CMOS 8-BIT MICROCONTROLLERS
TMP90C840AN/TMP90C841AN
TMP90C840AF/TMP90C841AF

1. OUTLINE AND CHARACTERISTICS

The TMP90C840A is a high-speed advanced 8-bit micro controller applicable to a variety of equipment.

With its 8-bit CPU, ROM, RAM, A/D converter, multi-function timer/event counter and general-purpose serial interface integrated into a single CMOS chip, the TMP90C840A allows the expansion of external memories for programs (up to 56K byte) and data (1M byte). The TMP90C841A is the same as the TMP90C840A but without the ROM.

The TMP90C840AN/841AN is a 64-pin shrink DIP product. (SDIP64-P-750)
The TMP90C840AF/841AF is a 64-pin flat package product. (QFP64-P-1420A)

The characteristics of the TMP90C840A include:

(1) Powerful instructions: 163 basic instructions, including
Multiplication, division, 16-bit arithmetic operations, bit manipulation instructions

(2) Minimum instruction executing time: 320 ns (at 12.5 MHz oscillation frequency)

(3) Internal ROM: 8K byte (The TMP90C841A does not have a built-in ROM.)

(4) Internal RAM: 256 byte

(5) Memory expansion
Program memory: 64K byte
Data memory: 1M byte

(6) 8-bit A/D converter (6 channels)

(7) General-purpose serial interface (1 channel)
Asynchronous mode, I/O interface mode

(8) Multi-function 16-bit timer/event counter (1 channel)

(9) 8-bit timers (4 channel)

(10) Stepping motor control port (2 channel)

(11) Input/Output ports (90C840A: 54 pins, 90C841A: 28pins)

(12) Interrupt function: 10 internal interrupts and 4 external interrupts

(13) Micro Direct Memory Access (DMA) function (11 channels)

(14) Watchdog timer

(15) Standby function (4 HALT mode)

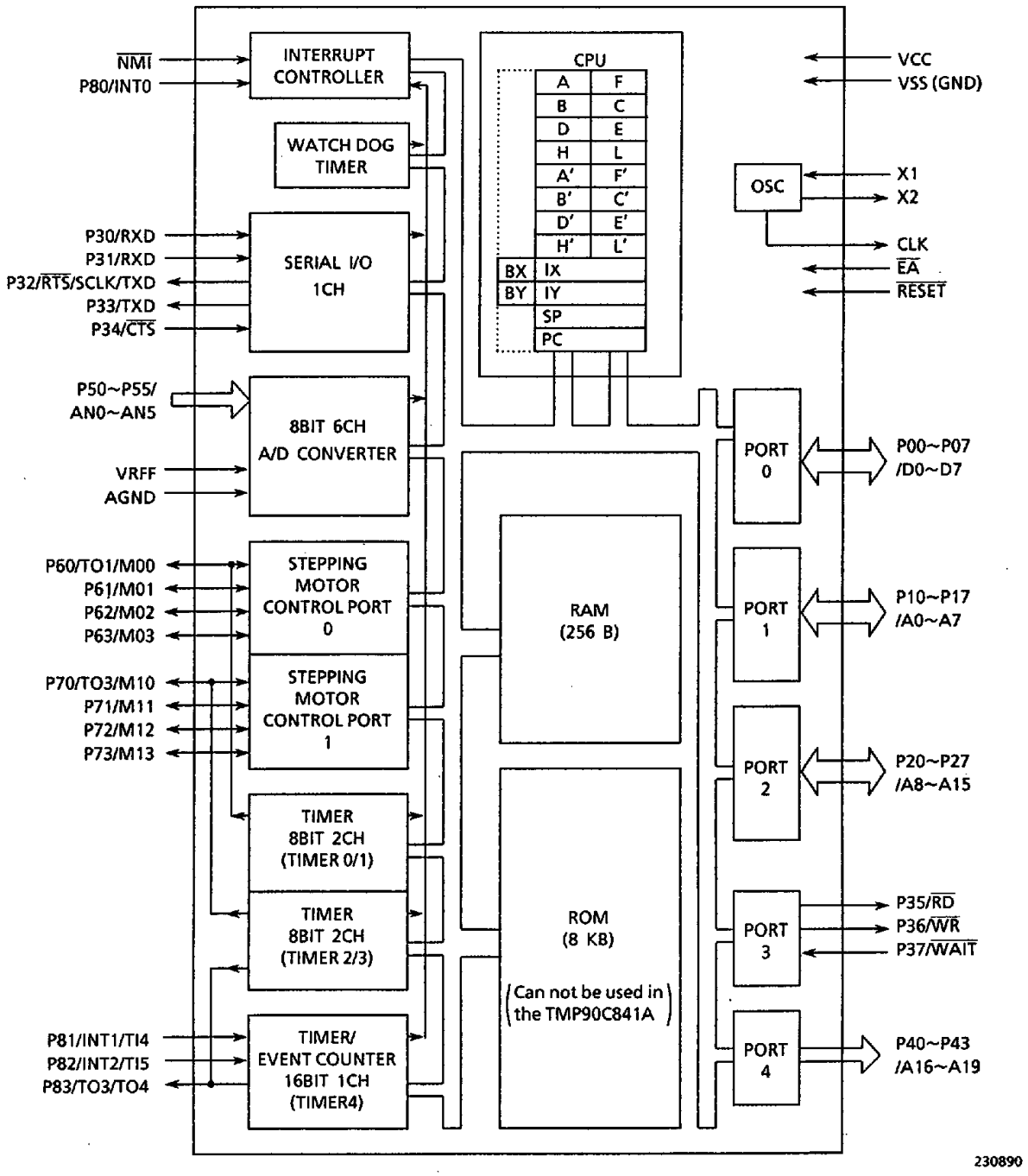


Figure 1 TMP90C840A Block Diagram

2. PIN ASSIGNMENT AND FUNCTIONS

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

2.1 Pin Assignment

Figure 2.1-(1)shows pin assignment of the TMP90C840AN/841AN.

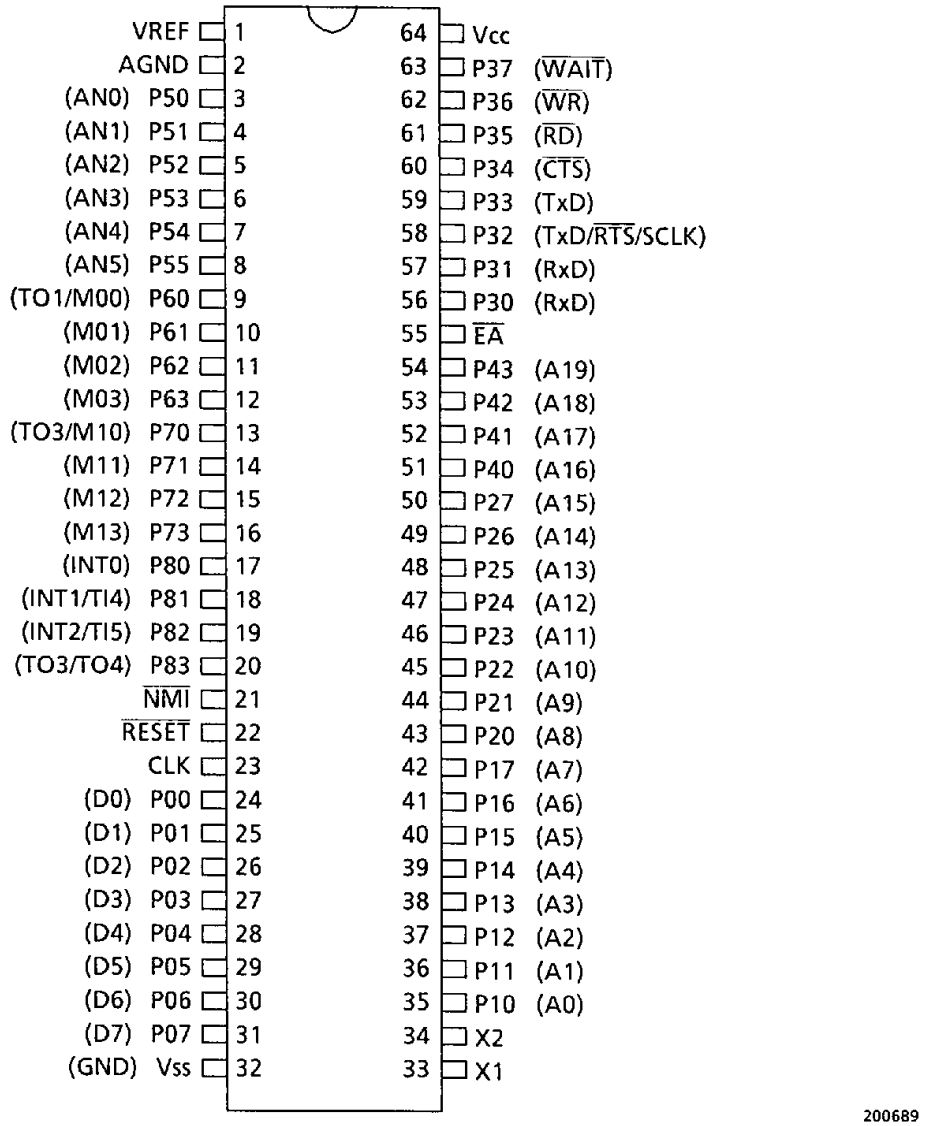
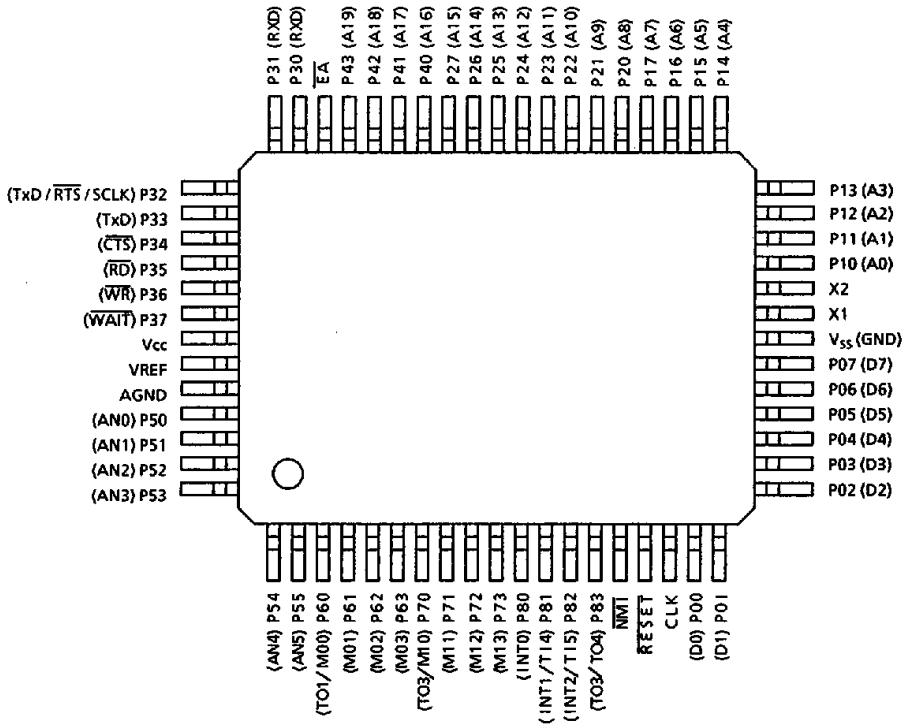


Figure 2.1- (1) Pin Assignment (Shrink Dual Inline Package)

Figure 2.1-(2) shows pin assigunment of the TMP90C840AF/841AF.



200689

Figure 2.1- (2) Pin Assignment (Flat Package)

2.2 Pin Names and Functions

The names of input/output pins and their functions are summarized in Table 2.2.

Table 2.2 Pin Names and Functions (1/2)

Pin Name	No. of pins	I/O 3 states	Function
P00~P07	8	I/O	Port 0: 8-bit I/O port that allows selection of input/output on byte basis
/D0~D7		3 states	Data bus: Also functions as 8-bit bidirectional data bus for external memory
P10~P17	8	I/O	Port 1: 8-bit I/O port that allows selection on byte basis
/A0~A7		Output	Address bus: The lower 8 bits address bus for external memory
P20~P27	8	I/O	Port 2: 8-bit I/O port that allows selection on bit basis
/A8~A15		Output	Address bus: The upper 8 bits address bus for external memory
P30	1	Input	Port 30: 1-bit input port
/RxD			Receiver Serial Data
P31	1	Input	Port 31: 1-bit input port
/RxD			Receiver Serial Data
P32	1	Output	Port 32: 1-bit output port
/TxD			Transmitter serial Data
/$\overline{RTS}$			Request to send serial data
/SCLK			Serial clock output
P33	1	Output	Port 33: 1-bit output port
/TxD			Transmitter Serial Data
P34	1	Input	Port 34: 1-bit input port
/$\overline{CTS}$			Clear to send Serial data
P35	1	Output	Port 35: 1-bit output port
/$\overline{RD}$			Read: Generates strobe signal for reading external memory
P36	1	Output	Port 36: 1-bit output port
/$\overline{WR}$			Write: Generates strobe signal for writing into external memory
P37	1	Input	Port 37: 1-bit input port
/$\overline{WAIT}$			Wait: Input pin for connecting slow speed memory or peripheral LSI
P40~P43	4	Output	Port 4: 4-bit output port that allows selection of Port/Address Bus on bit basis
/A16~A19			Address bus: Also functions as address bus for external memory (4 bits of bank address)
P50~P55	6	Input	Port 5: 6-bit input port
/AN0~AN5			Analog input: 6 analog inputs to A/D converter
VREF	1		Input of reference voltage to A/D converter

200689

Table 2.2 Pin Names and Functions (2/2)

Pin Name	No. of pins	I/O 3 states	Function
AGND	1		Ground pin for A/D converter
P60~P63 /M00~M03 /TO1	4	I/O	Port 6: 4-bit I/O port that allows I/O selection on bit basis
		Output	Stepping motor control port 0
		Output	Timer output 1: Output of Timer 0 or 1
P70~P73 /M10~M13 /TO3	4	I/O	Port 7: 4-bit I/O port that allows I/O selection on bit basis
		Output	Stepping motor control port 1
		Output	Timer output 3: Output of Timer 2 or 3
P80 /INT0	1	Input	Port 80: 1-bit input port
			Interrupt request pin 0: interrupt request pin (Level/rising edge is programmable)
P81 /INT1 /TI4	1	Input	Port 81: 1-bit input port
			Interrupt request pin 1: interrupt request pin (Rising/falling edge is programmable)
			Timer input 4: Counter/capture trigger signal for Timer 4
P82 /INT2 /TI5	1	Input	Port 82: 1-bit input port
			Interrupt request pin 2: rising edge interrupt request pin
			Timer input 5: capture trigger signal for Timer 4
P83 /TO3 /TO4	1	Output	Port 83: 1-bit output port
			Timer output 3/4: Output of Timer 2, 3 or 4
$\overline{NMI}$	1	Input	Non-maskable interrupt request pin: Falling edge interrupt request pin
CLK	1	Output	Clock output: Generates clock pulse at 1/4 frequency of clock oscillation. It is Pulled up internally during resetting.
$\overline{EA}$	1	Input	External access: Connects with Vcc pin in the TMP90C840A using internal ROM, and with GND pin in the TMP90C841A with no internal ROM.
$\overline{RESET}$	1	Input	Reset : Initializes the TMP90C840A/841A. (Built-in pull-up resister)
X1/X2	2	Input/ Output	Pins for quartz crystal or ceramic resonator
Vcc	1		Power supply (+ 5V)
Vss (GND)	1		Ground (0V)

230890

3. OPERATION

This chapter describes the functions and the basic operations of the TMP90C840A /841A in every block.

The following is a description of TMP90C840A which can also be applied to TMP90C841A, if not specifically defined otherwise.

3.1 CPU

TMP90C840A includes a high performance 8 bit CPU. For the function of the CPU, see the previous chapter "TLCS-90 CPU". This chapter explains exclusively the functions of the CPU of TMP90C840A which are not described in the chapter "TLCS-90 CPU".

3.1.1 Reset

The basic timing of the reset operation is indicated in Figure 3.1 (1). In order to reset the TMP90C840A, the $\overline{\text{RESET}}$ input must be maintained at the "0" level for at least ten system clock cycles (10 states : 2 μs at 10 MHz) within an operating voltage band and with a stable oscillation. When a reset request is accepted, all I/O ports (Port 0 /data bus D0 to D7, Port 1/address bus A0 to A7, Port 2/address bus A8 to A15, Port 6 and Port 7) function as input ports (high impedance state). The P35 ($\overline{\text{RD}}$), P36 ($\overline{\text{WR}}$) and CLK pins that always function as output ports turn to the "1" level, and the other output ports (Port 4/address bus A16 to A19 and P83) turn to the "0" level. The dedicated input ports remain unchanged. The registers of the CPU also remain unchanged. Note, however, that the program counter PC, the interrupt enable flag IFF and the bank registers BX and BY are cleared to "0". Register A shows an undefined status.

When the reset is cleared, the CPU starts executing instructions from the address 0000H.

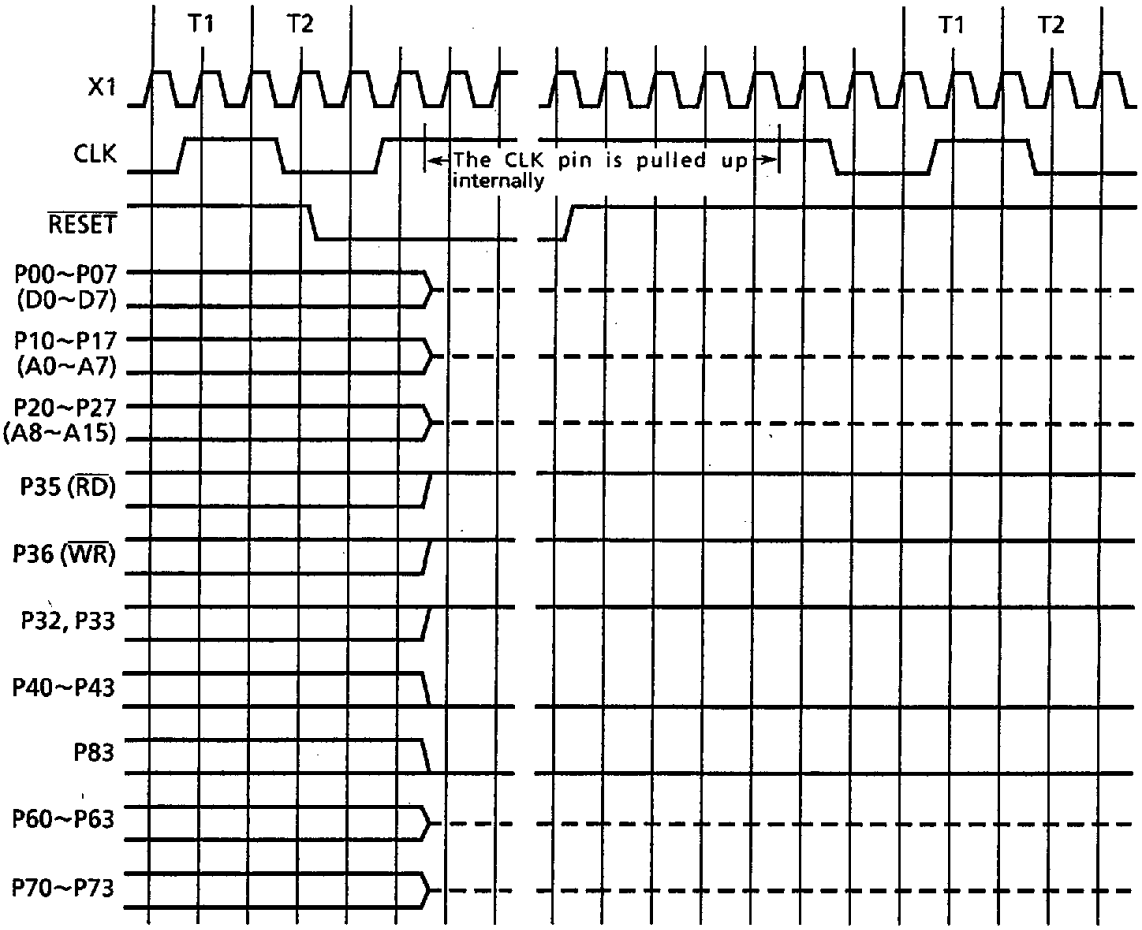


Figure 3.1 (1a) TMP90C840A Reset Timing

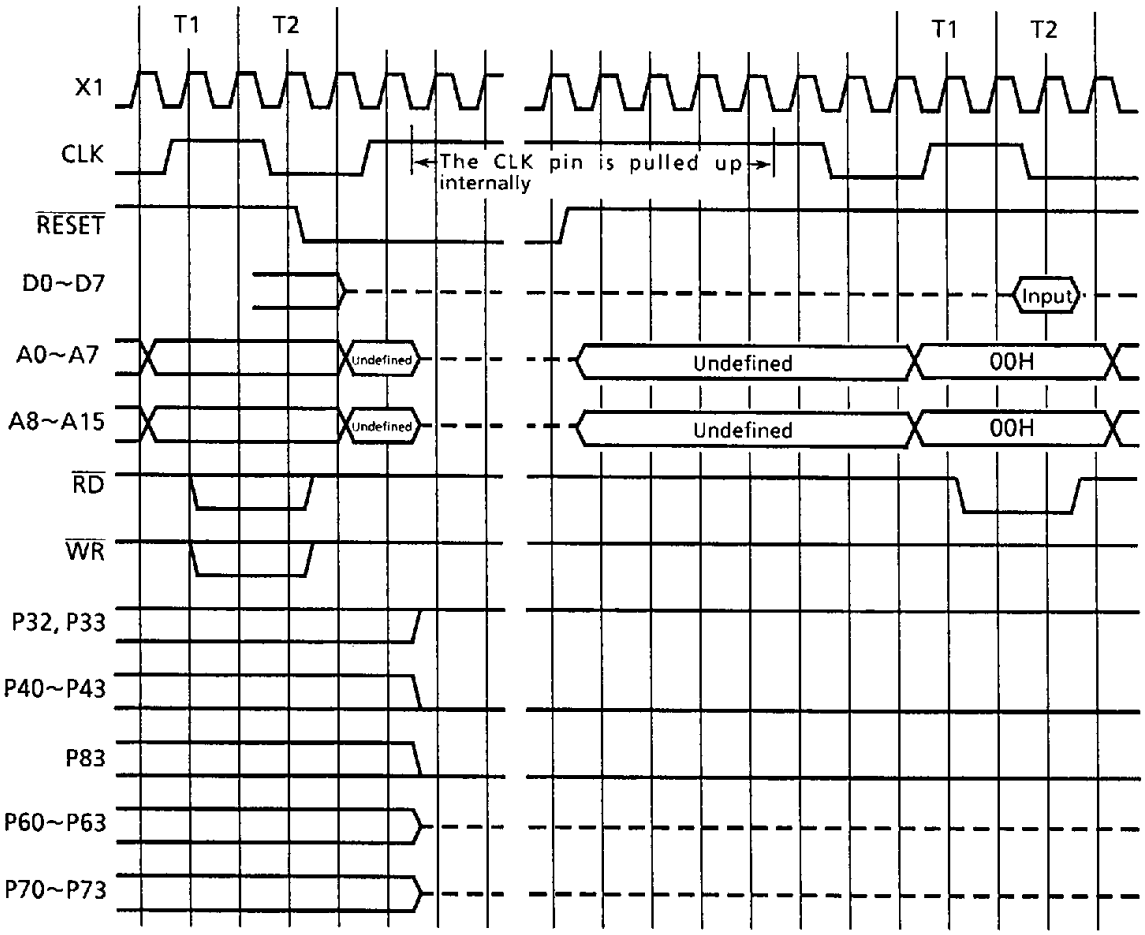


Figure3.1 (1b) TMP90C841A Reset Timing

3.1.2 EXF (Exchange Flag)

For TMP90C840A, "EXF", which is inverted when the command "EXX" is executed to transfer data between the main register and the auxiliary register, is allocated to the first bit of memory address FFD2H.

WDMOD (FFD2H)	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	EXF	DRIVE
Read/Write	R/W	R/W		R/W	R/W		R	R/W
Resetting Value	1	0	0	0	0	0	Undefined	0
Function	1: WDT Enable	WDT Detecting time 00: 2^{14}/fc 01: 2^{16}/fc 10: 2^{18}/fc 11: 2^{20}/fc		Warming up time 0: 2^{14}/fc 1: 2^{16}/fc	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Invert each time EXX instruction is executed	1: to drive pins in STOP mode

230890

3.1.3 Wait Control

For TMP90C840A, a wait control register (WAITC) is allocated to the 6th and 7th bits of memory address FFC7H.

P3CR (FFC7H)

	7	6	5	4	3	2	1	0
bit Symbol	WAITC1	WAITC0	RDE	ODE	TXDC1	TXDC0	RXDC1	RXDC0
Read/Write	R/W		R/W	R/W	R/W		R/W	
Resetting Value	0	0	0	0	0	0	0	0
Function	Wait control 00: 2state wait 01: normal wait 10: non wait 11: reserved		RD control 0: $\overline{RD}$ for only external access 1: Always $\overline{RD}$	P33 control 0: CMOS 1: Open drain	P33 00: Out 01: Out 10: TxD 11: TxD	P32 Out TxD Out $\overline{RTS}$/SCLK	P31 00: In 01: In 10: RxD 11: Not used	P30 In RxD In

230890

3.1.4 Bank Register

For TMP90C840A, BX and BY registers are allocated to memory addresses FFECH (BX register) and FFEDH (BY register), respectively. In these registers, only the low-order 4 bits are valid, and the high-order 4 bits are undefined. These undefined bits become "1" whenever they are read.

BX (FFECH)

	7	6	5	4	3	2	1	0
bit Symbol					BX3	BX2	BX1	BX0
Read/Write					R/W			
Resetting Value					0	0	0	0

BY (FFEDH)

	7	6	5	4	3	2	1	0
bit Symbol					BY3	BY2	BY1	BY0
Read/Write					R/W			
Resetting Value					0	0	0	0

230890

3.2 Memory Map

The TMP90C840A supports a program memory of up to 64K bytes and a data memory of maximum 1M bytes.

The program memory may be assigned to the address space from 00000H to 0FFFFH, while the data memory can be allocated to any address from 00000H to FFFFFH.

(1) Internal ROM

The TMP90C840A internally contains an 8K-byte ROM. The address space from 0000H to 1FFFH is provided to the ROM. The CPU starts executing a program from 0000H by resetting.

The addresses 0010H to 007FH in this internal ROM area are used for the entry area for the interrupt processing.

The TMP90C841A does not have a built-in ROM; therefore, the address space 0000H ~1FFFH is used as external memory space.

(2) Internal RAM

The TMP90C840A also contains a 256 byte RAM, which is allocated to the address space from FEC0H to FFBFH. The CPU allows the access to a certain RAM area (FF00H to FFBFH, 192 bytes) by a short operation code (opcode) in a "direct addressing mode".

The addresses from FF10H to FF7FH in this RAM area can be used as parameter area for micro DMA processing (and for any other purposes when the micro DMA function is not used).

(3) Internal I/O

The TMP90C840A provides a 48-byte address space as an internal I/O area, whose addresses range from FFC0H to FFEFH. This I/O area can be accessed by the CPU using a short opcode in the "direct addressing mode".

Figure 3.2 is a memory map indicating the areas accessible by the CPU in the respective addressing mode.

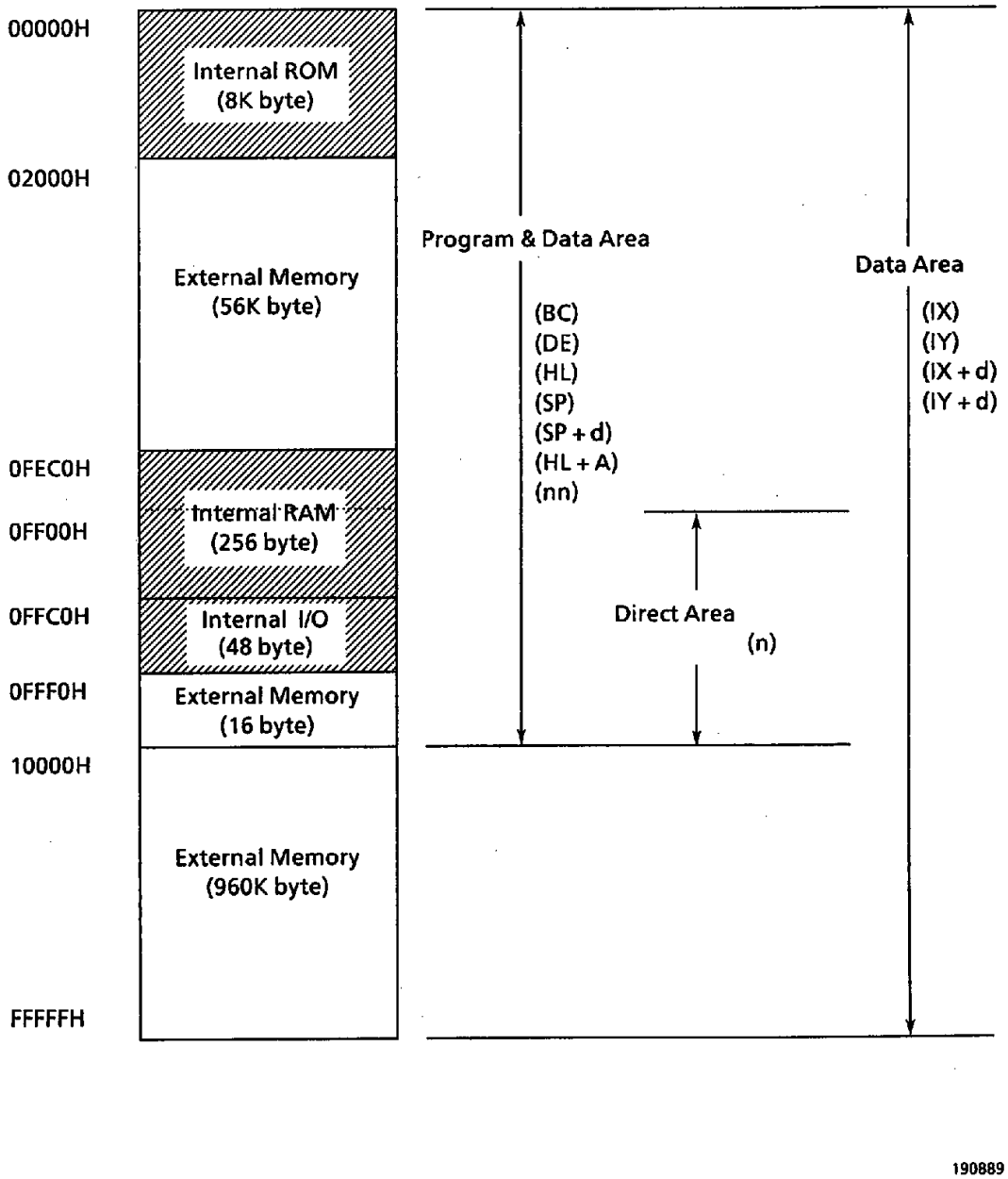


Figure 3.2 Memory Map

3.3 Interrupt Functions

The TMP90C840A supports a general purpose interrupt processing mode and a micro DMA processing mode that enables automatic data transfer by the CPU for internal and external interrupt requests. After the reset state is released, all interrupt requests are processed in the general purpose interrupt processing mode. However, they can be processed in the micro DMA processing mode by using a DMA enable register to be described later.

Figure 3.3 (1) is a flowchart of the interrupt response sequence.

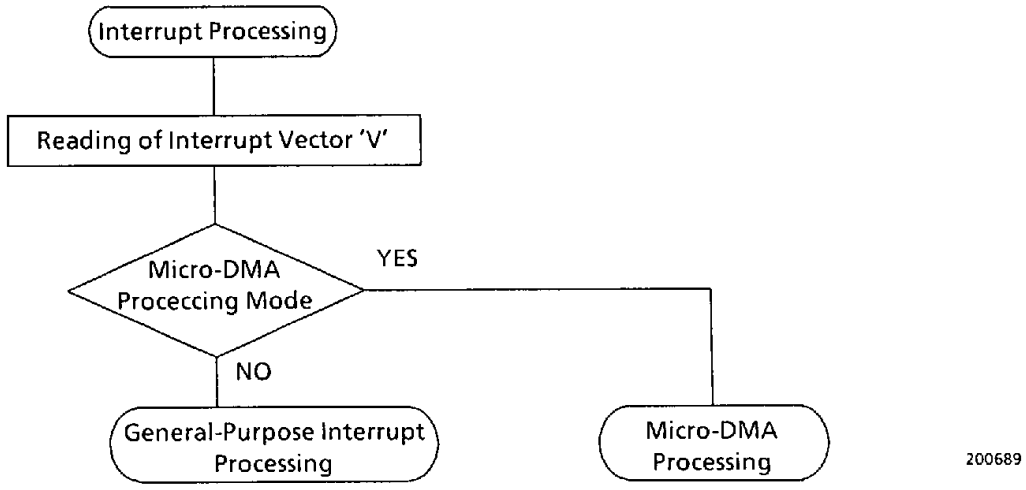


Figure 3.3 (1) Interrupt Response Flowchart

When an interrupt is requested, the source of the interrupt transmits the request to the CPU via an internal interrupt controller. The CPU starts the interrupt processing if it is a non-maskable or maskable interrupt requested in the EI state (interrupt enable flag (IFF) ="1"). However, a maskable interrupt requested in the DI state (IFF="0") is ignored. An interrupt request is sampled by the CPU at the falling edge of the CLK signal in the last bus cycle of each instruction.

Having acknowledged an interrupt, the CPU reads out the interrupt vector from the internal interrupt controller to find out the interrupt source.

Then, the CPU checks if the interrupt requests the general purpose interrupt processing or the micro DMA processing, and proceeds to each processing.

As the reading of an interrupt vecotors is performed in the internal operating cycles, the bus cycle results in dummy cycles.

3.3.1 General Purpose Interrupt Processing

A general purpose interrupt is processed as shown in Figure 3.3. (2).

The CPU stores the contents of the program counter PC and the register pair AF (including the interrupt enable flag (IFF) before the interrupt) into the stack, and resets the interrupt enable flag IFF to "0" (disable interrupts). In then transfers the value of the interrupt vector "V" to the program counter, and the processing jumps to an interrupt processing program.

The overhead for the entire process from accepting an interrupt to jumping to an interrupt processing program is 20 states.

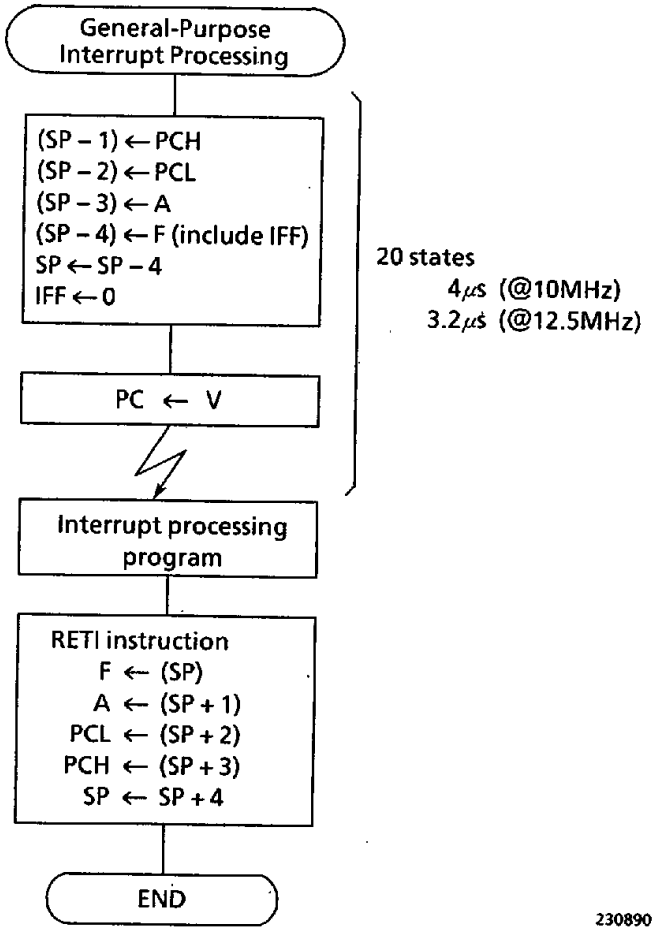


Figure 3.3 (2) General Purpose Interrupt Processing Flowchart

An interrupt (Maskable and Nonmaskable) processing program ends with a RETI instruction.

When this instruction is executed, the data previously stacked from the program counter PC and the register pair AF are restored. (Returns to the interrupt enable flag (IFF) before the interrupt.)

After the CPU reads out the interrupt vector, the interrupt source acknowledges that the CPU accepts the request, and clears the request.

A non-maskable interrupt cannot be disabled by programming. A maskable interrupt, on the other hand, can be enabled or disabled by programming. An interrupt enable flip flop (IFF) is provided on the bit 5 of Register F in the CPU. The interrupt is enabled or disabled by setting IFF to "1" by the EI instruction or to "0" by the DI instruction, respectively. IFF is reset to "0" by the reset operation or the acceptance of any interrupt (including non-maskable interrupt). The interrupt can be enabled after the subsequent instruction of EI instruction is executed.

Table 3.3 (1) lists the possible interrupt sources.

Table3.3 (1) Interrupt Sources

Priority order	Type	Interrupt sourece	Vector value ÷8	Vector value	Start address of general purpose interrupt processing	Start address of Mico DMA processing parameter
1	Non maskable	SWI instruction		10H	0010H	–
2		NMI (Input from $\overline{\text{NMI}}$ pin)		18H	0018H	–
3		INTWD (watchdog)		20H	0020H	–
4	Maskable	INT0 (External input 0)	05H	28H	0028H	FF28H
5		INTT0 (Timer 0)	06H	30H	0030H	FF30H
6		INTT1 (Timer 1)	07H	38H	0038H	FF38H
7		INTT2 (Timer 2)	08H	40H	0040H	FF40H
7		INTAD (A/D Converter)	08H	40H	0040H	FF40H
8		INTT3 (Timer 3)	09H	48H	0048H	FF48H
9		INTT4 (Timer 4)	0AH	50H	0050H	FF50H
10		INT1 (External input 1)	0BH	58H	0058H	FF58H
11		INTT5 (Timer 5)	0CH	60H	0060H	FF60H
12		INT2 (External input 2)	0DH	68H	0068H	FF68H
13		INTRX (End of serial receiving)	0EH	70H	0070H	FF70H
14		INTTX (End of serial transmission)	0FH	78H	0078H	FF78H

Note : Either INTT2 or INTAD is selected by INTEH<ADIS>. 200689

The "priority order" in the table shows the order of the interrupt source to be acknowledge by the CPU when more than one interrupt are requested at one time.

If interrupt of fourth and fifth orders are requested simultaneously, for example, an interrupt of the "5th" priority is acknowledged after a "4th" priority interrupt processing has been completed by a RETI instruction. However, a lower priority interrupt can be acknowledged immediately by executing an EI instruction in a program that processes a higher priority interrupt.

The internal interrupt controller merely determines the priority of the sources of interrupts to be acknowledged by the CPU when more than one interrupt are requested at a time. It is, therefore, unable to compare the priority of interrupt being executed with the one being requested.

To permit another interrupt during a certain interrupt operation, set the interrupt enabling flag for the source of the interrupt to be allowed, and execute the EI command.

3.3.2 Micro DMA Processing

Figure 3.3 (3) is a flowchart of the micro DMA processing. Parameters (addresses of source and destination, and transfer mode) for the data transfer between memories are loaded by the CPU from an address modified by an interrupt vector value. After the data transfer between memories according to these parameter, these parameters are updated and saved into the original locations. The CPU then decrements the number of transfers, and completes the micro DMA processing unless the result is "0".

If the number of transfer becomes "0", the CPU proceeds to the general purpose interrupt handling described in the previous chapter.

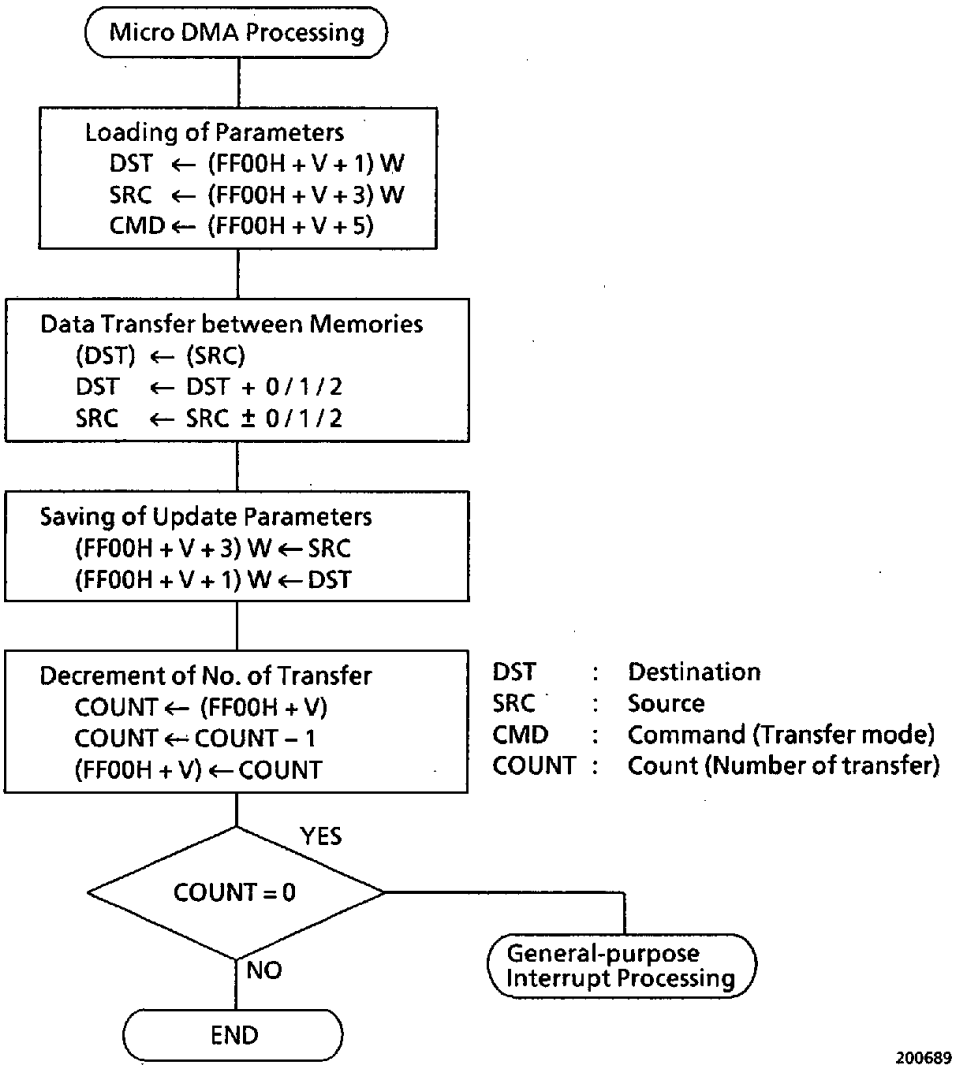


Figure 3.3 (3) Micro DMA Processing Flowchart

The micro DMA processing is performed by using only hardware to process interrupts mostly completed by simple data transfer. The use of hardware allows the micro DMA processing to handle the interrupt in a higher speed than the conventional methods using software. The CPU registers are not affected by the micro DMA processing.

Figure 3.3 (4) shows the functions of parameters used in the micro DMA processing.

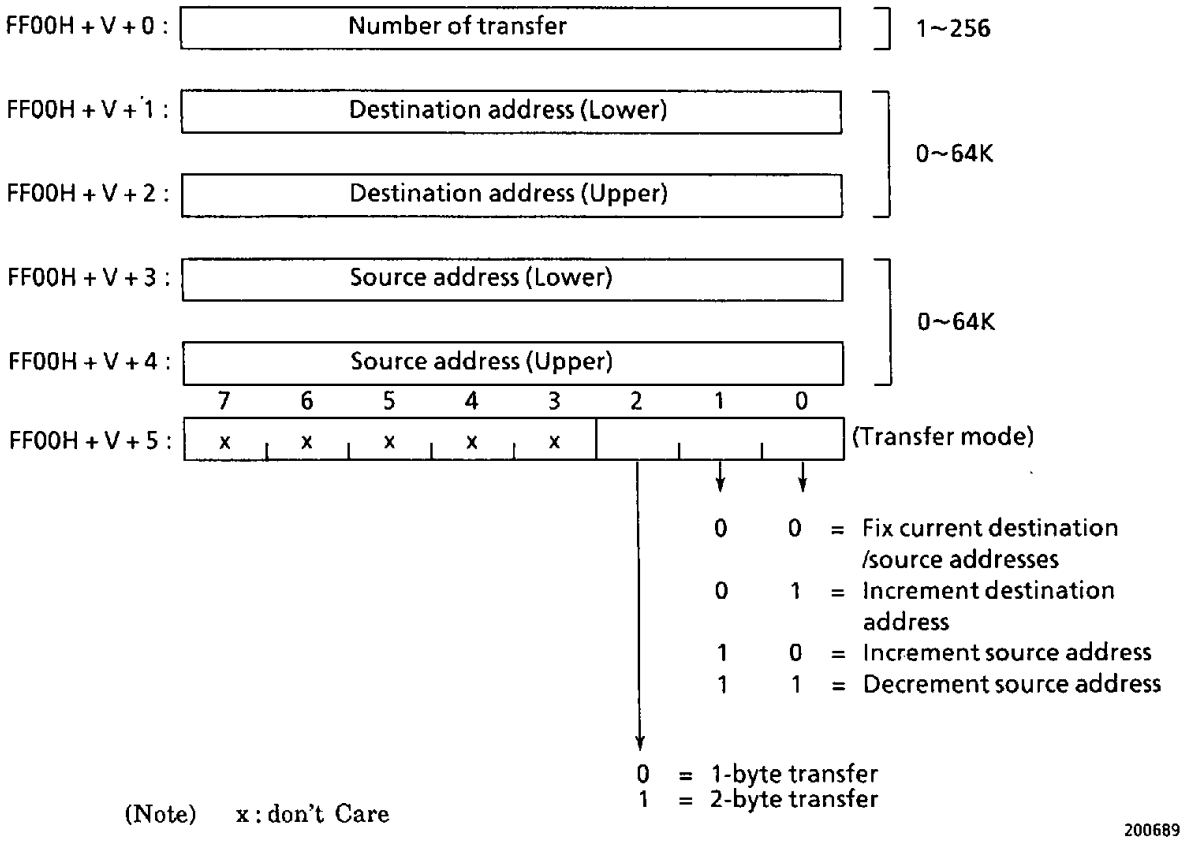


Figure 3.3 (4) Parameters for Micro DMA Processing

Parameters for the micro DMA processing are located in the internal RAM area (See Table 3.3 (1) Interrupt Sources). The start address of each parameter is "FF00H + interrupt vector value", from which a six bytes' space is used for the parameter. This space can be used for any other memory purposes if the micro DMA processing is not used.

The parameters normally consist of the number of transfer, addresses of destination and source, and transfer mode. The number of transfer indicates the number of data transfer accepted in the micro DMA processing.

The amount of data transferred by a single micro DMA processing is one or two bytes. The number of transfers is 256 when the number of transfers value is "00H". Both the destination and source addresses are specified by 2-byte data. The address space available for the micro DMA processing ranges from 0000H to FFFFH.

Bits 0 and 1 of the transfer mode indicates the mode updating the source and/or destination, and the bit 2 indicates the data length (one byte or two bytes).

Table 3.3 (2) shows the relation between the transfer mode and the result of updating the destination/source addresses.

Table 3.3 (2) Addresses Updated by Micro DMA Processing

Transfer mode	Function		Destination address	Source address
000	1-byte transfer :	Fix the current source/ destination addresses	0	0
001	1-byte transfer :	Increment the destination address	+1	0
010	1-byte transfer :	Increment the source address	0	+1
011	1-byte transfer :	Decrement the source address	0	−1
100	2-byte transfer :	Fix the current source/ destination addresses	0	0
101	2-byte transfer :	Increment the destination address	+2	0
110	2-byte transfer :	Increment the source address	0	+2
111	2-byte transfer :	Decrement the source address	0	−2

200689

In the 2-byte transfer mode, data are transferred as follows :

(Destination address) ← (Source address)
(Destination address+1) ← (Source address+1)

Similar data transfers are made in the modes that "decrement the source address", but the updated address are different as shown in the table 3.3 (2).

Address increment/decrement modes are applied to memory address space and fixed the addressing modes are applied to the I/O address space. Because of that, this micro-DMA was designed for both I/O to memory transfers and memory to I/O transfers.

Figure 3.3 (5) shows an example of the micro DMA processing that handles data receiving of internal serial I/O.

This is an example of executing "an interrupt processing program after serial data receiving" after receiving 7-frame data (Assume 1 frame=1 byte for this example) and saving them into the memory addresses from FF00H to FF06H.

```
CALL  SIOINIT          ;  Initial setting for serial receiving.
SET   1,(0FFE6H)       ;  Enable an interrupt for serial data receiving.
SET   1,(0FFE8H)       ;  Set the micro DMA processing mode for the interrupt.
LD    (0FF70H),7       ;  Set the number of transfer = 7
LDW   (0FF71H),0FF00H  ;  Set FF00H for the destination address.
LDW   (0FF73H),0FFEBH  ;  Set FFEBH for the source (serial receiving buffer) address.
LD    (0FF75H),1       ;  Set the transfer mode (1-byte transfer : Increment destination
                             address).

EI
:
:
ORG     0070H
```

Interrupt processing program after serial data receiving

```
RETI
```

Figure 3.3 (5) Example of Micro DMA Processing

The bus operation in the general purpose interrupt processing and the micro DMA processing is shown in "Table 1.4 (2) Bus Operation for Executing Instructions" in the previous section.

The micro DMA processing time (when the number of transfer is not decremented to 0) is 46 states without regard to the 1-byte/2-byte transfer mode.

Figure 3.3 (6) shows the interrupt processing flowchart.

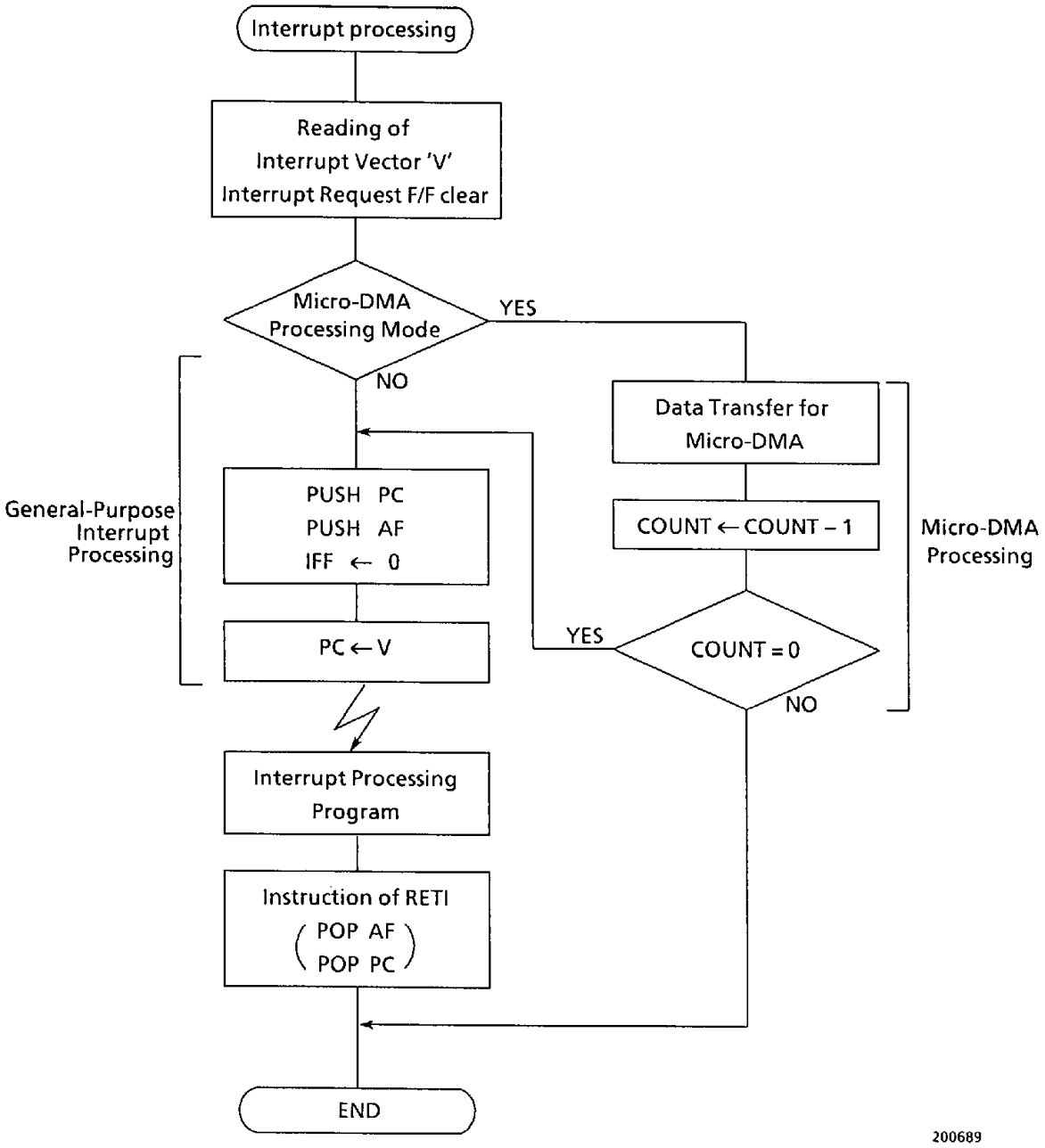


Figure 3.3 (6) Interrupt Processing Flowchart

3.3.3 Interrupt Controller

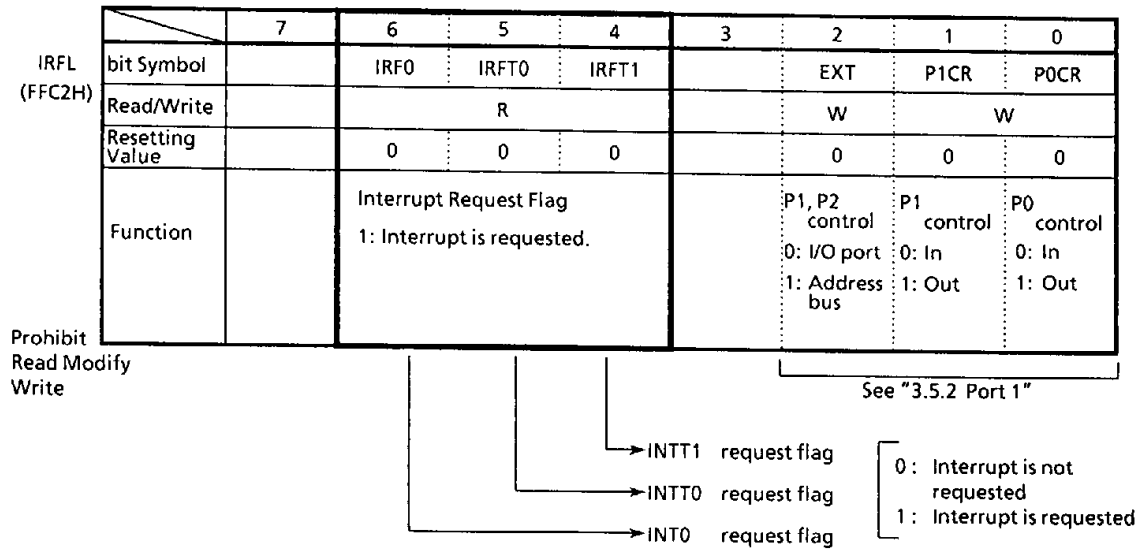
Figure 3.3 (8) outlines the interrupt circuit. The left side of this figure represents an interrupt controller, and the right side comprises the CPU's interrupt request signal circuit and HALT release signal circuit.

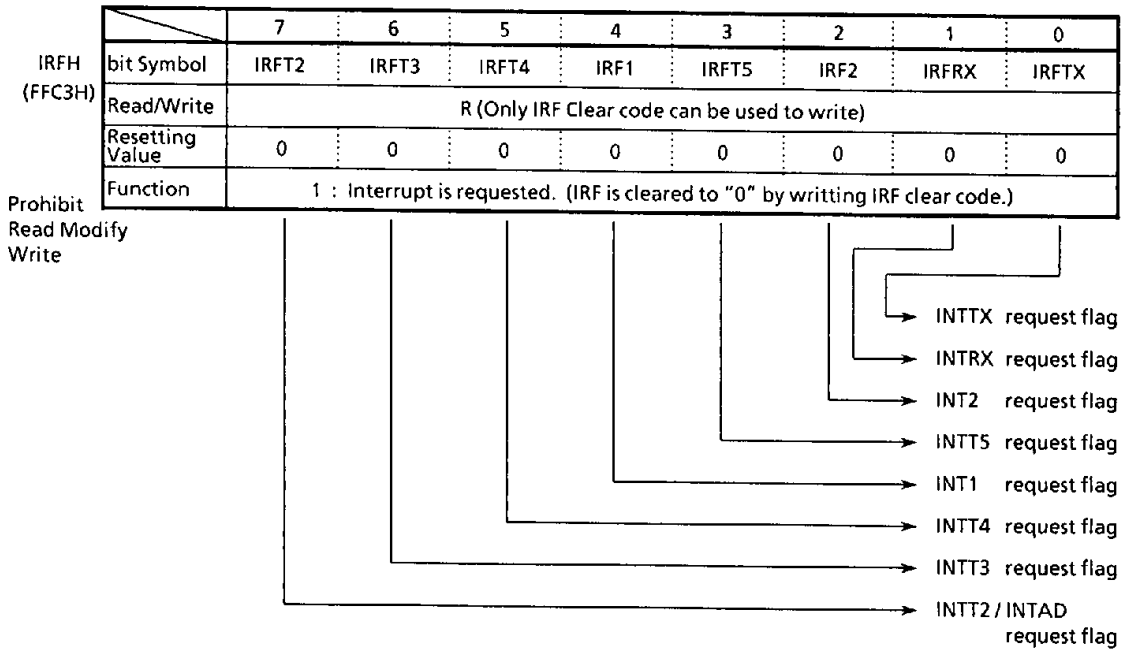
The interrupt controller consists of Interrupt Request Flip-flops, Interrupt Enable flags, and micro DMA enable flags allocated to each of 14 channels. The Interrupt Request Flip-flops serve to latch interrupt requests from peripherals. Each flip-flop is reset to "0" when a reset or interrupt is acknowledged by the CPU and the vector of the interrupt channel is read into the CPU, or when the CPU executes an instruction that clears an interrupt request flip-flop for the specified channel (write "vector divided by 8" into the memory address FFC3H). For example, by executing.

LD (FFC3H), 58H/8,

The Interrupt Request Flip-flops for the interrupt channel "INT1" whose vector is 58H is reset to "0". (When clearing the interrupt request flag assigned to address FFC2H, also write a clear code to the address FFC3H.)

The status of an Interrupt Request Flip-flops is found out by reading the memory address FFC2H or FFC3H. "0" denotes there is no interrupt request, and "1" denotes that an interrupt is requested. Figure 3.3 (7) illustrates the bit configuration indicating the status of Interrupt Request Flip-flops.





(Caution) Writing "vector divided by 8" into the memory address FFC3H clears the Flip-Flop for the specified interrupt request.

190990

Figure 3.3 (7) Configration of Interrupt Request Flip-flops

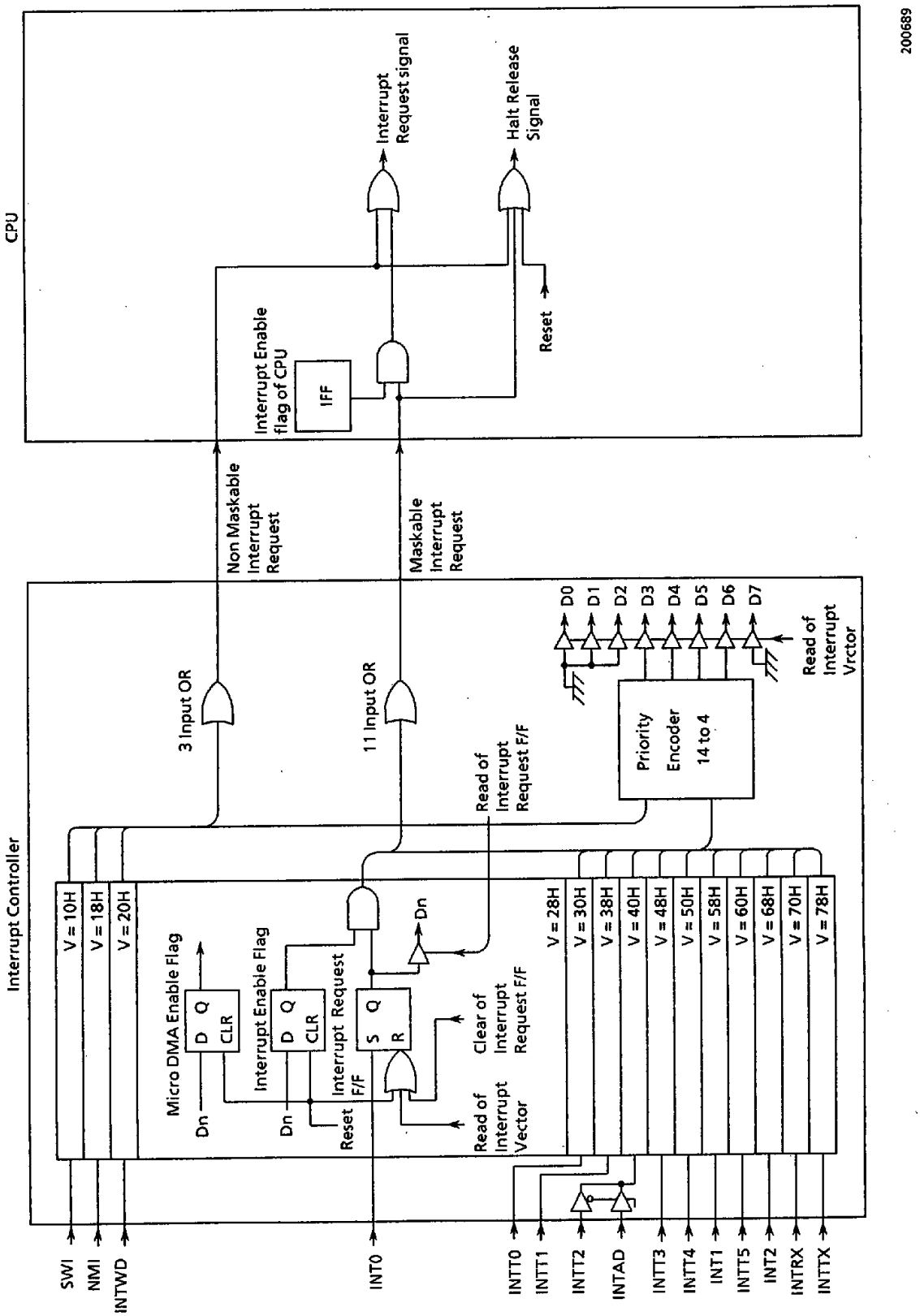


Figure 3.3 (8) Block Diagram of Interrupt Controller

The interrupt enable flags provided for all interrupt request channels are assigned to the memory address FFE6H or FFE7H. Setting any of these flags to "1" enables an interrupt of the respective channel. These flags are initialized to "0" by resetting.

Clear the interrupt enable flag in the DI status.

The micro DMA enable flag also provided for each interrupt request channel is assigned to the memory address FFE7H or FFE8H. The interrupt processing for each channel is placed in the micro DMA processing mode by setting this flag to "1". This flag is initialized to "0" (general purpose interrupt processing mode) by resetting.

Figure 3.3. (9) shows the bit configuration of the interrupt enable flags and micro DMA enable flags.

Interrupt by Timer 2 (INTT2) and that by A/D converter (INTAD) use a common interrupt request channel. The interrupt controller first accepts INTT2 after a reset. INTAD can be used by setting the "INTT2/INTAD selection bit" (ADIS: Bit 3 of memory address FFE7H) to "1".

The function of the external interrupts is as follows.

Interrupt	Common terminal	Mode	How to set
$\overline{NMI}$	—	Falling edge	———
INT0	P80	Level	P8CR<EDGE> = 0
		Rising edge	P8CR<EDGE> = 1
INT1	P81	Rising edge	T4MOD<CAPM1, 0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T4MOD<CAPM1, 0> = 1, 0
INT2	P82	Rising edge	———

For the pulse width for the external interrupts, see section 4.8 "Interrupt Operation".

Attention should be paid to the following three modes having special circuits:

INT0 Level mode	IF INT0 is not an edge-based interrupt, the function of Interrupt Request Flip-flop is canceled. Therefore the interrupt request signal must be held until the interrupt request is acknowledged by the CPU. A change in the mode (edge to level) automatically clears the interrupt request flag.
	When the CPU has been put in the interrupt response sequence with INT0 level mode, it is necessary to leave INT0 at "1" until the second bus cycle of the interrupt response sequence is completed. Also, "1" must always be held until HALT is cleared when using the INT0 level mode to clear HALT. (Use care to prevent noise changing "1" back to "0".)
	When switching from the level mode to the edge mode, the interrupt request flag set in the level mode is not cleared; therefore, use the following sequence to clear the interrupt request flag.
	DI
	LD (0FFD1H), 01H; switch from level to edge
	LD (0FFC3H), 05H; clear interrupt request flag
	EI

INTAD level mode	The Interrupt Request Flip-flop can be cleared only by resetting or reading the register that stores A/D conversion value, and cannot be cleared by an instruction. A change in the interrupt source (between INTAD and INTT2) automatically clears the interrupt request flag.
INTRX level mode	The Interrupt Request Flip-flop is cleared only by resetting or reading the serial channel receiving buffer, and not by an instruction.

031090

Interrupt enable flags

		7	6	5	4	3	2	1	0
INTEL (FFE6H)	bit Symbol	IET2	IET3	IET4	IE1	IET5	IE2	IERX	IETX
	Read/Write	R/W							
	Resettig Value	0	0	0	0	0	0	0	0
	Function	1 : Enable				0 : Disable			

- IET2 → INTTX interrupt enable flag
- IET3 → INTRX interrupt enable flag
- IET4 → INT2 interrupt enable flag
- IE1 → INTT5 interrupt enable flag
- IET5 → INT1 interrupt enable flag
- IE2 → INTT4 interrupt enable flag
- IERX → INTT3 interrupt enable flag
- IETX → INTT2/INTAD interrupt enable flag

Micro DMA and interrupt enable flags

		7	6	5	4	3	2	1	0
INTEH (FFE7H)	bit Symbol	0	DE0	DET0	DET1	ADIS	IE0	IET0	IET1
	Read/Write	R/W				R/W			
	Resettig Value	0	0	0	0	0	0	0	0
	Function	1 : Enable		0 : Disable		1 : INTAD	1 : Enable		0 : Disable

- IET1 → INTT1 interrupt enable flag
- IET0 → INTT0 interrupt enable flag
- IE0 → INT0 interrupt enable flag
- IET0 → INTT2/INTAD selection

0	INTT2
1	INTAD

- DET1 → INTT1 DMA enable flag
- DET0 → INTT0 DMA enable flag
- DE0 → INT0 DMA enable flag
- 0 → Write "0"

Micro DMA enable flags

		7	6	5	4	3	2	1	0
DMAEH (FFE8H)	bit Symbol	DET2	DET3	DET4	DE1	DET5	DE2	DERX	DETX
	Read/Write	R/W							
	Resettig Value	0	0	0	0	0	0	0	0
	Function	1 : Enable				0 : Disable			

- DET2 → INTTX DMA enable flag
- DET3 → INTRX DMA enable flag
- DET4 → INT2 DMA enable flag
- DE1 → INTT5 DMA enable flag
- DET5 → INT1 DMA enable flag
- DE2 → INTT4 DMA enable flag
- DERX → INTT3 DMA enable flag
- DETX → INTT2/INTAD DMA enable flag

230890

Figure 3.3 (9) Interrupt/Micro DMA Enable Flags

3.4 Standby Function

When a HALT instruction is executed, the TMP90C840A selects one of the following modes as determined by the halt mode set register:

(1)RUN : Suspends only the CPU operation. The power consumption remains unchanged.

(2)IDLE1 : Suspends all internal circuits except the internal oscillator. In this mode, the power consumption is less than 1/10 of that in the normal operation.

(3)IDLE2 : Operate only the internal oscillator and specific internal I/O devices. The power consumption is about 1/3 of that in the normal operation.

(4)STOP : Suspends all internal circuits including the internal oscillator. In this mode, the power consumption is considerably reduced.

The HALT mode set register WDMOD<HALTM 1,0> is assigned to the bits 2 and 3 of the memory address FFD2H in the internal I/O register area (other bits are used to control other functions). The register is reset to "00" (RUN mode) by resetting.

These HALT state can be released by resetting or requesting an interrupt.

The methods for releasing the HALT status are shown in Table 3.4 (2).

Either a non-maskable or maskable interrupt with EI (enable interrupt) condition is acknowledged and interrupt processing is processed. A maskable interrupt with DI (disable interrupt) condition is also acknowledged and CPU starts executing an instruction that follows the HALT instruction, but the interrupt request flag is held at "1".

But if interrupt request occur before MPU practices "HALT" command in the state of DI and it latches interrupt request flag, it causes to release HALT state and to state will be released as soon after MPU practices "HALT" command. (MPU doesn't HALT state.)

Therefore clear interrupt request flag or disable interrupt enable flag before MPU practices "HALT" command.

ex) MPU becomes STOP mode in the state of DI and refease it by INT0 interrupt. (But "built-in I/O" uses only "Timer 0")

```
DI
SET   2, (INTEH)        ;  INT0 interrupt enable
RES   1, (INTEH)        ;  INTT0 interrupt disable
LD    (WDMOD), 04H      ;  STOP mode
HALT
```

After release "HALT", Practice Program

When the halt status is released by a reset, the status in effect before entering the halt status (including built-in RAM) is held. The RAM contents may not be held, however, if the HALT instruction is executed within the built-in RAM.

WDMOD (FFD2H)	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	EXF	DRIVE
Read/Write	R/W	R/W		R/W	R/W		R	R/W
Resetting Value	1	0	0	0	0	0	Undefined	0
Function	1: Enable	WDT Detecting time 00: 2^{14}/fc 01: 2^{16}/fc 10: 2^{18}/fc 11: 2^{20}/fc		Warming up time 0: 2^{14}/fc 1: 2^{16}/fc	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Invert each time EXX instruction is executed	1: to drive pin in STOP mode.

See "3.10 Watchdog Timer"

See "3.4.4 STOP mode"

Exchange flag
See "3.1.2 Registers"

280391

Figure 3.4 (1) HALT Mode Set Register

3.4.1 RUN Mode

Figure 3.4 (2) shows the timing for releasing the HALT state by interrupts in the RUN/ IDLE 2 mode.

In the RUN mode, the system clock in the MCU continues to operate even after a HALT instruction is executed. Only the CPU stops executing the instruction. Until the HALT state is released, the CPU repeats dummy cycles. In the HALT state, an interrupt request is sampled with the rising edge of the "CLK" signal.

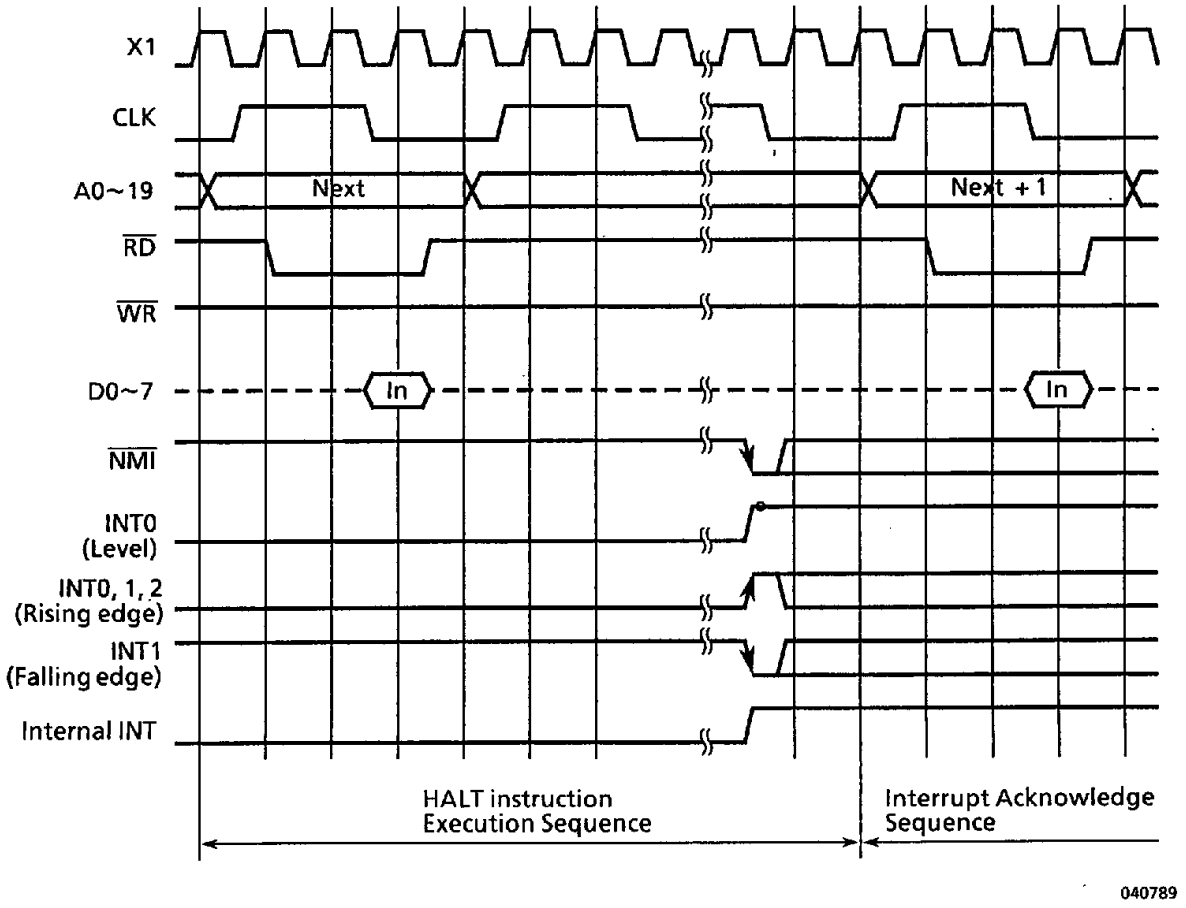


Figure 3.4 (2) Timing Chart for Releasing the HALT State by Interrupts in RUN/IDLE 2 Modes

3.4.2 IDLE 1 Mode

Figure 3.4 (3) illustrates the timing for releasing the HALT state by interrupts in the IDLE 1 mode.

In the IDLE 1 mode, only the internal oscillator and the watchdog timer operate. The system clock in the MCU stops, and the CLK signal is fixed at the "1" level.

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

Note : An interrupt requested by the watchdog timer is prohibited through the HALT period in this mode.

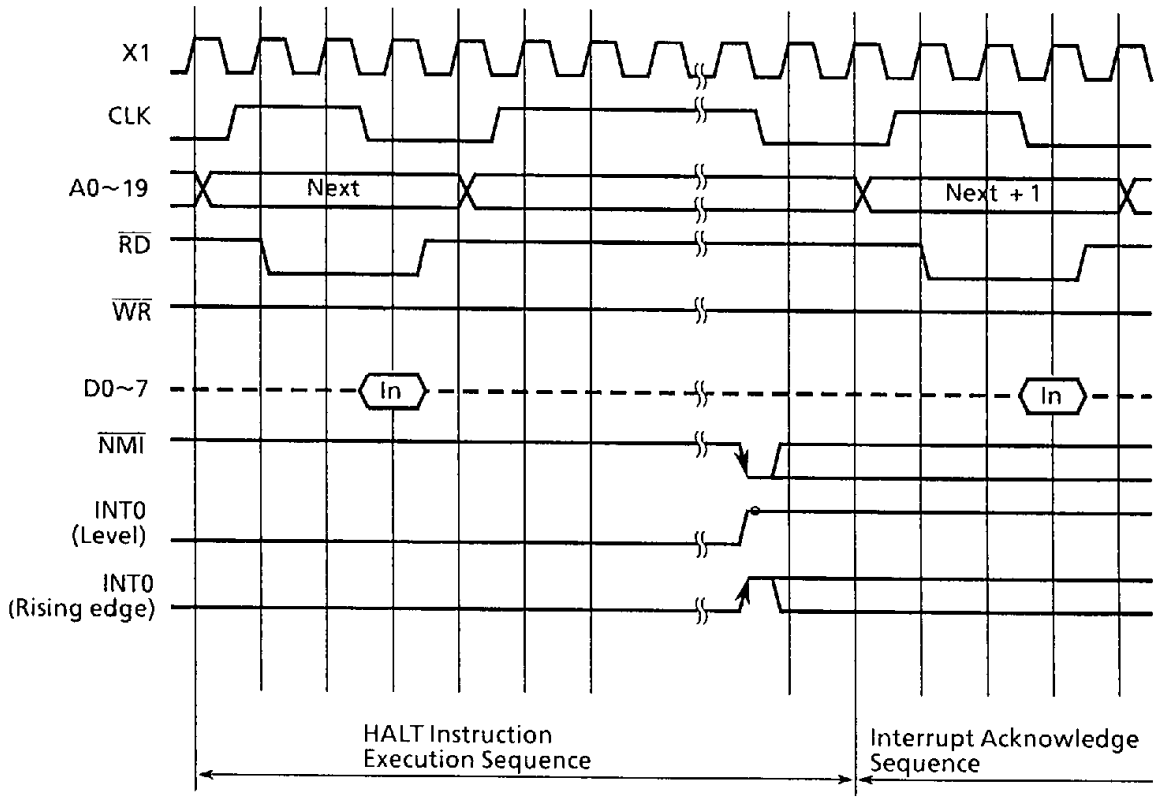


Figure 3.4 (3) Timing Chart of HALT Released by Interrupts in IDLE1 Mode

3.4.3 IDLE 2 Mode

Figure 3.4 (2) shows the timing of HALT release caused by interrupts in the RUN/IDLE 2 mode.

In the IDLE 2 mode, the HALT state is released by an interrupt with the same timing as in the RUN mode, except the internal operation of the MCU. In the RUN mode, only the CPU stops executing the current instruction, and the system clock is supplied to all internal devices. In the IDLE 2 mode, however, the system clock is supplied to only specific internal I/O devices. As a result, the HALT state in the IDLE 2 mode requires only a 1/3 of the power consumed in the RUN mode. In the IDLE 2 mode, the system clock is supplied to the following I/O devices:

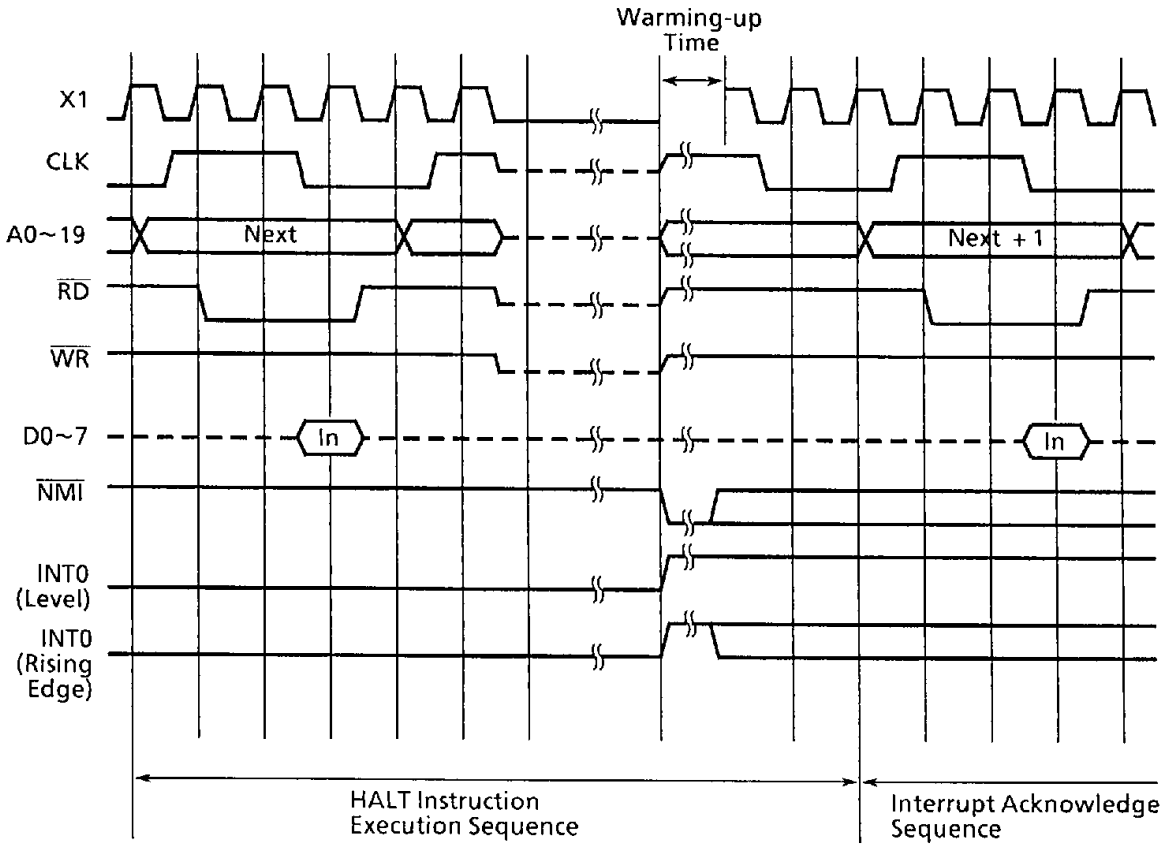
- 8-bit timer
- 16-bit timer
- Serial interface
- Watchdog timer

3.4.4 STOP Mode

Figure 3.4 (4) is a timing chart for releasing the HALT state by interrupts in the STOP mode.

The STOP mode is selected to stop all internal circuits including the internal oscillator. In this mode, all pins except special ones are put in the high-impedance state, independent of the internal operation of the MCU. Table 3.4 (1) summarizes the state of these pins in the STOP mode. Note, however, that the pre-halt state (The status prior to execution of HALT instruction) of all output pins can be retained by setting the internal I/O register WDMOD<DRVE> (Drive enable: Bit 0 of memory address FFD2H) to "1". The content of this register is initialized to "0" by resetting.

When the CPU accepts an interrupt request, the internal oscillator is restarted immediately. However, to get the stabilized oscillation, the system clock starts its output after the time set by the warming up counter WDMOD<WARM> (Warming up: Bit 4 of memory address FFD2H). A warming-up time of either the clock oscillation time $\times 2^{14}$ or $\times 2^{16}$ can be set by setting this bit to either "0' or "1". This bit is initialized to "0" by resetting.



200689

Figure 3.4 (4) Timing Chart of HALT Released by Interrupt in STOP Mode

The internal oscillator can be also restarted by the input of the $\overline{\text{RESET}}$ signal at "0" to the CPU. In the Reset restart mode, however, the warming-up counter remains inactive in order to get the quick response of MCU when the power is turned on (Power on Reset). As a result, the normal operation may not be performed due to the unstable clock supplied immediately after restarting the internal oscillator. To avoid this, it is necessary to keep the $\overline{\text{RESET}}$ signal at "0" long enough to release the HALT state in the STOP mode.

Table 3.4 (1) State of Pins in STOP Mode

	IN/OUT	90C840A		90C841A	
		DRVE = 0	DRVE = 1	DRVE = 0	DRVE = 1
P0	Input mode Output mode	— OUT	— OUT	— —	— —
P1	Input mode Output mode	— —	IN OUT	— —	 OUT
P2	Input mode Output mode	— —	IN OUT	— —	 OUT
P3	Input pin Output pin	— —	IN OUT	←	←
P4	Output pin	—	OUT	←	←
P5	Input pin	—	—	←	←
P6	Input mode Output mode	— OUT	IN OUT	←	←
P7	Input mode Output mode	— OUT	IN OUT	←	←
P80 (INT0) P81 (INT1) P82 (INT2) P83 (TO3/TO4)	Input pin Input pin Input pin Output pin	IN — — —	IN IN * IN * OUT	←	←
$\overline{\text{NMI}}$ CLK $\overline{\text{RESET}}$ X1 X2	Input pin Output pin Input pin Input pin Output pin	IN — IN — " 1 "	IN " 1 " IN — " 1 "	←	←

230890

*: Intermediate bias is still applied to this pin in the zero cross detect mode.

——: Indicates that input mode/input pin cannot be used for input and that the output mode/output pin have been set to high impedance.

IN: The input enable status.

IN: The input gate is operating. Fix the input voltage at either "0" or "1" to prevent the pin floating.

OUT: The output status.

It is necessary to leave INT0 at "1" until the second bus cycle of the interrupt response sequence is completed, when the STOP mode is released by the level mode of INT0.

Table 3.4 (2) I/O Operation During Halt and How to Release the Halt Command

Halt mode			RUN	IDLE2	IDLE1	STOP
WDMOD <HALTM1,0>			00	11	10	01
Operating block	CPU		Halt			
	I/O port		Keeps the state when the halt command was executed.			See Table 3.4 (1)
	8 bit timer		Operation		Halt	
	16 bit timer					
	Stepping motor controller					
	Serial interface					
	A/D converter					
	Watchdog timer					
	Interrupt controller					
Halt releasing source	Interrupt	NMI	○	○	○	○
		INTWD	○	○	—	—
		INT0	○	○	○	○
		INTT0	○	○	—	—
		INTT1	○	○	—	—
		INTT2	○	○	—	—
		INTAD	○	—	—	—
		INTT3	○	○	—	—
		INTT4	○	○	—	—
		INT1	○	○	—	—
		INTT5	○	○	—	—
		INT2	○	○	—	—
		INTRX	○	○	—	—
		INTTX	○	○	—	—
	Reset		○	○	○	○

280391

○ : Can be used to release the halt commnad.
— : Cannot be used to release the halt command.

3.5 Function of Ports

The TMP90C840A contains total 54 pins (TMP90C841A: 28 pins) input/output ports. These ports function not only for the general-purpose I/O but also for the input/output of the internal CPU and I/O. Table 3.5 describes the functions of these ports.

Table 3.5 Functions of Ports

Port name	Pin name	No. of pins	Direction	Direction set unit	Resetting value	Pin name for internal function
Port 0	P00~P07	8	I/O	Byte	Input	D0~D7
Port 1	P10~P17	8	I/O	Byte	Input	A0~A7
Port 2	P20~P27	8	I/O	Bit	Input	A8~A15
Port 3	P30	1	Input	—	Input	RxD
	P31	1	Input	—	Input	RxD
	P32	1	Output	—	Output	TxD/$\overline{\text{RTS}}$/SCLK
	P33	1	Output	—	Output	TxD
	P34	1	Input	—	Input	$\overline{\text{CTS}}$
	P35	1	Output	—	Output	$\overline{\text{RD}}$
	P36	1	Output	—	Output	$\overline{\text{WR}}$
	P37	1	Input	—	Input	$\overline{\text{WAIT}}$
Port 4	P40~P43	4	Output	—	Output	A16~A19
Port 5	P50~P55	6	Input	—	Input	AN0~AN5
Port 6	P60~P63	4	I/O	Bit	Input	M00~M03/TO1
Port 7	P70~P73	4	I/O	Bit	Input	M10~M13/TO3
Port 8	P80	1	Input	—	Input	INT0
	P81	1	Input	—	Input	INT1/TI4
	P82	1	Input	—	Input	INT2/TI5
	P83	1	Output	—	Output	TO3/TO4

230890

These port pins function as the general-purpose input/output ports by resetting. The port pins, for which input or output is programmably selectable, function as input ports by resetting. A separate program is required to use them for an internal function.
The TMP90C841A functions in the same way as the TMP90C840A except:

- Port 0 always functions as a data bus (D0 to D7).
- Port 1 always functions as Address bus (A0 to A7).
- Port 2 always functions as Address bus (A8 to A15).
- P35 and P36 of Port 3 always function as $\overline{\text{RD}}$ and $\overline{\text{WR}}$ pins, respectively.

3.5.1 Port 0 (P00~P07)

Port 0 is an 8-bit general-purpose I/O port PO whose I/O function is specified by the control register P01CR<P0C> in byte. By resetting all bits of the control register are initialized to "0", whereby Port 0 turns to the input mode, and the contents of the output latch register are undefined.

In addition to the general-purpose I/O port function, it functions as a data bus (D0~D7). Access of an external memory makes it automatically function as a data bus and <P0C> are cleared to "0".

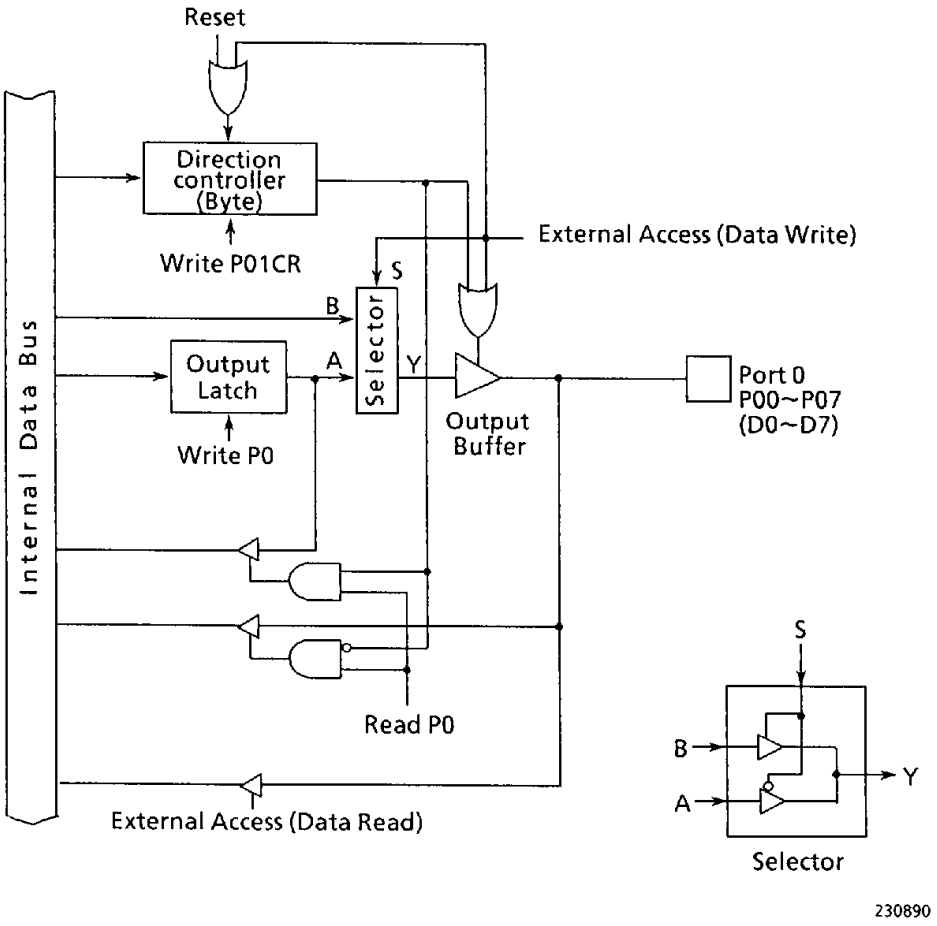


Figure 3.5 (1) Port 0

3.5.2 Port 1 (P10~P17)

Port 1 is an 8-bit general-purpose I/O port P1 whose I/O function is specified by the control register P01CR<P1C> in byte. All bits of the output latch and the control register are initialized to "0" by resetting, whereby Port 1 is put in the input mode.

In addition to the general-purpose I/O port function, it functions as an address bus (A0~A7). The address bus function can be selected by setting only the external extension control register P01CR<EXT> to "1" regardless of the status of the above control register <P1C>. The register <EXT> is reset to "0" whereby Port 1 and Port 2 turn to the general-purpose I/O mode.

The register <EXT> of the TMP90C841A is always set to "1" so that Port 1 functions as an address bus (A0~A7).

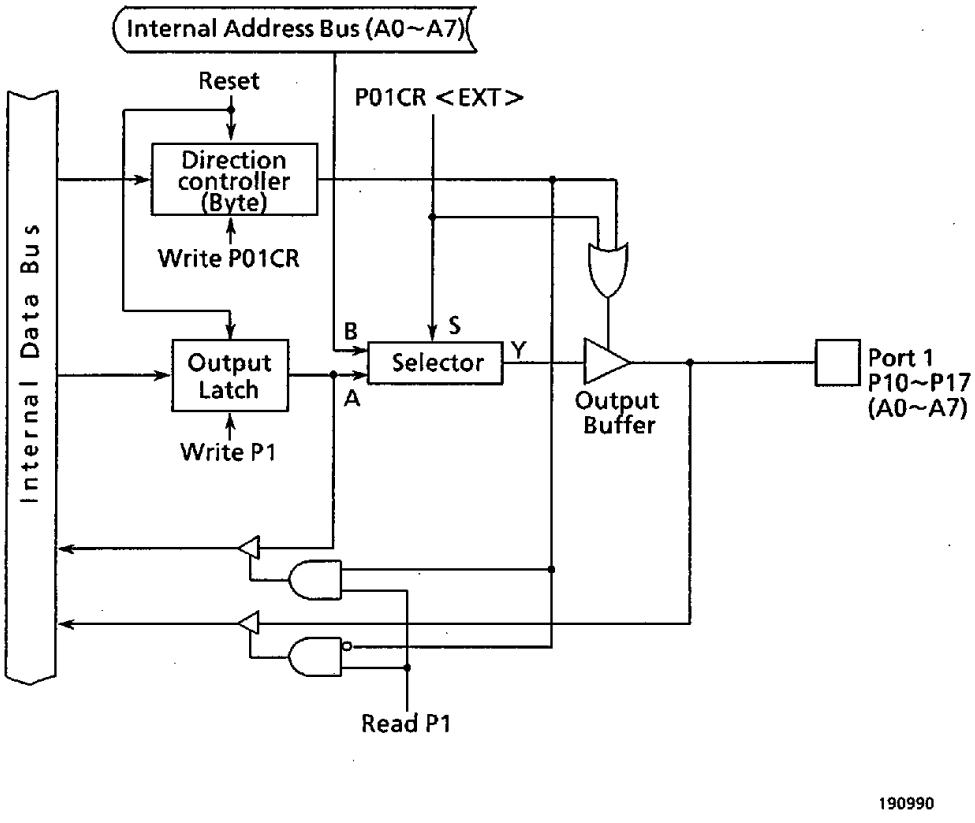


Figure 3.5 (2) Port 1

Port 0 Register

		7	6	5	4	3	2	1	0
P0 (FFC0H)	bit Symbol	P07	P06	P05	P04	P03	P02	P01	P00
	Read/Write	R/W							
	Resetting Value	Input Mode							

Port 1 Register

		7	6	5	4	3	2	1	0
P1 (FFC1H)	bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
	Read/Write	R/W							
	Resetting Value	Input Mode							

Port 0,1 Control Register

		7	6	5	4	3	2	1	0
P01CR (FFC2H)	bit Symbol		IRF0	IRFT0	IRFT1		EXT	P1C	P0C
	Read/Write		R				W	W	W
	Resetting Value		0	0	0		0	0	0
	Function		Interrupt Request Flag 1: Interrupt is requested				P1, P2 Control 0: I/O port 1: Address bus	P1 Control 0: In 1: Out	P0 Control 0: In 1: Out

Prohibit Read Modify Write.

R

See "3.3.3 Interrupt controller"

Set I/O of Port 0

0	Input
1	Output

Set I/O of Port 1

0	Input
1	Output

Set general-purpose port/address bus of Ports 1 and 2

0	General-purpose port
1	Address bus

280391

Figure 3.5 (3) Registers for Port 0 and 1

3.5.3 Port 2 (P20~P27)

Port 2 is an 8-bit general-purpose I/O port P2 whose I/O functions are specified by the control register P2CR for each bit. All bits of the output latch and the control register are initialized to "0" by resetting, where by Port 2 turns to the input mode.

In addition to the general-purpose I/O port function, it functions as an address bus (A8~A15). The address bus function can be selected by setting the register P01CR <EXT> (shared with port 1) to "1" and setting the Port 2 control register P2CR to the output mode. When the Port 2 control register P2CR is set to "0", Port 2 functions as an input port, regardless of the status of the EXT register.

For the TM90C841A, all bits of the register <EXT> and the control register are always set to "1", and Port 2 functions as an address bus (A8 to A15).

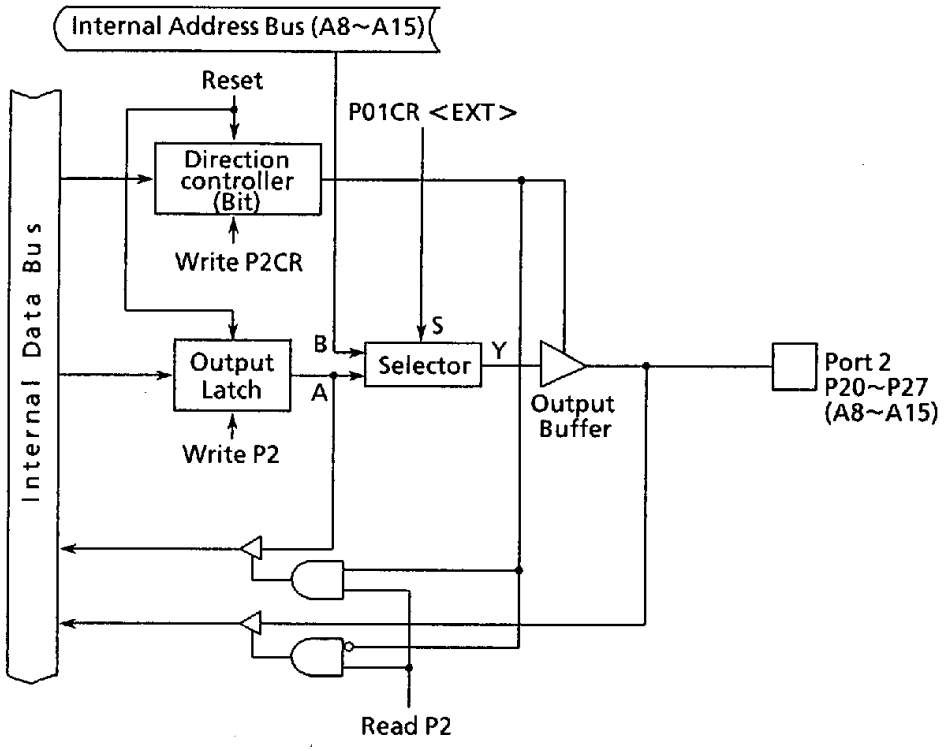


Figure 3.5 (4) Port 2

Port 2 Register

		7	6	5	4	3	2	1	0
P2 (FFC4H)	bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R / W							
	Resetting Value	Input Mode							

Port 2 Control Register

		7	6	5	4	3	2	1	0
P2CR (FFC5H)	bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
	Read/Write	W							
	Resetting Value	0	0	0	0	0	0	0	0
Prohibit Read Modify Write	Function	0 : In 1 : Out (I/O selected bit by bit)							

Set I/O of Port 2 and Address Bus

P2CR <P2XC> \ P01CR <EXT>	0	1
0	Input Port	Input Port
1	Output Port	Address Bus

Note : Settings can be in units of bits. Here, P2CR<P2XC> is the Xth bit of P2CR.

190990

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 P3 with fixed I/O function. All bits of the output latch are initialized to "1" by resetting, and "High level" is generated to the output port.

In addition to the I/O port function, P30~P34 have the I/O function for the internal serial interface, while P35~P37 have the external memory control function. The additional functions can be selected by the control register P3CR. All bits of the control register are initialized to "0" by resetting, and the port turns to the general-purpose I/O Ports mode.

However, access of an external memory makes P35~P36 automatically function as the memory control pins ($\overline{RD}$ and $\overline{WR}$), and access of an internal memory makes them function as general-purpose I/O ports.

When an external memory is accessed, therefore, the output latch registers P35 ($\overline{RD}$) and P36 ($\overline{WR}$) should be kept at "1" which is the initial value after the reset.

The P3CR <RDE> of the control register is intended for a pseudostatic RAM. When set to "1", it always functions as an $\overline{RD}$ pin. Therefore the $\overline{RD}$ pin outputs "0" (Enable) when it is an internal memory read and internal I/O read cycle.

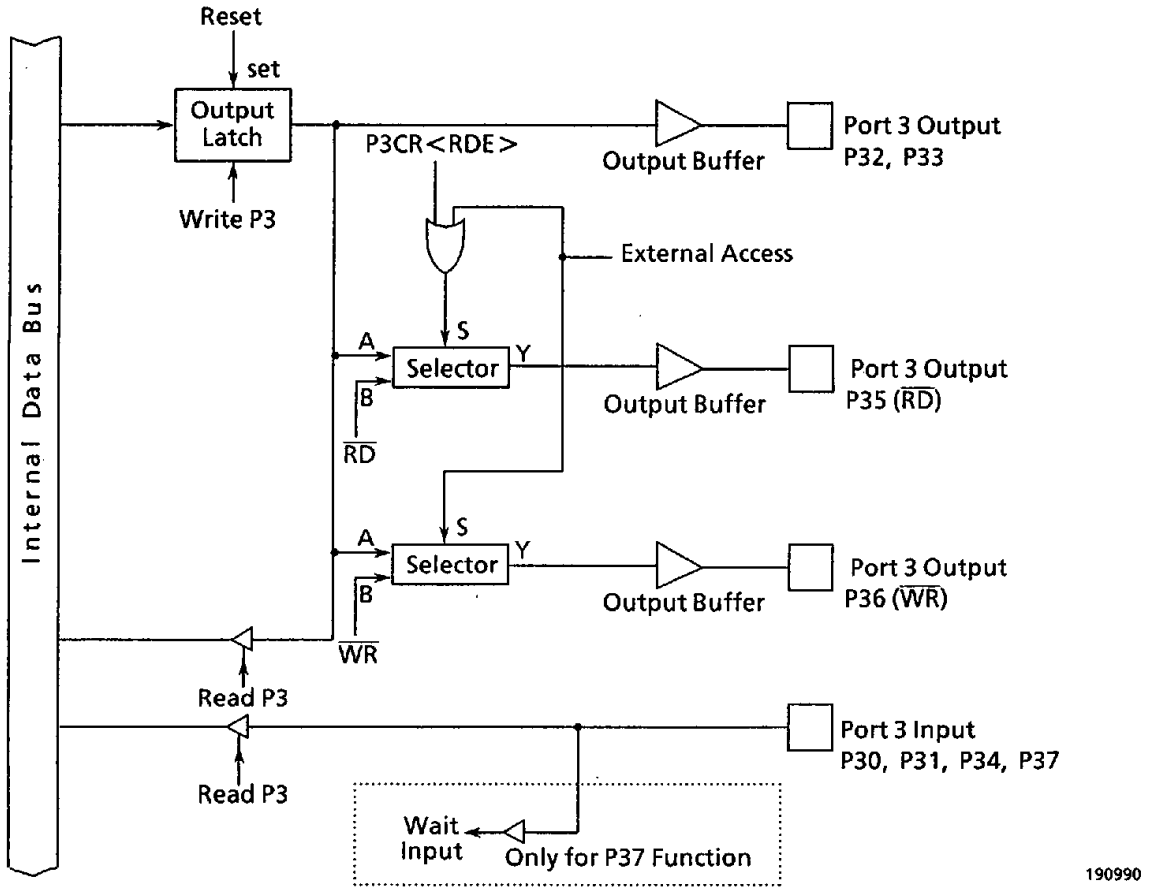


Figure 3.5 (6) Port 3

Port 3 Register

		7	6	5	4	3	2	1	0
P3 (FFC6H)	bit Symbol	P37	P36	P35	P34	P33	P32	P31	P30
	Read/Write	R	R/W	R/W	R	R/W	R/W	R	R
	Resetting Value	Input	1	1	Input	1	1	Input	Input

Port 3 Control Register

		7	6	5	4	3	2	1	0
P3CR (FFC7H)	bit Symbol	WAITC1	WAITC0	RDE	ODE	TXDC1	TXDC0	RXDC1	RXDC0
	Read/Write	R/W		R/W	R/W	R/W		R/W	
	Resetting Value	0	0	0	0	0	0	0	0
	Function	Wait Control 00: 2state wait 01: normal wait 10: non wait 11: Not used		RD Control 0: $\overline{RD}$ for only external access 1: Always $\overline{RD}$	P33 Control 0: CMOS Output 1: Open Drain Output	P33 00: Out 01: Out 10: TxD 11: TxD	P32 Out TxD Out $\overline{RTS}$/SCLK	P31 00: In 01: In 10: RxD 11: Not used	P30 In RxD In

See "3.8 Serial Channel"

Set port P33 to open drain output

0	CMOS output
1	Open drain output

Set port P35 to fixed $\overline{RD}$ mode

0	General-purpose I/O port
1	Fixed as $\overline{RD}$ Pin

WAIT control

00	2-state wait
01	Normal wait
10	No wait (used as an Input port)
11	Not used

230890

Figure 3.5 (7) Register for Ports 3

3.5.5 Port 4 (P40~P43)

Port 4 is a 4-bit port P4 intended only for the output. All bits of the output latch are initialized to "0" by resetting, and "0" is generated to the port.

In addition to the output port function, it works as an address bus (A16~A19). The selection of the address bus function is made by the control register P4CR. The output port or address bus function can be selected for each bit. All bits of the control register are initialized to "0" by resetting, and the port turns to the output mode.

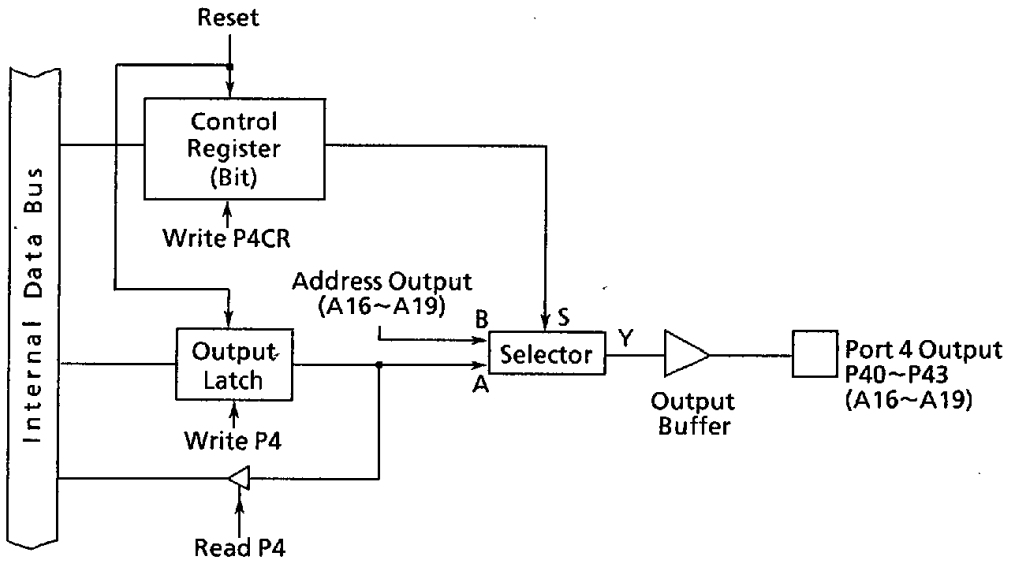


Figure 3.5 (8) Port 4

Port 4 Register

P4 (FFC8H)

	7	6	5	4	3	2	1	0
bit Symbol					P43	P42	P41	P40
Read/Write					R/W			
Resetting Value					0	0	0	0

Port 4 Control Register

P4CR (FFC9H)

Prohibit Read Modify Write.

	7	6	5	4	3	2	1	0
bit Symbol					P43C	P42C	P41C	P40C
Read/Write					W			
Resetting Value					0	0	0	0
Function					0 : Out (Port)		1 : Address Output	

Set Output port/address bus

0	Output port
1	Address bus

Bank Register BX

BX (FFECH)

	7	6	5	4	3	2	1	0
bit Symbol					BX3	BX2	BX1	BX0
Read/Write					R/W			
Resetting Value					0	0	0	0

Address bus A19~A16

Bank Register BY

BY (FFEDH)

	7	6	5	4	3	2	1	0
bit Symbol					BY3	BY2	BY1	BY0
Read/Write					R/W			
Resetting Value					0	0	0	0

Address bus A19~A16

Note: It is necessary that the BX and BY registers of bits for which an address bus has not been specified always be left at "0".

230890

Figure 3.5 (9) Registers for Port 4

3.5.6 Port 5 (P50~P55)

Port 5 is a 6-bit input port P5 and also use as an analog input pin (AN0~AN5).
(The 7th bit of "FFCAH" is a test bit and must always be set to zero.)

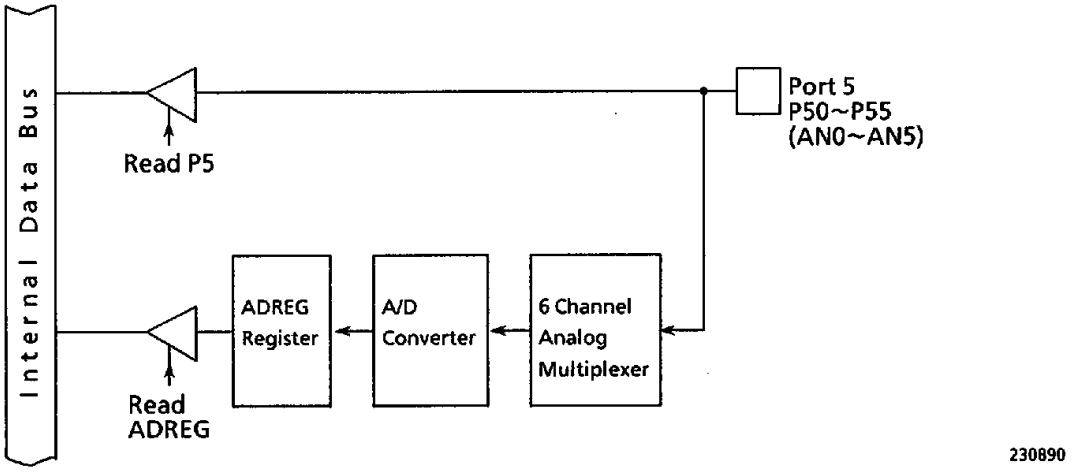


Figure 3.5 (10) Port 5

Port 5 Register

		7	6	5	4	3	2	1	0
P5 (FFCAH)	bit Symbol	"0" Fix		P55	P54	P53	P52	P51	P50
	Read/Write			R					
	Resetting Value	0		Input Only					
Writing data prohibited	Funcrion			Shared with analog input pin (AN0~AN5)					

230890

Figure 3.5 (11) Register for Port 5

3.5.7 Port 6 (P60~P63)

Port 6 is a 4-bit general-purpose I/O port P6 whose function is specified by the control register P67CR for each bit. The control register is initialized to "0" by resetting, and Port 6 enters in the input mode. This port is also serviceable as a stepping motor control port channel 0 (M00~M03), so either the general-purpose I/O or the stepping motor control port can be selected by the control register SMMOD. This port is served as the general-purpose I/O port by resetting.

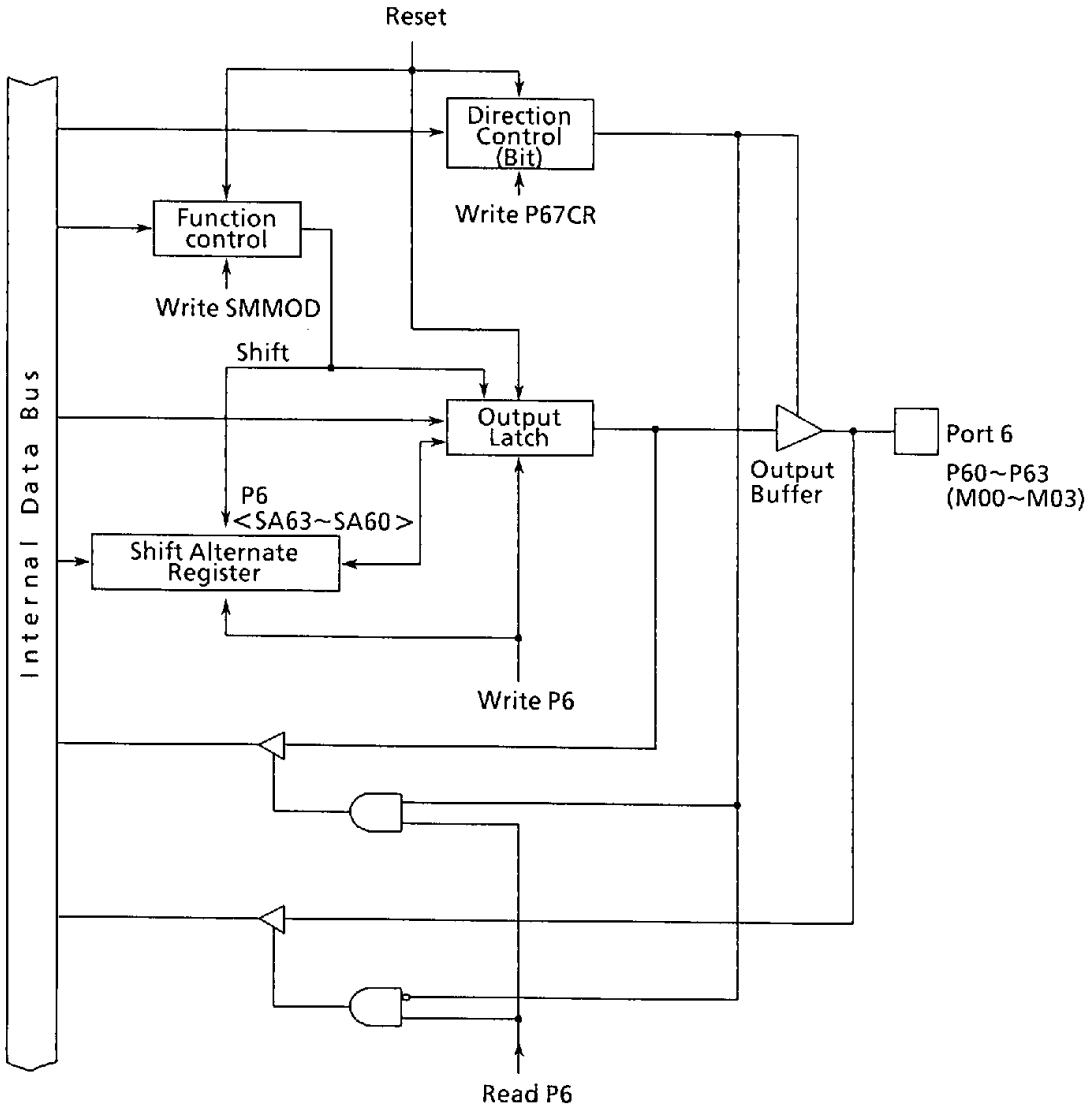


Figure 3.5 (12) Port 6

3.5.8 Port 7 (P70~P73)

Port 7 is a 4-bit general-purpose I/O port P7 whose I/O function is specified by the control register P67CR. The control register is initialized to "0" by resetting, whereby Port 7 turns to the input mode.

This port is also serviceable as a stepping motor control port channel 1 (M10~M13), so either the general-purpose I/O or the stepping motor control port can be selected by the control register SMMOD<P70C1, 0>. It is served as the general-purpose I/O port by resetting.

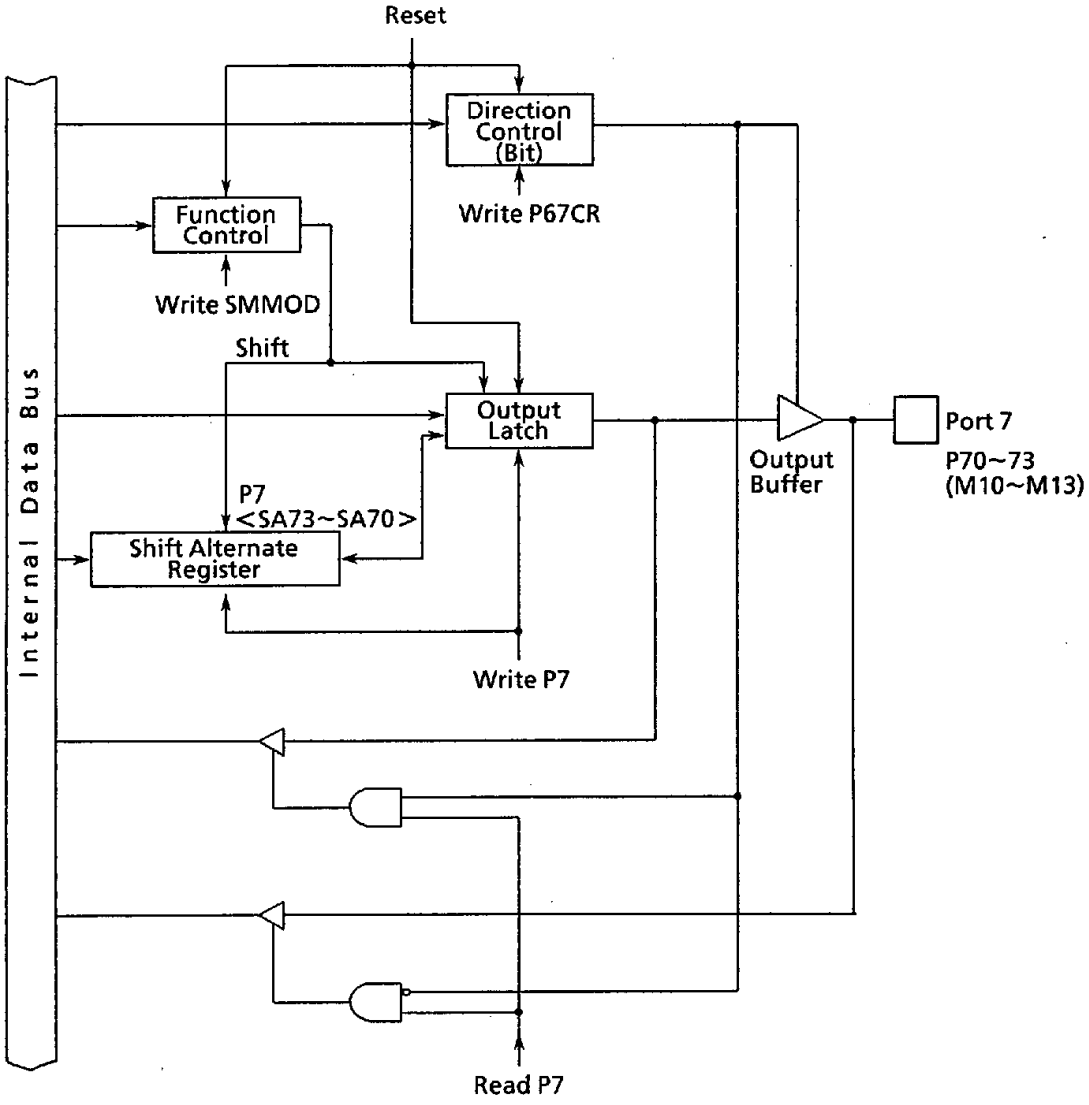


Figure 3.5 (13) Port 7

Port 6 Register

P6 (FFCCH)	7	6	5	4	3	2	1	0
bit Symbol	SA63	SA62	SA61	SA60	P63	P62	P61	P60
Read/Write	W				R / W			
Resetting Value	Undefined				Input Mode			
Function	Stepping motor control Port 0 Shifter alternate reg.				Shared with stepping morter control Port 0 (M0)			

See "3.7 Stepping moter control port" — Port 6

Port 7 Register

P7 (FFCDH)	7	6	5	4	3	2	1	0
bit Symbol	SA73	SA72	SA71	SA70	P73	P72	P71	P70
Read/Write	W				R / W			
Resetting Value	Undefined				Input Mode			
Function	Stepping motor control Port 1 Shifter alternate reg.				Shared with stepping morter control Port 1 (M1)			

See "3.7 Stepping moter control port" — Port 7

Port 6, 7 Control Register

P67CR (FFCEH)	7	6	5	4	3	2	1	0
bit Symbol	P73C	P72C	P71C	P70C	P63C	P62C	P61C	P60C
Read/Write	W				W			
Resetting Value	0	0	0	0	0	0	0	0
Function	0 : In 1 : Out				0 : In 1 : Out			

Prohibit Read Modify Write.

Select input/output of port 6

0	Input
1	Output

Select input/output of port 7

0	Input
1	Output

Note : When P6 and P7 are used as stepping motor control ports, read modify write is not available.

200990

Figure 3.5 (14a) Registers for Ports 6 and 7

Stepping motor mode register

SMMOD (FFCBH)	7	6	5	4	3	2	1	0
bit Symbol	–	SM7M0	P7OC1	P7OC0	–	SM6M0	P6OC1	P6OC0
Read/Write	R/W		R/W		R/W		R/W	
Resetting Value		0	0	0		0	0	0
Function		0: Full Step 1: Half Step	00 : IN/OUT 01 : IN/OUT, TO3 10 : IN/M1 (Timer 2, 3) 11 : ∗ (Timer 4)			0: Full Step 1: Half Step	00 : IN/OUT 01 : IN/OUT, TO1 1X : IN/M0 (Timer 0, 1)	

See "3.7 Stepping motor control port"

See "3.7 Stepping motor control port"

Select function of Port 6 (Port 6 Output Control)

P67CR / P6OC1, 0	P63		P62		P61		P60		
	P63C		P62C		P61C		P60C		SMC trigger signal
	0	1	0	1	0	1	0	1	
00	IN	OUT	IN	OUT	IN	OUT	IN	OUT	don't occurrence
01	IN	OUT	IN	OUT	IN	OUT	IN	TO1	
10	IN	M03	IN	M02	IN	M01	IN	M00	TFF1 (Timer 0 or Timer 1)
11									

IN : Input Port
OUT : Output Port
TO1 : Timer Output Port
M00-3 : Stepping Motor Control Port 0

Select function of port 7 (Port 7 Output Control)

P67CR &P8CR / P7OC1, 0	P73		P72		P71		P70		P83		
	P73C		P72C		P71C		P70C		P83OC		SMC trigger signal
	0	1	0	1	0	1	0	1	0	1	
00	IN	OUT	IN	OUT	IN	OUT	IN	OUT	OUT	TO4	don't occurrence
01	IN	OUT	IN	OUT	IN	OUT	IN	TO3			
10	IN	M13	IN	M12	IN	M11	IN	M10			TFF3 (Timer 2, 3)
11									OUT	TO3	TFF4 (Timer 4)

IN : Input Port
OUT : Output Port
TO3, TO4 : Timer Output Port
M10-3 : Stepping Motor Control Port 1

230890

Figure 3.5 (14b) Registers for Port 6 and 7

3.5.9 Port 8 (P80~P83)

Port 8 is a 4-bit general-purpose I/O port P8, with P80 to P82 intended only for the input and P83 only for the output. The output latch of P83 is reset to "0", and "0" is generated to the port.

Port 8 also has the functions of interrupt request input, clock input for a timer/event counter, and timer output.

(1) P80/INT0

P80 is a general-purpose input port, also used as the external interrupt request input pin INT0. INT0 allows the selection of either an "H" level interrupt or rising edge interrupt by using the control register P8CR <EDGE>.

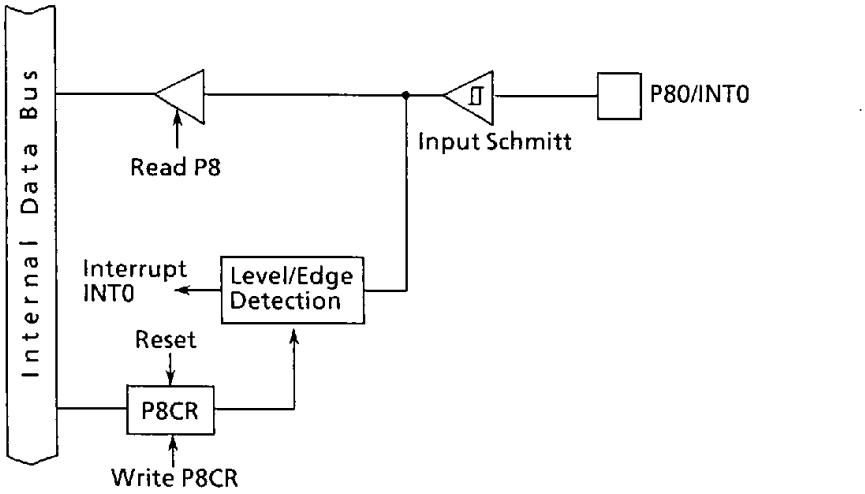


Figure 3.5 (15) Port P80/INT0

(2) P81/INT1/TI4

P81 is a general-purpose input port, also used as the external interrupt request input pin INT1 and the clock input pin TI4 for the timer/event counter.

This port incorporates a zero-cross detection circuit, and enables zero-cross detection by connecting an external capacitor. The zero-cross detection can be disabled/enabled by using the control register P8CR <ZCE1>. This control register is reset to "0", making the zero-cross detection disabled by resetting.

Rise edge/fall edge interrupts are controlled together with capture by the T4MOD <CAPM1, 0> of the 16-bit timer.

(3) P82/INT2/TI5

Like P81, P82 is a general-purpose input port, also used as the external interrupt request input pin INT2 and the clock input pin TI5 for the timer/event counter. This port also contains a zero-cross detection circuit, and disables/enables the zero-cross detection by using the control register P8CR <ZCE2>.

The zero-cross detection is disabled by resetting.

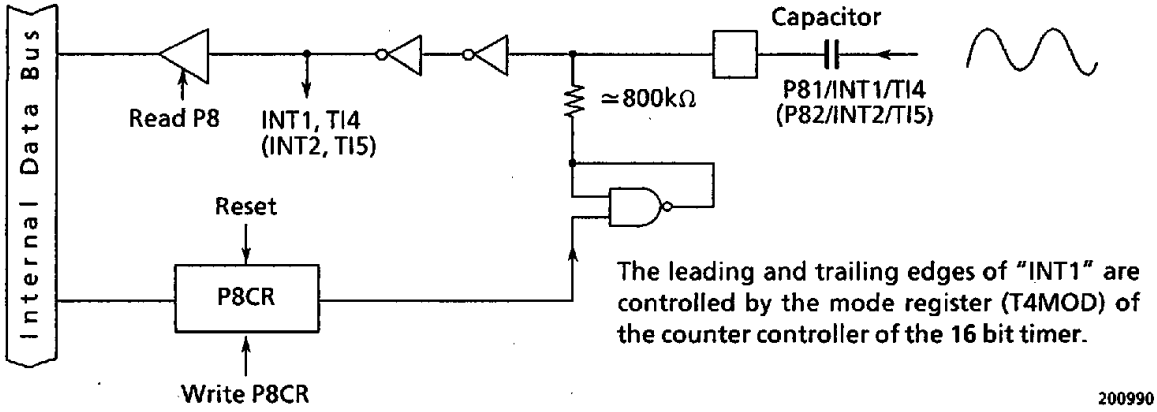


Figure 3.5 (16) Port P81/INT1/TI4 and P82/INT2/TI5

(4) P83/TO3/TO4

P83 is a general-purpose output port, also used as the timer output pins TO3/TO4. Either function can be selected by using two control registers P8CR<P83OC>, SMMOD <P7OC1, 0>.

P83OC	P7OC1, 0	P83 Function
0	X, X	Output port
1	0, X	TO4
1	1, 0	
1	1, 1	TO3

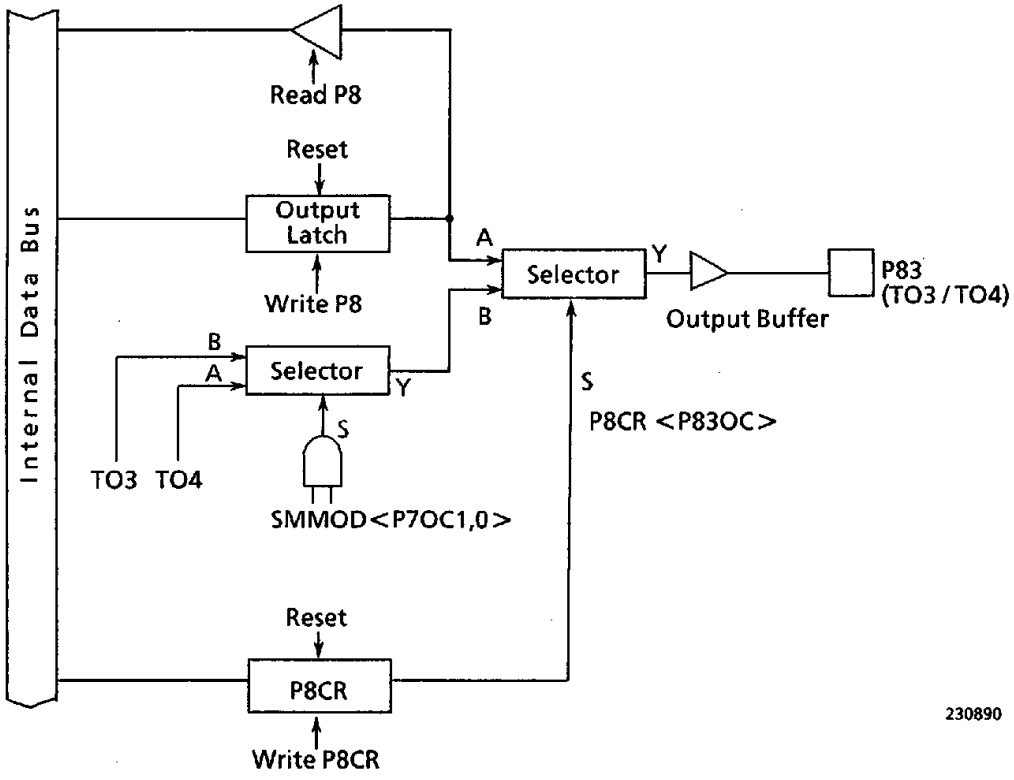


Figure 3.5 (17) Port P83/TO3/TO4

Port 8 Register

		7	6	5	4	3	2	1	0
P8 (FFD0H)	bit Symbol					P83	P82	P81	P80
	Read/Write					R/W	R	R	R
	Resetting Value					0	Input Mode		

Port 8 Control Register

		7	6	5	4	3	2	1	0
P8CR (FFD1H)	bit Symbol					P83OC	ZCE2	ZCE1	EDGE
	Read/Write					W	W	W	W
	Resetting Value					0	0	0	0
	Function					P83 Control 0: P83 1: TO3/TO4	INT2/TI5 Control 1: ZCD enable	(Note) INT1/TI4 Control 1: ZCD enable	INT0 Control 0: level 1: ↑ edge

Prohibit Read Modify Write.

Control INT0

0	Interrupt by "H" level detection
1	Interrupt by "H" rising edge detection

P81/INT1/TI4 zero-cross enable (Zero-Cross Enable P81)

0	Disable
1	Enable

P82/INT2/TI5 zero-cross enable (Zero-Cross Enable P82)

0	Disable
1	Enable

Select P83 function (P83 Output Control)

0	P83
1	TO3/TO4

Note : The INT1 rise edge or fall edge is selected by the T4MOD <CAPM1, 0> of the 16-bit timer. Refer to the 16-bit timer section.

230890

Figure 3.5 (18) Registers for Port 8

3.6 Timers

The TMP90C840A incorporates four 8-bit timers and a 16-bit multi-function timer /event counter.

The four 8-bit timers can operate independently, and also function as two 16-bit timers through cascade connection. 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)
  Possible arrangements: 8-bit×2 and 16-bit×1, or 16-bit×2
- 8-bit programmable square wave (pulse) generation (PPG: variable duty with variable cycle) mode (2 timers)
- 8-bit pulse width modulation (PWM: variable duty with constant cycle) mode (2 timers)

The 16-bit multi-function timer/event counter can operate in the following six modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square wave (pulse) generation (PPG: variable duty with variable cycle) mode
- Frequency measurement mode
- Pulse width measurement mode
- Timer deviation measurement mode

3.6.1 8-bit Timers

The TMP90C840A incorporates four 8-bit interval timers (Timers 0, 1, 2 and 3) , each of which can be operated independently. The cascade connection of Timer 0 and 1, or Timer 2 and 3 allows these timers used as 16-bit internal timers.

Figure 3.6 (1) is a block diagram of the 8-bit timers (Timer 0 and Timer 1) .

Timer 2 and 3 have the same circuit configuration as Timer 0 and Timer 1 respectively.

Each interval timer is composed of an 8-bit up-counter, an 8-bit comparator and an 8-bit timer register, with a Timer Flip-flop (TFF1/TFF3) provided to each pair of Timer 0/1 and Timer 2/3.

Internal clocks (ϕT1, ϕT16 and ϕT256), some of the input clock sources for the interval timers, are generated by the 9-bit prescaler shown in Figure 3.6 (2) .

Their operating modes of the 8-bit timers and flip-flops are controlled by four control registers (TCLK, TFFCR, TMOD and TRUN) .

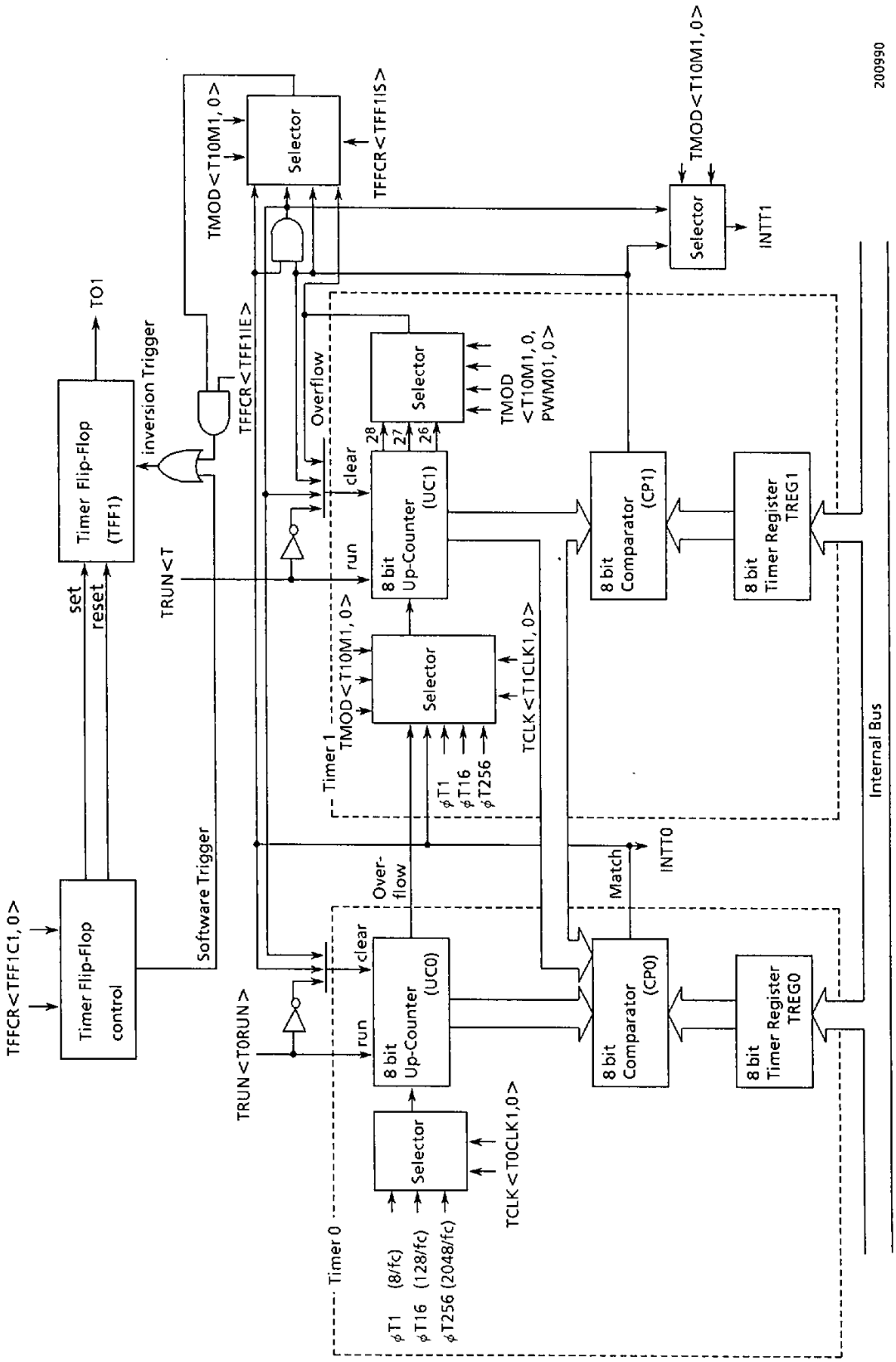


Figure 3.6 (1) Block Diagram of 8-bit Timers (Timers 0 and 1)

① Prescaler

A 9-bit prescaler is provided to further devide the clock frequency already divided to a 1/4 of the frequency of the source clock (fc).

It generates a input clock pulse for the 8-bit timers, 16-bit timer/event counter, the baud-rate generator, etc.

For the 8-bit timers, three types of clock are generated (ϕT1, ϕT16 and ϕT256).

The prescaler can be run or stopped by using the 5th bit TRUN<PRRUN> of the timer control register TRUN. Setting <PRRUN> to "1" makes the prescaler count, and setting it to "0" clears the prescaler to stop.

By resetting, <PRRUN> is initialized to "0", making the prescaler clear and stop.

Cycle

Input clock \ fc	8MHz	10MHz
ϕT1 (8/fc)	1.0μs	0.8μs
ϕT16 (128/fc)	16μs	12.8μs
ϕT256 (2048/fc)	256μs	204.8μs

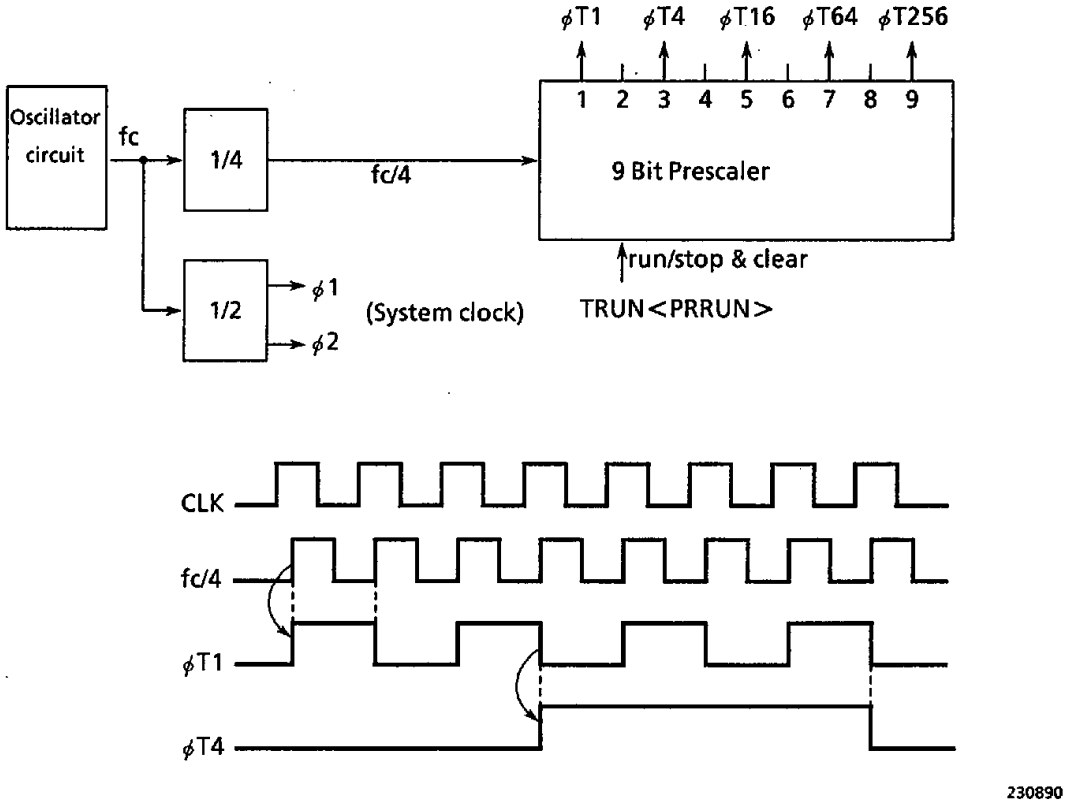


Figure 3.6 (2) Prescaler

② Up-counter

This is an 8-bit binary counter that counts up by an input clock pulse specified by an 8-bit timer clock control register TCLK and an 8-bit timer mode register TMOD.

The input clock pulse for Timer 0 and 2 is selected from ϕT1 (8/fc), ϕT16 (128/fc) and ϕT256 (2048/fc) according to the setting of the TCLK register.

Example : When setting TCLK<T0CLK1, 0>=0, 1, ϕT1 is selected as the input clock pulse for Timer 0.

The input clock pulse to Timer 1 and 3 is selected according to the operating mode. In the 16-bit timer mode, the overflow output of Timer 0 and 2 is automatically selected as the input clock pulse, regardless of the setting of the TCLK register.

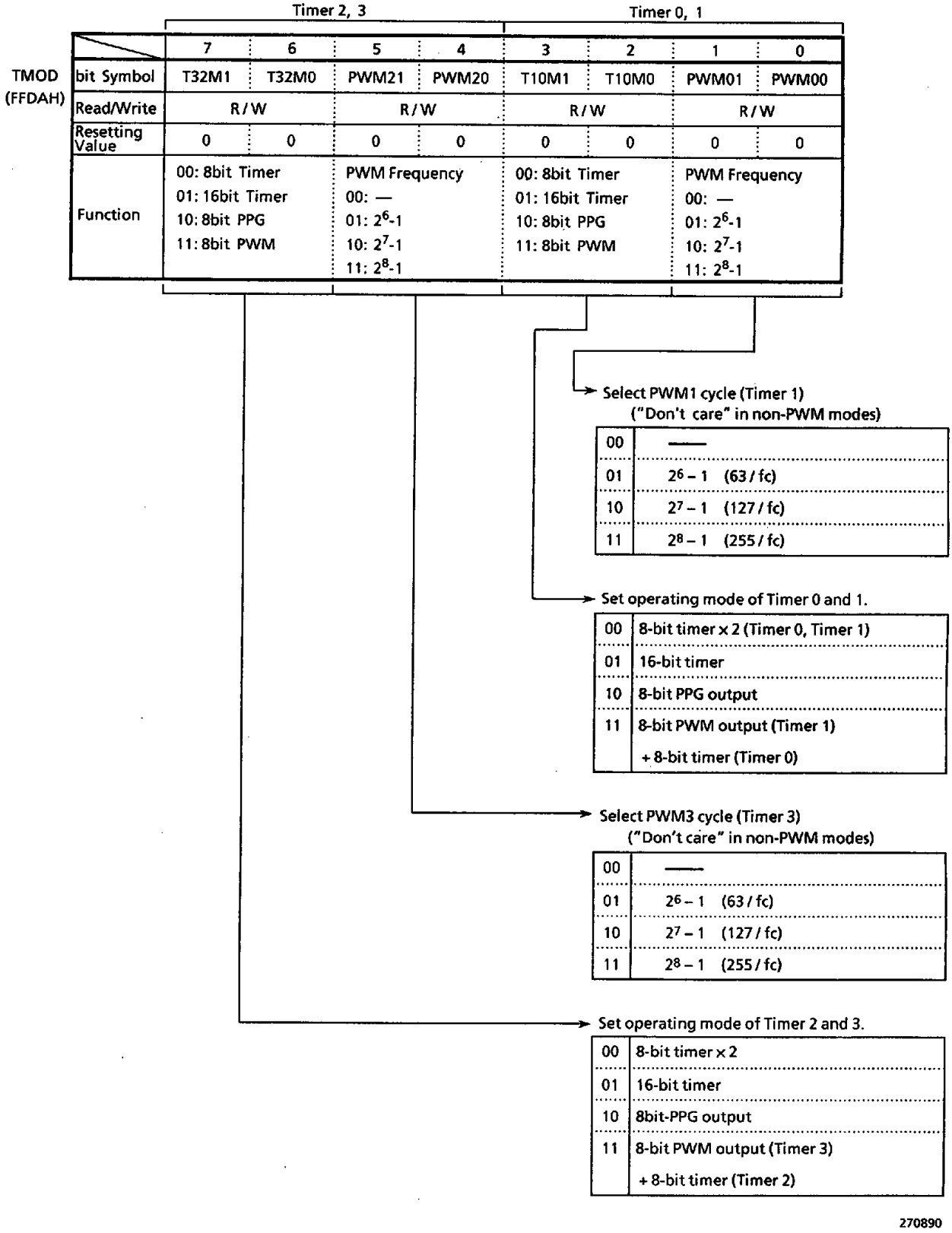
In the other operating modes, the clock pulse is selected among the internal clocks ϕT1, ϕT16 and ϕT256, and the output of the Timer 0 and 2 comparator (match signal).

Example : If TMOD<T10M1, 0>=0, 1, the overflow output of Timer 0 is selected as the input clock to Timer 1. (16 bit timer mode)
If TMOD<T10M1, 0>=0, 0 and TCLK<T1CLK1, 0>=0, 1, ϕT1 is selected as the input clock to Timer 1. (8 bit timer mode)

The operating mode is selected by the TMOD register. This register is initialized to TMOD<T10M1, 0>=0, 0 /TMOD<T32M1, 0>=0, 0 by resetting, whereby the up-counter is place in the 8-bit timer mode.

Functions, count, stop or clear of the up-counter can be controlled for each interval timer by the timer control register TRUN.

By resetting, all up-counters are cleared to stop the timers.



		Timer 2, 3				Timer 0, 1			
		7	6	5	4	3	2	1	0
TMOD (FFDAH)	bit Symbol	T32M1	T32M0	PWM21	PWM20	T10M1	T10M0	PWM01	PWM00
	Read/Write	R/W		R/W		R/W		R/W	
	Resetting Value	0	0	0	0	0	0	0	0
	Function	00: 8bit Timer 01: 16bit Timer 10: 8bit PPG 11: 8bit PWM		PWM Frequency 00: — 01: 2^6-1 10: 2^7-1 11: 2^8-1		00: 8bit Timer 01: 16bit Timer 10: 8bit PPG 11: 8bit PWM		PWM Frequency 00: — 01: 2^6-1 10: 2^7-1 11: 2^8-1	

Select PWM1 cycle (Timer 1)
("Don't care" in non-PWM modes)

00	——
01	$2^6 - 1$ (63 / fc)
10	$2^7 - 1$ (127 / fc)
11	$2^8 - 1$ (255 / fc)

Set operating mode of Timer 0 and 1.

00	8-bit timer x 2 (Timer 0, Timer 1)
01	16-bit timer
10	8-bit PPG output
11	8-bit PWM output (Timer 1) + 8-bit timer (Timer 0)

Select PWM3 cycle (Timer 3)
("Don't care" in non-PWM modes)

00	——
01	$2^6 - 1$ (63 / fc)
10	$2^7 - 1$ (127 / fc)
11	$2^8 - 1$ (255 / fc)

Set operating mode of Timer 2 and 3.

00	8-bit timer x 2
01	16-bit timer
10	8bit-PPG output
11	8-bit PWM output (Timer 3) + 8-bit timer (Timer 2)

270890

Figure 3.5 (3) 8-bit Timer Mode Register TMOD

TCLK (FFD8H)		Timer 3		Timer 2		Timer 1		Timer 0	
		7	6	5	4	3	2	1	0
	bit Symbol	T3CLK1	T3CLK0	T2CLK1	T2CLK0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
	Read/Write	R/W		R/W		R/W		R/W	
	Resetting Value	0	0	0	0	0	0	0	0
	Function	00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: — 01: ϕT1 10: ϕT16 11: ϕT256		00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: — 01: ϕT1 10: ϕT16 11: ϕT256	

Timer 0 input clock

00	—
01	Internal clock ϕT1
10	〃 ϕT16
11	〃 ϕT256

Timer 1 input clock

	TMOD<T10M1, 0> ≠ 0, 1	TMOD<T10M1, 0> = 0, 1
00	Timer 0 comparator output	Timer 0 overflow output (16 bit timer mode)
01	Internal clock ϕT1	
10	〃 ϕT16	
11	〃 ϕT256	

Timer 2 input clock

00	—
01	Internal clock ϕT1
10	〃 ϕT16
11	〃 ϕT256

Timer 3 input clock

	TMOD<T32M1, 0> ≠ 0, 1	TMOD<T32M1, 0> = 0, 1
00	Timer 2 comparator output	Timer 2 overflow output (16 bit timer mode)
01	Internal clock ϕT1	
10	〃 ϕT16	
11	〃 ϕT256	

270391

Figure 3.6 (4) 8-bit Timer Clock Control Register TCLK

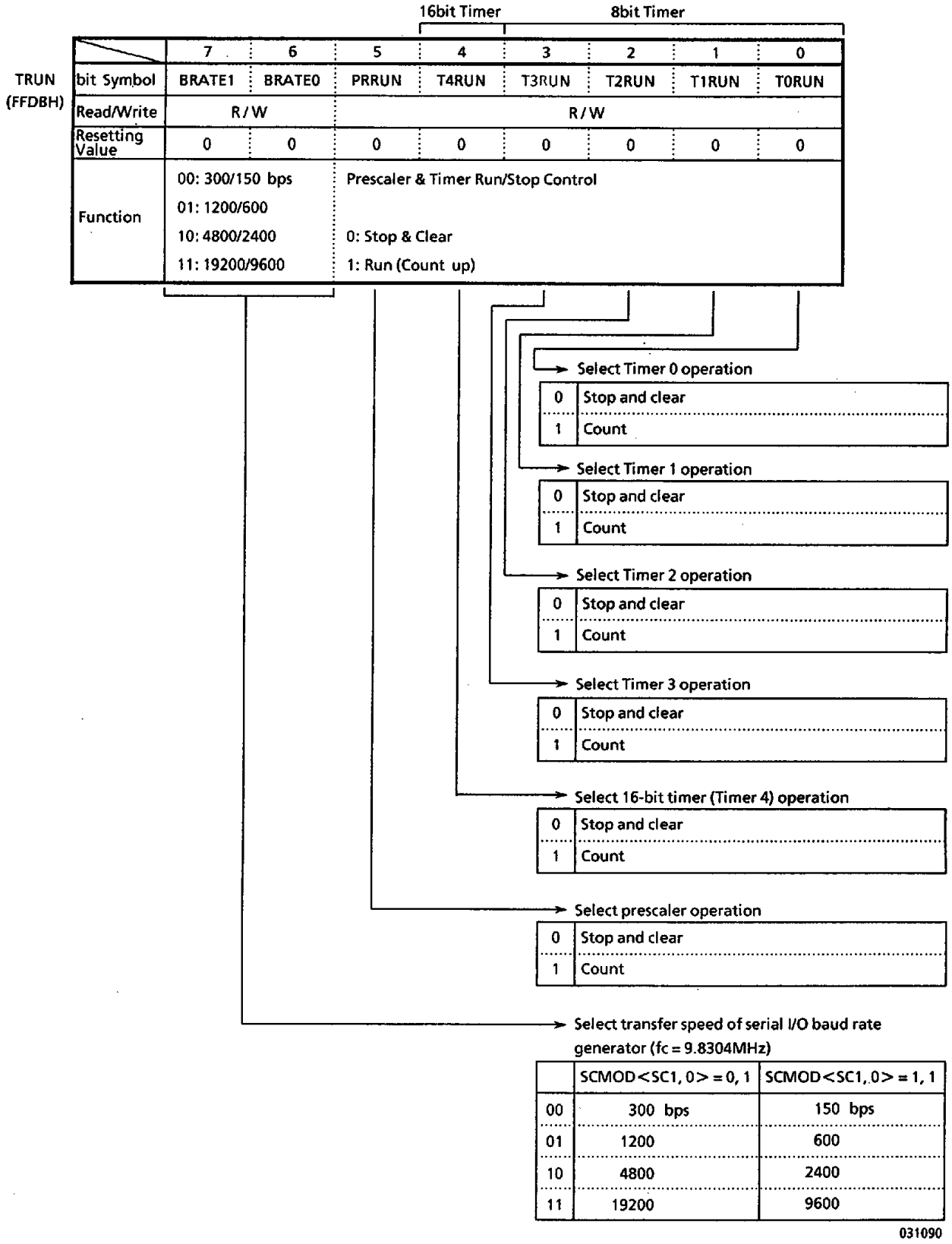


Figure 3.6 (5) Timer/Serial Channel Control Registers TRUN

TFF3 TFF1

TFFCR (FFD9H)	7	6	5	4	3	2	1	0
bit Symbol	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
Read/Write	W		R/W		W		R/W	
Resetting Value	—		0	0	—		0	0
Function	00: Clear TFF3 01: Set TFF3 10: Invert TFF3 11: Don't care		1: TFF3 Invert Enable	0: Invert by 8-bit timer 2 1: Invert by 8-bit timer 3	00: Clear TFF1 01: Set TFF1 10: Invert TFF1 11: Don't care		1: TFF1 Invert Enable	0: Invert by 8-bit timer 0 1: Invert by 8-bit timer 1

Select inverse signal of timer flip-flop TFF1

TMOD<T10M 1, 0> / TFFCR <TFF1IS>	0, 0	0, 1	1, 0	1, 1
0	8-bit timer mode (Timer 0)			
1	8-bit timer mode (Timer 1)	16-bit timer mode (Timers 0 & 1)	PPG mode (Timer 1)	PWM mode (Timer 1)

Invert timer flip-flop TFF1

0	Disable
1	Enable

Control timer flip-flop TFF1

00	Clear TFF1 to "0"
01	Set TFF1 to "1"
10	Invert value of TFF1 (Software inversion)
11	Don't care (Always set at "11" when read out)

Select inverse signal of timer flip-flop TFF3

TMOD<T32M 1, 0> / TFFCR <TFF3IS>	0, 0	0, 1	1, 0	1, 1
0	8-bit timer mode (Timer 2)			
1	8-bit timer mode (Timer 3)	16-bit timer mode (Timers 2 & 3)	PPG mode (Timer 3)	PWM mode (Timer 3)

Invert timer flip-flop TFF3

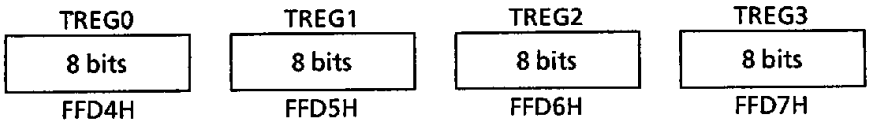
0	Disable
1	Enable

Control timer flip-flop TFF3

00	Clear TFF3 to "0"
01	Set TFF3 to "1"
10	Invert value of TFF3 (Software inversion)
11	Don't care (Always set at "11" when read out)

Figure 3.6 (6) 8-bit Timer Flip-Flop Control Register (TFFCR) 270890

③ Timer registers



Note : Write only.

8-bit registers are provided to set the interval time. When the set value of a timer register matches that of an up-counter, the match signal of their comparators turn to the active mode. If "00H" is set, this signal becomes active when the up-counter overflows. When a new value is written to this register, it is then immediately input to the comparator.

④ Comparators

A comparator compares the values in an up-counter and a timer register. When they matches, the up-counter is cleared to "0", and an interrupt signal (INTT0~5) is generated. If the timer flip-flop inversion is enabled by the Timer Flip-Flop control register, the Timer Flip-flop is inverted.

⑤ Timer flip-flops (Timer F/Fs)

The status of the Timer Flip-flop is inverted by the match signal (output by comparator). Its status can be output to the timer output pin TO1 (also used as P60) and TO3 (used as P70 or P80).

A Timer F/F is provided to each of the timer pairs Timer 0 - Timer 1 and Timer 2 - Timer 3, and is called TFF1 and TFF3, respectively. The status of TFF1 is output to T01, and that of TFF3 to TO3. TO3 has 2 pins (P70 and P83).

The Timer F/Fs are controlled by a timer flip-flop control register (TFFCR).

In the case of TFF1 (timer F/F for the Timer 0 and Timer 1), the flip-flop operation is described as follows (refer to Figure 3.6 (6)):

- TFFCR<FF1IS> is a timer selection bit for inversion of TFF1. In the 8-bit timer mode, inversion is enabled by the match signal from Timer 0 if this bit is set to "1", or by the signal from Timer 1 if set to "0".
  In any other mode, <FF1IS> must be always set to "1". It is initialized to "0" by resetting.
- TFFCR<FF1IE> controls the inversion of TFF1. Setting this bit to "1" enables the inversion and setting it to "0" disable.
  <FF1IE> is initialized to "0" by resetting.
- The bits TFFCR are used to set/reset TFF1 or enable its inversion by software. TFF1 is reset by writing "0, 0", set by "0, 1" and inverted by "1, 0".
  Similarly, TFF3 is controlled by TFFCR<TFF3C1, 0, TFF3IE, S>.

The 8-bit timers operate as follows:

(1) 8-bit Timer Mode

The four interval timers 0, 1, 2 and 3 can operate independently as an 8-bit interval timer. Only the operation of Timer 1 is described because their operations are the same.

① Generating interrupts at specified intervals

Periodic interrupts can be generated by using Timer 1 (INTT1) in the following procedure: Stop Timer 1, set the desired operating mode, input clock and cycle time in, the registers TMOD, TCLK and TREG1 enable INTT1, and start the counting of Timer 1.

Example: To generate Timer 1 interrupt every 4.0μs at fc=10 MHz, the registers should be set as follows:

	MSB 7 6 5 4 3 2 1 0 LSB	
TRUN ←	- - - - - - - 0 -	Stop Timer 1, and clear it to "0".
TMOD ←	- - - - 0 0 X X	Set the 8-bit timer mode.
TCLK ←	- - - - 0 1 - -	Select ϕT1 (0.8μs @fc=10 MHz) as the input clock.
TREG1 ←	0 0 1 1 0 0 1 0	Set the timer register at 40μs/ϕT1=32H.
INTEH ←	- - - - - - - 1	Enable INTT1.
TRUN ←	- - 1 - - - 1 -	Start Timer 1.

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

Use the following table for selecting the input clock:

Table 3.6 (1) 8-bit timer interrupt cycle and input clock

Interrupt cycle @fc = 10 MHz	Resolution	Input clock
0.8 μs ~ 204.8 μs	0.8 μs	ϕT1 (8/fc)
12.8 μs ~ 3.2768 ms	12.8 μs	ϕT16 (128/fc)
204.8 μs ~ 52.4288 ms	204.8 μs	ϕT256 (2048/fc)

270890

Caution for using Timer 2

Interrupts generated by Timer 2 (INTT2) uses the same interrupt mask flag INTEL<IET2> as used for those of the A/D converter (INTAD). To select either interrupt, another flag INTEH<ADIS> is provided. Setting this flag to "0" enables interrupts by Timer 2 and disables those by the A/D converter.

INTEH<ADIS>	Interrupt Source
0	Timer 2
1	A/D Convertor

270890

② Generating pulse at 50% duty

The Timer Flip-flop is inverted at specified intervals, and its status is output to a timer output pin TO1.

Example: To output pulse from TO1 at fc=10 MHz every 4.8μs, the registers should be set as follows:
This example uses Timer 1, but the same operation can be effected by using Timer 0.

```
                    MSB          LSB
                    7 6 5 4 3 2 1 0
TRUN  ←   - - - - - - - 0 -    Stop Timer 1, and clear it to "0".
TMOD  ←   - - - - 0 0 X X      Set the 8-bit timer mode.
TCLK  ←   - - - - 0 1 - -      Select φT1 as the input clock.
TREG1 ←   0 0 0 0 0 0 1 1      Set the timer register at 4.8μs/φT1/2=3.
TFFCR ←   - - - - 0 0 1 1      Clear TFF1 to "0", and set to invert by the match
                               signal from Timer 1.
SMMOD ←   - - - - X X 0 1  }   Select P60 as T01 pin.
P67CR ←   - - - - - - - 1  }
TRUN  ←   - - 1 - - - 1 -      Start Timer 1.
```

(Note) ×: Don't care -: No change

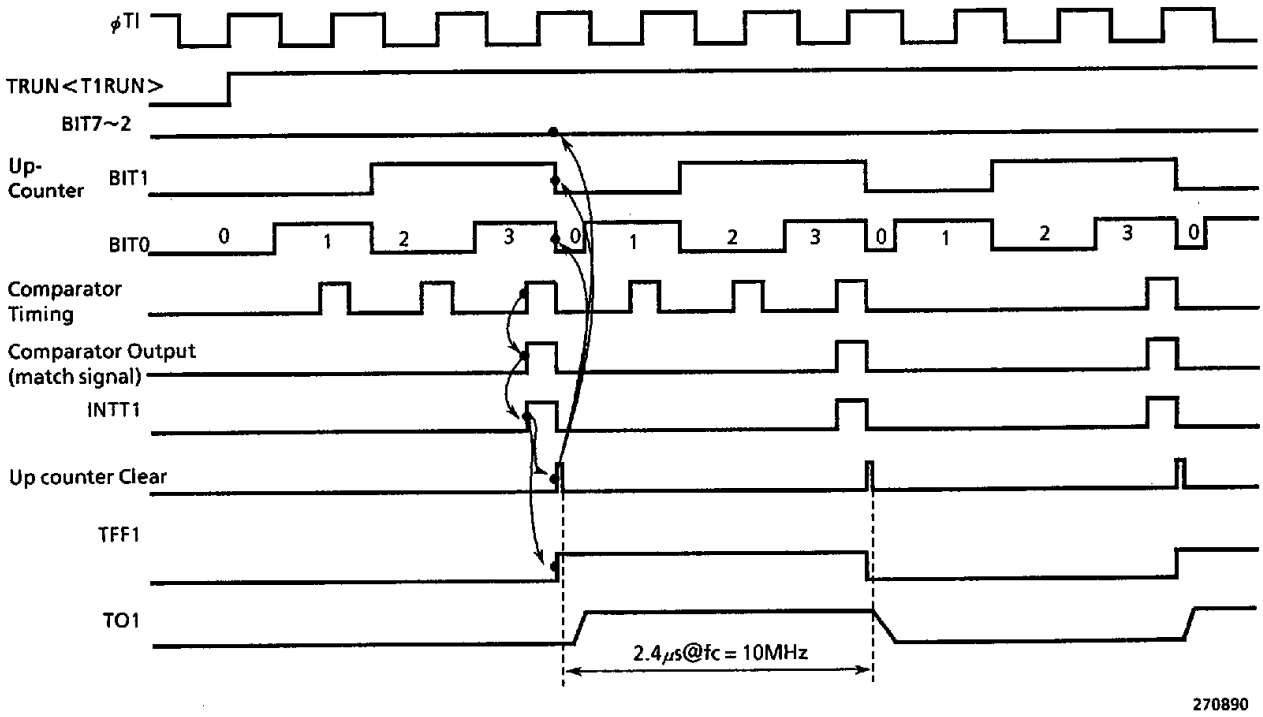


Figure 3.6 (7) Pulse Output (50% duty) Timing Chart

③ Making Timer 1 count up by match signal from Timer 0 comparator.

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

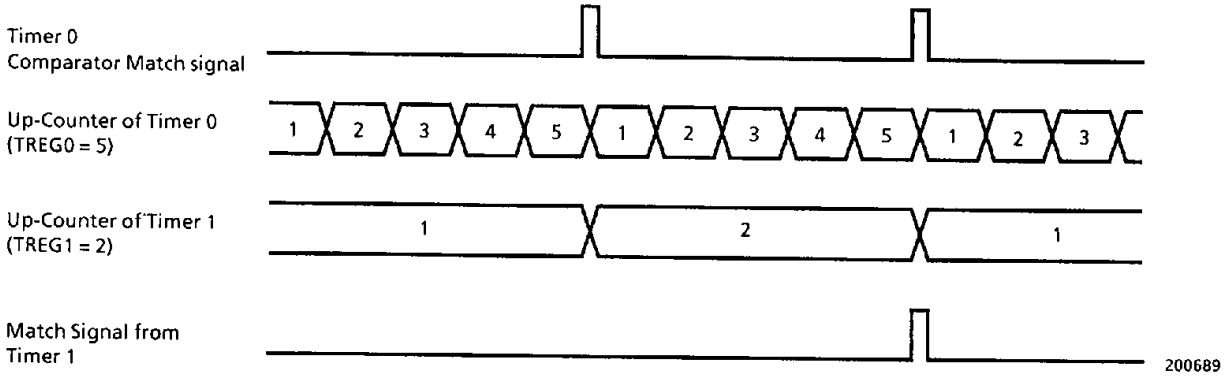


Figure 3.6 (8)

④ Software inversion

The timer flip-flops can be inverted by software independent of the timer operation.

Writing "1, 0" into the bits TFFCR<TFF1C1, 0> inverts TFF1, and writing the same into TFFCR<TFF3C1, 0> inverts TFF3.

⑤ Initial setting of Timer Flip-Flops

The Timer Flip-flops can be initialized to either "0" or "1" without regard to the timer operation.

TFF1 is initialized to "0" by writing "0, 0" into TFFCR<TFF1C1, 0>, and "1" by writing "01" into these bits.

Note : Reading the data from the Timer Flip-flops and timer registers is prohibited.

(2) 16-bit Timer Mode

The Timer 0 and Timer 1 or Timer 2 and Timer 3 can be used as one 16-bit interval timer.

As both timer pairs have the same function, only the timer pair Timer 0 and Timer 1 is discussed.

Cascade connection of Timer 0 and Timer 1 to use them as a 16-bit interval timer requires to set the <T10M1, 0> of the mode register TMOD to "0, 1".

By selecting the 16-bit timer mode, the overflow output of Timer 0 is automatically selected as the input clock to Timer 1, regardless of the set value of the clock control register TCLK. The input clock to Timer 0 is selected by TCLK. Table 3.6 (2) shows the combinations of timer (interrupt) cycle and input clock.

Table 3.6 (2) 16-bit Timer (Interrupt) Cycle and Input Clock

Timer (interrupt) cycle @fc = 10MHz	Resolution	Input clock to Timer 0
0.8μs~52.43ms	0.8μs	φT1 (8/fc)
12.8μs~838.86ms	12.8μs	φT16 (128/fc)
204.8μs~13.42s	204.8μs	φT256 (2048/fc)

270890

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

Example: To generate interrupts INTT1 at fc=8MHz every 1 second, the timer registers TREG0 and TREG1 should be set as follows:

As φT16 (=16μs @8MHz) is selected as the input clock,

1 s/16μs = 62500 = F424H

Therefore,

TREG1 = F4H

TREG0 = 24H

The match signal is generated by Timer 0 comparator each time the up-counter UC0 matches TREG0. In this case, the up-counter UC0 is not cleared, but the interrupt INTT0 is generated.

Timer 1 comparator also generates the match signal each time the up-counter UC1 match TREG1. When the match signal is generated simultaneously from comparators of Timer 0 and Timer 1, the up-counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If the Timer Flip-flop inversion is enabled by the Timer Flip-flop control register, the timer flip-flop TFF1 is inverted at the same time.

	Timer 0			Timer 1		
	INTT0	TO1	match	INTT1	TO1	match
16 bit Timer Mode (Count-up Timer 1 by overflow of Timer 0)	Interrupt is generated.	Can't output (Can't output the matching with TREG0.)	TREG0 (Continue counting when match)	Interrupt is generated.	Can output (Can output the matching with both TREG0 and TREG1)	TREG1*2^8 + TREG0(16 bit) (Cleared by matching with both registers.)
8 bit Timer Mode (Count-up Timer 1 by matching of Timer 0)	Interrupt is generated.	Can output (Timer 0 or Timer 1)	TREG0 (Clear when match)	Interrupt is generated.	Can output (Timer 0 or Timer 1)	TREG1*TREG0 (Multiplied Valve) (Cleared by matching)

031090

Example: Given TREG1 = 04H and TREG0 = 80H,

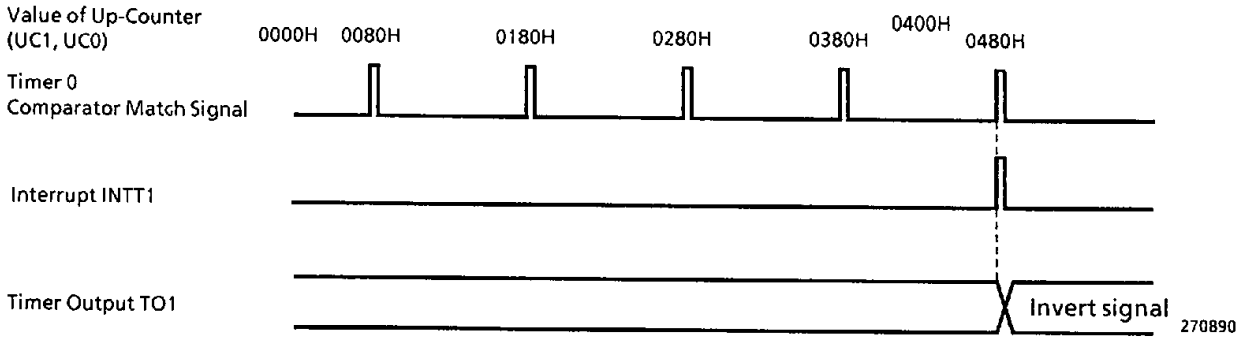


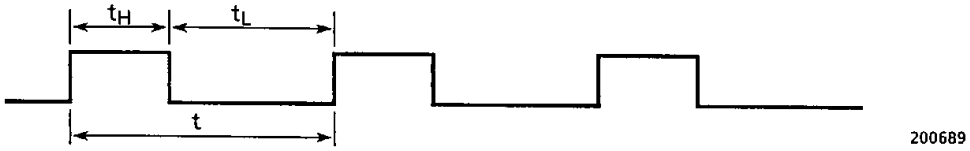
Figure 3.6 (9)

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

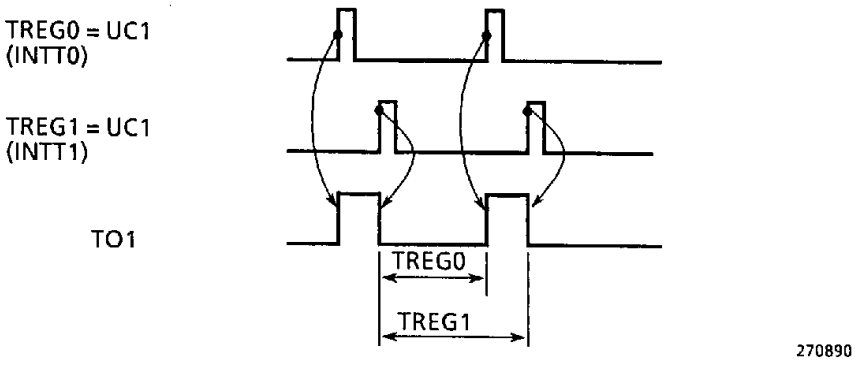
Pulse can be generated at any frequency and duty rate by Timer 1 or Timer 3. The output pulse may be either low-or high-active.

In this mode, Timers 0 and Timer 2 can not be used.

If Timer 1 is used, pulse is output to TO1 (shared with P60), and the use of Timer 3 results in the output to TO3 (shared with P70/P83).



Following is the timing of Timer 1 (The operation is the same as when Timer 3 is selected):



In the 8-bit PPG mode, programmable pulse is generated by the inversion of the timer output each time the 8-bit up-counter 1 (UC1) matches the timer register TREG0 or TREG1.

Note that the set value of TREG0 must be smaller than that of TREG1.

In this mode, the up-counter UC0 of Timer 0 can not be used (Set TRUN<T0RUN> = 1, and count the Timer 0).

The PPG mode can be illustrated as follows:

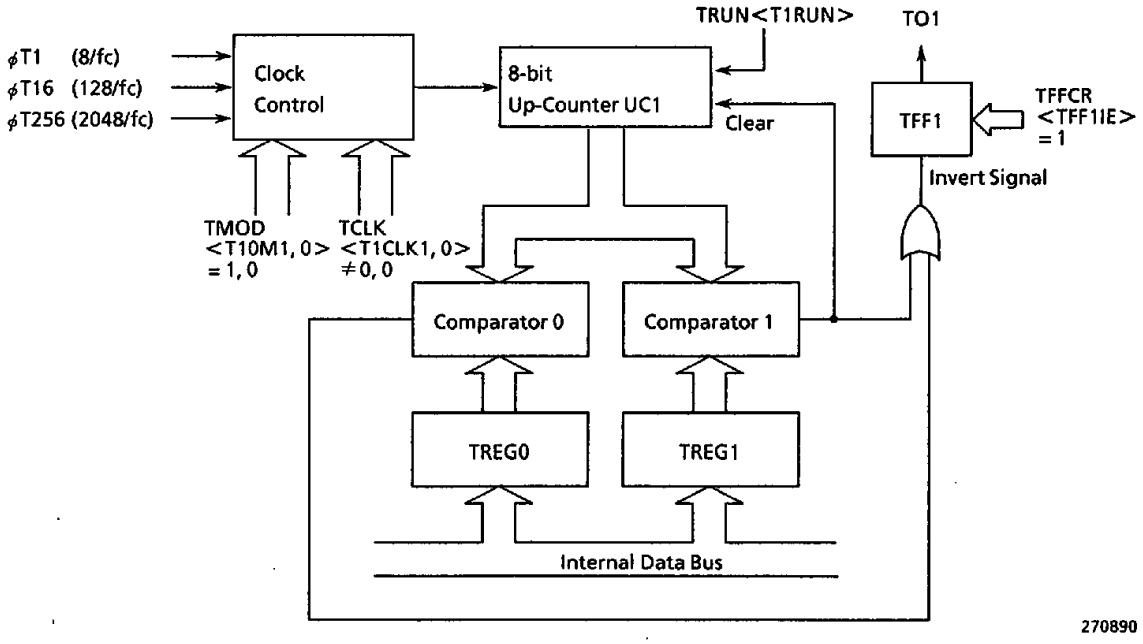


Figure 3.6 (10) Block Diagram of 8-bit PPG Mode

Example: Generate pulse at 50kHz and 1/4 duty rate (@fc = 8MHz)

- Calculate the set values of the timer registers.

  To obtain the frequency of 50kHz, the pulse cycle t should be: 1/50kHz = 20μs.
  Given ϕT1 = 1μs (@8MHz),

  20μs/1μs = 20

  Consequently, the timer register 1 (TREG1) should be set to 20 = 14H.
  Given a 1/4 duty, t × 1/4 = 20 × 1/4 = 5 μs

  5 μs/1 μs = 5

  As a result, the timer register 0 (TREG0) should be set to 5 = 05H.

```
TRUN  ← - - - - - - 0 0   Stop Timer 0 and Timer 1, and clear them to "0".
TCLK  ← - - - - 0 1 x x   Select φT1 as the input clock.
TMOD  ← - - - - 1 0 x x   Set 8-bit PPG mode.
TFFCR ← - - - - 0 1 1 1   Set the output "H", and enable the inversion by Timer 1.
                          Writing "00" provides negative logic pulse
TREG0 ← 0 0 0 0 0 1 0 1   Write "5H".
TREG1 ← 0 0 0 1 0 1 0 0   Write "14H".
SMMOD ← - - - - x x 0 1
P67CR ← - - - - - - - 1   } Select P60 as the T01 pin.
TRUN  ← - - 1 - - - 1 1   Start Timer 1.
```

(Note) ×: Don't care -: No change

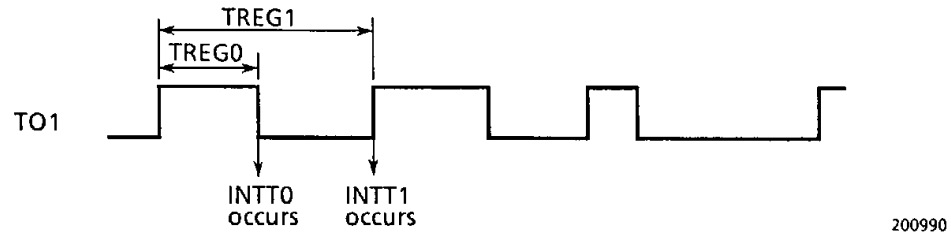
Precautions for PPG Output

By rewriting the content of the TREG (timer register), it is possible to make TMP90C840A output PPG. However, be careful, since the timing to rewrite TREG differs depending on the pulse width of PPG to be set. This problem is explained below by an example.

Example : To output PPG through 8 bit timers 0 and 1

TREG0 : Pulse width

TREG1 : Cycle



The pulse width is normally changed by the interrupt (INTT1) process routine in each cycle. However, when the pulse width to be set (the value to be written in TREG0) is small, trouble may occur, in that the timer counter exceeds the value of TREG0 before the interrupt process routine is set. Therefore, it is recommended to make the following decisions in INTT0 and INTT1 interrupt processes.

INTT0 process routine : The value of TREG0 is rewritten only when the value to be written in TREG0 is smaller than the current value of TREG0.

INTT1 process routine : On the contrary to INTT0, TREG0 is rewritten only when the value to be written in TREG0 is larger than the current value of TREG0.

TMP90C840A cannot read the content of TREG, so it is necessary to buffer the content of TREG in a RAM (or the like) for making the above judgment.

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

This mode is only available for Timer 1 and Timer 3, and generates two 8-bit resolution PWM (PWM1 and PWM3).

Pulse width modulation is output to T01 pin (shared with P60) when using Timer 1, and to TO3 pin (shared with P70 or P83) when using Timer 3.

Timer 0 and Timer 2 can be also used as 8-bit timers.

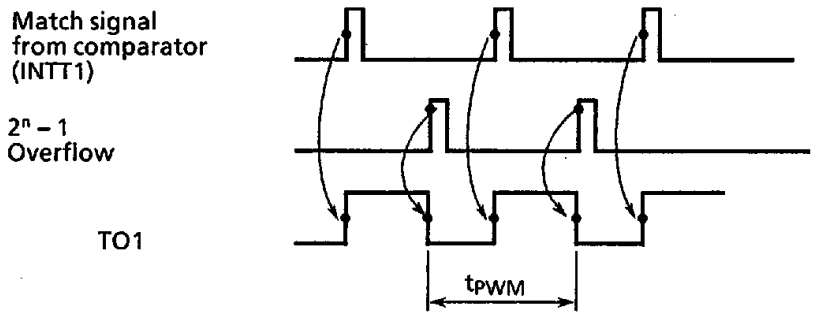
Following is the timing of Timer 1 (PWM1) (The operation is the same as when Timer 3 is selected):

The inversion of the timer output occurs when the up-counter (UC1) matches the set value of the timer register TREG1, as well as when an overflow of 2^n-1 (n=6, 7 or 8 selected by TMOD<PWM01, 00> occurred at the counter. The up-counter UC1 is cleared by the occurrence of an overflow of 2^n-1. For example, 6 bit PWM is selected when n=6.

The following condition must be obtained in this PWM mode:

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

(Set value of timer register) ≠ 0



270890

The PWM mode can be illustrated as follows:

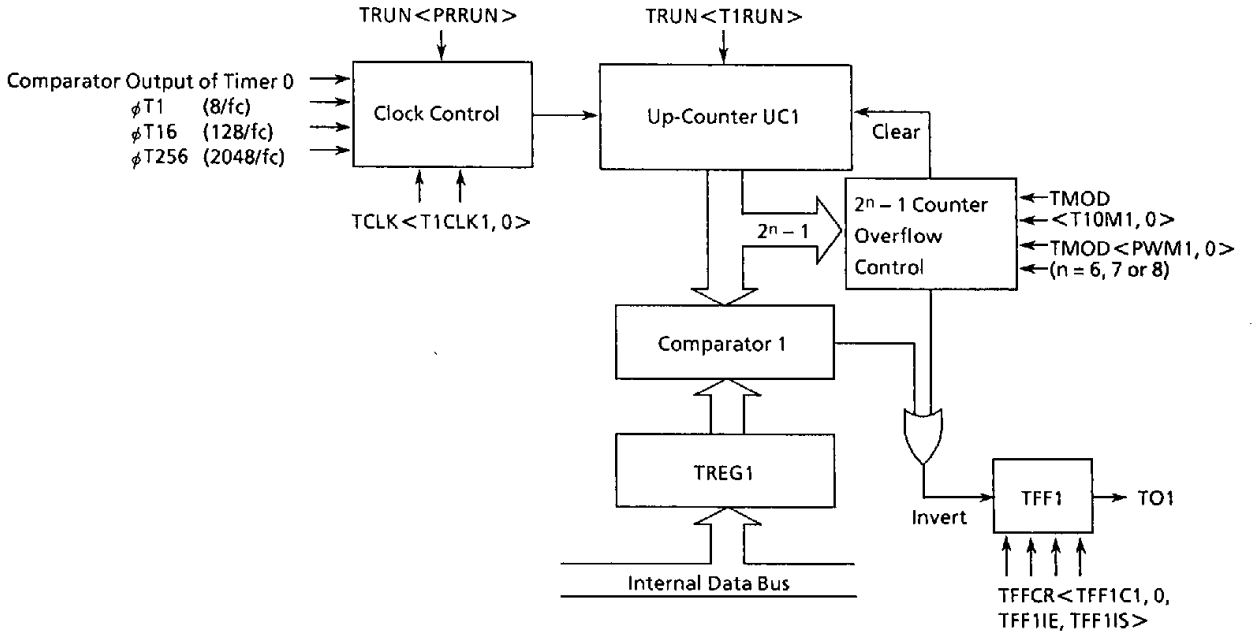
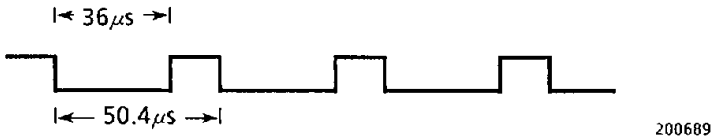


Figure 3.6 (11) Block Diagram of 8-bit PWM Mode

Example: Generate the following PWM to the TO3 pin (P83) at fc = 10MHz.



Assuming the PWM cycle is 50.4 μs when φT1 = 0.8 μs and @fc = 10MHz,

$50.4\mu s/0.8\mu s = 63 = 2^6 - 1$

Consequently, n should be set at 6 (TMOD1, 0 = 01).

Given the "H" level period of 36μs, setting φT1 = 0.8μs results:

$36\mu s/0.8\mu s = 45 = 2DH$

As a result, TREG3 should be set at 2DH.

TRUN ← - - - - 0 - - -	Stop Timer 3.	
TCLK ← 0 1 - - - - - -	Select φT1 as the input clock.	
TMOD ← 1 1 0 1 - - - -	Set the 2^6 – 1cycle in the PWM mode.	
TFFCR ← 0 0 1 1 - - - -	Set the initial output to 0 ("L" level).	
TREG3 ← 0 0 1 0 1 1 0 1	Write "2DH".	
SMMOD ← x x 1 1 - - - -	} Select P83 as the TO3 pin.	
P8CR ← - - - - 1 - - -		
TRUN ← - - - - 1 - - -	Start Timer 3.	

(Note) ×: Don't care -: No change

Table 3.6 (3) PWM Cycle and Selection of $2^n - 1$ counter

	Expression	PWM cycle (@fc = 10MHz)		
		ϕT1 (8/fc)	ϕT16 (128/fc)	ϕT256 (2048/fc)
26-1	(26-1) × ϕTn	50.4μs	806.4μs	12.9ms
27-1	(27-1) × ϕTn	101.6μs	1625.6μs	26.0ms
28-1	(28-1) × ϕTn	204.0μs	3264.0μs	52.2ms

270890

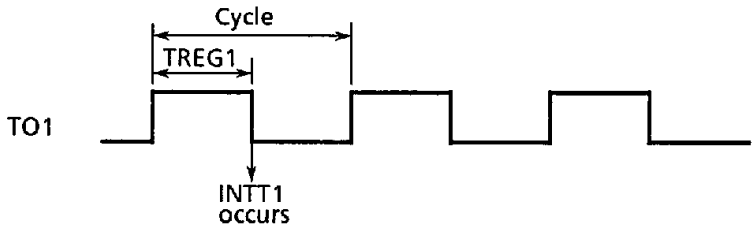
Precautions for PWM output

TMP90C840A can output PWM by the 8 bit timer. However, changing the pulse width of PWM requires special care. This problem is explained by the following example.

Example : To output PWM by 8 bit timer

TREG1 : Pulse width

Cycle : Fixed ($2^6-1, 2^7-1, 2^8-1$)



200990

In the PWM mode, INTT1 occurs at the coincidence with TREG1. However, the pulse width cannot be changed directly using this interrupt. (Depending on the value of TREG1 to be set, coincidence with TREG1 may be detected again in a single cycle, inverting the timer output.)

To eliminate this problem in changing the pulse width, it is effective to halt the timer with the INTT1 process, modify the value of TREG1, set the timer output to "1", and restart the timer. In the meantime, the output waveform loses shape when the pulse width is changed. This method is valid for a system that allows a deformed output waveform.

(5) A Table of All Timer Mode

Table 3.6 (4) Timer Mode

Register name	TMOD		TCLK		TFFCR
bit Symbol	T10M (T32M)	PWM1 (PWM3)	T1CLK (T3CLK)	T0CLK (T2CLK)	FF1IS (FF3IS)
Function	Timer mode	PWM cycle	Upper Input	Lower Input	Inversion select
16 bit Timer mode	01	—	—	ϕT1, 16, 256 (01, 10, 11)	1 (*)
8 bit Timer × 2ch	00	—	Comparator output from Lower timer, ϕT1, 16, 256 (00, 01, 10, 11)	ϕT1, 16, 256 (01, 10, 11)	0: Lower timer 1: Upper timer
8 bit PPG × 1ch	10	—	ϕT1, 16, 256 (01, 10, 11)	—	1
8 bit PWM × 1ch	11	$2^6-1, 2^7-1, 2^8-1$ (01, 10, 11)	ϕT1, 16, 256 (01, 10, 11)	—	1
8 bit Timer × 1ch		—	—	ϕT1, 16, 256 (01, 10, 11)	Impossible to output

031090

(Note) — : don't care
* : It is possible to set to "0", when timer F/F is not used.

3.6.2 Multi-Function 16-bit Timer/Event Counter (Timer 4)

The TMP90C840A incorporates a multi-function 16-bit timer/event counter that functions in the following operating modes:

- 16-bit timer
- 16-bit event counter
- 16-bit PPG mode
- Frequency measurement
- Pulse width measurement
- Time difference measurement

Figure 3.6 (12) is a block diagram of the 16-bit timer/event counter.

The timer/event counter is composed of a 16-bit up-counter, two 16-bit timer registers, two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its control circuit.

The timer/event counter is controlled by four control registers (T4MOD, T4FFCR,, TRUN, SMMOD and P8CR). The TRUN register also controls the 8-bit timers. The P8CR register is the control register for the port P8. The SMMOD is the mode register for stepping motor control port.

- T4MOD : Timer 4 clock source selection, Capture control and Counter control
- T4FFCR : Timer 4 Flip-Flop control
- TRUN : Prescaler and Timer Run/Stop control
- P8CR, SMMOD } Timer 4 output (TO4) control

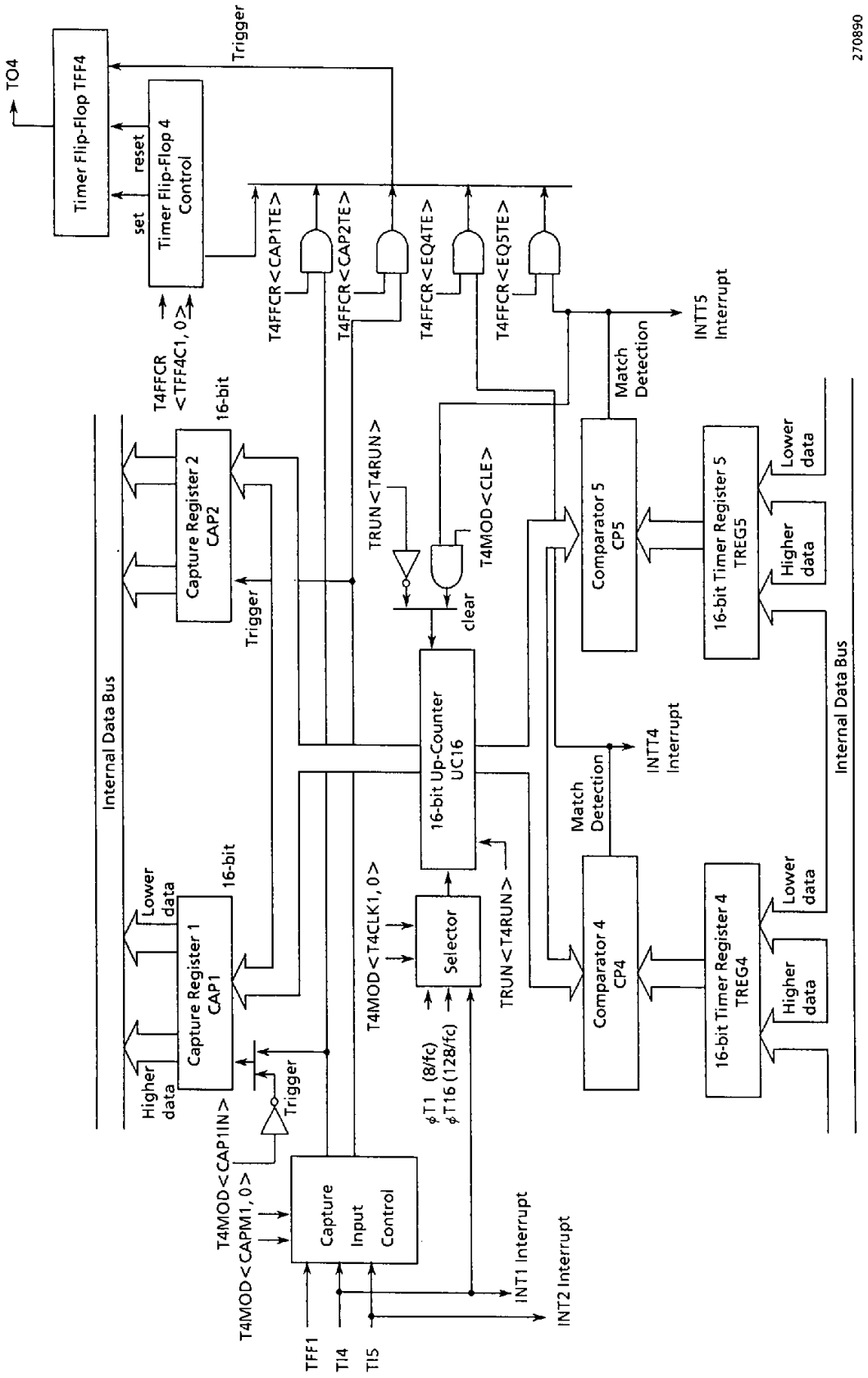


Figure 3.6 (12) Block Diagram of 16-Bit Timer/Event Counter (Timer 4)

T4MOD (FFE4H)	7	6	5	4	3	2	1	0
bit Symbol			CAP1IN	CAPM1	CAPM0	CLE	T4CLK1	T4CLK0
Read/Write			W	R/W		R/W	R/W	
Resetting Value			1	0	0	0	0	0
Function			0: Soft-Capture 1: don't care	Capture Timing 00: Disable INT1 01: TI4↑ TI5↑ INT1 10: TI4↑ TI4↓ INT1 11: TFF1↑ TFF1↓ INT1		1: UC16 Clear Enable	Timer 4 Source Clock 00: TI4 01: φT1 (8/fc) 10: φT16 (128/fc) 11: —	

Timer 4 input clocks

00	External clock (TI4)
01	Internal clock φT1
10	Internal clock φT16
11	Not used

Clear up-counter UC16

0	Clear disable
1	Clear if equal to TREG5

Control capture/interrupt INT1

	Capture control	IINT1 control
00	Capture disable	Interrupt occurs at rising edge of TI4 (INT1) input
01	CAP1 at TI4 rise CAP2 at TI5 rise	
10	CAP1 at TI4 fall CAP2 at TI5 fall	Interrupt occurs at falling edge of TI4 (INT1) input
11	CAP1 at TFF1 rise CAP2 at TFF1 fall	Interrupt occurs at falling edge of TI4 (INT1) input

Trigger software capture

0	Capture up-counter value into CAP1 (Software capture)
1	Always set at "1" when read out

200990

Figure 3.6 (13) 16-Bit Timer/Event Counter (Timer 4) Control Mode Registers

T4FFCR (FFE5H)

	7	6	5	4	3	2	1	0
bit Symbol			CAP2TE	CAP1TE	EQ5TE	EQ4TE	TFF4C1	TFF4C0
Read/Write			R/W				W	
Resetting Value			0	0	0	0		
Function			TFF4 反転トリガ 0: Disable trigger 1: Enable trigger				00: Clear TFF4 to "0" 01: Set TFF4 to "1" 10: Invert TFF4 11: Don't care ※ Always set at "1, 1" when read out	
			Invert when capturing UC value into CAP2	Invert when capturing UC value into CAP1	Invert when matching UC with TREG5	Invert when matching UC with TREG4		

Control of Timer Flip-Flop 4 (TFF4)

00	Clear TFF4 to "0"
01	Set TFF4 to "1"
10	Invert TFF4 value (software inversion)
11	Don't care (Always set at "11" when read out)

Trigger inversion of Timer Flip-Flop 4 (TFF4)

0	Disable trigger
1	Enable trigger

CAP2TE ; When capturing up-counter value into CAP2
CAP1TE ; When capturing up-counter value into CAP1
EQ5TE ; When equalizing up-counter with TREG5
EQ4TE ; When equalizing up-counter with TREG4

270890

Figure 3.6 (14) 16-bit Timer/Event Counter Timer Flip-Flop 4 Control Registers

TRUN (FFDBH)

	7	6	5	4	3	2	1	0
bit Symbol	BRATE1	BRATE0	PRRUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
Resetting Value	0	0	0	0	0	0	0	0
Function	00: 300/150 bps 01: 1200/600 10: 4800/2400 11: 19200/9600		Prescaler & Timer Run/Stop Control 0: Stop & Clear 1: Run (Count up)					

270890

Figure 3.6 (15) Timer Run Control Register

P83 control to set it TO4 function

		7	6	5	4	3	2	1	0
P8CR (FFD1H)	bit Symbol					P83OC	ZCE2	ZCE1	EDGE
	Read/Write					W	W	W	W
	Resetting Value					0	0	0	0
	Function					P83 Control 0: P83 1: TO3 / TO4	INT2/TI5 Control 1: ZCD enable	INT1/TI4 Control 1: ZCD enable	INT0 Control 0: Level 1: ↑ edge

Prohibit Read Modify Write.

Timer selection for stepping motor control port

		7	6	5	4	3	2	1	0
SMMOD (FFCBH)	bit Symbol		SM7M0	P7OC1	P7OC0		SM6M0	P6OC1	P6OC0
	Read/Write		R/W	R/W			R/W	R/W	
	Resetting Value		0	0	0		0	0	0
	Function		0: 1 step/ 2 step excitation 1: 1-2step excitation	00: IN/OUT 01: IN/OUT, TO3 10: IN/M1 (Timer 2, 3) 11: IN/M1 (Timer 4)			0: 1 step/ 2 step excitation 1: 1-2 step excitation	00: IN/OUT 01: IN/OUT, TO1 1X: IN/M0 (Timer 0, 1)	

※ Refer a following table, when P83 is used as timer output port.

Output Port P83 Control

P8CR<P83OC>	SMMOD<P7OC1, 0>	P83 Function
0	XX	P83 Output Port
1	0X	TO4
1	10	
1	11	TO3

200990

Table 3.6 (16) P83 Control for Timer Output (TO4/TO3)

① Up-counter (UC16)

UC16 is a 16-bit binary counter that counts up by the input clock specified by the register T4MOD<T4CLK1, 0>.

Its input clock is selected from the internal clocks ϕT1 and ϕT16 generated by the 9-bit prescaler (also used for the 8-bit timer), or the external clock from the TI4 pin (also used as P81/INT1). By resetting, <T4CLK1,0> of T4MOD are initialized to "0, 0", whereby TI4 is selected as the input clock to this up-counter.

The Timer Control Register TRUN<T4RUN> controls the counter to start, stop and clear.

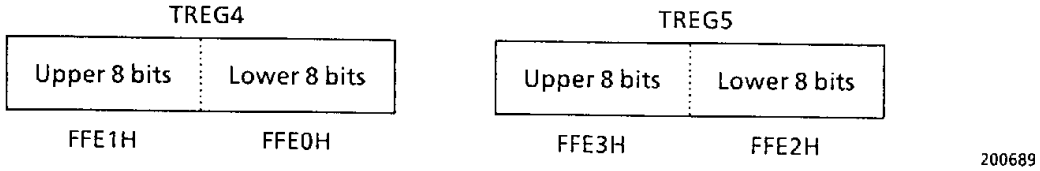
The up-counter UC16 is cleared to "0" only when matching the timer register TREG5, if T4MOD<CLE> = "1".

When UC16 is set in a clear Disable Mode (T4MOD<CLE> = "0"), this counter operates as a free-running counter.

② Timer registers (TREG4 and TREG5)

Two 16-bit registers are provided to set the counter value. When the values in these timer registers equal with the value of the up-counter UC16, the match signal of the comparator becomes active.

Data for the timer registers TREG4 and TREG5 are provided by a 2-byte data load instruction or two 1-byte data load instructions, from the lower eight bits followed by the upper eight bits.

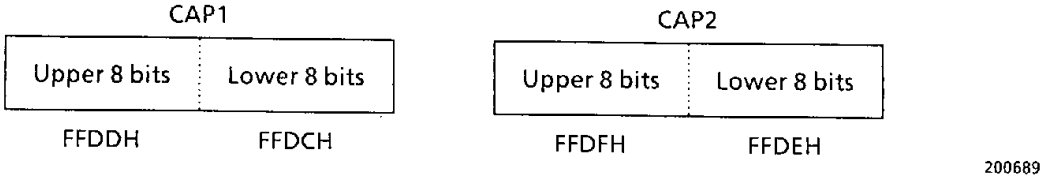


200689

③ Capture registers (CAP1 and CAP2)

CAP1 and CAP2 are 16-bit registers that capture the values of the up-counter UC16.

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



200689

④ Capture input control circuit

This circuit controls the timing that the capture registers (CAP1 and CAP2) latch the UC16 up-counter value.

The latch timing is set by the register T4MOD<CAPM1,0>.

- If T4MOD<CAPM1,0> = 0, 0,

  The capture function is disabled. These bits are initialized to this mode by resetting.

- If T4MOD<CAPM1, 0> = 0, 1,

  The up-counter value is latched into CAP1 at the rising edge of the TI4 (also used as P81/INT1) input, and into CAP2 at the rising edge of TI5 (also used as P82/INT2) input. (Time difference measurement)

- If T4MOD<CAPM1, 0> = 1, 0,

  The up-counter value is latched into CAP1 at the rising edge of the TI4 input, and into CAP2 at its falling edge. Only in this mode, the interrupt INT1 occurs at the falling edge. (Pulse width measurement)

- If T4MOD<CAPM1, 0> = 1, 1,

  The up-counter value is latched into CAP1 at the rising edge of the timer flip-flop TFF1, and into CAP2 at its falling edge. (Frequency measurement)

The up-counter value can be also latched into the capture registers by using software. In this case, the current up-counter value is latched into CAP1 each time "0" is written into the T4MOD<CAPIN> register.

⑤ Comparators (CP4 and CP5)

The 16-bit comparators detect the match of the up-counter UC16 and the timer register TREG4 or TREG5.

When the up-counter matches registers TREG4 or TREG5, the interrupt INTT4 or INTT5 occurs, respectively. The up-counter is cleared only when it matches TREG5 (the clear operation can be disabled by setting T4MOD<CLE> = "0", if necessary).

⑥ Timer Flip-flop (TFF4)

This Timer Flip-flop is inverted by the match signal from the comparators (CP4 and CP5) and the latch signal to the capture registers (CAP1 and CAP2).

It is possible to enable/disable the inversion for each of these sources by using T4FFCR<CAP2TE, CAP1TE, EQ5TE, EQ4TE>.

Also TFF4 can be inverted, cleared to "0" and set to "1" by setting T4FFCR.

Its value can be output to the timer output pin TO4 (also used as P83 or TO3). Either pin function may be selected by the registers P8CR and SMMOD. TO4 is selected by setting P8CR<P83OC> = 1 and SMMOD<P70C1,0> ≠ 1, 1.

(1) 16-bit Timer Mode

In this example, the interval time is set in the timer register TREG5 to generate the interrupt INTT5. The up-counter (UC16) is not cleared by matching with TREG4. Therefore, the TREG4 should be selected for interval timer.

```
TRUN  ←  - - - 0 - - - -        Stop Timer 4.
INTEL ←  - - 0 - 1 - - -        Enable INTT5 and disable INTT4.
T4FFCR←  X X 0 0 0 0 1 1        Disable trigger.
T4MOD ←  X X 1 0 0 1 * *        Select internal clock for input, and
                 (**=01,10,11)  disable the capture function.
TREG5 ←  **** **** **** ****    Set the interval time (16 bits).
TRUN  ←  - - 1 1 - - - -        Start Timer 4.
```

(Note) ×: Don't care - : No change

(2) 16-bit Event Counter Mode

This timer can be used as a 16-bit event counter by selecting the external clock (TI4) as the input clock in the above timer mode (1).

When reading the counter value, use "software capture" and read the capture value.

The counter counts up at the rising edge of the TI4 input clock.

The TI4 pin is also used as P81 or INT1.

```
TRUN  ←  - - - 0 - - - -        Stop Timer 4.
P8CR  ←  - - - - - - - * -      *=0; TI4 is square wave.
                                *=1; TI4 is sign wave (zero-cross)
INTEL ←  - - 0 0 1 - - -        Enable INTT5 and disable INTT4 and INT1.
T4FFCR←  X X 0 0 0 0 1 1        Disable trigger.
T4MOD ←  X X 1 0 0 1 0 0        Select TI4 as the input clock.
TREG5 ←  **** **** **** ****    Set the number of counts (16 bits).
TRUN  ←  - - 1 1 - - - -        Start Timer 4 and Prescaler.
```

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

The PPG mode is obtained by inversion of the Timer Flip-flop TFF4 to be enabled by match of the up-counter UC16 with the timer register TREG 4 or 5 and output it to TO4 (also used as P83/TO3). In this mode the following condition must be satisfied.

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

TRUN ← - - - 0 - - - -	Stop Timer 4.
TREG4 ← **** **** **** ****	Set the duty rate.
TREG5 ← **** **** **** ****	Set the cycle.
T4FFCR← X X 0 0 1 1 0 0	Set the TFF4 inversion to be effected by match with TREG4 or 5. Initialize TFF4 to "0".
T4MOD ← X X 1 0 0 1 * *	Select the internal clock for the input, and disable the capture function.
SMMOD ← - - * * - - - -	Select P83 as the TO4 pin.
(**=01,10,11)	
P8CR ← - - - - 1 - - -	
(**=00,01,10)	
TRUN ← - - 1 1 - - - -	Start Timer 4.

(Note) ×: Don't care -: No change

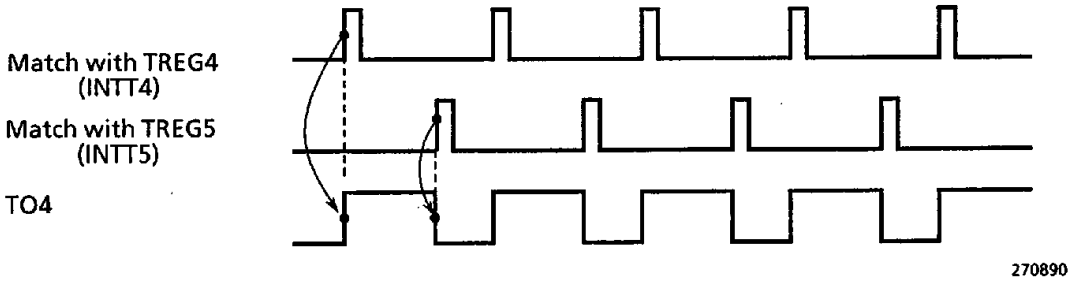


Figure 3.6 (17) Programmable Pulse Output

(4) Application Examples of Capture Function

It can be enabled or disabled that loading the up-counter (UC16) value into CAP1 or CAP2, the Timer Flip-flop TFF4 inversion by the match signal from CP4 or CP5, and the output of the TFF4 status to TO4 pin.

Various applications are available by combining with the interrupt function:

① One-shot pulse output by using external trigger pulse.
② Frequency measurement.
④ Pulse width measurement.
⑤ Time difference measurement.

① One-shot pulse output from the rising edge of external trigger pulse.

Keep the up-counter (UC16) counting (free-running) with the internal clock by setting T4MOD<CLE> = "0". Input an external trigger pulse through the TI4 pin, and also load the up-counter value into the capture register CAP1 at the rising edge of TI4 (T4MOD4, 3 = 01).

When the interrupt INT1 is generated at the rising edge of TI4 input, set the CAP1 value (c) plus a delay time (d) to TREG4 (=c+d), and set the above set value (c+d) plus a one-shot pulse width (p) to TREG5 (=c+d+p). In the interrupt routine of INT1, the T4FFCR register should be set that the TFF4 inversion is enabled only when the up-counter value matches TREG4 or 5. In the interrupt routine INTT5 this inversion should be disabled.

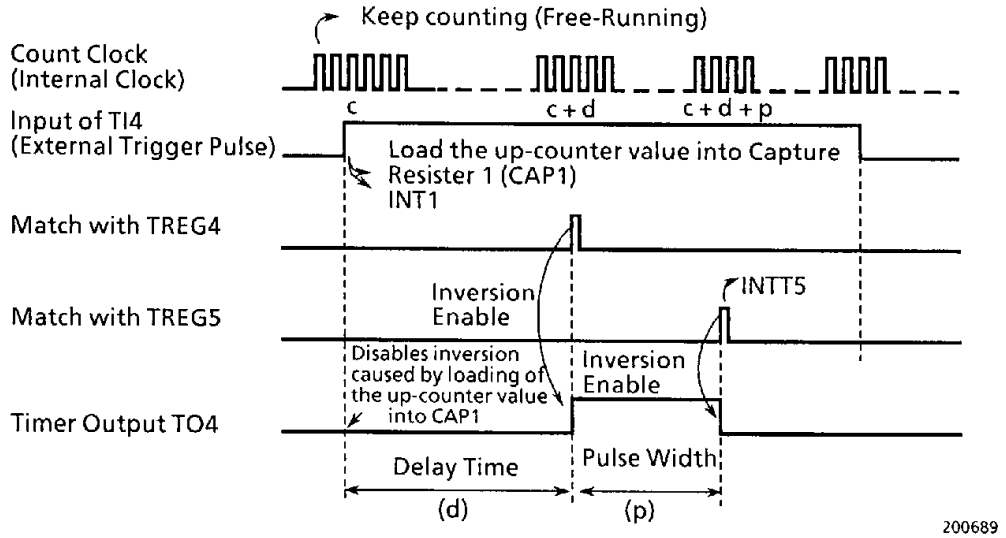
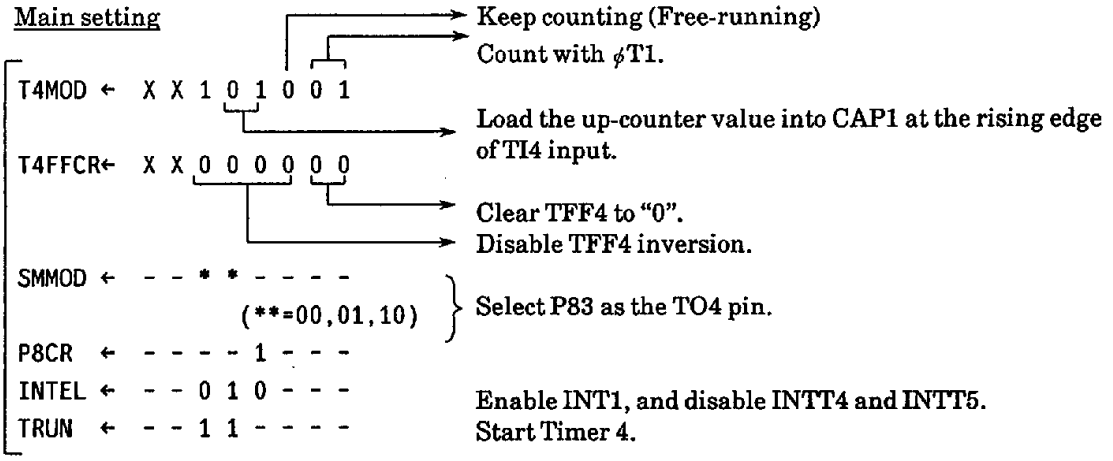
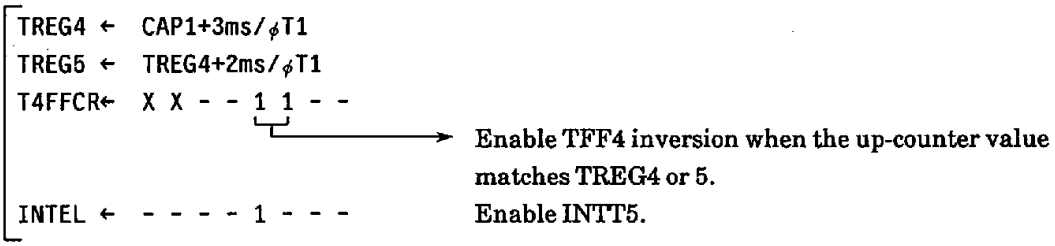


Figure 3.6 (18) One-Shot Pulse Output (with delay)

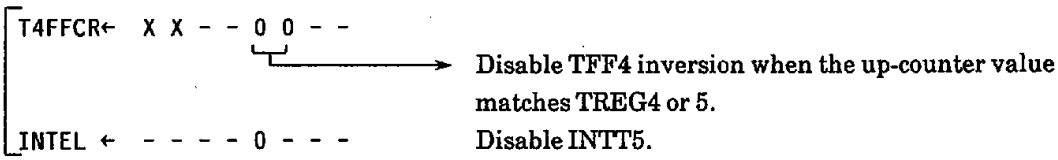
Example: Generate a 2ms one-shot pulse with a 3ms delay from the rising edge of an external trigger pulse.



Setting of INT1



Setting of INT5



(Note) ×: Don't care -: No change

When no delay time is required, invert the TFF4 when the up-counter value is loaded into CAP1, and set the CAP1 value (c) plus the one-shot pulse width (p) to TREG5 when the interrupt INT1 occurs. The TFF4 inversion should be enabled before the up-counter (UC16) value matches TREG5, and disabled when generating the interrupt INTT5.

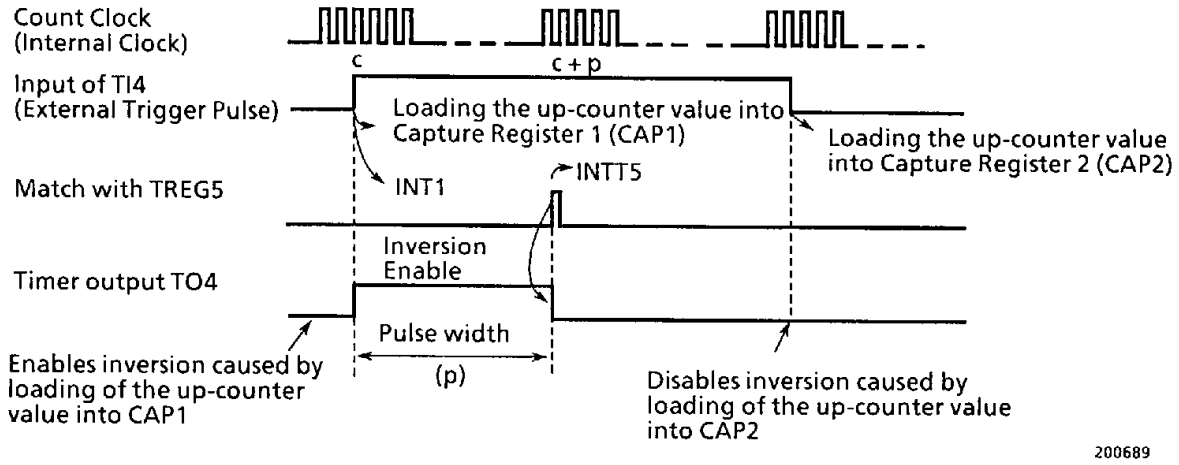


Figure 3.6 (19) One-Shot Pulse Output (with no delay)

② Frequency measurement

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

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

The frequency is calculated by using the number of input clock in a certain period (the difference between the loaded values in CAP1 and CAP2 when the interrupt (INTT0 or INTT1) is generated by either 8-bit timer.)

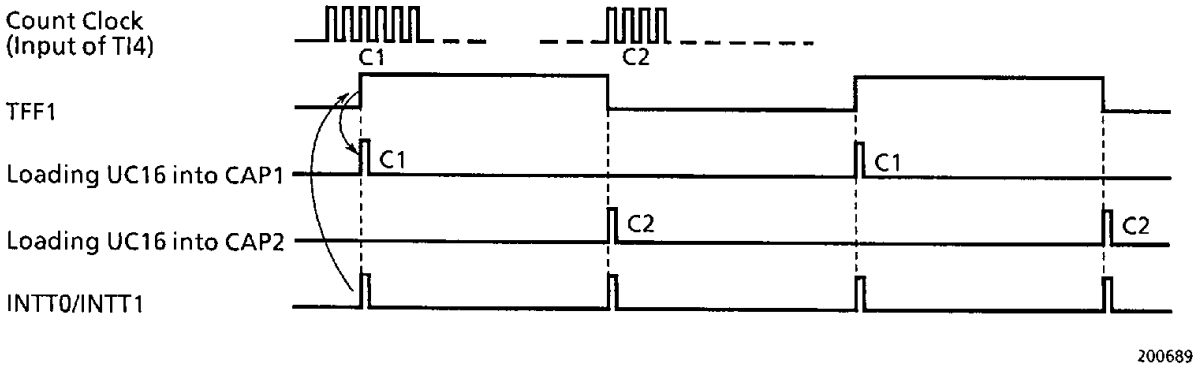


Figure 3.6 (20) Frequency Measurement

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

③ Pulse width measurement

This mode allows to measure the "H" level width of an external pulse. While keeping the 16-bit timer/event counter counting (free-running) with the internal clock input, the external pulse is input through the TI4 pin. Then the capture function is used to load the UC16 values into CAP1 and CAP2 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT1 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 μs and the difference between CAP1 and CAP2 is 100, the pulse width will be $100 \times 0.8\ \mu s = 80\ \mu s$.

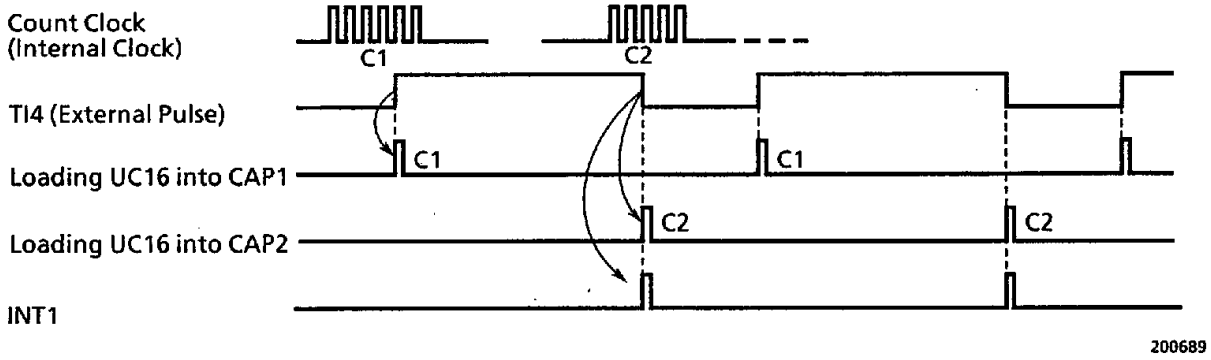


Figure 3.6 (21) Pulse Width Measurement

Note: The external interrupt INT1 occurs at the falling edge of TI4 only in this pulse width measurement mode T4MOD<CAPM1, 0>=1, 0. In any other mode, the interrupt occurs at its rising edge.

The width of "L" level can be obtained from the difference between the first C2 and the second C1 when the second INT1 interrupt is generated.

In this case, the first C2 should be stored in a RAM or register in the first INT1 interrupt routine.

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 UC16 value into CAP1 at the rising edge of the input pulse to TI4. Then the interrupt INT1 is generated.

Similarly, the UC16 value is loaded into CAP2 at the rising edge of the input pulse to TI5, generating the interrupt INT2.

The time difference between these pulses can be obtained from the difference between the time loading the up-counter value into CAP1 and CAP2.

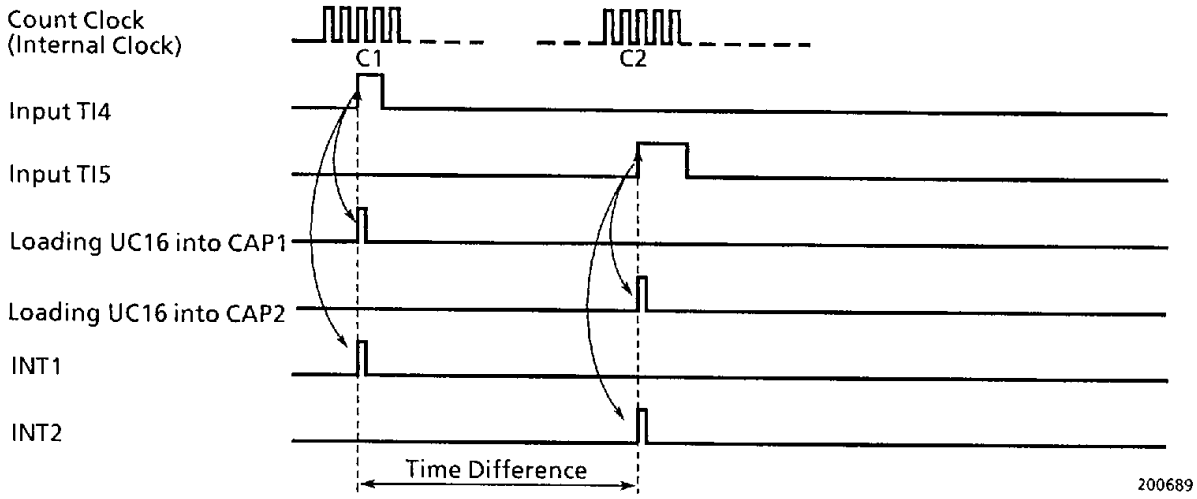


Figure 3.6 (22) Time Difference measurement

3.7 Stepping Motor Control Ports (P6 and P7)

The TMP90C840A has two 4-bit hardware Stepping Motor Control ports (SMC ports M0 and M1) that synchronize with the (8-bit/16-bit) timers. These SMC ports M0 and M1 are also used as the I/O ports P6 and P7.

The channel 0 (M0) synchronizes with the trigger signal of the Timer Flip-flop TFF1, and the channel 1 (M1) is synchronous with that of the Timer Flip-flop TFF3 or TFF4, for output.

The SMC ports are controlled by three control registers (P67CR, SMMOD, and SMCR), and allow the selection of driving method: 4-phase 1-step/2-step excitation and 4-phase 1-2 step excitation.

3.7.1 Control Registers

(1) Port 6 and Port 7 I/O Selection Register (P67CR)

This register specifies either input or output for each bit of the 4-bit I/O ports 6 and 7.

When all bits of P67CR are initialized to "0" by resetting, Ports 6 and 7 function as input ports.

P67CR is a write-only register (can not be readout).

P67CR (FFCEH)	7	6	5	4	3	2	1	0
bit Symbol	P73C	P72C	P71C	P70C	P63C	P62C	P61C	P60C
Read/Write	W				W			
Resetting Value	0	0	0	0	0	0	0	0
Function	0 : In 1 : Out				0 : In 1 : Out			

Prohibit Read Modify Write.

Select Port 6 I/O (On bit basis)

0	Input
1	Output

Select Port 7 I/O (On bit basis)

0	Input
1	Output

270890

Figure 3.7 (1) Ports 6 and 7 I/O Selection Registers

(2) Stepping Motor Control Port Mode Register (SMMOD)

Ports 6 and 7 also function as SMC Ports (M0 and M1) or Timer Output Ports (TO1 and TO3), as selected by SMMOD<P60C1, 0> or SMMOD<P70C1, 0>.

When Port 6 is used as SMC Port (M0), SMMOD<P60C1, 0> should be set to "1, 0" or "1, 1".

When Port 7 is used as SMC Port (M1), SMMOD<P70C1, 0> should be set to "1, 0" to make it synchronize with the trigger pulse of the timer flip-flop TFF3 and set to "1, 1" to make it synchronize with the trigger pulse of the Timer Flip-flop TFF4. P83 (also used as TO3 and TO4) can function as TO3 only when SMMOD<P70C1, 0> are set to "1, 1".

SMMOD<SM6M0> and SMMOD<SM7M0> serve to select the excitation method. With "0" the full-step (1-step/2-step) excitation is selected and with "1" the half-step (1-2 step) excitation is selected. When full-step excitation is selected, the selection of either 1 or 2-step excitation is determined by the initial output value.

SMMOD (FFCBH)

	7	6	5	4	3	2	1	0
bit Symbol	—	SM7M0	P7OC1	P7OC0	—	SM6M0	P6OC1	P6OC0
Read/Write	—	R/W	R/W		—	R/W	R/W	
Resetting Value	—	0	0	0	—	0	0	0
Function	—	0: 1 or 2 step excitation 1: 1-2 step excitation	00: IN/OUT 01: IN/OUT, TO3 10: IN/M1 (Timer 2, 3) 11: IN/M1 (Timer 4)		—	0: 1 or 2 step excitation 1: 1-2 step excitation	00: IN/OUT 01: IN/OUT, TO1 1X: IN/M0 (Timer 0, 1)	

Select Port 6 Function (On bit basis)

P67CR / P6OC1, 0	P63		P62		P61		P60		
	P63C		P62C		P61C		P60C		SMC Trigger Signal
	0	1	0	1	0	1	0	1	
00	IN	OUT	IN	OUT	IN	OUT	IN	OUT	Not generated
01	IN	OUT	IN	OUT	IN	OUT	IN	TO1	
10	IN	M03	IN	M02	IN	M01	IN	M00	TFF1(Timer 0 or Timer 1)
11									

IN : Input Port
OUT : Output Port
TO1 : Timer Output Port
M00-3 : Stepping motor control port 0

Stepping motor control port 0 mode (Excitation)

0	1 or 2 excitation (Full-step)
1	1-2 Excitation (Half-step)

280890

Figure 3.7 (2a) Stepping Motor Control Port Mode Registers

(3) Stepping Motor Control Port Rotating Direction Control Register (SMCR)

This register controls the rotating direction. The direction of the channel 0 (M0) is set by SMCR<CCW6> and that of the channel 1 (M1) is set by SMCR<CCW7>.
"0" denotes normal rotation and "1" denotes reverse rotation.

(4) Port 6/Port 7

They are 4-bit I/O ports allocated to the address FFCCH for Port 6 and FFCDH for Port 7.

The lower four bits are assigned as Port function, while the upper four bits operate as the shifter alternate register for driving the stepping motor with the half-step (1-2) excitation.

SMMOD (FFCBH)

	7	6	5	4	3	2	1	0
bit Symbol		SM7M0	P7OC1	P7OC0		SM6M0	P6OC1	P6OC0
Read/Write		R/W	R/W			R/W	R/W	
Resetting Value		0	0	0		0	0	0
Function		0: 1 or 2 step excitation 1: 1-2 step excitation	00 : IN / OUT 01 : IN / OUT, TO3 10 : IN / M1 (Timer 2, 3) 11 : ⁄ (Timer 4)			0: 1 or 2 step excitation 1: 1-2 step excitation	00 : IN / OUT 01 : IN / OUT, TO1 1X : IN / M0 (Timer 0, 1)	

Select Port 7 Function (On bit basis)

	P73		P72		P71		P70		P83		
P67CR &P8CR	P73C		P72C		P71C		P70C		P83OC		SMC Trigger Signal
P7OC1, 0	0	1	0	1	0	1	0	1	0	1	
00	IN	OUT	IN	OUT	IN	OUT	IN	OUT	OUT	TO4	Not generated
01	IN	OUT	IN	OUT	IN	OUT	IN	TO3			
10	IN	M13	IN	M12	IN	M11	IN	M10			TFF3 (Timer 2, 3)
11									OUT	TO3	TFF4 (Timer 4)

IN : Input Port
OUT : Output Port
TO3, TO4 : Timer Output Port
M10-3 : Stepping motor control port 1

Stepping motor control port 1 mode (Excitation)

0	1 or 2 excitation (Full-step)
1	1-2 Excitation (Half-step)

280890

Figure 3.7 (26) Stepping Motor Control Port Mode Registers

SMCR (FFCFH)

	7	6	5	4	3	2	1	0
bit Symbol				CCW7				CCW6
Read/Write				R / W				R / W
Resetting Value				0				0
Function				0: Normal 1: Reverse				0: Normal 1: Reverse

Stepping motor control port 1 rotating direction

0	Normal rotation
1	Reverse rotation

Stepping motor control port 0 rotating direction

0	Normal rotation
1	Reverse rotation

280890

Figure 3.7 (3) Stepping Motor Control Port Rotating Direction Control Registers

P6 (FFCCH)

	7	6	5	4	3	2	1	0
bit Symbol	SA67	SA62	SA61	SA60	P63	P62	P61	P60
Read/Write	W				R / W			
Resetting Value	Undefined				Input mode			
Function	Shifter alternate register 0 for stepping motor control				Shared with stepping motor control port 0 (M0)			

P7 (FFCDH)

	7	6	5	4	3	2	1	0
bit Symbol	SA73	SA72	SA71	SA70	P73	P72	P71	P70
Read/Write	W				R / W			
Resetting Value	Undefined				Input mode			
Function	Shifter alternate register 1 for stepping motor control				Shared with stepping motor control port 1 (M1)			

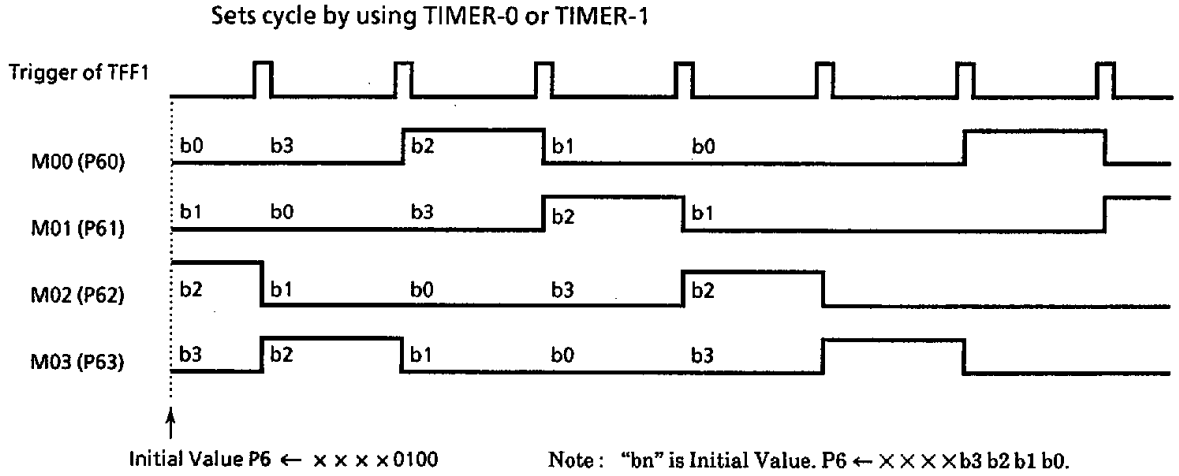
Note : Read Modify Write instructions can not be used for writing P6/P7 register, when these port (P6/P7) are used as Stepping Motor Control port.

280890

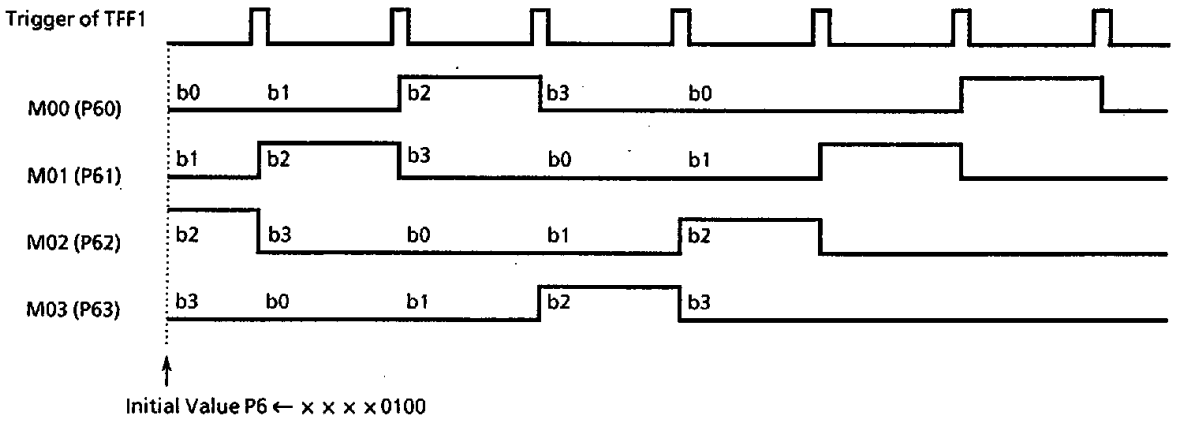
Figure 3.7 (4) Port 6 and 7

3.7.2 4-Phase 1-Step/2-Step Excitation

Figures 3.7 (5) and (6) show the output waveforms of 4-phase 1-step excitation and 2-step excitation when Channel 0 is selected.



① Normal rotation



② Reverse rotation

190889

Figure 3.7 (5) Output Waveforms of 4-phase 1-step Excitation (Normal Rotation and Reverse Rotation)

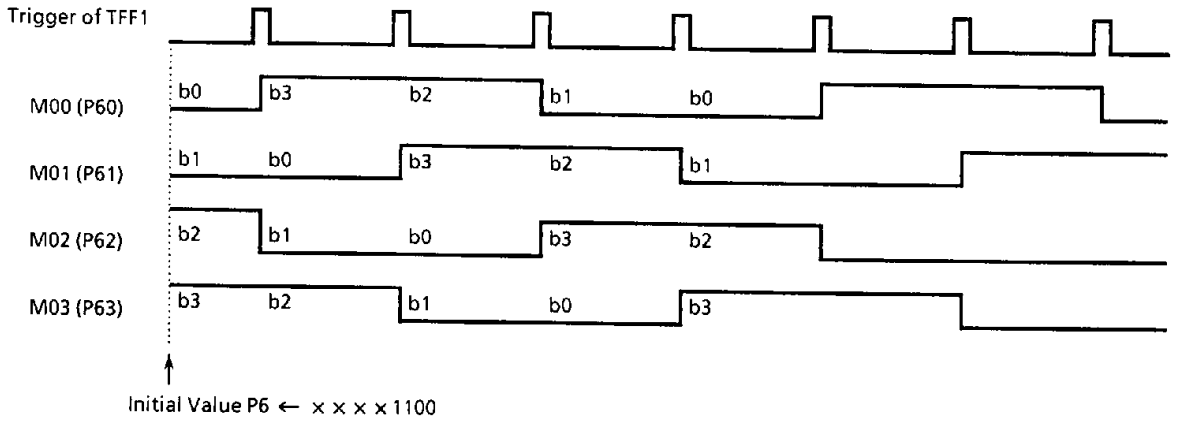


Figure 3.7 (6) Output Waveforms of 4-phase 2-step Excitation (Normal Rotation)

When Channel 0 is selected, for example, the stepping motor control port is controlled as follows:

The output data in the output latch of M0 (also used as P6) rotates at the rising edge of the TFF1 trigger pulse (that inverts the value of TFF1) and is output to the port.

The rotating direction is selected by SMCR<CCW6>. Setting SMCR<CCW6> to "0" selects the normal rotation (M00 → M01 → M02 → M03), and setting it to "1" results in the reverse rotation (M00 ← M01 ← M02 ← M03). The 4-phase 1-step excitation can be obtained by initializing any one bit of Port 6 to "1", and setting two succeeding bits to "1" produces the 4-phase 2-step excitation.

Figure 3.7 (7) is a block diagram of the two excitations.

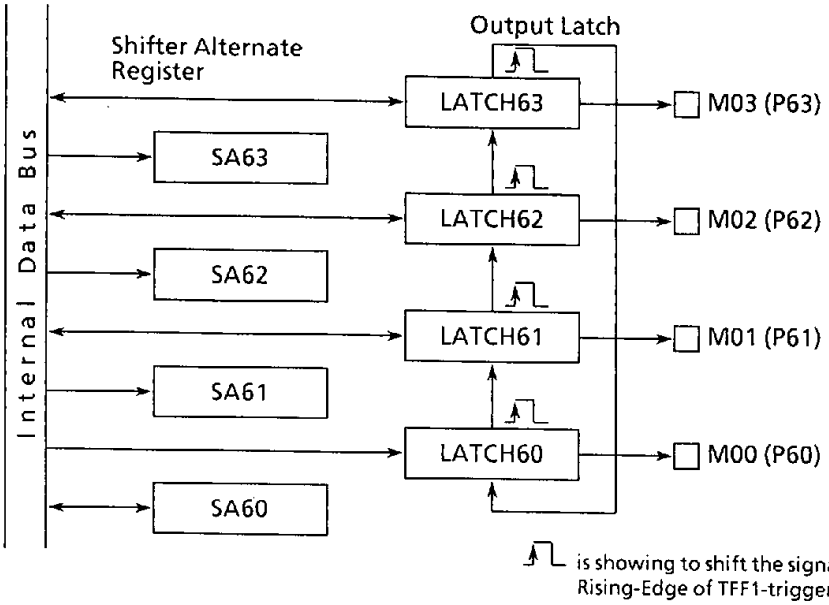
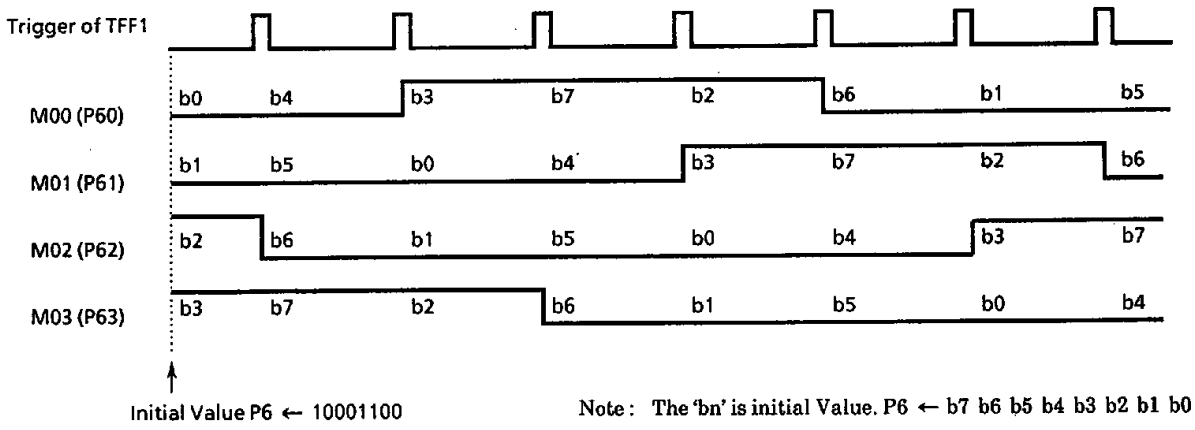


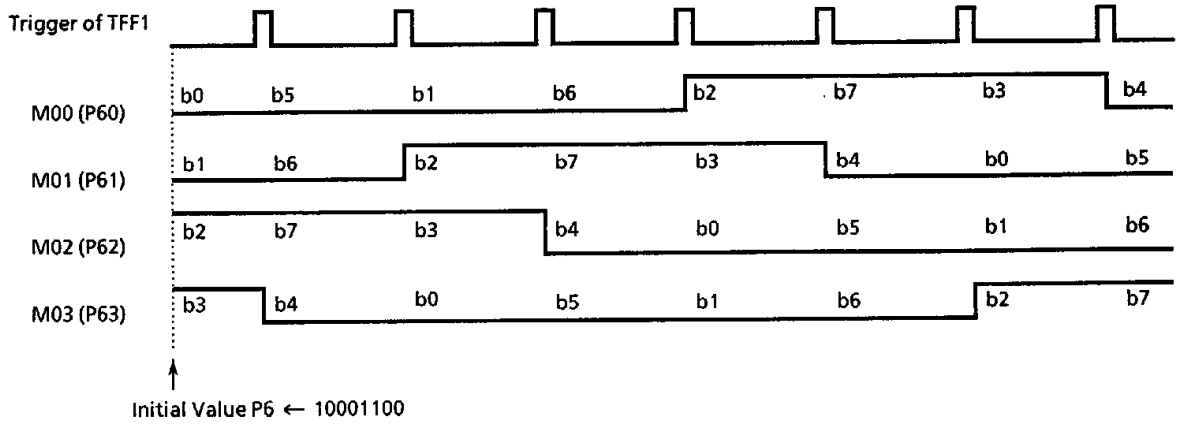
Figure3.7 (7) Block Diagram of 4-phase 1-step/2-step Excitation (Normal Rotation)

3.7.3 4-Phase 1-2 Step Excitation

Figures 3.7 (8) shows the output waveforms of 4-phase 1-2 step excitation when Channel 0 is selected.



① Normal rotation



② Reverse rotation

200689

Figure 3.7 (8) Output Waveforms of 4-phase 1-2 step Excitation (Normal Rotation and Reverse Rotation)

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

Given the Port bits | b7 b6 b5 b4 b3 b2 b1 b0 | initialially arranged as, | b3 b7 b2 b6 b1 b5 b0 b4 | set three succeeding bits to "1", and the other bits to "0" (Positive logic).

For example, initializing b3, b7 and b2 to "1" provided the initial value "10001100", which produces the waveforms in Figure 3.7 (8).

To obtain the negative logic output, the above initial value of Port 6 should be inverted. That is, the waveforms in Figure 3.7 (8) can be converted to the negative logic output by initializing the P6 bits to "011110011".

When Channel 0 is selected, for example, the stepping motor control port is controlled as follows:

The data in the output latch of M0 (also used as P6) and in the stepping motor control shifter alternate register (SA6) rotate at the rising edge of the TFF1 trigger pulse and are output to the port. The rotating direction is selected by SMCR<CCW6>.

Figure 3.7 (9) is a block diagram of this excitation.

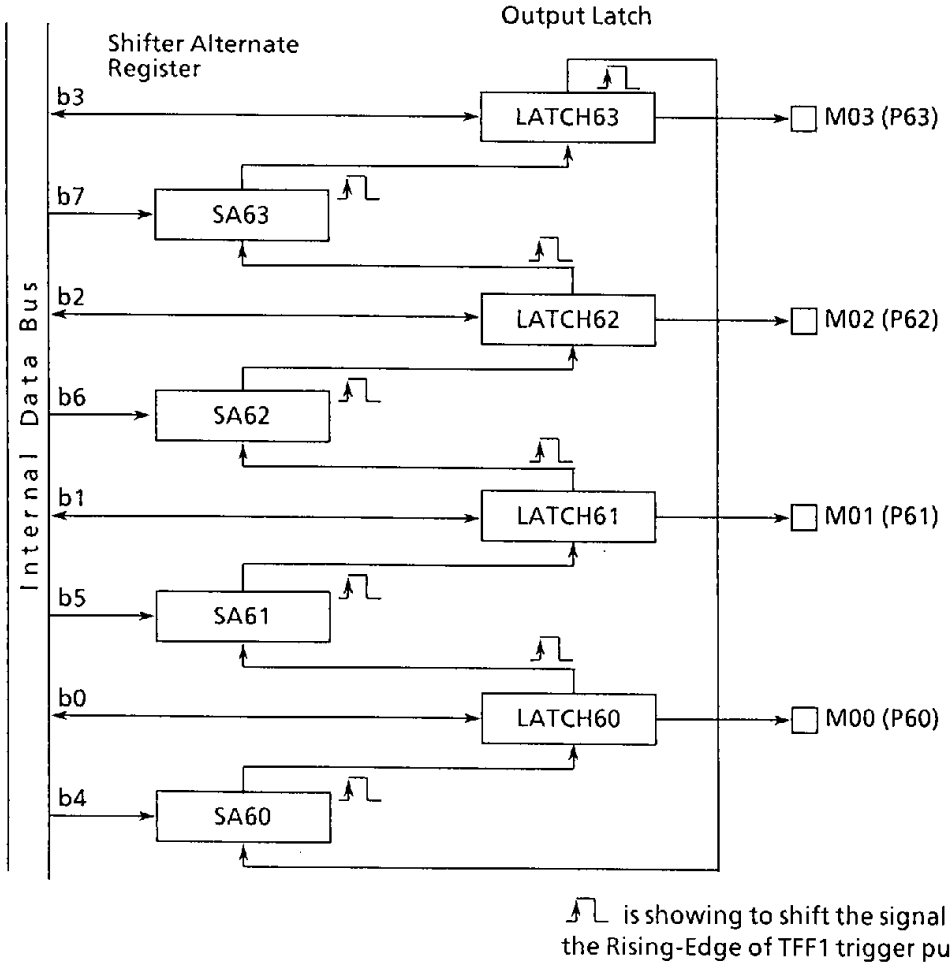


Figure 3.7 (9) Block Diagram of 4-phase 1-2 step Excitation (Normal Rotation)

280890

Example: The registers should be set as follows to drive Channel 0 (M0) with the 4-phase 1-2 step excitation (normal rotation) when Timer 0 is selected:

```
        7 6 5 4 3 2 1 0
TRUN  ← - - - - - - - - 0    Stop Timer 0, and clear it to "0".
TMOD  ← - - - - - 0 0 X X    Set the 8-bit timer mode.
TCLK  ← - - - - - - - - 0 1  Select φT1 as the input clock.
TREG0 ← * * * * * * * * *    Set the cycle in Timer register.
TFFCR ← - - - - - 0 0 1 0    Clear TFF1 to "0", and enable inversion trigger by
                             Timer 0
SMMOD ← - - - - - 1 1 1 X    Select 4-phase 1-2 step excitation mode.
SMCR  ← - - - - - - - - - 0  Select normal rotation.
P67CR ← - - - - - 1 1 1 1    Set all P6 bits to the output mode.
P6    ← 1 0 0 0 1 1 0 0      Set the initial value of P6.
TRUN  ← - - - 1 - - - - 1    Start Timer 0.
```

(Note) ×: Don't care -: No change

The upper bits of the port register (P6 and P7) cannot be read. In order to modify the data in the port register (P6 and P7) when these ports are setting to the 4 phase 1-2 step excitation, refer a table 3.7 (1).

Table 3.7 (1) The actual data of shifter

Readed Data		Actual Data
F1H	→	31H
F2H	→	62H
F3H	→	23H
F4H	→	C4H
F6H	→	46H
F8H	→	98H
F9H	→	19H
FCH	→	8CH

280890

3.8 Serial Channel

The TMP90C840A incorporates a serial I/O channel for full-duplex asynchronous transmission (UART) and I/O expansion.

The serial channel has the following operating modes:

- I/O interface mode ——— Mode 0: Transmit/Receive I/O data for expand I/O and transmit its synchronous signals SCLK.
- Asynchronous transmission (UART) mode
  - Mode1 : 7-bit data
  - Mode2 : 8-bit data
  - Mode3 : 9-bit data

The Mode 1 and Mode 2 allow the addition of a parity bit. The Mode 3 accommodates a wake-up function to start the slave controllers in a controller serial link (multi-controller system).

Figure 3.8 (1) shows the data format (1-frame data) in each mode.

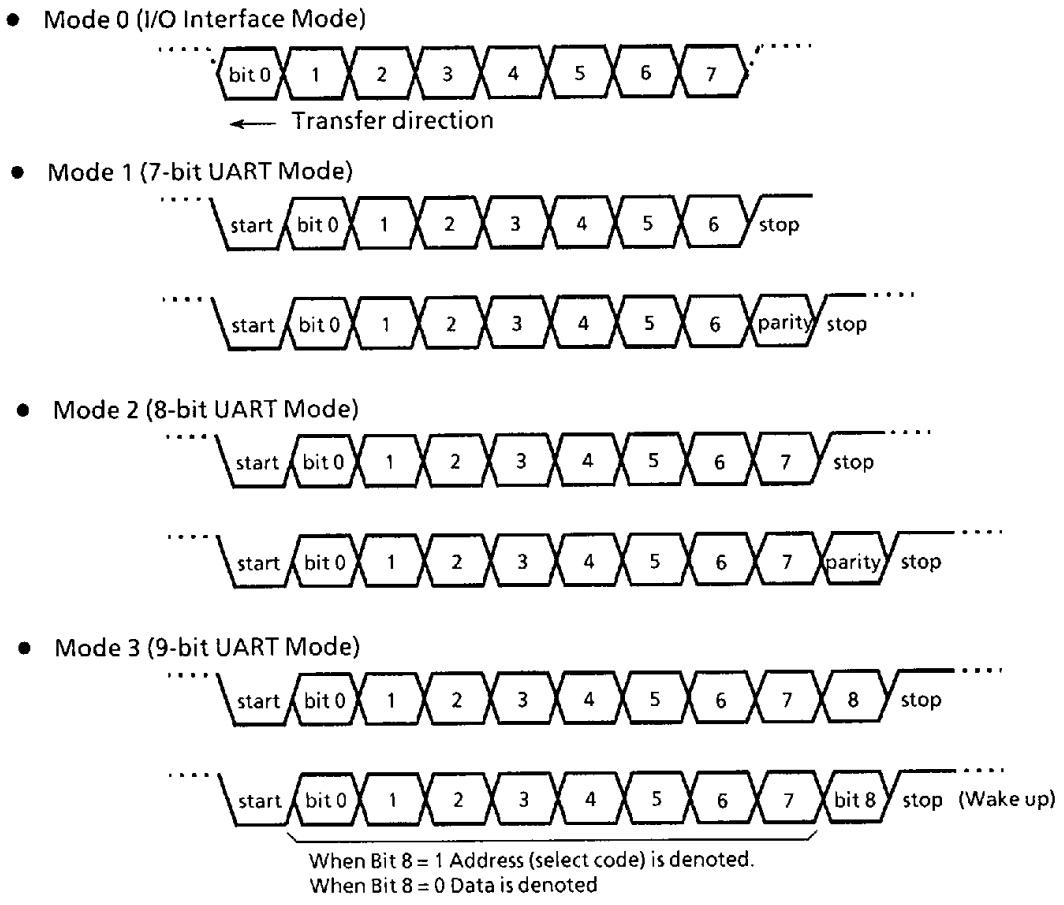


Figure 3.8 (1) Data Formats

Data received and transmitted are stored temporarily into separate buffer registers to allow independent transmission and receiving (Full-duplex).

In the I/O interface mode, however, the data transfer is half-duplex due to the single SCLK (serial clock) pin is used for transmission and receiving.

Two pins are provided for receiving (RxD/P30 and RxD/P31) and transmission (TxD/P32 and TxD/P33), which allows two serial transmission/receiving using the time-sharing . The pin function (Port function or serial I/O function) is selected by the port 3 control register P3CR. For example, P30 can be used as the RxD pin by setting P3CR<RXDC1, 0> to 01.

The receiving buffer register has a double-buffer structure to prevent overruns. The one buffer receives the next frame data while the other buffer stores the received data.

The use of $\overline{CTS}$ and $\overline{RTS}$ allows to halt the data transmission until the CPU completes reading of the receive data for each frame (Hand-shake 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 an request is issued to the CPU to transmit data after the transmitting buffer becomes empty or to read data after the receiving buffer completed to store data, the interrupt INTTX or INTRX occurs respectively. In receiving data, the occurrence of an overrun error, parity error or framing error sets the flag SCCR<OERR, PERR, FERR> accordingly.

3.8.1 Control Registers

The serial channel is controlled by four control registers (SCMOD, SCCR, TRUN, and P3CR). The received/transmitted data are stored into SCBUF.

SCMOD (FFE9H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	"0"	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
Resetting Value	Unknown	0	0	0	0	0	0	0
Function	Transmission Data Bit 8 (9bit UART)	"0"	1: Receive Enable	1: Wake up Enable	00: I/O interface 01: UART 7bit 10: UART 8bit 11: UART 9bit		00: TO2TRG 01: BR 10: ϕ1 11: BR 1/2	U A R T

Serial transfer clock

	UART mode	I/O interface
00	Timer 2 match signal	fc/8
01	Baud rate generator	
10	Internal clock ϕ1	
11	Baud rate generator 1/2 clock	

Serial transfer mode

00	I/O interface mode	
01	UART mode	7-bit data
10		8-bit data
11		9-bit data

Wake-up function

	9-bit UART	Other modes
0	Interrupt if data are received	don't care
1	Interrupt only if RB8 = 1.	

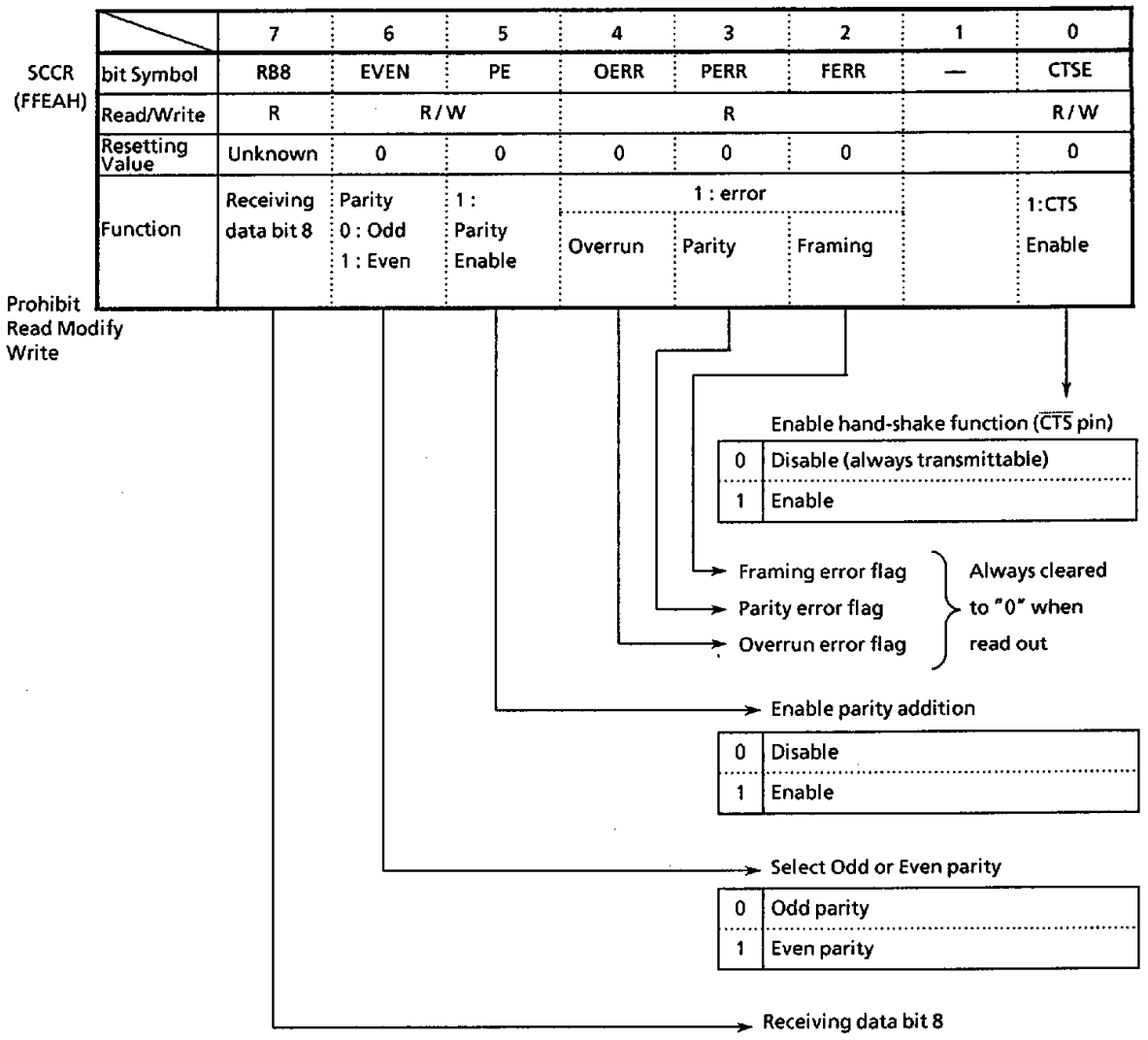
Enable receiving function

00	Disable
00	Enable

Transmission data bit 8

280890

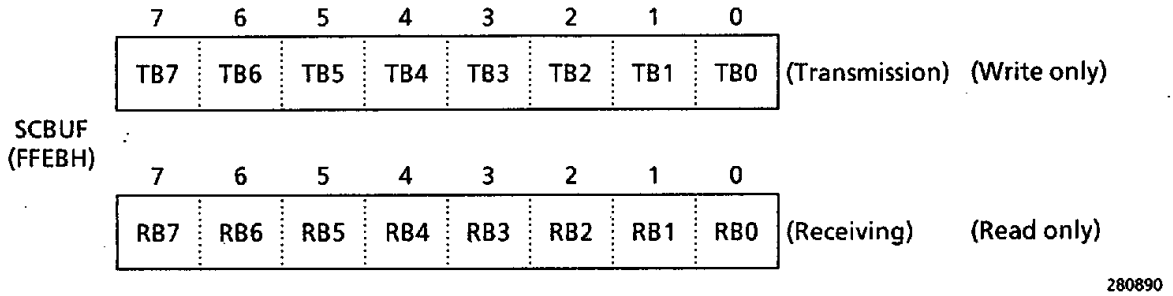
Figure 3.8 (2) Serial Channel Mode Register



(caution) Since all error flags are cleared after readout, avoid testing for only one bit using a bit-testing instruction

280890

Figure 3.8 (3) Serial Channel Control Register



280890

Figure 3.8 (4) Serial Transmission/Receiving Buffer Registers

TRUN (FFDBH)

	7	6	5	4	3	2	1	0
bit Symbol	BRATE1	BRATE0	PRRUN	T4RUN	T3RUN	T2RUN	T1RUN	TORUN
Read/Write	R/W		R/W					
Resetting Value	0	0	0	0	0	0	0	0
Funtion	00: 300/150 bps 01: 1200/600 10: 4800/2400 11: 19200/9600		Prescaler & Timer Run/Stop Control 0: STOP & Clear 1: Run (Count up)					

Timer 2 function

0	Stop & clear
1	Count

Prescaler function

0	Stop & clear
1	Count

Select transfer speed of serial I/O baud rate generator

	SCMOD<SC1, 0> = 01	SCMOD<SC1, 0> = 11
00	300 bps	150 bps
01	1200 bps	600 bps
10	4800 bps	2400 bps
11	19200 bps	9600 bps

@fc = 9.8304MHz

280890

Figure 3.8 (5) Timer/Serial Chanel Operation Control Register

P3CR (FFC7H)

	7	6	5	4	3	2	1	0
bit Symbol	WAITC1	WAITC0	RDE	ODE	TXDC1	TXDC0	RXDC1	RXDC0
Read/Write	R/W		R/W	R/W	R/W		R/W	
Resetting Value	0	0	0	0	0	0	0	0
Function	Wait Control 00: 2state wait 01: normal wait 10: non wait 11: reserved		RD Control 0: $\overline{RD}$ for only external access 1: Always $\overline{RD}$	P33 Control 0: CMOS 1: Open Drain	P33 00: Out 01: Out 10: TxD 11: TxD	P32 Out TxD Out $\overline{RTS}$/SCLK	P31 00: In 01: In 10: RxD 11: Not used	P30 In RxD In

Select P30 and P31 functions

	P31	P30
00	Input port	Input port
01	Input port	RxD pin
10	RxD pin	Input port
11	not used	

Select P32 and P33 functions

	P33	P32
00	Output port	Output port
01	Output port	TxD pin
10	TxD pin	Output port
11	TxD pin	$\overline{RTS}$/SCLK pin

Select P33 CMOS/Open-drain output

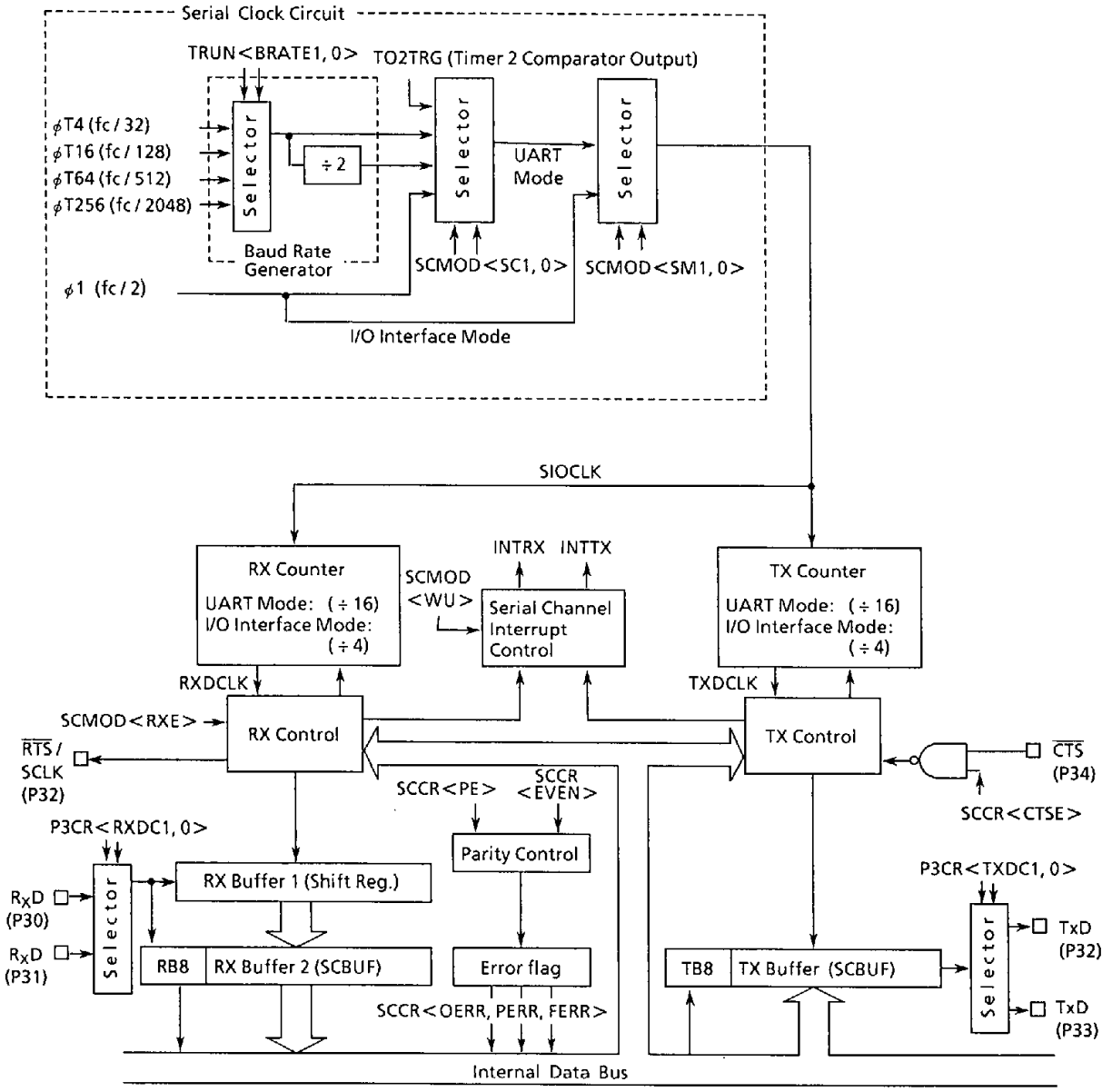
0	CMOS output
1	Open-drain output

280890

Figure 3.8 (6) Port 3 Control Register

3.8.2 Architecture

Figure 3.8 (7) is a block diagram of the serial channel.



Fifure 3.8 (7) Block Diagram of Serial Channel

① Baud-rate generator

The baud-rate generator comprises a circuit that generates a clock pulse to determine the transfer speed for transmission/receiving in the asynchronous communication (UART) mode.

The input clock to the baud-rate generator ϕT4 (fc/32), ϕT16 (fc/128), ϕT64 (fc/512) or ϕT256 (fc/2048) is generated by the 9-bit prescaler. One of these input clocks is selected by the timer/serial channel control register TRUN<BRATE1, 0>.

Also, either no frequency division or 1/2 division can be selected by the serial channel mode register SCMOD<SC1, 0>.

Table 3.8 (1) shows the baud-rate when fc=9.8304 MHz.

Table 3.8 (2) shows the baud-rate when use timer 2 (input clock: ϕT1).

Table 3.8 (1) Baud Rate Selection (1) [bps]

<BRATE1, 0>	Input clock	No division (SC1, 0 = 01)	1/2 division (SC1, 0 = 11)
00	ϕT256 (fc/2048)	300	150
01	ϕT64 (fc/512)	1200	600
10	ϕT16 (fc/128)	4800	2400
11	ϕT4 (fc/32)	19200	9600

@fc = 9.8304MHz

280890

Table 3.8 (2) Baud Rate Selection (2) (When using Timer 2 with ϕT1) [k bps]

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

280890

$$\text{Baud Rate} = \frac{1}{\text{TREG2}} \times \frac{1}{16} \times \text{Input clock of Timer 2}$$

Input Clock of Timer 2

ϕT1 = fc/8

ϕT16 = fc/128

ϕT256 = fc/2048

erating circuit

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

1) I/O interface mode

It generates a clock at a 1/8 frequency (1.25 M bit/s at 10 MHz) of the system clock (fc). This clock is output from the SCLK pin (also used as P32/$\overline{RTS}$).

2) Asynchronous communication (UART) mode

A basic clock (SIOCLK) is generated based on the above baud rate generator clock, the internal clock ϕ1 (fc/2) (SIOCLK = 5MHz, Transfer speed = 312.5kb.p.s. at 10MHz), or the match signal from Timer 2, as selected by SCMOD<SC1, 0> register.

③ Receiving counter

The receiving counter is a 4-bit binary counter used in the asynchronous communication (UART) mode and is counted by using SIOCLK. 16 pulses of SIOCLK is used for receiving 1 bit data. The data are sampled three times at 7th, 8th and 9th pulses and evaluated by the rule of majority. For example, if data sampled at the 7th, 8th and 9th clock are "1", "0" and "1", 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

The RxD signal is sampled on the rising edge of the shift clock which is output to the SCLK pin.

2) Asynchronous communication (UART) mode

The receiving control features 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 normal start bit and the receiving operation is started.
Data being received are also evaluated by the rule of majority.

⑤ Receiving buffer

The receiving buffer has a double-buffer structure to prevent overruns. Received data are stored into the Receiving buffer 1 (shift register type) for each 1 bit. When 7 or 8 bits data are stored in the Receiving buffer 1, the stored data is transferred to the Receiving buffer 2 (SCBUF), and the interrupt INTRX occurs at the same time. The CPU reads out the Receiving buffer 2 (SCBUF). Data can be stored into the Receiving buffer 1 before the CPU reads out the Receiving buffer 2 (SCBUF).

Note, however, that an overrun occurs unless the CPU reads out the Receiving buffer 2 (SCBUF) before the Receiving buffer 1 receives all bits of the next data.

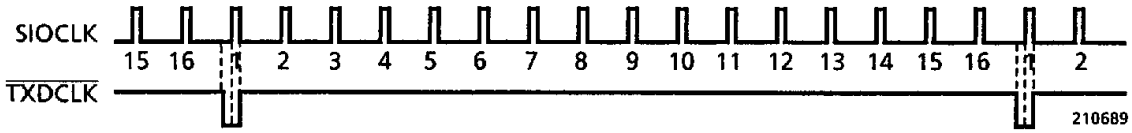
When an overrun occurred, the data in the buffer 2 and SCCR<RB8> are not lost, however, that in the buffer 1 are lost.

SCCR<RB8> stores the parity bit in the case adding parity in the 8-bit UART mode and the MSB in the 9-bit UART mode.

In the 9-bit UART mode, setting SCMOD<WU> to "1" enables the wake-up function of the slave controllers, and the interrupt INTRX occurs only if SCCR<RB8> = 1.

⑥ Transmission counter

This is a 4-bit binary counter used in the asynchronous communication (UART) mode. Like the receiving counter, it counts based on SIOCLK to generate a transmission clock TXDCLK for every 16 counts.



⑦ Transmission control

1) I/O interface mode

Data in the transmission buffer are output to the TxD pin bit by bit at the rising edge of the shift clock output from the SCLK pin.

2) Asynchronous communication (UART) mode

When the CPU have written data into the transmission buffer, transmission is started with the next rising edge of TxDCLK, and a transmission shift clock TxDSFT is generated.

Hand-shake function

The TMP90C840A supports a hand-shake function by the connection of $\overline{\text{CTS}}$ of one TMP90C840A and $\overline{\text{RTS}}$ of the other TMP90C840A.

The hand-shake function allows receiving/transmitting data on a frame basis to prevent overrun errors. This function is enabled or disabled by the control register SCCR<CTSE>.

When the last bit (parity bit or MSB) of 1-frame data is received by the receiving unit, the $\overline{\text{RTS}}$ pin turns to the "H" level to request the transmission unit to halt transmission.

When the $\overline{\text{CTS}}$ pin turned to the "H" level, the transmission unit halts transmission, after completing the current data transmission, until the pin turns to the "L" level. At this time, the interrupt INTTX is generated, to request the CPU to transfer data. Then the data is written into the transmission buffer, and the transmission unit is placed in the standby until the $\overline{\text{CTS}}$ pin turned to the "L" level.

When the received data are read by the CPU, the $\overline{\text{RTS}}$ pin returns to the "L" level, requesting that the transmission is restarted.

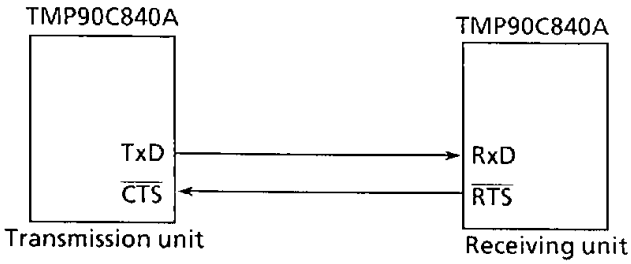
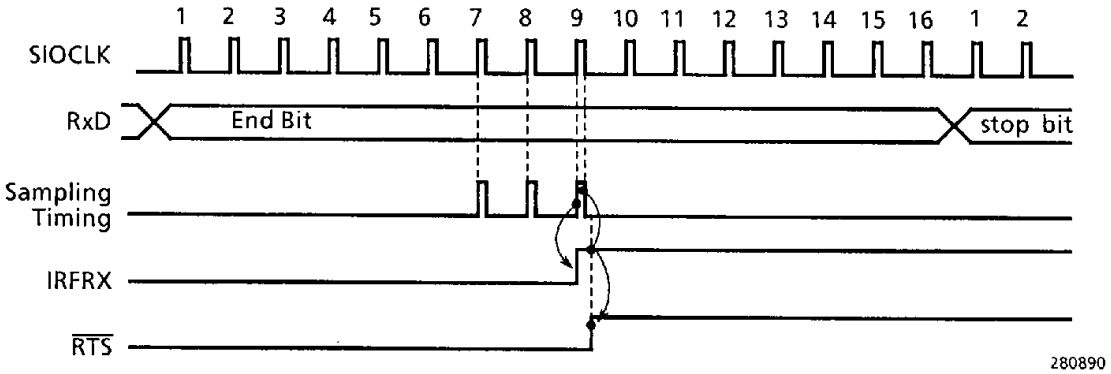
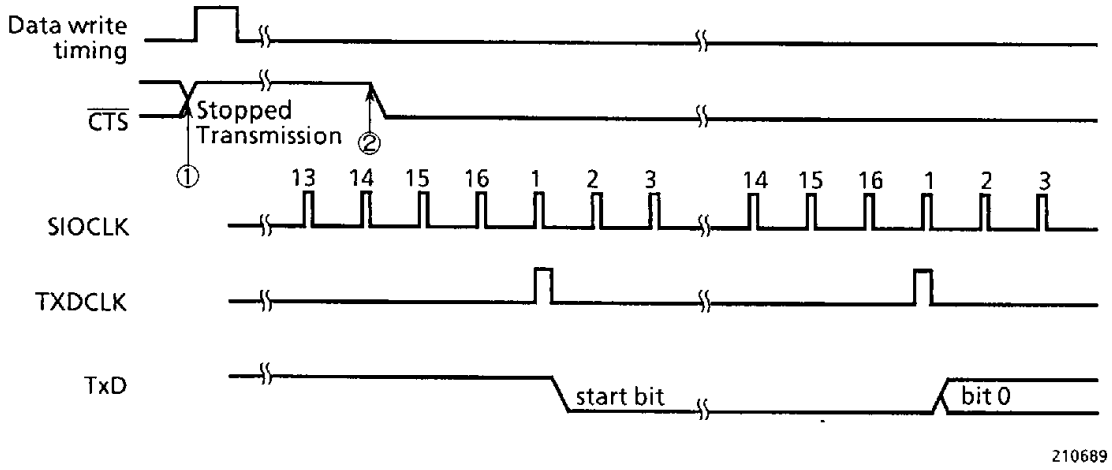


Figure 3.8 (8) Hand-shake Function



Note : In case of 8-bit asynchronous communication (UART), the last bit is the bit 7 in the non-parity mode, and the parity bit in the parity-added mode.

Figure 3.8 (9) Timing Chart of $\overline{RTS}$ (request to send) Signal



Note : ① A Rise of the $\overline{CTS}$ signal during the data transmission halts the transmission of the next data after the current data transmission.
② The transmission is restarted from the first fall of TXDCLK after a fall of the $\overline{CTS}$ signal.

Figure 3.8 (10) Hand-shake by $\overline{CTS}$ (clear to send) Signal

⑧ Transmission buffer

The transmission buffer SCBUF shifts out the data written by the CPU from the LSB as based on the shift clock TXDSFT (Same period as TXDCLK) generated by the transmission control unit. When all bits are shifted out, the transmission buffer becomes empty, generating the interrupt INTTX.

⑨ Parity control circuit

Setting the serial channel control register SCCR<PE> to "1" allows the addition of a parity bit in transmitting/receiving data, only in the 7-bit UART or 8-bit UART mode. Either even or odd parity can be selected by the SCCR<EVEN> register.

In the transmission mode, the parity is automatically generated as based on the data written into the transmission buffer SCBUF, storing into the SCBUF<TB7> in the 7-bit UART mode or into <TB8> in the 8-bit UART mode for transmission. <PE> and <EVEN> should be designated before writing data into the transmission buffer.

In the receiving mode, the receiving data is shifted into the receiving buffer 1 and transferred to the receiving buffer 2 (SUBUF). The parity is generated from the data in the receiving buffer 2. A parity error is detected and the SCCR<PERR> flag is set if the parity status mismatches the SCBUF<RB7> in the 7-bit UART mode or <RB8> in the 8-bit UART mode.

⑩ Error flag

There error flags are prepared to increase the reliability of received data.

1) Overrun error (SCCR<OERR>)

Overrun error occurs if all the bits of the next data are received by the receiving buffer 1 while valid data are still stored in the receiving buffer 2 (SCBUF).

2) Parity error (SCCR<PERR>)

The parity generated from the data that is transferred to the receiving buffer 2 (SCBUF) is compared with the parity bit received from the RxD terminal. Parity error occurs if they are not equal.

3) Framing error (SCCR<FERR>)

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

⑪ Generation 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	↑
Parity error timing	Center of last bit (Bit 8)	Center of last bit (Parity Bit)	↑
Over-run error timing	Center of last bit (Bit 8)	Center of last bit (Parity Bit)	↑

210689

Note: The occurrence of a framing error is delayed until after interruption. Therefore, to check for framing error during interrupt operation, an additional operation, such as waiting for 1 bit time, becomes necessary.

Transmitting

mode	9 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt timing	Just before the stop bit	←	←

210689

2) I/O interface mode

Interrupt timing of receiving	Just after the last SCLK rising ⌐
Interrupt timing of transmitting	↑

210689

3.8.3 Operation

(1) Mode 0 (I/O Interface Mode)

This mode is used to increase the number of I/O pins of the TMP90C840A.

The TMP90C840A supplies the transmitting/receiving data and a synchronous clock (SCLK) to an external shift register.

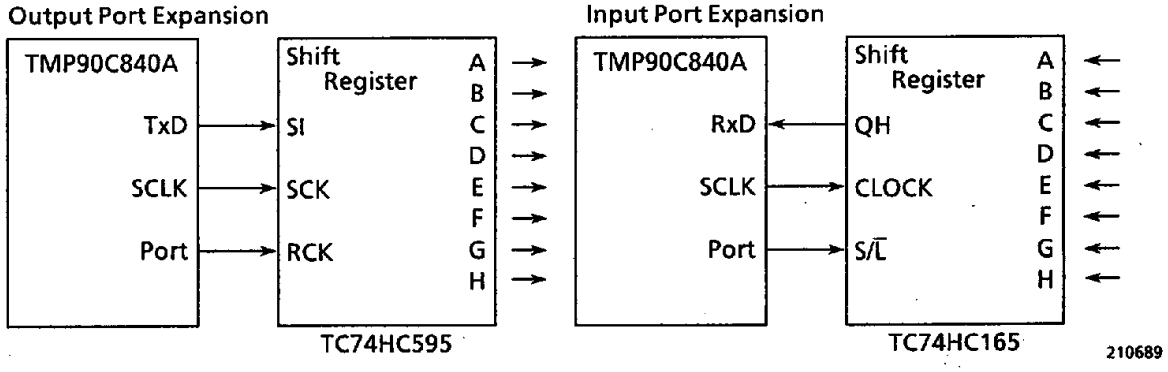


Figure 3.8 (11) I/O Interface Mode

① Transmission

Each time the CPU writes data into the transmission buffer, 8-bit data are output from TxD pin. When all data are output, IRFH<IRFTX> is set, and the interrupt INTTX occurs.

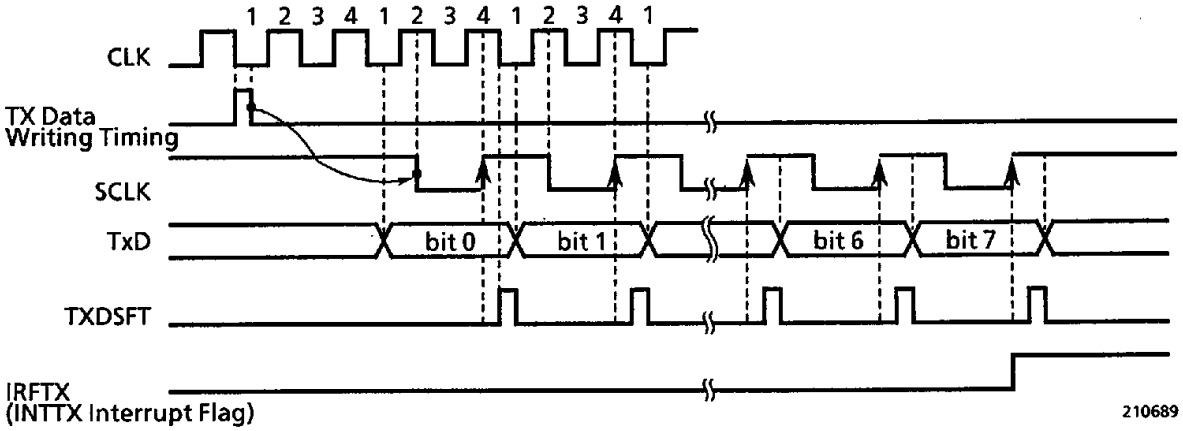


Figure 3.8 (12) Transmitting Operation (I/O Interface Mode)

Example : When transmitting data from P33 pin, the control registers should be set as described below.

```
P3CR  ← - - - - 1 1 0 0    Select P32 as the SCLK pin, and P33 as the TXD pin.
SCMOD ← X 0 0 0 0 0 X X    Set I/O Interface Mode.
INTEL ← - - - - - - - 1    Enable INTTX interrupt.
SCBUF ← * * * * * * * *    Set data for transmission.
```

Note : × ; don't care -; no change

280890

② Receiving

Each time the CPU reads the receiving data and clears the receiving interrupt flag IRFH<IRFRX>, the next data are shifted into the receiving buffer 1. When 8-bit data are received, the data are transferred to the receiving buffer 2 (SCBUF), which sets <IRFRX> and generates interrupt INTRX.

For receiving data, the receiving enable state is previously set SCMOD<RXE> = 1.

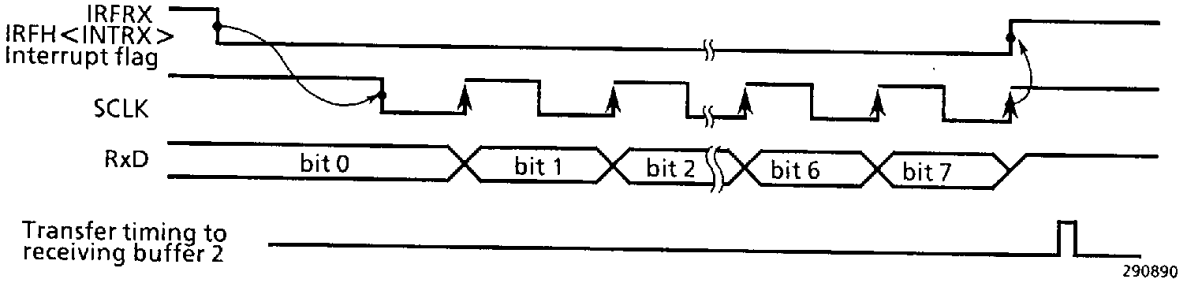


Figure 3.8 (13) Receiving Operation (I/O Interface Mode)

Example: When receiving from P30 pin, the control registers should be set as described below.

P3CR ← - - - - 1 1 0 1	Select 32 as the SCLK pin, and P30 as the RXD pin.
SCMOD ← X 0 0 0 0 0 X X	Set I/O Interface Mode.
INTEL ← - - - - - - - 1 -	Enable INTRX interrupt.
SCMOD ← - - 1 - - - - -	Set RXE to "1".

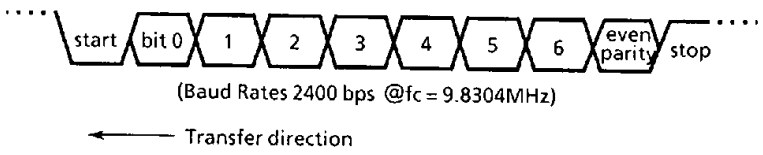
Note: × ; don't care - ; no change 280890

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

The 7-bit UART mode is selected by setting the serial channel mode register SCMOD<SM0, 1> to "01".

This mode allows the addition of a parity bit, which is enabled or disabled by the serial channel control register SCCR<PE>. When <PE> = 1 (enable), even or odd parity can be selected by SCCR<EVEN>.

Example: When transmitting data with the following format, the control registers should be set as described below.



280890

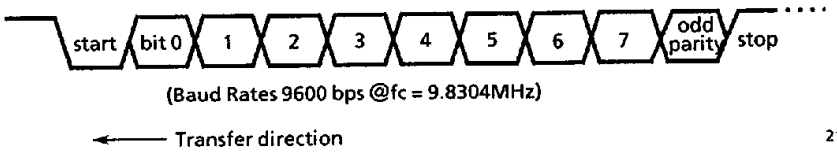
P3CR ← - - - - 0 1 - -	Select P32 as the TxD pin.
SCMOD ← X 0 - X 0 1 1 1	Set the transfer speed at 2,400 bps in the 7-bit UART mode.
TRUN ← 1 0 1 - - - - -	
SCCR ← X 1 1 X X X X 0	Add an even parity.
INTEL ← - - - - - - - 1	Enable INTTX interrupt.
SCBUF ← * * * * * * * *	Set data for transmission.

(Note) × ; don't care - ; no change

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

The 8-bit UART mode is selected by setting SCMOD<SM1, 0> to "10". This mode also allows the addition of a parity bit, as enabled or disabled by SCCR<PE>. When PE = 1 (enable), even or odd parity can be selected by SCCR<EVEN>.

Example: When receiving data with the following format, the control registers should be set as described below.



Main setting:

```
P3CR  ← - - - - - - - 0 1      Select P30 as the RxD pin.
SCCR  ← X 0 1 X X X X 0        Add an odd parity.
TRUN  ← 1 1 1 - - - - -  }     Set the transfer speed at 9,600 bps in
SCMOD ← - 0 1 X 1 0 1 1  }     the 8-bit UART mode.
INTEL ← - - - - - - - 1 -      Enable INTTX interrupt.
```

INTRX processing:

```
Acc  ← SCCR Λ 00011100         Check errors.
if Acc ≠ 0 then error
Acc  ← SCBUF                   Read out the received data.
```

(Note) ×: Don't care -: No change

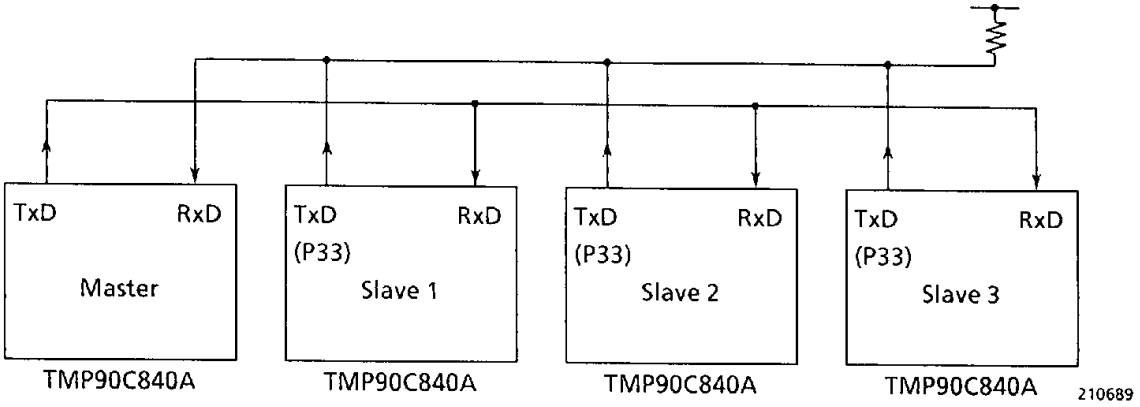
(4) Mode 3 (9-bit UART Mode)

The 9-bit UART mode is selected by setting SCMOD<SM1, 0> = "11". The addition of a parity bit is disabled in this mode.

The MSB (9th bit) is written into SCMOD<TB8> for transmission, and into SCCR<RB8> for receiving. Writing into or reading from the buffer must begin with the MSB (9th bit) followed by SCBUF.

Wake-up function

In the 9-bit UART mode, setting SCMOD<WU> to "1" allows the wake-up operation as the slave controllers. The interrupt INTRX occurs only when SCCR<RB8> = 1.



Note : For the wake-up operation, P33 should be always selected as the TxD pin of the slave controllers, and put in the open drain output mode.

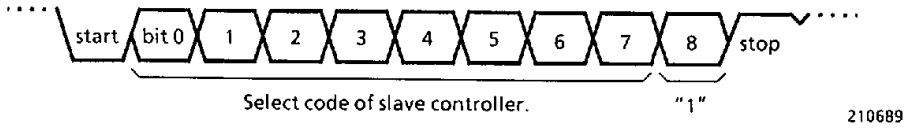
Figure 3.8 (14) Serial Link Using Wake-up Function

Protocol

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

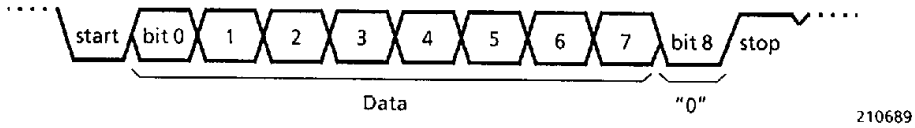
② Set the SCMOD<WU> bit of each slave controller to "1" to enable data receiving.

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



④ Each slave controller receives the above frame, and clears the 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 <WU> bit is cleared to "0") with setting the MSB (bit 8) <TB8> to "0".

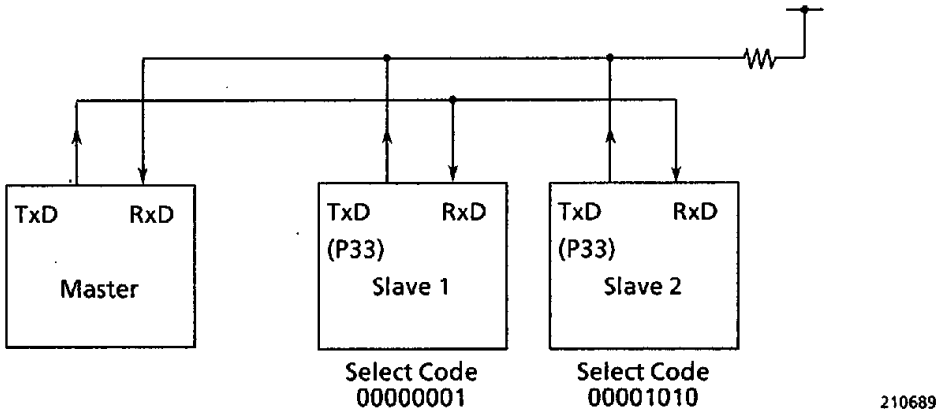


⑥ The other slave controllers (with the SCMOD<WU> bit remaining at "1") ignore the receiving data because their MSBs SCCR<RB8> are set to "0" to disable the interrupt INTRX.

When the <WU> bit is cleared to "0", the interrupt INTRX occurs, making it possible to read the receiving data.

The slave controllers (WU=0) transmits data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

Example: Link two slave controllers serially with the master controller, and use the internal clock ϕ1 (fc/2) as the transfer clock.



- Set the master control

Main

```
P3CR  ← - - - 0 0 1 1 0    Select P32 as TxD pin and P31 as RxD pin.
INTEL ← - - - - - - 1 1    Enable INTRX and INTTX.
SCCR  ← X X X X X X X 0    Disable the hand-shake function.
SCMOD ← 1 0 1 0 1 1 1 0    Select φ1 (fc/2) as the transfer clock in the 9-bit UART mode.
SCBUF ← 0 0 0 0 0 0 0 1    Set the select code for the slave controller 1.
```

INTTX interrupt

```
SCMOD ← 0 - - - - - - -    Set SCMOD<TB8> to "0".
SCBUF ← * * * * * * * *    Set data for transmission.
```

- Set the slave controller 2

Main

```
P3CR  ← - - - 1 1 0 1 0    Select P33 as TxD pin and P31 as RxD pin .
INTEL ← - - - - - - 1 1    Enable INTRX and INTTX.
SCCR  ← X X X X X X X 0    Disable the hand-shake function.
SCMOD ← 0 0 1 1 1 1 1 0    Set <WU> to 1 in the 9-bit UART mode
                           (transfer clock : φ1(fc/2) ).
```

INTRX interrupt

```
Acc  ← SCBUF
if Acc = Select code
then SCMOD ← - - - 0 - - - -    Clear <WU> to "0".
```

(Note) ×: Don't care -: No change

3.9 Analog/Digital Converter (A/D Converter)

The TMP90C840A incorporates a high-speed A/D converter with six analog input channels that features 8-bit sequential comparison.

Figure 3.9 (1) is a block diagram of A/D converter. The 6-channel analog input pin (AN5 ~AN0) is also used as the input port P5.

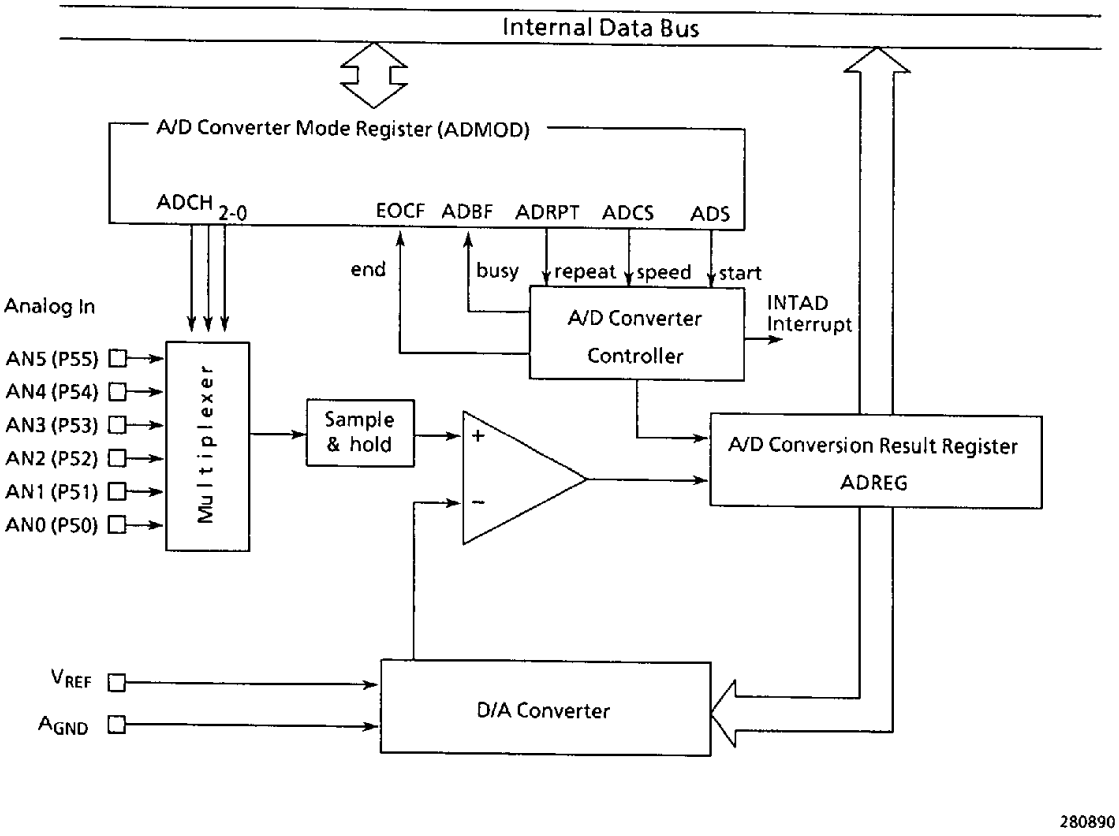


Figure 3.9 (1) Block Diagram of A/D Converter

3.9.1 Control Registers

Fig. 3.9 (2) shows the registers related to the A/D converter.

The A/D converter is controlled by the A/D converter mode register ADMOD, and the result of A/D conversion is stored in the A/D conversion value storing register ADREG.

A/D Converter Mode Register

		7	6	5	4	3	2	1	0
ADMOD (FFEFH)	bit Symbol	ADRPT	EOCF	ADBF	ADCS	ADS	ADCH2	ADCH1	ADCH0
	Read/Write	R/W	R				R/W		
	Resetting Value	1	0	0	0	0	0	0	0
Prohibit Read Modify Write	Function	1: single mode 0: repeat mode	1 : end 0 : not end	1 : busy 0 : not busy	1 : 190 states 0 : 95 states	1 : A/D start 0 : don't care	Analog Input Channel Select		

Analog Input Channel Select

000	AN0
001	AN1
010	AN2
011	AN3
100	AN4
101	AN5
110	Can not be used
111	

A/D Conversion Start

0	—
1	Start A/D Conversion

Note : Always "0" when data is read out.

A/D Conversion Speed Selection

0	95 states (15.2μs)
1	190 states (30.4μs)

Note : Conversion speed calculated at fc = 12.5MHz

A/D Conversion busy flag

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

End of Conversion Flag

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

* <EOCF> is reset to zero by reset operation, reading ADREG, or restarting A/D conversion.

A/D conversion repeat flag

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

Note : <ADBF> and <EOCF> cannot be used in the A/D conversion repeat mode.

A/D Conversion Result Register

		7	6	5	4	3	2	1	0
ADREG (FFEEH)	bit Symbol								
	Read/Write	R							
	Resetting Value	Undefined							
Prohibit Read Modify Write	Function	The results of A/D conversion are stored.							

200990

Figure 3.9 (2) Registers for A/D Conversion

3.9.2 Operation

(1) Analog Reference Voltage

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

The reference voltage between VREF and AGND is divided by 256 by ladder resistance, and compared with the analog input voltage for A/D conversion.

(2) Analog Input Channels

For the A/D conversion, one of the six analog input channels (AN0 (P50) to AN5 (P55)) is selected by the register ADMOD<ADCH2, 1, 0>.

The analog input channel select registers ADMOD<ADCH2, 1, 0> is initialized to "000" by resetting, whereby the AN0 pin is selected.

The pins not used for the analog input can be used for ordinary input P5 .

(3) Selection of A/D Conversion Speed

Normally, the A/D converter is used in the high-speed conversion mode that completes the operation in 95 states (15.2 μs @fc = 12.5 MHz).

When used in the low-speed conversion mode (do not use for new programs), set the speed select register ADMOD<ADCS> to "1" to set the conversion speed to the 190 state.

It is not possible to change the conversion speed and start conversion at the same time; therefore, first change the conversion speed and then start the conversion.

ADMOD<ADCS> is initialized to "0" by resetting, by which the A/D converter turns to the high-speed conversion mode.

(4) Starting A/D Conversion

The A/D conversion is started by writing "1" into the start register ADMOD<ADS>. When the A/D conversion is started, the busy flag ADMOD<ADBF> is set to "1", indicating that the conversion is in progress.

A/D conversion may stop when restarted while the ADMOD<ADBF> is set to "1"; therefore, always make sure that the ADMOD<ADBF> is set to "0" before restarting conversion.

(5) Setting A/D Conversion Repeat

The repeat flag ADMOD<ADRPT> can be used to select repeat mode or single mode A/D conversion.

In the repeat mode, A/D conversion is restarted after the completion of one A/D conversion cycle.

Reset operation initializes <ADRPT> to "1", making the system into one-time conversion mode. To use this in repeat mode, write zero in <ADRPT>.

Write "1" to ADMOD<ADRPT> to end the repeat mode. The repeat mode will be exited as soon as the conversion in progress is completed.

Do not use A/D interrupts in the repeat mode.

(6) A/D Conversion End and Interrupt

- A/D conversion single mode

When the A/D conversion is completed, the end of conversion flag ADMOD <EOCF> is set to "1", indicating that the A/D conversion is completed. Then the <ADBF> flag is cleared to "0" and the A/D conversion completion interrupt INTAD is generated.

INTAD (interrupt by the A/D converter) is controlled by the interrupt enable flag INTEL<IET2>, also used for INTT2 (interrupt by Timer 2). Either INTAD or INTT2 is selected by the A/D interrupt select register INTEH<ADIS>. To generate INTAD interrupt, both <IET2> and <ADIS> should be set to "1".

Both INTAD and INTT2 jump to the same vector address (0040H), but can be distinguished by the <ADIS>.

The INTAD interrupt Request Flip-flop can be cleared only by resetting or reading the register that stores A/D conversion value, and cannot be cleared by an instruction. A change in the interrupt source (between INTAD and INTT2) automatically clears the interrupt request flag.

- A/D conversion repeat mode

The A/D conversion completion interrupt INTAD cannot be used when the repeat mode is set. <ADBF> and <EOCF> cannot be used during A/D conversion in repeat mode.

To end the repeat mode operation, write "1" to <ADRPT>. The repeat mode will end when the current conversion is completed.

(7) Reading A/D Conversion Results

The results of A/D conversion are stored into the A/D conversion result register ADREG.

By reading the contents of the ADREG, ADMOD<EOCF> is cleared to "0".

When the contents of the ADREG is read while the A/D conversion is performed, the reading data is undefined. (In A/D conversion repeat mode, the current conversion value is always latched to ADREG, which can be read at any time except during the initial A/D conversion.)

A/D conversion is stopped by a HALT instruction except in the RUN mode, by which the A/D conversion results become undefined.

In the RUN mode, A/D conversion is not stopped by a HALT instruction.

Example: ①Analog input voltage to the AN3 pin is converted to a digital value in the high-speed conversion mode (95 states), and the result is stored into the memory address FF10H by using A/D interrupt INTAD routine.

Main setting

```
INTEH ← - - - - 1 - - -      Set <ADIS> to "1"
INTEL ← 1 - - - - - - -      Enable INTT2
ADMOD ← 1 x x 0 1 0 1 1      Select AN3 as the analog input channel, and start the
                             A/D conversion in the high-speed conversion mode.
```

A/D interrupt routine

```
A ← ADREG                    Load the contents of ADREG into the accumulator.
                             Store the accumulator value into the memory address
(FF10H) ← A                  FF10H.
```

Example: ②Analog input voltage to the AN2 pin is converted to a digital value in the high-speed conversion mode (95 states), and the result is loaded into the accumulator when the end of conversion is detected by the EOCF flag.

```
ADMOD ← 1 X X 0 1 0 1 0      Select AN2 as the analog input channel, and start
                             the A/D conversion in the high-speed conversion
loop:                        mode.
  if EOCF = 0 then loop
              else A ← ADREG
```

(Note) ×: Don't care -: No change

3.10 Watchdog Timers (Runaway Detecting Timer)

When the malfunction (runaway) of the CPU occurs due to any cause such as noise, the watchdog timer (WDT) detects it to return to the normal state. When WDT has detected malfunction, a non-maskable interrupt is generated to indicate it to the CPU.

3.10.1 Architecture

Figure 3.10 (1) is a block diagram of the watchdog timer (WDT).

The watchdog timer consists of a 20-stage binary counter (input clock: ϕ @fc/2), a flip-flop that disables/enables the selector, a selector that selects one of the four output clocks generated from the binary counter, and two control registers.

The watchdog timer generates INTWD (watchdog timer interrupt) after a time specified by the register WDMOD<WDTP1, 0>. The binary counter for the watchdog timer is cleared to "0" by software (instruction) before the interrupt occurs. If the CPU caused a malfunction (runaway) for reason such as noise and fails to execute the instruction to clear the watchdog timer, the counter will overflow and the watchdog timer interrupt INTWD occurs. The CPU detects the malfunction (runaway) by this interrupt and is possible to be recovered to the normal state by the software for malfunction.

The watchdog timer starts its operation as soon as the reset state is cleared.

The watchdog timer stops its operation only in the STOP mode. When the STOP mode is released, the watchdog timer starts its operation after a specified warming-up time.

In the other standby mode (IDLE 1, IDLE 2 or RUN modes), the watchdog timer is enabled. However, the function can be disabled before entering any of these modes.

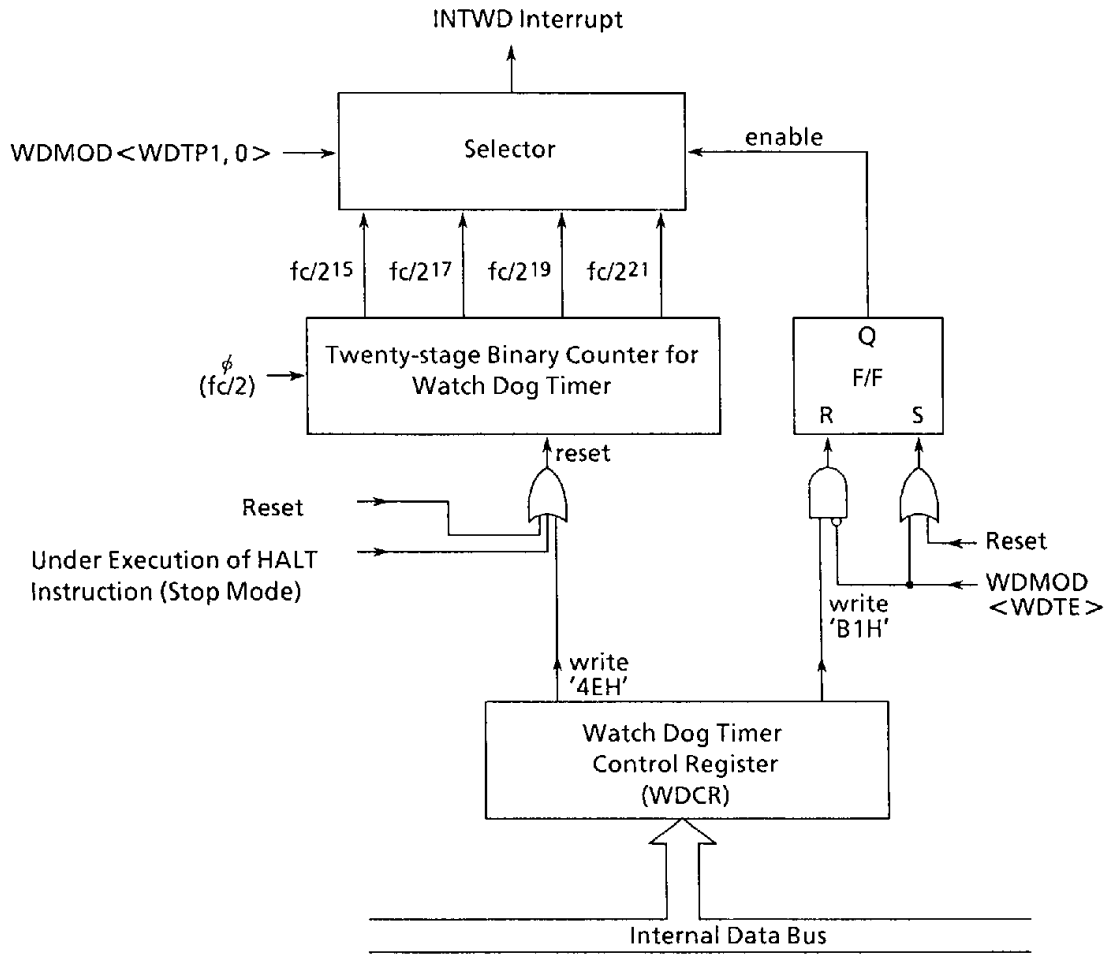


Figure 3.10 (1) Block Diagram of Watchdog Timer

3.10.2 Control Registers

WDT is controlled by two control registers (WDMOD and WDCR).

The watchdog timer (WDT) is controlled by two control registers (WDMOD and WDCR). Fig. 3.9 (2) shows the registers related to WDT.

Watchdog Timer Mode Register

WDMOD (FFD2H)	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	EXF	DRIVE
Read/Write	R/W						R	R/W
Resetting Value	1	0	0	0	0	0	Undefined	0
Function	1: WDT Enable	WDT detecting period 00: 2^{14}/fc 01: 2^{16}/fc 10: 2^{18}/fc 11: 2^{20}/fc		Warming up time 0: 2^{14}/fc 1: 2^{16}/fc	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Invert each time EXX instruction is executed	1: to drive pin in STOP mode.

Prohibit Read Modify Write

See "3.4.4 STOP Mode".

Invert each time EXX instruction is executed

Select standby mode by HALT instruction

00	RUN Mode
01	STOP Mode
10	IDLE1 Mode
11	IDLE2 Mode

Select warming-up time when returned from STOP mode.

0	2^{14}/fc (Approx 1.6ms)
1	2^{16}/fc (Approx 6.6ms)

Note: fc=10MHz

Select detecting period of watchdog timer (WDTP)

00	2^{14}/fc (Approx 1.6ms)
01	2^{16}/fc (Approx 6.6ms)
10	2^{18}/fc (Approx 26.2ms)
11	2^{20}/fc (Approx 105ms)

Note: fc=10MHz

Watchdog timer enable/disable control

0	Disable (Interactive with WDCR) (Note)
1	Enable

Note: To disable, it is necessary to also write the disable code to the WDCR register. Disabling is not possible by writing to this register alone.

Watchdog timer control register

WDCR (FFD3H)	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	W							
Resetting Value	—							
Function	B1H: WDT Disable code				4EH: WDT Clear code			

Disable/clear watchdog timer

B1H	Disable code
4EH	Clear code
Other	—

200990

Figure 3.10 (2) Registers for Watchdog Timer

3.10.3 Operation

(1) Watchdog Timer Mode Register (WDMOD)

① The register of detecting period of watchdog timer: WDMOD<WDTP1, 0>

The WDT interrupt period is set by this 2-bit registers. <WDTP1, 0> is initialized to "00" by resetting, providing the initial set value of 2^{14}/fc (s.) (approx. 8,192 states).

Here, the interrupt vector address of INTWD is "0020H".

② The WDT enable/disable control register: WDMOD<WDTE>

<WDTE> is initialized to "1" by resetting, which enables the watchdog timer function.

To disable the function, the bit should be cleared to "0" and the disable code "B1H" should be written into the Watchdog timer control register WDCR. By using this dual procedure, it becomes hard to disable the WDT even if the malfunction occurs.

The disable state can be returned to the enable state easily by setting <WDTE> to "1".

(2) Watchdog Timer Control Register (WDCR)

This is the register to control the disabling and clearing of the watchdog timer.

① Disabling the watchdog timer

To disable the watchdog timer, write "0" in WDMOD<WDTE>, and also write B1H in WDCR.

② Clearing the watchdog timer

To clear the watchdog timer, write 4EH in WDCR.

During reset operation, or while the HALT command is being executed after being set to the STOP mode, the CLEAR signal already exists in the watchdog timer, and the timer is already reset.

Example: ① Clear the Watchdog timer.

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

② Set 2^{16}/fc for the detecting time of watchdog timer.

```
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 Select the IDLE 2 mode.

```
WDMOD  ← 0 - - - 1 1 X X      Disable WDT and set IDLE 2 mode
WDCR   ← 1 0 1 1 0 0 0 1
```

Execute HALT instruction . Falling into the standby mode.

⑤ Select the STOP mode (Warming-up time: 2^{16}/fc)

```
WDMOD  ← - - - 1 0 1 X X      Select STOP mode
```

Execute HALT instruction. Falling into the standby mode.

4. ELECTRICAL CHARACTERISTICS

TMP90C840AN/TMP90C840AF/TMP90C841AN/TMP90C841AF

4.1 Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit
V_{CC}	Supply voltage	−0.5 ~ +7	V
V_{IN}	Input voltage	−0.5 ~ Vcc + 0.5	V
P_D	Power dissipation (Ta = 85°C)	F 500	mW
		N 600	
T_{SOLDER}	Soldering temperature (10 s)	260	°C
T_{STG}	Storage temperature	−65 ~ 150	°C
T_{OPR}	Operating temperature	−40 ~ 85	°C

210689

4.2 DC Characteristics

Vcc = 5V ± 10% TA = −40~85°C (1~10MHz)
TA = −20~70°C (1~12.5MHz)
Typical Values are for TA = 25°C V_{CC} = 5V.

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (P0)	−0.3	0.2Vcc − 0.1	V	
V_{IL1}	P1, P2, P3, P4, P5, P6, P7, P8	−0.3	0.3Vcc	V	
V_{IL2}	$\overline{RESET}$, INT0 (P80), $\overline{NMI}$	−0.3	0.25Vcc	V	
V_{IL3}	$\overline{EA}$	−0.3	0.3	V	
V_{IL4}	X1	−0.3	0.2Vcc	V	
V_{IH}	Input High Voltage (P0)	0.2Vcc + 1.1	Vcc + 0.3	V	
V_{IH1}	P1, P2, P3, P4, P5, P6, P7, P8	0.7Vcc	Vcc + 0.3	V	
V_{IH2}	$\overline{RESET}$, INT0 (P80), $\overline{NMI}$	0.75Vcc	Vcc + 0.3	V	
V_{IH3}	$\overline{EA}$	Vcc − 0.3	Vcc + 0.3	V	
V_{IH4}	X1	0.8Vcc	Vcc + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 1.6mA
V_{OH} V_{OH1} V_{OH2}	Output High Voltage	2.4 0.75Vcc 0.9Vcc		V V V	I_{OH} = −400μA I_{OH} = −100μA I_{OH} = −20μA
I_{DAR}	Darlington Drive Current (8 I/O pins) (Note)	−1.0	−3.5	mA	V_{EXT} = 1.5V R_{EXT} = 1.1 kΩ
I_{LI}	Input Leakage Current	0.02 (Typ)	±5	μA	0.0 ≦ Vin ≦ Vcc
I_{LO}	Output Leakage Current	0.05 (Typ)	±10	μA	0.2 ≦ Vin ≦ Vcc − 0.2
I_{CC}	Operating Current (RUN) Idle 1 Idle 2	20 (Typ) 1.5 (Typ) 8 (Typ)	40 5 15	mA mA mA	tosc = 10MHz (25%Up @12.5MHz)
	STOP (TA = −40~85°C) STOP (TA = 0~50°C)	0.2 (Typ)	50 10	μA μA	0.2 ≦ Vin ≦ Vcc − 0.2
V_{STOP}	Power Down Voltage (@STOP)	2 RAM BACK UP	6	V	V_{IL2} = 0.2Vcc, V_{IH2} = 0.8Vcc
R_{RST}	$\overline{RESET}$ Pull Up Registor	50	150	kΩ	
CIO	Pin Capacitance		10	pF	testfreq = 1MHz
V_{TH}	Schmitt width $\overline{RESET}$, $\overline{NMI}$, INT0	0.4	1.0 (Typ)	V	

Note : I_{DAR} is guaranteed for a total of up to 8 optional ports. 280890

4.3 AC Characteristics

Vcc = 5V ± 10% TA = −40~85°C (1~10MHz)
CL = 50pF TA = −20~70°C (1~12.5MHz)

Symbol	Parameter	Variable		10MHz Clock		12.5MHz Clock		Unit
		Min	Max	Min	Max	Min	Max	
t_{OSC}	OSC. Period = x	80	1000	100		80		ns
t_{CYC}	CLK Period	4x	4x	400		320		ns
t_{WL}	CLK Low width	2x − 40		160		120		ns
t_{WH}	CLK High width	2x − 40		160		120		ns
t_{AC}	Address Setup to $\overline{RD}$, $\overline{WR}$	x − 45		55		35		ns
t_{RR}	$\overline{RD}$ Low width	2.5x − 40		210		160		ns
t_{CA}	Address Hold Time After $\overline{RD}$, $\overline{WR}$	0.5x − 30		20		10		ns
t_{AD}	Address to Valid Data In		3.5x − 95		255		185	ns
t_{RD}	$\overline{RD}$ to Valid Data In		2.5x − 80		170		120	ns
t_{HR}	Input Data Hold After $\overline{RD}$	0		0		0		ns
t_{WW}	$\overline{WR}$ Low width	2.5x − 40		210		160		ns
t_{DW}	Data Setup to $\overline{WR}$	2x − 50		150		110		ns
t_{WD}	Data Hold After $\overline{WR}$	30	90	30	90	30	90	ns
t_{CWA}	$\overline{RD}$, $\overline{WR}$ to Valid $\overline{WAIT}$		1.5x − 100		50		20	ns
t_{AWA}	Address to Vaild $\overline{WAIT}$		2.5x − 130		120		70	ns
t_{WAS}	$\overline{WAIT}$ Setup to CLK	70		70		70		ns
t_{WAH}	$\overline{WAIT}$ Hold After CLK	0		0		0		ns
t_{RV}	$\overline{RD}$, $\overline{WR}$ Recovery Time	1.5x − 35		115		85		ns
t_{CPW}	CLK to Port Data Output		x + 200		300		280	ns
t_{PRC}	Port Data Setup to CLK	200		200		200		ns
t_{CPR}	Port Data Hold After CLK	100		100		100		ns
t_{CHCL}	$\overline{RD}$ / $\overline{WR}$ Hold After CLK	x − 60		40		20		ns
t_{CLC}	$\overline{RD}$ / $\overline{WR}$ Setup to CLK	1.5x − 50		100		70		ns
t_{CLHA}	Address Hold After CLK	1.5x − 80		70		40		ns
t_{ACL}	Address Setup to CLK	2.5x − 80		170		120		ns
t_{CLD}	Data Setup to CLK	x − 50		50		30		ns

280890

- AC output level High 2.2V/Low 0.8V
- AC input level High 2.4V/Low 0.45V (D0 − D7)
  High 0.8Vcc/Low 0.2Vcc (excluding D0 − D7)

4.4 A/D Conversion Characteristics

Vcc = 5V ± 10% TA = −40~85°C (1~10MHz)
TA = −20~70°C (1~12.5MHz)

Symbol	Parameter	Min	Typ	Max	Unit
V_{REF}	Analog reference voltage	Vcc − 1.5	Vcc	Vcc	V
A_{GND}	Analog reference voltage	V_{SS}	V_{SS}	Vss	
V_{AIN}	Allowable analog input voltage	V_{SS}		Vcc	
I_{REF}	Supply current for analog reference voltge		0.5	1.0	mA
Error	Total error (TA = 25°C, $V_{CC} = V_{REF} = 5.0V$)		1.0		LSB
	Total error			2.5	

280890

4.5 Zero- Cross Characteristics

Vcc = 5V ± 10% TA = −40~85°C (1~10MHz)
TA = −20~70°C (1~12.5MHz)

Symbol	Parameter	Condition	Min	Max	Unit
Vzx	Zero- cross detection input	AC coupling C = 0.1μF	1	1.8	VAC p − p
Azx	Zero- cross accuracy	50/60Hz sine wave		135	mV
Fzx	Zero- cross detection input frequency		0.04	1	kHz

280890

4.6 Serial Channel Timing – I/O Interface Mode

Vcc = 5V ± 10% TA = −40~85°C (1~10MHz)
CL = 50pF TA = −20~70°C (1~12.5MHz)

Symbol	Parameter	Variable		10MHz Clock		12.5MHz Clock		Units
		Min	Max	Min	Max	Min	Max	
t_{SCY}	Serial Port Clock Cycle Time	8x		800		640		ns
t_{OSS}	Output Data Setup SCLK Rising Edge	6x − 150		450		330		ns
t_{OHS}	Output Data Hold After SCLK Rising Edge	2x − 120		80		40		ns
t_{HSR}	Input Data Hold After SCLK Rising Edge	0		0		0		ns
t_{SRD}	SCLK Rising Edge to Input DATA Valid		6x − 150		450		330	ns

280890

4.7 16-bit Event Counter

Vcc = 5V ± 10% TA = −40~85°C (1~10MHz)
TA = −20~70°C (1~12.5MHz)

Symbol	Parameter	Variable		10MHz Clock		12.5MHz Clock		Units
		Min	Max	Min	Max	Min	Max	
t_{VCK}	TI4 clock cycle	8x + 100		900		740		ns
t_{VCKL}	TI4 Low clock pulse width	4x + 40		440		360		ns
t_{VCKH}	TI4 High clock pulse width	4x + 40		440		360		ns

280890

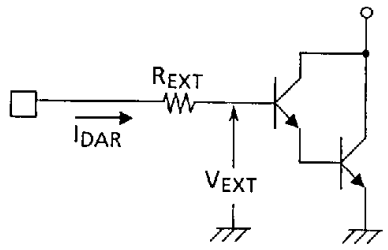
4.8 Interrupt Operation

Vcc = 5V ± 10% TA = −40~85℃ (1~10MHz)
TA = −20~70℃ (1~12.5MHz)

Symbol	Parameter	Variable		10MHz Clock		12.5MHz Clock		Units
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{NMI}$, INT0 Low level pulse width (⊔)	4x		400		320		ns
t_{INTAH}	$\overline{NMI}$, INT0 High level pulse width (⊓)	4x		400		320		ns
t_{INTBL}	INT1, INT2 Low level pulse width (⊔)	8x + 100		900		740		ns
t_{INTBH}	INT1, INT2 High level pulse width (⊓)	8x + 100		900		740		ns

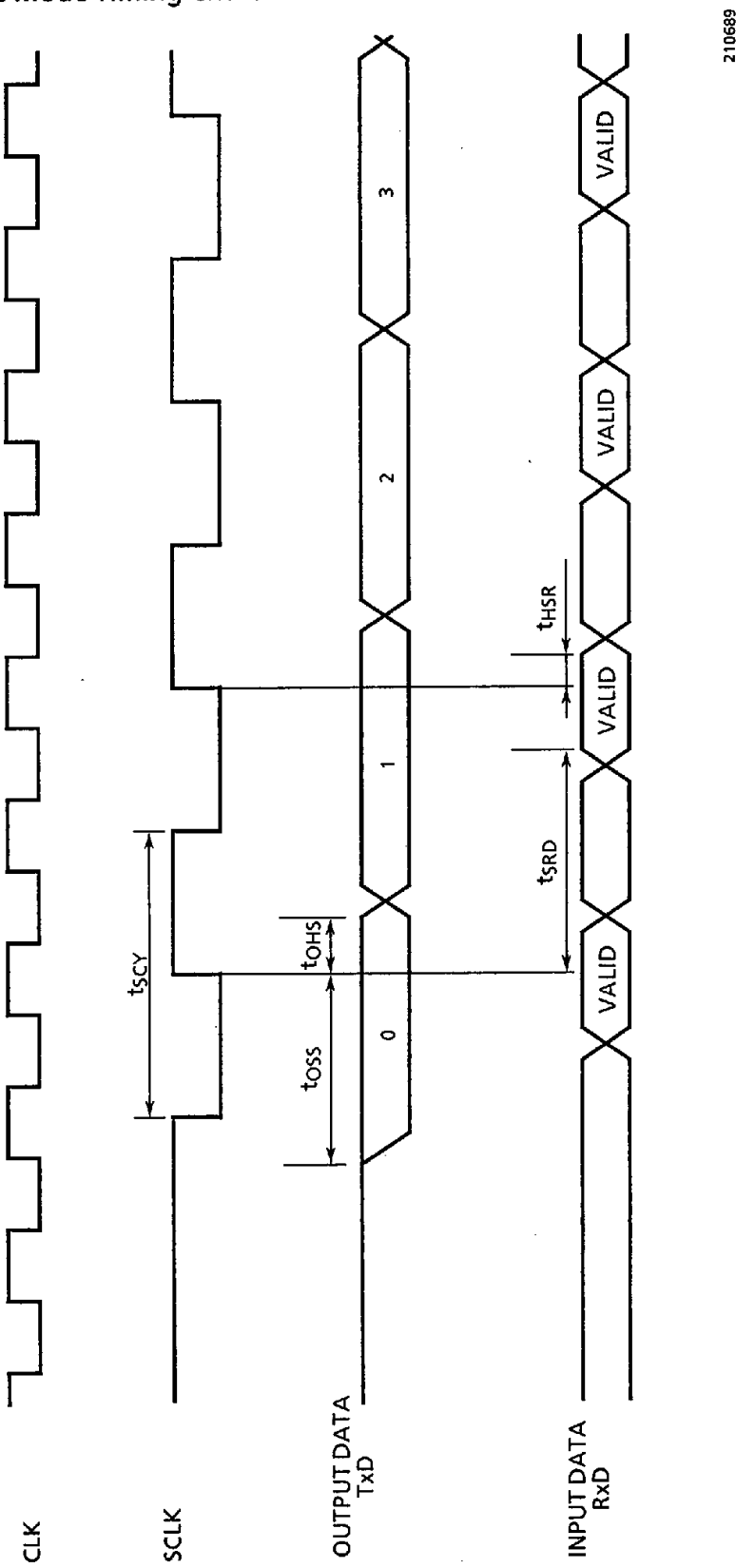
280890

(Reference) Definition of I_{DAR}



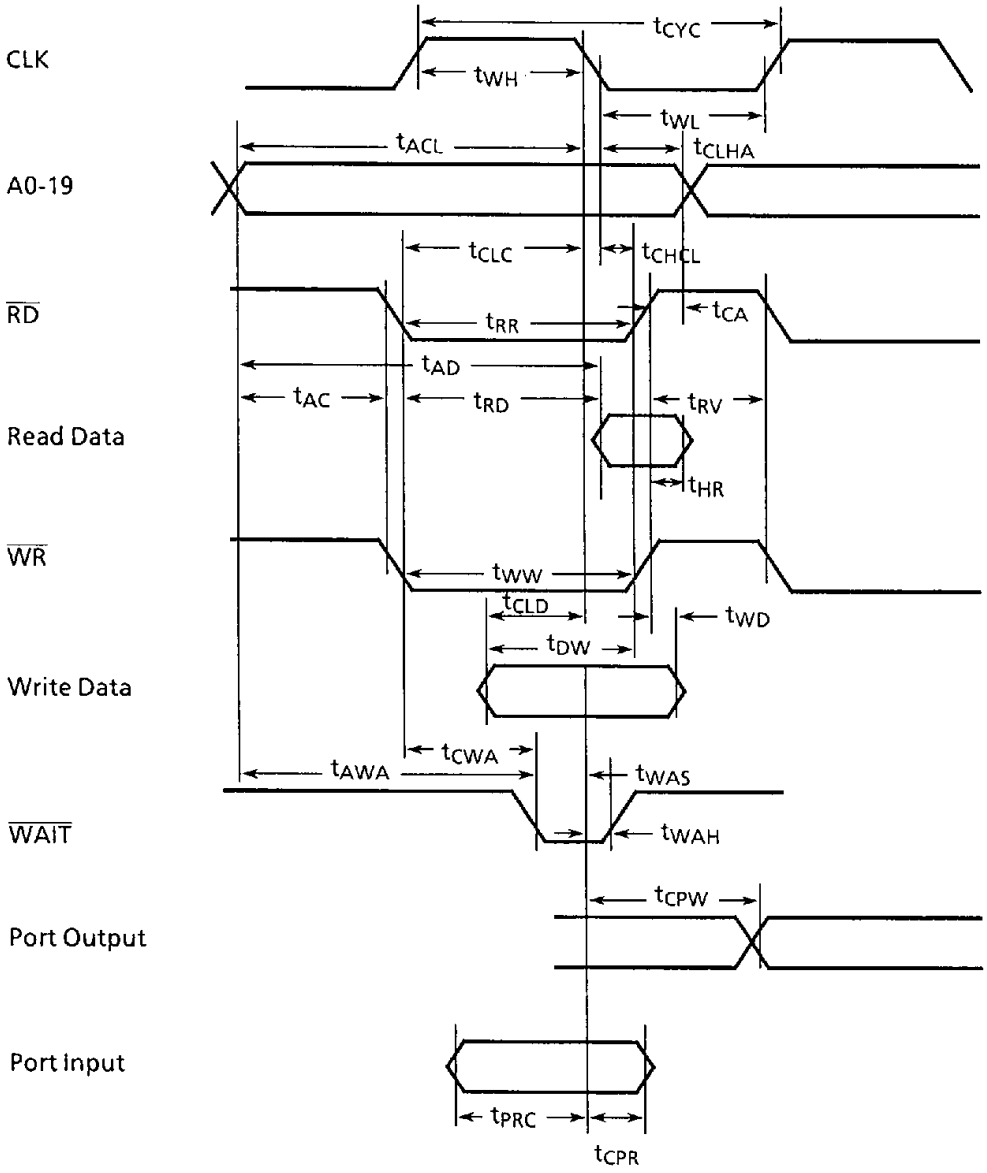
020289

4.9 I/O Interface Mode Timing Chart



TMP90C840A I/O Interface Mode Timing Waveforms

4.10 Timing Chart



210689

4.11 Typical Characteristics

Vcc = 5V, Ta = 25°C, unless otherwise noted.

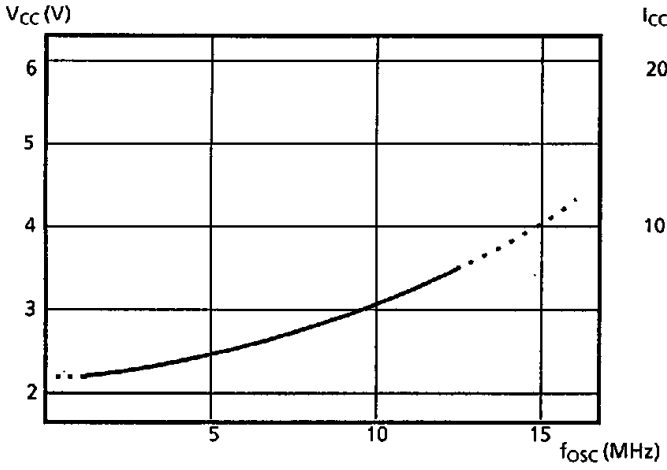


Figure 4.11 (1) V_{CC} – f_{OSC} TYPICAL CURVE

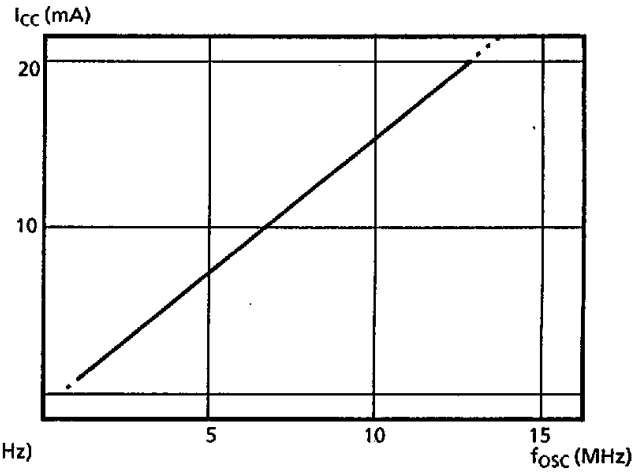


Figure 4.11 (2) f_{OSC} – I_{CC} TYPICAL CURVE

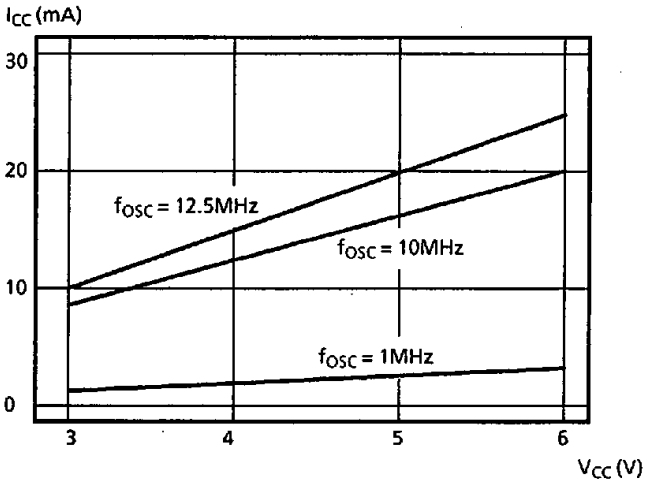


Figure 4.11 (3) I_{CC} – V_{CC} TYPICAL CURVE

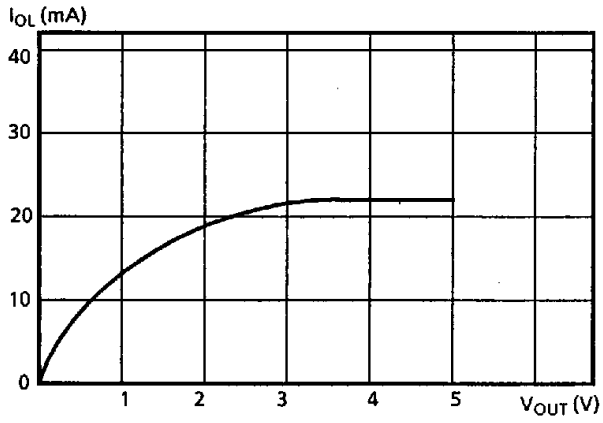


Figure 4.11 (4) V_{OUT} – I_{OL} TYPICAL CURVE

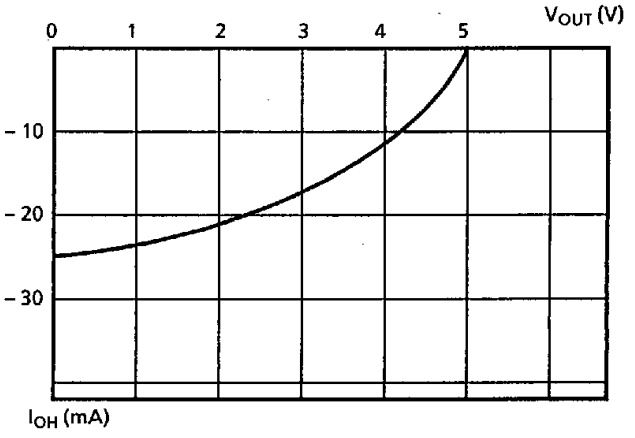


Figure 4.11 (5) V_{OUT} – I_{OH} TYPICAL CURVE

200990

5. TABLE OF SPECIAL FUNCTION REGISTERS

The special function registers include the I/O ports, peripheral control registers and bank registers (BX and BY) allocated to the 48-byte addresses from 0FFC0H to 0FFEFH.

(1) I/O port

(2) I/O port control

(3) Stepping motor port control

(4) Watchdog timer control

(5) Timer/event counter control

(6) Serial channel control

(7) A/D converter control

(8) Interrupt control

(9) Bank register

Format of table

Symbol	Name	Address	7	6			1	0	
									→bit Symbol
									→Read/Write
									→Resetting Value
									→Function

280890

(1) I/O Port

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
P0	Port 0	0FFC0H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Input mode							
P1	Port 1	0FFC1H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Input mode							
P2	Port 2	0FFC4H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Input mode							
P3	Port 3	0FFC6H	P37	P36	P35	P34	P33	P32	P31	P30
			R	R/W	R/W	R	R/W	R/W	R	R
			Input	1	1	Input	1	1	Input	Input
P4	Port 4	0FFC8H					P43	P42	P41	P40
							R/W			
							0	0	0	0
P5	Port 5	0FFCAH	0		P55	P54	P53	P52	P51	P50
					R					
			0		Input only					
					Shared with analog input pin (AN0-AN5)					
P6	Port 6	0FFCCH prohibit RMW	SA63	SA62	SA61	SA60	P63	P62	P61	P60
			W				R/W			
			Undefined				Input mode			
			Stepping motor control Port 0 Shifter alternate reg.				Shared with stepping motor control port 0 (M0)			
P7	Port 7	0FFCDH prohibit RMW	SA73	SA72	SA71	SA70	P73	P72	P71	P70
			W				R/W			
			Undefined				Input mode			
			Stepping motor control Port 1 Shifter alternate reg.				Shared with stepping motor control port 1 (M1)			
P8	Port 8	0FFD0H					P83	P82	P81	P80
							R/W	R	R	R
							0	Input mode		

210689

Note: Read/Write

R/W : Either read or write is possible.
R : Only read is possible.
W : Only write is possible.
prohibit RMW : Prohibit Read Modify Write (Prohibit RES/SET Instruction etc.)

(2) I/O Port Control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
P01CR (IRFL)	Port 0 / 1 Control Reg.	0FFC2H		IRF0	IRFT0	IRFT1		EXT	P1C	P0C
				R				W	W	W
				0	0	0		0	0	0
		prohibit RMW		Interrupt Request Flag 1 : Interrupt being requested				P1, P2 control 0:I/O Port 1:Address bus	P1 control 0:In 1:Out	P0 control 0:In 1:Out
P2CR	Port2 Control Reg.	0FFC5H	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
		prohibit RMW	0	0	0	0	0	0	0	0
			0: In 1:Out (I/O selected bit by bit)							
P3CR	Port3 Control Reg.	0FFC7H	WAITC1	WAITC0	RDE	ODE	TXDC1	TXDC0	RXDC1	RXDC0
			R/W		R/W	R/W	R/W		R/W	
			0	0	0	0	0	0	0	0
			Wait control 00 : 2state wait 01 : normal wait 10 : non wait 11 : reserved		RD control 0:$\overline{RD}$ for only external access 1:Always $\overline{RD}$	P33 control 0 :CMOS 1 :Open drain	P33 00 : Out 01 : Out 10 : TxD 11 : TxD	P32 Out TxD Out $\overline{RTS}$/ SCLK	P31 00 : In 01 : In 10 : RxD 11 : Not used	P30 In RxD In
P4CR	Port4 Control Reg.	0FFC9H					P43C	P42C	P41C	P40C
							W			
		prohibit RMW					0	0	0	0
							0 : Out (Port) 1 : Address output			
P67CR	Port6/7 Control Reg.	0FFCEH	P73C	P72C	P71C	P70C	P63C	P62C	P61C	P60C
			W				W			
		prohibit RMW	0	0	0	0	0	0	0	0
			0 : In		1 : Out		0 : In		1 : Out	
P8CR	Port8 Control Reg.	0FFD1H					P830C	ZCE2	ZCE1	EDGE
							W	W	W	W
							0	0	0	0
		prohibit RMW					P83 control 0:P83 1:TO3/ TO4	INT2/TI5 control 1:ZCD enable	* INT1/TI4 control 1:ZCD enable	INT0 control 0 : level 1 : ↑ edge

210689

Symbol in ( ) denotes another name. * : Refer T4MOD register of 16 Bit Timer.

(3) Stepping Motor Control Port Control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
SMMOD	Stepping Motor Mode Reg.	0FFCBH	—	SM7M0	P7OC1	P7OC0	—	SM6M0	P6OC1	P6OC0
				R/W	R/W			R/W	R/W	
				0	0	0		0	0	0
				0 : 1-step /2step excitation 1 : 1-2 step excitation	00 : IN / OUT 01 : IN / OUT, TO3 10 : IN / M1 (Timer2, Timer3) 11 : (Timer4)			0 : 1-step /2step excitation 1 : 1-2 step excitation	00 : IN/OUT 01 : IN/OUT, TO1 1X : IN/M0 (Timer 0, Timer 1)	
SMCR	Stepping Motor Control Reg.	0FFCFH				CCW7				CCW6
						R/W				R/W
						0				0
						0 : Normal rotation 1:Reverse rotation				0:Normal rotation 1:Reverse rotation

280890

Also refer to P67CR, P6 and P7 registers.

(4) Watchdog Timer Control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
WDMOD	Watch Dog Timer Mode Reg.	0FFD2H	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	EXF	DRVE
			R/W	R/W		R/W	R/W		R	R/W
			1	0	0	0	0	0	Undefined	0
			1:WDT Enable	WDT Detecting time 00 : 2^{14}/fc 01 : 2^{16}/fc 10 : 2^{18}/fc 11 : 2^{20}/fc		Warming up time 0 : 2^{14}/fc 1 : 2^{16}/fc	Standby mode 00:RUN mode 01:STOP mode 10: IDLE1 mode 11: IDLE2 mode		Invert each time EXX inst-ruction is executed	1: to drive pin in STOP mode.
WDCR	Watch Dog Timer Control Reg.	0FFD3H prohibit RMW	—							
			W							
			—							
			B1H : WDT Disable code 4EH : WDT Clear code							

210689

(5) Timer/event Counter Control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TREG0	8bit Timer Register 0	0FFD4H prohibit RMW	—							
			W							
			Undefined							
TREG1	8bit Timer Register 1	0FFD5H prohibit RMW	—							
			W							
			Undefined							
TREG2	8bit Timer Register 2	0FFD6H prohibit RMW	—							
			W							
			Undefined							
TREG3	8bit Timer Register 3	0FFD7H prohibit RMW	—							
			W							
			Undefined							
TCLK	8bit Timer Source Clock Control Reg.	0FFD8H	T3CLK1	T3CLK0	T2CLK1	T2CLK0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W		R/W		R/W		R/W	
			0	0	0	0	0	0	0	0
			8-bit 00 : TO2TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : — 01 : ϕT1 10 : ϕT16 11 : ϕT256		8-bit 00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : — 01 : ϕT1 10 : ϕT16 11 : ϕT256	
TFFCR	8bit Timer Flip-Flop Control Reg.	0FFD9H	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R/W		W		R/W	
			—		0	0	—		0	0
			00 : Clear TFF3 01 : Set TFF3 10 : Invert TFF3 11 : Don't care		1 : TFF3 Invert Enable	0 : Invert by 8bit timer 2 1 : Invert by 8 bit Timer 3	00 : Clear TFF1 01 : Set TFF1 10 : Invert TFF1 11 : Don't care		1 : TFF1 Invert Enable	0 : Invert by 8bit timer 0 1 : Invert by 8 bit Timer 1
TMOD	8bit Timer Mode reg.	0FFDAH	T32M1	T32M0	PWM21	PWM20	T10M1	T10M0	PWM01	PWM00
			R/W		R/W		R/W		R/W	
			0	0	0	0	0	0	0	0
			00 : 8bit Timer 01 : 16bit Timer 10 : 8bit PPG 11 : 8bit PWM		PWM Frequency 00 : — 01 : 2^6-1 10 : 2^7-1 11 : 2^8-1		00 : 8bit Timer 01 : 16bit Timer 10 : 8bit PPG 11 : 8bit PWM		PWM Frequency 00 : — 01 : 2^6-1 10 : 2^7-1 11 : 2^8-1	

280890

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TRUN	8bit Timer/ Serial Channel Baud Rate Control Reg.	0FFDBH	BRATE1	BRATE0	PRRUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
			R/W		R/W					
			0	0	0	0	0	0	0	0
			00 : 300/150 bps 01 : 1200/600 10 : 4800/2400 11 : 19200/9600		Prescaler & Timer Run/Stop Control 0: Stop & Clear 1: Run (Count up)					
CAP1L	16bit Timer/ Event Counter Capture Register 1	0FFDCH	—							
			R							
			Undefined							
CAP1H		0FFDDH	—							
			R							
			Undefined							
CAP2L	16bit Timer/ Event Counter Capture Register 2	0FFDEH	—							
			R							
			Undefined							
CAP2H		0FFDFH	—							
			R							
			Undefined							
TREG4L	16bit Timer/ Event Counter Register 4	0FFE0H prohibit RMW	—							
			W							
			Undefined							
TREG4H		0FFE1H prohibit RMW	—							
			W							
			Undefined							
TREG5L	16bit Timer/ Event Counter Register 5	0FFE2H prohibit RMW	—							
			W							
			Undefined							
TREG5H		0FFE3H prohibit RMW	—							
			W							
			Undefined							

280890

MSB LSB

Symbol	Name	Address	7	6	5	4	3	2	1	0
T4MOD	16bit Timer/ Event Counter Mode reg.	0FFE4H			CAP1IN	CAPM1	CAPM0	CLE	T4CLK1	T4CLK0
					W	R/W		R/W	R/W	
					—	0	0	0	0	0
					0 : Software Capture 1: don't care	Capture timing 00 : Disable INT1 01 : TI4 ↑ TI5 ↑ INT1 10 : TI4 ↑ TI4 ↓ INT1 11 : TFF1 ↑ TFF1 ↓ INT1		1: UC16 Clear Enable	Timer 4 source clock 00 : TI4 01 : ϕT1 10 : ϕT16 11 : —	
T4FFCR	16bit Timer Flip-Flop4 Control reg.	0FFE5H			CAP2TE	CAP1TE	EQ5TE	EQ4TE	TFF4C1	TFF4C0
					R/W				W	
					0	0	0	0	—	
					TFF4 inversion trigger 0 : Disable trigger 1 : Enable trigger				00 : Clear TFF4 01 : Set TFF4 10 : Invert TFF4 11 : Don't care	

210689

(6) Serial Channel Control

MSB LSB

Symbol	Name	Address	7	6	5	4	3	2	1	0
SCMOD	Serial Channel Mode Reg.	0FFE9H	TB8	Fixed at "0"	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Trans-mission Bit-8 data in 9bit UART	Write "0"	1 : Receive Enable	1 : Wake up Enable	00 : I/O interface 01 : UART 7bit 10 : UART 8bit 11 : UART 9bit		00 : TO2TRG 01 : BR 10 : ϕ1 11 : BR 1/2	UART
SCCR	Serial Channel Control Register	0FFEAH	RB8	EVEN	PE	OERR	PERR	FERR		CTSE
			R	R/W		R (Cleared to "0" by reading)				R/W
			Undefined	0	0	0	0	0		0
			Receiving Bit-8 data	Parity 0 : Odd 1 : Even	1 : Parity Enable	1 : Error Overrun	Parity	Flaming		1:CTS Enable
SCBUF	Serial Channel Buffer Register	0FFEBH	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
		prohibit RMW	R (Receiving)/W (Transmission)							
			Undefined							

210689

Also refer to P3CR, TRUN register. Note : BR:Baud Rate Generator

(7) A/D Converyter Control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
ADMOD	A/D Converter mode Reg	0FFEFH	ADRPT	EOCF	ADBF	ADCS	ADS	ADCH2	ADCH1	ADCH0
				R		R/W	R/W	R/W		
			1	0	0	0	0	0	0	0
			0:Repeat 1: Single	1:End	1:Busy	0:95state 1:190state	1: Start	Analog Input Channel Select		
ADREG	A/D Result Register	0FFEEH	—							
			R							
			—							

210689

(8) Interrupt Control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
INTEL	Interrupt Enable Mask Reg.	0FFE6H	*IET2	IET3	IET4	IE1	IET5	IE2	IERX	IETX
			R/W							
			0	0	0	0	0	0	0	0
			1:Enable				0:Disable			
INTEH (DMAEL)	Micro DMA Enable Register	0FFE7H	0	DE0	DET0	DET1	ADIS	IE0	IET0	IET1
			R/W				R/W			
			0	0	0	0	0	0	0	0
			1 : Enable		0 : Disable		1 : INTAD	1 : Enable	0 : Disable	
DMAEH		0FFE8H	DET2	DET3	DET4	DE1	DET5	DE2	DERX	DETX
			R/W							
			0	0	0	0	0	0	0	0
			1:Enable				0:Disable			
IRFL (P01CR)	Interrupt Request Flag & IRF Clear	0FFC2H prohibit RMW		IRF0	IRFT0	IRFT1		EXT	P1CR	P0CR
				R				W	W	
				0	0	0		0	0	0
				Interrupt Request Flag 1 : Interrupt being requested				P1, P2 Controls 0:I/O port 1:Address bus	P1 Controls 0 : In 1 : Out	P0 Controls 0 : In 1 : Out
IRFH		0FFC3H prohibit RMW	IRFT2	IRFT3	IRFT4	IRF1	IRFT5	IRF2	IRFRX	IRFTX
			R (Only IRF Clear code can be used to write)							
			0	0	0	0	0	0	0	0
			1: Interrupt being requested (IRF is cleared to "0" by writing IRF Clear code).							

280890

Symbol in ( ) denotes another name. * Share with IEAD (If ADIS = 1, use as INTAD Mask Reg.)

(9) Bank Register

MSB LSB

Symbol	Name	Address	7	6	5	4	3	2	1	0
BX	Bank Register X	0FFECH					BX3	BX2	BX1	BX0
							R/W			
							0	0	0	0
BY	Bank Register Y	0FFEDH					BY3	BY2	BY1	BY0
							R/W			
							0	0	0	0

210689

6. PORT SECTION EQUIVALENT CIRCUIT DIAGRAM

- Reading The Circuit Diagram

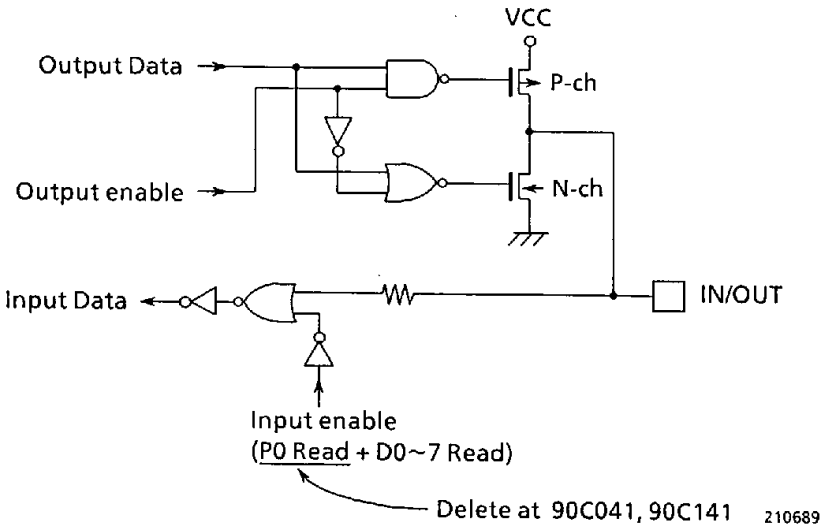
  Basically, the gate singles written are the same as those used for the standard CMOS logic IC [74HCXX] series.

  The dedicated signal is described below.

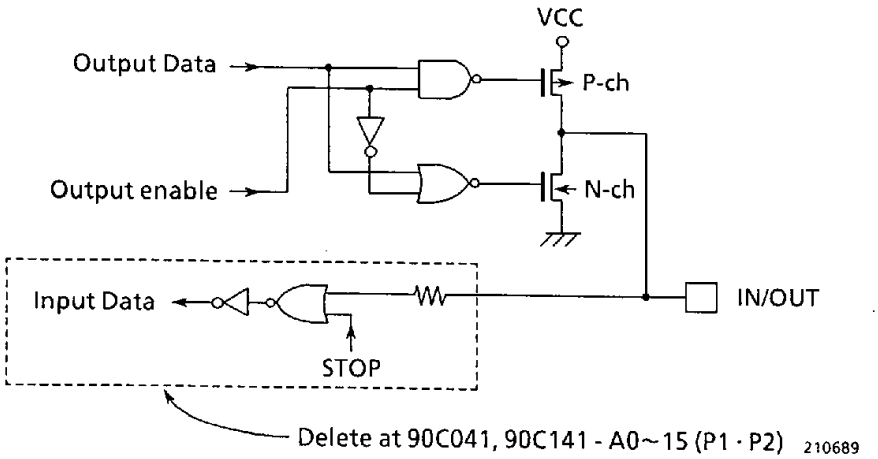
  STOP : This signal becomes active "1" when the hold mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit [DRIVE] is set to "1", however, STP remains at "0".

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

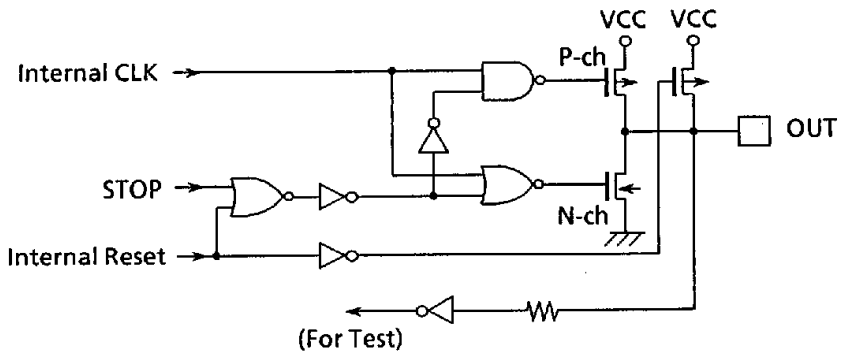
■ P0 (D0~D7)



■ P1, P2, P6, P7

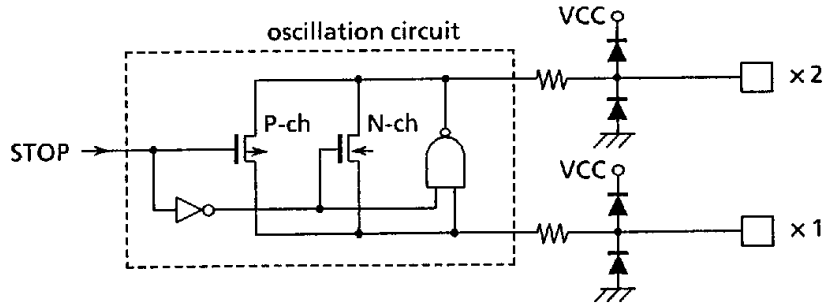


■ CLK



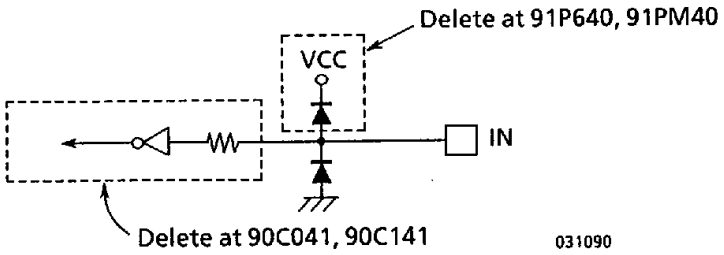
210689

■ X1, X2



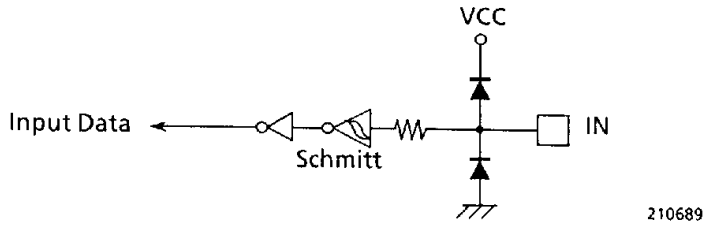
210689

■ $\overline{EA}$

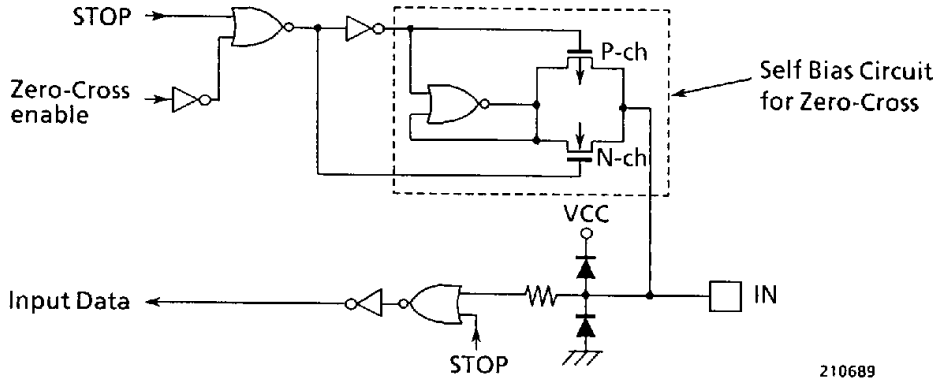


031090

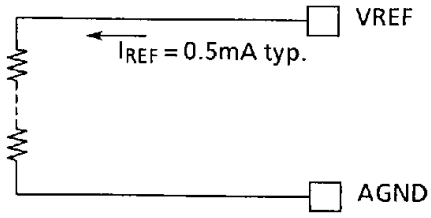
■ P80, $\overline{NMI}$



■ P81, P82

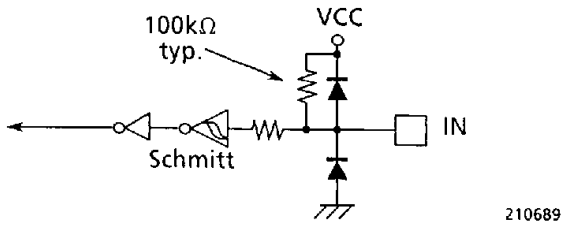


■ VREF, AGND

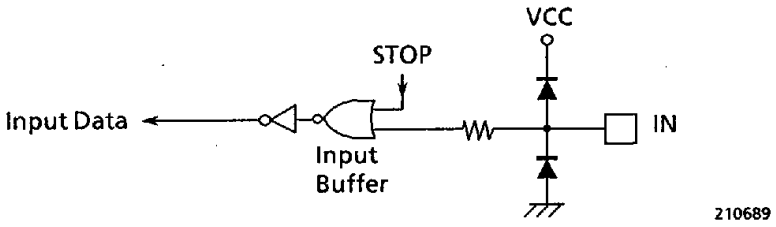


Note : I_{REF} continues to flow even during standby.

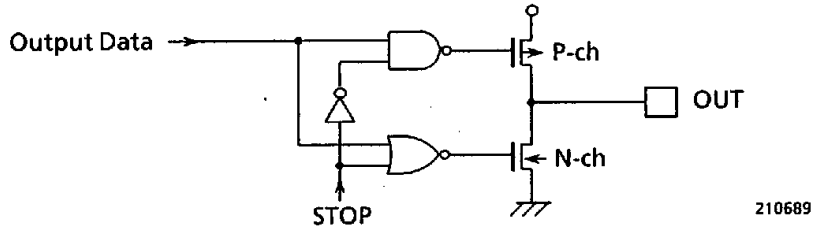
■ $\overline{RESET}$



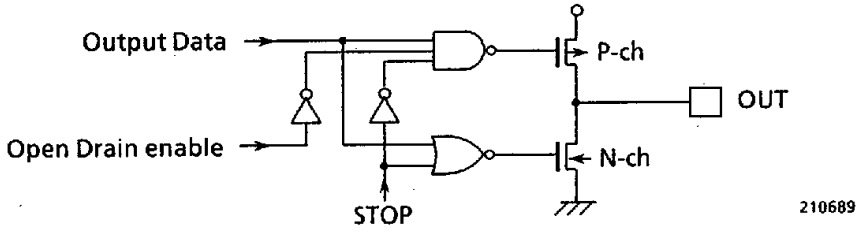
■ P30, P31, P34, P37



■ P32, P35, P36, P4, P83



■ P33



■ P5