $2^7 - 1$ 10: $2^8 - 1$ 11: Don't care			
PFFCR	PWM Flip-flop Control	2AH	FF3C1	FF3C0	FF3TRG1	FF3TRG0	FF2C1	FF2C0	FF2TRG1	FF2TRG0	
			W		R/W		W		R/W		
			0	0	0	0	0	0	0	0	
			00: Don't care 01: Set TFF3 10: Clear TFF3 11: Don't care		00: Prohibit TFF3 Inverted 01: Invert if matched 10: Set if matched; Clear if overflowed 11: Clear if matched; set if overflowed		00: Don't care 01: Set TFF2 10: Clear TFF2 11: Don't care		00: Prohibit TFF2 Inverted 01: Invert if matched 10: Set if matched; Clear if overflowed 11: Clear if matched; set if overflowed		
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								

(3) Timer Control (3/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
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 ↑ T15 ↑ 10: T14 ↑ T14 ↓ 11: TFF1 ↑ TFF1 ↓		1: UC4 Clear Enable	Source Clock 00: T14 01: φT1 10: φT4 11: φT16	
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				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 5	PG0 shift trigger 0: Timer 0, 1 1: Timer 4	1: Double Buffer Enable	
TREG6L	16bit Timer Register 6L (Prohibit RMW)	40H	-							
			W							
			Undefined							
TREG6H	16bit Timer Register 6H (Prohibit RMW)	41H	-							
			W							
			Undefined							
TREG7L	16bit Timer Register 6L (Prohibit RMW)	42H	-							
			W							
			Undefined							
TREG7H	16bit Timer Register 6H (Prohibit RMW)	43H	-							
			W							
			Undefined							

(3) Timer Control (4/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
CAP3L	Capture Register 3L	44H	-							
			R							
			Undefined							
CAP3H	Capture Register 3H	45H	-							
			R							
			Undefined							
CAP4L	Capture Register 4L	46H	-							
			R							
			Undefined							
CAP4H	Capture Register 4H	47H	-							
			R							
			Undefined							
T5MOD	16bit Timer 5 Source CLK and MODE	48H			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
					W	R/W				
					0	0	0	0	0	0
			0 : Soft-Capture 1 : Don't care		Capture Timing 00 : Disable 01 : T16 ↑ T17 ↑ 10 : T16 ↑ T16 ↓ 11 : TFF1 ↑ TFF1 ↓			1 : UC5 Clear Enable		Source Clock 00 : T16 01 : φT1 10 : φT4 11 : φT16
T5FFCR	16bit Timer 5 Flip-flop Control	49H			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
					R/W				W	
					0	0	0	0	0	0
					TFF6 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF6 01 : Set TFF6 10 : Clear TFF6 11 : Don't care	

(4) Pattern Generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
PG0REG	PG0 Register (Prohibit RMW)	4CH	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
			W				R/W			
			0				Undefined			
PG1REG	PG1 Register (Prohibit RMW)	4DH	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
			W				R/W			
			0				Undefined			
PG01CR	PG0, 1 Control	4EH	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
			R/W							
			0							
			0 : 8bit write 1 : 4bit write	0 : Normal Rotation 1 : Reverse Rotation	0 : 4bit Step 1 : 8bit Step	PG1 trigger input enable 1 : Enable	0 : 8bit write 1 : 4bit write	0 : Normal Rotation 1 : Reverse Rotation	0 : 4bit Step 1 : 8bit 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 <sup>16</sup> /fc 01 : 2 <sup>18</sup> /fc 10 : 2 <sup>20</sup> /fc 11 : 2 <sup>22</sup> /fc	Warming up Time 0 : 2 <sup>14</sup> /fc 1 : 2 <sup>16</sup> /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	RBO TBO		
			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	0	0	
			Receiving data bit 8	Parity 0 : Odd 1 : Even	1 : Parity Enable	1 : Error			Fix at "0"	Fix at "0"		
			Overrun	Parity	Framing							
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	0	
			Transmission data bit 8	1 : CTS Enable	1 : Receive Enable	1 : Wake up Enable	00 : Unused 01 : UART 7bit 10 : UART 8bit 11 : UART 9bit			00 : T00 Trigger 01 : Baud rate generator 10 : Internal clock $\phi$ 1 11 : Don't care		
BR0CR	Baud Rate Control	53H	-		BROCK1	BROCK0	BR053	BR052	BR051	BR050		
			R/W	R/W								
			0		0	0	0	0	0	0	0	
			Fix at "0"		00 : $\phi$ 0 (fc/4) 01 : $\phi$ 2 (fc/16) 10 : $\phi$ 8 (fc/64) 11 : $\phi$ 32 (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	RBO TBO		
			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	0	0	
			Receiving data bit 8	Parity 0 : Odd 1 : Even	1 : Parity Enable	1 : Error			0 : SCLK0 1 : SCLK0	1 : Input SCLK1 pin		
			Overrun	Parity	Framing							
SC1-MOD	Serial Channel 1 Mode	56H	TB8	-	RXE	WU	SM1	SM0	SC1	SC0		
			R/W									
			0	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 7bit 10 : UART 8-bit 11 : UART 9bit			00 : T00 Trigger 01 : Baud rate generator 10 : Internal clock $\phi$ 1 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 : $\phi t0$ (tc/4) 01 : $\phi t2$ (tc/16) 10 : $\phi t8$ (tc/64) 11 : $\phi t32$ (tc/256)	Set frequency divisor 0 ~ F ("1" prohibited)					
ODE	Special Open Drain Enable	58H	-						ODE1	ODE0	
									R/W		
									0	0	
									1 : P93 Open-drain	1 : P90 Open-drain	

(7) A/D Converter Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0		
ADMOD	A/D Converter Mode Register	5EH	EOCF	ADBF	REPET	SCAN	ADCS	ADS	ADCH1	ADCH0		
			R		R/W	R/W	R/W		R/W			
			0: END	1: BUSY	1: Repeat Mode Set	1: Scan Mode Set	1: Slow Mode	1: START	Analog Input Channel Select			
ADCH	A/D Control Select Register	5FH	---	---	---	---	---	---	---	ADCH2		
										R/W		
										0: AN0-3	1: AN4-5	
AD REG04L	A/D Result Register CH0, 4 (L)	60H	ADR041	ADR040								
			R		Undefined							
			Lower 2 bits of AD conversion result for AN0 or AN4 are stored.									
AD REG04H	A/D Result Register CH0, 4 (H)	61H	ADR049	ADR048	ADR047	ADR046	ADR045	ADR044	ADR043	ADR042		
			R		Undefined							
			Upper 8 bits of AD conversion result for AN0 or AN4 are stored.									
AD REG15L	A/D Result Register CH1, 5 (L)	62H	ADR151	ADR150								
			R		Undefined							
			Lower 2 bits of AD conversion result for AN1 or AN5 are stored.									
AD REG15H	A/D Result Register CH1, 5 (H)	63H	ADR159	ADR158	ADR157	ADR156	ADR155	ADR154	ADR153	ADR152		
			R		Undefined							
			Upper 8 bits of AD conversion result for AN1 or AN5 are stored.									
AD REG2L	A/D Result Register CH2 (L)	64H	ADR21	ADR20								
			R		Undefined							
			Lower 2 bits of AD conversion result for AN2 are stored.									
AD REG2H	A/D Result Register CH2 (H)	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22		
			R		Undefined							
			Upper 8 bits of AD conversion result for AN2 are stored.									
AD REG3L	A/D Result Register CH3 (L)	66H	ADR031	ADR030								
			R		Undefined							
			Lower 2 bits of AD conversion result for AN3 are stored.									
AD REG3H	A/D Result Register CH3 (H)	67H	ADR039	ADR038	ADR037	ADR036	ADR035	ADR034	ADR033	ADR032		
			R		Undefined							
			Upper 8 bits of AD conversion result for AN3 are stored.									



(8) Interrupt Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE-0AD	INTerrupt Enable 0 & A/D (Prohibit RMW)	70H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE45	INTerrupt Enable 4/5 (Prohibit RMW)	71H	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 (Prohibit RMW)	72H	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE10	INTerrupt Enable Timer 1/0 (Prohibit RMW)	73H	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
INTE-PW10	INTerrupt Enable PWM 1/0 (Prohibit RMW)	74H	INTT3 (Timer 3/PWM1)				INTT2 (Timer 2/PWM0)			
			IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE54	INTerrupt Enable Treg 5/4 (Prohibit RMW)	75H	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
INTE76	INTerrupt Enable Treg 7/6 (Prohibit RMW)	76H	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE50	INTerrupt Enable Serial 0 (Prohibit RMW)	77H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE51	INTerrupt Enable Serial 1 (Prohibit RMW)	78H	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INT-DMA0	INTerrupt Enable DMAC 1/0 (Prohibit RMW)	79H	DMA1				DMA0			
			ID1C	ID1M2	ID1M1	ID1M0	ID0C	ID0M2	ID0M1	ID0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INT-DMA1	INTerrupt Enable DMAC 3/2 (Prohibit RMW)	7AH	DMA3				DMA2			
			ID3C	ID3M2	ID3M1	ID3M0	ID2C	ID2M2	ID2M1	ID2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

(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 : DMA0V5 : DMA0V4							
			W							
			0 : 0 : 0 : 0 : 0							
DMA1V	DMA 1 request Vector	7DH (Prohibit RMW)	μDMA1 start vector							
			DMA1V8 : DMA1V7 : DMA1V6 : DMA1V5 : DMA1V4							
			W							
			0 : 0 : 0 : 0 : 0							
DMA2V	DMA 2 request Vector	7EH (Prohibit RMW)	μDMA2 start vector							
			DMA2V8 : DMA2V7 : DMA2V6 : DMA2V5 : DMA2V4							
			W							
			0 : 0 : 0 : 0 : 0							
DMA3V	DMA 3 request Vector	7FH (Prohibit RMW)	μDMA3 start vector							
			DMA3V8 : DMA3V7 : DMA3V6 : DMA3V5 : DMA3V4							
			W							
			0 : 0 : 0 : 0 : 0							
IIMC	Interrupt Input Mode Control	7BH (Prohibit RMW)	IOIE : IOLE : NMIREE							
			W : W : W							
			0 : 0 : 0							
			1: INTO input enable : 0: INTO edge mode : 1: Operate even at NMI rise edge 1: INTO level mode							

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

Symbol	Name	Address	B7	B6	B5	B4	B3	B2	B1	B0
B0CSL	Block0 CS/WAIT Control Register Low	68H	B0E	B0SYS	B0CAS	B0BUS	—	B0W2	B0W1	B0W1
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System Only	0: /CS0 1: /CAS0	0: 16 bit Bus 1: 8 bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B0CSH	Block0 CS/WAIT Control Register High	69H	B0M3	B0M2	B0M1	B0M0	B0A3	B0A2	B0A1	B0A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparision 0000: 7F0H~7FFFH Other: Compare A23~A20			
B1CSL	Block1 CS/WAIT Control Register Low	6AH	B1E	B1SYS	B1CAS	B1BUS	—	B1W2	B1W1	B1W0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System Only	0: /CS1 1: /CAS1	0: 16 bit Bus 1: 8 bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B1CSH	Block1 CS/WAIT Control Register High	6BH	B1M3	B1M2	B1M1	B1M0	B1A3	B1A2	B1A1	B1A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparision 0000: 0080H~7FFFH Other: Compare A23~A20			
B2CSL	Block2 CS/WAIT Control Register Low	6CH	B2E	B2SYS	B2CAS	B2BUS	—	B2W2	B2W1	B2W0
			W	W	W	W		W	W	W
			1	0	0	0		0	0	0
			1: CS/CAS Enable	1: System Only	0: /CS2 1: /CAS2	0: 16 bit Bus 1: 8 bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B2CSH	Block2 CS/WAIT Control Register High	6DH	B2M3	B2M2	B2M1	B2M0	B2A3	B2A2	B2A1	B2A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparision 0000: 8000H~3FFFFH Other: Compare A23~A20			

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

Symbol	Name	Address	B7	B6	B5	B4	B3	B2	B1	B0
B3CSL	Block3 CS/WAIT Control Register Low	6EH	B3E	B3YS	B3CAS	B3BUS	—	B3W2	B3W1	B3W1
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System Only	0: /CS3 1: /CAS3	0: 16 bit Bus 1: 8 bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B3CSH	Block3 CS/WAIT Control Register High	6FH	B3M3	B3M2	B3M1	B3M0	B3A3	B3A2	B3A1	B3A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparison 0000: Address areas are not specified Other: Compare A23~A20			
B4CSL	Block4 CS/WAIT Control Register Low	4AH	B4E	B4YS	B4CAS	B4BUS	—	B4A2	B4A1	B4A0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			1: CS/CAS Enable	1: System Only	0: /CS4 1: /CAS4	0: 16 bit Bus 1: 8 bit Bus		Wait Control 000: 2 Wait 001: 1 Wait 010: 1 Wait + n 011: 0 Wait 100: 0~2 Wait 101: 0~1 Wait 110: Reserved 111: Reserved		
B4CSH	Block4 CS/WAIT Control Register High	4BH	B4M3	B4M2	B4M1	B4M0	B4A3	B4A2	B4A1	B4A0
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Address (A23~A20) Mask Set 0: Non-Mask 1: Mask				Address (A23~A20) Comparison 0000: Address areas are not specified Other: Compare A23~A20			

### 6. Port Section Equivalent Circuit Diagram

• Reading The Circuit Diagram

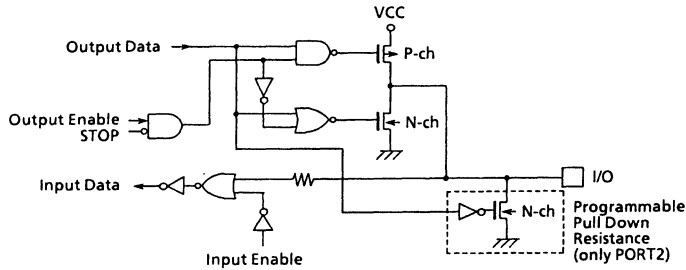
Basically, the gate singles written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The dedicated signal is described below.

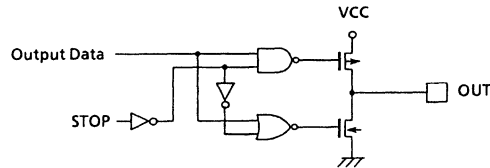
STOP: This signal becomes active "1" when the hold mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit [DRIVE] is set to "1", however, STP remains at "0".

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

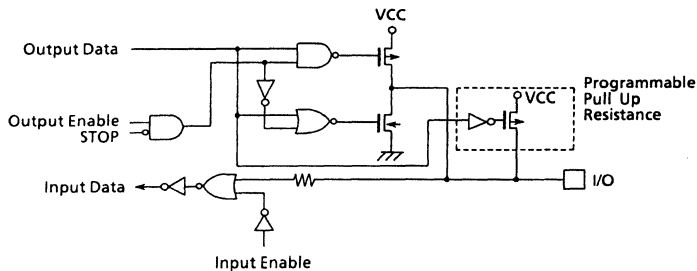
- AD0 ~ AD7, AD8 ~ 15, A8 ~ 15, P2 (A16 - 23, A0 ~7)



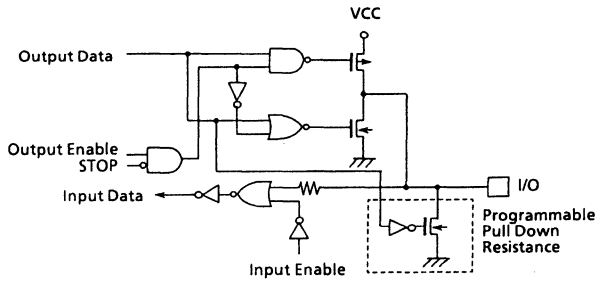
- $\overline{RD}$ ,  $\overline{WR}$



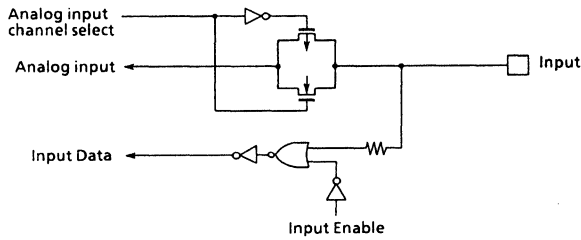
- P32 ~ 37, P40 ~ 41, P43 ~ 44



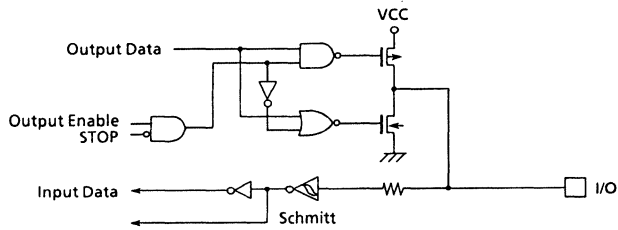
- P42 ( $\overline{CS2}$ ,  $\overline{CAS2}$ )



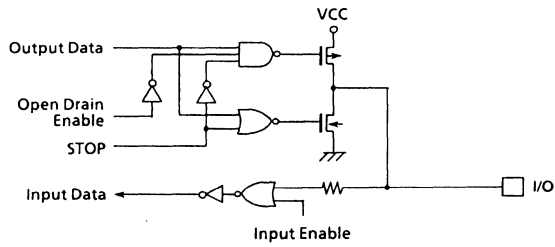
- P5 (AN0 ~ 5)



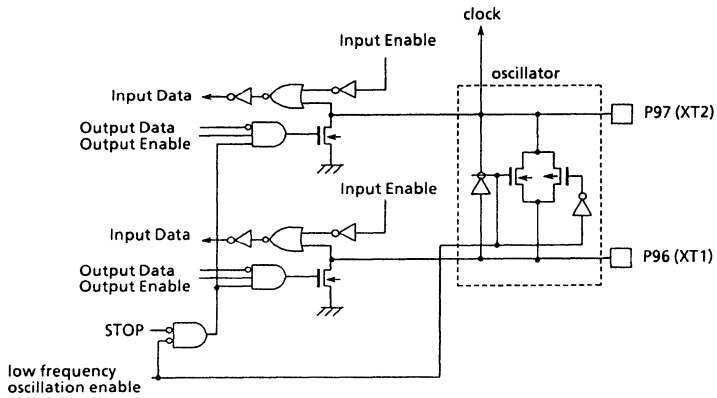
- P87 (INT0)



- P90 (TXD0), P93 (TXD1)



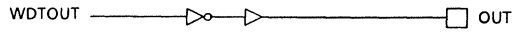
- P96 (XT1), P97 (XT2)



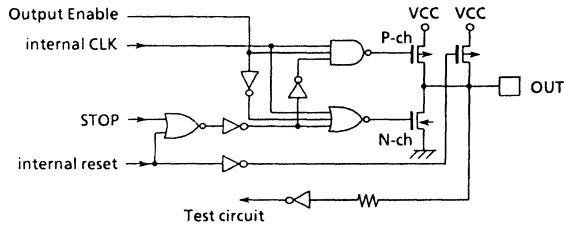
- $\overline{\text{NMI}}$



•  $\overline{\text{WDTOU}}$



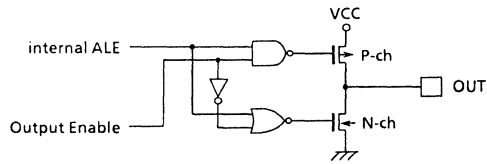
• CLK



• CSEL,  $\overline{\text{AM8/16}}$

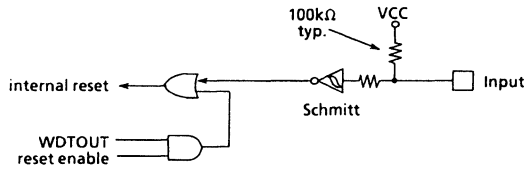


• ALE

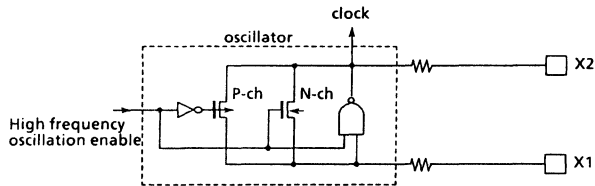




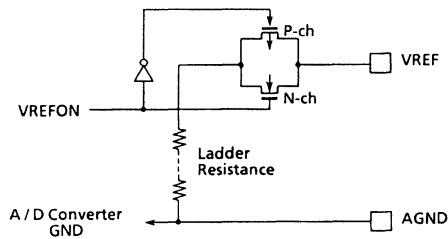
•  $\overline{\text{RESET}}$



• X1, X2



• VREF, AGND



## 7. Guidelines and Restrictions

### (1) Special Expression

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

Symbol <Bit Symbol>

ex) TRUN <TRUN> ... Bit TORUN 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) The case of oscillation frequency is 20MHz

$$2/20\text{MHz} = 100\text{ns} = 1 \text{ state}$$

### (2) Guidelines

① CESL, AM8/ $\overline{\text{T6}}$  pin

Fix these pins V<sub>CC</sub> or GND unless changing voltage.

② Standby Mode (IDLE1)

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

③ 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.

④ High Speed  $\mu$ DMA (DRAM refresh mode)

When the Bus is released ( $\overline{\text{BUSAK}} = "0"$ ) for waiting to accept the interrupt, DRAM refresh is not performed because of the high-speed  $\mu$ DMA is generated by an interrupt.

⑤ 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.

⑥ 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.

⑦ 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.

⑧ Watch Dog Timer

When the Bus is released, the watch dog timer cannot be operated.

⑨ A/D Converter

The ladder resistor between VREF and AGND pin can be cut by program to reduce the power consumption. When the standby mode is used, cut by program before "HALT" instruction is executed.

⑩ CPU (High Speed  $\mu$ DMA)

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.

**TAEC Regional  
Sales Offices**

**NORTHWEST  
Sunnyvale, CA**

TEL: (408) 737-9844  
FAX: (408) 737-9905

**SOUTHWEST  
Irvine, CA**

TEL: (714) 453-0224  
FAX: (714) 453-0125

**Denver, CO**

TEL: (303) 220-9848  
FAX: (303) 220-9856

**CENTRAL**

**Chicago, IL**

TEL: (708) 945-1500  
FAX: (708) 945-1044

**Dallas, TX**

TEL: (214) 480-0470  
FAX: (214) 235-4114

**NORTHEAST**

**Boston, MA**

TEL: (617) 224-0074  
FAX: (617) 224-1096

**Edison, NJ**

TEL: (908) 248-8070  
FAX: (908) 248-8030

**SOUTHEAST**

**Atlanta, GA**

TEL: (404) 368-0203  
FAX: (404) 368-0075

**Orlando, FL**

TEL: (407) 332-0966  
FAX: (407) 339-3377