TOSHIBA

CMOS 8–Bit Microcontroller

TMP90C846F

1. Outline and Characteristics

The TMP90C846 is an advanced 8-bit microcontroller developed for application in the control of HDD/FDD high-speed mechanisms. The built-in functions include a high-speed A/D converter (minimum sampling rate: 400ns @ 10MHz) with an external start function, and a D/A converter.

The TMP90C846, integrates 8-bit CPU, ROM, RAM, high-speed A/D converter, D/A converter, and multi-function timer/event counter in a single-chip.

The TMP90C846 uses a 44-pin mini flat package (QFP44-P-1414D).

The following are the features of the TMP90C846:

(1) Highly efficient instruction set:
    163 basic instructions
    Division and multiplication instructions, 16-bit operation instruction and bit manipulation operation instructions and bit operation instructions.

(2) Minimum instruction executing time: 400ns (@ 10MHz)

(3) Built-in ROM: 8K bytes

(4) Built-in RAM: 256 bytes

(5) Memory expansion capability
    External program memory: 56K bytes
    External data memory: 56K bytes

(6) Highly-speed A/D converter (2 channels)
    • Minimum sampling rate: 400ns (@ 10MHz)
    • 16-byte FIFO RAM (conversion data storage)
    • External start, software start (one-time conversion, repeat conversion

(7) 8-bit voltage output type D/A converter (2 channels)

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

(9) 8-bit timer (4 channels)

(10) Interrupt function: 9 internal, 4 external

(11) Micro DMA function (10 channels)

(12) Watchdog timer function

(13) Zero cross detector (2 pins)

(14) I/O ports (28 pins)
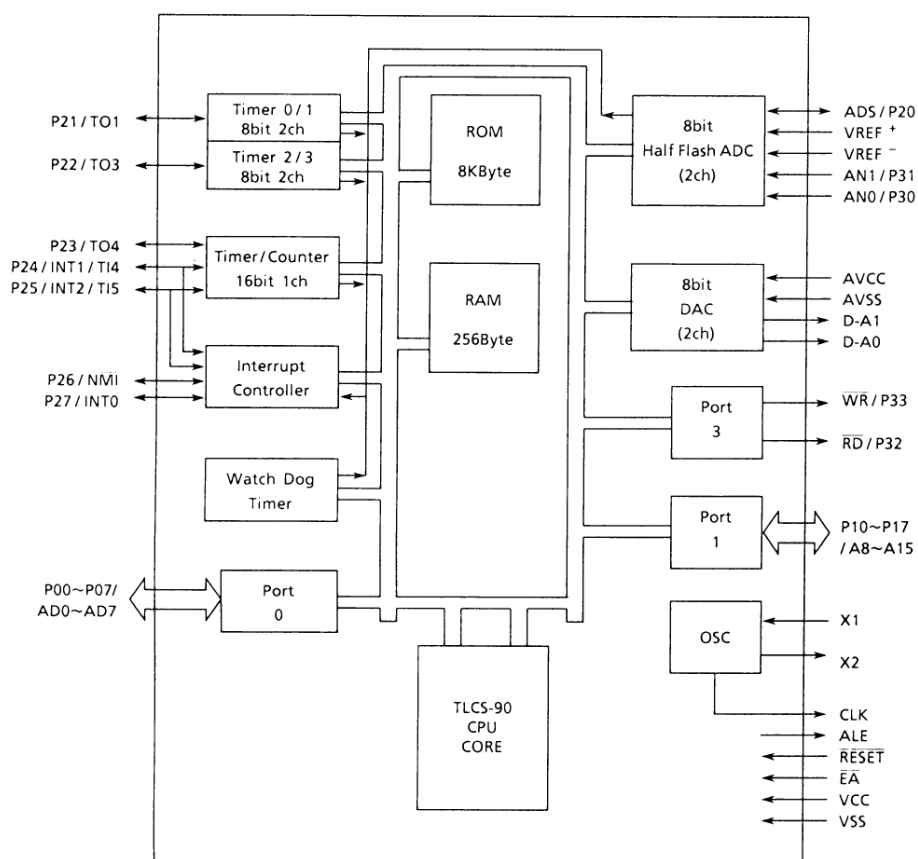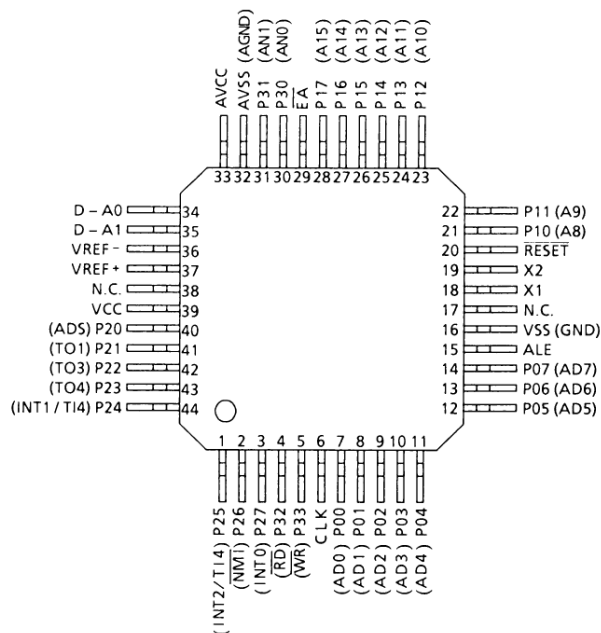
(15) Standby function (4 HALT modes)

Figure 1. TMP90C846 Block Diagram

2. Pin Assignments and Functions

The TMP90C846F, pin assignments, I/O pin names and functions are described in this section.

Figure 2.1 shows the TMP90C846F pin assignments.

2.1 Pin Assignments

(Note) N.C.: No Connection

Figure 2.1. Pin Assignment (Mini Flat Package)

2.2 Pin Names and Functions

Table 2.2 shows the I/O pin names and functions.

Table 2.2 Pin Assignments and Functions

Pin Name	No. of pins	I/O or tristate	Function
P00 ~ P07 /AD0 ~ AD7	8	I/O	Port 0: An 8-bit I/O port. Each bit can be set for input or output.
		Tristate	Operates as an address/data bus during external program execution.
P10 ~ P17 /A8 ~ A15	8	I/O	Port 1: An 8-bit I/O port. Each bit can be set for input or output.
		Output	Address bus: Operates as the 8 upper bits of the address bus when using external memory.
P20 /ADS	1	I/O	Port 20: A 1-bit I/O port. A/D conversion start: The input pin for the A/D conversion start signal.
P21 /TO1	1	I/O	Port 21: A 1-bit I/O port. Timer output 1: Timer 0 or timer 1 output.
P22 /TO3	1	I/O	Port 22: A 1-bit I/O port. Timer output 3: Timer 2 or timer 3 output.
P23 /TO4	1	I/O	Port 23: A 1-bit I/O port. Timer output 4: Timer 4 output.
P24 /INT1 /TI4	1	I/O	Port 24: A 1-bit I/O port. Interrupt request pin 1: A rising/falling edge programmable interrupt request pin. Timer input 4: Timer 4 count input/capture trigger signal input.
P25 /INTT2 /TI5	1	I/O	Port 25: A 1-bit I/O port. Interrupt request pin 2: A rising edge interrupt request pin. Timer input 5: Timer 4 capture trigger signal input.
P26 /$\overline{\text{NMI}}$	1	I/O	Port 26: A 1-bit I/O port. Non-maskable interrupt request pin: A falling edge interrupt request pin (after register setting).
P27 /INT0	1	I/O	Port 27: A 1-bit I/O port. Interrupt request pin 0: A level/rising edge programmable interrupt request pin.
P30, P31 /AN0, AN1	2	Input	Port 30, 31: 2-bit input ports. Analog input: Two analog inputs to the A/D converter.
P32 /$\overline{\text{RD}}$	1	Output	Port 32: A 1-bit output port. Read: The strobe signal output for reading external memory.
P33 /$\overline{\text{WR}}$	1	Output	Port 33: A 1-bit output port. Write: The strobe signal output for writing external memory.
D - A0, D - A1	2	Output	D/A output: The analog voltage output pin for D/A converters 0/1.
VREF+	1	–	A/D converter High reference voltage input.
VREF-	1	–	A/D converter Low reference voltage input.
A Vcc	1	–	Used as both A/D converter and D/A converter power supply, and D/A reference voltage input.
A Vss	1	–	Used as both the analog GND pin and D/A reference voltage.
ALE	1	Output	Address latch enable: The falling edge of this signal is used as the timing for latching addresses on AD0 - AD7 when accessing external memory.
CLK	1	Output	Clock output: Outputs 1/4 frequency of clock oscillation. Pulled up internally during reset.
$\overline{\text{EA}}$	1	Input	Connect to the V_{CC} pin when the built-in ROM is used; connect to the GND pin when an external memory is used instead.
$\overline{\text{RESET}}$	1	Input	Reset: Initializes the TMP90C846. (pull-up resistor is built-in).
X1/X2	2	I/O	The crystal/ceramic oscillator connection pin.
V_{CC}	1	–	Power supply (+5V)
V_{SS} (GND)	1	–	GND pin (0V)

3. Operation

This section explains the functions and basic operations of the TMP90C846.

3.1 CPU

The TMP90C846 has a built-in high-performance 8-bit CPU. Refer to the book TLCS-90 Series CPU Core Architecture concerning the CPU operation.

Following section explains the CPU functions unique to the TMP90C846 that are not explained in that book.

3.1.1 Resets

The basic reset timing is shown in Figure 3.1.

To reset the TMP90C846, it is necessary to maintain the $\overline{\text{RESET}}$ input at "0" for at least 10 system clocks (10 states: 2µsec @ 10MHz) with the power supply voltage within the operating range and with stabilized oscillation.

When a reset is received, I/O port 0 (address data bus AD0 ~ AD7), port 1 (address bus A8 ~ A15), and port 2 are all set to input status (high impedance). Output ports P32 ($\overline{\text{RD}}$) and P33 ($\overline{\text{WR}}$) and CLK are all set to "1". ALE is cleared to "0".The registers of the CPU also remain unchanged. Note, however, that the program counter PC, the interrupt enable flag IFF are cleared to "0". Register A shows an reset, because $\overline{\text{WR}}$ is set to "1" before undefined address/data is outputted.

When the reset is released, instruction execution starts from address 0000H.
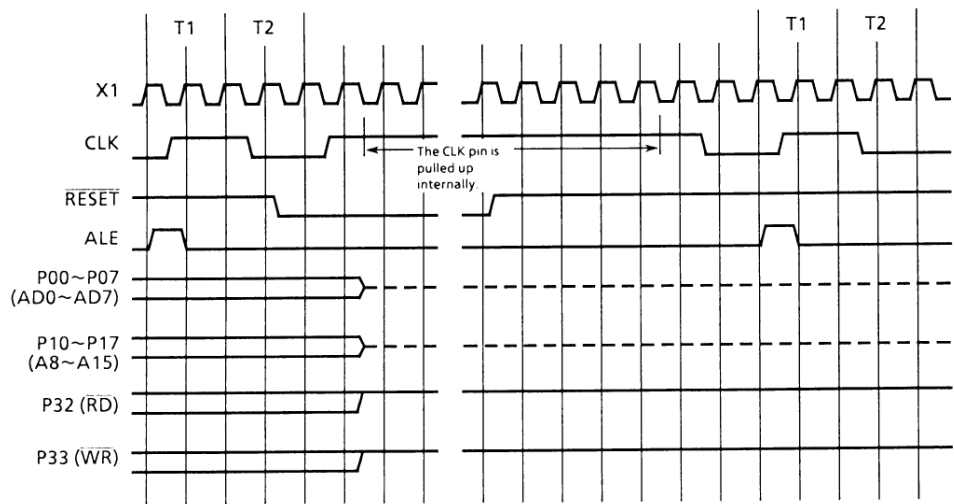
Figure 3.1. Reset Timing

3.1.2 EXF (Exchange Flag)

The exchange flag <EXF> is assigned to bit 1 of memory address FFD2H. This flag is inverted when the register exchange instruction [EXX] is executed between main registers and auxiliary registers.

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	Indeterminate	0
	Function	1 : WDT Enable	WDT Detection 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		Inverts each time the EXX instruction is executed.	1 : Pin is driven even in the STOP mode.

3.2 Memory Map

The TMP90C846 can handle up to 64K bytes of program memory and data memory.

Program and data memory can be located at address 00000H ~ FFFFH.

(1) Built-in ROM

The TMP90C846 has 8 bytes of the built-in ROM located at addresses 0000H ~ 1FFFH. After the CPU is reset, The instruction execution starts from address 0000H.
Addresses 0010H ~ 0077H in the built-in ROM area are used as the entry are for interrupt processing.

(2) Built-in RAM

The TMP90C846 has 256 bytes of the built-in RAM

located at addresses FFC0H ~ FFBFH. In the direct addressing mode, the CPU allows the access to a certain RAM area (192 bytes at addresses FF00H ~ FFBFH) using short instruction codes.
Addresses of FF28H ~ FF77H this RAM area can be used as the parameter area for micro DMA processing. (This area can be used as RAM when not using micro DMA processing.)

(3) Built-in I/O

The TMP90C846 uses 48 bytes of the address space as a built-in I/O area. This area is assigned to addresses FFC0H ~ FFEFH. In the direct addressing mode, the CPU can access the built-in I/O area using short instruction codes.
Figure 3.2 shows the memory map and the access ranges of the CPU for each addressing mode.

```
0000H  ┌────────────────┐ ───────────────────────────────────────┐
       │  Built-in      │                                         ▲
       │  ROM           │                                         │
       │  (8K bytes)    │                                         │
2000H  ├────────────────┤                                         │
       │                │          Program and                    │
       │                │          data areas                     │
       │  External memory│                                        │
       │  (56K bytes)   │          (BC)                            │
       │                │          (DE)                            │
       │                │          (HL)                            │
       │                │          (SP)                            │
       │                │          (IX)                            │
       │                │          (IY)                            │
       │                │          (IX + d)                        │
       │                │          (IY + d)                        │
FEC0H  ├────────────────┤          (SP + d)                        │
       │                │          (HL + A)                        │
       │  Built-in RAM  │          (nn)              ──────┐       │
FF00H  ├╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌┤                                  ▲       │
       │                │                                  │       │
       │  (256 bytes)   │                                  │       │
       │                │                          Direct area    │
FFC0H  ├╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌┤                                (n)       │
       │                │                                  │       │
       │  Built-in I/O  │                                  │       │
       │  (48 bytes)    │                                  │       │
       │                │                                  │       │
FFF0H  ├────────────────┤                                  ▼       │
       │  External memory│                          ──────┘       │
       │  (16 bytes)    │                                         ▼
FFFFH  └────────────────┘ ───────────────────────────────────────┘
```
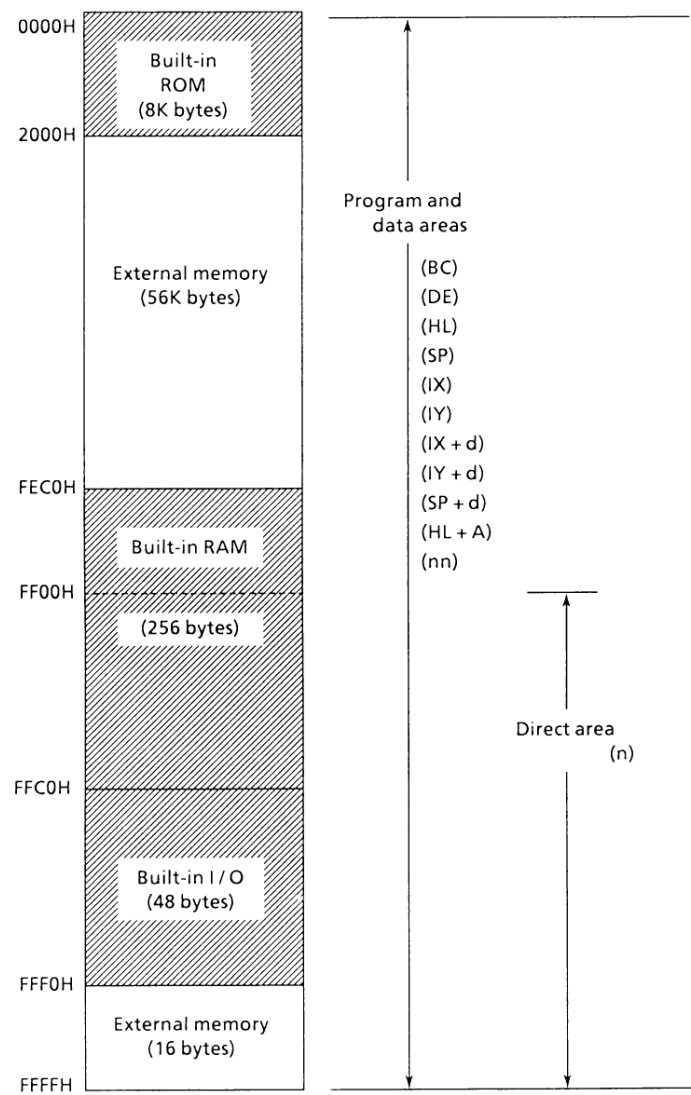
Note) The memory area is 64K bytes because there are no BX and BY registers as with the TMP90C840A.

Figure 3.2. Memory Map

3.3 Interrupt Functions

The TMP90C846 has a general-purpose interrupt processing mode and a micro DMA processing mode in which the CPU automatically transfers data for internal and external interrupt requests.

After a reset is released, all responses to interrupt requests are set to the general-purpose interruot processing mode. The interrupt request can be set to the micro DMA processing mode with the DMA enable register which is described later.

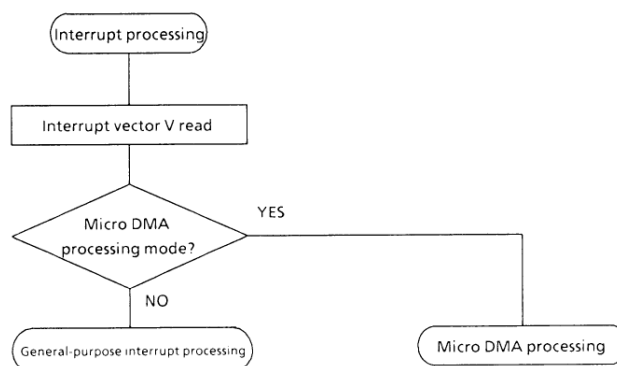The interrupt response flow chart is shown in Figure 3.3 (1).

Figure 3.3 (1). Interrupt Response Flow Chart

When an interrupt request is generated, this is reported to the CPU via built-in interrupt controller. The CPU starts the interrupt processing if it is a non-maskable interrupt or maskable interrupt requested in the EI state (interrupt enable flag (IFF bit of the F register) = "1"). A maskable interrupt requested in the DI state (IFF = "0") is ignored and not received. (The CPU samples interrupt requests at the falling edge of CLK signal of the last bus cycle of each instruction.)

When an interrupt is received, the CPU first reads the interrupt vector from the built-in interrupt controller to determine the interrupt request source.

Next, the CPU checks whether this request is for processed in the general-purpose interrupt processing or micro DMA processing, and performs the corresponding processing.

The interrupt vector is read in an internal operation cycle, so the bus cycle results in dummy cycle.

3.3.1 General-Purpose Interrupt Processing

The general-purpose interrupt processing flow chart is shown in Figure 3.3 (2).

The CPU first saves the contents of the program counter PC and the register AF (including the interrupt enable flag IFF before an interrupt) to the stack, and resets the interrupt enable flag IFF to "0" (interrupt disable).Then it transfers the content of the interrupt vector "V" to the program counter and jumps to the interrupt processing program.

There is a 20-state overhead from the time the interrupt is received until the jump is made to the interrupt processing program.
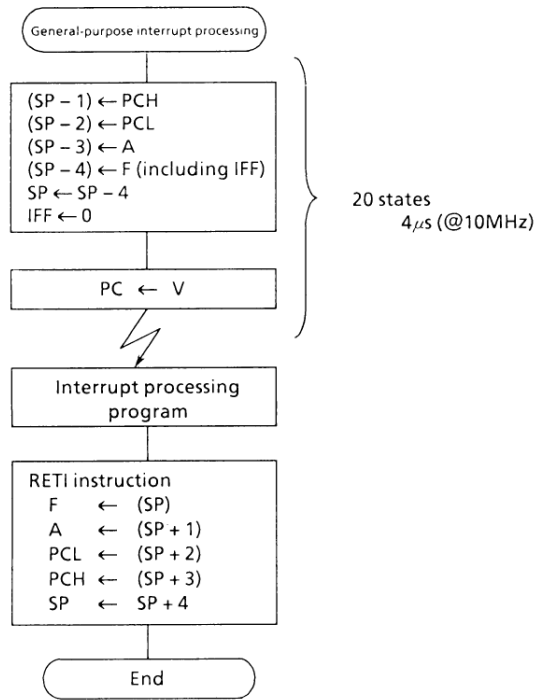

Figure 3.3 (2). General-Purpose Interrupt Processing Flowchart

The interrupt processing program ends with the RETI instruction for both maskable and non-maskable interrupts.

Executing this instruction (RETI) restores the contents of the program counter PC and the register AF from the stack. (Returns to the interrupt enable flag before the interrupt.)

When the CPU reads an interrupt vector, the interrupt request source confirms that the interrupt has been received, and clears the interrupt request.

A non-maskable interrupt cannot be disabled by program. A maskable interrupt, however, can be enabled or disabled by a program.

Bit 5 of the register F is the interrupt enable/disable flip-flop (IFF). Interrupts are enabled by setting this bit to "1" with the EI (endable interrupt) instruction and disabled by resetting to "0" with the DI (disable interrupt) instruction. IFF is reset to "0" by resetting or by receiving an interrupt (including non-maskable interrupts).

Interrupt enabled with the EI instruction become effective when the next instruction after the EI is executed.

The interrupt sources are shown in Table 3.3 (1).

Table 3.3 (1) Interrupt Sources

Priority	Type	Interrupt request source	Vector Value ÷ 8	Vector Value	General-purpose interrupt processing start address	Micro DMA processing parameter start address
1	Non-maskable	SWI instruction	–	10H	0010H	–
2		NMI (NMI pin input (programmable))	–	18H	0018H	–
3		INT WD (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	0030H	FF38H
7		INTAD (A/D converter)	08H	40H	0040H	FF40H
8		INTT2 (Timer 2)	09H	48H	0048H	FF48H
9		INTT3 (Timer 3)	0AH	50H	0050H	FF50H
10		INTT4 (Timer 4)	0BH	58H	0058H	FF58H
11		INT1 (External input 1)	0CH	60H	0060H	FF60H
12		INTT5 (Timer 5)	0DH	68H	0068H	FF68H
13		INT2 (External input 2)	0EH	70H	0070H	FF70H

The "priority" used in Table 3.3 (1) indicates the priority in which interrupt sources are received by the CPU when multiple interrupt requests are generated simultaneously.

For example, if the interrupt requests with the priority 4 and 5 are generated simultaneously, the CPU will receive the interrupt request with the priority 4 first. When the priority 4 interrupt processing is ended with the RETI instruction, the CPU will receive the interrupt with the priority 5. If the interrupt processing program with the priority 4 is interrupted by execting th EI instuction, the CPU will receive the priority 5 interrupt request.

When multiple interrupt request are generated simultaneously, the built-in interrupt controller only determines the priority of the interrupt sources received by the CPU. There is no function for comparing the priority between the currently processed interrupt and the currently request interrupt.

Another interrupt can be enabled while an interrupt is processed by setting the interrupt enable flag IFF to enable.

3.3.2 Micro DMA Processing

The micro DMA processing flow chart is shown in Figure 3.3 (3). The CPU first loads the parameters (transfer source and destination addresses, and transfer mode) for data transfer between memories from an address supplied by an interrupt vector value, and then transfers the data in accordance with those parameters. After that, parameters are updated and saved to the original location. The transfer count is decremented, and the micro DMA processing is ended unless the count is not "0". If the count is "0", the general-purpose interrupt processing is performed as described in the previous item.

Micro DMA processing

Loads parameters
    DST ← (FF00H + V + 1) W
    SRC ← (FF00H + V + 3) W
    CMD ← (FF00H + V + 5)

Transfers data between memories
    (DST) ← (SRC)
    DST   ← DST + 0 / 1 / 2
    SRC   ← SRC ± 0 / 1 / 2

Saves updated parameters
    (FF00H + V + 3) W ← SRC
    (FF00H + V + 1) W ← DST

Decrements transfer count
    COUNT ← (FF00H + V)
    COUNT ← COUNT – 1
    (FF00H + V) ← COUNT

COUNT = 0
    YES → General-purpose interrupt processing
    NO → End

DST   :   destination
SRC   :   source
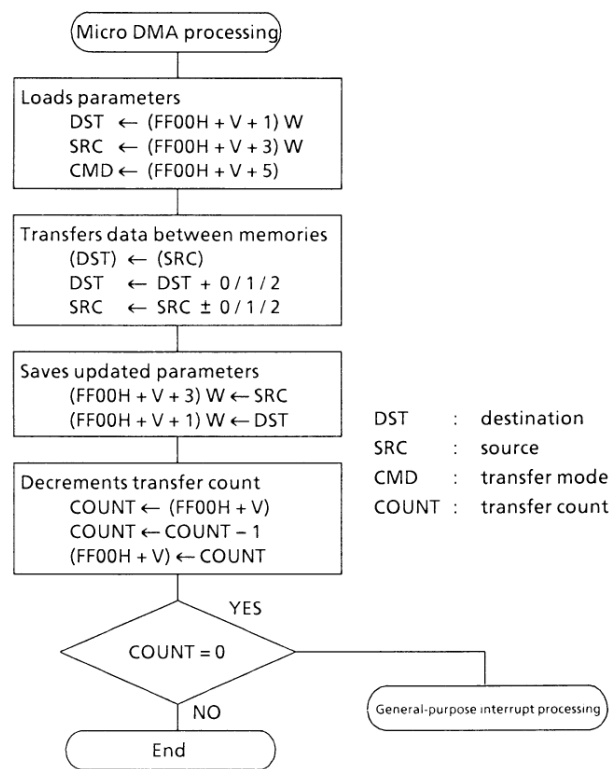CMD   :   transfer mode
COUNT :   transfer count

Figure 3.3 (3). Micro DMA Processing Flow Chart

The micro DMA processing is performed by using only hardware to process interrupts mostly completed by simple data transfer. Consequently, the micro DMA processing is faster than the conventional software processing, which in turn improves the interrupt processing speed. The micro DMA processing has absolutely no influence on the CPU registers.

The functions of the parameters used in the micro DMA processing are shown in Figure 3.3 (4).
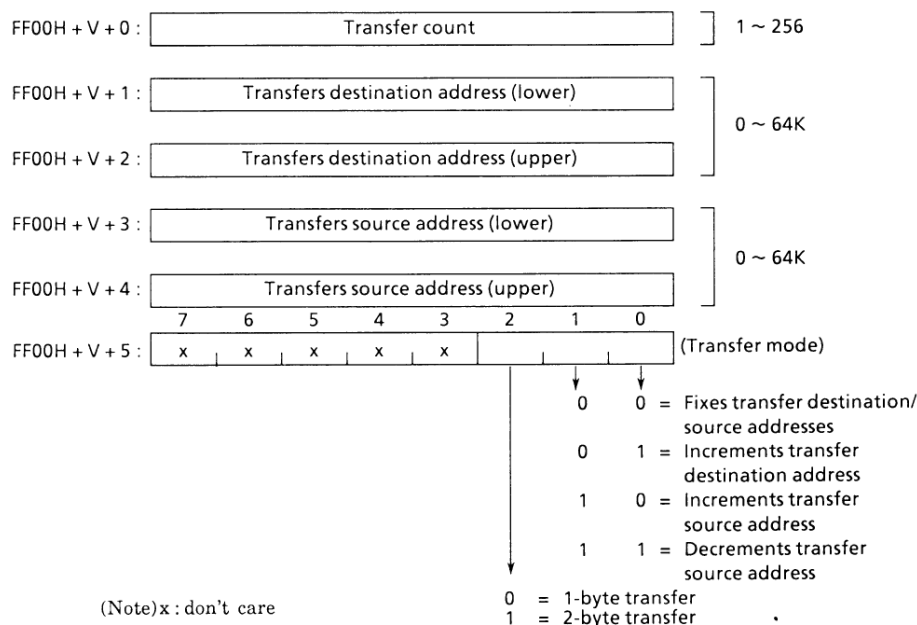
Figure 3.3 (4). Parameters for Micro DMA Processing

The parameters used for the micro DMA processing are located in the internal RAM (See Table 3.3 (1) Interrupt sources). The start address of each parameter is [FF00H + interrupt vector value], from which 6 bytes are used for the parameters. When micro DMA processing mode is not used, this area can be used as user memory.

The parameters include transfer count, transfer destination addresses, transfer source address, and transfer mode. The count indicates the number of data transfer accepted in the micro DMA processing. Either 1 or 2 bytes of data are transferred at one time with the micro DMA processing. Data are transferred 256 times with a transfer count of "00H". Transfer destination and transfer source addresses are each specified with 2 bytes of data. The address space 0000H - FFFFH is available for the micro DMA processing.

Bits 0 and 1 of the transfer mode specifies the mode updating the transfer source and/or destination. Bit 2 specifies the data length (1 or 2 bytes).

The relationship between the transfer mode and increment/ destination values of the transfer destination source addresses are shown in Table 3.3 (2).

Table 3.3 (2) Addresses Updated by Micro DMA Processing

Transfer Mode	Function	Destination address	Source address
000	Transfer 1 byte: fixes transfer destination/source addresses.	0	0
001	Transfer 1 byte: increments transfer destination address.	+1	0
010	Transfer 1 byte : increments transfer source address.	0	+1
011	Transfer 1 byte: decrements transfer source address.	0	-1
100	Transfer 2 byte: fixes transfer destination/source addresses.	0	0
101	Transfer 2 byte: increments transfer destination address.	+2	0
110	Transfer 2 byte: increments transfer source address.	0	+2
111	Transfer 2 byte: decrements transfer source address.	0	-2

In the 2-byte transfer mode, data are transferred as follows:
(Destination address) ← (Source address)
(Destination address + 1) ← (Source address + 1)
Though transfers are performed as shown above in "decrement transfer source address mode", addresses are updated as shown in the Table 3.3 (2).

Address increment and decrement are used for the memory area, but fixed addresses are used for ordinary I/O addresses. Because of that, I/O to memory and memory to I/O transfers were taken into consideration during the micro DMA design.

An example using the micro DMA processing mode is shown in Figure 3.3 (5). Conversion data of Built-in A/D converter are processed in this example.

This is an example of executing and "A/D conversion data processing program" after saving 3-byte conversion data into the memory addresses from FF00H to FF02H (built-in RAM area), by using INTAD which is enabled when the ADS1 conversion ends in a case where the external A/D conversion request signal (ADS) is entered as shown in Figure 3.3 (5).


```
CALL   ADINIT          ;   Initializes A/D converter settings
SET    6, (FFE6H)      ;   Enables INTAD
SET    6, (FFE8H)      ;   Sets INTAD to micro DMA processing mode
LD     (FF40H), 3      ;   Sets transfer count to 3
LDW    (FF41H), FF00H  ;   Sets transfer destination start address to FF00H
LDW    (FF43H), FFCEH  ;   Sets transfer source (ADREG1) address to FFCEH
LD     (FF45H), 01H    ;   Sets transfer mode
                           (1-byte transfer, increments transfer destination address)
EI
 :
 :
ORG    0040H
```

A/D conversion data processing program

```
RETI
```

Figure 3.3 (5). Micro DMA Processing Example

"Table 1.4 (2) Bus operation for each instruction" of the book TLCS-90 Series CPU Core Architecture shows the bus operations for general-purpose interrupt processing and micro DMA processing.

The execution time (when the transfer count is not 0 after decrementation) for micro DMA processing is 46 states (9.2μs @ 10MHz), regardless of whether the 1-byte or 2-byte transfer mode is used.

The interrupt processing flow chart is shown in Figure 3.3 (6).
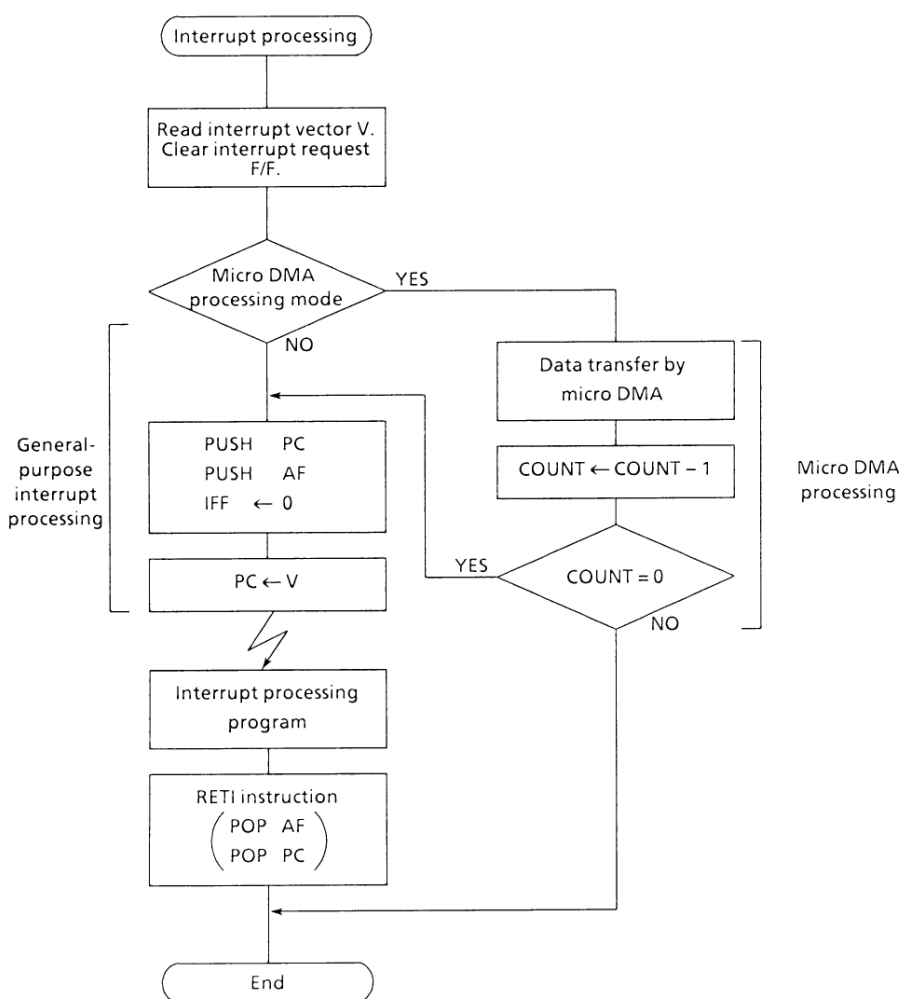
Figure 3.3 (6). Interrupt Processing Flow Chart

3.3.3 Interrupt Controller

The interrupt circuit diagram is shown in Figure 3.3 (8). The left side of this diagram shows the interrupt controller, and the right side shows the CPU interrupt request signal circuit and halt release circuit.

The interrupt controller has an interrupt request flipflop and interrupt enable flag, and a micro DMA enable flag (10 channels) for each of 13 interrupt channels. The interrupt request flip-flop latches interrupt requests that arrive from peripherals.

This flipflop is cleared to "0", when the CPU receives a reset or an interrupt and reads the vector of that interrupt channel, or when an instruction that clears the interrupt request (writes [vector/8] to the memory address FFC5H) for that channel is executed.

For example, when "LD (FFC5H), 38H/8" is executed, the interrupt request flip-flop for the interrupt channel [INT1] with the vector value 38H is cleared to "0" (write to FFC5H even when clearing the interrupt request flag assigned to FFC4H). When clearing an interrupt request, ensure that the interrupt source does not generate an interrupt request.

The status of the interrupt request flipflop can be checked by reading the memory address FFC4H or FFC5H. "0" means no interrupt request and "1" means an interrupt request. Figure 3.3 (7) shows the bit layout of the interrupt request flipflop.

(Note)　The specified interrupt request flipflop is cleared by writing [vector value/8] to the memory address FFC5H. However, IRFAD (INTAD interrupt request flag) cannot be cleared by instructions ; only by resetting or reading all FIFO data.
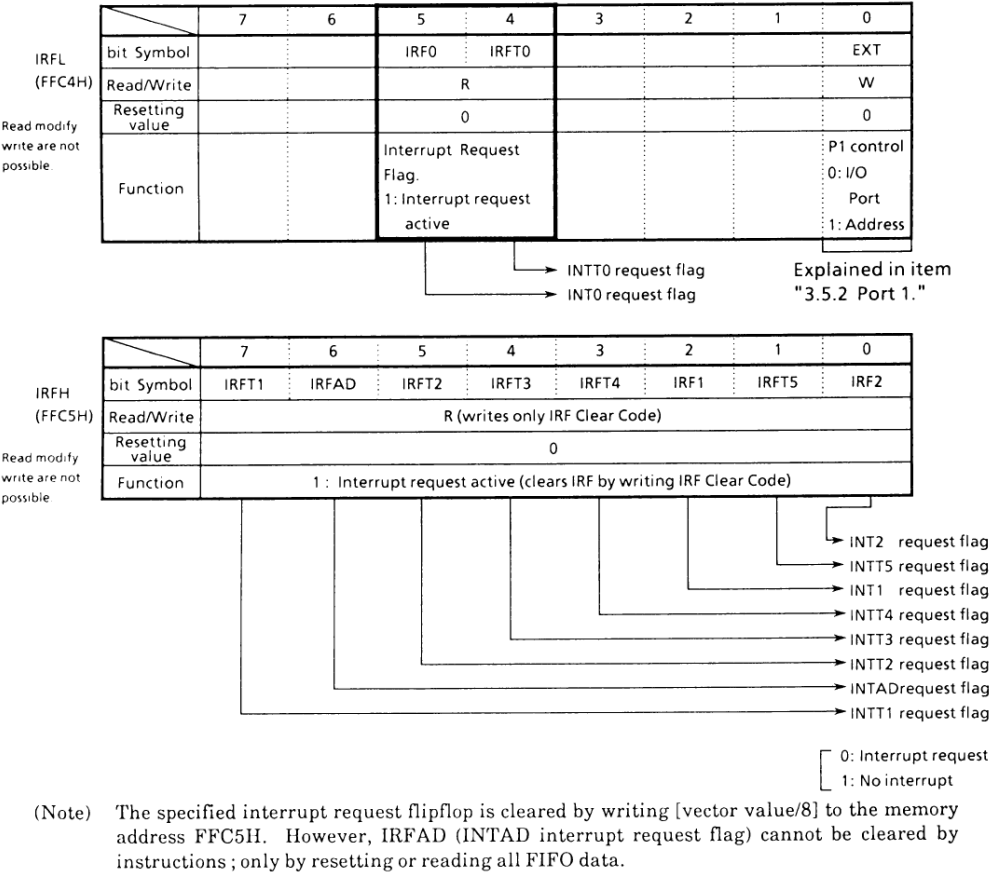
Figure 3.3 (7). Interrupt Request Flipflop Read
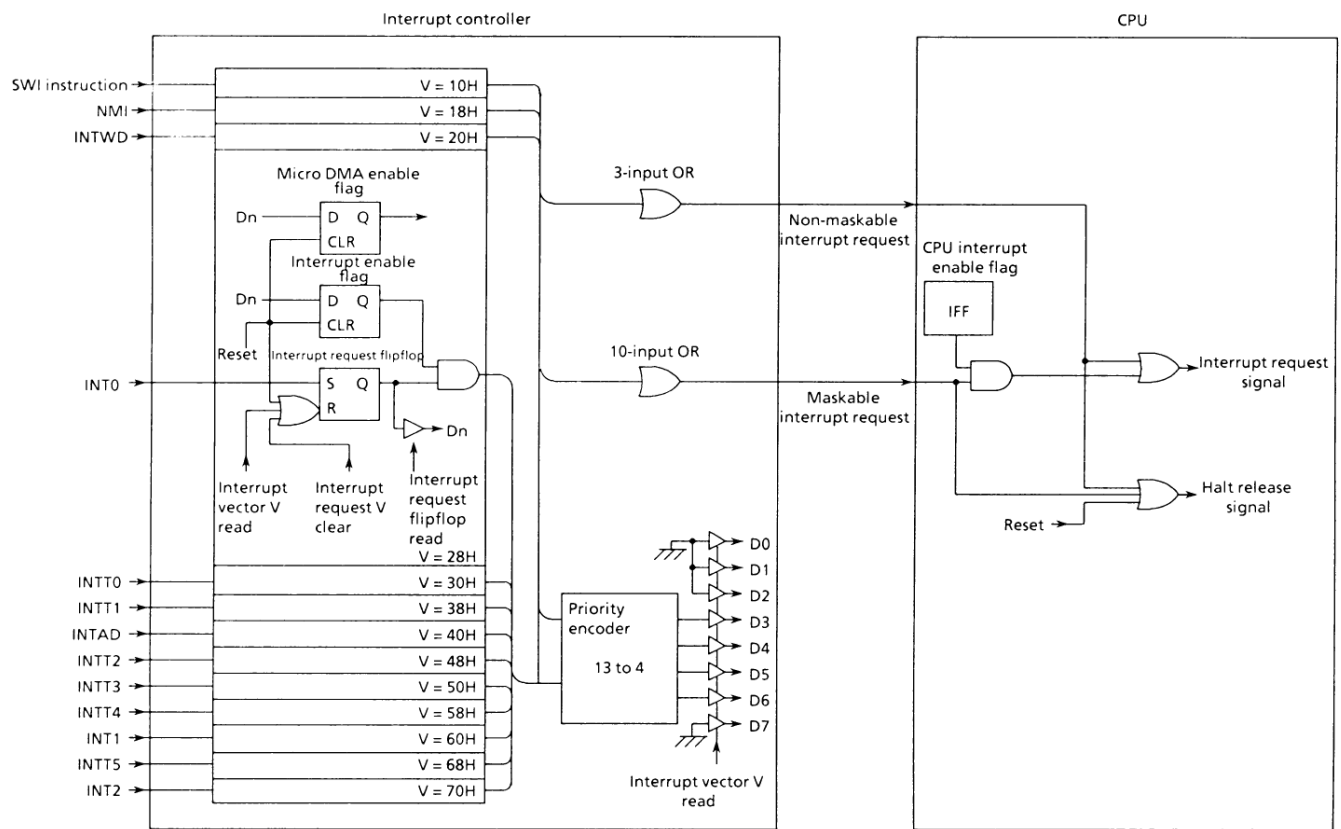
Figure 3.3 (8). Interrupt Circuit

An interrupt enable flag is assigned to the memory address FFE6H or FFE7H for each interrupt request channel. Interrupts for a channel are enabled by setting a flag to "1". The flag is cleared to "0" by resetting.

Clear the interrupt enable flag in the Disable Interrupts (DI) state.

The micro DMA enable flag for each interrupt request channel is assigned to the memory address FFE7H or FFE8H.

The interrupt requests for each channel are set to the micro DMA processing mode by setting the flag to "1". The flag is cleared to "0" by resetting ("0" is the general-purpose interrupt processing mode).
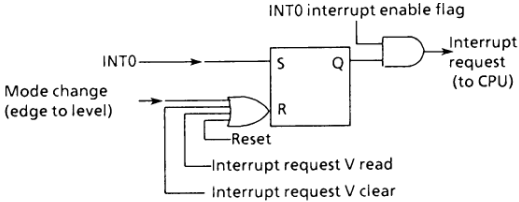
Figure 3.3 (9) shows the bit layout for the interrupt enable flags and micro DMA enable flags.

The table below shows the external interrupt functions.

Interrupt	Also Used as	Mode	Setting Method
$\overline{\text{NMI}}$	P26	Falling edge	NMI enable (by setting P2FR <NMIC> = 1)
INT0	P27	Level	P2FR <EDGE> = 0
		Rising edge	P2FR <EDGE> = 0
INT1	P24	Rising edge	T4MOD <CAPM1, 0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T4MOD <CAPM1, 0> = 1, 0
INT2	P25	Rising edge	–

Refer to "4.8 Interrupt Operation" concerning the external interrupt function pulse width.

Caution is required in the following two points as exceptions.

INT0 Level mode	This is not an edge-based interrupt, therefore interrupt request flipflop is cancelled. The peripheral interrupt request passes right through the flipflop S input and becomes the Q output.
	When the mode is changed over (from edge type to level type), the previous interrupt request flag will be cleared automatically.
	When the mode is changed from level to edge, the interrupt request flag set in the level mode is not cleared. Thus, use the following sequence to clear the interrupt request flag.
	DI
	LD 6, (FFFC9H), 80H
	LD (FFC5H), 05H
	EI
	When the INT0 level mode is used to clear the HALT (STOP) mode (the CPU in the EI state), an execution can be restarted from the interrupt vector address (0028H) by holding "1" until time preset by the warming-up counter. Note that if INT0 is cleared to "0" during warming-up, an execution will restart from the instruction following the HALT instruction.
	With the TLCS-90 and other products (TMP90C840A etc.), it is necessary to leave INT0 at "1" until the second bus cycle of this interrupt response sequence is completed when INT0 is set to "level". This restriction does not apply to the TMP90C846.
INTAD	The interrupt request flipflop cannot be cleared by instructions; only by resetting or reading all FIFO data.

Interrupt enable flags

INTEL (FFE6H)		7	6	5	4	3	2	1	0	FFE6H~FFE8H:
	bit Symbol	IET1	IEAD	IET2	IET3	IET4	IE1	IET5	IE2	0: Interrupt/DMA disable
	Read/Write	R / W								1: Interrupt//DMA enable
	Resetting value	0								
	Function	1 : Enable			0 : Disable					

→ INT2 interrupt enable flag
→ INTT5 interrupt enable flag
→ INT1 interrupt enable flag
→ INTT4 interrupt enable flag
→ INTT3 interrupt enable flag
→ INTT2 interrupt enable flag
→ INTAD interrupt enable flag
→ INTT1 interrupt enable flag

Micro DMA interrupt enable flags

INTEH (DMAEL) (FFE7H)		7	6	5	4	3	2	1	0
	bit Symbol			DE0	DET0			IE0	IET0
	Read/Write			R / W				R / W	
	Resetting value			0				0	
	Function			1 : Enable 0 : Disable				1 : Enable 0 : Disable	

→ INTT0 interrupt enable flag
→ INT0 interrupt enable flag
→ INTT0 DMA enable flag
→ INT0 DMA enable flag

Micro DMA enable flags

DMAEH (FFE8H)		7	6	5	4	3	2	1	0
	bit Symbol	DET1	DEAD	DET2	DET3	DET4	DE1	DET5	DE2
	Read/Write	R / W							
	Resetting value	0							
	Function	1 : Enable			0 : Disable				

→ INT2 DMA enable flag
→ INTT5 DMA enable flag
→ INT1 DMA enable flag
→ INTT4 DMA enable flag
→ INTT3 DMA enable flag
→ INTT2 DMA enable flag
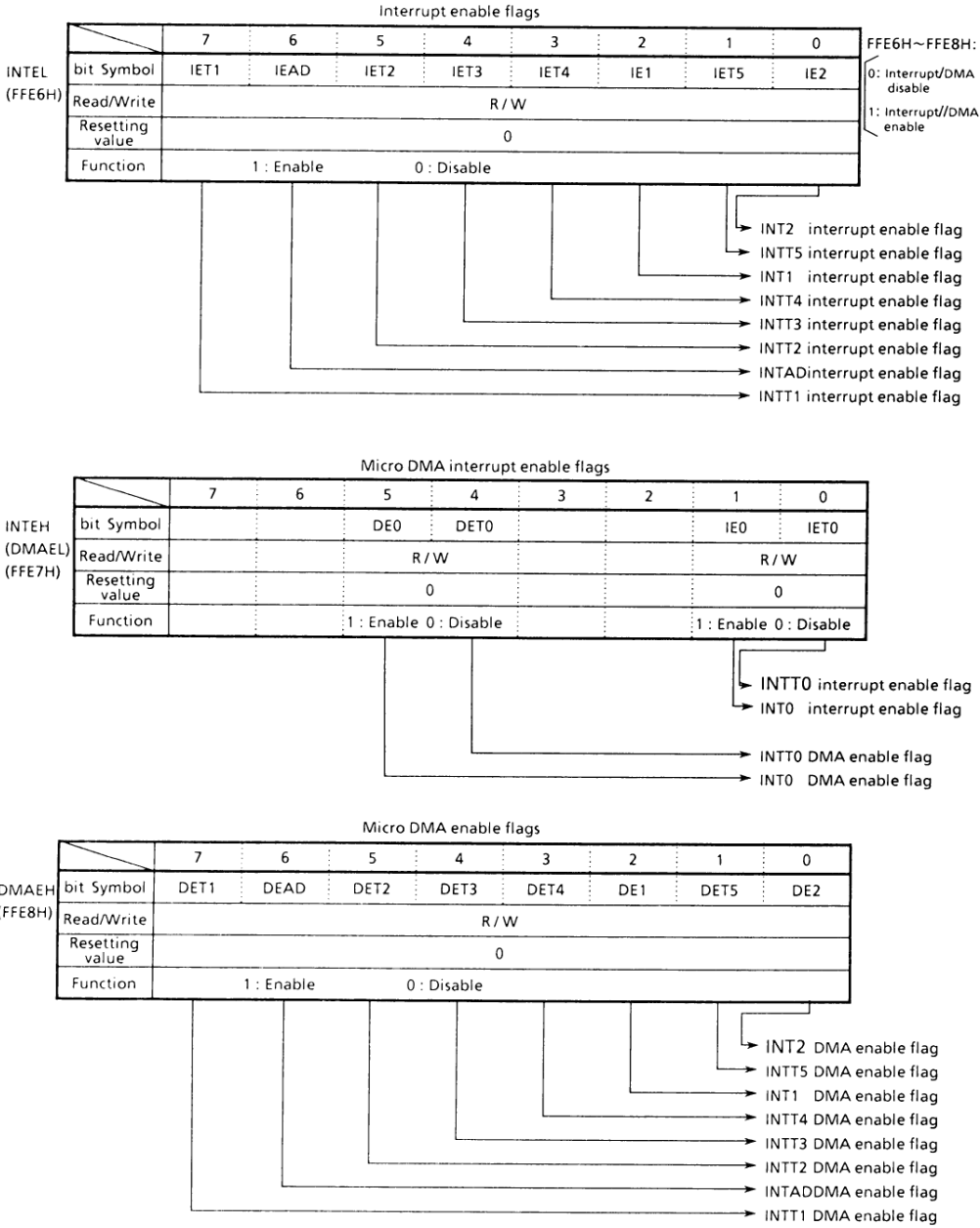→ INTAD DMA enable flag
→ INTT1 DMA enable flag

Figure 3.3 (9). Interrupt/Micro DMA Enable Flags

3.4 Standby Function

The TMP90C846 can be set to the RUN, IDLE1, IDLE2 or STOP modedepending on the contents of the halt mode set register, by executing the HALT instruction. The features are shown below:

(1) RUN: Only the CPU halts, and the power consumption remains the same.

(2) IDLE1:   Only the internal oscillators operate: all other internal circuits halt. The power consumption is 1/10 or less than that during the operation.

(3) IDLE2:   Only the internal oscillator and specific built-in I/O operate. In this mode, the power consumption is about 1/3 or less during the operation.

(3) STOP: All internal circuits halt, including the oscillator. The power consumption is extremely reduced.

The HALT mode set register WDMOD <HALTM1, 0> is assigned to the bits 2 and 3 of memory address FFD2H in the built-in I/O register area (all other bits are used to control other functions). The RUN mode ("00") is entered by resetting.

The HALT is released by interrupt requests or resets. The methods for releasing the halt state are shown in Table 3.4 (2). The CPU receives non-maskable interrupt or maskable interrupts EI state and starts interrupt processing. If maskable interrupts are disable (DI state), the CPU restarts an execution from the instruction following the HALT instruction, but the interrupt request flag remains at "1".

When Halt state is released by a reset , the state in effect before entering the halt state (including the 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)	bit Symbol	7 WDTE	6 WDTP1	5 WDTP0	4 WARM	3 HALTM1	2 HALTM0	1 EXF	0 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	Detection 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		Inverted each time the EXX instruction is executed.	1 : Pin is driven even in the STOP mode.

Explained in 3.7 Watchdog Timer

Explained in 3.4.4 STOP Mode

Exchange flag. Explained in 3.1.2 Registers.

Figure 3.4 (1). Halt Mode Set Register

3.4.1 RUN Mode

The timing for releasing the halt state by interrupts in the RUN/IDLE2 modes is shown in Figure 3.4 (2).

In the RUN mode, the system clock in the MCU does not stop even after HALT instruction is executed; the CPU merely stops executing the instruction. The CPU repeats dummy cycles until halt state is released. In the halt state, interrupt request are sampled at the falling edge of the CLK signal.

The halt state is released by external interrupt (INT1, INT2) requests only in the RUN mode.
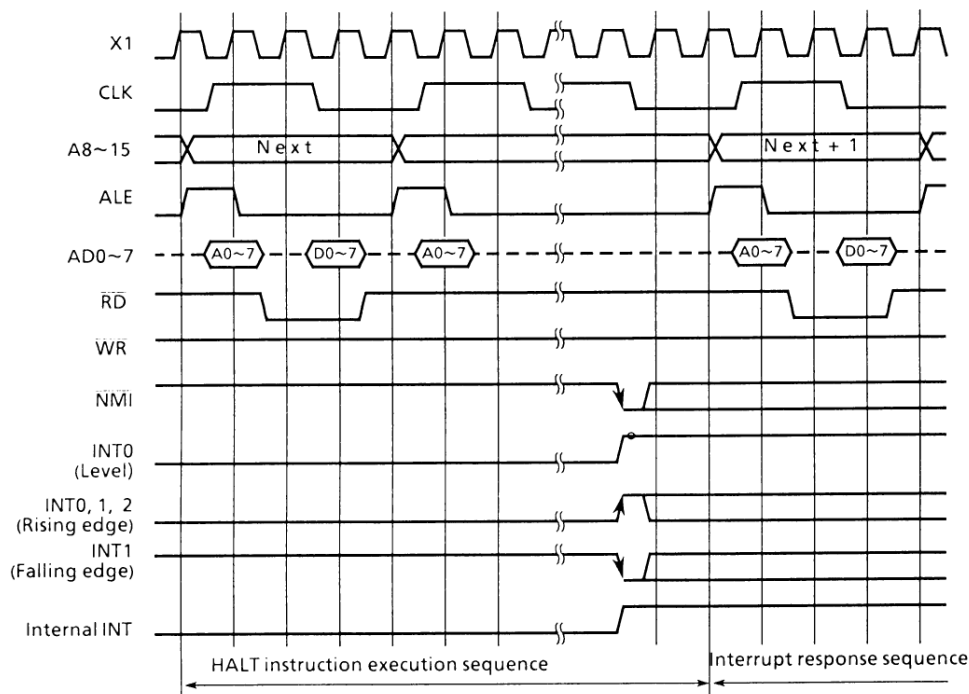
Figure 3.4 (2). Halt Release Timing Using Interrupts in the RUN/IDLE2 Modes

3.4.2 IDLE 1 Mode

The timing for releasing the halt state by interrupts in the IDLE1 mode is shown in Figure 3.4 (3).

In the IDLE1 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".

In the HALT mode, interrupt requests are sampled asynchronously with the system clock. However, the halt release (restart of operation) is performed synchronously with the system clock.

(Note) In this mode, only the external interrupt requests ($\overline{\text{NMI}}$, INT0) are enabled during the halt interval in the IDLE1 mode.
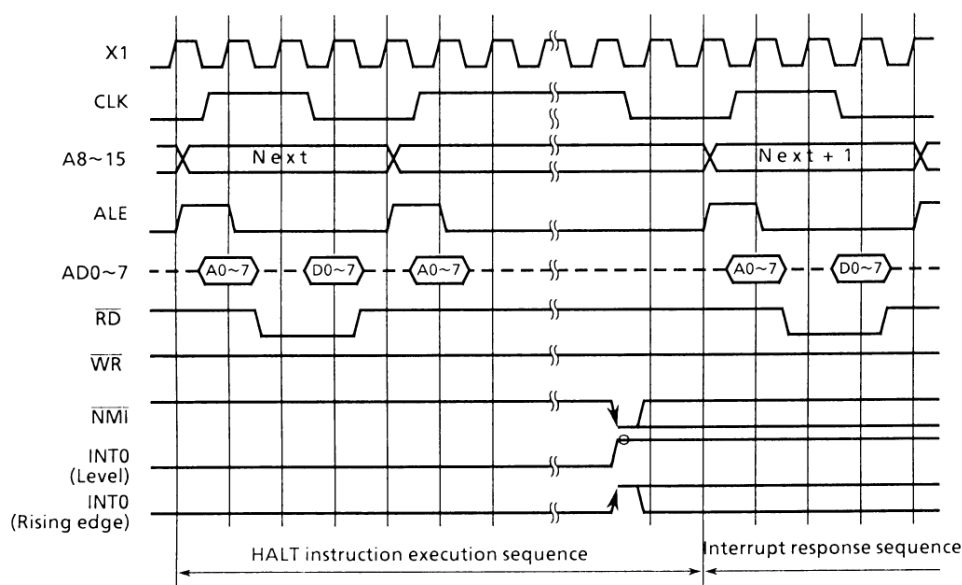
Figure 3.4 (3). Halt Release Timing Using Interrupts in the IDLE1 Mode

3.4.3 IDLE2 Mode

The timing used for releasing the halt state by interrupts in the RUN/IDLE2 mode is shown in Figure 3.4 (2).

The timing for releasing the halt state by interrupts in the IDLE2 mode is the same as in the RUN mode. However, the internal operating mode of the MCU differs. In the RUN mode, only the execution of instruction by the CPU is halted, but the system clock is still supplied to all MCUs. In the IDLE2 mode, however, the system clock is supplied only to specific I/O. Because of that, the power consumption in the halt state of the IDLE2 mode is about 1/3 or less of that of the RUN mode.

In the IDLE2 mode, the system clock is supplied to the following built-in I/O.

- 8-bit timers
- 16-bit timers
- Watchdog Timer

3.4.4 STOP Mode

The timing for releasing the halt with state by interrupts in the STOP mode is shown in Figure 3.4 (4).

In the STOP mode, all internal circuits stop, including the internal oscillator. When the STOP mode is activated, all but certain pins are isolated from the MCU by being set to high impedance.

The state of each pin in the STOP mode is shown in Table 3.4 (1). The status in effect before the halt state continues if WDMOD <DRVE> (drive enable: bit 0 of memory address FFD2H) of the built-in I/O register is set to "1". This register is cleared to "0" by resetting.

The internal oscillator starts first when the CPU receives an interrupt request; however, to allow oscillation to stabilize, the system clock starts its output after the time set by the warming-up counter. WDMOD <WARM> (warming-up: bit 4 at the memory address FFD2H) is used to set the warming-up time. Clearing this bit to "0" sets the warming-up time to the time required for 2^{14} clock oscillations. Setting this bit to "1" sets the warming-up time to the time required for 2^{16} clock oscillations. This bit is initialized to "0" by resetting.
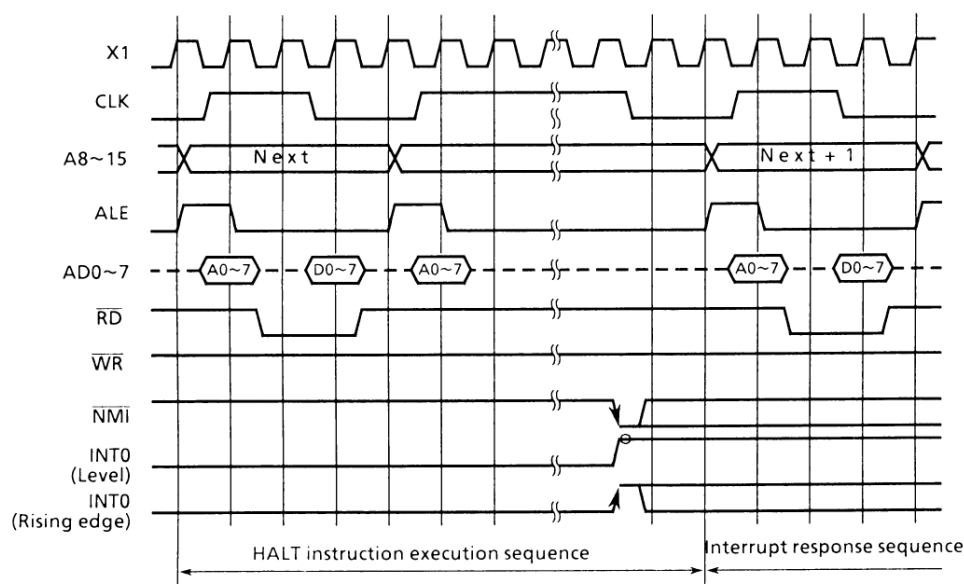
Figure 3.4 (4). HALT Release Timing Using Interrupt in STOP Mode

The internal oscillator can be also restarted by the input of the $\overline{\text{RESET}}$ signal "0" to the CPU; however, the warming-up counter does not operate to permit operation to start quickly immediately after power-on. 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 must at "0" long enough to release the halt state in the STOP mode.

Table 3.4 (1) STOP Mode Pin Status

	In/Out	DRVE = 0	DRVE = 1
P0	Input mode Output mode	– –	– Output
P1	Input mode Output mode	– –	– Output
P20	Input mode Output mode	Input –	Input Output
P21 ~ 23	Input mode Output mode	– –	Input Output
P24 ~ 25	Input pin Output pin	– –	Input* Output
P26 ($\overline{\text{NMI}}$) P27 (INT0)	Input pin Output pin	Input –	Input Output
P30 ~ P31	Output mode	–	Output
P32 ~ P33	Output mode	–	Output
D - A0, D - A1	Output mode	0V	0V
ALE CLK $\overline{\text{RESET}}$ X1 X2	Output pin Output pin Input pin Input pin Output pin	"0" – Input – "1"	"0" "1" Input – "1"

*:   This pin remains intermediately biased in the zero-cross detect mode.
–:   Input mode/input pin indicates that input is invalid.
Input: Input is enabled.
Input: The input gate is working. Fix the input voltage at "0" or "1" to prevent the pin floating.
Output: Output status.

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

Halt mode			RUN	IDLE2	IDLE1	STOP
WDMOD <HALTM1, 0>			00	11	10	01
Operation Block	CPU		Stopped			
	I/O port		Holds status before HALT instruction.			See Table 3.4 (2).
	8-bit timer		Operation			Stopped
	16-bit timer					
	Watchdog timer					
	A/D converter					
	D/A converter		Holds status before HALT instruction.			See Table 3.4 (2).
	Interrupts controller		Operating			
Halt Releasing Source	Interrupt	NMI	O	O	O	O
		INTWD	O	O	–	–
		INT0	O	O	O	O
		INTT0	O	O	–	–
		INTT1	O	O	–	–
		INTAD	O	–	–	–
		INTT2	O	O	–	–
		INTT3	O	O	–	–
		INTT4	O	O	–	–
		INT1	O	O	–	–
		INTT5	O	O	–	–
		INT2	O	O	–	–
	RESET		O	O	O	O

O: Can be used for HALT release.
–: Cannot be used for HALT release.

3.5 Port Functions

The TMP90C846 has a total of 28 I/O port pins. These ports pins can be used not only for the general-purpose I/O function but also for the I/O function of the internal CPU and I/O. The functions of each port pin are shown in Table 3.5.

Table 3.5 Port Functions

Port name	Pin name	No. of pins	Direction	Direction set unit	Resetting status	Internal function pin name
Port 0	P00 ~ P07	8	I/O	Bit	Input	AD0 ~ AD7
Port 1	P10 ~ P17	8	I/O	Bit	Input	A8 ~ A15
Port 2	P20	1	I/O	Bit	Input	ADS
	P21	1	I/O	Bit		TO1
	P22	1	I/O	Bit		TO3
	P23	1	I/O	Bit		TO4
	P24	1	I/O	Bit		INT1/TI4
	P25	1	I/O	Bit		INT2/TI5
	P26	1	I/O	Bit		$\overline{\text{NMI}}$
	P27	1	I/O	Bit		INT0
Port 3	P30	1	Input	–	Input	AN0
	P31	1	Input	–	Input	AN1
	P32	1	Output	–	Output	$\overline{\text{RD}}$
	P33	1	Output	–	Output	$\overline{\text{WR}}$

These port pins function as the general-purpose input/output pins by resetting. All port pins that can be programmed for input or output are set as input ports. The port pins must be set for the internal functions by a program.

3.5.1 Port 0 (P00 ~ P07)

Port 0 is the 8-bit general-purpose I/O port whose I/O function is specified by the control register P0CR in bit basis. A reset operation clears all bits of the control register (P0) to "0" and sets Port 0 to the input mode. The contents of the output latch register become undefined.

In addition to the general-purpose I/O port function, Port 0 also functions as an address/data bus (AD0 ~ AD7). When the external memory is accessed, Port 0 automatically functions as the address/data bus and P0CR is cleared to "0". Therefore, to use port 0 as an output port again after the external access, it is necessary to set the P0CR to "1" again.

Figure 3.5 (1). Port 0

3.5.2 Port 1 (P10 ~ P17)

Port 1 is the 8-bit general-purpose I/O port whose I/O function is specified by the control register P1CR in bit basis.

A reset operation clears all bits of the output latch (P1) and the control register (P1CR) to "0"and sets Port 1 to the input mode.

In addition to the general-purpose I/O port function, Port 1 also functions as the address bus (A8 ~ A15). The function is selected by setting the external extension control register IRFL<EXT> to "1" and setting P1CR to the output mode. When the P1CR cleared to "0", Port 1 is set to the input mode, regardless of the external extension control register <EXT> value. Resetting clears <EXT> to "0" and sets Port 1 to the general-purpose I/O port mode.

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 (output latch register is undefined)							
	Function	Becomes address data bus during external memory access or $\overline{EA} = 0$							

Port 0 control register

		7	6	5	4	3	2	1	0
P0CR (FFC1H)	bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
	Read/Write	W							
	Resetting value	0							
	Function	0 : IN 1 : OUT (I/O specification in bit)							

Read modify write are not possible.

Port 0 I/O settings.

0	input
1	output

Port 1 register

		7	6	5	4	3	2	1	0
P1 (FFC2H)	bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
	Read/Write	R / W							
	Resetting value	0 (Input mode)							
	Function	Becomes the address bus when $\overline{EA} = 0$ or EXT·P1XC = 1. P1XC : Bit X of P1CR							

Port 1 control register

		7	6	5	4	3	2	1	0
P1CR (FFC3H)	bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
	Read/Write	W							
	Resetting value	0							
	Function	0 : IN 1 : OUT (I/O specification in bit)							

Read modify write are not possible.

Port 1 I/O and address bus settings.

P1CR <P1XC> \ IRFL <EXT>	0	1
0	Input port	Input port
1	Output port	Address bus

Note) Settings in bit are possible.
P1CR <P1 x C> is bit x of P1CR

		7	6	5	4	3	2	1	0
IRFL (FFC4H)	bit Symbol			IRF0	IRFT0				EXT
	Read/Write			R					W
	Resetting value			0					0
	Function			Interrupt Request Flag 1 : Active interrupt request					P1 control 0: IN/OUT 1 : IN/ address

Read modify write are not possible.

Figure 3.5 (3). Registers for Port 0/1
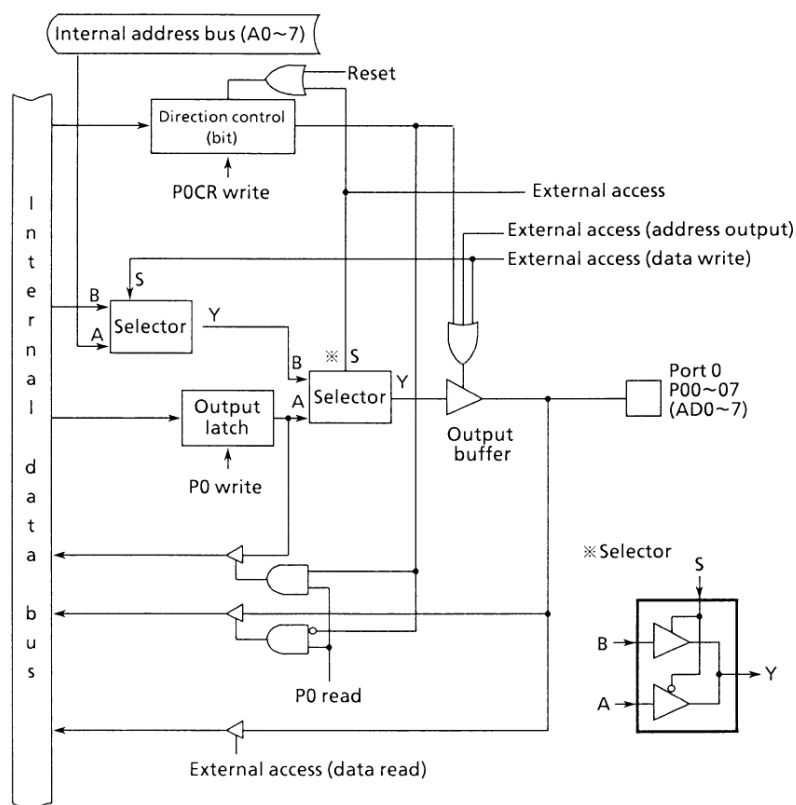
3.5.3 Port 2 (P20 ~ P27)

Port 2 is a 8-bit general-purpose I/O port whose I/O function is specified by the control register P2CR in bit basis.

A reset operation clears all bits of the output latch (P2) and control register (P2CR) to "0" and sets Port 2 to the input mode.

In addition to its I/O port function, Port 2 also functions as an external A/D conversion start pin, timer output pin, timer/ event counter clock input pin and external interrupt request pin.

(1) P20 (ADS)

P20 is a general-purpose I/O port used both as the external A/D conversion start: ADS pin.

The Port 2 function register P2FR <ADSE> is used to set P20 to function as the ADS pin.

P20 is set as the ADS input pin by writing <ADSE>= "1".

P20 is set as the ADS input pin by writing <ADSE> = "1".

A reset operation clears P2CR <P20C> and P2FR <ADSE> to "0" and sets P20 to input port.
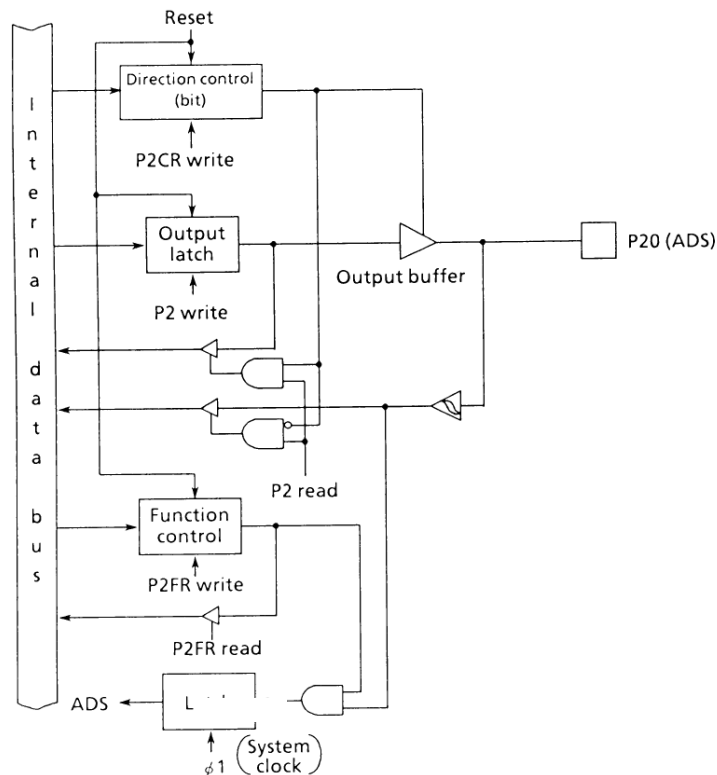
(2) P21 (TO1), P22 (TO3), P23 (TO4)

P21 - P23 are general-purpose I/O ports also used as timer output pins.
The port 2 function register P2FR <TO1E, TO3E, TO4E> and port 2 control register P2CR <P21C,

P22C, P23C> are used to set the timer output pin.
The timer output pin is set by writing "1" to both the control register and function register.
A reset operation clears P2CR <P21C, P22C, P23C> and P2FR <TO1E, TO3E, TO4E> to "0" and sets P21 - P23 to the input port.
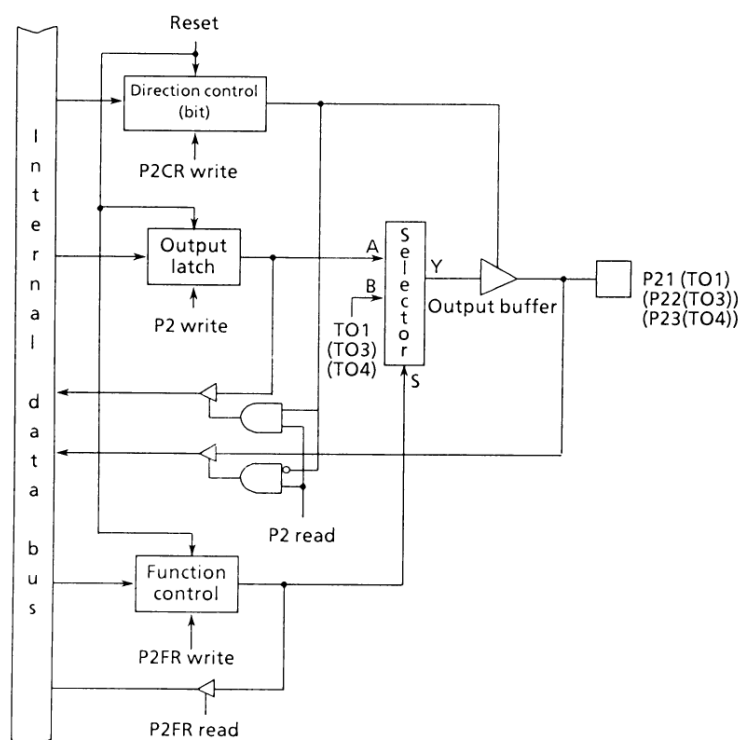
(3)   P24 (INT1/TI4), P25 (INT2/TI5)

P24 and P25 are general-purpose I/O ports, also used both as external interrupt request input pins (INT1, INT2) and timer/event counter clock input pins (TI4, TI5).

These ports have built-in zero cross detection circuits and can be used for zero cross detection by connecting an external capacitor. The zero cross detection function is enabled by writing "1" to <ZCE1, ZCE2> of the Port 2 function register P2FR.

A reset operation clears P2CR <P24C, P25C> and P2FR <ZCE1, ZCE2> to "0" and sets P24 - P25 to the input port.
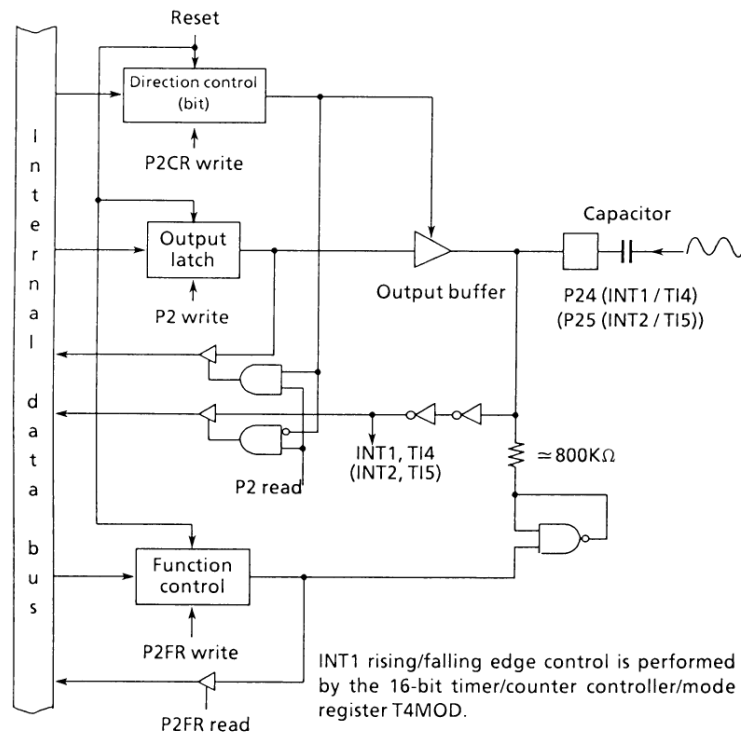
(4) P26 ($\overline{\text{NMI}}$)

P26 is a general-purpose I/O port, also used as the non-maskable interrupt $\overline{\text{NMI}}$ input pin. The Port 2 function register P2FR <NMIC> is used to set the $\overline{\text{NMI}}$ pin mode.
It is necessary to write "1" to <NMIC> when P26 is

used as the $\overline{\text{NMI}}$ input pin (this does not depend on the Port 2 control register P2CR <P26C>).
Once set the $\overline{\text{NMI}}$ input pin mode, P26 can only be returned to the general-purpose I/O port mode by resetting.
A reset operation clears P2CR <P26C> and P2FR <NMIC> to "0" and sets P26 to the input port.
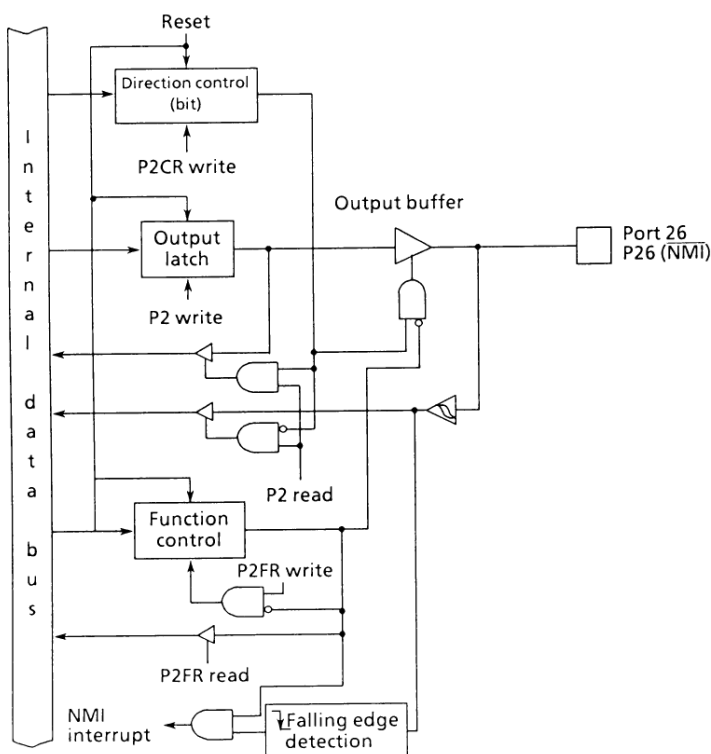
(5)   P27 (INT0)

P27 is a general-purpose I/O port, also used as the external interrupt request input pin INT0.

INT0 can be set for either "H" level interrupt or rising edge interrupt with the Port 2 function register P2FR <EDGE>.

Port 2 register

P2 (FFC6H)		7	6	5	4	3	2	1	0
	bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R / W							
	Resetting value	0 (Input mode)							
	Function	(INT0)	(NMI)	(INT2/TI5)	(INT1/TI4)	(TO4)	(TO3)	(TO1)	(ADS)

Port 2 function register

P2CR (FFC7H)		7	6	5	4	3	2	1	0
	bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
	Read/Write	W							
Read modify write are not possible.	Resetting value	0							
	Function	0 : IN 1 : OUT (I/O specified in bit)							

Port 2 function register

P2FR (FFC9H)		7	6	5	4	3	2	1	0
	bit Symbol	EDGE	NMIC	ZCE2	ZCE1	TO4E	TO3E	TO1E	ADSE
	Read/Write	R / W							
	Resetting value	0							
	Function	INT0 control 0 : level 1 : ↑ edge	NMI control 0 : Port 1 : NMI	INT2 / TI5 control 1 : ZCD enable	INT1 / TI4 control 1 : ZCD enable	Port / TO4 control 0 : In/Out 1 : In/TO4	Port / TO3 control 0 : In/Out 1 : In/TO3	Port / TO1 control 0 : In/Out 1 : In/TO1	Port / ADS control 0 : Port (ADS disable) 1 : ADS enable

Port 20 ADS control

0	general-purpose I/O port
1	ADS enable

Port 21 - 23 timer output control

0	input port/output port
1	input port/timer output

Port 24 - 25 zero-cross control

0	zero-cross disable
1	zero-cross enable

Port 26 NMI control

0	input port/output port
1	NMI enable

INT0 control

0	"H" level detect interrupt
1	rising edge detect interrupt

Figure 3.5 (6). Registers for Port 2

3.5.4 Port 3 (P30 ~ P33)

Port 3 is the 4-bit general-purpose I/O port. P30 and P31 are used for input only; P32 and P33 are used for output only.

P30 and P31 function both as a 2-bit input port and A/D converter analog input pins AN0 and AN1.

P32 and P33 are used both as a 2-bit output port and for the external memory control functions $\overline{RD}$ and $\overline{WR}$. The output latch for P32 and P33 is set to "1" by resetting and outputs "1".

When the external memory is accessed, P32 and P33 automatically function as the memory control pins $\overline{RD}$ and $\overline{WR}$. When the external memory is accessed, P32 and P33 automatically function as the memory control pins $\overline{RD}$ and $\overline{WR}$. When the internal memory is accessed, P32 and P33 function as the output port. Therefore, for accessing the external memory, leave the output latch register of P32 ($\overline{RD}$) and P33 ($\overline{WR}$) to "1", a value immediately set after resetting.

P3 (FFC8H)		7	6	5	4	3	2	1	0
	bit Symbol					P33	P32	P31	P30
	Read/Write					R / W		R	
	Resetting value					1		input mode	
	Function					($\overline{WR}$)	($\overline{RD}$)	(AN1)	(AN0)

Figure 3.5 (7). Port 3

3.6 Timers

The TMP90C846 has four 8-bit timers and one multi-function 16-bit timer/event counter.

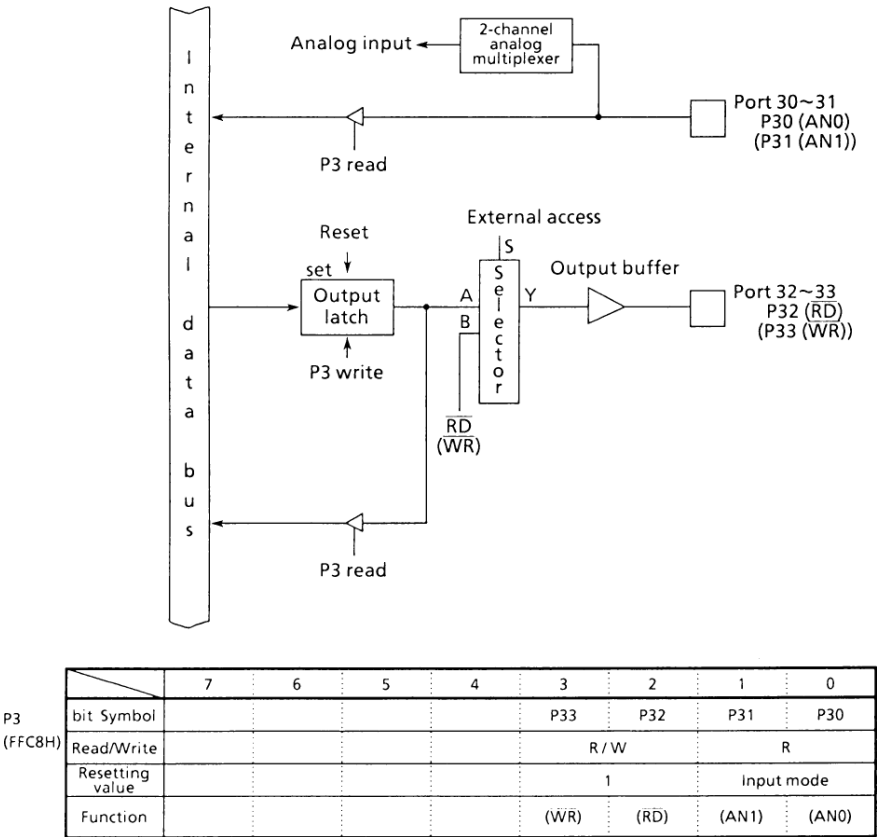These four 8-bit timers can be operated independently or cascade connected to form two 16-bit timers. The 8-bit timers have the following four operating modes.

- 8-bit interval timer modes (4)
- 16-bit interval timer modes (2)
  - The two above can be combined (8 bits x 2, 16 bits x 1)
- 8-bit Programmable Pulse Generation (PPG; variable duty at variable interval) output modes (2)
- 8-bit PWM (pulse width modulation: variable duty at fixed interval) output mode (2)

The multi-function 16-bit timer/event counter has the following six operating modes.

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit PPG (variable duty at variable interval) output mode
- Frequency measurement mode
- Pulse width measurement mode
- Time deviation measurement mode

3.6.1 8-bit Timers

The TMP90C846 has four 8-bit interval timers (timer 0, 1, 2 and 3), each of which can be operated independently. Timer 0 and 1, or Timer 2 and 3 can be cascade-connected and used as 16-bit interval timers.

The block diagram of the 8-bit timers (Timers 0 and 1) is shown in Figure 3.6 (1).

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

Each interval timer comprises an 8-bit up-counter, an 8-bit comparator, and an 8-bit timer register. One timer flip-flop (TFF1, TFF3) is provided for each pair of Timer 0/1 and Timer 2/3.

The internal clocks øT1, øT16, and øT256 used as the input clocks to the interval timers are obtained from the 9-bit prescaler shown in Figure 3.6 (2).

The operating modes and timer flip-flops for the 8-bit timers are controlled by four control registers (TCLK, TFFCR, TMOD and TRUN).
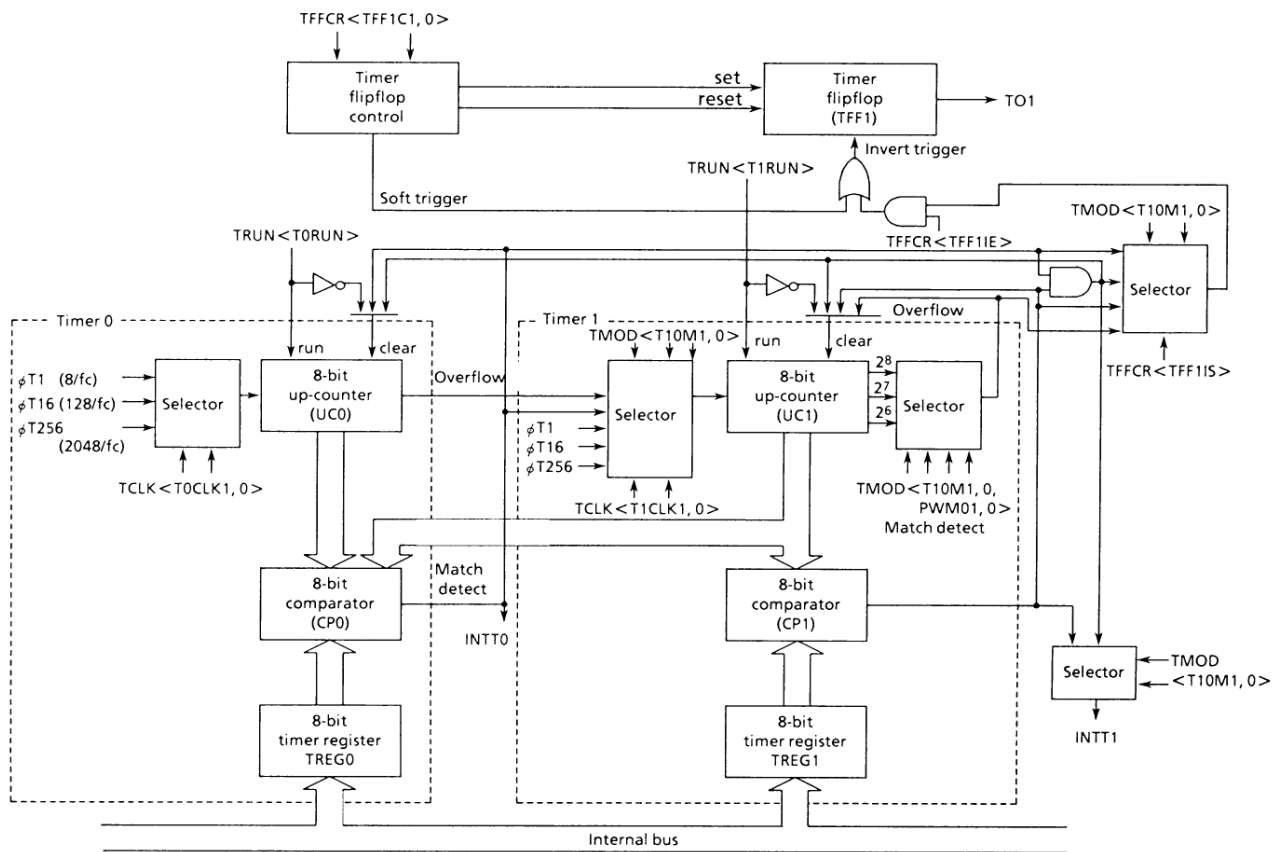
Figure 3.6 (1). 8-bit Timers Block Diagram (Timers 0 and 1)

① Prescaler

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

The three clocks øT1, øT16 and øT256 are used for the 8-bit timers.

This prescaler is run and stopped with the timer operation control register TRUN <PRRUN>. Setting <PRRUN> to "1" starts counting and setting <PRRUN> to "0" stops and clears the prescaler to "0" and stops. Resetting clears <PRRUN> to "0", which clears and stops the prescaler.

Input clock	fc	Cycle	
		8MHz	10MHz
ϕT1 (8/fc)		1.0μs	0.8μs
ϕT16 (128/fc)		16μs	12.8μs
ϕT256 (2048/fc)		256μs	204.8μs
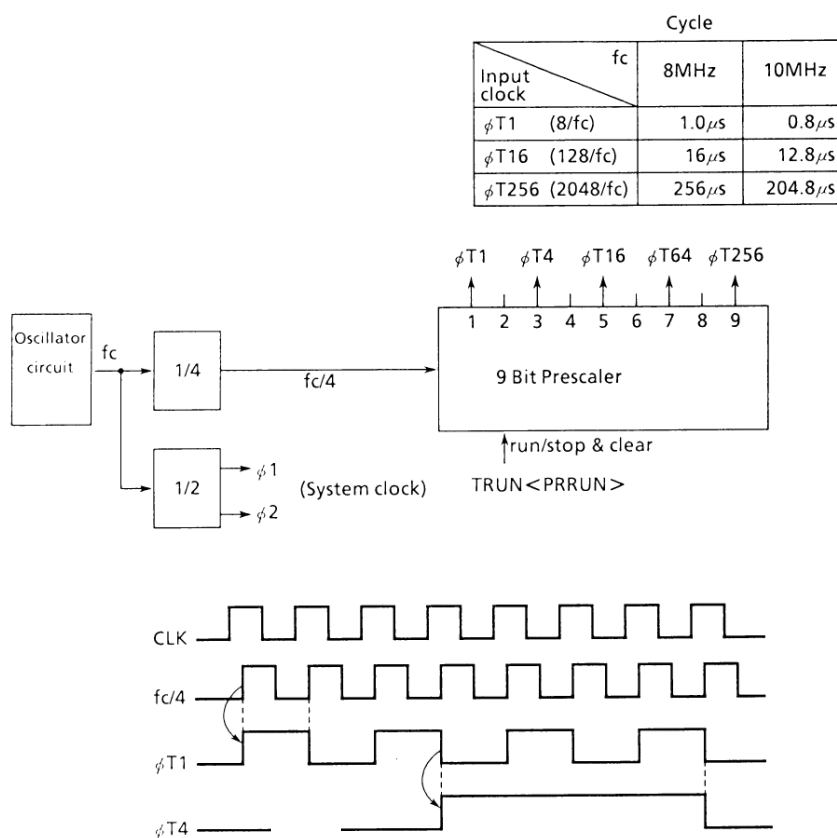
Figure 3.6 (2). Prescaler

② Up-counter

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

The input clock of Timers 0 and 2 can be selected from the three internal clocks øT1 (8/fc), øT16 (128/fc), øT256 (2048/fc) in accordance with the TCLK setting value.

Example: Setting TCLK <T0CLK1, 0> = 0, 1 selects øT1 as the input clock for Timer 0.

The input clock selection for Timers1 and 3 differs depending on the operating mode. When the 16-bit timer mode is set, the overflow output of Timer 0 or 2 is used as the input cock, regardless of the TCLK register setting.

In the other modes, the input clock is selected among the internal clocks øT1, øT16, øT256, and the output of the Timers 0 and 2 comparator (match detection) by setting the TCLK register.

Example: If TMOD <T10M1, 0> = 0, 1, the overflow output of Timer 0 is used as the input clock to Timer 1 (16-bit timer).
If TMOD <T10M1, 0> = 00 and T01MOD <T1CLK1, 0> = 0, 0, and <T1CLK1, 0> = 0, 1, øT1 is used as the input clock to Timer 1 (8-bit timer).

The TMOD register is also used to set the operating mode. Resetting initializes to TMOD <T10M1, 0> = 0, 0/TMOD <T32M1, 0> = 0, 0; therefore, 8-bit timer mode is set.

Each up-counter can be run, stopped and cleared with the timer control register TRUN. Resetting stops all timers and clears all up-counters.

		7	6	5	4	3	2	1	0
		Timer 2, 3				Timer 0, 1			
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		00 : — 01 : 2^{6-1} PWM 10 : 2^{7-1} cycle 11 : 2^{8-1}		00 : 8bit Timer 01 : 16bit Timer 10 : 8bit PPG 11 : 8bit PWM		00 : — 01 : 2^{6-1} PWM 10 : 2^{7-1} cycle 11 : 2^{8-1}	

PWM1 cycle (timer 1) selection
(Don't care in other than the PWM mode.)

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

Timer 0, 1 operating mode setting

00	8-bit timer × 2 (timer 0, 1)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 1) + 8-bit timer (timer 0)

PWM3 cycle (timer 3) selection
(Don't care in other than the PWM mode.)

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

Timer 2/3 operating mode setting

00	8-bit timer × 2 (timer2, 3)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 3) + 8-bit time (timer 2)

Figure 3.6 (3). 8-bit Timer Mode Register TMOD

		Timer 3		Timer 2		Timer 1		Timer 0	
		7	6	5	4	3	2	1	0
TCLK (FFD8H)	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	Internal clock ϕT16
11	Internal clock ϕT256

Timer 1 input clock

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

Timer 2 input clock

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

Timer 3 input clock

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

(Note) ϕT256 input is possible in the 16-bit timer mode.

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

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

16-bit timer: bits 5, 4
8-bit timer: bits 3, 2, 1, 0

Timer 0 operation

0	Stop and clear
1	Count

Timer 1 operation

0	Stop and clear
1	Count

Timer 2 operation

0	Stop and clear
1	Count

Timer 3 operation

0	Stop and clear
1	Count

16-bit timer (timer 4) operation

0	Stop and clear
1	Count

Prescaler operation

0	Stop and clear
1	Count

Figure 3.6 (5). Timer Control Register TRUN

		7	6	5	4	3	2	1	0
		TFF3				TFF1			
TFFCR (FFD9H)	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 timer3	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 timer 1

Timer flipflop TFF1 inversion signal select

TFFCR <TFF1IS> \ TMOD<T10M 1, 0>	0, 0	0, 1	1, 0	1, 1
0	8-bit timer mode (timer 0)	TFF1 inversion not possible		
1	8-bit timer mode (timer 1)	16-bit timer mode (timer 0 and timer 1)	PPG mode (timer 1)	PWM mode (timer 1)

Timer flipflop TFF1 inversion

0	Disable
1	Enable

Timer flipflop TFF1 control

00	Clears TFF1 to "0".
01	Sets TFF1 to "1".
10	Inverts the TFF1 value (software inversion).
11	Don't care (always read "11").

Timer flipflop TFF3 inversion signal select

TFFCR <TFF3IS> \ TMOD<T32M 1, 0>	0, 0	0, 1	1, 0	1, 1
0	8-bit timer mode (timer 2)	TFF3 inversion not possible		
1	8-bit timer mode (timer 3)	16-bit timer mode (timer 2 and timer 3)	PPG mode (timer 3)	PWM mode (timer3)

Timer flipflop TFF3 inversion

0	Disable
1	Enable

Timer flipflop TFF3 control

00	Clears TFF3 to "0".
01	Sets TFF3 to "1".
10	Inverts the TFF3 value (software inversion).
11	Don't care (always read "11").

Figure 3.6 (6). 8-bit Timer Flip-flop Control Register TFFCR

③ Timer registers

TREG0	TREG1	TREG2	TREG3	
8-bit	8-bit	8-bit	8-bit	
FFD4H	FFD5H	FFD6H	FFD7H	(Note) Write only.

This is an 8-bit register used to set the interval time. When the set value of a timer register matches to that of an up-counter, the match signal of comparators becomes active. When the set value is 00H, the match signal becomes active when an up-counter overflows. When a new value is written to this register, it is immediately input to the comparator.

④ Comparators

When a comparison of an up-counter value and a timer register value show a match, the up-counter is cleared to "0" and an interrupt signal (INTT0 ~ INTT3) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop value is inverted at the same time.

⑤ Timer flip-flop (Timer F/Fs)

This flip-flop is inverted by the match signals (comparator output) of the interval timer. The value can be output to the timer output pins TO1 (also used as P21) and TO3 (also used as P22).
A Timer F/F is provided to each of the Timer 0/1 pair (TFF1) and Timer 2/3 pair (TFF3). The TFF1 value is

output to the TO1 pin and the TFF3 value to the TO3 pin.
The timer F/F is controlled by the timer flip-flop control register TFFCR.
TFF1 (Timer 0/1 timer flip-flop) is used below for explanatory purposes. (Refer to Figure 3.6 (6).)

• TFFCR <FF1IS> is a selection bit for the TFF1 invert signal. In the 8-bit timer mode, <FF1IS> should be to "0" when the match signal from Timer 0 is used, and should be set to "1" when the match signal from Timer 1 is used.
In any other mode, always keep <FF1IS> set to "1". Resetting clears <FF1IS> to "0".
• TFFCR1 <FF1IS> is the TFF1 invert enable bit. Set this bit to "1" to enable inversion and clear to "0" to disable inversion.
Resetting clears <FF1IS> to "0".
• TFFCR <TFF1C 1, 0> is the TFF1 set/reset and software inversion bit. Writing "0, 0" resets TFF1, writing "0, 1" sets TFF1 and writing "1, 0" inverts the TFF1 value. Similarly, the TFF3 is controlled by the upper four bits TFFCR <TFF2C1, 0, TFF3IE, S>.

The following is an explation of the 8-bit timer operation:

(1)   8-bit timer mode

The four interval timers 0, 1, 2 and 3 can be used independently as an 8-bit interval timers. The operation is the same as for all of the timers. Thus, Timer 1 will be used here for explanatory purposes.

① Generating interrupts in a fixed cycle

To use Timer 1 for generating Timer 1 interrupts (INTT1) in a fixed cycle, first stop Timer 1 and then set the operation mode, input clock and cycle to the TMOD, TCLK and TREG1 registers. Next, enable the interrupt INTT1 and then start Timer 1 counting.

Example: Use the following procedure to set the registers to generate Timer 1 interrupts every 40µs (fc = 10MHz).

```
                  MSB          LSB
                  7 6 5 4 3 2 1 0
         ┌ TRUN  ←  – – – – – – 0 –      Stops Timer 1 and clears it to "0"
         │ TMOD  ←  – – – – 0 0 X X      Sets the 8-bit timer mode.
         │ TCLK  ←  – – – – 0 1 – –      Sets the input clock to ∅T1 (0.8 µs @ fc = 10MHz).
         │ TREG1 ←  0 0 1 1 0 0 1 0      Sets 40 µs ÷ ∅T1 = 32H to the timer register.
         │ INTEH ←  – – – – – – – 1      Enables INTT1.
         └ TRUN  ←  – – 1 – – – 1 –      Starts Timer 1.

         (Note) X ; don't care      - ; no change
```

Refer to the table below for selecting the input clock:

Table 3.6 (1) Interrupt Cycle and Input Clock Using 8-bit Timer

Interrupt cycle @fc = 10MHz	Resolution	Input clock
8µs ~ 204.8µs	8µs	∅T1 (8/fc)
12.8µs ~ 3.2768ms	12.8µs	∅T16 (128/fc)
204.8µs ~ 52.42ms	204.8µs	∅T256 (2048/fc)

② Generating pulse at 50% duty

Invert the timer flipflop at a fixed cycle and output the timer flipflop value to the timer output pin (TO1).

Example: Use the following procedure to set the register to output a pulse from the TO1 pin in a 4.8μs cycle at fc = 10MHz. Timer 0 or 1 is used in this case, but Timer 1 will be used for explanatory purposes.

```
                 MSB          LSB
                 7 6 5 4 3 2 1 0
TRUN    ←    – – – – – – 0 –      Stops Timer 1 and clears it to "0".
TMOD    ←    – – – – 0 0 X X      Sets the 8-bit timer mode.
TCLK    ←    – – – – 0 1 – –      Sets the input clock to φT1.
TREG1   ←    0 0 0 0 0 0 1 1      Sets (4.8 μs ÷ φT1) ÷ 2 = 3 to the timer register.
                                  Clears TFF1 to "0" and sets inversion with the match
TFFCR   ←    – – – – 0 0 1 1      signal from Timer 1.

                                  Sets P21 as the TO1 pin.
P2CR    ←    – – – – – – 1 –   ⎫
P2FR    ←    – – – – – – 1 –   ⎬  Starts Timer 1.
TRUN    ←    – – 1 – – – 1 –   ⎭
```

(Note) X ; don't care   - ; no change
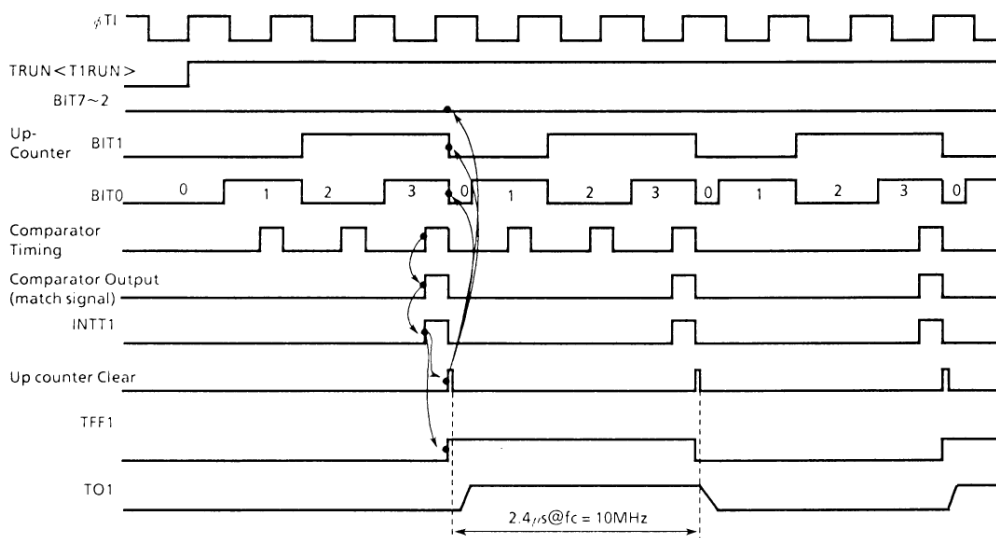
Figure 3.6 (7). Pulse (50% duty) Output Timing Chart

③ Starting Timer 1 counting up with Timer 0 match output

Set the 8-bit timer mode and set the comparator output of Timer 0 as the Timer 1 input clock.
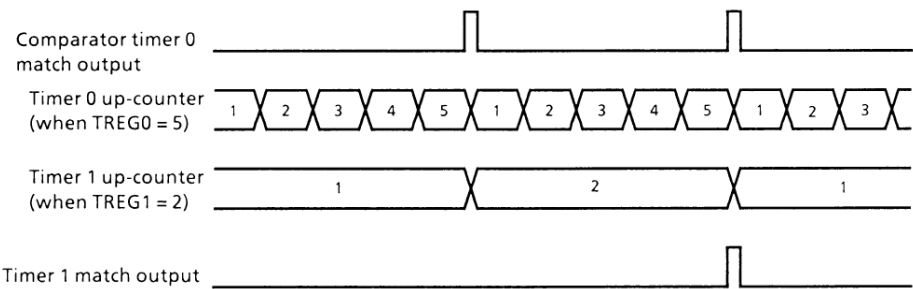
Figure 3.6 (8)

④ Output inversion with software

The timer flip-flop value can be inverted regardless of the timer operation.
Writing "1, 0" to TFFCR <TFF1C1, 0> inverts the TFF1 value; writing "0, 1" into TFFCR <FF3C1, 0> inverts the TFF3 value.

⑤ Initial setting of timer flipflop

The timer flipflop initial value can be set to either "0" or "1", regardless of the timer operation.
For example, write 0, 0 to TFFCR <TFF1C1, 0> to clear TFF1 to "0", and write 0, 1 in TFFCR <TFF1C1, 0> to set TFF1 to "1".

(Note) The timer flipflopand timer register values cannot be read.

(2) 16-bit timer mode

16-bit interval timers can be created by using Timer 0 and 1 as a pair of Timer 2 and 3 as a pair.
The operation of Timer 0 and 1 is the same as that of Timer 2 and 3, so Timer 0 and 1 are used for explanatory purposes.
Timer 0 and 1 can be used as a 16-bit interval timer by connecting them in a cascade configuration and writing "0, 1" to TMOD <T10M1, 0>.
When the 16-bit timer mode is set, the overflow output of timer 0 is used as the Timer 1 input clock, regardless of the TCLK setting value. TCLK sets the Timer 0 input clock. The relationship between the timer (interrupt) cycle and the input clock is shown in Table 3.6 (2).

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

Timer (interrupt) cycle @fc = 10MHz	Resolution	Input clock to Timer 0
8μs ~ 52.43ms	8μs	øT1 (8/fc)
12.8μs ~ 838.86ms	12.8μs	øT16 (128/fc)
204.8μs ~ 13.42s	204.8μs	øT256 (2048/fc)

The timer (interrupt) cycle is set by loading the lower eight bits to the timer register TREG0 and the upper eight bits to the timer register TREG1. In this case, always set TREG0 first. (A comparison is temporarily halted by writing data to TREG0 and a comparison is started by writing data into TREG1).

Example: To generate interrupts INTT1 every 1second
(at fc = 8MHz, set the following values for
timer register TREG0 and TREG1.
When counting by using øT16 (16µs @8MHz),
1s ÷ 16µs = 62500 = F424H
Therefore, set TREG1 = F4H and TREG0 = 24H.

The match signal of timer 0 comparator is output each time the up-counter UC0 matches TREG0. The up-counter UC0 is not cleared bit INTT0 is generated.

The match signal of timer 1 comparator is output at each comparator timing cycle, if the up-counter UC1 matches TREG1. When the match signals of both the Timer 0 and Timer 1 comparators are output at the same time, the up-counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If the inversion is enabled, the value of the timer flipflop TFF1 is inverted then.

	Timer 0			Timer 1		
	INTT0	TO1	Match	INTT1	TO1	Match value
16-bit Timer Mode (count-up Timer 1 by Timer 0 overflow)	Interrupt generated	Output disabled (cannot output a TREG0 match)	TREG0 (continue count-up)	Interrupt generated	Output enable (can output when both Timer 0 and Timer 1 match)	TREG1 * 2^8 + TREG0 16-bit) (Cleared both match)
8-bit Timer Mode (count-up Timer 1 by Timer 0 match)	Interrupt generated	Output enable (either Timer 0 or Timer 1)	TREG0 (clears on match)	Interrupt generated	Output enable (either Timer 0 or Timer 1)	TREG1*TREG0 (multiplication value) (clears on match)

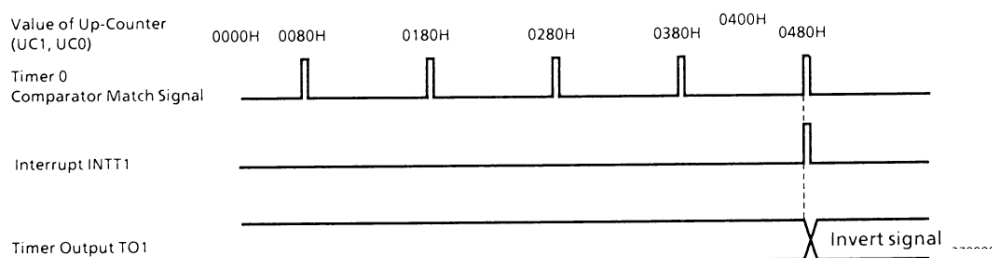Example: When TREG1 = 04H and TREG0 = 80H

Figure 3.6 (9)

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

Timer 1 or Timer 3 can be used to output pulse at any frequency and duty rate. The output pulse can be either low-active or high-active.

Timer 0 and/or Timer 2 cannot be used in this mode. For Timer 1, pulse is output to TO1 (also used as P21); for Timer 3, it is output to TO3 (also used as P22).
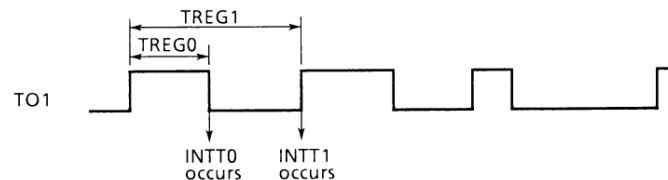
Timer 1 is used for explanatory purposes (operation is the   same for Timer 3).
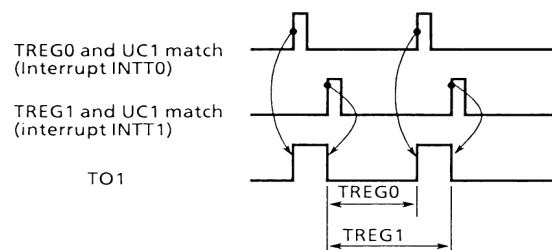
This mode outputs a programmable pulse by inverting the timer output each time the 8-bit up-counter 1 (UC1) matches the timer register TREG0 or TREG1.

It is necessary, however, to satisfy the condition (TREG0 set value) < (TREG1 set value).

The up-counter (UC0) of Timer 0 cannot be used in this mode. Timer 0 can be used for counting by setting TRUN <T0RUN> to "1".

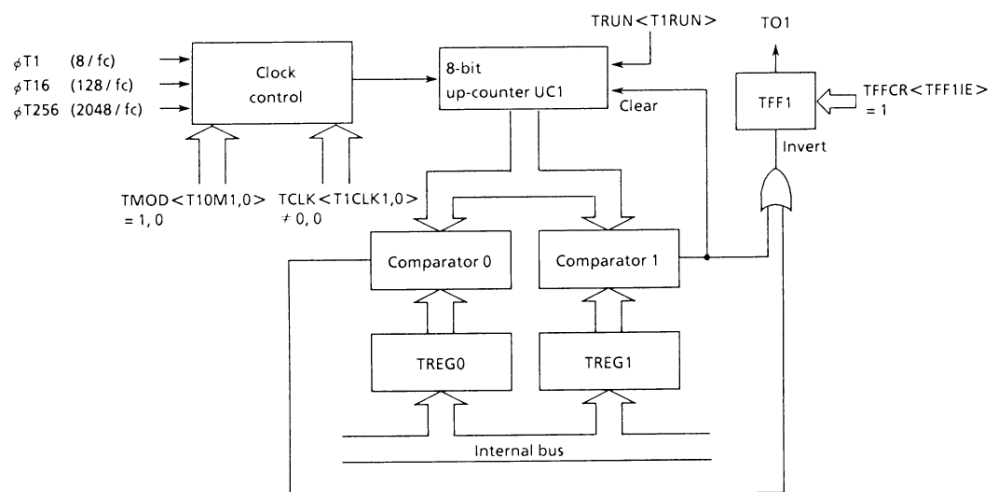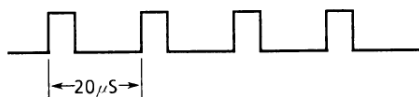Figure 3.6 (10) shows the block diagram of the PPG mode.

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

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

• Determine the set values of the timer registers.

The frequency is set to 50kHz, by making a
t = 1/50kHz = 20µspulse cycle.
When øT1 = 1µs (@ 8MHz) is used,
   20µs ÷ 1µs = 20
Therefore, the timer register 1 (TREG1) is set to
TREG1 = 20 = 14H
Next, to set a duty to 1/4,
t x 1/4 = 20µs x 1/4 = 5µs
   5µs ÷ 1µs = 5
Therefore, set timer register 0 (TREG0) to TREG0 = 5 = 05H.

|		
TRUN ← - - - - - - 0 0	Stops Timer 0 and Timer 1, and clears them to "0".
TCLK ← - - - - 0 1 x x	Sets the input clock to øT1.
TMOD ← - - - - 1 0 x x	Sets the 8-bit PPG mode.
TFFCR ← - - - - 0 1 1 1	Clears the output and enables the inversion by Timer 1.
	When set to 00, a negative logic pulse is obtained.
	Writes 5H.
TREG0 ← 0 0 0 0 0 1 0 1	Writes 14H.
TREG1 ← 0 0 0 1 0 1 0 0	
P2CR ← - - - - - - 1 -	Sets P21 as the TO1 pin.
P2FR ← - - - - - - 1 -	
TRUN ← - - 1 - - - 1 1	Starts Timer 1.

(注)   ×；don't care   -；no change

Cautions for PPG Output

The PPG output is possible by updating the contents of TREG (Timer Register); however, a caution is required concerning the

TREG update timing in relation to the PPG pulse width setting.

Example: PPG output by 8-bit Timer 0, 1
TREG0 = pulse width TREG1 = cycle
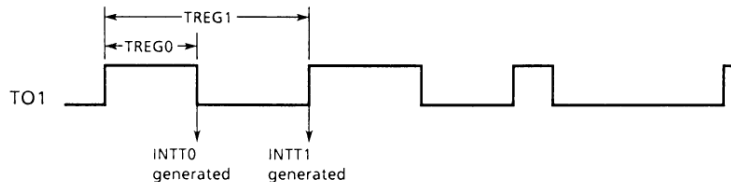
The pulse width is normally changed with the interrupt (INTT1) processing routine for each timer cycle. When a next pulse width (value written to TREG0) is small, a problem will occur if the timer has already exceeded the TREG0 value. Thus, it is recommended that the following decision be made during INTT0 and INTT1 interrupt processing.

INTT0 processing routine: Update the current TREG0 value only when writing a smaller value.

INTT1 processing routine: Update the current TREG0 value only when writing a larger value.

The TREG contents cannot be read; therefore, when making decisions as the above, it is necessary to store the TREG contents in RAM or a register.

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

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

Timer 1 outputs to the TO1 pin (also used as P21); Timer 3 outputs to the TO3 pin (also used as P22).

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

Timer 1 (PWM1) is used for explanatory purposes (The operation for timer 3 is the same.)

The timer output is inverted when up-counter (UC1) value matches the set value of timer register TREG1, and when a counter overflow of $2^n - 1$ (specify $n = 6, 7$ or 8 with TMOD <PWM01,00>) occurs. The up-counter UC1 is cleared by a counter overflow of $2^n - 1$.

The following conditions must be satisfied when the PWM mode is used.

(Set value of timer register) $< (2^n - 1$ counter overflow setting value)
(Set value of timer register) $\neq 0$
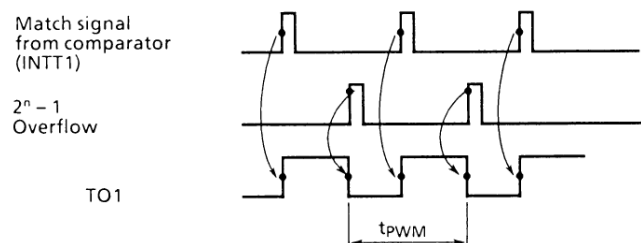(For example, $n = 6$: 6-bit PWM; $n = 7$: 7-bit PWM.)

Figure 3.6 (11) shows the block diagram of this mode.
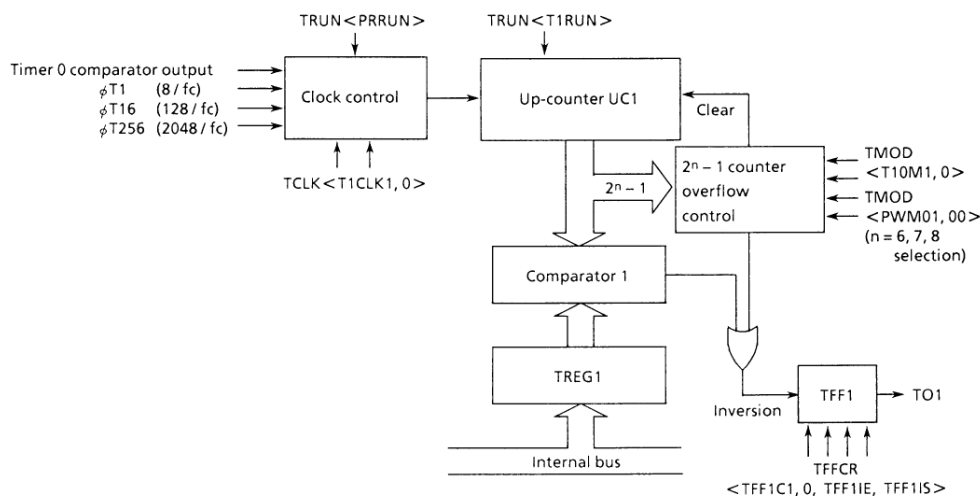
Figure 3.6 (11). 8-bit PWM Mode Block Diagram

Example: Outputting the PWM shown below to the TO3 pin (P22) by using Timer 3 at fc = 10MHz.
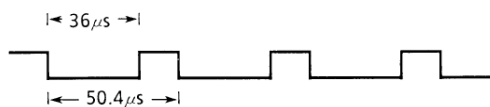
Setting a PWM cycle of 50.4μs with T1 = 0.8μs (@fc = 10MHz),

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

Therefore, n = 6 is set. (TMOD1, 0 = 01)

The "Low" level cycle is 36μs; therefore, at øT1 = 0.8μs, $36\mu s \div 0.8\mu s = 45 = 2DH$ is set to TREG3.

TRUN	←	–	–	–	–	0	–	–	–	Stops Timer 3.
TCLK	←	0	1	–	–	–	–	–	–	Sets the input clock to øT1.
TMOD	←	1	1	0	1	–	–	–	–	Sets the PWM mode cycle = 2^6 - 1.
TFFCR	←	0	0	1	1	–	–	–	–	Sets the output initial value to 0 ("L" level).
TREG3	←	0	0	1	0	1	1	0	1	Writes 2DH.
P2CR	←	–	–	–	–	–	1	–	–	Sets P22 as the TO3 pin.
P2FR	←	–	–	–	–	–	1	–	–	
TRUN	←	–	–	–	–	1	–	–	–	Starts Timer 3.

(Note) × ; don't care   – ; no change

Table 3.6 (3) PWM Cycle and $2^n - 1$ Counter Setting

	Expression	PWM cycle (@fc = 10Mhz)		
		øT1 (8/fc)	øT16 (128/fc)	øT16 (128/fc)
$2^6 - 1$	$(2^6 - 1) \times øTn$	50.4μs	806.4μs	12.9μs
$2^7 - 1$	$(2^7 - 1) \times øTn$	101.6μs	1625.6μs	26.0ms
$2^8 - 1$	$(2^8 - 1) \times øTn$	204.0μs	3264.0μs	52.2ms

Cautions for PWM Output

The TMP90C846 is capable of the PWM output with a 8-bit timer; however, it is necessary to pay a caution when the

PWM pulse width is changed. The following example is used for explanatory purposes.

Example: Using an 8-bit timer for PWM output.
TREG1 = pulse width Cycle: fixed ($2^6 - 1$, $2^7 - 1$, $2^8 - 1$)
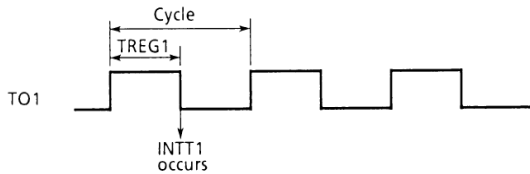
During the PWM output, INTT1 is generated as matching with TREG1. This interrupt cannot be used to change the pulse width directly.
(A new value of TREG1 may cause another inversion within the PWM cycle, when new value is larger than current one.)

One method of solving this problem, when the pulse width is changed, is to temporarily stop the timer with the INTT1 processing routine, and change the TREG1 value; then set Timer Out to "1" and restart the timer. In this case, the output waveform is disrupted when the pulse width is changed, but the usage is still possible with systems that can tolerate this situation.

(5)　　Table 3.6 (4) shows the list of 8-bit timer modes.

Table 3.6 (4) Timer Mode Setting Register

Register name	TMOD		T1CLK		TFFCR
Name of bit in register	T01M (T32M)	PWM1 (PWM3)	T1CLK (T3CLK)	T0CLK (T2CLK)	FF1IS (FF3IS)
Function	Timer Mode	PWM cycle	Upper timer input clock	Lower timer Input clock	Timer F/F invert signal select
16-bit timer mode	01	–	–	øT1, øT16, øT256 (01, 10, 11)	1 (*)
8-bit timer x 2 channels	00	–	Lower timer match øT1, øT16, øT256 (01, 01, 10, 11)	øT1, øT16, øT256 (01, 10, 11)	0: lower timer output 1: upper timer output
8-bit PPG x 1 channel	10	–	øT1, øT16, øT256 (01, 10, 11)	–	1
8-bit PWM x 1 channel	11	$2^6 - 1$, $2^7 - 1$, $2^8 - 1$ (01, 10, 11)	øT1, øT16, øT256 (01, 10, 11)	–	1–
8-bit timer x 1 channel	11	–	øT1, øT14, øT16 (01, 10, 11)	øT1, øT16, øT256 (01, 10, 11)	Output disabled

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

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

The TMP90C846 has one multi-function 16-bit timer/event counter with the following modes:

- 16-bit timer
- 16-bit event counter
- 16-bit programmable pulse generation (PPG)

- Frequency measurement
- Pulse width measurement
- Time difference measurement

　　A block diagram of the 16-bit timer/event counter is shown in Figure 3.6 (12).
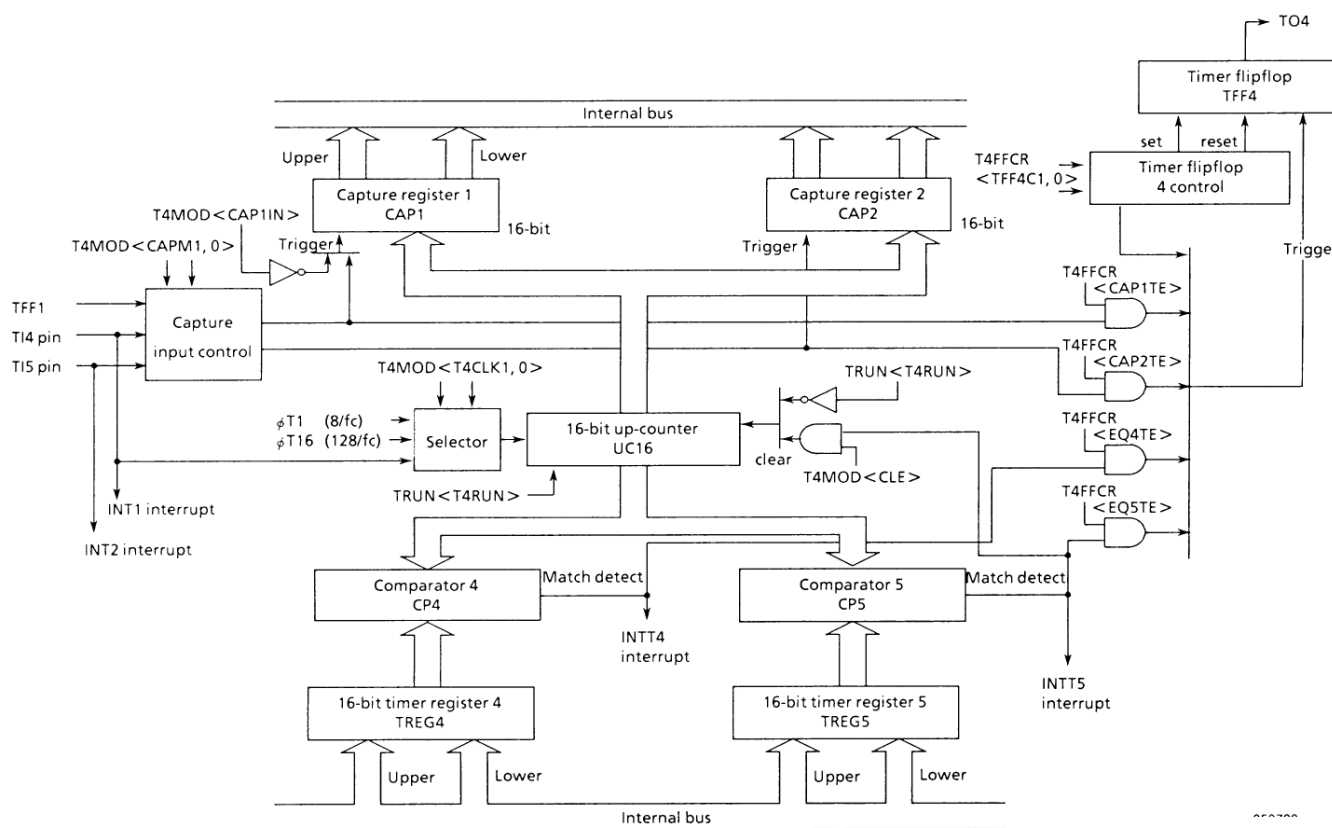
Figure 3.6 (12). 16-Bit Timer/Event (Timer 4) Block Diagram

　　A timer/event counter comprises a 16-bit up-counter, two 16-bit timer registers, and two 16-bit capture registers, two comparators, a capture input control, a timer flipflop its and the control circuit. A timer/event counter is controlled by five control registers.

- Timer mode register T4MOD
- Timer flip-flop control register T4FFCR
- Timer control register TRUN
- Timer flip-flop output port P2 control register P2CR
- Timer flip-flop output port P2 function register P2FR

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 is rising edge 01： TI4 ↑　TI5 ↑ 　INT1 is rising edge 10： TI4 ↑　TI4 ↓ 　INT1 is falling edge 11： TFF1 ↑　TFF1 ↓ 　INT1 is rising edge		1: UC16 Clear Enable	Timer 4 source clock 00: TI4 01: φT1 (8/fc) 10: φT16 (128/fc) 11: disable	

Up-counter UC16 clear control

0	Clear disable
1	Clear match with TREG5

Capture control/INT1 interrupt control

	Capture control	INT1 control
00	Capture disable	Interrupt generated at TI4 ⌐／ (INT1)　input rising edge.
01	CAP1 at TI4 rise CAP2 at TI5 rise	
10	CAP1 at TI4 rise CAP2 at TI4 fall	Interrupt generated at TI4 ↘ (INT1)　input falling edge
11	CAP1 at TFF1 rise CAP2 at TFF1 fall	Interrupt generated at TI4 ／ (INT1)　input rising edge.

Software capture trigger

0	Load up-counter value to CAP1.
1	Always read as "1".

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 invert trigger 0: Disable trigger 1: Enable trigger				00: Clear TFF4 01: Set TFF4 10: Invert TFF4 11: Don't care	
				When up-counter value is loaded to CAP2.	When up-counter value is loaded to CAP1.	When up-counter and TREG5 match	When up-counter and TREG4 match	※ Always read as "1, 1"	

Timer flipflop 4 (TFF4) control

00	Clears TFF4 to "0"
01	Sets TFF1 to "1"
10	Inverts the TFF4 value (software inversion)
11	Don't care Always read as "1, 1"

Timer flipflop 4 (TFF4) invert trigger

0	Trigger disable
1	Trigger enable

CAP2TE: When up-counter value is loaded to CAP2.
CAP1TE: When up-counter value is loaded to CAP1.
EQ5TE : When up-counter and TREG5 match
EQ4TE : When up-counter and TREG4 match

Figure 3.6 (14). 16-bit Timer/Event Counter Timer Flip-flop 4 Control Register

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

Prescaler operation

0	Stop and clear.
1	Run

16-bit timer (timer 4) operation

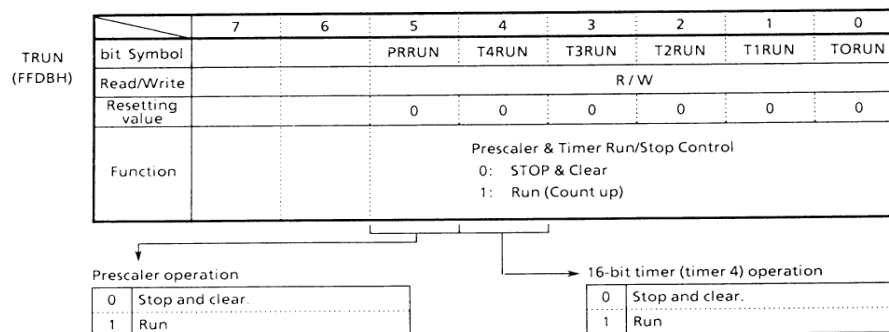| 0	Stop and clear.
1	Run

Figure 3.6 (15). Timer Operation Control Register

① Up-counter (UC16)

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

Either the internal clock øT1 or øT16 from the 9-bit prescaler (also used as 8-bit timer), or the external clock from the TI4 pin (also used as P24/INT1) can be selected as the input clock. Resetting initializes <T4CLK1, 0> to "0,0" which selects the external clock from the TI4 pin.

The timer control register TRUN <T4RUN> controls run, stop and clear the counter UC16.
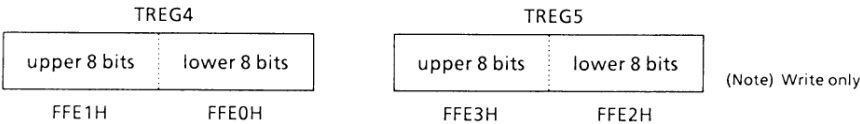
The up-counter UC16 is cleared to "0" when by matching with TREG5. T4MOD <CLE> is used to set clear enable/disable.

When clear is disabled, UC16 operates as a free-running counter.

② Timer registers (TREG4 and TREG5)

These two 16-bit registers are used to set the counter values. The comparator match signal becomes active when there is a match between the value of the Timer register and the UC16 value.
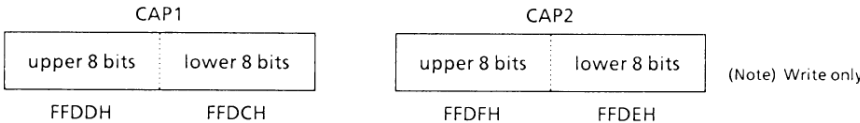
The 16-bit load instuction is used to load data to the timer registers (TREG4, TREG5). This can also be done by executing the 8-bit load instruction two times: once for the lower eight bits and then once more for the upper eight bits.

TREG4	
upper 8 bits	lower 8 bits
FFE1H	FFE0H

TREG5		
upper 8 bits	lower 8 bits	(Note) Write only
FFE3H	FFE2H	

③ Capture registers (CAP1, CAP2)

These two 16-bit registers are used to capture the up-counter UC16 values. The 16-bit load instuction is used to read the capture registers. This can also be done by executing the 8-bit load instruction two times: once for the lower eight bits and then once more for the upper eight bits.

CAP1	
upper 8 bits	lower 8 bits
FFDDH	FFDCH

CAP2		
upper 8 bits	lower 8 bits	(Note) Write only
FFDFH	FFDEH	

④ Capture input control circuit

This circuit controls the timing to latching the up-counter UC16 value in the capture register (CAP1, CAP2). The capture register latch timing is set by T4MOD <CAPM1, 0>.

• When T4MOD <CAPM1, 0> = 0, 0

The capture function is disabled. Disable is the default on reset.

• When T4MOD <CAPM1, 0> = 0, 1

Captures the up-counter values to CAP1 at the TI4 pin (also used as P24/INT1) rising edge; captures to CAP2 at the TI5 pin (also used as P25/INT2) rising edge (time differential measurement).

• When T4MOD <CAPM1, 0> = 1, 0

Captures the up-counter value to CAP1 at the TI4 pin rising edge and capture to CAP2 at the TI4 pin falling edge.
INT1 interrupt is generated at the falling edge only with this setting . (Pulse width measurement)

• When T4MOD <CAPM1, 0> = 1, 1

Captures the up-counter value to CAP1 at the timer flipflop rising edge and captures CAP2 at the timer flip-flop falling edge.
Up-counter values can also be captures to the capture register by software. The current up-counter is captures to CAP1 each time "0" is written to T4MOD <CAP1IN>. (The prescaler should be set in the RUN mode (TRUN <PRUN> = "1").

⑤ Comparators (CP4, CP5)

These are 16-bit comparators that compare and detect matches of the up-counter UC16 with the timer registers TREG4 and TREG5. When a match is

detected, the interrupt INTT4 and INTT5 is generated, respectively. The up-counter is cleared to "0" only when it matches TREG5. (Clearing can be disabled with T4MOD <CLE> = 0).

⑥ Timer flipflop (TFF4)

This flipflop is inverted by the match detect signal from the comparators (CP4 and CP5) and the latch signal to the capture registers (CAP1 and CAP2).
T4FFCR <CAP2TE, CAP1TE, EQ5TE, EQ4TE> are used to enable/disable the inversion for each element.
 TFF4 is cleared to "0" by writing "0, 0" to T4FFCR <TFFC1, 0>, set to "1' by writing "0, 1" and inverted by writing "1, 0".
TFF4 values can be output to the timer output pin TO4. TO4 is also used as P23. P2CR and P2FR are used to make this selection. Set P2CR <P23C> = 1, P2FR <TO4E> = 1 to use as T04.

(1)   16-bit Timer Mode

The following sets the interval time to the timer register TREG5 and generates the INTT5 interrupt.

```
TRUN   ←  - - - 0 - - - -        Stops Timer 4.
INTEL  ←  - - - - 0 - 1 -        Enables INTT5 and disables INTT4 with interrupt
                                 enable register INTEL <IET5,4>.
T4FFCR←  X X 0 0 0 0 1 1         Disables the trigger.
T4MOD  ←  X X 1 0 0 1 * *        Selects the internal clock as the input clock and
            (**=01,10,11)        disables the capture function.
TREG5  ←  **** **** **** ****    Sets the interval time (16 bits)
TRUN   ←  - - 1 1 - - - -        Starts Timer 4.
```

(Note) X ; don't care   - ; no change

(2) 16-bit Event Counter Mode

This timer can be used as an event counter by selecting the external clock (TI4 input pin) as the input clock in the above timer mode (1). To read the counter value, first perform a "software capture" and then read the

capture value.
The counter counts up the rising edge of the TI4 input pin. A minimum of two bus cycles can be counted. (For details, refer to item 4.7 or "4. Electrical Characteristics".)
The TI4 pin is also used as P24/INT1.

```
TRUN    ← - - - 0 - - - -        Stops Timer 4.
P2CR      - - - 0 - - - -         Sets P24 (I/O port) to the input mode.
P2FR    ← - - - * - - - -         When the * = 0 ; TI4 is a square wave.
                                  When the * = 1 ; TI4 is a sign wave.  (zero-cross)
INTEL   ← - - - - 0 0 1 -         Enables INTT5, and disables INTT4 and INT1.

T4FFCR← X X 0 0 0 0 1 1          Disables the trigger.
T4MOD   ← X X 1 0 0 1 0 0         Sets the input clock to TI4.
TREG5   ← **** **** **** ****     Sets the count number (16 bits).
TRUN    ← - - 1 1 - - - -         Starts Timer 4.
```
(Note)   Set the prescaler to "RUN" even when used as an event counter.

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

The timer flipflop TFF4 is inverted by a match between the up-counter UC16 and the timer registers TREG4 and TREG5. The programmable pulse generation output

mode is set by using TFF4 for output to timer output pin TO4 (also used as the P23 pin). However, it is necessary that the following conditions should be satisfied.

(value set to TREG4) < (value set to TREG5)

```
TRUN    ← - - - 0 - - - -        Stops Timer 4.
TREG4   ← **** **** **** ****     Sets the duty.
TREG5   ← **** **** **** ****     Sets the cycle.
T4FFCR← X X 0 0 1 1 0 0          Sets for inversion on match between TFF4 and
                                  TREG4/TREG5.  Also initializes TFF4 to "0".
T4MOD   ← X X 1 0 0 1 * *         Sets the internal clock as the input clock and disables
            (**=01,10,11)         the capture function.
P2CR    ← - - - - 1 - - -     }   Assigns P23 as TO4.
P2FR    ← - - - - 1 - - -
TRUN    ← - - 1 1 - - - -         Starts Timer 4.
```
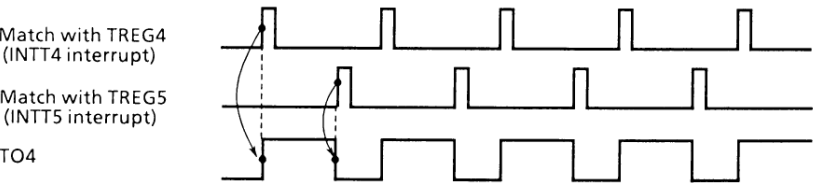(Note) X ; don't care   - ; no change

Figure 3.6 (17). Programmable Pulse Output

(4)   Examples using the capture function

The capturing of the up-counter UC16 value to the capture registers CAP1 and CAP2, the timer flip-flop TFF4 inversion due to the match signal from CP4 and CP5, and output TFF4 to the TO4 pin can be enabled or disabled. By combining with the interrupt function, many applications are possible, including the following examples:

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

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

The up-counter UC16 is set for free-running with the internal clock, The external trigger pulse is input from the TI4 pin, and the up-counter (UC16) value is captured to the capture register CAP1 at the rising edge of TI4 pin. (Set T4MOD <CAPM1, 0> = 0, 1).

In the example below, the value (c + d) obtained by adding the delay time (d) to the capture register CAP1 value (c) is loaded to TREG4 when the interrupt INT1 is generated at the rising edge of TI4 pin; and the value (c + d + p) obtained by adding the one-shot pulse width (p) to this TREG1 value is loaded to TREG5.

The interrupt INT1 sets the T4FFCR register to enable the timer flipflop TFF4 inversion only on a TREG4, or TREG5 match. The interrupt INT5 returns this to disable.
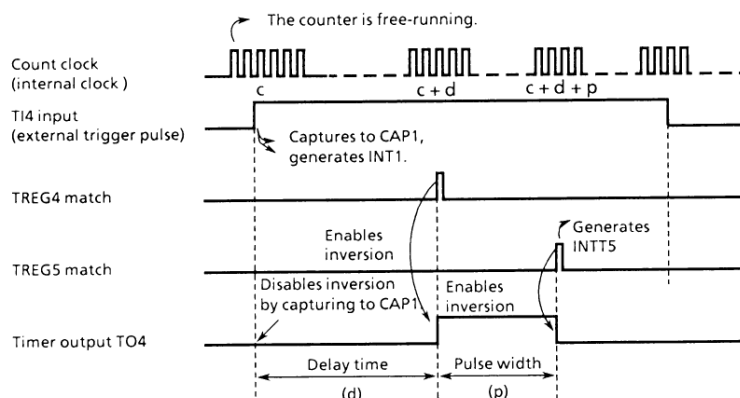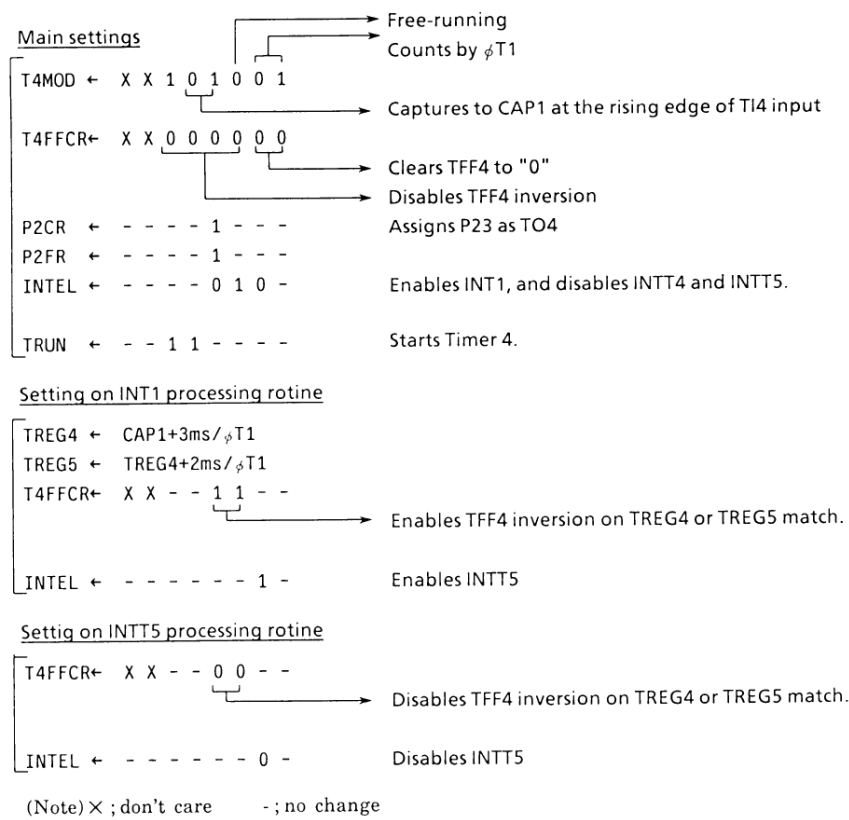

Figure 3.6 (18). One-Shot Pulse Output (with Delay)

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

```
Main settings                                          Free-running
                              ┌──────────────────────► Counts by φT1
                              │ ┌─►
T4MOD ←  X X 1 0 1 0 0 1
                │ │ │                                  Captures to CAP1 at the rising edge of TI4 input
                └─┘ └──────────────────────────►
T4FFCR← X X 0 0 0 0 0 0
              │ │ │ │   └─┘                            Clears TFF4 to "0"
              │ │ │ │     └───────────────────────►   Disables TFF4 inversion
              └─┴─┴─┴──────────────────────────►
P2CR  ←  - - - - 1 - - -                               Assigns P23 as TO4
P2FR  ←  - - - - 1 - - -
INTEL ←  - - - - 0 1 0 -                               Enables INT1, and disables INTT4 and INTT5.

TRUN  ←  - - 1 1 - - - -                               Starts Timer 4.

Setting on INT1 processing rotine

TREG4 ←  CAP1+3ms/φT1
TREG5 ←  TREG4+2ms/φT1
T4FFCR← X X - - 1 1 - -
                  └─┘
                    └──────────────────────────────►  Enables TFF4 inversion on TREG4 or TREG5 match.

INTEL ←  - - - - - - 1 -                               Enables INTT5

Settig on INTT5 processing rotine

T4FFCR← X X - - 0 0 - -
                  └─┘
                    └──────────────────────────────►  Disables TFF4 inversion on TREG4 or TREG5 match.

INTEL ←  - - - - - - 0 -                               Disables INTT5
```

(Note) ✕ ; don't care   - ; no change

When delay time is not required, the timer flipflop TFF4 is inverted by capturing to CAP1, and setting the value (c+p) obtained by adding the one-shot pulse width (p) to CAP1 to the timer register TREG5 with the interrupt INT1. The TFF4 inversion is enabled on a match between the up-counter UC16 and TREG5, and disabled by the interrupt INT5.
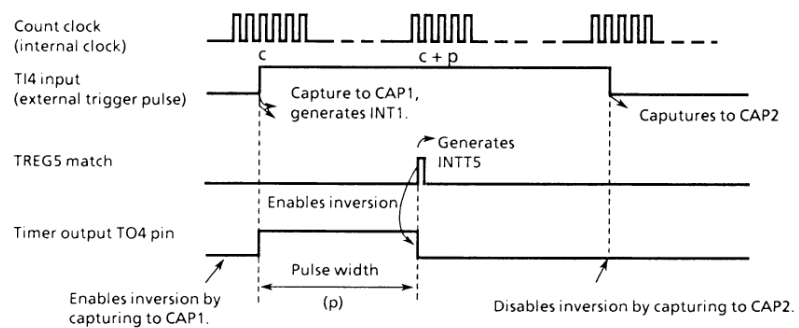

Figure 3.6 (19). One-shot Pulse Output (without Delay)

② Frequency measurement

This mode is used to measure the external clock frequency. The external clock is input to the TI4 pin, and the 8-bit timers (Timer 0 and Timer 1) and 16-bit timer/event measure (Timer 4) are used to measure the frequency. The Timer 4 input clock is input to the TI4 input and the up-counter UC16 value is captured to CAP1 at the rising edge; the up-counter UC16 value is captured to CAP2 at the falling edge of the timer flip-flop TFF4 of the 8-bit Timers 0 and 1.

The frequency is determined by the interrupt INTT0 and INTT1 from the difference between the capture register CAP1 and CAP2 values.
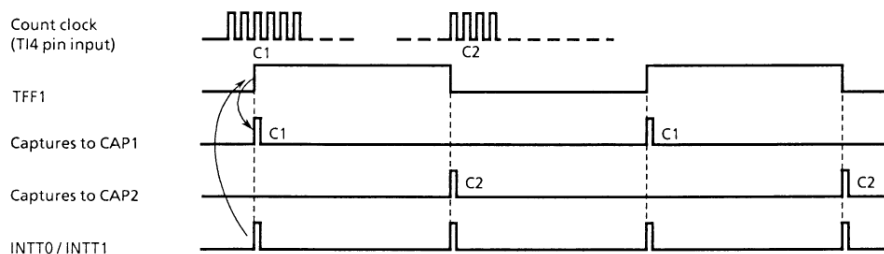

Figure 3.6 (20). Frequency Measurement

If the "H" level width of the TFF1 (8-bit timer) set to 0.5 sec and the difference between CAP1 and CAP2 is 100, the frequency will be $100 \div 0.5$ [sec] = 200 [Hz].

③ Pulse width measurement

This mode is used to measure the "H" level width of the external pulse.

The external pulse is input to the TI4 pin, while the 16-bit timer is set to free-running with the internal clock. A trigger is applied by the capture function at either the rising or falling edge of the external pulse, and the up-counter UC16 value is captured to CAP1 and CAP2. The interrupt INT1 is generated at the falling edge of the TI4 input.

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

If the difference between CAP1 and CAP2 values is 100, with an internal clock cycle of $0.8\mu s$, the pulse width will be $100 \times 0.8\mu s = 80 \ \mu s$.
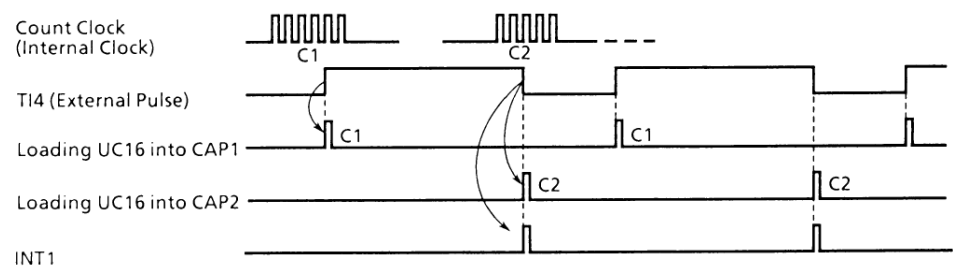
Figure 3.6 (21). Pulse Width Measurement

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

When the external pulse "L" level width is measured, it can be determined from the difference between the first C2 and the second C1 at the second interrupt INT1.

④ Time difference measurement

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

The 16-bit timer (Timer 4) is set to free-running with the internal clock, and the up-counter UC16 value is captured to CAP1 upon a detection of the rising edge of the counter UC16 value is captured to CAP1 upon detection of the rising edge of the input pulse to TI4 pin. The interrupt INT1 is generated.

In the same way, the up-counter UC16 value is captured to CAP2 upon detection of the rising edge of the input pulse to the TI5 pin, and the interrupt INT2 is generated. The timer difference can be determined when values have been captured to both CAP1 and CAP2.
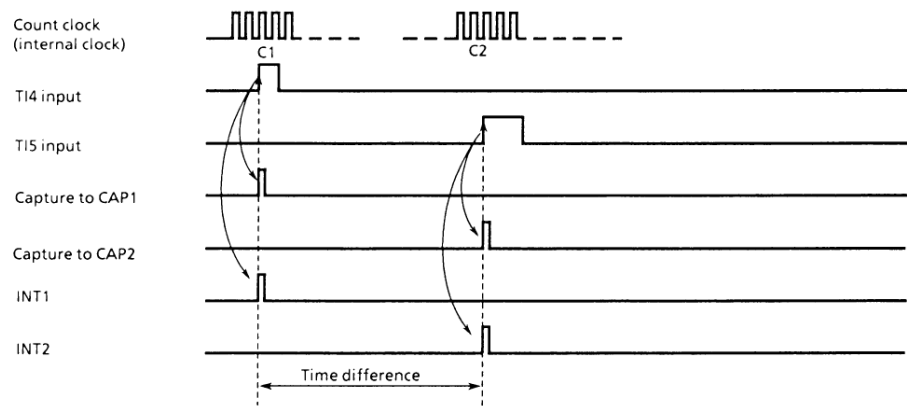

Figure 3.6 (22). Time Difference Measurement

3.7 Watchdog Timers (Runaway Detecting Timer)

The watchdog timer (WDT) detects the malfunction (runaway) of the CPU due to noises, etc., and returns it to the normal operation. When the WDT has detected malfunction, a non-maskable interrupt is generated and the CPU is notified.

3.7.1 Configuration

The watchdog timer (WDT) block diagram is shown in Figure 3.7 (1).

The watchdog timer comprises of a 20-stage binary counter used as the ø (fc/2) input clock, a flipflop that enable/disable a selector, the selector that selects one of the four binary counter outputs, and two control registers.

The watchdog timer generates the interrupt INTWD after the detection time is set with the watchdog timer mode register WDMOD <WDTP1, 0>, and should be cleared the watchdog timer binary counter by software (instruction) before the INTWD

interrupt is generated. If the CPU malfunctions (runaway) due to some cause such as noises, the binary counter will overflow and the INTWD interrupt will be generated if the watchdog timer clear instruction is not executed. The CPU is notified of malfunction (runaway) by the INTWD interrupt and runs the corrective program for malfunction (runaway) to return to the normal operation.

The watchdog timer starts an operation immediately after a reset is released.

The watchdog timer stops its operation only in the STOP mode. After the STOP mode is released mode is released and the warming-up time has elapsed, the watchdog timer resumes an operation.

The watchdog timer operates in the other standby mode (IDLE1, 2 and RUN modes), but can be disabled when entering one of these standby modes.
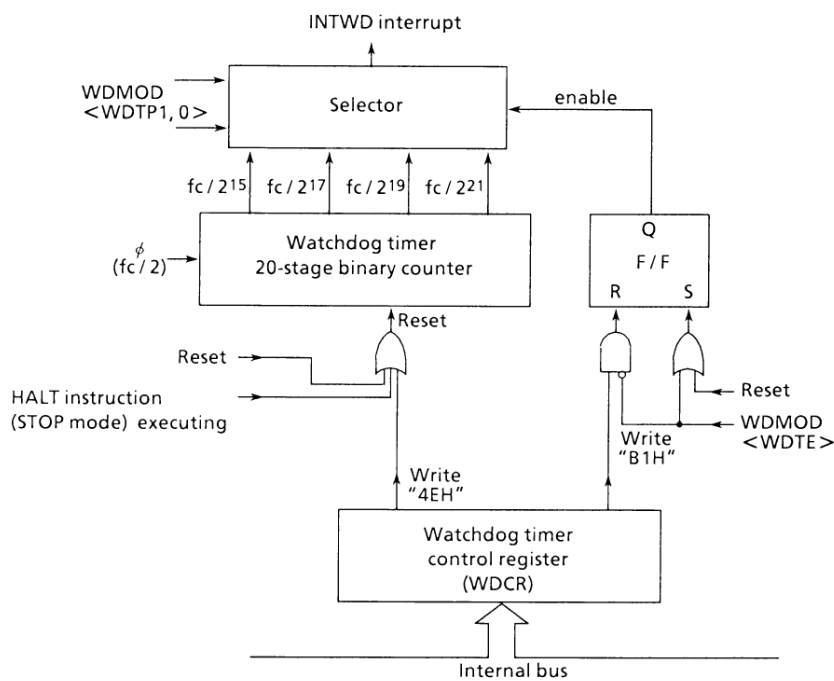

Figure 3.7 (1). Watchdog Timer Block Diagram

3.7.2 Control Registers

The watchdog timer (WDT) is controlled by two control registers (WDMOD and WDCR). The WDT related registers are shown in Figure 3.7 (2).

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	Detection 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 the EXX instruction is executed.	1 : Pin is driven even in the STOP mode.

Explained in 3.4.4 STOP Mode.

Invert each time the EXX instruction is executed.

Standby mode selection with the HALT instruction (HALT mode)

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

Warming-up time selection when returning from standby mode.

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

(Note)   The time is calculated in case of @fc = 10MHz.

Watchdog timer detection time selection (Watch Dog Timer Period)

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)   The time is calculated in case of @fc = 10MHz.

Watchdog timer enable/disable control

0	Disable (Note)
1	Enable

(Note)   To disable, it is also necessary to write the disable code to the WDCR register. Disable 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							
	After reset	—							
	Function	B1H : WDT Disable code				4EH : WDT Clear code			

Watchdog timer disable and clear

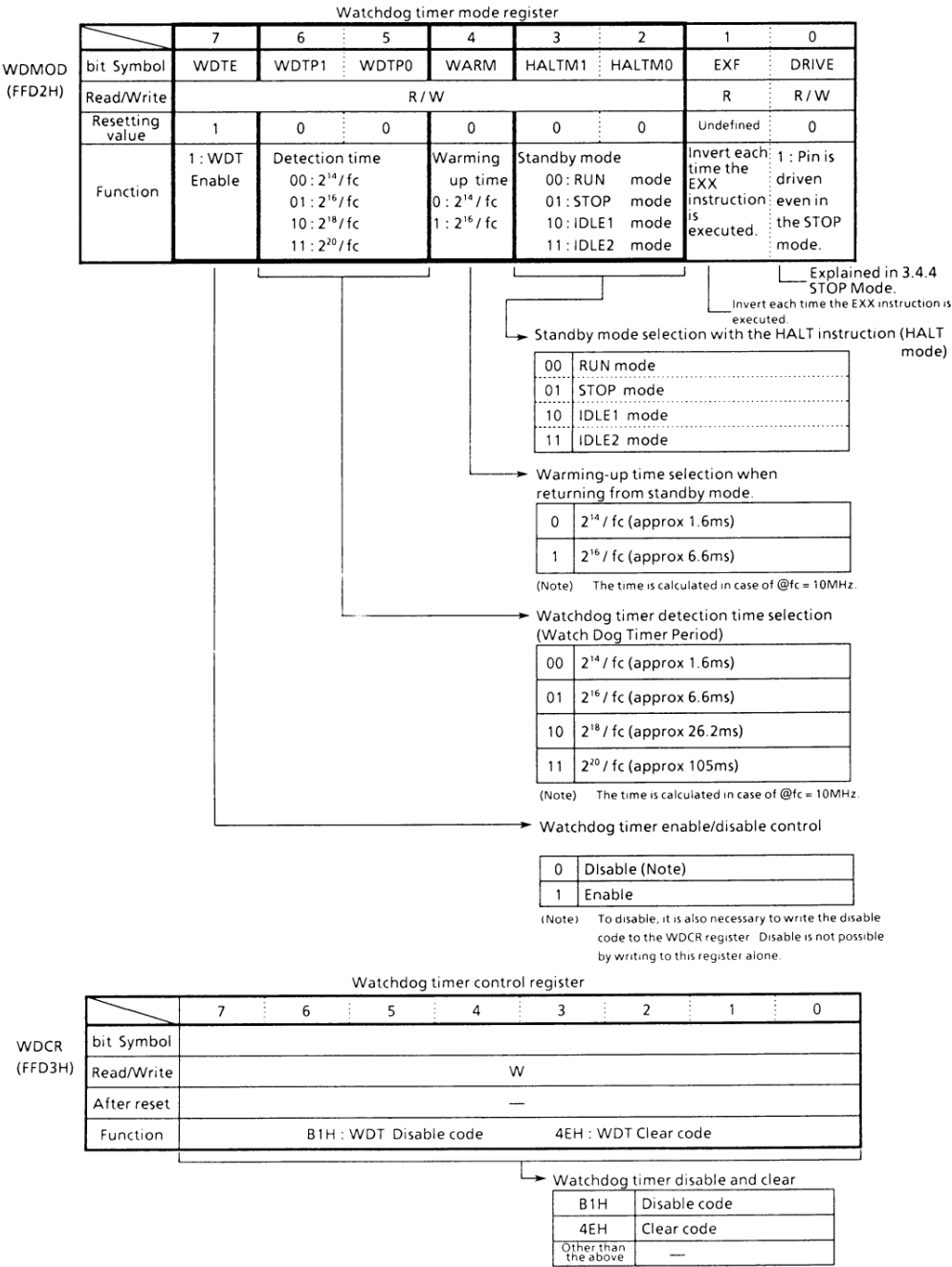| B1H	Disable code
4EH	Clear code
Other than the above	—

Figure 3.7 (2). Watchdog Timer Related Registers

3.7.3 Operation

(1) Watchdog timer mode register (WDMOD)

① Watchdog timer detection time select register:
WDMOD <WDTP1, 0>

This 2-bit register is used to set the watchdog timer interrupt period for detecting a malfunction (runway). This register is initialized to <WTRP1, 0> = 0, 0 by resetting therefore, 2^{14}/fc[sec] is set. (The number of states is approximately 8,192.)
The INTWD interrupt vector address is 0020H.

② Watchdog timer enable/disable control register:
WDMOD <WDTE>

<WDTE> is initialized to "1" by resetting, which enables the watchdog timer function.
To disable, it is necessary to clear this bit to "0" and write the disable code (B1H) to WDCR. This makes it difficult for the watchdog timer to be disabled, even if the malfunction occurs.

It is possible to return to the enable state by setting <WDTE> to "1".

(2) Watchdog Timer Control Register (WDCR)

This register is used to disable and clear the watchdog timer.

① Watchdog timer disable control

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

② Watchdog timer clear control

To clear the watchdog timer write 4EH to WDCR.
The clear signal input and resets the watchdog timer during reset operations or after the STOP mode is set.

Setting examples : ① Clear the watchdog timer
    WDCR ← 0 1 0 0 1 1 1 0   Writes the clear code 4EH.

② Set the watchdog timer detection time to 2^{16}/fc
    WDMOD ← 1 0 1 - - - X X

③ Disable the watchdog timer
    WDMOD ← 0 - - - - - X X   Clears <WDTE> to "0".
    WDCR ← 1 0 1 1 0 0 0 1   Writes the disable code B1H.

④ Set the IDLE2 mode
    WDMOD ← 0 - - - 1 1 X X   Disables the watchdog timer and sets
    WDCR ← 1 0 1 1 0 0 0 1   the IDLE2 mode.
    Executes the HALT instruction. Sets the standby mode.

⑤ Set the STOP mode (warming-up time : 2^{16}/fc)
    WDMOD ← - - - 1 0 1 X X   Sets the STOP mode.
    Executes the HALT instruction. Sets the standby mode.

3.8 8-bit Half-Flash A/D Converter

The TMP90C846 has an 8-bit high-speed, half-flash A/D converter with 2-channel analog input. The features are as follows.

- 8-bit half-flash A/D converter with 2-channels analog input pins.
- Minimum sampling rate 2 states (400nsec @fc = 10MHz)

- 16-bytes built-in FIFO (First In First Out) RAM for the storage of conversion results.
- Software start (register write) trigger with single or repeat conversion mode and external start trigger.
- A/D conversion interrupt function (an interrupt is generated by an input to FIFO RAM).

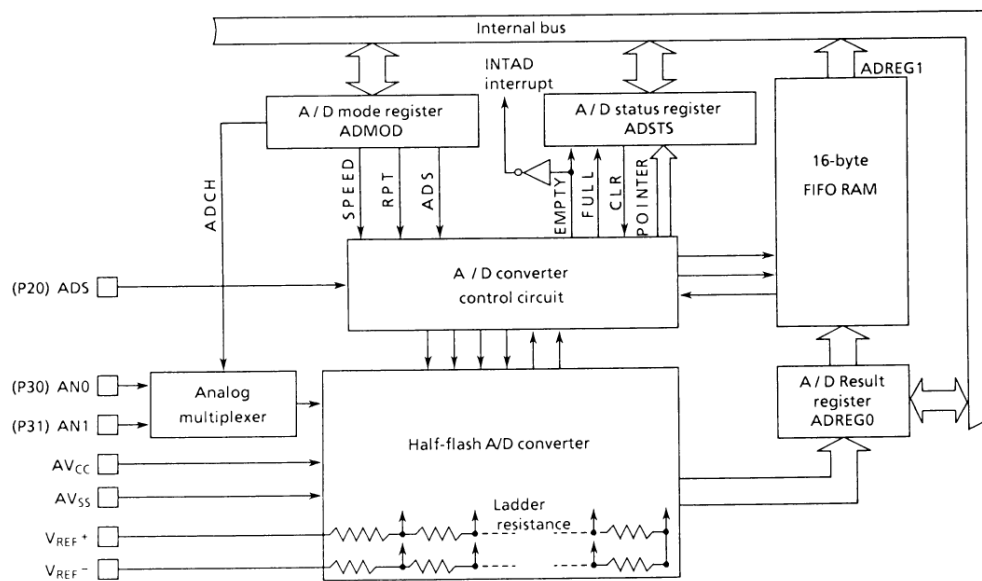The A/D converter block diagram is shown in Figure 3.8 (1).

Figure 3.8 (1). A/D Converter Block Diagram

3.8.1 Basic Operation of the Half-Flash A/D Converter

This is a two-time conversion type A/D converter that converts the upper four bits and lower four bits separately. The function outline is shown in Figure 3.8 (2).
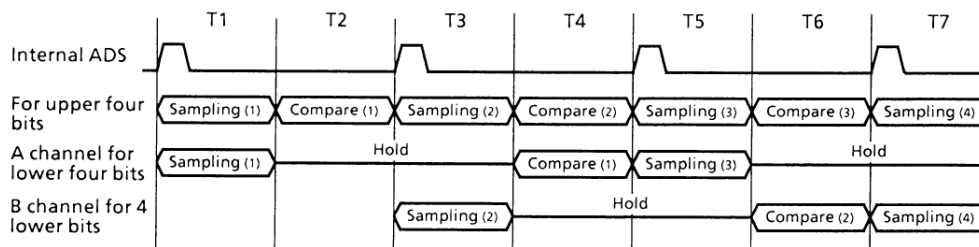

Figure 3.8 (2). Half-Flash A/D Converter Outline

When the A/D converter start signal (ADS) is input, the analog input voltage in T1 is sampled by the A/D converter for the upper four bits and the A channel for the lower four bits. The A/D converter for the upper four bits compares the output voltage from the internal ladder resistance with input voltage in T2, and outputs the conversion results of the upper four bits. The A channel A/D converter for the lower four bits compares in T4 in the same way that held the voltage in T1. The 8-bit conversion results can be obtained in T5.

The A/D converter for the lower four bits has two channels (A and B). The next analog input voltage in T3 is sampled by the A/D converter for the upper four bits and the B channel A/D converter for the lower four bits.

This type of processing enables the high-speed A/D conversion with a minimum sampling rate of 2 states (400ns @ 10MHz). The sampling rate of the low-speed conversion mode is 4 states (800ns @ 10MHz).

3.8.2 Operation

(1)   A/D converter start operation

The conversion by the A/D converter can be started either by inputting "1" to the ADS pin (also used as P20) or by writing "1" to the internal ADS register ADMOD <ADS> with the software.

① External start operation

The external start function can be enabled by writing "0" to the Port 20 control register P2CR <P20C> and writing "1" to the internal ADS register ADMOS <ADS>.
The external start signal performs a conversion only once. The value of rpeat mode register ADMOD <RPT> is ignored.

Note: The external start signal ADS is sensed with level in the 1/2 state zone from rising/falling edge of the CLK signal (ø1 zone in Figure 3.8 (3)).
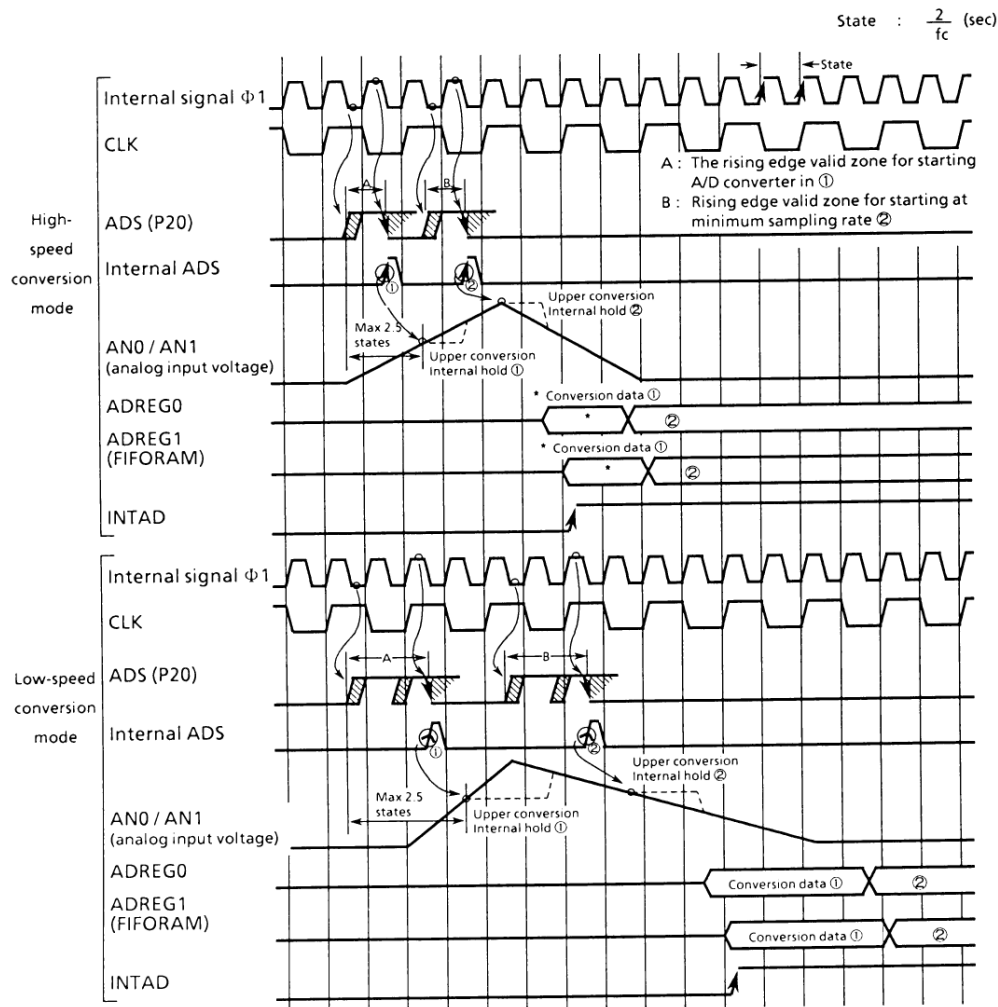
Figure 3.8 (3). External Start Conversion Timing

(Note) • Refer to "4.4 A/D Converter Electrical Characteristics" concerning the ADS (P20) AC specifications.

② Software start operation

When using the software start operation, the A/D converter starts by writing "1" to ADMOS <ADS>. <ADS> is always read as "0".

(2) A/D converter repeat specification

In the repeat mode, the A/D converter starts automatically after completion of each conversion.
The repeat mode can only be used with the software start operation. The A/D conversion in repeat mode is started by writing "1" to both <ADS> and <RPT>.

To end the repeat mode operation, write "0" to <RPT>. The repeat mode will end when the current conversion is completed.

Read the A/D conversion result storage register ADREG0 in the repeat mode since it contains the newest conversion data.

The repeat mode operation timing is shown in Figure 3.8 (4).

<RPT> is cleared to "0" by resetting; therefore, the A/D converter becomes the one-time conversion mode.
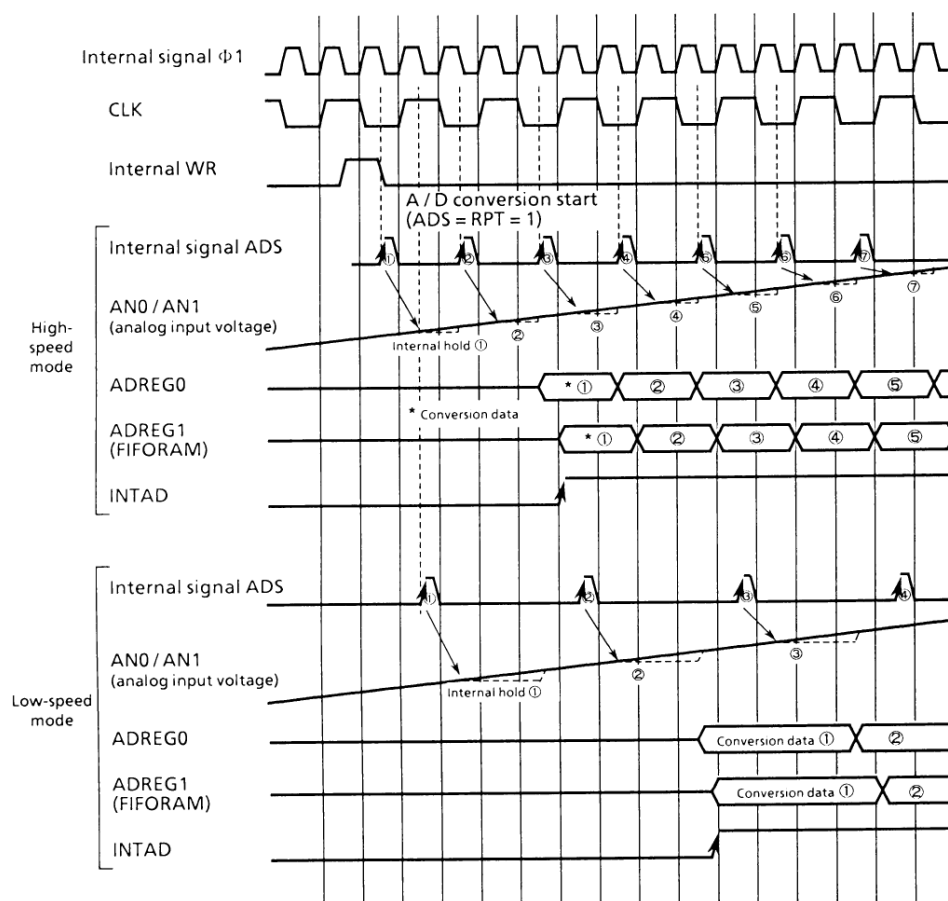
Figure 3.8 (4) Repeat Mode Operation Timing

(3)   A/D converter speed setting

The A/D converter has two speed modes, the high-speed conversion mode and the low-speed conversion mode, which can be selected with the speed specification register ADMOD <SPEED>.

The sampling rates are 2 states (400ns @ 10MHz) in the high-speed conversion mode and 4 states (800ns @ 10MHz) in the low-speed conversion mode. The low-speed conversion mode is selected by clearing <SPEED> to "0" with a reset operation.

To use the high-speed conversion mode, set <SPEED> to "1".

(4)   Analog input channel

Before starting the A/D conversion, select one of the two analog input channels (AN0, AN1) with ADMOD <ADCH>.

AN0 (P30) is set as the analog input pin by clearing <ADCH> to "0" by reset. To use AN1 (P31), write "1" to <ADCH>.

The pin which is not used as an analog input pin can be used as an ordinary input port.

A/D converter status register

ADSTS (FFCBH)		7	6	5	4	3	2	1	0
	bit Symbol	EMPTY	FULL		CLR			Pointer	
	Read/Write	R			W		R		
	Resetting value	1	0		0		0		
Read modify write is not possible.	Function	FIFO Empty Flag 1: Empty 0: Not Empty	FIFO Full Flag 1: Full 0: Not Full		1: FIFO Clear (Pointer & Flag) 0: Don't Care	FIFO Pointer Indicates the number of data stored to FIFO (number of readable data)				

→ FIFO RAM pointer (4-bit)
(indicates the number of data stored to FIFO RAM) ("0000" after reset)

→ FIFO RAM clear control

0	Don't care
1	FIFORAM clear (set "EMPTY")

→ FIFO RAM Full flag

0	Not Full
1	Full

→ FIFO RAM Empty flag

0	Not Empty
1	Empty

A/D converter mode register

ADMOD (FFCCH)		7	6	5	4	3	2	1	0
	bit Symbol				ADCH		SPEED	RPT	ADS
	Read/Write				R / W		R / W		
	Resetting value				0		0		
	Function				0 : AN0 1 : AN1		A/D conversion speed 0: low-speed conversion 1: high-speed conversion	A/D conversion repeat mode 0 : one-time conversion 1 : REPEAT	1 : A/D conversion start 0 : Don't Care

→ A / D conversion start

0	———
1	A / D conversion start

(Note) Always read as "0".

→ A / D conversion repeat mode

0	one-time conversion mode
1	repeat conversion mode

→ A / D conversion speed

0	low-speed mode (4 states)
1	high-speed mode (2 states)

→ Analog input channel

0	AN0
1	AN1

Figure 3.8 (5). A/D Converter Related Registers (1/2)

A/D converter conversion value storage register 0

	7	6	5	4	3	2	1	0
ADREG0 bit Symbol				—				
(FFCDH) Read/Write				R / W				
Resetting value				Undefined				

(Note) Do not write during the A/D conversion operation. The written data are stored to FIFO RAM.

A/D converter conversion value storage register 1

	7	6	5	4	3	2	1	0
ADREG1 bit Symbol				—				
(FFCEH / FFCFH) Read/Write				R				
Resetting value				Undefined				

(Note) Address used to read FIFO RAM. FFCEH and FFCFH are allocated as addresses for reading conversion values with a 16-bit load instruction. Use the address FFCEH when reading conversion values with an a 8-bit load instruction.
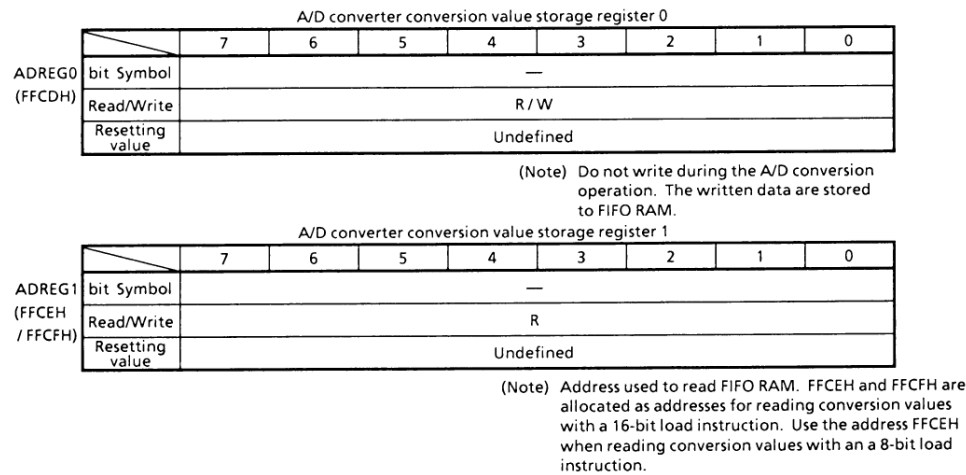
Figure 3.8 (5). A/D Converter Related Registers (2/2)

(5)　FIFO RAM

The A/D converter conversion result register ADREG0 (a 1-byte latch) and the 16-byte FIFO RAM (ADREG1 (FFCEH/FFCFH) is assigned for reading conversion results) are used to store the A/D converter conversion results.

The A/D conversion results are stored to ADREG0 after the completion of each conversion operation and are loaded to FIFO RAM until the FIFO RAM Full flag ADSTS <FULL> is set to "1".

① FIFO RAM clear control

The pointers ADSTS <POINTER> and <FULL> are cleared to "0" by writing "1" to FIFO RAM clear function register ADSTS <CLR>. Thus, the Empty flag ADSTS <EMPTY> is set to "1", and FIFO RAM is cleared.

② FULL flag

When the 16 byte FIFO RAM becomes full, <FULL> is set to "1". Then the loading of A/D converter conversion data to the FIFO RAM is stopped, and <POINTER> is cleared to "0".

ADREG0 can be updated even after <FULL> is set to "1". Thus, in the repeat mode, disable the interrupt INTAD and read the new conversion values from ADREG0.

③ Empty flag

This flag indicates that FIFO RAM is "empty". When <EMPTY> is set to "1" by resetting, it indicates that FIFO RAM is empty, however, when data is loaded to FIFO RAM due to the completion of the A/D conversion, <EMPTY> is cleared to "0".

The A/D conversion interrupt request INTAD is the <EMPTY> inversion signal. When <EMPTY> is cleared to "0" (data is input to FIFO RAM), INTAD is set to "1" and requests an interrupt.

Only the data FFH will be read when FIFO RAM (ADREG1) is read while <EMPTY> is set to "1". Make sure that <EMPTY> has been cleared to "0" before reading FIFO RAM data.

④ Pointer (4-bit)

This is the register ADSTS <POINTER> which indicates the remained number of data stored to FIFO RAM. The number of remaining data can be determined by reading this pointer.

<POINTER> indicates "0" when <EMPTY> or <FULL> flag is set to "1", that is, when FIFO RAM is either empty or full. <POINTER> remains to "0" as long as <EMPTY> or <FULL> is not cleared to "0". In other words, the pointer value remains at "0" when reading in the EMPTY status or writing in the FULL status.

⑤ Reading FIFO RAM data

FIFO RAM data can be obtained by reading ADREG1. The two addresses, FFCEH and FFCFH, are allocated to ADREG1 to enable the 16-bit load instruction either 8-bit or 16-bit load instructions can be used with the address FFCEH.

INTAD is set to "1" as long as FIFO RAM contains data and is held requesting. INTAD can be cleared to "0" by either reading all FIFO RAM data or writing "1" to the <CLR> register.

(6)　A/D conversion interrupt request sources (INTAD)

When data is loaded to FIFO RAM, INTAD is set to "1" because it is the inversion of the <EMPTY> signal, and an interrupt request is output to the internal interrupt controller.

This request is for a level interrupt, which is not reset by reading the interrupt vector V; therefore, it is necessary to use the interrupt routine to empty the FIFO RAM. Also,

this interrupt is not reset by writing the interrupt vector 40H/8 to the interrupt request register IRFH. For resetting INTAD, it is necessary to either write "1" to <CLR> or read all FIFO RAM data.

3.8.3 Analog Reference Voltage

The VREF$^+$ pin is the High A/D converter analog reference voltage input pin and the VREF$^-$ pin is the Low A/D converter analog reference voltage input pin. The A Vcc and A Vss pins are used as the A/D converter power supply.

The VREF$^+$ and VREF$^-$ pins are variable ($3.5 \leq$ VREF$^+ \leq$ Vcc, Vss $\leq$ VREF$^- \leq 2.5$); however, when the VREF$^+$ voltage is below 5V, the conversion error for the LBS tends to increase. Refer to "4.4 A/D Converter Electrical Characteristics" for the specifications.

3.8.4 Program Example

① To A/D convert the analog input voltage of the AN0 pin with the external A/D conversion start ADS in the high-speed conversion mode, and process the data with the A/D interrupt INTAD routine:

```
              bit  7 6 5 4 3 2 1 0
           ┌ P2CR  ←   – – – – – – – 0      Sets P20 to the input mode.
           │ P2FR  ←   – – – – – – – 1      Enables ADS.
  Main     │ INTEL ←   – 1 – – – – – –      Enables INTAD.
  settings │ ADSTS ←   X X X 1 X X X X      Clears FIFO RAM.
           │ ADMOD ←   X X X 0 X 1 – 1      Specifies the high-speed mode and channel 0.
           └    EI                          Enables maskable interrupts.


           ┌    A  ←   ADREG0               Reads the ADREG0 result into the accumulator.
  A/D      │ ADSTS ←   X X X 1 X X X X      Clears FIFO RAM.
  interrupt│
  routine  │
           └

                          (Note)  X : Don't care
                                  – : No change
```

② To A/D convert the analog input voltage of the AN1 pin with software in the low-speed repeat mode:

```
  ADMOD  ←   X X X 1 X 0 1 1      Starts conversion with channel 1 in the low-speed
                                  repeat mode.
```

③ To A/D convert the analog input voltage of the AN1 pin with the external A/D conversion start ADS in the high-speed conversion mode, and process the data when 16-bytes have been stored to FIFO RAM:

3.9 8-bit Voltage Output Type D/A Converter

The TMP90C846 has a 2-channels, 8-bit voltage output type D/A converter. The features are as follows.

```
P2CR   ←   - - - - - - - 0
P2FR   ←   - - - - - - - 1
ADSTS  ←   X X X 1 X X X X
```

• 2-channel, 8-bit voltage output type D/A converter
• Voltage output range of A Vcc/2 ± A Vcc (1.26 - 3.75 [V], @ A Vcc = 5 [V])
   The D/A converter block diagram is shown in Figure 3.9 (1).
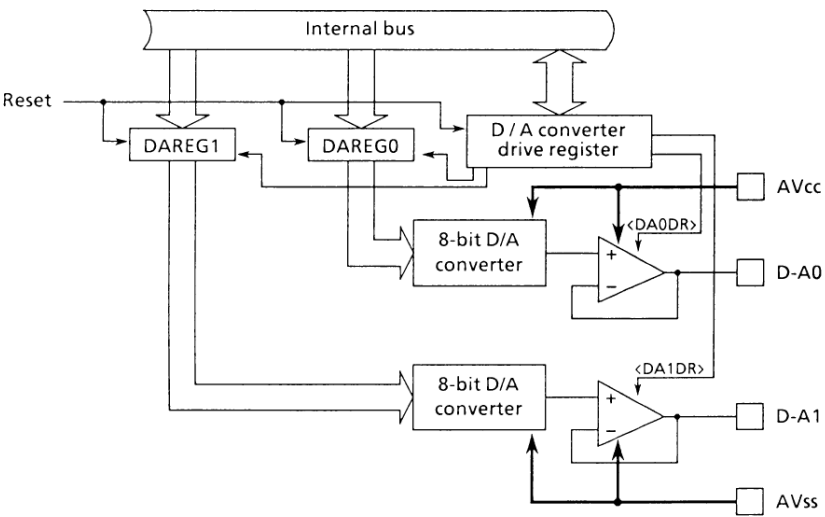
Figure 3.9 (1). D/A Converter Block Diagram

D/A converter drive register

DADRV (FFCAH)		7	6	5	4	3	2	1	0
	bit Symbol							DA1DR	DA0DR
	Read/Write							R / W	
	Resetting value							0	
	Function							0 : 0V output 1 : register value conversion output	

D / A output amplifier drive specification

0	0V output
1	register value conversion output

D/A converter conversion register 0

DAREG0 (FFD0H)		7	6	5	4	3	2	1	0
	bit Symbol	—							
	Read/Write	W							
	Resetting value	0							
	Function	D/A conversion started with register write, output to DA-0.							

D/A converter conversion register 1

DAREG1 (FFD1H)		7	6	5	4	3	2	1	0
	bit Symbol	—							
	Read/Write	W							
	Resetting value	0							
	Function	D/A conversion started with register write, output to DA-1.							

The contents of this register become the D/A converter digital input data. The relationship between register values and output voltage is shown below.

Drive register	Register value	Output voltage (V)	
		AV_{CC}	When AV_{CC} = 5V
1	00H	$AV_{CC} / 4 + (1 \times AV_{CC}) / 512$	1.260
1	01H	$AV_{CC} / 4 + (2 \times AV_{CC}) / 512$	1.270
1	02H	$AV_{CC} / 4 + (3 \times AV_{CC}) / 512$	1.279
⋮	⋮	⋮	⋮
1	N	$AV_{CC} / 4 + ((N + 1) \times AV_{CC}) / 512$	$(129 + N) \times 5 / 512$
⋮	⋮	⋮	⋮
1	FFH	$AV_{CC} / 4 + (256 \times AV_{CC}) / 512 = (3 \times AV_{CC}) / 4$	3.750
0	XX	0V output	←
X	XX	0V output after executing the HALT instruction in the STOP mode.[*1]	←

(Note)

XX : don't care

*1) : After the HALT instruction is released, the output voltage returns to the same value as before the HALT instruction is executed. In the HALT mode other than the STOP mode, an output of the conversion voltage continues in accordance with the register value, regardless of the HALT instruction.
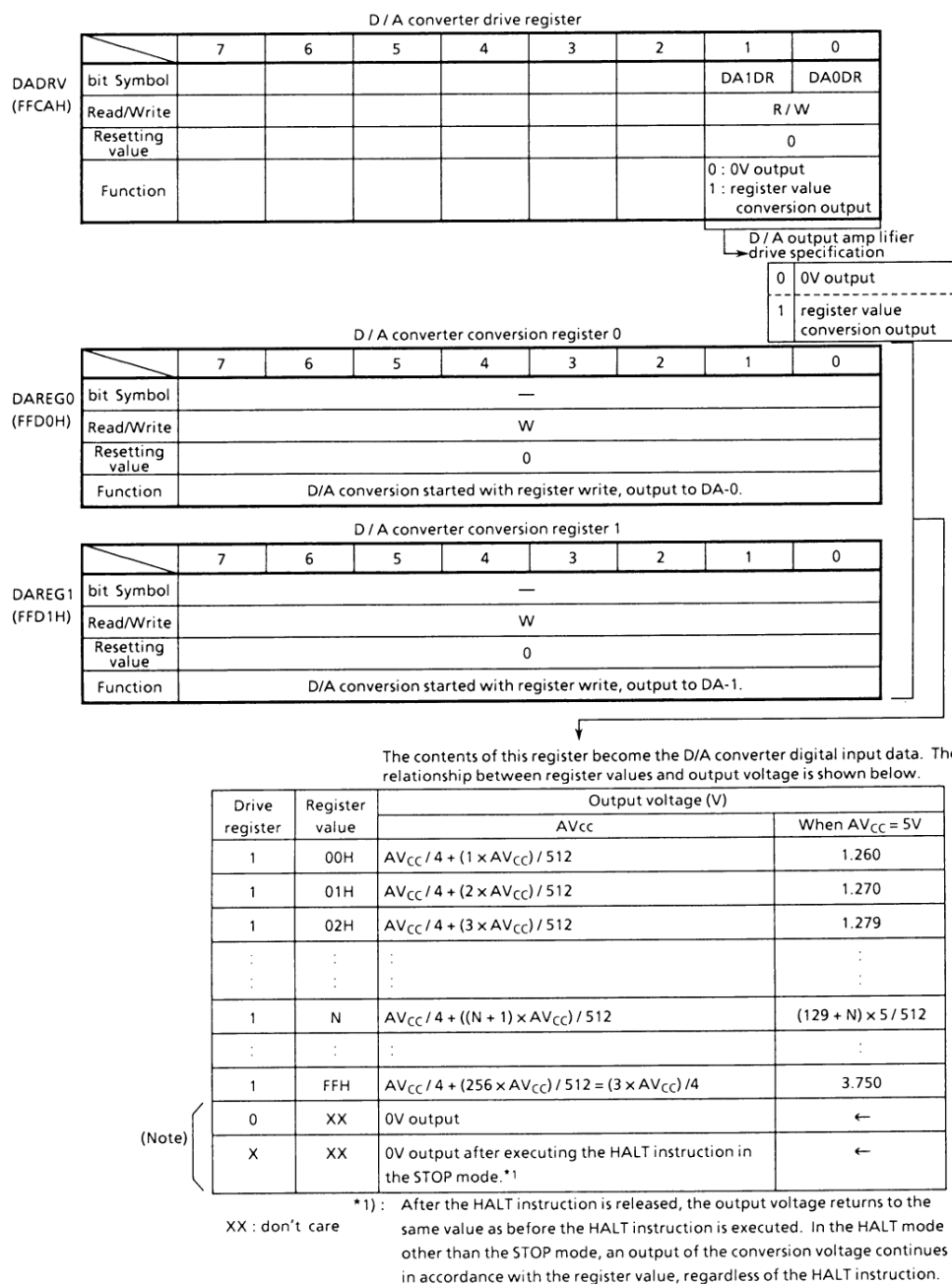
Figure 3.9 (2). D/A Converter Related Registers

3.9.1 Operation

When the value of the D/A converter drive register DADRV <DA1DR, DA0DR> is "1", the built-in D/A converter converts the digital values of the D/A converter conversion registers DAREG1 and DATEG0 to the analog values, and outputs the conversion voltages from the D - A1 and D - A0 pins. The relationship between input data and output voltages is shown in Figure 3.9 (2).

Because <DA1DR> and DA0DR> are cleared to "0" by resetting 0V is output from the D - A1 and D - A0 pins. DAREG1 and DAREG0 are cleared to "00H" by resetting. Thus, if DADRV is set to "1" after a reset, A Vcc/4 (see Figure 3.9 (2)) is output from the relevant pin. To output the relevant analog values using the D/A converter, first write "1" to the DADRV of the channel to be used, and write data to DAREG.

If the HALT instruction is executed after specifying the STOP mode (WDMOD <HALTM1, 0> = 0, 1), 0V is output from the D - A0 and D - A1 pins, regardless of the DADRV and DAREG values.

3.9.2 Example Program

① When outputting the voltage (3 x A Vcc)/4 (3.75V @ A Vcc = 5V) from the D - A0 pin:

```
      bit  7 6 5 4 3 2 1 0
DADRV      X X X X X X 1 1      Sets DA0DRV and DA1DRV to "1".
DAREG0     0 0 1 0 0 0 1 0      Writes "22H" to DAREG0.
DAREG1     0 1 0 0 0 1 1 1      Writes "47H" to DAREG1.
```

② When outputting the voltage ($A3_H$ x A Vcc)/ 512 (1.59V @A Vcc = 5V) from the D - A0 pin and the voltage ($C8_H$ x A Vcc)/512 (1.59V @A Vcc = 5V) from the D - A1 pin, the output value will be (81_H + N_H) X A Vcc/512 when N is written to the register, according to Figure 3.9 (2). In this case,

$$A3_H = 81_H + 22_H$$
$$C8_H = 81_H + 47_H$$

Therefore, 22_H, 47_H are written to DAREG.

```
      bit  7 6 5 4 3 2 1 0
DADRV      X X X X X X 1 1      Sets DA0DRV and DA1DRV to "1".
DAREG0     0 0 1 0 0 0 1 0      Writes "22H" to DAREG0.
DAREG1     0 1 0 0 0 1 1 1      Writes "47H" to DAREG1.
```

4. Electrical Characteristics
TMP90C846F

4.1 Absolute Maximum Ratings

Symbol	Item	Rating	Unit
V_{CC}	Power Supply voltage	$-0.5 \sim +7$	V
V_{IN}	Input voltage	$-0.5 \sim V_{CC} + 0.5$	V
P_D	Power dissipation (Ta = 70°C)	500	mW
T_{SOLDER}	Soldering temperature (10s)	260	°C
T_{STG}	Storage temperature	$-65 \sim 150$	°C
T_{OPR}	Operating temperature	$-20 \sim 70$	°C

4.2 DC Characteristics

$V_{CC} = 5V \pm 10\%$ TA = -20 ~ 70°C (1 ~ 10MHz)

Symbol	Item	Min	Max	Unit	Conditions
V_{IL}	Input Low Voltage (P0)	-0.3	0.8	V	–
V_{IL1}	P1, P21 ~ 25, P30 ~ 31	-0.3	$0.3V_{CC}$	V	–
V_{IL2}	$\overline{RESET}$, INT0, $\overline{NMI}$, ADS	-0.3	$0.25V_{CC}$	V	–
V_{IL3}	$\overline{EA}$	-0.3	0.3	V	–
V_{IL4}	X1	-0.3	$0.2V_{CC}$	V	–
V_{IH}	Input High Voltage (P0)	2.2	$V_{CC} + 0.3$	V	–
V_{IH1}	P1, P21 ~ 25, P30 ~ 31	$0.7V_{CC}$	$V_{CC} + 0.3$	V	–
V_{IH2}	$\overline{RESET}$, INT0, $\overline{NMI}$, ADS	$0.75V_{CC}$	$V_{CC} + 0.3$	V	–
V_{IH3}	$\overline{EA}$	$V_{CC} - 0.3$	$V_{CC} + 0.3$	V	–
V_{IH4}	X1	$0.8V_{CC}$	$V_{CC} + 0.3$	V	–
V_{OL}	Output Low Voltage	–	0.45	V	I_{OL} = 1.6mA
V_{OH} V_{OH1} V_{OH2}	Output High Voltage	2.4 $0.75V_{CC}$ $0.9V_{CC}$	–	V V V	I_{OH} = -400µA I_{OH} = -100µA I_{OH} = -20µA
I_{LI}	Input Leakage Current	–	±5	µA	$0.0 \leq Vin \leq V_{CC}$
I_{LO}	Output Leakage Current	–	±10	µA	$0.2 \leq Vin \leq V_{CC} - 0.2$
I_{CC} (Vcc - Vss)	Operating Current (RUN) Idle 1 Idle 2	–	20 3 10	mA mA mA	fosc = 10MHz
	STOP	–	10	µA	$0.2 \leq Vin \leq V_{CC} - 0.2$
Alcc (A Vcc - A Vss)	Operating Current	–	20	mA	fosc = 10MHz A Vcc = 5V ± 10%
V_{STOP}	Power Down Voltage (@STOP)	2 RAM BACK UP	6	V	V_{IL2} = 0.2V_{CC}, V_{IH2} = 0.8V_{CC}
R_{RST}	$\overline{RESET}$ Pull Up Register	50	150	KΩ	–
CIO	Pin Capacitance	–	10	pF	testfreq = 1MHz
V_{TH}	Schmitt width $\overline{RESET}$, $\overline{NMI}$, INT0, ADS	0.4	–	V	–

4.3 AC Characteristics

$V_{CC} = 5V \pm 10\%$ TA = -20 ~ 70°C (1 ~ 10MHz)
$C_L = 50pF$

Symbol	Item	Variable		10MHz Clock		Unit
		Min	**Max**	**Min**	**Max**	
t_{OSC}	Oscillation cycle (= x)	100	1000	100	–	ns
t_{CYC}	CLK cycle	4x	4x	400	–	ns
t_{WH}	CLK "L" pulse width	2x - 40	–	160	–	ns
t_{WL}	CLK "H" pulse width	2x - 40	–	160	–	ns
t_{AL}	A0 ~ A7 effective address→ALE fall	0.5x - 15	–	35	–	ns
t_{LA}	ALE fall →A0 ~ A7 hold	0.5x - 15	–	35	–	ns
t_{LL}	ALE pulse width	x - 40	–	60	–	ns
t_{LC}	ALE fall →$\overline{RD}/\overline{WR}$ fall	0.5x - 40	–	10	–	ns
t_{CL}	$\overline{RD}/\overline{WR}$ →ALE rise	0.5x - 30	–	20	–	ns
t_{ACL}	A0 ~ A7 effective address →$\overline{RD}/\overline{WR}$ fall	x - 35	–	65	–	ns
t_{ACH}	Upper effective address →$\overline{RD}/\overline{WR}$ fall	1.5x - 60	–	90	–	ns
t_{CA}	$\overline{RD}/\overline{WR}$ rise →Upper address hold	0.5x - 30	–	20	–	ns
t_{ADL}	A0 ~ A7 effective address →Effective data input	–	3.0x - 35	–	255	ns
t_{ADH}	Upper effective address →Effective data input	–	3.5x - 70	–	280	ns
t_{RD}	$\overline{RD}$ fall →Effective data input	–	2.0x - 50	–	150	ns
t_{RR}	$\overline{RD}$ pulse width	2.0x - 40	–	160	–	ns
t_{HR}	$\overline{RD}$ rise →Data hold	0	–	0	–	ns
t_{RAE}	$\overline{RD}$ rise→ Address enable	x - 20	–	80	–	ns
t_{WW}	$\overline{WR}$ pulse width	2.0x - 40	–	160	–	ns
t_{DW}	Effective data→$\overline{WR}$ rise	2.0x - 60	–	140	–	ns
t_{WD}	$\overline{WR}$ rise→Effective data hold	0.5x - 10	–	40	–	ns
t_{CPW}	CLK fall →Port Data Output	–	x + 200	–	300	ns
t_{PRC}	Port Data Input →CLK fall	200	–	200	–	ns
t_{CPR}	CLK fall →Port Data hold	100	–	100	–	ns

AC Measuring Conditions

• Output level: High 2.2V/Low 0.8V, C_L = 50pF

(however, CL = 100pF for AD0 ~ 7, A8 ~ 15, ALE, $\overline{RD}$, $\overline{WR}$)

• Input level: High 2.4V/Low 0.45V (AD0 ~ AD7)

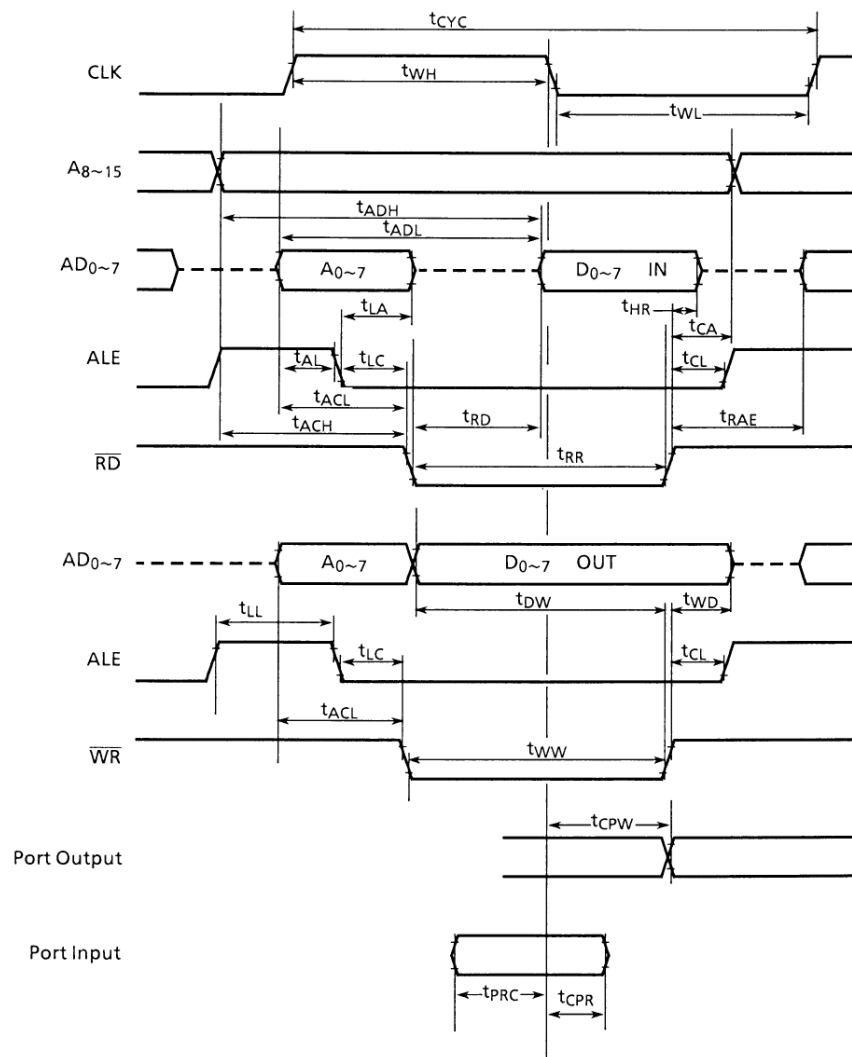High $0.8V_{CC}$/Low $0.2V_{CC}$ (excluding AD0 ~ AD7)

Figure 4.3 (1). AC Timing Diagram

4.4 A/D Conversion Characteristics

$V_{CC} = AV_{CC} = 5V \pm 10\%$ TA = -20 ~ 70°C (1 ~ 10MHz)

Symbol	Parameter	Condition	Min		Max		Unit
$V_{REF}+$	Analog reference voltage (+)	–	3.5	Vcc	Vcc		
$V_{REF}-$	Analog reference voltage (-)	–	Vss	Vss	2.5		
ΔV_{REF}	Analog reference voltage range	$V_{REF}+ - V_{REF}-$	1.0	Vcc	Vcc		V
AVss	Analog power supply voltage	–	Vss	Vss	Vss		
V_{AIN}	Analog input voltage range	–	Vss	–	Vcc		
I_{REFAD}	Analog current for analog reference voltage	–	–	.80	2		mA

This A/D Converter is guaranteed only monotonicity because it has an offset value (when V_{AIN} = 0V), but the 8-bit resolution is gotten except an offset value.

The A/D converted data is recommended to be processed relatively.
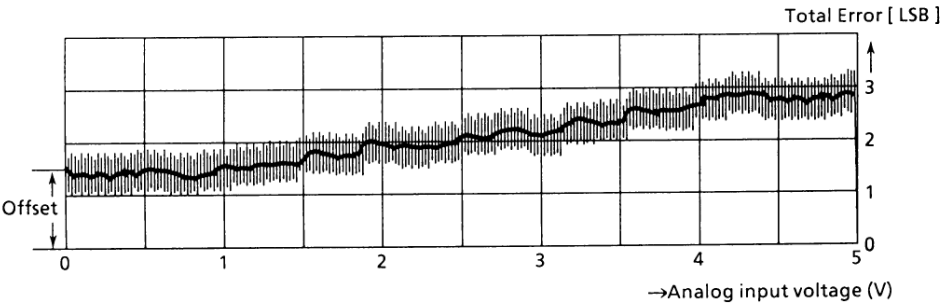

Figure 4.4 (1). A/D Converter typical conversion characterics (VREF $^+$ = 5V, VREF $^-$ = 0V)

Symbol	Item		Variable		10MHz Clock		Unit
			Min	Max	Min	Max	
t_{HADS}	High-speed conversion	ADS "H" level pulse width	2x	–	200	–	ns
t_{HADCYC}		ADS cycle	4x + 20	–	420	–	
t_{LADS}	Low-speed conversion	ADS "H" Level pulse width	2x	–	200	–	
t_{LADCYC}		ADS cycle	8x + 20	–	820	–	

x = 1/fc [ns]
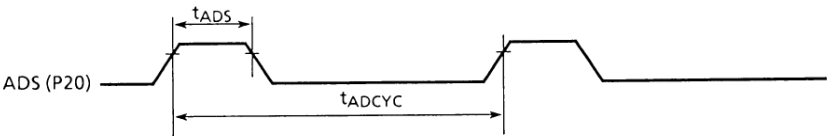

Figure 4.4 (1). A/D Converter ADS timing chart

4.5 D/A Converter Characteristics

$V_{CC} = AV_{CC} = 5V \pm 10\%$ TA = -20 ~ 70°C (1 ~ 10MHz)

Symbol	Item	Condition	Min	Typ	Max	Unit
Conversion Error *	Total Error		–	2	6	LSB
I_{DAOUT}	Output Current (When 3.75V is drived)	Vcc = A Vcc = 5V, Vss = A Vss = 0V	-0.2	-0.8	–	mA
I_{REFDA}	Analog reference voltage supply current		–	0.8	2.0	mA

*) Total Error: 1LSB = 9.77mV

4.6 Zero-Cross Characteristics

$V_{CC} = 5V \pm 10\%$ TA = -20 ~ 70°C (1 ~ 10MHz)

Symbol	Parameter	Condition	Min	Max	Unit
V_{ZX}	Zero-cross detection input	For AC, C = 0.1μF	1	1.8	$V_{AC\ P-P}$
A_{ZX}	Zero-cross accuracy	50/60Hz sine wave	–	135	mV
F_{ZX}	Zero-cross detection input frequency	–	0.04	1	kHz

4.7 16-bit Event Counter

$V_{CC} = 5V \pm 10\%$ TA = -20 ~ 70°C (1 ~ 10MHz)

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

4.8 Interrupt Operation

$V_{CC} = 5V \pm 10\%$ TA = -20 ~ 70°C (1 ~ 10MHz)

Symbol	Parameter	Variable		10MHz Clock		Unit
		Min	Max	Min	Max	
t_{INTAL}	$\overline{NMI}$, INT0 Low level pulse width	4x	–	400	–	ns
t_{INTAH}	$\overline{NMI}$, INT0 High level pulse width	4x	–	400	–	ns
t_{INTBL}	INT1, INT2 Low level pulse width	8x + 100	–	900	–	ns
t_{INTBH}	INT1, INT2 High level pulse width	8x + 100	–	900	–	ns

5. Special Function Registers (SFR) List

Special function registers (SFR) are I/O port and peripheral control registers allocated to the 48 bytes at address 0FFC0H - 0FFEFH.

(1)   Port control
(2)   Interrupt control
(3)   Timer/event counter control
(4)   Watchdog timer control
(5)   A/D and D/A converter control

Format of table

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

TMP90C846 Special Function Register Address List

FFC0	P0	FFD0	DAREG0	FFE0	TREG4L
FFC1	P0CR	FFD1	DAREG1	FFE1	TREG4H
FFC2	P1	FFD2	WDMOD	FFE2	TREG5L
FFC3	P1CR	FFD3	WDCR	FFE3	TREG5H
FFC4	IRFL	FFD4	TREG0	FFE4	T4MOD
FFC5	IRFH	FFD5	TREG1	FFE5	T4FFCR
FFC6	P2	FFD6	TREG2	FFE6	INTEL
FFC7	P2CR	FFD7	TREG3	FFE7	INTEH (DMAEL)
FFC8	P3	FFD8	TCLK	FFE8	DMAEH
FFC9	P2FR	FFD9	TFFCR	FFE9	
FFCA	DADRV	FFDA	TMOD	FFEA	
FFCB	ADSTS	FFDB	TRUN	FFEB	
FFCC	ADMOD	FFDC	CAP1L	FFEC	
FFCD	ADREG0	FFDD	CAP1H	FFED	
FFCE	ADREG1	FFDE	CAP2L	FFEE	
FFCF	ADREG1	FFDF	CAP2H	FFEF	

(Note) Addresses FFE9H - FFEFH are used as a built-in I/O reserved area and cannot be used.

(1) Port Control

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	Port0	0FFC0H	P07	P06	P05	P04	P03	P02	P01	P00
			R / W							
			Input mode (indeterminate)							
			Becomes the address/data bus when $\overline{EA}$ = 0, or during external memory access.							
P0CR	Port0 Control Reg.	0FFC1H (RMW disabled)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0							
					0 : IN	1 : OUT (bitwise I/O specification)				
P1	Port1	0FFC2H	P17	P16	P15	P14	P13	P12	P11	P10
			R / W							
			T·P put mode (0)							
			Becomes the address bus when $\overline{EA}$ = 0, or EXT·P1XC = 1. *) P1 x C:bit x of P1CR							
P1CR	Port1 Control Reg.	0FFC3H (RMW disabled)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0							
					0 : IN	1 : OUT (bitwise I/O specification)				
P2	Port2	0FFC6H	P27	P26	P25	P24	P23	P22	P21	P20
			R / W							
			Input mode (0)							
			(INT0)	($\overline{NMI}$)	(INT2/TI5)	(INT1/TI4)	(TO4)	(TO3)	(TO1)	(ADS)
P2CR	Port2 Control Reg.	0FFC7H (RMW disabled)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0							
					0 : IN	1 : OUT (bitwise I/O specification)				
P3	Port3	0FFC8H					P33	P32	P31	P30
							R / W		R	
							1		Input mode	
							($\overline{WR}$)	($\overline{RD}$)	(AN1)	(AN0)
P2FR	Port2 Function Reg.	0FFC9H	EDGE	NMIC	ZCE2	ZCE1	TO4E	TO3E	TO1E	ADSE
			R / W							
			0							
			INT0 control 0 : level 1 : ↑edge	NMI control 0 : Port 1 : $\overline{NMI}$	INT2 / TI5 control 1 : ZCD enable	INT1 / TI4 control** 1 : ZCD enable	Port / TO4 control 0 : In / Out 1 : In/TO4	Port / TO3 control 0 : In / Out 1 : In/TO3	Port / TO1 control 0 : In / Out 1 : In/TO1	Port / ADS control 0 : Port (ADS disable) 1 : ADS enable
IRFL	Port1 Function Reg & Interrupt Request Flag	0FFC4H (RMW disabled)			IRF0	IRFT0				EXT
					R					W
					0					0
					Interrupt Request Flag. 1 : Interrupt request in progress					P1 control 0: I / O Port 1: Address

(Note) Read / Write
R/W : both read and write
R : Read only
W : Write only
RMW disable : Disable Read Modify Write
(BIT, RES and SET instructions cannot be used)
Brackets indicate another name for symbols ( ).

* * : The 16-bit timer T4MOD < CAPM1, CAPM0 > is used to select INT1 rising or falling edge.

(2) Interrupt Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
			MSB							LSB
INTEL	Interrupt Enable Mask Reg.	FFE6H	IET1	IEAD	IET2	IET3	IET4	IE1	IET5	IE2
			R/W							
			0							
			1 : Enable 0 : Disable							
INTEL (DMAEL)		FFE7H			DE0	DET0			IE0	IET0
					R/W				R/W	
					0				0	
					1 : Enable 0 : Disable				1 : Enable 0 : Disable	
DMAEH	Micro DMA Enable Reg.	FFE8H	DET1	DEAD	DET2	DET3	DET4	DE1	DET5	DE2
			R/W							
			0							
			1 : Enable 0 : Disable							
IRFH	Interrupt Request Flag & IRF Clear	FFC5H	IRFT1	IRFAD	IRFT2	IRFT3	IRFT4	IRF1	IRFT5	IRF2
			R (IRF Clear Code Write only)							
			0							
			1 : Interrupt request in progress (clear IRF by writing the RF clear code)							

(Note)   IRFAD (INTAD interrupt request flipflop) is cleared by reset operations and by reading all FIFO RAM data ; therfore, it cannot be cleared by instruction.

Brackets indicate another name for symbols (   ).

(3) Timer/Event Counter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG0	8Bit Timer Reg.0	0FFD4H (RMW disabled)	–							
			W							
			Indeterminate							
TREG1	8Bit Timer Reg.1	0FFD5H (RMW disabled)	–							
			W							
			Indeterminate							
TREG2	8Bit Timer Reg.2	0FFD6H (RMW disabled)	–							
			W							
			Indeterminate							
TREG3	8Bit Timer Reg.3	0FFD7H (RMW disabled)	–							
			W							
			Indeterminate							
TCLK	8Bit Timer Source clock Control Reg.	0FFD8H	T3CLK1	T3CLK0	T2CLK1	T2CLK0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R / W							
			0							
			8-bit 00 : TO2TRG 01 : ϕT1 L0 : ϕT16 11 : ϕ256		00 : – 01 : ϕT1 10 : ϕT16 11 : ϕT256 (8-bit only)		8-bit 00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : – 01 : ϕT1 10 : ϕT16 11 : ϕT256 (8-bit only)	
TFFCR	8Bit Timer Flip–Flop Control Reg.	0FFD9H	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R / W		W		R / W	
			–		0		–		0	
			00 : Clear TFF3 01 : Set TFF3 10 : Invert TFF3 11 : Don't Care		1: TFF3 Invert Enable	0:inverted by 8-bit timer 2	00 : Clear TFF1 01 : Set TFF1 10 : Invert TFF1 11 : Don't Care		1: TFF1 Invert Enable	0:inverted by 8-bit timer 0
TMOD	8Bit Timer Mode Reg.	0FFDAH	T32M1	T32M0	PWM21	PWM20	T10M1	T10M0	PWM01	PWM00
			R / W							
			0							
			00 : 8Bit Timer 01 : 16Bit Timer 10 : 8Bit PPG 11 : 8Bit PWM		00 : – 01 : $2^6 - 1$ 10 : $2^7 - 1$ 11 : $2^8 - 1$	PWM cycle	00 : 8Bit Timer 01 : 16Bit Timer 10 : 8Bit PPG 11 : 8Bit PWM		00 : – 01 : $2^6 - 1$ 10 : $2^7 - 1$ 11 : $2^8 - 1$	PWM cycle
TRUN	8bit Timer Control Reg.	0FFDBH			PRRUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
					R / W					
					0					
					Prescaler & Timer RUN/STOP Control 0 : Stop & Clear 1 : RUN (Count up)					
CAP1L	16bit Timer/ Event Counter Capture Reg.1	0FFDCH	–							
			R							
			Indeterminate							
CAP1H		0FFDDH	–							
			R							
			Indeterminate							

Symbol	Name	Address	7	6	5	4	3	2	1	0
CAP2L	16bit Timer/ Event Counter Capture Reg.2	0FFDEH				–				
						R				
						Indeterminate				
CAP2H		0FFDFH				–				
						R				
						Indeterminate				
TREG4L	16bit Timer Event Counter Reg.4	0FFE0H (RMW disabled)				–				
						W				
						Indeterminate				
TREG4H		0FFE1H (RMW disabled)				–				
						W				
						Indeterminate				
TREG5L	16bit Timer/ Event Counter Reg.5	0FFE2H (RMW disabled)				–				
						W				
						Indeterminate				
TREG5H		0FFE3H (RMW disabled)				–				
						W				
						Indeterminate				
T4MOD	16bit Timer/ Event Counter Mode Reg.	0FFE4H			CAP1IN	CAPM1	CAPM0	CLK	T4CLK1	T4CLK0
					W			R/W		
					–			0		
					0: Soft – Capture 1: Don't care	Capture timing 00: Disable 01: T14↑ T15↑ 10: T14↑ T14↓ 11: TFF1↑ TFF1↓		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			–	
					TFF4 inversion trigger 0: Disable trigger 1: Enable trigger			00 : Clear TFF4 01 : Set TFF4 10 : Invert TFF4 11 : Don't Care	

(4) Watchdog Timer Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	Watch Dog Timer Mode Reg.	0FFD2H	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	EXF	DRIVE
			R/W						R	R/W
			0						Indeterminate	0
			1 : WDT Enable 0*:WDT Disable	Detection 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		Inverts each time the EXX instruction is executed.	1: Pin is also driven in STOP mode.
WDCR	Watch Dog Timer Control Reg.	0FFD3H (RMW disabled)	–							
			W							
			–							
			B1H:WDT Disable code 4EH:WDT Clear code							

* : To disable the watchdog timer, it is necessary to also write B1H to WDCR. Disable is not possible by merely writing "0" to WDMOD <WDTE>.

(5) A/D and D/A Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADSTS	A / D Status Reg.	FFCBH (RMW disabled)	EMPTY / R / 1 / FIFO Empty Flag 1: Empty 0: Not Empty	FULL / R / 0 / FIFO Full Flag 1: Full 0: Not Full		CLR / W / 0 / 1 : FIFO Clear (Pointer& Flag) 0 : Don't Care	Pointer / R / 0 / FIFO Pointer Indicates the number of data written to FIFO RAM (number of readable data)			
ADMOD	A / D Mode Reg.	FFCCH				ADCH / R/W / 0 / 0 : AN0 1 : AN1		SPEED / R/W / 0 / A/D conversion speed specification 0: low-speed conversion 1: high-speed conversion	RPT / R/W / 0 / A/D conversion repeat mode specification 0: one-time conversion 1: Repeat	ADS / 0 / 1 : A/D conversion start 0 : Don't Care
ADREG0	A / D Result Reg.0	FFCDH	− / *R / W / Indeterminate							
ADREG1	A/D Result Reg.1	** FFCEH (FFCFH)	− / R / Indeterminate							
DADRV	D / A Drive Reg.	FFCAH								DA0DR / R / W / 0 / 0 : 0V output 1 : register value conversion output
DAREG0	D / A Conversion Reg.0	FFD0H (RMW disabled)	− / W / 0 / Starts D/A conversion with register write and outputs to D-A0.							
DAREG1	D / A Conversion Reg.1	FFD1H (RMW disabled)	− / W / 0 / Starts D/A conversion with register write and outputs to D-A1.							

* : Do not write to ADREG0 during A/D conversion.

** : ADREG1 has two addresses: FFCEH and FFCFH. Both can be read but use address FFCEH when using the 1-byte load instruction.

6. Port Block Equivalent Circuit Diagram

• Reading the circuit diagram

The gate symbols used are basically the same as those used for the standard CMOS logic IC "74HCXX" Series.
The signal names include the following special cases.

STOP:This signal sets the halt mode specification register to the STOP mode (WDMOD <HALTM1, 0> = 0, 1) and becomes active "1" when the CPU executes the HALT instruction.

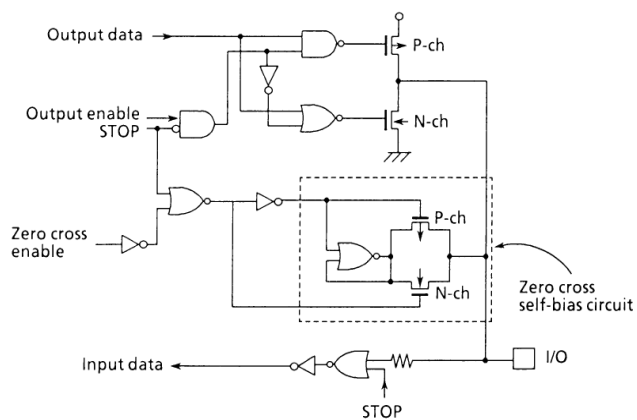• Guaranteed input resistance ranges from several tens of ohms to several hundred ohms.

• P0 (AD0 ~ AD7), P1 (A8 ~ A15), P21 ~ 23
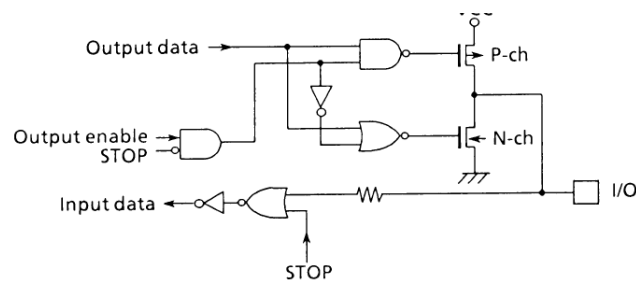
• P20 (ADS), P26 ($\overline{\text{NMI}}$), P27 (INT0)
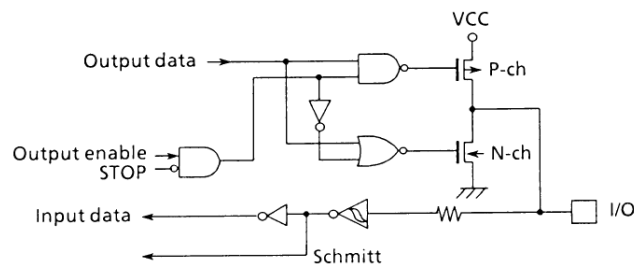
• P24 (INT1/TI4), P25 (INT2/TI5)

• P30 (AN0), P31 (AN1)

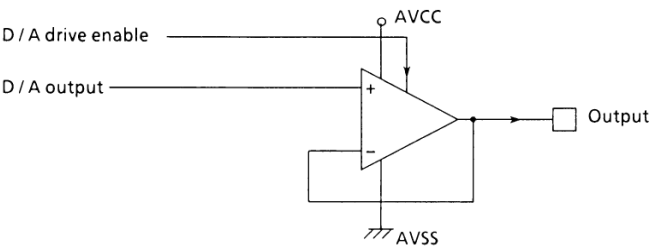• P32 ($\overline{\text{RD}}$), P33 ($\overline{\text{WR}}$)

● VREF$^+$, VREF$^-$

VREF +

I_{REFAD} = 0.8mA typ.

A / D converter
ladder resistance

(Note) I_{REFAD} continues to flow even
during standby.

VREF −

● AVCC, AVSS

I_{REFDA} = 0.8mA typ. AVCC

D / A converter
ladder resistance

(Note) I_{REFDA} continues to flow even
during standby.

AVSS

A / D, D / A converter power
supply

● D - A0, D - A1

D / A drive enable

AVCC

D / A output

+

−

Output

AVSS

• CLK

• X1, X2

• $\overline{\text{EA}}$

* $\overline{\text{RESET}}$

* ALE

TOSHIBA CORPORATION