CMOS 8–Bit Microcontroller

TMP90CM36F/TMP90CM36T

1. Outline and Characteristics

The TMP90CM36 is a high-speed, high performance 8-bit microcontroller developed for application in the control of various devices.

The TMP90CM36, CMOS 8-bit microcontroller, integrates an 8-bit CPU, ROM, RAM, A/D converter, D/A converter, multi-function timer/event counter, general-purpose serial interface, signal measure circuit, timing pulse generation circuit and PWM output in a single chip, and with which external program memory and data memory can be extended up to 31KB.

The TMP90CM36F is a device with an 80-pin flat package.

The TMP90CM36T is a device with an 84-pin QF (PLCC) package.

The following are the features of the TMP90CM36:

(1) Highly efficient instruction set: 167 basic instructions
Division and multiplication instructions, 16-bit operation instructions and bit operation instructions

(2) Minimum instruction executing time: 250ns (at 16MHz oscillation frequency)

(3) Built-in ROM: 32K bytes

(4) Built-in RAM: 1K bytes

(5) Memory extension capability
External program memory: 31K bytes
External data memory: 31K bytes

(6) Interrupt functions: 18 internal, 5 external

(7) 8-bit A/D converter (8 channels)

(8) 8-bit D/A converter (2 channels)

(9) 12-bit D/A conversion (pulse width change of tone) output (2 channels)

(10) General-purpose serial interface mode (3 channels)
- With asynchronous mode and I/O interface mode (1 channel)
- With synchronous mode (1 channel)
- I/O interface mode (1 channels)

(11) Timer function
(1) 20-bit time base counter
(2) 24-bit capture with 4 step FIFO
(3) 24-bit timing pulse generation circuit (2 channels)
(Comparator data 16-bit + timing output 8-bit) x 4 step FIFO
(4) 16-bit timer/event counter (1 channel)
----- Built-in 2 capture register and 2 comparator
(5) 8-bit timer (4 channels)
----- Built-in 1 comparator in each channel
(6) Watchdog timer function ($\overline{\text{WDTOUT}}$ pin having)

(12) I/O ports: MAX64 pins

(13) HDMA function (2 channels) ----- 1 byte transmission: 1.75μs (@16.0MHz)

(14) Software standby function ----- RUN, STOP, IDLE modes
Hardware standby function ----- STOP mode

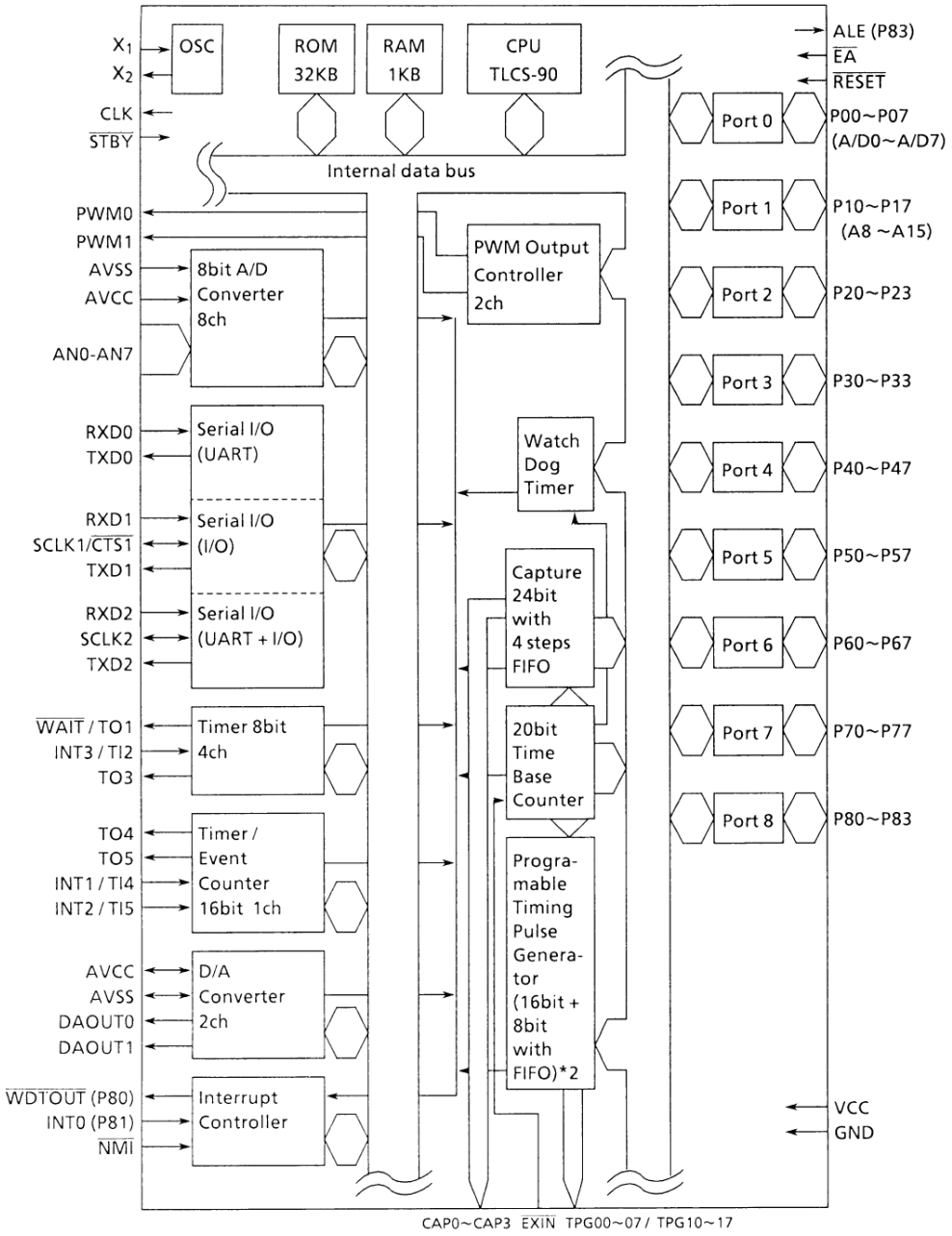


Figure 1. TMP90CM36 Block Diagram

2. Pin Assignment and Functions

The assignment of input/output pins for TMP90CM36, their name and outline functions are described below.

2.1 Pin Layout Diagram

Figure 2.1 (1) shows the pin assignment of TMP90CM36F.

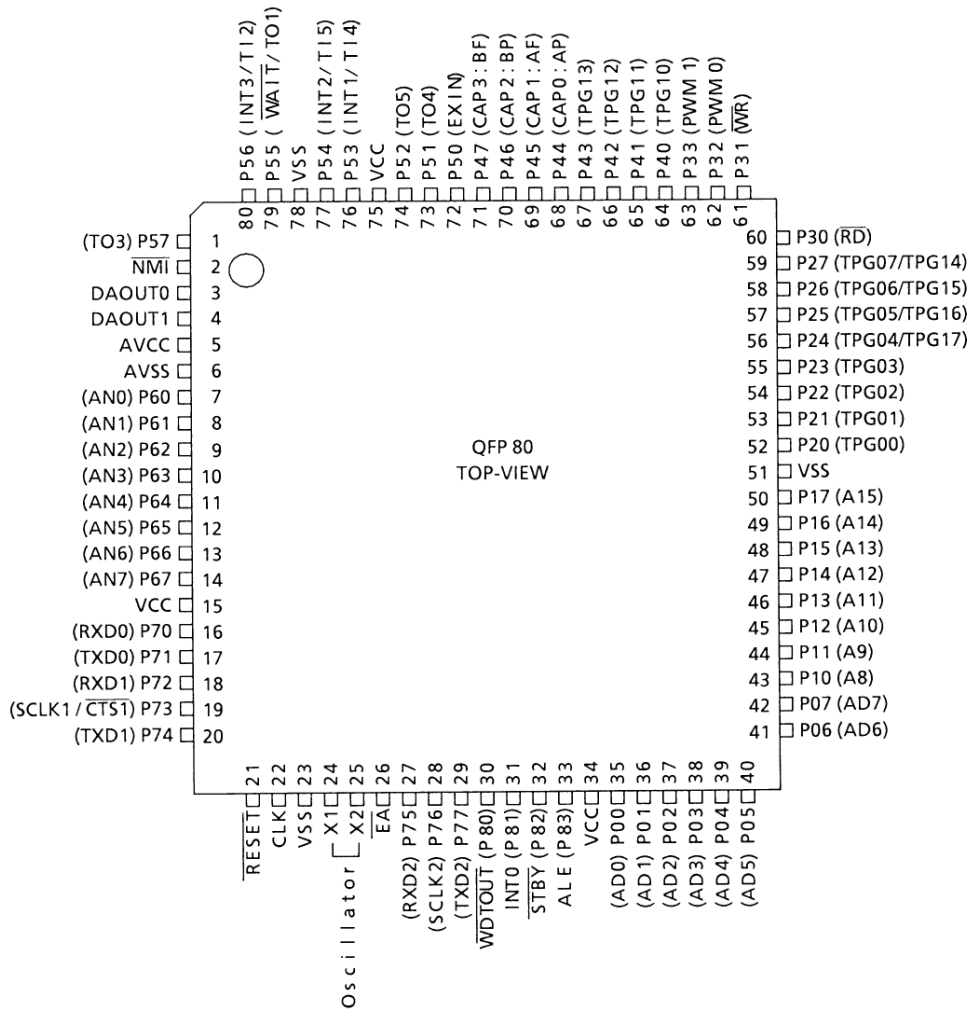


Figure 2.1 (1). Pin Assignments (Flat Package)

Figure 2.1 (2) shows the pin assignment of TMP90CM36T.

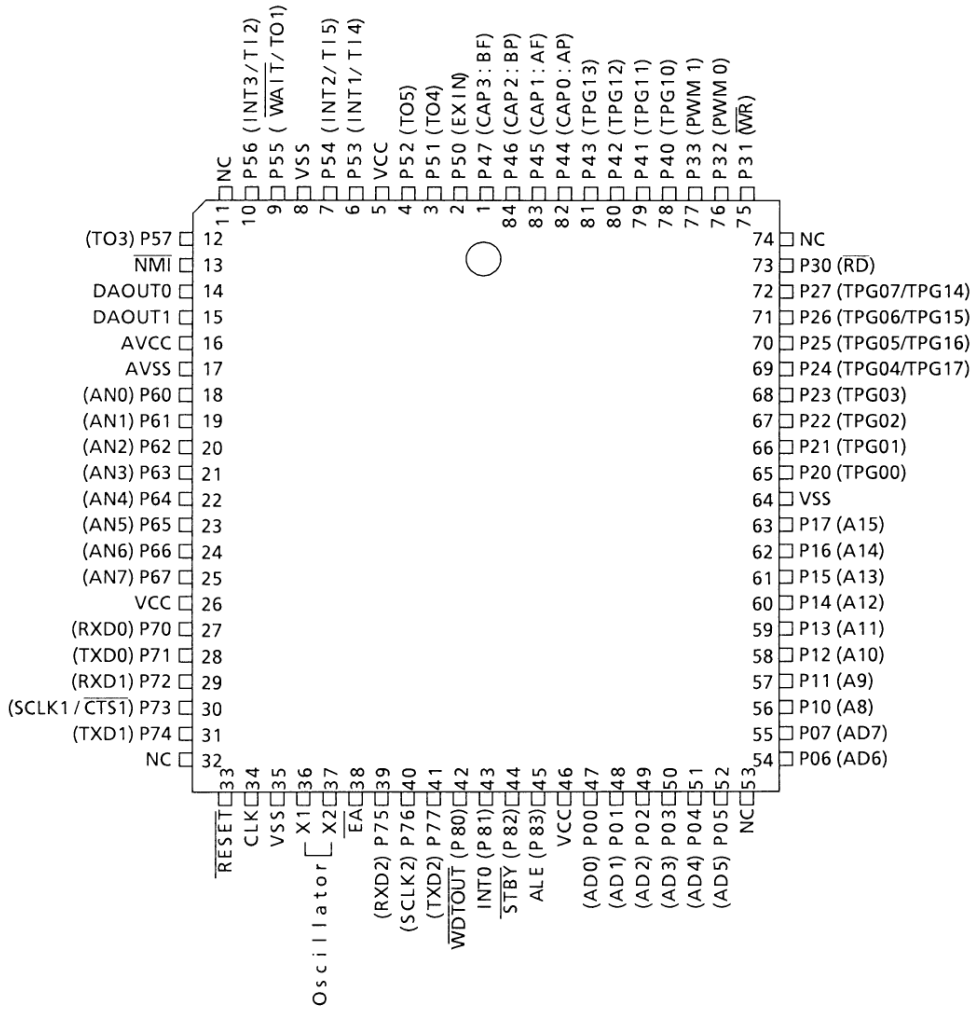


Figure 2.1 (2). Pin Assignments (QFJ (PLCC) Package)

2.2 Pin Names and Functions

The names of input/output pins and their functions are described below. Table 2.2 Shows the input/output Pin Names and Functions.

Table 2.2. Pin Names and Functions (1/4)

Pin name	No. of pins	I/O or tristate	Function
P00~P07	8	I/O	Port 0 : An 8-bit I/O port. Each bit can be set for input or output.
/AD0~AD7		Tristate	Address/data bas : Operates as an 8-bit bi-directional address bus or data bus when using external memory. Operates port when built-in memory used.
P10~P17	8	I/O	Port1: An 8-bit I/O port. Each bit can be set for input or output.
/A8~A15		Output	Address bus: Operates as the 8 upper bits of the address bus when using external memory. Operates port when built-in memory used.
P20~P27 /TPG00~ TPG07 (TPG14~ TPG17)	8	I/O	Port2: An 8-bit I/O port. Each bit can be set for input or output.
		Output Tristate	TPG0 output: Operates of the TPG00~TPG07 output pin. 4 upper bits are shared in TPG14~TPG17. Operates as Reset at once of the 3 state condition. Operates output condition of agreement for output data write and comparator data.
P30	1	Output	Port30: A 1-bit output port.
/$\overline{RD}$			Read: Strobe signal output for reading external memory.
P31	1	Output	Port31: A 1-bit output port.
/$\overline{WR}$			Write: Strobe signal output for writing external memory.
P32, P33	2	I/O	Port32, Port33: A 2-bit I/O port. Each bit can be set for input or output.
/PWM0, PWM1		Output Tristate	Motor control output : PWM0/1 motor control output pin.
P40~P43 /TPG10~ TPG13	4	I/O	Port40~43: A 4-bit I/O port. Each bit can be set for input or output.
		Output Tristate	TPG1output : Operates of the TPG10~TPG13 output pin. TPG14~TPG17 are shared in TPG04~07. Operates as Reset at once of the 3 state condition. Operates output condition of agreement for output data write and comparator data.

Table 2.2. Pin Names and Functions (2/4)

Pin name	No. of pins	I/O or tristate	Function
P44~P47 /CAP0~ CAP3	4	I/O	Port44~47:A 4-bit I/O port. Each bit can be set for input or output.
		Input	Capture input : CAP0~CAP3 input pin.
P50 /EXIN	1	I/O	Port50: A 1-bit I/O port.
		Input	Time base counter to external clock input pin.
P51 /TO4	1	I/O	Port51: A 1-bit I/O port.
		Output	Timer Output4: Timer 4 output pin.
P52 /TO5	1	I/O	Port52: A 1-bit I/O port.
		Output	Timer Output5: Timer 5 output pin.
P53 /INT1 /TI4	1	I/O	Port53: A 1-bit I/O port.
		Input	Interrupt request pin1: A rising/falling edge programmable interrupt request pin.
		Input	Timer input4: Timer 4 counter input pin.
P54 /INT2 /TI5	1	I/O	Port54: A 1-bit I/O port.
		Input	Interrupt request pin2: A rising edge interrupt request input pin.
		Input	Timer input5: Timer 5 counter input pin.
P55 /TO1/$\overline{\text{WAIT}}$	1	I/O	Port55: A 1-bit I/O port.
		Output / Input	Timer output0/1: Timer 0 or timer 1 output. / Wait: Input pin for connecting a memory or peripheral LSI with delayed access time.
P56 /TI2 /INT3	1	I/O	Port56: A 1-bit I/O port.
		Input	Timer input2/3: Timer 2 or Timer 3 counter input pin.
		Input	Interrupt request pin3: A rirising edge interrupt request input pin.
P57 /TO3	1	I/O	Port57: A 1-bit I/O port.
		Output	Timer Output2/3: Timer 2 or Timer 3 output.
$\overline{\text{NMI}}$	1	Input	Non-maskable Interrupt request pin: A falling edge interrupt request pin (Sctimitt input).
/DAOUT0, DAOUT1	2	Output	D/A Output: D/A converter 0/1 analog current output pin.

Table 2.2. Pin Names and Functions (3/4)

Pin name	No. of pins	I/O or tristate	Function
AVCC	1	Input	Reference Voltage to A/D converter.
AVSS	1	Input	GND pin for A/D converter (0V).
P60~P67	8	Input	Port60~67: An 8-bit input ports.
/AN0~AN7			Analog input: 8 analog inputs to A/D converter.
VCC	2	Input	Power supply pin (+ 5V).
VSS	3	Input	GND pin (0V).
P70	1	I/O	Port70: A 1-bit I/O port.
/RXD0		Input	Serial channel 0 receive data input pin.
P71	1	I/O	Port71: A 1-bit I/O port.
/TXD0		Output	Serial channel 0 send data output pin.
P72	1	I/O	Port72: A 1-bit I/O port.
/RXD1		Input	Serial channel 1 receive data input pin.
P73	1	I/O	Port73: A 1-bit I/O port.
/SCLK1		I/O	Serial clock I/O : External clock for SCLK1 do input or send clock for internal boudrate generator do output when I/O interface mode is condition.
/CTS1		Input	$\overline{CTS1}$ input pin: Serial data send possible signal (Clear To Send).
P74	1	I/O	Port74: A 1-bit I/O port.
/TXD1		Output	Serial channel 1 send data output pin.
P75	1	I/O	Port75: A 1-bit I/O port.
/RXD2		Input	Serial channel 2 receive data input pin.
P76	1	I/O	Port76: A 1-bit I/O port.
/SCLK2		I/O	Serial clock I/O: External clock for SCLK2 do input or send clock for internal boudrate generator do output when I/O interface mode is condition.
P77	1	I/O	Port77: A 1-bit I/O port.
/TXD2		Output	Serial channel 2 send data output pin.
$\overline{RESET}$	1	Input	Reset: Reset input pin to initialize the TMP90CM36.

Table 2.2. Pin Names and Functions (4/4)

Pin name	No. of pins	I/O or Tristate	Function
CLK	1	Output	Clock output: Output 1/4 frequency of the clock oscillation. Pulled up during reset.
X1, X2	2	I/O	The crystal oscillator connectopn pin.
$\overline{EA}$	1	Input	External access: Connects to the Vcc pin when the TMP90CM36 built-in ROM is used.
P80	1	Output	Port80: A 1-bit output port.
/$\overline{WDTOUT}$		Output	Wachdog out: Operates $\overline{WDTOUT}$ output pin when watchdog timer register is D1 = 1
P81	1	Input	Port81: A 1-bit input port.
/INT0		Input	Interrupt request pin0: A level/rising edge programmable interrupt request pin.
P82	1	Input	Port82: A 1-bit input port.
/$\overline{STBY}$		Input	Hardware standby input pin (schmitt input).
P83	1	Output	Port83: A 1-bit output port.
/ALE		Output	Address latch enable : The falling edge of this signal is used for latching addresses on AD0-AD7 when accessing external memory.

3. Operation

This section explains the functions and basic operations of the TMP90CM36 in blocks.

3.1 CPU

The TMP90CM36 has a built-in high performance 8 bit CPU.

For the operation of the CPU, see the book TLCS 90 Series CPU Core Architecture.

This section explains the CPU functions unique to the TMP90CM36 that are not explained in that book.

3.1.1 Reset

Figure 3.1 (1) shows the basic timing of reset.

To reset the TMP90CM36, it is required that the power supply voltage is within operating range, the internal oscillator is stably functioning, and $\overline{\text{RESET}}$ input be kept at “0” for at least 10 system clocks (10 states: 2 microseconds with 10MHz system clock)

When a reset is accepted, among I/O common ports, port 0 (address data bus A0 - A7), port 1 (address data bus A8 - A15) and port 2 are set to input status (with high impedance). Output ports P30 ($\overline{\text{RD}}$) and P31 ($\overline{\text{WR}}$) and CLK are set to “1” and ALE (P83) is cleared to “0”.

CPU registers and external memory are not changed. However, program counter PC and interrupt enable/disable flag IFF are cleared to “0”. The A register becomes undefined.

When the reset is released, instruction execution starts from address 0000H.

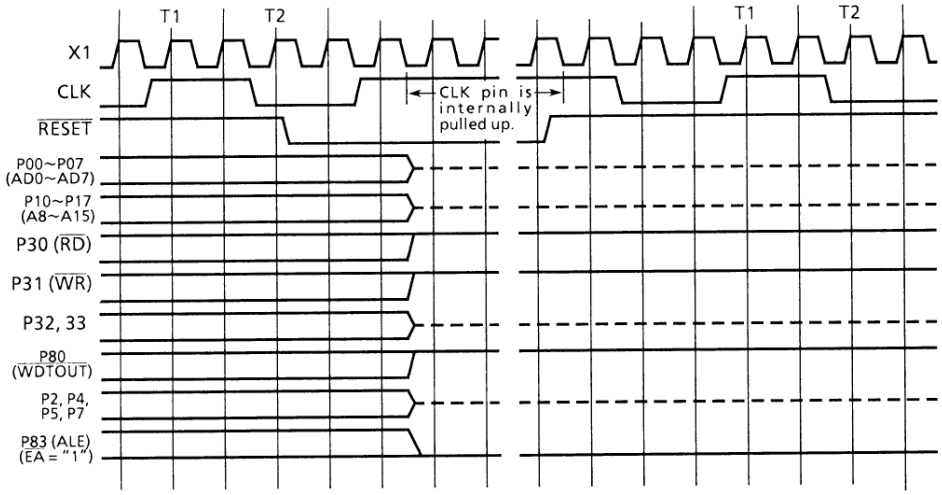


Figure 3.1 (1). Reset Timing of TMP90CM36

3.1.2 EXF (Exchange Flag)

The exchange flag EXF is inverted when the EXX instruction is executed to exchange data between the TMP90CM36 main registers and auxiliary registers. This flag is assigned to bit 7 at memory address FFE1H.

IRF1 (FFE1H)

	7	6	5	4	3	2	1	0
bit Symbol	EXF	—	IRFT5	IRFT4	IRFT3	IRFT2	IRFT1	IRFT0
Read/Write	R	R						
After reset	Un-deFined		0	0	0	0	0	0
Function	Inverts each times the EXX instructio n is executed	Interruput Request Flag 1: Interrupt is currently requested.						

3.1.3 Wait Control

For the TMP90CM36, a wait control register (WAITC) is assigned to bits 4 and 5 at memory address FFB4H.

P5FR (FFB4H)

	7	6	5	4	3	2	1	0
bit Symbol	EXINE	–	WAITC1	WAITC0	TO5S	T04S	T03S	TO1S
Read/Write	R/W		R/W	R/W	R/W			
After reset	0		0	0	0	0	0	0
Function	P50 Control 0: Port 1: EXIN		WAIT Control 00: 2 state wait 01: normal Wait 10: non wait 11: –		P52 Control 0: Port 1: TO5	P51 Control 0: Port 1: TO4	P57 Control 0: Port 1: TO3	P55 Control 0: Port 1: TO1

3.2 Memory Map

The TMP90CM36 can provide a maximum 64K byte program memory and data memory.

The program and data memories may be allocated to the addresses 0000H ~ FFFFH.

(1) Built-in ROM

The TMP90CM36 has an internal 32K byte ROM. This ROM is located at addresses 0000H ~ 7FFFH. Program execution starts from address 000H after a reset operation.

Addresses 0008H ~ 0078H in the internal ROM area are used as the interrupt processing entry area.

(2) Built-in RAM

The TMP90CM36 contains a 1K byte built-in RAM which is allocated to the addresses FBA0H ~ FF9FH.

The CPU can also access some portions of the RAM (160 byte area FF00H ~ FF9FH) using short instruction codes in the direct addressing mode.

(3) Built-in I/O

The TMP90CM36 uses 96 bytes of the address space as a built-in I/O area. The area is allocated to the addresses FFA0H ~ FFFFH. The CPU can access the built-in I/O using short instruction codes in the direct addressing mode.

Figure 3.2 shows the memory map and the access ranges of the CPU for each addressing mode.

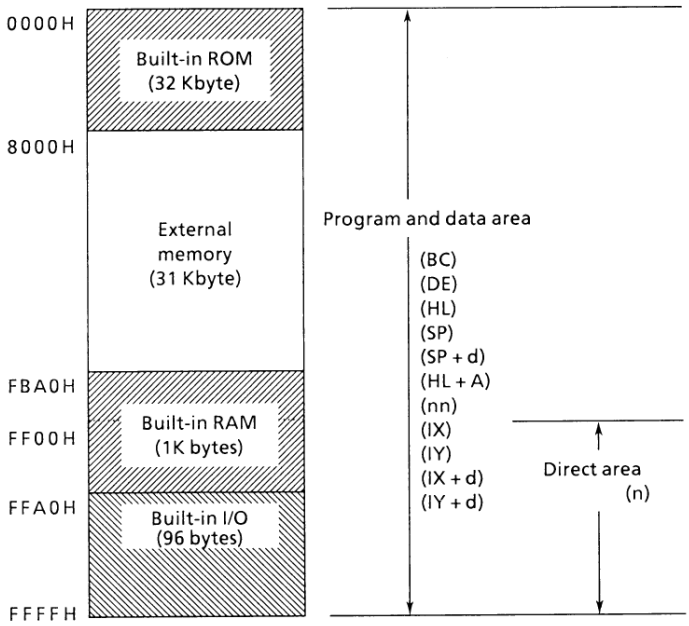


Figure 3.2. Memory Map

3.3 Interrupt Functions

The TMP90CM36 has a general-purpose interrupt processing routine for responding to both internal and external interrupt request, and a high-speed micro DMA (HDMA) processing mode in which the CPU automatically transfers data.

Immediately after a reset is released, all responses to interrupt requests are set to the general-purpose interrupt processing mode.

The high-speed DMA processing mode can be set by loading a vector value to the DMAV 0/1 register.

Figure 3.3 (1) shows the interrupt response flow.

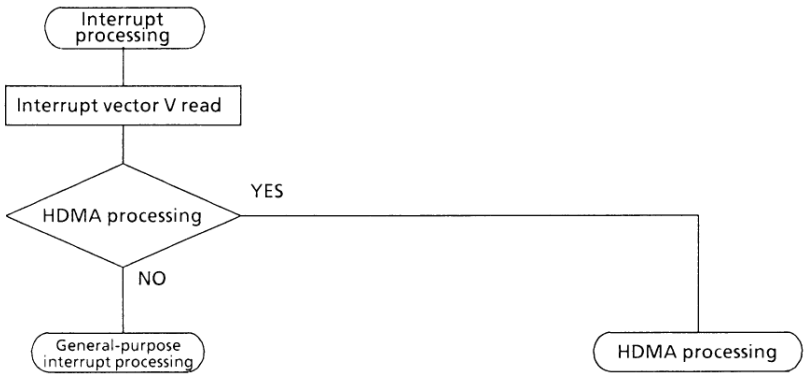


Figure 3.3 (1). Interrupt Response Flow

When an interrupt request is generated, this is reported to the CPU via the built-in interrupt controller. If the request is for a non-maskable interrupt or an enabled maskable interrupt, the CPU starts interrupt processing. If for a disabled maskable interrupt, the request is ignored and not received.

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

Next, a check is made as to whether this request is for general-purpose interrupt processing, micro DMA processing or high-speed DMA (HDMA) processing, and then the corresponding processing is performed.

The interrupt vector is read in an internal operation cycle so the bus cycle becomes a dummy cycle.

3.3.1 General-Purpose Interrupt Processing

Figure 3.3 (2) shows the general-purpose interrupt processing flow.

The CPU first saves the contents of the program counter PC and register AF (including the interrupt enable/disable flag IFF immediately before an interrupt) to the stack and then resets the interrupt enable/disable flag IFF to "0" (interrupt disable). Finally, the interrupt vector contents [V] are transferred to the program counter and a jump is made to the interrupt processing program.

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

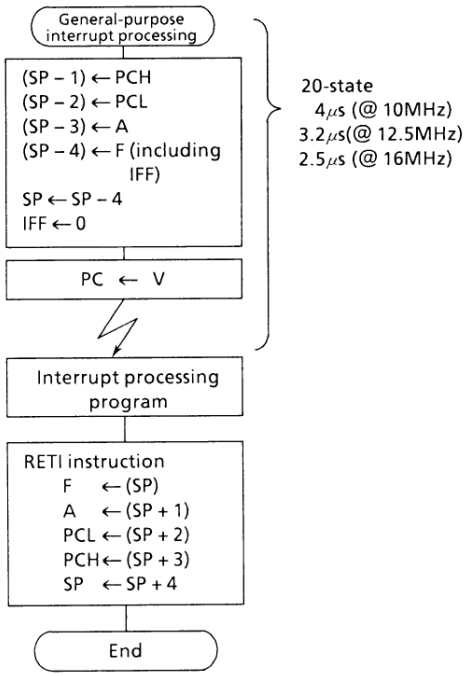


Figure 3.3 (2). General-Purpose Interrupt Processing Flow

Interrupt processing program is ended with the RETI instruction for both maskable and non-maskable interrupts.

Executing this instruction restores the program counter PC and register AF contents from the stack. (Resets the interrupt enable/disable flag immediately before an interrupt.)

When the CPU reads the interrupt vector, the interrupt request source confirms that the interrupt has been received and then clears the interrupt request. Non-maskable interrupts cannot be disabled by program. Maskable interrupts, however, can be enabled and disabled by program. Bit 5 of CPU register F is an interrupt enable/disable flipflop (IFF). Interrupts are enabled by setting this bit to "1" with the EI (interrupt enable) instruction and disabled by resetting this bit to "0" with the DI (interrupt disable) instruction. IFF is reset to "0" by resetting and when an interrupt is received (including non-maskable interrupts).

The EI instruction is actually executed after the next instruction is executed.

Table 3.3 (1) shows the interrupt sources.

Table 3.3 (1) Interrupt Sources

Priority sequence	Type	Interrupt function name	Interrupt vector value	General purpose interrupt processing start address	Interrupt sources		Comments
					Internal	External	
1	Non maskable	SWI instruction	10H	0010H	0	–	–
2		NMI	18H	0018H	–	NMI pin	–
3		INTWD	20H	0020H	WDT	–	–
4	maskable	INT0	28H	0028H	–	INT0 pin	External 0
5		INTCAP	30H	0030H	Capture Input	–	Capture
6		INTTPG0	38H	0038H	TPG0	–	TPG
7		INTTPG1	40H	0040H	TPG1	–	TPG
8		INTRX0	48H	0048H	Serial receive and	–	SIO
9		INTTX0	50H	0050H	Serial send end	–	SIO
10		INTT0	58H	0058H	Timer 0	–	Timer
11		INTT1	60H	0060H	Timer 1	–	Timer
12		INTT2	68H	0068H	Timer 2	–	Timer
13		INTAD	70H	0070H	AD	–	AD
14		INTT3	78H	0078H	Timer 3	–	Timer
15		INTT4	80H	0080H	Timer 4	–	Timer
16		INT1	88H	0088H	–	INT1 pin	External 1
17		INTT5	90H	0090H	Timer 5	–	Timer
18		INT2	98H	0098H	–	INT2 pin	External 2
19		INTRX1	A0H	00A0H	Serial receive and	–	SIO
20		INTTX1	A8H	00A8H	Serial send end	–	SIO
21		INT3	B0H	00B0H	–	INT3 pin	External 3
22		INTRX2	B8H	00B8H	Serial receive and	–	SIO
23		INTTX2	C0H	00C0H	Serial send end	–	SIO

The “priority sequence” shown in Table 3.3 (1) indicates the sequence in which interrupt sources are received by the CPU when multiple interrupt requests are generated simultaneously.

For example, if interrupt requests with the priority sequences 4 and 5 are generated simultaneously, the CPU will receive the interrupt request with priority sequence 4 first. When processing of the interrupt with priority sequence 4 is ended with the RETI instruction, the CPU will then receive the interrupt with priority sequence 5.

If the interrupt processing program for the priority sequence 4 interrupt is interrupted by executing the EI instruction, the CPU will receive the priority sequence 5 interrupt request. When multiple interrupt requests are generated simultaneously, the built-in interrupt controller only determines the priority sequence of the interrupt sources received by the CPU. There is no function to compare the priority sequence of the interrupt currently being processed and the interrupt currently being requested.

Another interrupt can be enabled while another interrupt is being processed by resetting the interrupt enable/disable flag IFF to enable.

3.3.2 High-Speed Micro DMA Processing

The TMP90CM36 has two built-in DMA channels called HDMA.

HDMA has three times the processing capacity of μDMA and is used for high-speed data transfers. HDMA execution time (decrease the value of transfer number and the value is not "0" data) is 14 states, regardless of whether the 1-byte transfer mode or 2-byte transfer mode is used. HDMA and micro DMA (the TMP90CM36 has not the micro DMA) transfer speeds.

Table 3.3 (3) shows the high-speed micro.

Table 3.3 (3) Transfer Speeds

fxtal (MHz)	HDMA	Micro DMA
10	2.8 μs	9.2 μs
12.5	2.24 μs	7.36 μs
16	1.75 μs	5.75 μs

* At 1 byte transfer mode.

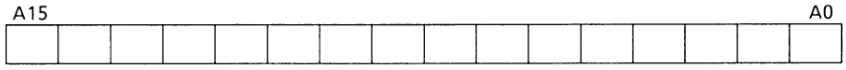
Table 3.3 (4) Shows the DHMA Functions

Number of channels	2
Transfer speed	14 states (for 1 byte) or 18 states (for 2 bytes)
Start method	By interrupt (all external and internal interrupt sources)
Transfer mode	1-byte transfer or 2-byte transfer
Address output method	Dual address (source/destination)
Access area	0~8M-byte memory area (64k-byte units)

(1) HDMA Setting Registers

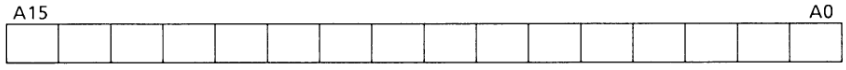
The following describes the registers required for HDMA operation.

(a) DMAS0, 1 : source address register (16-bit)



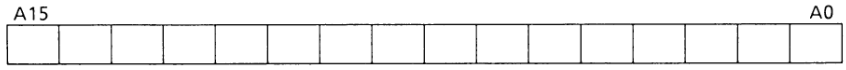
Sets the transfer source address.

(b) DMAD0, 1 : destination address register (16-bit)



Sets the transfer destination address.

(c) DMAC0, 1 : byte count register (16-bit)



Sets the number of bytes to be transferred.

The set value is decremented (− 1) for each HDMA started.
A generation purpose interrupt is when the value reaches "0".

(d) DMAM0, 1 : transfer mode register (8-bit)

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–					

D4	Transfer byte count
0	1-byte transfer
1	2-byte transfer

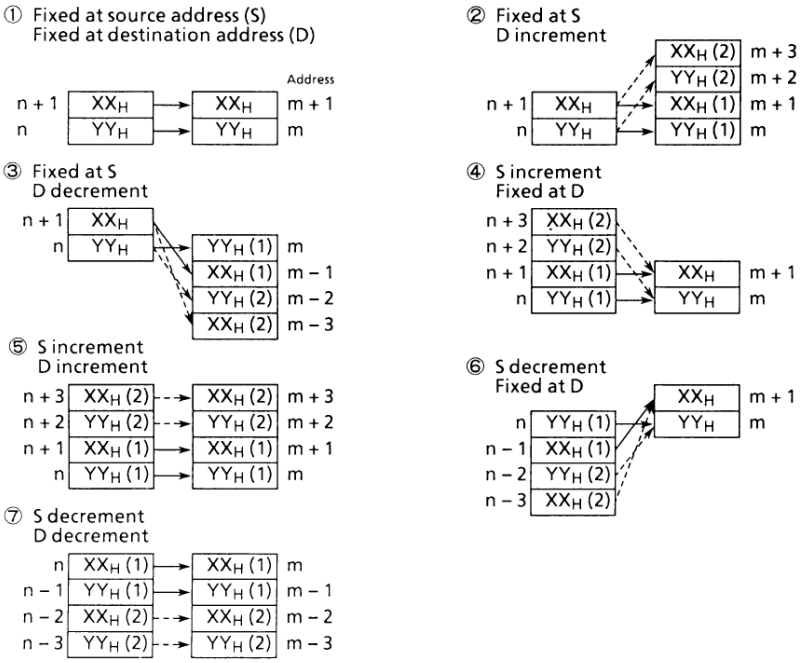
D3	D2	Source address
0	0	Fixed
0	1	Increment
1	0	Decrement

D1	D0	Destination address
0	0	Fixed
0	1	Increment
1	0	Decrement

This register determines the HDMA transfer mode.
The bits of this register are as shown in the table above.
The source and destination addresses shown here are the addresses loaded to DMAS0, 1 and DMAD0, 1 above.

Example 1 : ×××00001
Transfers 1 byte, fixes transfer source address (DMAS0, 1), and increments the transfer destination address after each transfer.

Example 2 : The renewal of address at 2-byte transfer mode.



Note: It is ineffective to set decrement for a destination address when a source address being increment; and to set increment for a destination address when a source address being decrement.

(e) DMAV0, 1 : DMA vector register

DMAV0 (FFE8H)

	7	6	5	4	3	2	1	0
bit Symbol	DV07	DVO6	DV05	DV04	DV03	DV02	DV01	DV00
Read/Write	W							
Resetting Value	0	0	0	0	0	0	0	0

DMAV11 (FFEBH)

	7	6	5	4	3	2	1	0
bit Symbol	DV17	DV16	DV15	DV14	DV13	DV12	DV11	DV10
Read/Write	W							
Resetting Value	0	0	0	0	0	0	0	0

HDMA is started by interrupts. Consequently, the vector address of the interrupt that assigns HDMA start is loaded to the DMA vector register (DMAV0/1).

HDMA compares the interrupt vector and the contents of this register. If they match, HDMA operation starts.

It is necessary to set the vector address before generating the interrupt that starts HDMA.

(2) Register Loading

(a) DMAS0, 1
DMAD0, 1
DMAC0, 1
DMAM0, 1
} Loaded with the LDC instruction.
(* The LDC instruction is a new TLCS-90 instruction.)

(b) DMAV0, 1 Load the input / output address with the LD instruction.
(See the separate address map concerning input / output addresses.)

(3) HDMA Start

HDMA can be started by any of the following TMP90CM36 maskable interrupt sources

(a) Internal start factors

- SWI (software)
- All internal I/O interrupts

Assign starting of HDMA channel 0 or channel 1 to the INT0 - INT3 external interrupts, connect any of the bits of ports 0 - 8 (output mode) externally to INT0 - INT3 to generate a start interrupt.

(b) External start factors

- $\overline{\text{NMI}}$ pin
- INT0 ~ 3 pin

(4) HDMA Channel 0 and Channel 1 Priority Sequence

The channel where an interrupt is generated first has priority.

Note: HDMA, regardless of an interrupt enable flag, compares the vector and the values of the DMA V0/1 register. If they match in EI mode, the HDMA starts.

Do not write the vector value of the non-maskable interrupt to the DMA V0/1 register. If doing so, the HDMA does not operate normally.

To stop the HDMA from being started, set DI mode before generating the interrupt to start the HDMA, or set the DMA V0/1 register to 00H.

(5) HDMA Operation Flow

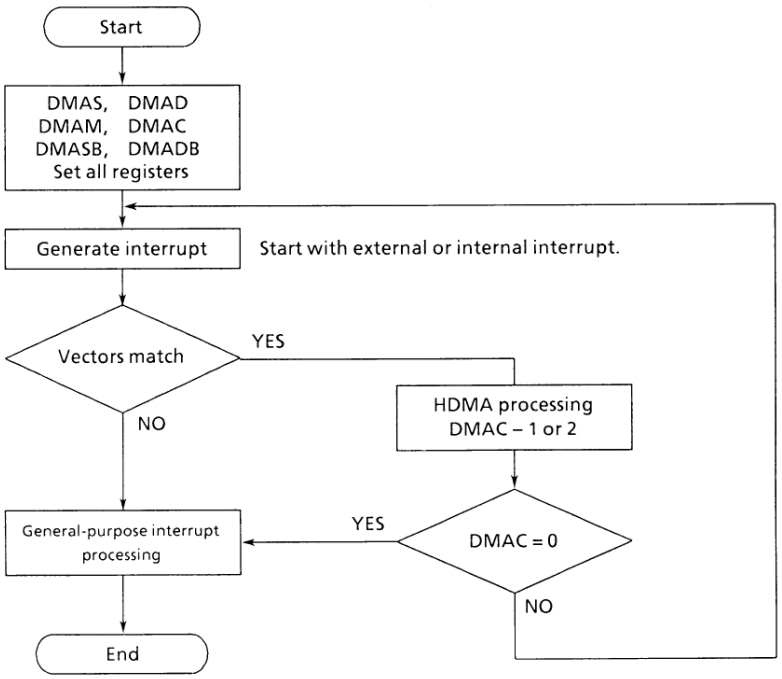
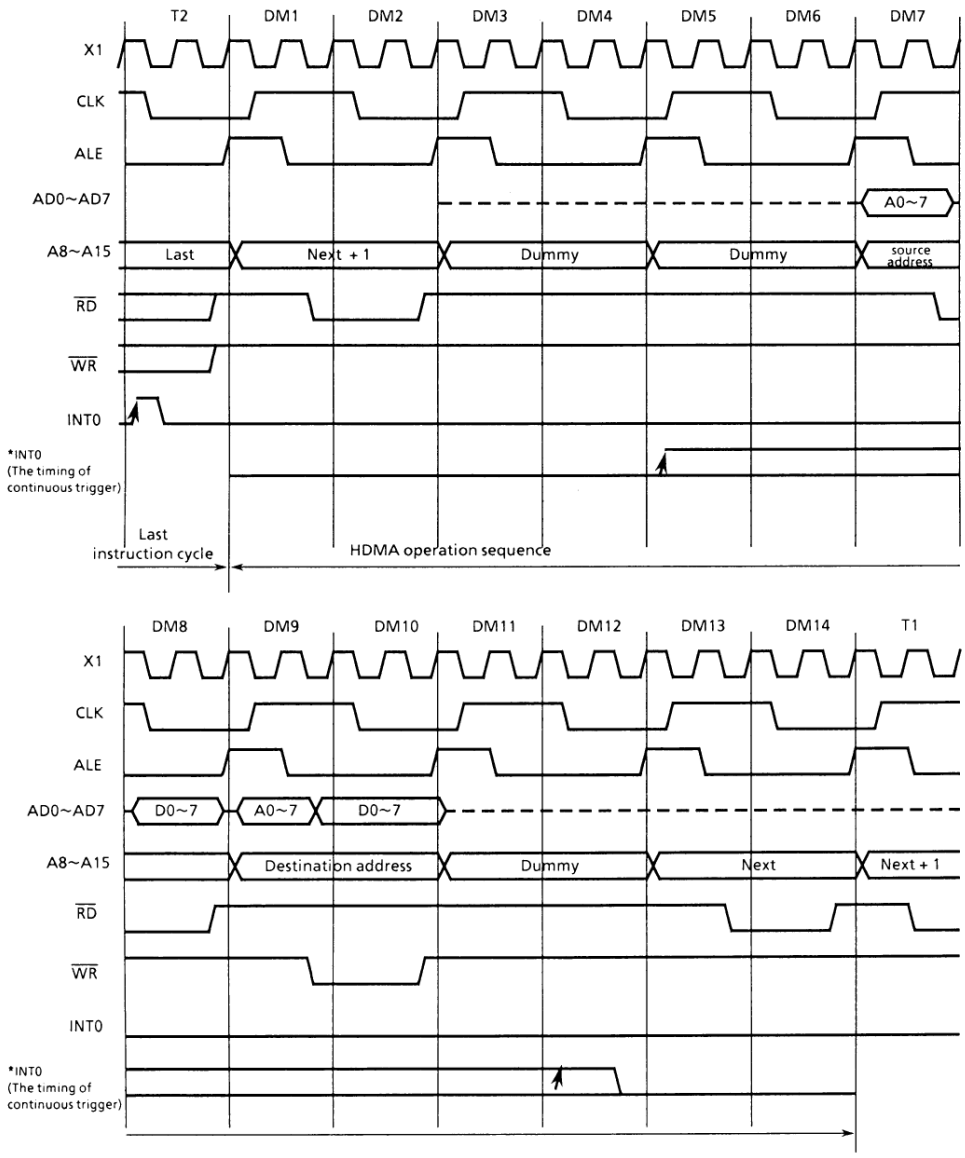


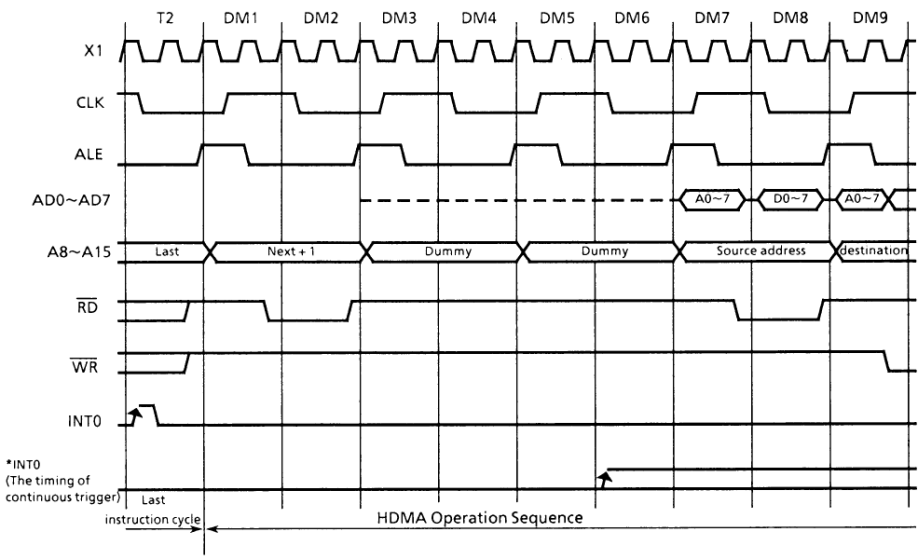
Figure 3.3 (6). HDMA Operation Flow

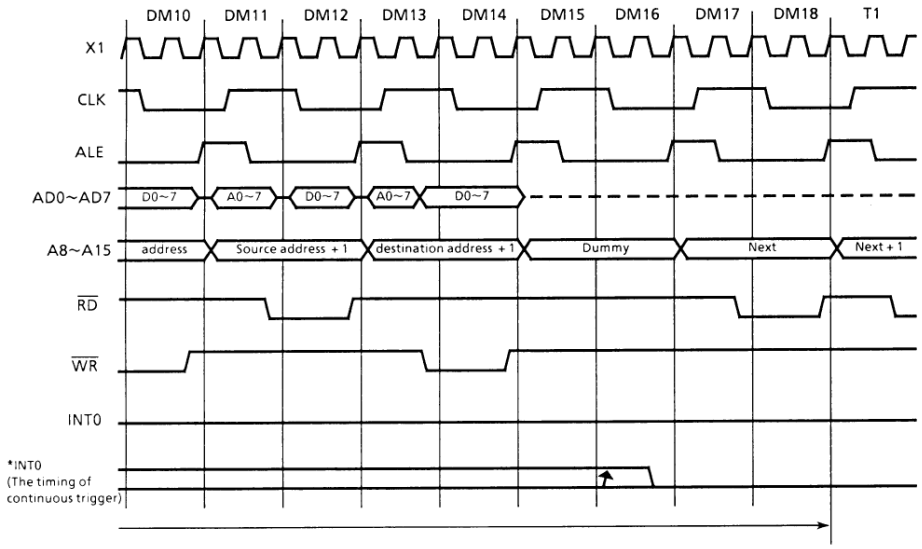
(6) HDMA Operation Timing

(a) 1-byte transfer mode



(b) 2-byte transfer mode





3.3.3 Interrupt Controller

Figure 3.3 (9) shows an abbreviated interrupt circuit diagram. The left half of this diagram shows the interrupt controller and the right half shows the CPU interrupt request signal circuit and hold release circuit.

The interrupt controller has an interrupt request flipflop and interrupt enable/disable flag for each interrupt channel (total: 23 channels), and a micro DMA enable/disable flag. The interrupt request flipflop latches interrupt requests that arrive from the periphery. This flipflop is reset to "0" when there is a reset, when the CPU receives an interrupt and reads the vector of that interrupt channel, and when an instruction that clears the interrupt request (writes "vector value/8" to memory address FFE0H) for that channel is executed.

LD (0FFE0H), 60H/8

For example, when LD (0FFE0H), 60H/8 is executed, the interrupt request flipflop for the interrupt channel [INTT1] with the vector value 38H is reset to "0" (to clear the flipflop, also write to address FFC9H when the interrupt request flag is assigned to FFE1H and FFE2H).

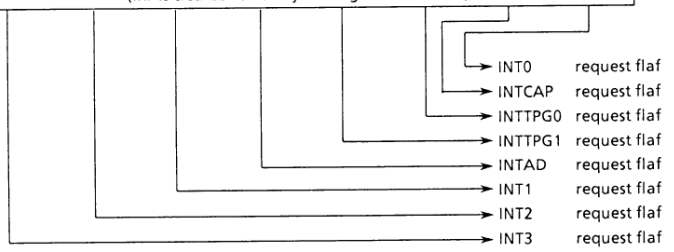
Table 3.3 (5) shows the "interrupt vector value/8" values. The status of the interrupt request flipflop can be determined by reading memory address FFC9H, FFCAH or FFCBH. "0" means no interrupt request and "1" means an interrupt request. Figure 3.3 (8) shows the bit layout when the interrupt request flipflop is read.

Table 3.3 (4) Interrupt Vector Value/8 Values

Priority sequence	Type	Interrupt function name	Interrupt vector value	Vector value ÷ 8
1	Non Maskable	SWI Instruction	10H	
2		NMI	18H	
3		INTWD	20H	
4	Maskable	INT0	28H	05H
5		INTCAP	30H	06H
6		INTTPG0	38H	07H
7		INTTPG1	40H	08H
8		INTRX0	48H	09H
9		INTTX0	50H	0AH
10		INTT0	58H	0BH
11		INTT1	60H	0CH
12		INTT2	68H	0DH
13		INTAD	70H	0EH
14		INTT3	78H	0FH
15		INTT4	80H	10H
16		INT1	88H	11H
17		INTT5	90H	12H
18		INT2	98H	13H
19		INTRX1	A0H	14H
20		INTTX1	A8H	15H
21		INT3	B0H	16H
22		INTRX2	B8H	17H
23		INTTX2	C0H	18H

IRF0 (FFE0H)

	7	6	5	4	3	2	1	0
bit Symbol	IRF3	IRF2	IRF1	IRFAD	IRFTPG1	IRFTPG0	IRFCAP	IRF0
Read/Write	R (Only IRF Clear Code can be written)							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt Request Flag 1: Interrupt requested (IRF is cleared to "0" by writing IRF clear code)							



Note : The specified interrupt request flipflop is cleared by writing 「vector value/8」 to memory address FFE0H.

Figure 3.3 (5). Interrupt Request Flipflop Read (1/2)

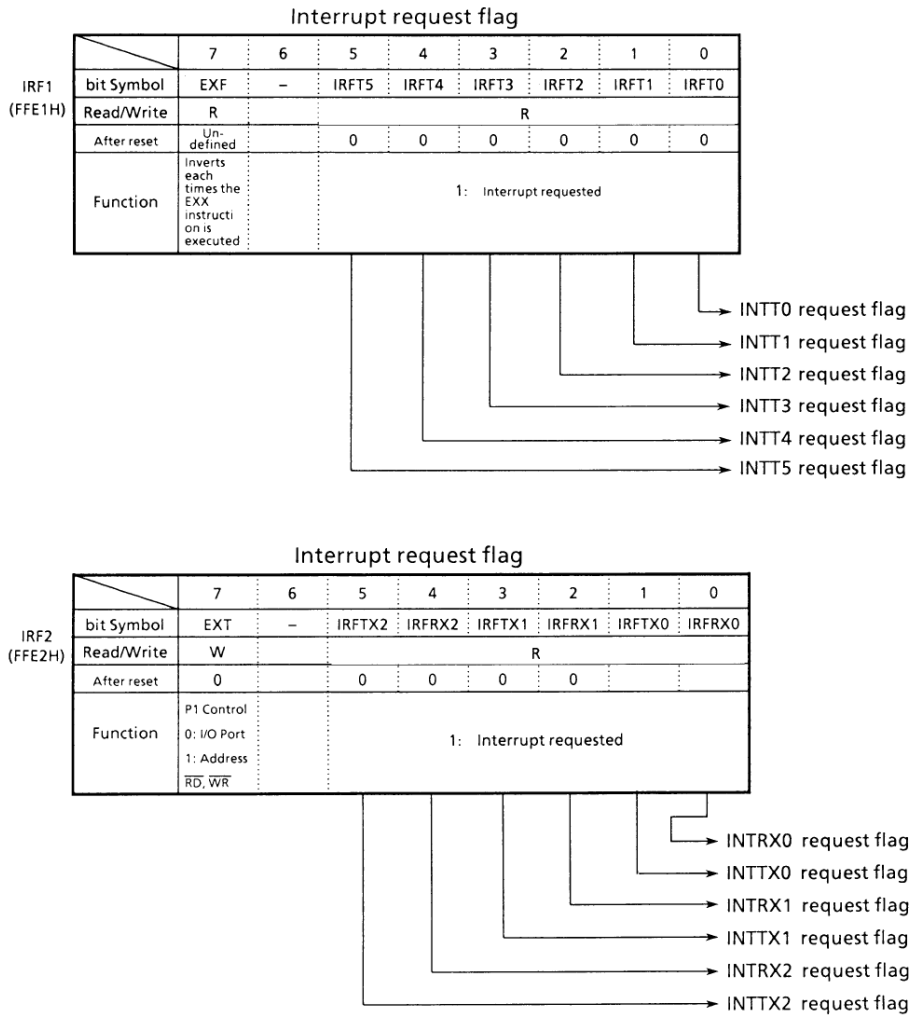


Figure 3.3 (6). Interrupt Request Flipflop Read (2/2)

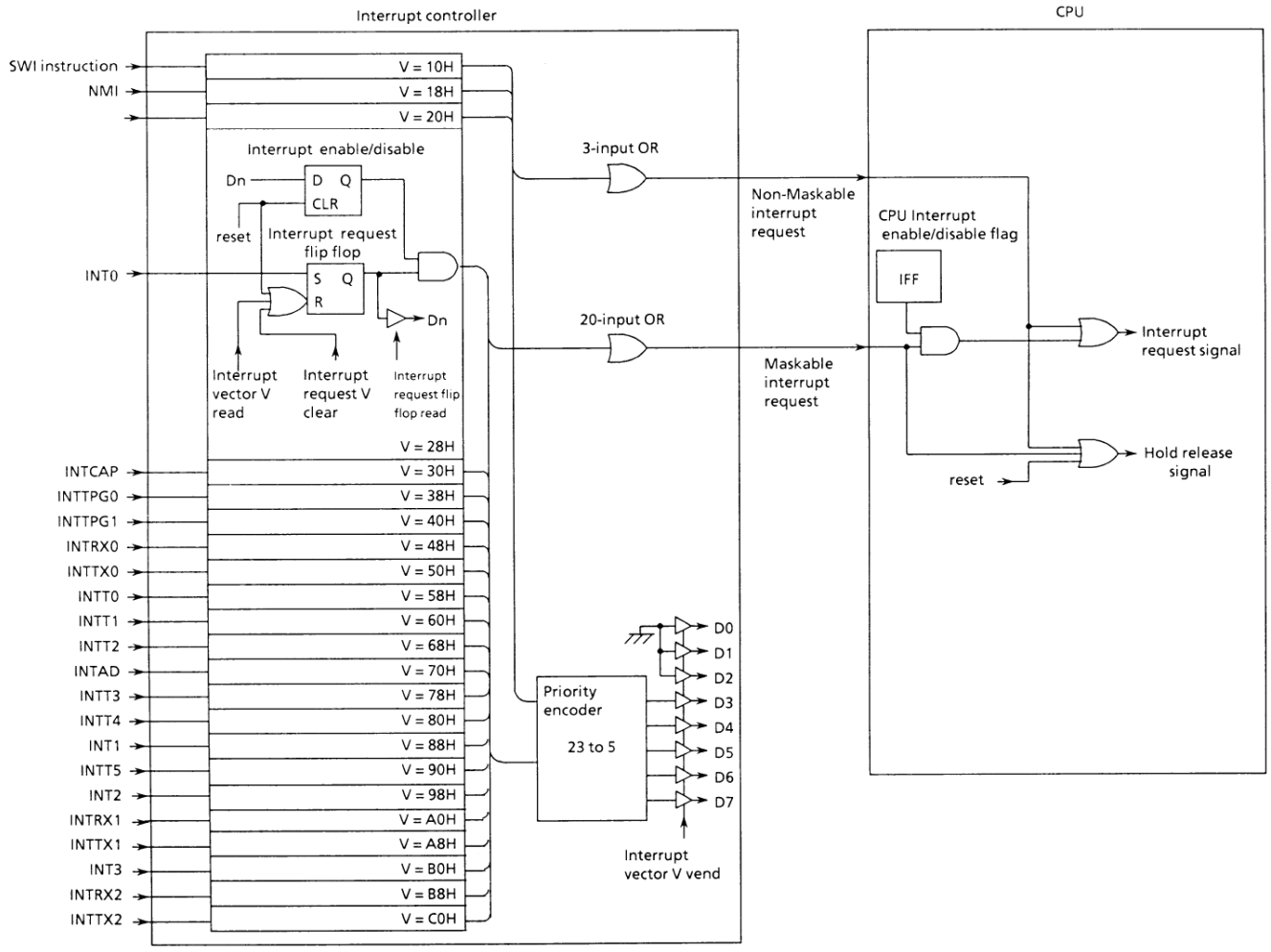


Figure 3.3 (7). Interrupt Controller Block

The interrupt enable/disable flags for each interrupt request channel are assigned to memory addresses FFE3H - FFE5H. Interrupts are enabled for a channel by setting the flag to “1”. The flags are reset to “0” by reseting.

Interrupt	Common Terminal	Mode	How to set
INT0	P81	 Level	INTE2 <EDGE> = 0
		 Rise edge	INTE2 <EDGE> = 1
INT1	P53	 Rise edge	T4MOD <CAPM1, 0> = 0, 0 or 0, 1 or 1, 1
		 Fall edge	T4MOD <CAPM1, 0> = 1, 0
INT2	P54	 Rise edge	–
INT2	P56	 Rise edge	–

For the pulse width for external interrupt, refer to “4.7 Interrupt Operation”.

Be careful that the following five are exceptional circuits.

INT0 Level mode	As the INT0 is not an edge type interrupt, the interrupt request flipflop is cancelled, and thus an interrupt request from peripheral devices passes through S input of the flipflop to become 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 SET 6, (0FFE5H): Switch the mode from level to edge LD (0FFE0H), FEH: Clear interrupt request flag EI
INTRX1, INTRX2	The interrupt request flipflop cannot be cleared only by reset operation or reading the serial channel receiving buffer, and cannot be cleared by an instruction.

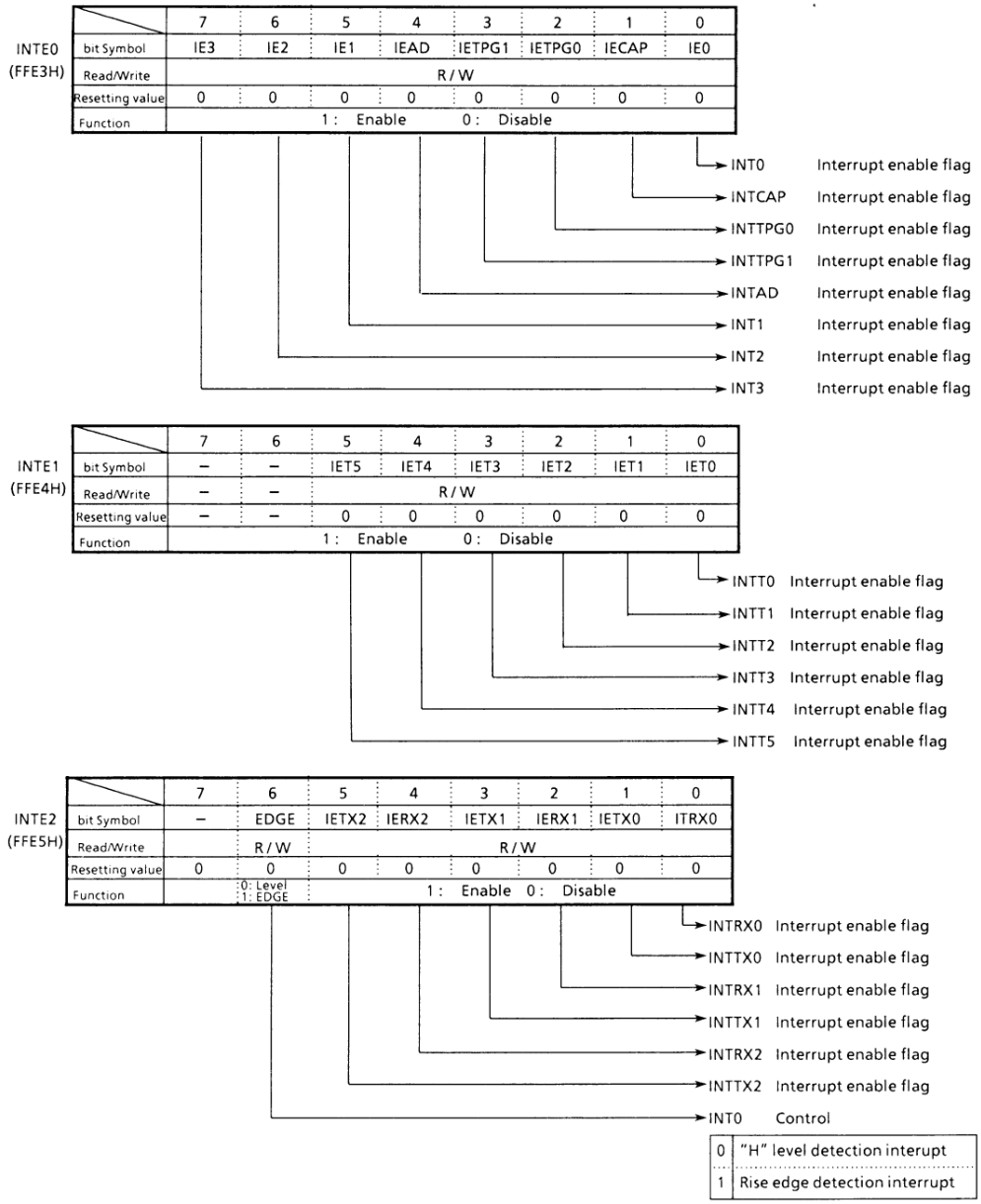


Figure 3.3 (8). Interrupt Enable Flags

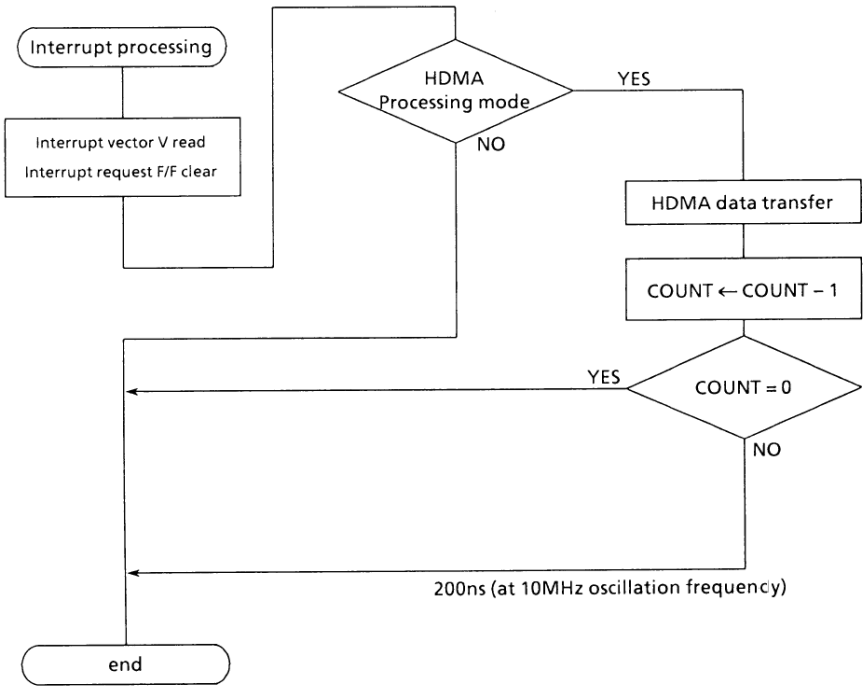


Figure 3.3 (9). Interrupt Processing Flow Chart

3.4 Standby Functions

When a HALT instruction is executed, TMP90CM36 enters the RUN, IDLE1 or STOP mode according to the contents of the halt mode setting register. The features are shown below.

(1) Run: Only the CPU halts, power consumption remains unchanged.

(2) IDLE: Only the internal oscillators operate, while all other internal circuits halt. Power consumption is 1/10 or less than that during normal operation.

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

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

These HALT states can be released by requesting an interrupt or resetting. Table 3.4 (2) shows how to release the HALT state. If the CPU is in the EI state for non-maskable or maskable interrupt, the interrupt will be acknowledged by the CPU and the CPU starts interrupt processing. If the CPU is in the DI state fro maskable interrupt, the CPU starts the execution from the instruction following HALT instruction, but the interrupt request flag remains at "1".

Even when HALT state is released by reset operation, the state (including the built-in RAM) just before entering the HALT can be retained. However, if HALT instruction has already been executed in the built-in RAM, the RAM contents may not be retained.

WDMOD (FFD2H)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRIVE
Read/Write	R/W	R/W		R/W	R/W		R/W	R/W
After reset	1	0	0	0	0	0	1	0
Function	1: WDT Enable	00 : 2^{16}/fc 01 : 2^{18}/fc 10 : 2^{20}/fc 11 : 2^{22}/fc	WDT Detecting time	Warming up time 0 : 2^{14}/fc 1 : 2^{16}/fc	Standby mode 00 : RUN mode 01 : STOP mode 10 : IDLE1 mode 11 : Reserved		Watchdog time out Control 1: P81 is set as the WDTOUT, pin and is connected internally to the RESER pin.	1 : Drives the pin even in STOP mode

Explained in 3.11 "Watchdog Timer"

Explained in 3.4.2 "STOP mode"

Figure 3.4 (1). HALT Mode Setting Register

3.4.1 RUN Mode

Figure 3.4 (2) shows the timing for releasing the HALT state by an interrupt during RUN mode. In the RUN mode, the system clock inside MCU does not stop even after HALT instruction has been executed; the CPU merely stops executing instructions. Accordingly, the CPU repeats dummy cycle until HALT state, interrupt requests are sampled at the fall edge of CLK signal.

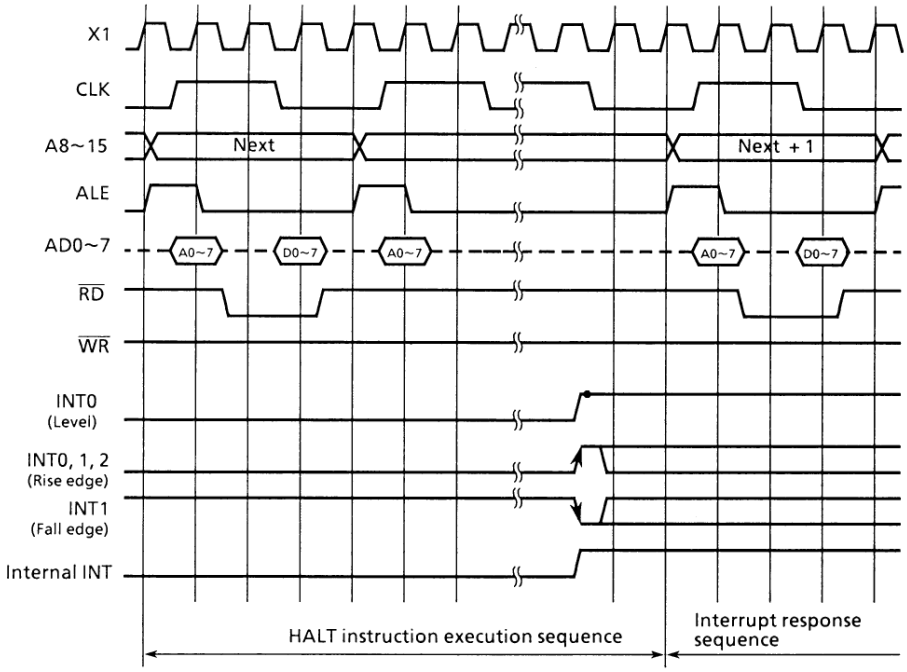


Figure 3.4 (2). HALT Release Timing Using Interrupts in RUN Mode

3.4.2 IDLE1 Mode

Figure 3.4 (3) shows the timing used for releasing the HALT mode by interrupts in the IDLE1 mode.

In the IDLE1 mode, only the internal oscillator operates, the system clock inside MCU stops and CLK signal is fixed to "1".

In the HALT state, interrupt requests are sampled asynchronously with the system clock but sampling is performed synchronously with the system clock, whereas the HALT release (restart of operation) is performed synchronously with it.

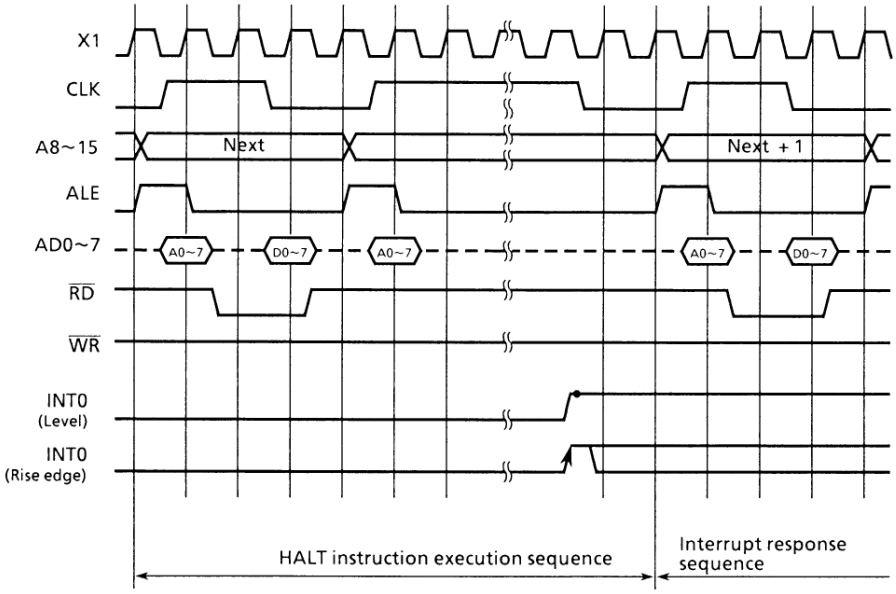


Figure 3.4 (3). HALT Release Timing Using Interrupts in the IDLE1 Mode

3.4.3 STOP Mode

Figure 3.4 (4) shows the timing of HALT release caused by interrupts in STOP mode.

In the STOP mode, all interval circuits stop, including internal oscillator. When the STOP mode is activated, all pins except special ones are put in the high-impedance state, isolated from the internal operation of MCU. Table 3.4 (1) shows the state of each pin in the STOP mode. However, if WDMOD <DRVE> (drive enable: bit 0 of memory address FFECH) of th built-in I/O register is set to "1", the pre-halt state of the pins can be retained. The register is cleared to "0" by reset operation.

When the CPU accepts an interrupt request, the internal oscillator first restarts. However, to get the stabilized oscillation, the system clock starts its output after the time set by warming up counter has passed. WDMOD <WARM> (warming up: bit 4 at memory address FFECH) is used to set up the warming up time. Warming up is executed for 2^{14} clock oscillation time when this bit is set to "0", while 2^{16} clock oscillation time when set to "1". This bit is cleared to "0" by reset operation.

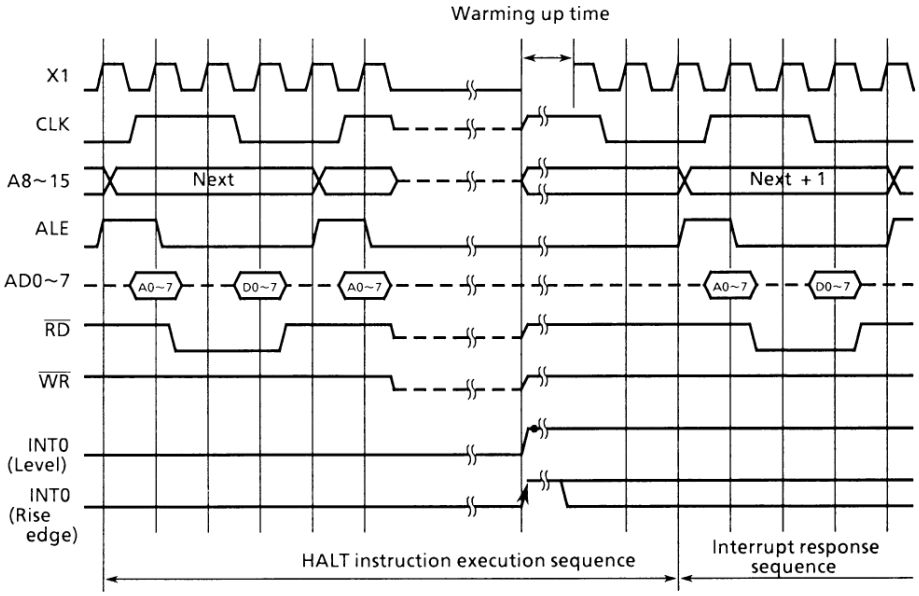


Figure 3.4 (4). HALT Release Timing Using Interrupts in STOP Mode

The internal oscillator can also be restarted by inputting the $\overline{\text{RESET}}$ signal "0" to the CPU.

However, the warming up counter remains inactive in order to make the CPU rapidly operate when the power is turned on. Accordingly, wrong operation may occur due to unstable clocks immediately after the internal oscillator has restarted. To release the HALT state by resetting in the STOP mode, $\overline{\text{RESET}}$ signal must be kept at "0" for a sufficient period of time.

Table 3.4 (1) State of Pins in STOP Mode

	I/O	DRVE = 0	DRVE = 1
P00 ~ P07	Input Mode Output Mode	– –	Input Output
P10 ~ P17	Input Mode Output Mode	–	Input Output
P20 ~ P27	Input Mode Output Mode	–	Input Output
P30 ~ P31	Input Mode Output Mode	–	Input Output
P32 ~ P33	Input Mode Output Mode	–	Input Output
P40 ~ P47	Input Mode Output Mode	–	Input Output
P56 ~ P57	Input Mode Output Mode	–	Input Output
P60 ~ P67	Input Mode	–	–
P70 ~ P77	Input Mode Output Mode	– Output	Input Output
$\overline{NMI}$ DAOUT0 DAOUT1 $\overline{NMI}$ CLK X1 $\underline{X2}$ $\overline{EA}$	Input Mode Output Mode Output Mode Input Mode Output Mode Input Mode Output Mode Input Mode	Input 0V 0V Input – – “1” Input	Input 0V 0V Input – – “1” Input
P80 ($\overline{WDTOUT}$) P81 ($\underline{INT0}$) P82 ($\overline{STBY}$) P83 (ALE)	Output Mode Input Mode Input Mode Output Mode	– Input Input –	Output Input Input Output

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

Input: Input is enabled.

Input: The input gate is operating. Fix the input voltage at either “0” or “1” to prevent the pin floating.

Output: Output status.

Table 3.4 (2) I/O Operation Release in HALT Mode

HALT mode			RUN	IDLE1	STOP
WDMOD<HALTM1, 0>			00	10	01
Operating block	CPU		Halt		
	I/O port		Retains the state when HALT instruction is executed.		See Table 3.4 (1)
	8-bit timer		Operation	Halt	
	16-bit timer				
	Time Base Counter				
	Capture circuit				
	Timming pulse generation				
	Watchdog timer				
	PWM Output				
	Serial interface				
	A/D converter				
	D/A converter				
	Interrupt controller				
HALT releasing source	Interrupt	NMI	○	○	○
		INTWD	○	—	—
		INT0	○	○	○
		INTCAP	○	—	—
		INTTPG0	○	—	—
		INTTPG1	○	—	—
		INTRX0	○	—	—
		INTTX0	○	—	—
		INTT0	○	—	—
		INTT1	○	—	—
		INTT2	○	—	—
		INTAD	○	—	—
		INTT3	○	—	—
		INTT4	○	—	—
		INT1	○	—	—
		INTT5	○	—	—
		INT2	○	—	—
		INTRX1	○	—	—
		INTTX1	○	—	—
		INT3	○	—	—
		INTRX2	○	—	—
		INTTX2	○	—	—
		INTTX2	○	—	—
		INTTBC	○	—	—
Reset			○	○	○

○ : Can be used for HALT release
— : Cannot be used for HALT release

3.5 Function of Ports

The TMP90CM36 contains total of 64 I/O port pins. These port pins function not only as the general-purpose I/O ports but also as the I/O ports for the internal CPU and built-in I/O. Table 3.5 shows the functions of these port pins.

Table 3.5 Functions of Ports

name	Pin name	NO. of pins	Direction	Direction Setting unit	Direction setting unit	Pin mane for internal function
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~P23	4	I/O	Bit	Input	TPG00~07 (TPG14~17)
Port 3	P30	1	Output		Output	$\overline{RD}$
	P31	1	Output		Output	$\overline{WR}$
	P32, P33	2	I/O		Input	PWM0, PWM1
Port 4	P40~P43	4	I/O	Bit	Input	TPG10~13
	P44~P47	4	I/O	Bit	Input	CAP0~CAP3
Port 5	P50	1	I/O		Input	EXIN
	P51	1	I/O		Input	TO4
	P52	1	I/O		Input	TO5
	P53	1	I/O		Input	INT1/TI4
	P54	1	I/O		Input	INT2/TI5
	P55	1	I/O		Input	TO1
	P56	1	I/O		Input	INT3/TI2
	P57	1	I/O		Input	TO3
Port 6	P60~P67	8	Input	Bit	Input	AN0~AN7
Port 7	P70	1	I/O		Input	RXD0
	P71	1	I/O		Input	TXD0
	P72	1	I/O		Input	RXD1
	P73	1	I/O		Input	SCLK1/$\overline{CTS1}$
	P74	1	I/O		Input	TXD1
	P75	1	I/O		Input	RXD2
	P76	1	I/O		Input	SCLK2
	P77	1	I/O		Input	TXD2
Port 8	P80	1	Output		Output	$\overline{WDTOUT}$
	P81	1	Input		Input	INT0
	P82	1	Input		Input	$\overline{STBY}$
	P83	1	Output		Output	ALE

These port pins function as the general-purpose I/O port pins by resetting. The port pins, for which input or output is programmably selectable, function as input ports by resetting. A separate program is required to use them for an internal function.

3.5.1 Port 0 (P00 ~ P07)

Port 0 is the 8-bit general-purpose I/O port P0, each bit of which can be set independently for input or output. The control register P0CR is used to set input or output. Reset operations clear all output latch and control register bits to “0” and set port 0 to the input mode.

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.

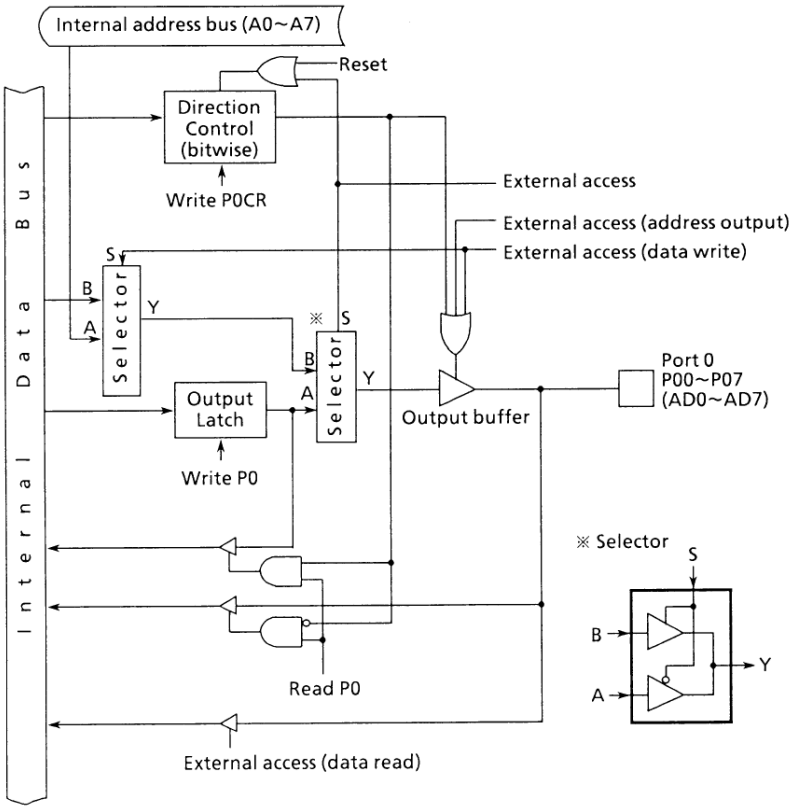


Figure 3.5 (1). Port 0 (P00 ~ P07)

Port 0 register

P0 (FFA0H)

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

Port 0 control register

P0CR (FFA1H)
Read, modify, write are not possible.

	7	6	5	4	3	2	1	0
bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
Read/Write	W							
After reset	0							
Function	0: IN 1: OUT (Select I/O on bit basis)							

Port 0 I/O setting

0	Input
1	Output

Figure 3.5 (2). Registers for Port 0

3.5.2 Port 1 (P10 ~ P17)

Port 1 is the 8-bit general-purpose I/O port P1, each bit of which can be set to input or output. The port 1 control register P1CR is used to set input or output. Reset operations clear all output latch and the control register bits to “0” and sets all port 1 bits to the input mode.

In addition to the general-purpose I/O port function, port 1 also functions as an address bus (A8 ~ A15). This is specified by setting the external extended specification 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 extended specification register value.

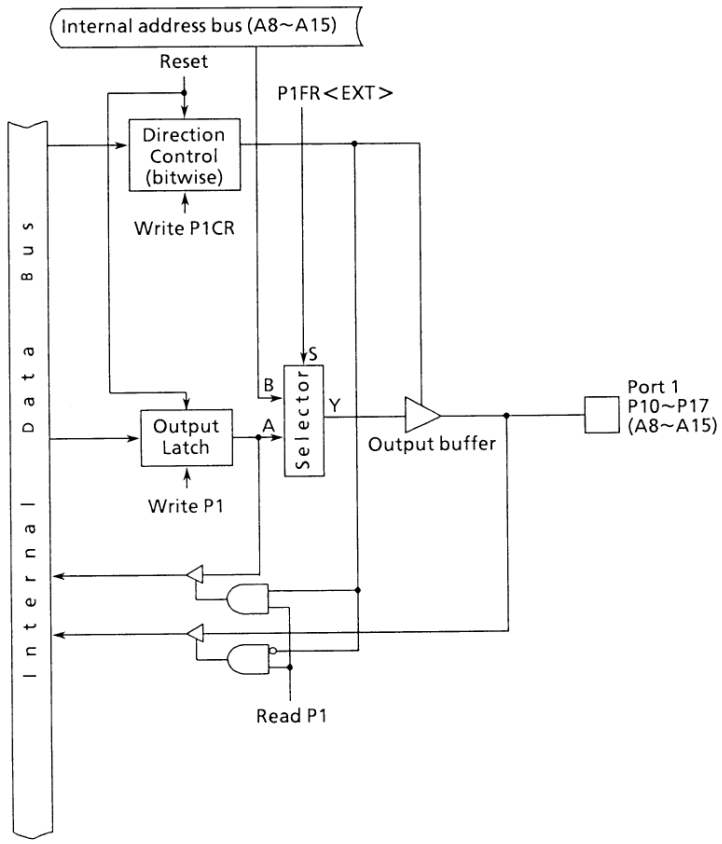


Figure 3.5 (3). Port 1 (P10 ~ P17)

Port 1 register

P1 (FFA2H)

	7	6	5	4	3	2	1	0
bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W							
After reset	Input mode							

Port 1 control register

P1CR (FFA3H)

	7	6	5	4	3	2	1	0
bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: IN 1: OUT (input or output can be specified for each bit)							

→ Port 1 I/O setting

0	Input
1	Output

IRF2 (FFE2H)

	7	6	5	4	3	2	1	0
bit Symbol	EXT	—	IRFTX2	IRFRX2	IRFTX1	IRFRX1	IRFTX0	IRFRX0
Read/Write	W			R				
After reset	0			0	0	0	0	0
Function	P1 control 0:I/O Port 1:Address			Interrupt Request Flag 1: Interrupt requested				

→ Explained under interrupt controller.

→ Port 1 general-purpose port/address bus settings

0	General-purpose port
1	Address bus

P1CR and IRFL <EXT> settings

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

Note: Each bit can be set independently.
P1CR <P1XC> is the X bit of P1CR.

Figure 3.5 (4). Registers for Port 1

3.5.3 Port 2 (P20 ~ P27)

Port 2 is the 8-bit general-purpose I/O port P2, each bit of which can be set to input or output. The port 2 control register P2CR is used to set input or output. Reset operations clear all output latch and the control register bits to “0” and set port 2 to the input mode.

In addition to the general-purpose I/O port function, port 2 also has a timing pulse generator (TPG) output (TPG00 ~ 07, TPG14 ~ TPG17). To use TPG as the output port function, port should be set to output mode, and and port output latch set to “0”. (After reset “0”) P24 ~ P27 is also used as TPG04 and TPG17 (P24), TPG05 and TPG16 (P25), TPG06 and TPG15 (P26), TPG07 and TPG14 (P27), it selects by the port 2 function register P2FR <P24TPG ~ P27TPG>. P27 is TPG07 output by setting P2 FR <P27M> “1”. TPG07 is <P27M> = “1” by resetting, and the port turns to output mode.

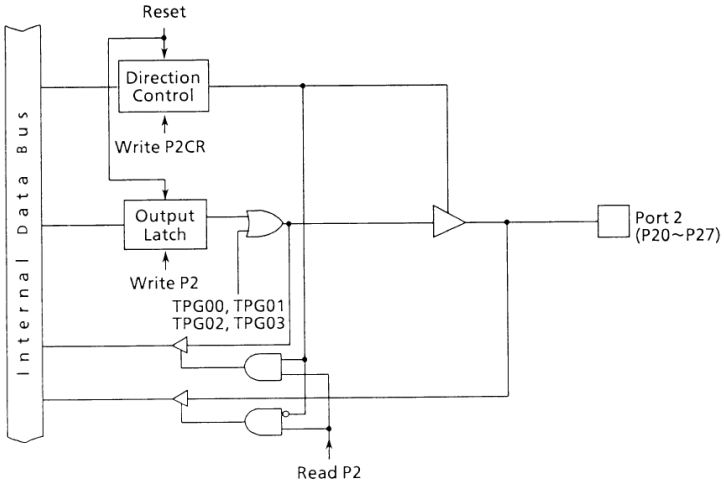


Figure 3.5 (5). Port 2 (P20 ~ P27)

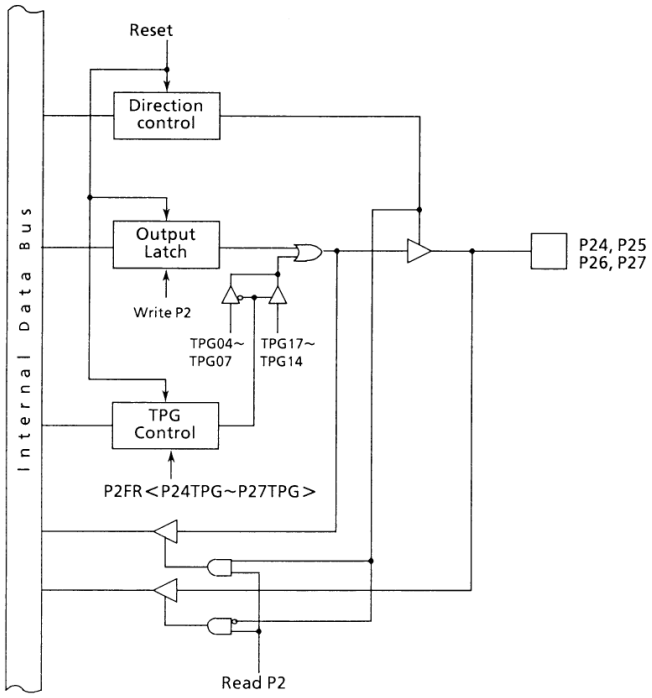


Figure 3.5 (6) Port 2 (P24, P25, P26, P27)

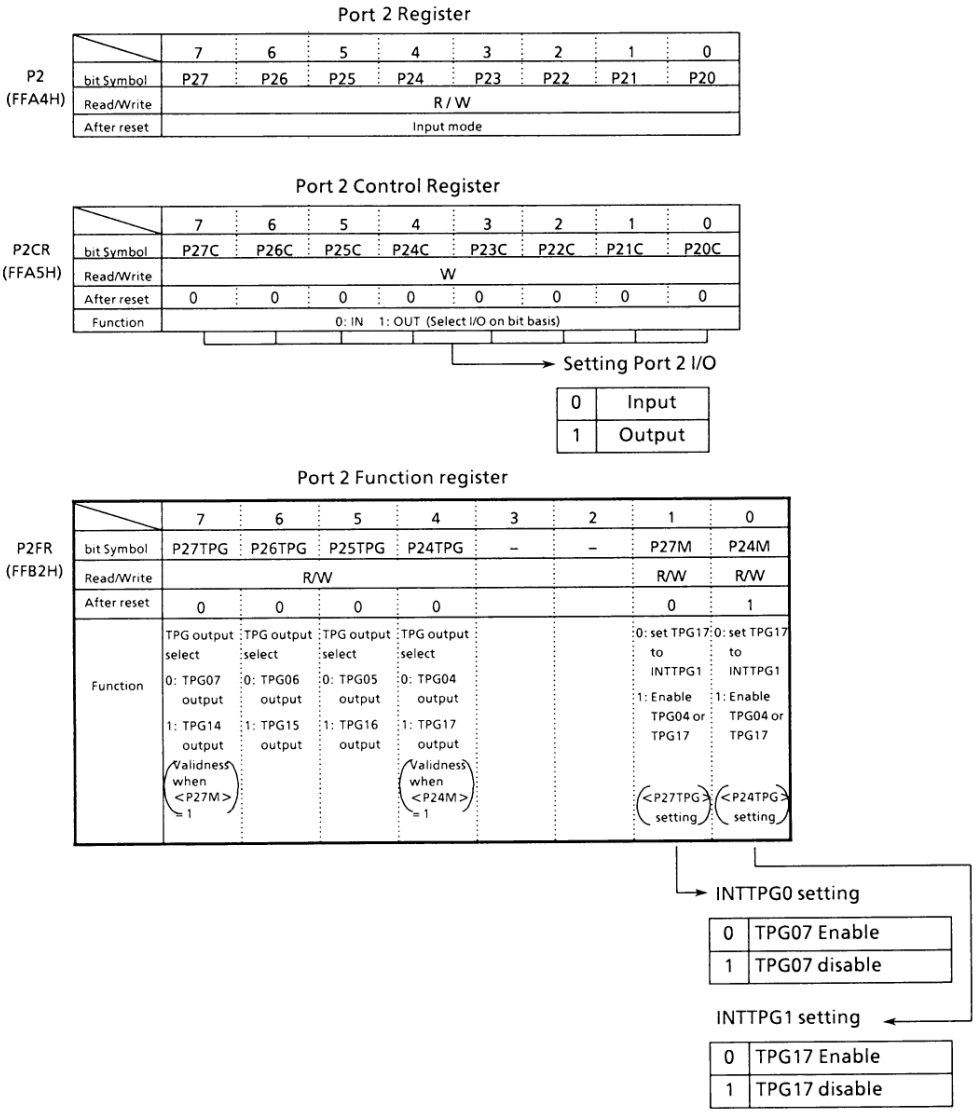


Figure 3.5 (7). Registers for Port 2

3.5.4 Port 3 (P30 ~ P33)

P32, P33 are a 4-bit general-purpose I/O port. The control register P38CR <P33C, P32C> is used for input or output.

P30, P31 are output ports. All bits of the output latch are set to "1" by resetting, and "1" is generated to the output port. Access of external memory makes P30, P31 function as the memory control pins ($\overline{RD}$ and $\overline{WR}$), when set IRF2 <EXT> to "1". When access of an internal memory makes them function, "1" is generated always.

Also function register P38CR <RDE> is intended for a pseudostatic RAM. When set IRF2 <EXT> to "1", and set P38CR <$\overline{RDE}$> to "1", it always functions as $\overline{RD}$ pin. Therefore, the $\overline{RD}$ pin outputs "0" (Enable) when it is an internal memory read and internal I/O read cycle.

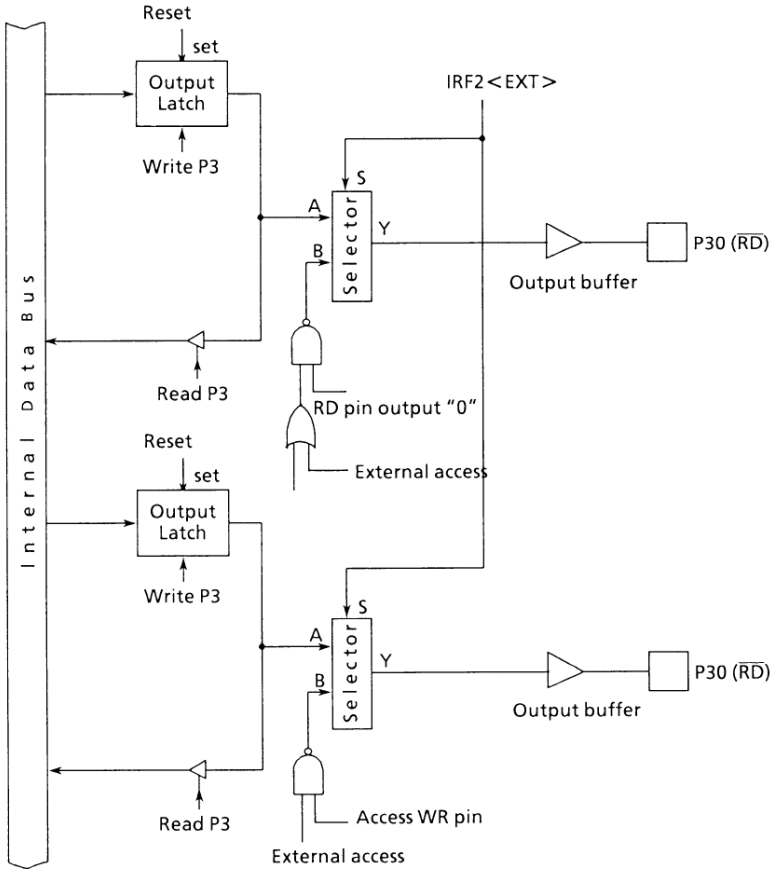


Figure 3.5 (8). Port 3 (P30, 31)

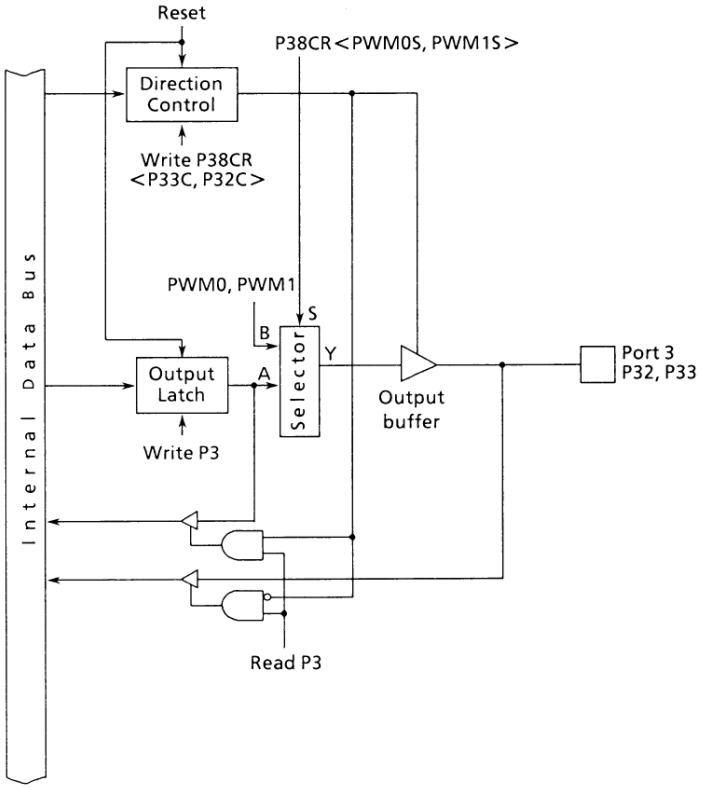


Figure 3.5 (9). Port 3

Port 3 Register

P3 (FFA6H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	P33	P32	P31	P30
Read/Write	R/W							
After reset					0	–	–	1

Port 38 Control register

P38CR (FFA7H)

	7	6	5	4	3	2	1	0
bit Symbol	ALEE	STBYS	PWM1S	PWM0S	P33C	P32C	WDTOUTC	RDE
Read/Write	R/W	W	R/W	R/W	W	W	W	R/W
After reset	(Note)	0	0	0	0	0	0	0
Function	ALE Control 0: Port 1: ALE	P82 Control 0: Port 1: $\overline{STBY}$	P33 Control 1: PWM1 Enable	P32 Control 1: PWM0 Enable	0: IN 1: OUT (Select I/O on bit basis)		P80 Control 0: Disable 1: Enable	RD Control 1: Always $\overline{RD}$ output (when <EXT> = 1)

Note : When $\overline{EA}$ pin is set to "1", port is output port after RESET, and <ALEE> is "0".
When $\overline{EA}$ pin is set to "0", port is always ALE output, <ALEE> is "1".

When IRF2<EXT> is "1", and P30 fixed to $\overline{RD}$ pin.

0	always "1" output
1	Fixed to $\overline{RD}$ pin

Select Port 32 PWM0 output

0	Input
1	Output

Select Port 32 PWM0 output

0	General-purpose port
1	PWM0

Select Port 33 PWM1 output

0	General-purpose port
1	PWM1

P38CR<P32C, P33C> and P3CR<PWM0S, PWM1S> setting

P3 <P32C, P33C> \ P3CR <PWM0S, 1S>	0	1
0	Input Port	Input Port
1	Output Port	PWM Output

Figure 3.5 (10). Register for Port 3

3.5.5 Port 4 (P40 - P47)

Port 4 is the 8-bit general-purpose I/O port, each bit of which can be set for input or output. The control register P4CR is used to set input or output.

All bits of the function register are cleared to “0” by resetting, and the port turns of general-purpose I/O port mode.

In addition to the general-purpose I/O port function, P40 ~ 43 have an input/output function for the Timing Pulse Generators output (TPG10 ~ TPG13) function, and P44 ~ P47 have the capture input (CAP0 ~ CAP3) function.

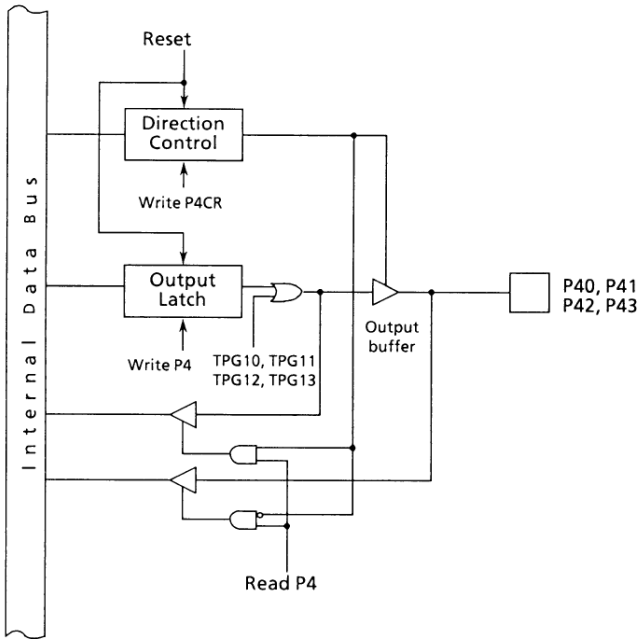


Figure 3.5 (11). Port 4 (P40, P41, P42, P43)

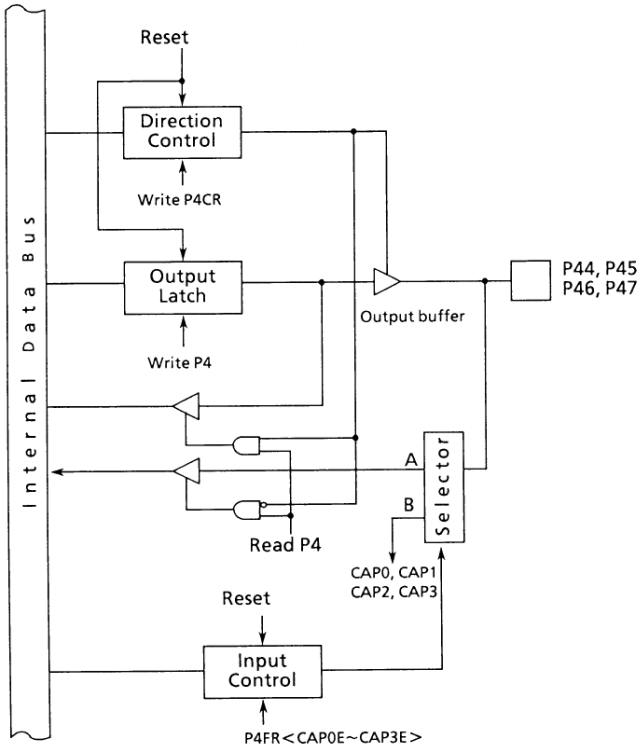


Figure 3.5 (12). Port 4 (P44, P45, P46, P47)

Port 4Register

P4 (FFA8H)

	7	6	5	4	3	2	1	0
bit Symbol	P47	P46	P45	P44	P43	P42	P41	P40
Read/Write	R/W							
After reset	Input mode							

Port 4 Control register

P4CR (FFA9H)

	7	6	5	4	3	2	1	0
bit Symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: IN 1: OUT (Select I/O on bit basis)							

Select Port 3 I/O

0	Input
1	Output

Port 4Function register

P4FR (FFB3H)

	7	6	5	4	3	2	1	0
	CAPE				CAPS			
bit Symbol	CAP3E	CAP2E	CAP1E	CAP0E	CAP3S	CAP2S	CAP1S	CAP0S
Read/Write	R/W				R/W			
After reset	0	0	0	0	0	0	0	0
Function	Capture input control 0: Disable 1: Enable				Capture edge degection control 0: ↑ rise 1: ↓ fall			

Figure 3.5 (13). Register for Port 4

3.5.6 Port 5 (P50 - P57)

Port 4 is the 8-bit general-purpose I/O port, each bit of which can be set for input or output. The control register P5CR is used to set input or output.

By reset operation, the output latch and the control register is reset to “0”, and port 5 is placed in the input mode.

In addition to the general-purpose I/O port function, these ports function as interrupt request input, clock input for timer or event counter, or timer output, or wait input, or clock input for time base counter.

(1) P55, P57, P51, P52

When specified by port 5 function register P5FR <TO1S ~ TO5S>, these ports become the timer output.

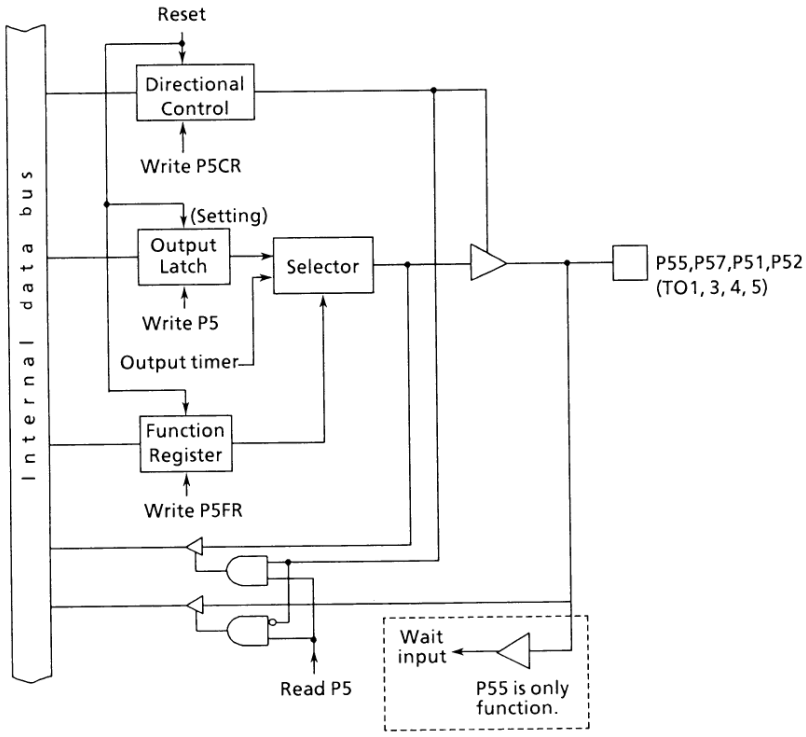


Figure 3.5 (14). Port 5 (P55, P57, P51, P52)

(2) P56 external interrupt request input (INT3).

P56 is also used as clock input (TI0) for 8-bit timer 0

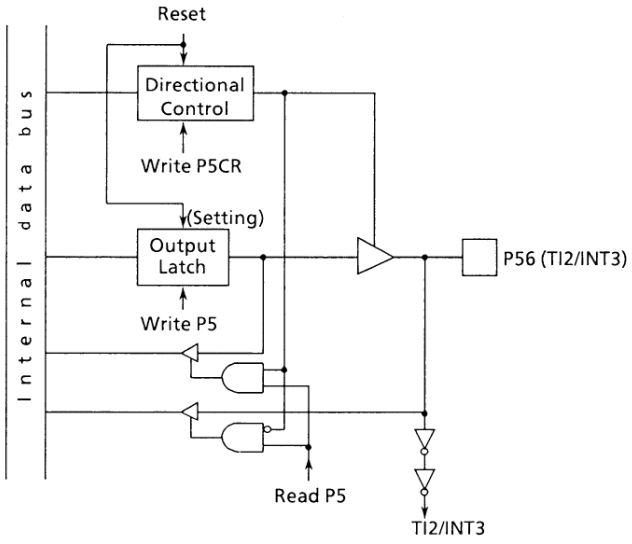


Figure 3.5 (15). Port 5 (P56)

(3) P53, P54

These ports are also used as the clock input for 16-bit timer or event counter as well as external interrupt request input.

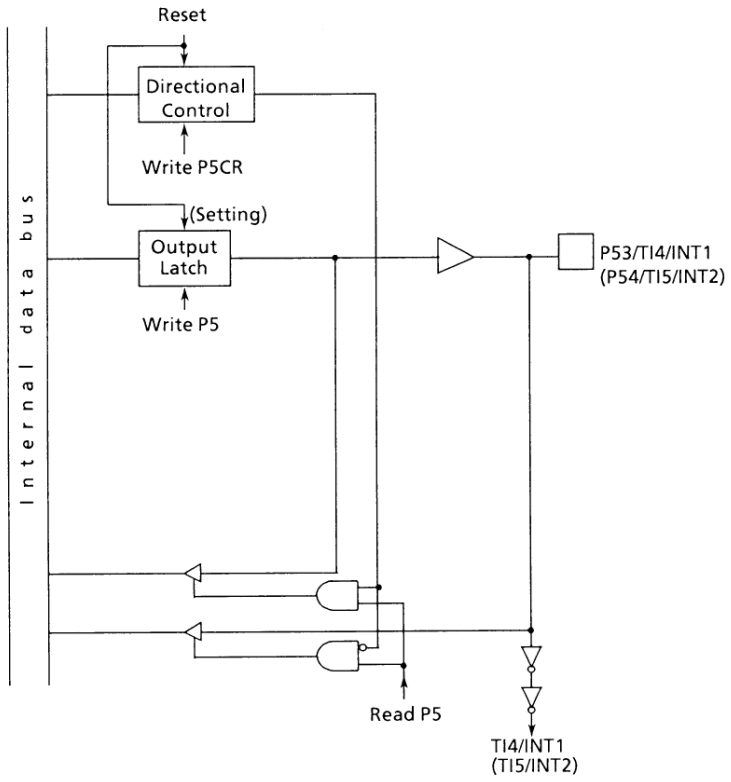


Figure 3.5 (16). Port 5 (P53, P54)

(4) P50

P50 is also used as input external input (EXIN) for clock input for time base counter. It selects by setting the port 5 function register P5FR <EXINE>.

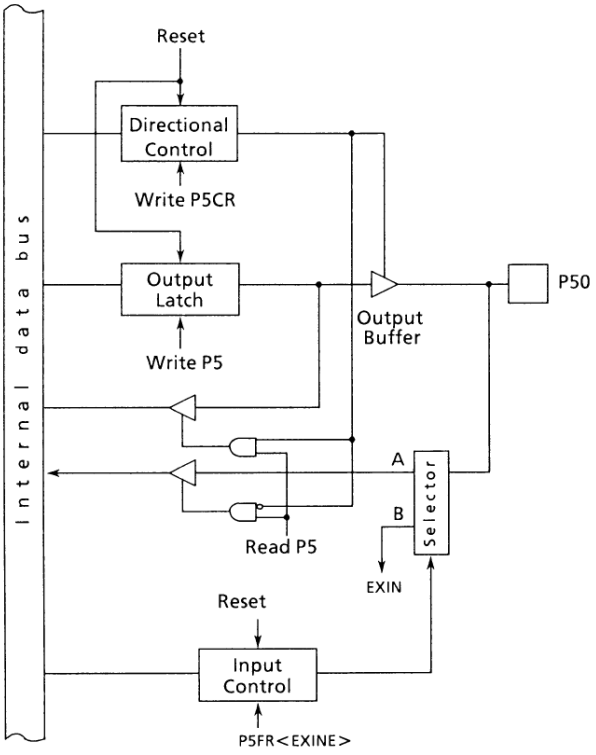


Figure 3.5 (17). Port (P50)

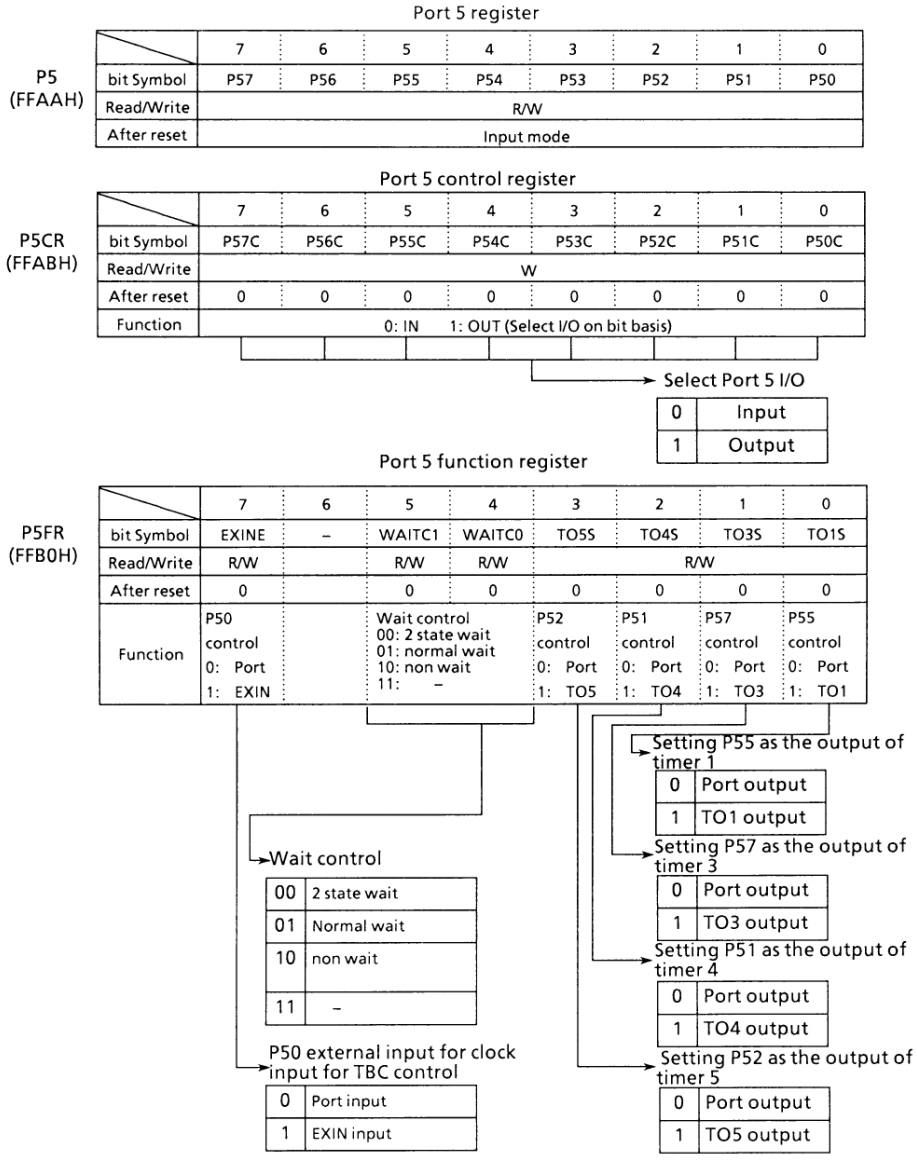


Figure 3.5 (18). Registers for Port 5

3.5.7 Port 6 (P60 - P67)

Port 6 is an 8-bit general-purpose input port with fixed input function.

In addition to its general-purpose input port function, these ports function as analog input pins (AN0 ~ AN7).

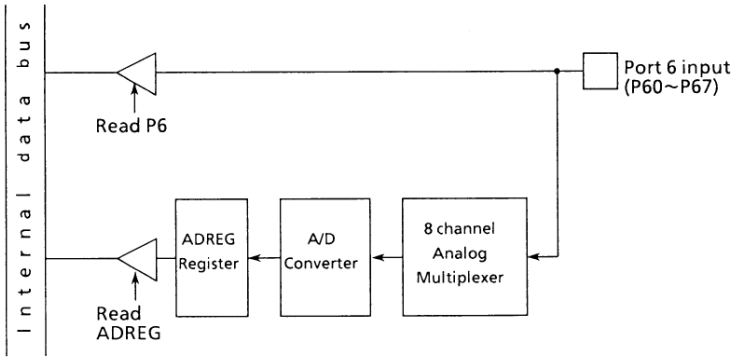


Figure 3.5 (19). Port 6 (P60 ~ P67)

Port 6 register

		7	6	5	4	3	2	1	0
P6 (FFACH)	bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
	Read/Write	R							
	After reset	Input-only							
	Function	Used also as analog input pins (AN0~7)							

Figure 3.5 (20). Registers for Port 6

3.5.8 Port 7 (P70 - P77)

Port 3 is the 8-bit general-purpose I/O port, each bit of which can be set for input or output. The control register P3CR is used to set input or output. By reset operations, all bits of the output latch are set to “0”, while all bits of control register are to “0”, and port 7 is placed in the input mode. In addition to the general-purpose I/O port function, port 7 function have an internal serial interface input/output function. This is specified by function register P23FR. All bits of the function register are cleared to “0” by resetting, and the port turns to general-purpose I/O mode.

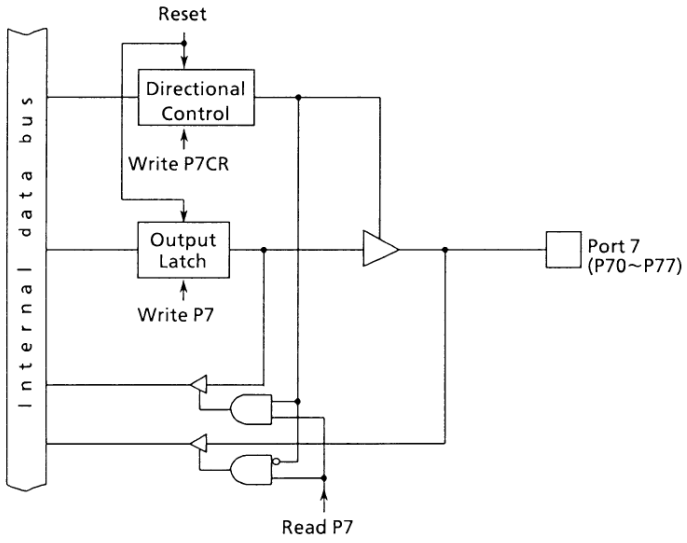


Figure 3.5 (21). Port 7

Port 7 register

		7	6	5	4	3	2	1	0
P7 (FFADH)	bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
	Read/Write	R/W							
	After reset	Input mode							

Port 7 control register

		7	6	5	4	3	2	1	0
P7CR (FFAEH)	bit Symbol	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: IN 1: OUT (Select I/O on bit basis)							

Select Port 7 I/O

0	Input
1	Output

Figure 3.5 (22). Registers for Port 7

Port 7 function register

P7FR (FFB1H)

	7	6	5	4	3	2	1	0
bit Symbol	ODE2	TXDC2	SCLKC2	ODE1	TXDC1	SCLKC1	ODE0	TXDC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	P77 0: CMOS 1: Open Drain	P77 control 0: Port 1: TxD output	P76 control 0: Port 1: SCLK output	P74 0: CMOS 1: Open Drain	P74 control 0: Port 1: TxD output	P73 control 0: Port 1: SCLK output	P71 0: CMOS 1: Open Drain	P71 control 0: Port 1: TxD output

Setting P71 as TxD0 output

0	Port output
1	TxD0 output

Setting P71 as open drain output

0	CMOS output
1	Open drain output

Setting P73 as SCLK1 output

0	Port output
1	SCLK1 output

Setting P74 as TxD output

0	Port output
1	TxD1 output

Setting P74 as open drain output

0	CMOS output
1	Open drain output

Setting P76 as SCLK2 output

0	Port output
1	SCLK2 output

Setting P77 TxD2 output

0	Port output
1	TxD2 output

Setting P77 as open drain output

0	CMOS output
1	Open drain output

Figure 3.5 (23). Registers for Port 7

3.5.9 Port 8 (P80 - P83)

Port 8 is the 4-bit general-purpose I/O port, P81, P82 are input-only ports. P80, P83 are output-only ports.

In addition to its general-purpose input port function, or watch dog timer out output, these port function as external interrupt request input, or hardware input, or ALE output.

(1) P81/INT0

P81 is the general-purpose input port, is also use as external interrupt request input (INT0). INT0 is to be used as “H” level detection interrupt or rise edge detection interrupt by control register INTE2 <EDGE>.

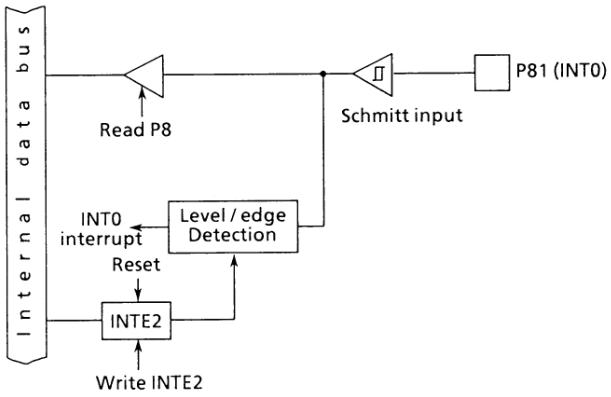


Figure 3.5 (24). Port 8 (P81)

(2) P80

P80 is used both as a general-purpose output port and for $\overline{\text{WDTOUT}}$ output. Bit 1 of the watchdog timer mode register (WDMOD: memory address FFDDH) and Bit 1 of the P38 control register (P38CR: memory address FFA7H) is used to set P80 for $\overline{\text{WDTOUT}}$ output.

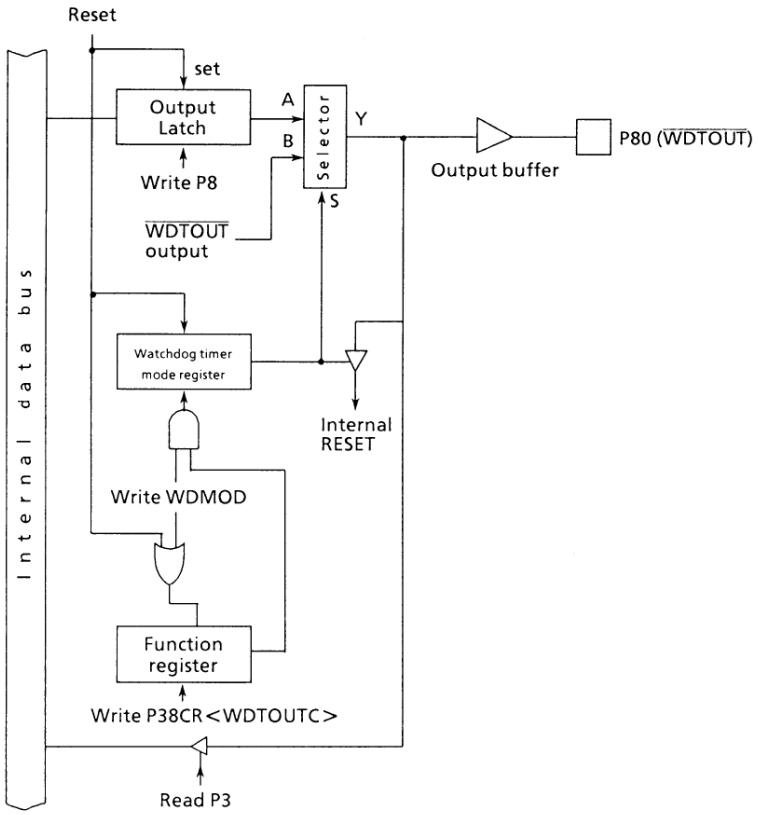


Figure 3.5 (25). Port 8 (P80)

(3) P82/$\overline{\text{STBY}}$

P82 is a general purpose input port, and this port can be used also as Hardware standby. By reset operations, the control register P38CR <STBY> is "0", and P82 is placed in the general-purpose input port. When the control register P38CR <STBYS> is "1", and P82 is placed in the hardware standby input pin.

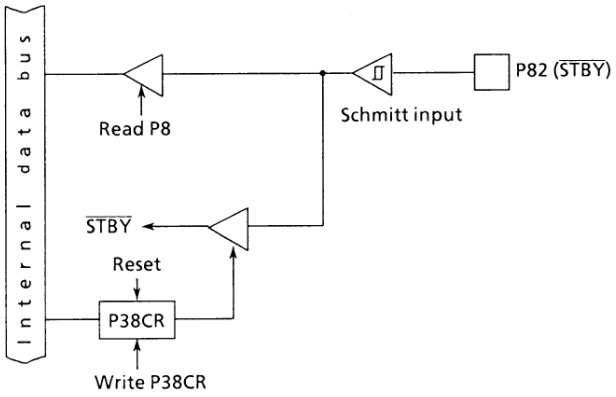


Figure 3.5 (26). Port 8 (P82)

(4) P83

P83 is output port, and is also used as ALE pin. When P83 was 1 chip mode ($\overline{EA}$ = 1), by reset operations, the control register P38CR <ALEE> is “0”, and P83 is placed in the output port. When ALE pin uses, and <ALEE> is set to “1”. When Multi chip mode was <ALEE> is always “1”, and P83 become the ALE output.

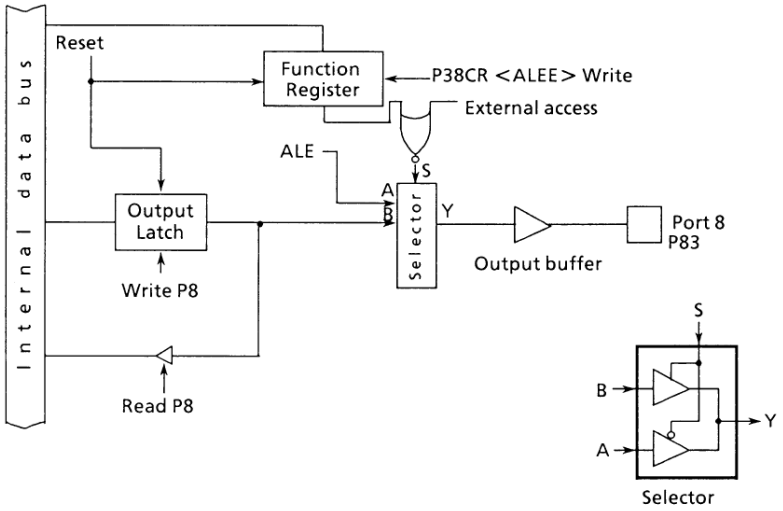


Figure 3.5 (27). Port 8 (P83)

Port 8 register

P8 (FFAFH)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	P83	P82	P81	P80
Read/Write	R/W							
After reset					0	–	–	1

Port 38 control register

P38CR (FFA7H)

	7	6	5	4	3	2	1	0
bit Symbol	ALEE	STBYS	PWM1S	PWM0S	P33C	P32C	WDTOUTC	RDE
Read/Write	R/W	W	R/W	R/W	W	W	W	R/W
After reset	(注)	0	0	0	0	0	0	0
Function	ALE control 0: Port 1: ALE	P82 control 0: Port 1: $\overline{STBY}$	P33 control 1: PWM1 Enable	P32 control 1: PWM0 Enable	0: IN 1: OUT (Select I/O on bit basis)		P80 control 0: Disable 1: Enable	RD control 1: Always $\overline{RD}$ output (When <EXT> is "1")

(Note) When $\overline{EA}$ pin is set to "1", port is output port after RESET, and <ALEE> is "0".

When EA pin is set to "0", port is always ALE output, <ALEE> is "1"

Figure 3.5 (28). Registers for Port 8

3.6 Timers

The TMP90CM36 contains four 8-bit timers (timers 0, 1, 2 and 3), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timers.

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

- 8-bit interval timer mode (4 timers)
- 16-bit interval timer mode (2 timers)
- 8-bit programmable square wave pulse generation (PPG: variable duty with variable cycle) output mode (2 timers)
- 8-bit pulse width modulation (PWM: variable duty with constant cycle) output mode (2 timers)

The upper two can be combined (two 8-bit timers and one 16-bit timer).

Figure 3.6 (1) shows the block diagram of 8-bit timer (timer 0 and timer 1).

8-bit timer (timer 2, 3) are connected to the external clock pin TI2 in the timer 2 up counter input clock.

Other timer 2 and timer 3 have the same circuit configuration as timer 0 and timer 1. Each interval consists of an 8-bit up-counter, 8-bit comparator, and 8-bit timer register. Besides, one timer flipflop (TFF1 or TFF3) is provided for each pair of timer 0 and timer 1 as well as timer 2 and timer 3.

Among the input clock sources for the interval timers, the internal clocks of øT1, øT4, øT16, and øT256 are obtained from the 9-bit prescaler shown in Figure 3.6 (2).

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

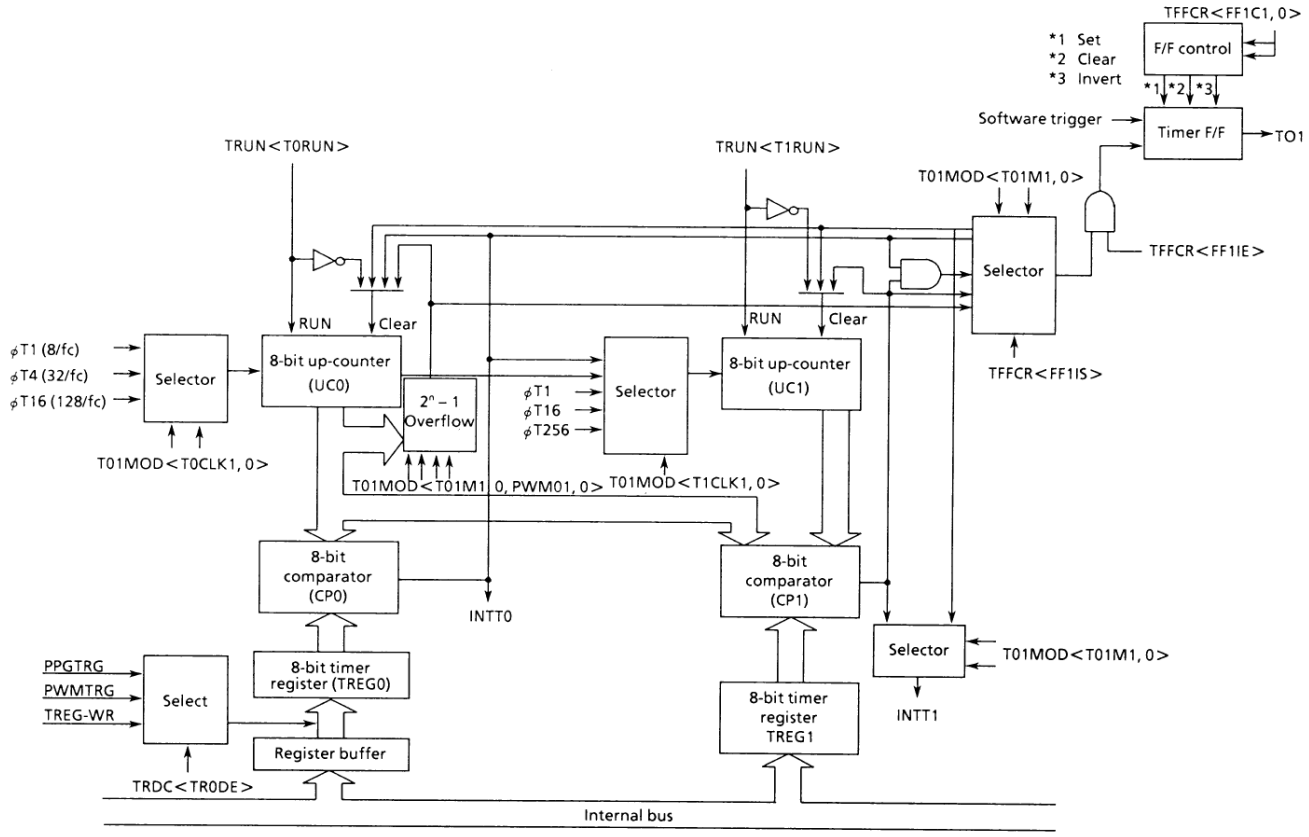


Figure 3.6 (1). Block Diagram of 8-bit Timers (Timers 0 and 1)

① Prescaler

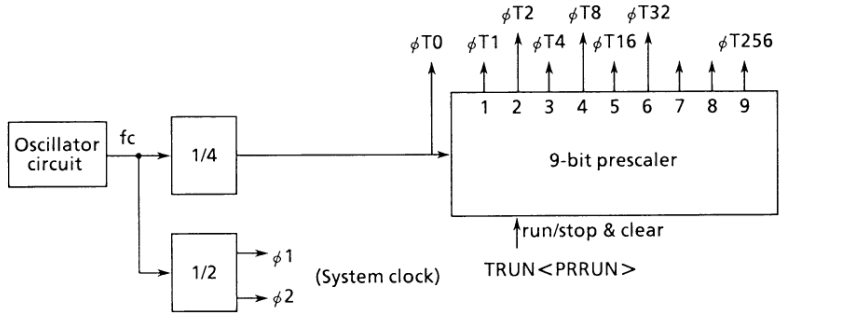
This 9-bit prescaler generates the clock input to the 8-bit timers, 16-bit timer/event counters, and baud rate generators by further dividing the fundamental clock (fc) after it has been divided by 4 (fc/4).

Among them, 8-bit timer uses 4 types of clock: øT1, øT16, and øT256.

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

Cycle

Input clock \ fc	12.5MHz	16MHz
ϕT1 (8/fc)	0.64μs	0.5μs
ϕT4 (32/fc)	2.56μs	2.0μs
ϕT16 (128/fc)	10.24μs	8.0μs
ϕT256 (2048/fc)	163.84μs	128μs



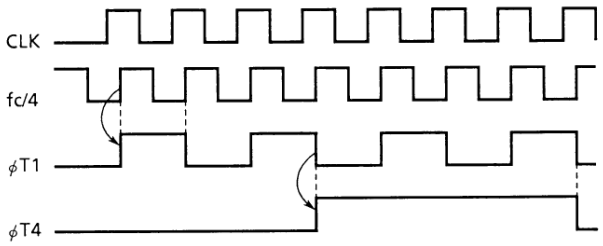


Figure 3.6 (2). Prescaler

② Up-counter

This is an 8-bit binary counter that counts up the input clock pulse specified by the timer 0/timer 1 mode register T01MOD and timer 2/timer 3 mode register T23MOD.

The input clock pulse for timer 0 is selected from øT1 (8/fc), øT4 (32/fc) and øT16 (128/fc). Timer 2 input clock is selected from external clock (TI2 pin = P55/INT3) and same the timer 0 in three kinds internal clock. According to the set value of T01MOD and T23MOD.

The input clock of timer 1 and timer 3 differs depending on the operating mode. When set to 16-bit timer mode, the overflow output of timer 0 and timer 2 is used as the input clock.

When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks øT1 (8/fc), øT16 (128/fc), and øT256 (2048/fc) as well as the comparator output (match detection signal) of timer 0 and timer 2, according to the set value of T01MOD and T23MOD.

Example: When TMOD <T01M1,0> = 01, the overflow output of timer 0 becomes the input clock of timer 1 (16-bit timer). When TMOD <T01M1,0> = 00 and T01MOD <T1CLK1,0> = 01, øT1 (8/fc) becomes the input of timer 1. Operation mode is also set by T01MOD and T23MOD. When reset, it is initialized to

T01MOD <T01M1, 0> = 00 and T23MOD <T23M1, 0> = 00, whereby the up-counter is placed in the 8-bit timer mode.

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

③ Timer registers

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

Timer registers TREG0 and TREG2 are of double buffer structure, each of which makes a pair with register buffer.

The TREG0 and TREG2 control whether the double buffer should be enabled or disabled through the timer register double buffer control register TRDC <TR0DE, TR2DE>. It is disabled when <TR0DE>/<TR2DE> = 0, and enabled when they are set to 1.

The timing to transfer data from the register buffer to the timer register in the double buffer enable state is the moment 2^n - 1 overflow occurs in PWM mode or the moment compare cycles will be equal in PPG mode.

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

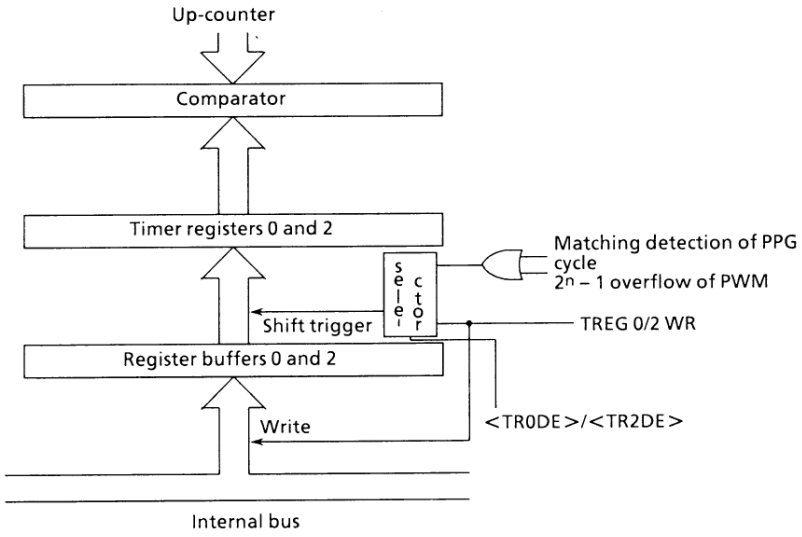


Figure 3.6 (3). Configuation of Timer Registers 0 and 2

Note: Timer register and the register buffer are allocated to the same memory address. When <TR0DE>/<TR2DE> = 0, the same value written in the register buffer as well as the timer register, while when <TR0DE>/<TR2DE> = 1 only the register buffer is written.

The memory address of each timer register is as follows.

TREG0: FFC6H
TREG1: FFC7H
TREG2: FFC8H
TREG3: FFC9H

All the registers are write-only and cannot be read.

T01MOD (FFCAH)

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

Input clock of timer 0

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

Input clock of timer 1

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

Select PWM0 cycle
("Don't care" except in PWM mode)

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

Set the operation mode of timer 0 and 1.

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

Figure 3.6 (4). Timer 0/Timer 1 Mode Register (T01MOD)

T23MOD (FFCBH)

	7	6	5	4	3	2	1	0
bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8bit Timer 01: 16bit Timer 10: 8bit PPG 11: 8bit PWM		00: – 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI2 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 2 input clock

00	External clock TI2
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Timer 3 input clock

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

Select PWM2 cycle
("Don't care" except in PWM mode)

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

Set the operation mode of timer 2 and timer 3

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

Figure 3.6 (5). Timer 2/Timer 3 Mode Register (T23MOD)

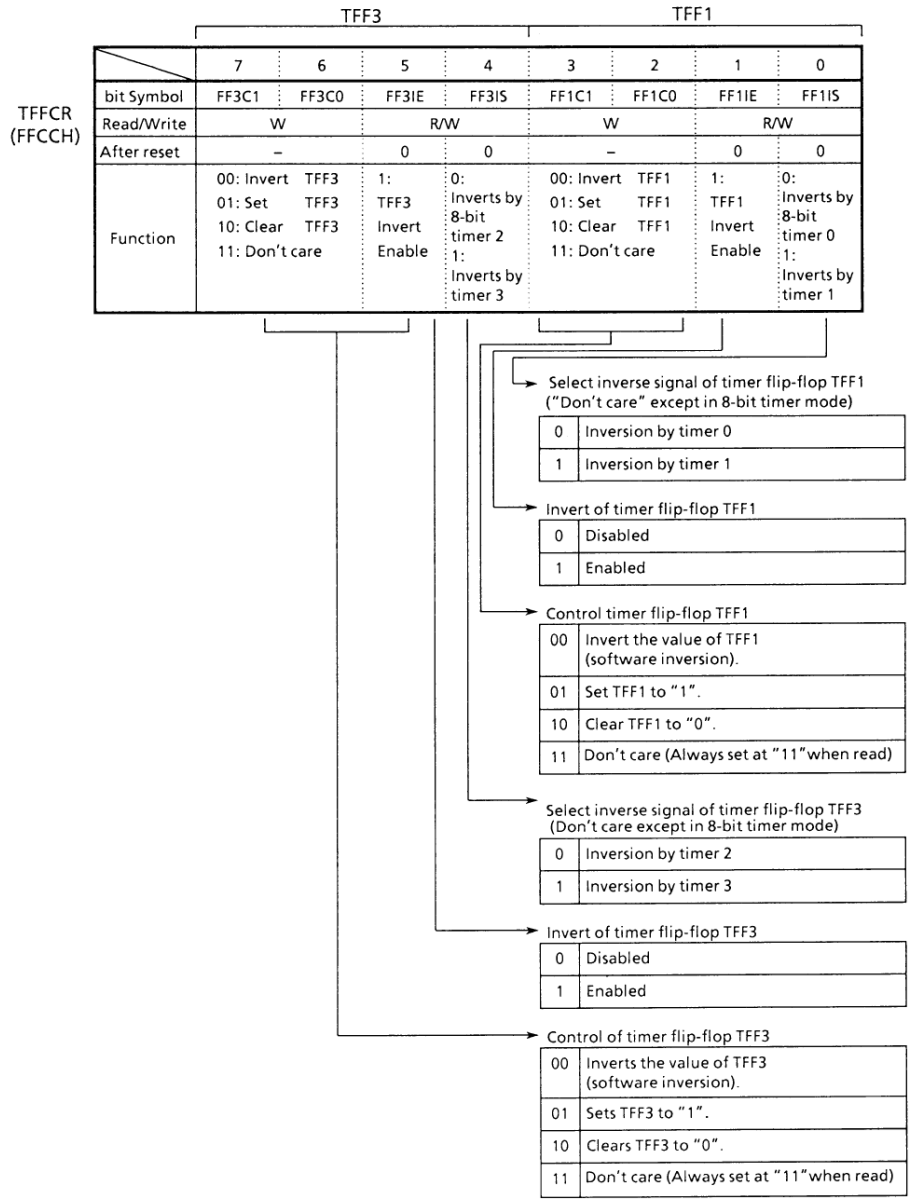


Figure 3.6 (6). 8-Bit Timer Flipflop Control Register (TFFCR)

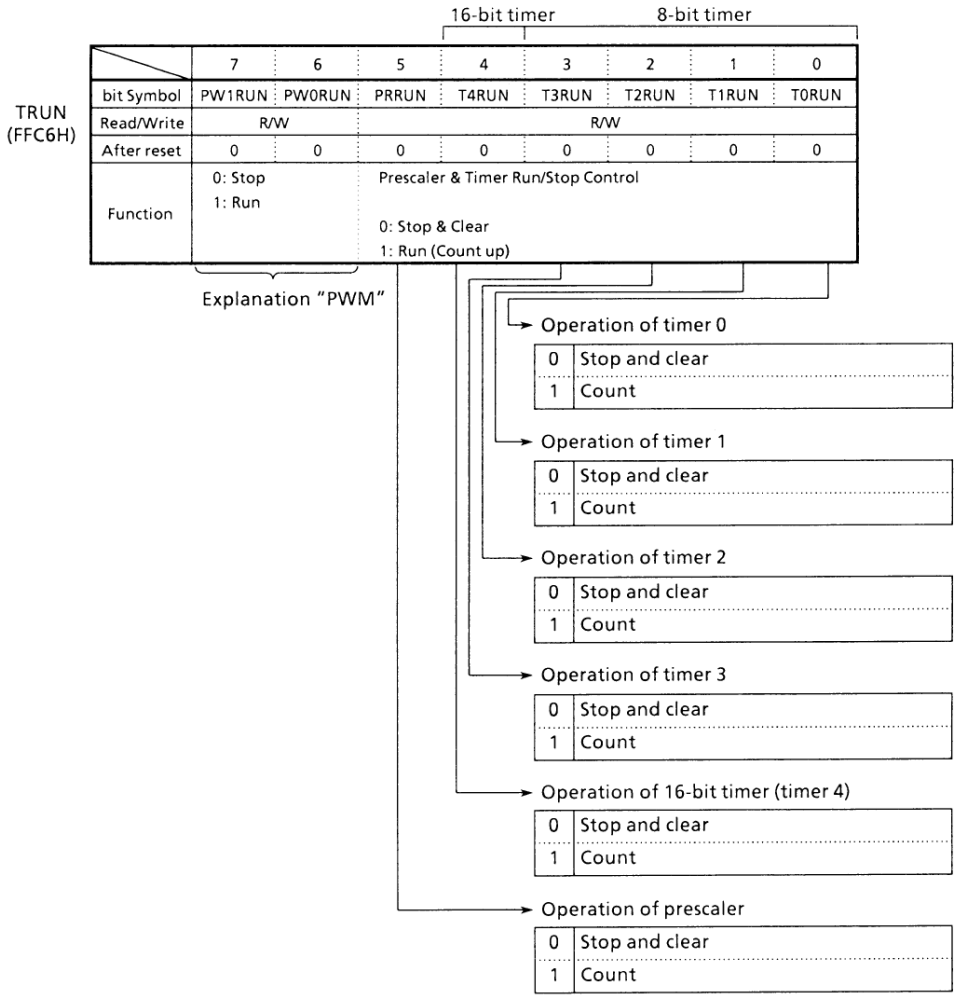


Figure 3.6 (7). Timer Operation Control Register (TRUN)

TRDC (FFCDH)

	7	6	5	4	3	2	1	0
bit Symbol						TR4DE	TR2DE	TR0DE
Read/Write						R/W		
After reset						0	0	0
Function						0: Double Buffer Disable 1: Double Buffer Enable		

Control the double buffer of timer register 0

0	Disabled
1	Enabled

Control the double buffer of timer register 2

0	Disabled
1	Enabled

Control the double buffer of timer register 4

0	Disabled
1	Enabled

Figure 3.6 (8). Timer Register Double Buffer Control Register (TRDC)

④ Comparators

A comparator compares the value in the up-counter with the values to which the timer register is set. When they match, the up-counter is cleared to zero and an interrupt signal (INTT0 ~ INTT3) is generated. If the timer flipflop inversion is enabled, the timer flipflop is inverted at the same time.

⑤ Timer flipflop (timer F/F)

The status of the timer flipflop is inverted by the match detect signal (comparator output) of each interval timer and the value can be output to the timer output pins TO1 (also used as P55) and TO3 (also used as P57).

A timer F/F is provided for each pair of timer 0 and timer 1 as well as that of timer 2 and timer 3 and is called TFF1 and TFF3. TFF1 is output to TO1 pin, while TFF3 is output to TO3 pin.

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

(1) 8-bit timer mode

Four interval timers 0, 1, 2, and 3 can be used independently as an 8-bit interval timers. All interval timers operate in the same manner, thus only the operation of timer 1 will be explained below.

① Generating interrupts in a fixed cycle

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

Example: To generate timer 1 interrupt every 40 microseconds at fc = 16MHz, set each register in the following manner.

```
          MSB          LSB
          7 6 5 4 3 2 1 0
TRUN  ←   - - - - - - 0 -     Stop timer 1, and clear it to "0".
T01MOD←   0 0 X X 0 1 - -     Set the 8-bit timer mode, and select φT1 (0.5 μs @
                              fc = 16 MHz) as the input clock.
TREG1 ←   0 1 0 1 0 0 0 0     Set the timer register at 40 μs φT1 = 50.
INTEH ←   X - - - - 1 - -     Enable INTT1.
TRUN  ←   X X 1 - - - 1       Start timer 1 counting.
```

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

Use the following table for selecting the input clock:

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

Interrupt cycle (at fc = 16MHz)	Resolution	Input clock
0.5μs ~ 128ms	0.5μs	øT1 (8/fc)
2μs ~ 512ms	2μs	øT16 (32/fc)
8μs ~ 2,048ms	8μs	øT256 (128/fc)
128μs ~ 32,768ms	128μs	øT256 (2048/fc)

② Generating pulse a 50% square wave pulse

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

Example: To output a 3.0μs square wave pulse from TO1 pin at fc = 16MHz, set each register in the following procedures. Either timer 0 or timer 1 may be used, but this example uses timer 1.

	MSB 7	6	5	4	3	2	1	LSB 0	
TRUN ←	–	–	–	–	–	–	0	–	Stop timer 1, and clear it to "0".
TO1MOD ←	0	0	X	X	0	1	–	–	Set the 8-bit timer mode, and select ϕT1 (0.5 μs @ fc = 16 MHz) as the input clock.
TREG1 ←	0	0	0	0	0	0	1	1	Set the timer register at 3.0 μs ÷ ϕT1 ÷ 2 = 3.
TFFCR ←	–	–	–	–	1	0	1	1	Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.
P5CR ←	–	–	1	–	–	–	–	–	} Select P55 as TO1 pin. (To be non wait mode in wait control bit)
P5FR ←	–	–	X	X	–	–	–	1	
TRUN ←	X	X	1	–	–	–	1	–	Start timer 1 counting.

(Note) X; Don't care – ; No change

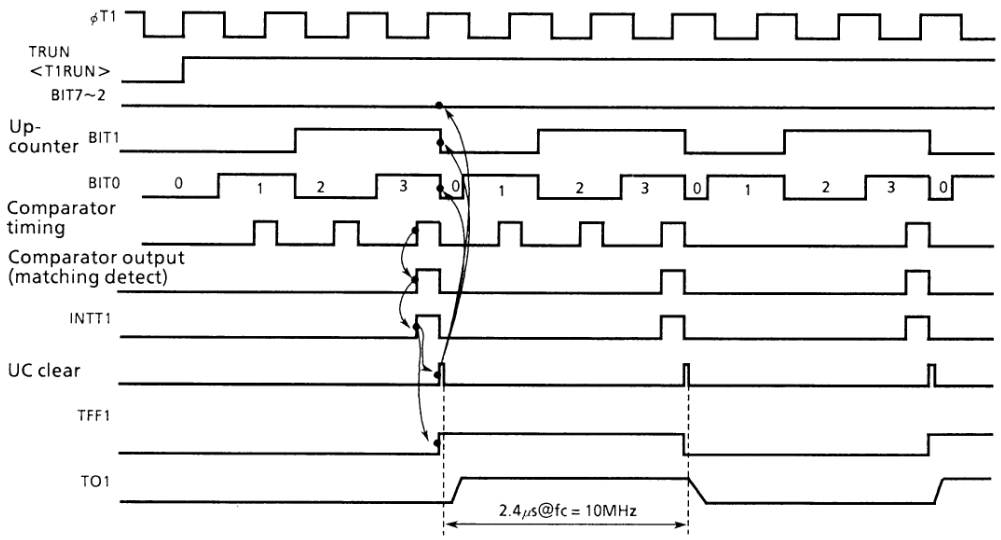


Figure 3.6 (9). Square Wave (50% Duty) Output Timing Chart

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

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

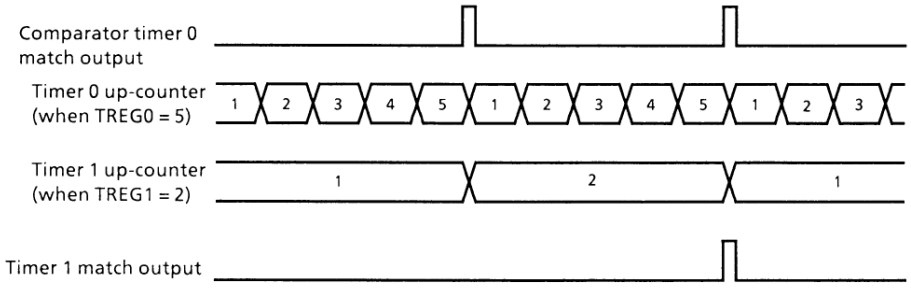


Figure 3.6 (10)

④ Output inversion with software

The value of timer flipflop (timer F/F) can be inverted, independent of the timer operation.

Writing “00” to TFFCR <FF1C1, 0> inverts the value of TFF1, and writing “00” into TFFCR <FF3C1,0> inverts TFF3.

⑤ Initial setting of timer flipflop (timer F/F)

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

For example, write “10” in TFFCR <FF1C1,0> to clear TFF1 to “0”, while write “01” in TFFCR <FF3C1,0> to set TFF1 to “1”.

Note: The value of timer F/F and timer register cannot be read.

(2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of timer 0 and timer 1 or that of timer 2 and timer 3.

As the above two pairs operate in the same manner, only the case of combining timer 0/timer 1 is discussed.

To make a a 16-bit interval timer by cascade connecting timer 0 and timer 1, set timer 0/timer 1 mode register T01MOD <T01M1,0> to “0,1”.

When set in 16-bit timer mode, the overflow output of timer 0 will become the input clock of timer 1, regardless of the set value of T01MOD <T1CLK1,0>. Table 3.6 (2) shows the relation between the cycle of timer (interrupt) and the selection of input clock.

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

Interrupt cycle (at fc = 16MHz)	Resolution	Input clock
0.5μs ~ 32.768ms	0.5μs	øT1 (8/fc)
2μs ~ 131.072ms	2μs	øT16 (32/fc)
8μs ~ 524.288ms	8μs	øT256 (128/fc)

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

Setting example: To generate interrupt INTT1 every 0.5 seconds at fc = 16MHz, set the following values for timer registers TREG0 and TREG1.

When counting by using øT16 (8μs @16MHz),

0.5sec ÷ 8μs = 62500 = F424H

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

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

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

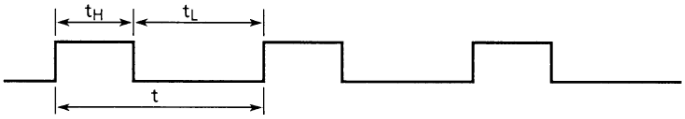
Example: When TREG1 = 04H and TREG0 = 80H

Figure 3.6 (11)

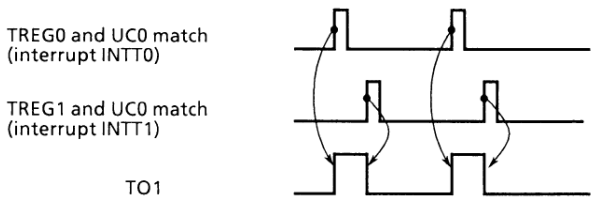
(3) 8-bit PPG (Programmable Pulse Generation) Mode

Square wave pulse can be generated at any frequency and duty by timer 0 or timer 2. The output pulse may be either low-active or high-active.

In this mode, timer 1 and timer 3 cannot be used. Timer 0 outputs pulse to TO1 pin, (also used as P55), and timer 2 outputs to TO3 pin (also used as P57).



As an example, the case of timer 0 will be explained below. (Timer 2 also functions in the same way).



In this mode, a programmable square wave is generated by inverting timer output each time the 8-bit up-counter (UC0) matches the timer registers TREG0 and TREG1.

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

Though the up-counter (UC1) of timer 1 cannot be used in this mode, timer 1 can be used for counting by setting TRUN <T1RUN> to "1".

Figure 3.6 (12) shows the block diagram for this mode.

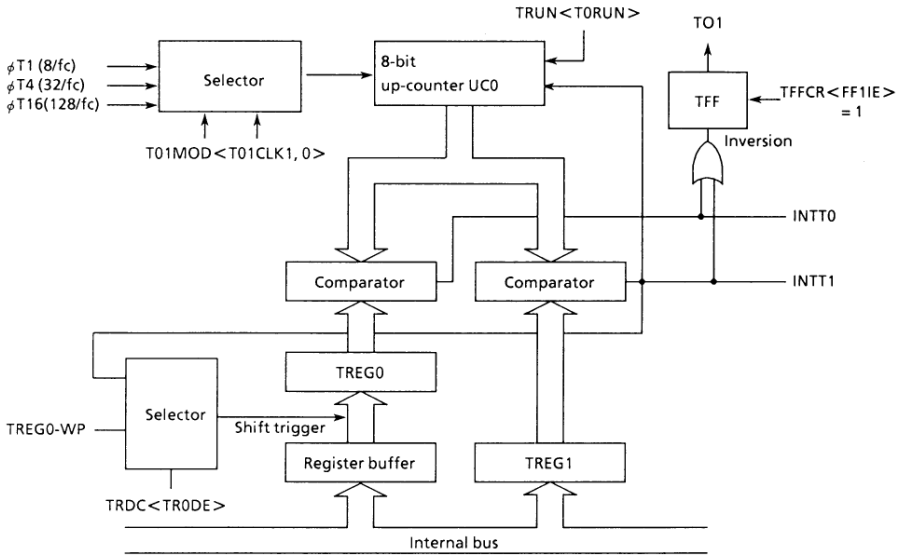
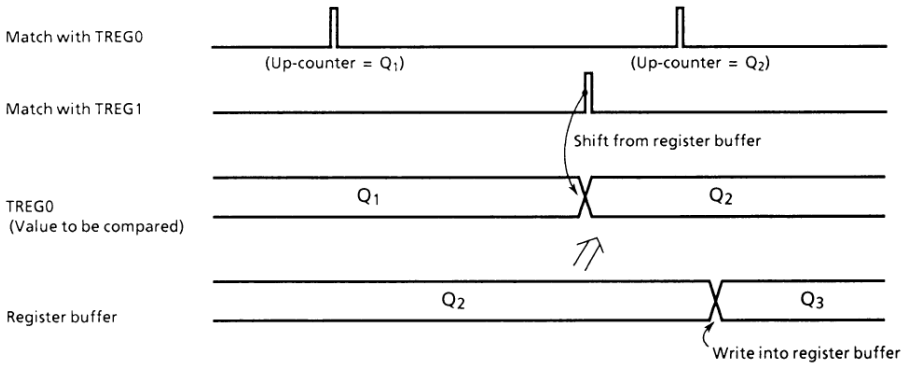


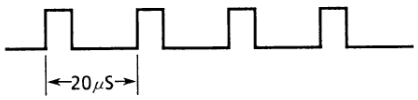
Figure 3.6 (12). Block Diagram of 8-bit PPG Mode

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

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



Example: Generating 1/4 duty 50KHz pulse (@fc = 16MHz)



- Calculate the value to be set for timer register

To obtain the frequency 50KHz, the pulse cycle t should be: 1/50KHz = 20μs.

Given øT1 = 0.5μs (@ 16MHz)

20μs ÷ 0.5μs = 40

Consequently, to set the timer register 1 (TREG1) to TREG1 = 40 = 28H and the duty to 1/4, t x 1/4 = 20μs x 1/4 5μs

5μs ÷ 0.5μs = 10

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

```
          MSB              LSB
       ←  7 6 5 4 3 2 1 0
TRUN   ←  X X - - - - 0 0    Stop timer 0, and clear it to "0".
T01MOD ←  1 0 X X X X 0 1    Set the 8-bit PPG mode, and select φT1 as input clock.
TFFCR  ←  - - - - 0 1 1 x    Sets TFF1 and enable the inversion.
                             Writing "10" provides negative logic pulse.
TREG0  ←  0 0 0 0 1 0 1 0    Write "0AH".
TREG1  ←  0 0 1 0 1 0 0 0    Write "28H".
P5CR   ←  - - 1 - - - - 1  }
P5FR   ←  - - X X - - - 1  }  Set P55 as the TO1 pin.  ( To be non wait mode in wait control bit )
TRUN   ←  X X 1 - - - 1 1    Start timer 0 counting.
```

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

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

This mode is valid only for timer 0 and timer 2. In this mode, maximum two PWMs of 8-bit resolution (PWM0 and PWM2) can be output.

PWM pulse is output to TO1 pin (also used as P55) when using timer 0, and to TO3 pin (also used as P57) when using timer 2.

Timer 1 and timer 3 can also be used as 8-bit timer.

As an example, the case of timer 0 will be explained below. (Timer 2 also operates in the same way.)

Timer output is inverted when up-counter (UC0) matches the set value of timer register TREG0 or when 2n - (n = 6, 7, or 8; specified by T01MOD <PWM0,10>) counter overflow occurs. Up-counter UC0 is cleared when 2n - 1 counter overflow occurs. For example, when n = 6, 6-bit PWM will be output, while when n = 7, 7-bit PWM will be output.

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

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

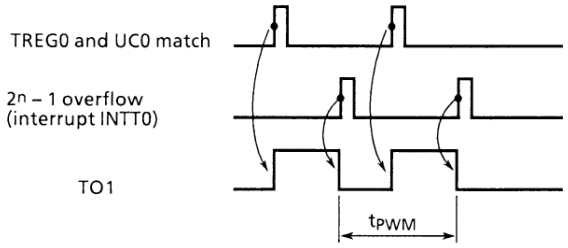


Figure 3.6 (13) shows the block diagram of this mode.

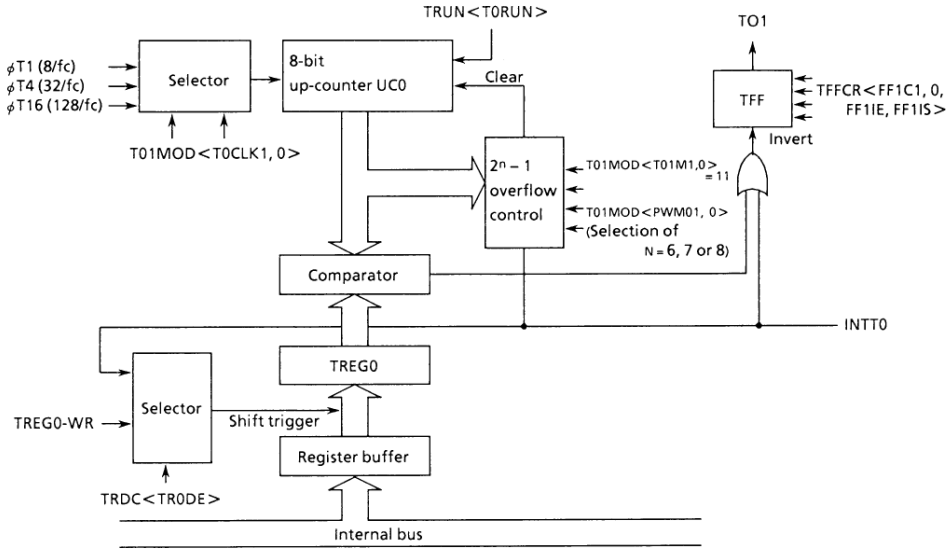
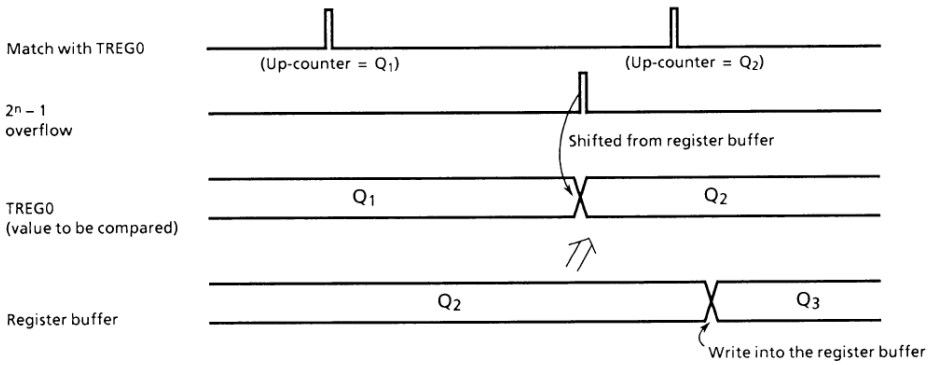


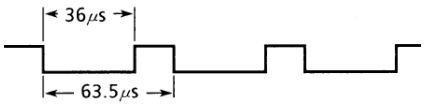
Figure 3.6 (11). Block Diagram of 8-bit PWM Mode

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

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



Example: To output the following PWM waves to TO1 pin using timer 0 at fc = 16MHz.



- Calculate the value to be set for timer register

To realize 63.5μs of PWM cycle by øT1 = 0.5μs (@ = 16MHz),

$63.5\mu s \div 0.5\mu s = 127 = 2^7 - 1$

Consequently, n should be set to 7.

As the period of low level is 36μs, for øT1 = 0.5μs, set the following value for TREG0.

$36\mu s \div 0.5\mu s = 72 = 48H$

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

TO1MOD  ←         1 1 1 0 - - 0 1      Set 8-bit PWM mode (cycle: 2^7 – 1) and select φT1 as the
                                       input clock.
TFFCR   ←         - - - - 1 0 1 X      Clears TFF1 to enable the inversion.
TREG0   ←         0 1 0 0 1 0 0 0      Writes "48H".
P5CR    ←         - - 1 - - - - -  }   Set P55 as the TO1 pin.   ( To be non wait mode in
P5FR    ←         - - - - - - - 1  }                               wait control bit )
TRUN    ←         X X 1 - - - - 1      Start timer 0 counting.

(Note)   X ; Don't care      - ; No change
```

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

	Formula	PWM cycle (@ fc = 16MHz)		
		øT1 (8/fc)	øT4 (32/fc)	øT16 (128/fc)
2^6 - 1	(2^6 - 1) x øTn	31.5μs	126μs	504μs
2^7 - 1	(2^7 - 1) x øTn	63.5μs	254μs	1.01ms
2^8 - 1	(2^8 - 1) x øTn	127μs	510μs	2.04ms

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

Table 3.6 (4) Timer Mode Setting Registers

Register name	T01MOD (T23MOD)				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	–	–	External clock øT1, øT16, øT256 (01, 10, 11)	–
8-bit timer x 2 channels	00	–	Lower timer match øT1, øT16, øT256 (01, 01, 10, 11)	External clock øT1, øT16, øT256 (01, 01, 10, 11)	0: Lower timer output 1: Upper timer output
8-bit PPG x 1 channel	10	–	–	External clock øT1, øT16, øT256 (01, 01, 10, 11)	–
8-bit PWM x 1 channel	11	2^6 - 1, 2^7 - 1, 2^8 - 1 (01, 10, 11)	–	External clock øT1, øT16, øT256 (01, 01, 10, 11)	–
8-bit timer x 1 channel	11	–	øT1, øT14, øT16 (01, 10, 11)	–	Output disabled

(Note) –: Don't care

• Lower timer external input clock has T2CLK. But it does not have T0CLK.

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

The TMP90CM36 has one multifunctional 16-bit timer/event counter with the following operation modes:

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

Figure 3.7 (1) shows the block diagram of 16-bit timer/event counter.

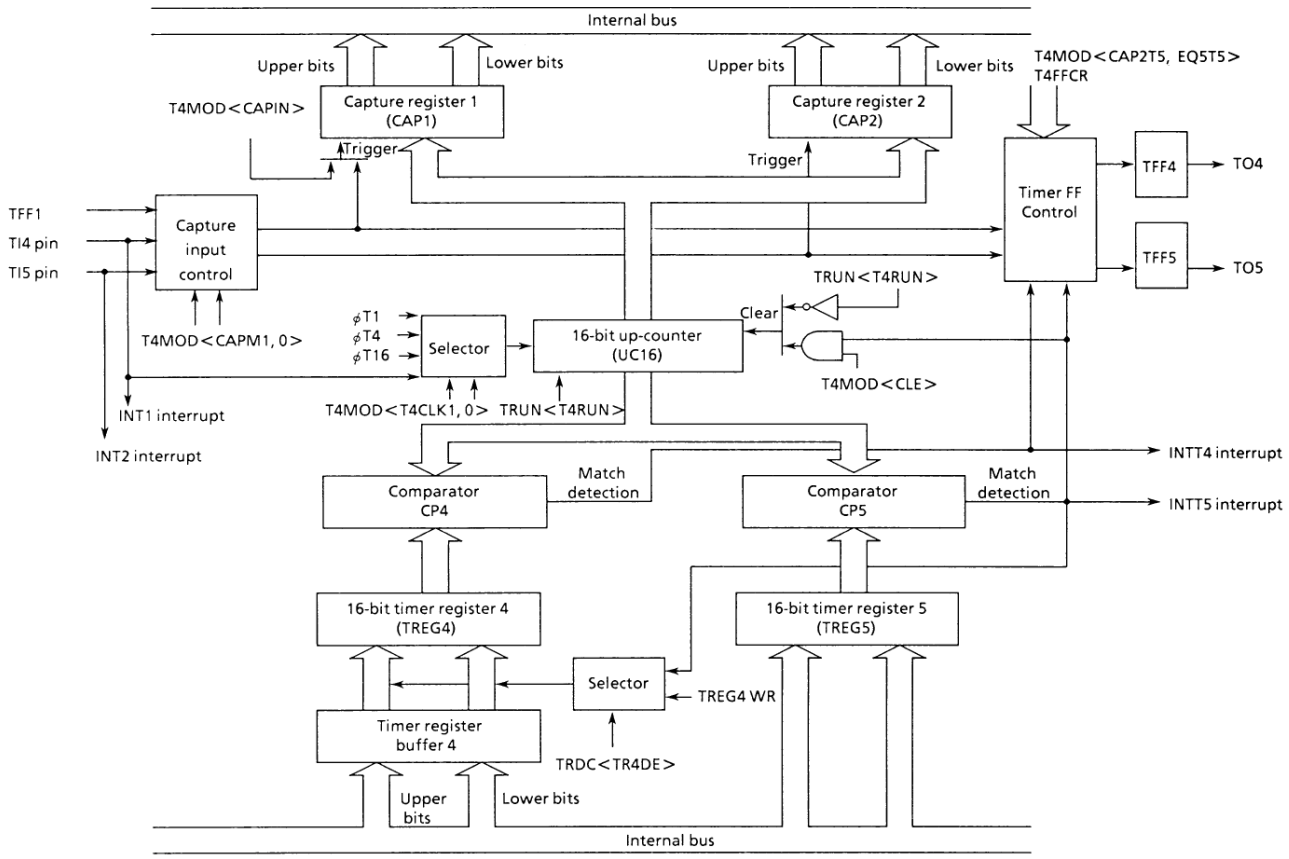


Figure 3.7 (1). Block Diagram of 16-Bit Timer/Event Counter (Timer 4)

Timer/event counter con sists of 16-bit up-counter, two 16-bit timer registers, and two 16-bit capture registers, two comparators, register buffer, capture input controller, timer flipflop, and the control circuit.

Timer/event counter is controlled by 4 control registers: T4MOD, T4FFCR, TRUN, and TRDC. TRUN register includes 8-bit timer controller. For TRUN and TRDC registers, see Figure 3.6 (7) and Figure 3.6 (8).

T4MOD (FFD3H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T5	EQ5T5	CAP1IN	CAPM1	CAPM0	CLE	T4CLK1	T4CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF5 invert trigger 0: Disable 1: Enable		0: Soft- Capture 1: don't care	Capture timing 00: Disable INT1 occurs at rise edge. 01: T14↑ T15↑ INT1 occurs at rise edge. 10: T14↑ T14↓ INT1 occurs at fall edge. 11: TFF1↑ TFF1↓ INT1 occurs at rise edge.		1: UC16 Clear Enable	Timer 4 source clock 00: TI4 01: ϕT1 (8/fc) 10: ϕT4 (32/fc) 11: ϕT16 (128/fc)	

Timer 4 input clock

00	External clock (TI4)
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Clearing the up-counter UC16

0	Clear disable
1	Clear by match with TREG5.

Figure 3.7 (2). 16-Bit Timer/Event Counter (Timer 4) Control/Mode Register (1/2)

T4MOD (FFD3H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T5	EQ5T5	CAP1IN	CAPM1	CAPM0	CLE	T4CLK1	T4CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF5 invert trigger 0: Disable 1: Enable		0: Soft-Capture 1: don't care	Capture timing 00: Disable INT1 occurs at rise edge. 01: TI4↑ TI5↑ INT1 occurs at rise edge. 10: TI4↑ TI4↓ INT1 occurs at fall edge. 11: TFF1↑ TFF1↓ INT1 occurs at rise edge.		1: UC16 Clear Enable	Timer 4 source clock 00: TI4 01: ϕT1 (8/fc) 10: ϕT4 (32/fc) 11: ϕT16 (128/fc)	

Capture control/INT1 interrupt control

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

Software capture trigger

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

Timer flip-flop 5 (TFF5) invert trigger

0	Trigger disable
1	Trigger enable

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

Figure 3.7 (2). 16-Bit Timer/Event Counter (Timer 4) Control/Mode Register (2/2)

T4FFCR (FFD4H)

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

Timer flip-flop 4 (TFF4) control

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

Timer flip-flop 4 (TFF4) invert trigger

0	Trigger disable
1	Trigger enable

CAP2T4 ; When the up-counter value is loaded to CAP2
CAP1T4 ; When the up-counter value is loaded to CAP1
EQ5T4 ; When up-counter matches TREG5
EQ4T4 ; When up-counter matches TREG4

Timer flip-flop 5 (TFF5) control

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

Figure 3.7 (3). 16-Bit Timer/Event Counter Timer flipflop Control Register

① Up-counter (UC16)

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

As the input clock, one of the internal clocks øT1 (8/fc), øT4 (32/fc), and øTI6 (128/fc) from 9-bit prescaler (also used as 8-bit timer), and external clock from TI4 pin (commonly used as P46/INT1 pin) can be selected. When reset, it will be initialized to <T4CLK1,0> = 00 to select TI4 input mode. Counting, stop, or clearing of the counter in controlled by timer operation control register TRUN <T4RUN>.

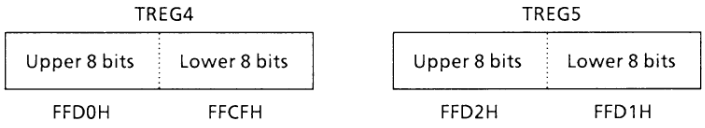
When clearing is enabled, up-counter UC16 will be cleared to zero each time it coincides matches the timer register TREG5. The “clear enable/disable” is set by T4MOD <CLE>.

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

② Timer registers (TREG4 and TREG5)

These two 16-bit registers are used to set the value of the counter. When the value of up-counter UC16 matches the set value of this timer register, the comparator match detect signal will become active.

Setting data for timer register (TREG4 and TREG5) is executed using 16-bit transfer instruction od using 8-bit transfer instruction twice for lower 8 bits and upper 8 bits in order.



TREG4 timer register is of double buffer structure, which is paired with register buffer. TREG4 controls whether the double buffer should be enabled or disabled, using the timer register double buffer control register TRDC <TR4DE>; disable when <TR4DE> = 0, while enable when <TR4DE> = 1.

When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up-counter and TREG5.

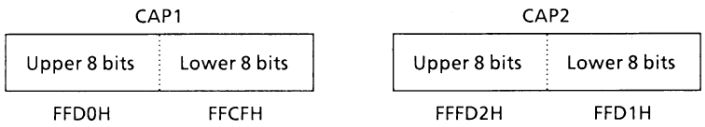
When reset, it will be initialized to <TR4DE> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the register buffer.

TREG4 and register buffer 4 are allocated to the same memory addresses FFCFH and FFD0H. When <TR4DE> = 1, the value is written into only the register buffer.

③ Capture register (CAP1 and CAP2)

These 16-bit registers are used to hold the values of the up-counter UC16.

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



④ Capture input control circuit

This circuit controls the timing to latch the value of up-counter UC16 into CAP1 and CAP2. The latch timing of capture register is controlled by register T4MOD <CAPM1,0>.

• When T4MOD <CAPM1, 0> = 00

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

• When T4MOD <CAPM1, 0> = 01

Data is loaded to CAP1 at the rise edge of TI4 pin (commonly used as P53/INT1) input, while data is loaded to CAP2 at the rise edge of TI5 pin (commonly used as P57/INT2) input. (Time difference measurement)

• When T4MOD <CAPM1, 0> = 10

Data is loaded to CAP1 at the rise edge of TI4 pin input, while to CAP2 at the fall edge. Only in this setting, interrupt INT1 occurs at fall edge. (Pulse width measurement)

• When T4MOD <CAP1, 0> = 11

Data is loaded to CAP1 at the rise edge of timer flipflop TFF1, while to CAP2 at the fall edge. (Frequency measurement)

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

⑤ Comparators (CP4, CP5)

These are 16-bit comparators which compare the up-counter UC16 value with the set value of TREG4 or TREG5 to detect the match. When a match is detected, the comparators generate an interrupt INTT4 and INTT5, respectively. The up-counter UC16 is cleared only when UC16 matches TREG5. (The clearing of up-counter UC16 can be disabled by setting 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 signals to the capture registers (CAP1 and CAP2). Disable/enable of inversion can be set for each element by T4FFCR <CAP2T4, CAP1T4, EQ5T4, EQ4T4>. TFF4 will be inverted when "00" is written in T4FFCR <TFF4C1,0>. Also, it is set to "1" when "10" is written, and cleared to "0" when "10" is written. The value of TFF4 can be output to the timer output pin TO4 (commonly used as P51).

⑦ Timer flipflop (TFF5)

This flipflop is inverted by the match detect signal from the comparator CP5 and the latch signal to the capture register CAP2. TFF5 will be inverted when "00" is written in T4FFCR <TFF5C1,0>. Also, it is set to "1" when "10" is written, and cleared to "0" when "10" is written. The value of TFF5 can be output to the timer output pin TO5 (commonly used as P52).

(1) 16-bit Timer Mode

In this example, the interval time is set in the timer register TREG5 to generate the interrupt INTT5.

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

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

(2) 16-bit Event Counter Mode

In timer mode as described in above (1), the timer can be used as an event counter by selecting the external clock (TI4 pin input) as the input clock. To read the value of the counter, first perform “software capture” once and read the captured value.

The counter counts at the rising edge of TI4 pin input. TI4 pin can also be used as P46/INT1.

```
TRUN  ←  - - - 0 - - - -          Stop timer 4.

INTEL ←  0 0 1 - - - - -          Enable INTT5, while disables INTT4 and INT1.
T4FFCR←  1 1 0 0 0 0 1 1          Disable trigger.
T4MOD ←  0 0 1 0 0 1 0 0          Select TI4 as the input clock.
TREG5 ←  **** **** **** ****      Set the number of counts (16 bits).
TURN  ←  - - 1 1 - - - -          Start timer 4.
```

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

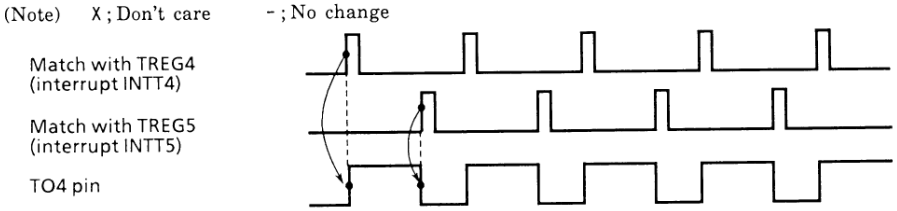
(3) 16-bit Programmable Pulse Generation (PPG) Mode

The PPG mode is entered by inversion of the timer flip-flop TFF4 that is to be enabled bu the match of the up-counter UC16 with the timer register TREG4 or 5 and to be output to TO4 (also used as P51). In this mode, the following conditions must be satisfied.

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

```
TRUN  ←  - - - 0 - - - -          Stop timer 4.
TREG4 ←  **** **** **** ****      Set the duty.
TREG5 ←  **** **** **** ****      Set the cycle.
T4FFCR←  1 1 0 0 1 1 0 0          Set the TFF4 inversion to be effected by match with TREG4 or
                                  TREG5. Initialize TFF4 to “0”.
T4MOD ←  0 0 1 0 0 1 * *          Select the internal clock for the input, and disable the capture
                 (**=01,10,11)    function.
P5CR  ←  - - - - - - - 1 -
                                  Assign P51 as TO4.
P5FR  ←  - - - - - - 1 - -
TRUN  ←  - - 1 1 - - - -          Start timer 4.
```

(Note) X ; Don’t care - ; No change



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

(4) Application examples of capture function

The loading of up-counter (UC16) vaules into the capture registers CAP1 and CAP2, the timer flipflop TFF4 inversion due to the match detection by comparators CP4 and CP5, and the output of the TFF4 status to TO4 pin can be enabled or disabled. Combined with interrupt function, they can be applied in many ways, for example.

① One-shot pulse output 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 external trigger pulse.

Set the up-counter UC16 in free-running mode with the internal input clock, input the external trigger pulse from TI4 pin, and load the value of up-counter into capture register CAP1 at the rise edge of the TI4 pin. Then set to T4MOD <CAPM1,0> = 01.

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

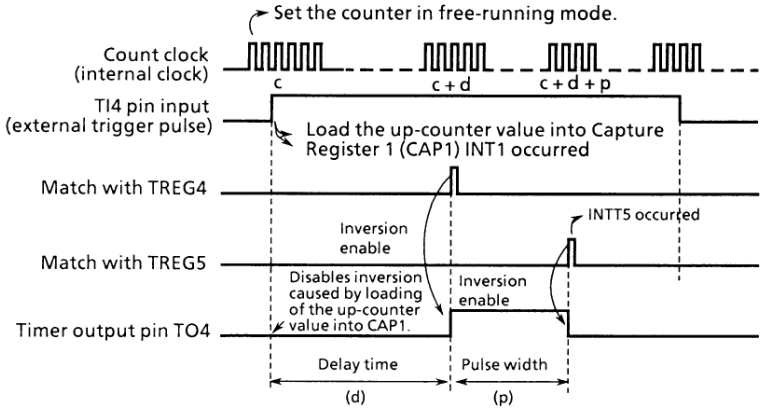


Figure 3.7 (4). One-Shot Pulse Output (with Delay)

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

Main setting

```
T4MOD ← 0 0 1 0 1 0 0 1   → Keep counting (Free-runnig)
                          → Count with φT1.
                          → Load the up-counter value into CAP1 at the rise edge
T4FFCR← 1 1 0 0 0 0 0 0     of T14 pin input.
                          → Clear TFF4 to zero.
                          → Disable TFF4 inversion.
P5CR  ← - - - - - - - 1 -
            (**=00,01,10) } Select P51 as the TO4 pin.
P5FR  ← - - - - - - 1 - -
INTEL ← 0 1 0 - - - - -   → Enable INT1, and disable INTT4 and INTT5.

TRUN  ← - - 1 1 - - - -   → Start timer 4.
```

Setting of INT1

```
TREG4 ← CAP1+3ms/φT1
TREG5 ← TREG4+2ms/φT1
T4FFCR← X X - - 1 1 - -   → Enable TFF4 inversion when the up-counter value
                            matches TREG4 or 5.
INTEL ← - - 1 - - - - -   → Enable INTT5.
```

Setting of INT5

```
T4FFCR← 1 1 - - 0 0 - -   → Disable TFF4 inversion when the up-counter value
                            matches TREG4 or 5.
INTEL ← - - 0 - - - - -   → Disable INTT5.
```

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

When delay time is unnecessary, invert timer flipflop TFF4 when the up-counter value is loaded into loaded into capture register 1 (CAP1), and set the CAP1 value (c) plus the one-shot pulse width (p) to TREG5 when the interrupt INT1 occurs. The TFF4 inversion should be enabled when the up-counter (UC16) value matches TREG5, and disabled when generating the interrupt INTT5.

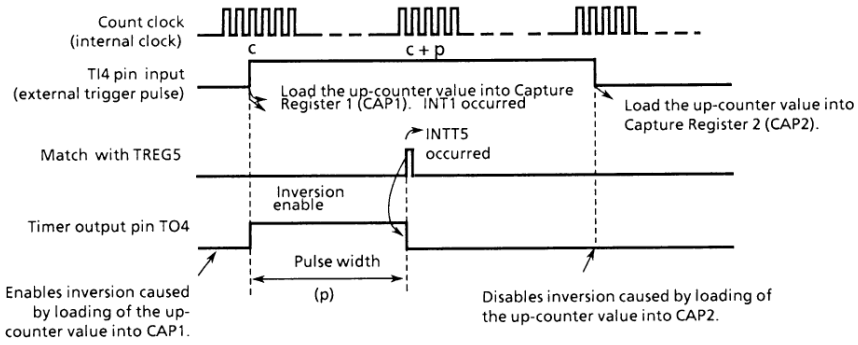


Figure 3.7 (5). One-Shot Pulse Output (without Delay)

② Frequency measurement

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

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

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

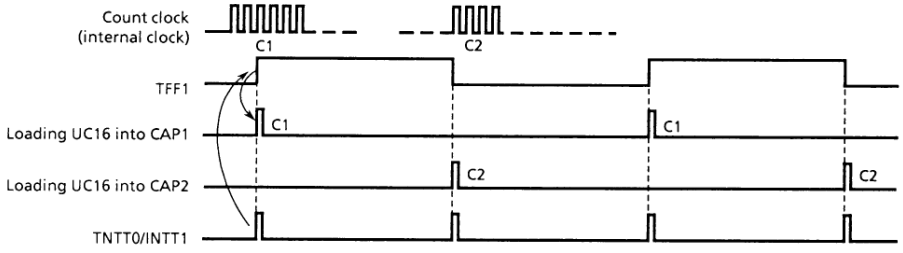


Figure 3.7 (6). Frequency Measurement

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

③ Pulse width measurement

This mode allows to measure the “H” level width of an external pulse. While keeping the 16-bit timer/event counter counting (free-running) with the internal clock input, the external pulse is input though the TI4. Then the capture function is used to load the UC16 values into CAP1 and CAP2 at the rising edge and falling edge of the external trigger pulse, respectively. The interrupt INT1 occurs at the falling edge of TI4.

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

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

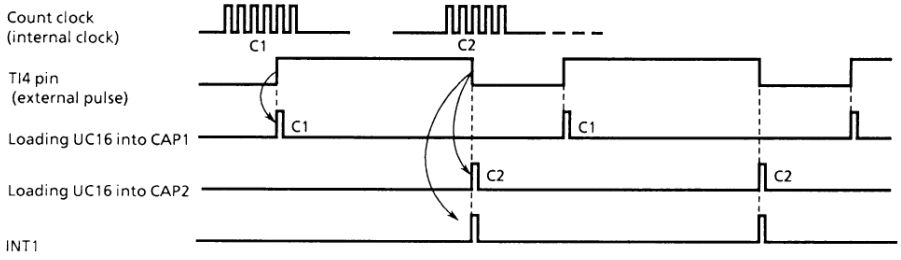


Figure 3.7 (7). Pulse Width Measurement

Note: Only in this pulse width measuring mode (T4MOD <CAPM1, 0> = 10), external interrupt INT1 occurs at the falling edge of TI4 pin input. In other modes, it occurs at the rising edge.

The width of “L” level can be measured from the difference between the first C2 and the second C1 at the second INT1 interrupt.

④ Time difference measurement

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

Keep the 16-bit timer/event counter (Timer 4) counting (free-running) with the internal clock, and load the UC16 value into CAP1 at the rising edge of the input pulse to TI4. Then the interrupt INT1 is generated.

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

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

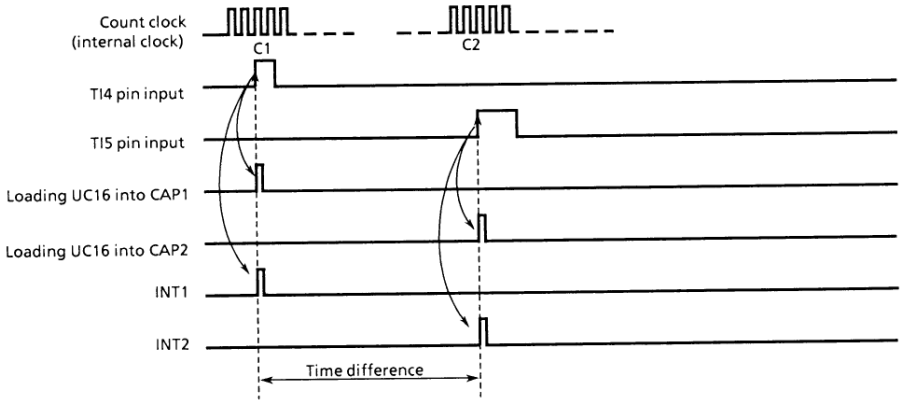


Figure 3.7 (8). Time Difference Measurement

3.8 Serial Channels

The TMP90CM36 contains three serial channels (SIO0,1, 2).

The three serial channels have the following operation modes.

Channel	Action mode
SIO0	• Asynchronous transmission (UART) mode — mode 1 : 7-bit data; mode 2 : 8-bit data; mode 3 : 9-bit data
SIO1	• I/O interface mode 1 — mode 0 : To transmit and receive I/O data as well as the synchronizing signal SCLK for extending I/O. • Asynchronous transmission (UART) mode — mode 1; mode 2; mode 3
SIO2	• I/O interface mode 2 — mode 4

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

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

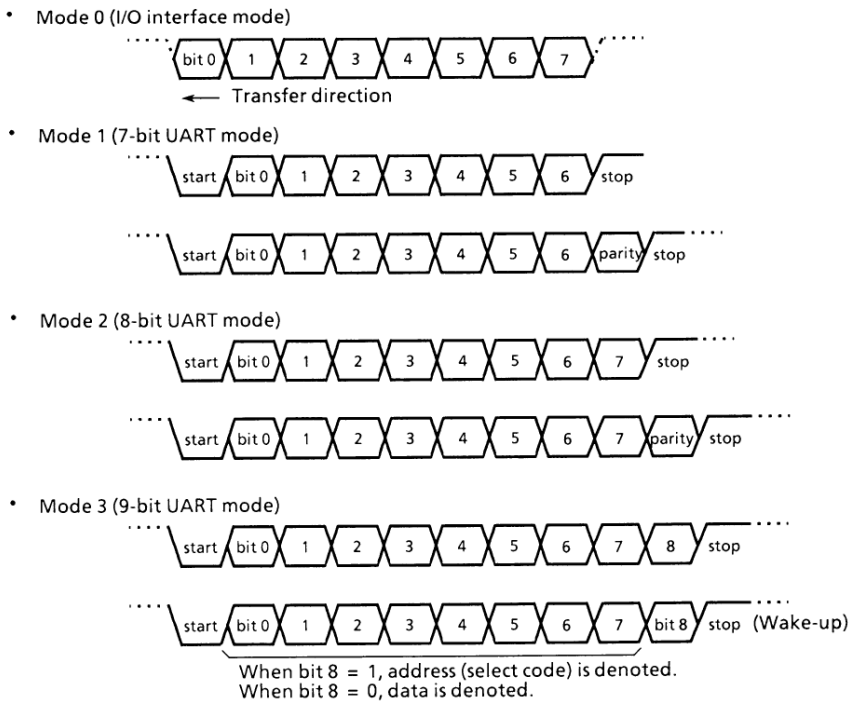


Figure 3.9 (1). Data Formats

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

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

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

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

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

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

The SIO0 or SIO1 includes a special baud rate generator, which can set any baud rate can be set by dividing by the frequency of 4 clocks (øT0, øT2, øT8, and øT32) from the internal prescaler (shared by 8-bit/16-bit timer) by the value of 2 to 16.

Internal clock (SIO2) is able to select in speed from øT0, øT1, øT4, and øT16.

(1) Serial Channel (SIO0) (3.8.1 ~ 3.8.3)

3.8.1 Control Registers

The serial channel SIO0 is controlled by 4 control registers SCMOD0, SCCR0, BRGCR0, and P7FR. Transmitted and received data are stored in register SCBUF0.

SCMOD0 (FFEEH)

	7	6	5	4	3	2	1	0
bit Symbol	TB80	Fixed to "0"	RXE0	WU0	SM0	SM00	SC10	SC00
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data Bit 8		1: Receive Enable	1: Wake Up Enable	00: 01: UART 7Bit 10: UART 8Bit 11: UART 9Bit		00: TO2TRG 01: BRG Mode 10: ϕ1 11: –	

Serial transmission clock

	UART mode
00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	Reserved (cannot be used)

Serial transmission mode

00	Cant be used	
01	UART mode	7-bit length
10		8-bit length
11		9-bit length

Wake-up function

	9-bit UART	Other mode
0	Interrupt when data are received.	don't care
1	Interrupt only when RB80 = 1.	

Enable receiving

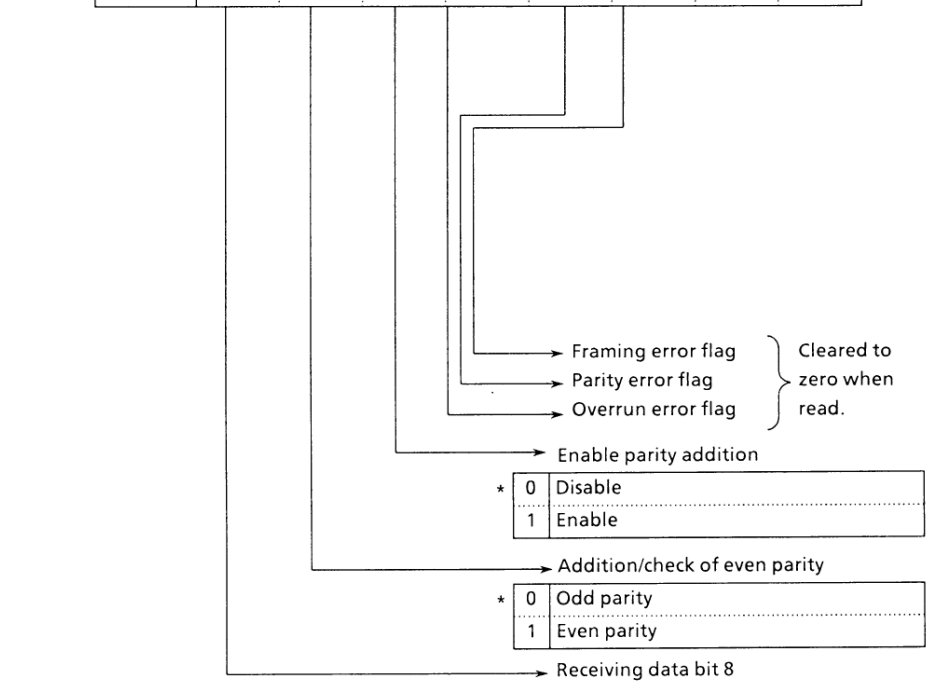
0	Disable
1	Enable

Transmission data bit 8

Figure 3.8 (2). Serial Channel Mode Register (SCMOD0)

SCCR0 (FFEFH)

	7	6	5	4	3	2	1	0
bit Symbol	RB80	EVEN0	PE0	OERR0	PERR0	FERR0		
Read/Write	R	R/W		R (Cleared to zero when read)				
After reset	Undefined	0	0	0	0	0		
Function	Received data Bit 8	Parity 0: Odd 1: EVEN	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing		



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

Figure 3.8 (3). Serial Channel Mode Register (SCCR0)

SCBUF0 (FFF0H)

	7	6	5	4	3	2	1	0	
	TB70	TB60	TB50	TB40	TB30	TB20	TB10	TB00	(Transmission)
	RB70	RB60	RB50	RB40	RB30	RB20	RB10	RB0	(Receiving)

Figure 3.8 (4). Serial Transmission/Receiving Buffer Registers (SCBUF0)

BRGCR0 (FFF1H)

	7	6	5	4	3	2	1	0
bit Symbol	Fixed to "0"		BG10	BG00	PS30	PS20	PS10	PS00
Read/Write	R/W							
After reset	0		0	0	0	0	0	0
Function			00: fc/4 01: fc/16 10: fc/64 11: fc/256		Divided frequency of prescaler			

Setting of the divided frequency of baud rate generator (SIO0)

0000	16 divisions
0010 ↓ 1111	2 to 15 divisions
"0001" cannot be set.	

Selecting the input clock of baud rate generator (SIO0)

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

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

Figure 3.8 (5). Baud Rate Generator Control Register (BRGCR0)

Port 7 Function Register

P7FR (FFB1H)

	7	6	5	4	3	2	1	0
bit Symbol	ODE2	TXDC2	SCLKC2	ODE1	TXDC1	SCLKC1	ODE0	TXDC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	P77 0: CMOS 1: Open Drain	P77 0: Port 1: TxD Output	P76 0: Port 1: SCLK Output	P74 0: CMOS 1: Open Drain	P74 0: Port 1: TxD Output	P73 0: Port 1: SCLK Output	P71 0: CMOS 1: Open Drain	P710: Port 1: TxD Output

Setting P71 as TxD0 output

0	Port output
1	TxD0 output

Setting P71 as open drain output

0	CMOS output
1	Open drain output

Setting P73 as SCLK1 output

0	Port output
1	SCLK1 output

Setting P74 as oTxD output

0	Port output
1	TxD1 output

Setting P74 as open drain output

0	CMOS output
1	Open drain output

Setting P76 as SCLK2 output

0	Port output
1	SCLK2 output

Setting P77 as TxD output

0	Port output
1	TxD2 output

Setting P77 as open drain output

0	CMOS output
1	Open drain output

Figure 3.8 (6). Port 7 Function Register

3.8.2 Configuration

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

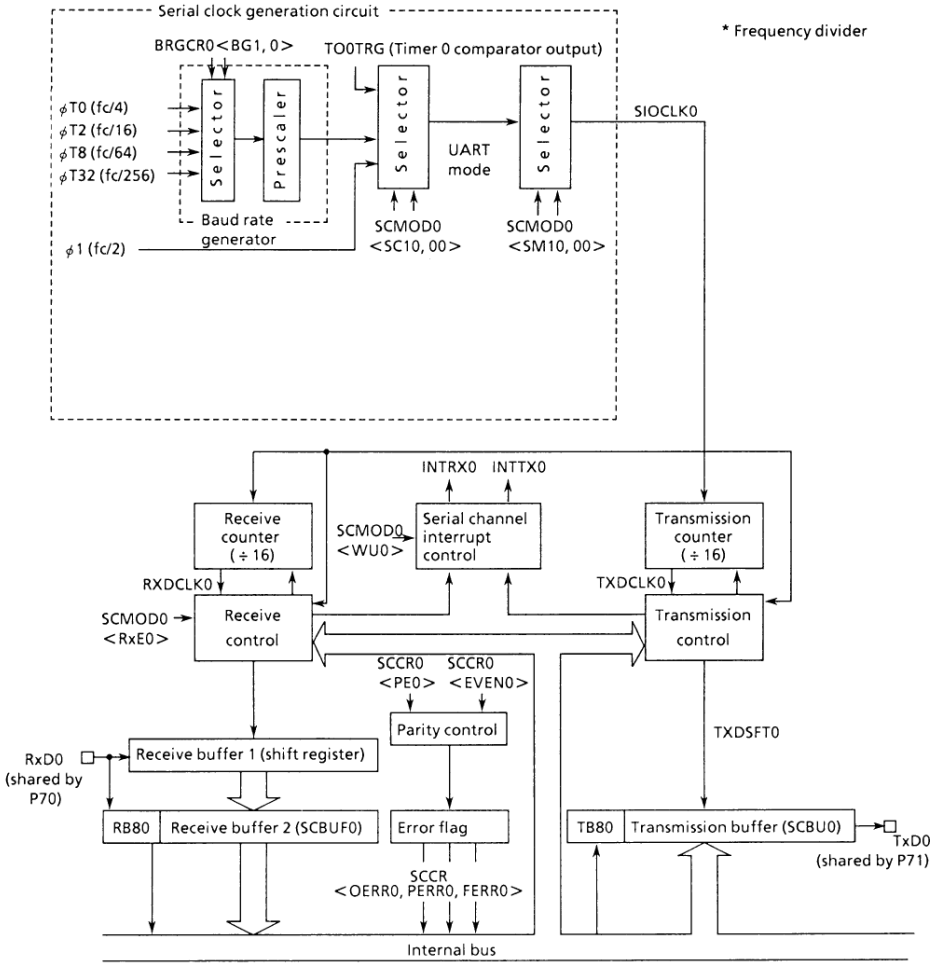


Figure 3.8 (7). Block Diagram of the Serial Channel (SIO0)

① Baud rate generator

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

The input clock to the baud rate generator, øT0 (fc/4), øT2 (fc/16), øT8 (fc/64), or øT32 (fc/256) is generated by the 9-bit prescaler which is shared by the timers. One of these input clocks is selected by the baud rate genorator control register BRGCR0 <BG10,00>.

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

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

• UART mode

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

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

øT0 = fc/4
øT2 = fc/16
øT8 = fc/64
øT32 = fc/256

Accordingly, when source clock fc is 12.288MHz, input clock øT2 (fc/16), frequency divider is 5, the transfer in UART mode becomes as follows.

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

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

Table 3.8 (1) shows an example of the transfer in UART mode.

Also with 8-bit timer 0, the serial channel (SIO0) can get a transfer rate. Table 3.8 (2) shows an example of baud rate using timer 0.

Table 3.8 (1) Selection of Transfer Rate (1) (When Baud Rate Generator is Used) Unit: Kbps

Source clock (fc)	Input clock / Frequency clock	øT0 (fc/4)	øT2 (fc/16)	øT8 (fc/64)	øT32 (fc/256)
9.8304MHz	–	2457.600	614.400	153.600	38.400
–	2	76.800	19.200	4.800	1.200
–	4	38.400	9.600	2.400	0.600
–	8	19.200	4.800	1.200	0.300
–	0	9.600	2.400	0.600	0.150
12.288MHz	–	3072.000	786.000	192.000	48.000
–	5	38.400	9.600	2.400	0.600
–	A	19.200	4.800	1.200	0.300
14.7456MHz	–	3686.400	921.600	230.400	57.600
–	3	76.800	19.200	4.800	1.200
–	6	38.400	9.600	2.400	0.600
–	C	19.200	4.800	1.200	0.300

Table 3.8 (2). Selection of Transfer Rate (2) (When Timer 0 (Input Clock øT1) is Used) Unit: Kbps

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

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

$$\text{Baud rate} = \frac{1}{\text{TREG0}} \times \frac{1}{16} \times \text{Input clock of timer 0}$$

Input clock of timer 0

øT1 = fc/8
øT4 = fc/32
øT16 = fc/128

② Serial clock generation circuit

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

Asynchronous communication (UART) mode

According to the setting of SCMOD0 <SC10,00>, the above baud rate generator clock, internal clock ø1 (fc/2) (312.5 Kbaud at 10MHz), or the match detect signal from timer 0 will be selected to generate the basic clock SIOCK0.

③ Receiving counter

The receiving counter is a 4-bit binary counter used in asynchronous communication (UART) mode and counts up by SIOCLK0 clock. 16 pulses of SIOCLK0 are used for receiving 1 bit of data, and the data is sampled three times at the 7th, 8th and 9th clock.

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

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

④ Receiving control

Asynchronous communication (UART) mode

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

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

⑤ Receiving buffer

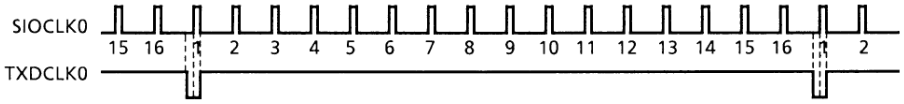
To prevent overrun from occurring, the receiving buffer has a double structure. Received data are stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data are stored in the receiving buffer 1, the stored data are transferred to another receiving buffer 2 (SCBUF0), generating an interrupt INTRX0. The CPU reads only receiving buffer 2 (SCBUF0). Even before the CPU reads the receiving buffer 2 (SCBUF0), the received data can be stored in receiving buffer 1 . However, unless the receiving buffer 2 (SCBUF0) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of receiving buffer 2 and SCCR0 <RB80> is still preserved.

The parity bit added in 8-bit UART mode and the most significant bit (MSB) in 9-bit UART mode are stored in SCCR0 <RB80>.

When in 9-bit UART, the wake-up function of the slave controllers is enabled by setting SCMOD0 <WU0> to “1”, and interrupt INTRX occurs only when SCCR0 <RB80> is set to “1”.

⑥ Transmission counter

Transmission counter is a 4-bit binary counter which is used in asynchronous communication (UART) mode and, like a receiving counter, counts by SIOCLK0 clock, generating TxDCLK0 every 16 clock pulses.



⑦ Transmission controller

Asynchronous communication (UART) mode

When the transmission data are written in the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TxDCLK0, generating a transmission shift clock TxDSFT0.

⑧ Transmission buffer

Transmission buffer SCBUF0 shifts out and sends the transmission data written from the CPU from the least significant bit (LSB) in order, using transmission shift clock TxDSFT0 which is generated by the transmission control. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX0 interrupt.

⑨ Parity control circuit

When serial channel control register SCCR0 <PE0> is set to “1” , it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART or 8-bit UART mode. With SCCR0 <EVEN0> register, even (odd) parity can be selected.

For transmission, parity is automatically generated according to the data written in the transmission buffer SCBUF, and data are transmitted after being stored in SCBUF0 <TB70> when in 7-bit UART mode while in SCMOD0 <TB80> in 8-bit UART mode. <PE0> and <EVEN0> must be set before transmission data are written in the transmission buffer.

For receiving, data are shifted in the receiving buffer 1, and parity is added after the data are transferred in the receiving buffer 2 (SCBUF0), and then compared with <RB70> of SCBUF0 when in 7-bit UART mode and with SCCR0 <RB80> when in 8-bit UART mode. If they are not equal, a parity error occurs, and SCCR0 <PERR0> flag is set.

⑩ Error flag

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

1) Overrun error (SCCR0 <OERR0>)

If all bits of the next data are received in receiving buffer 1 while valid data are still stored in receiving buffer 2 (SCBUF0), an overrun error will occur.

2) Parity error (SCCR0 <PERR0>)

The parity generated for the data shifted in receiving buffer 2 (SCBUF0) is compared with the parity bit received from the RxD0 pin. If they are not equal, a parity error occurs.

3) Framing error (SCCR0 <FERR0>)

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

⑪ Generation Timing

1) UART mode

Receiving

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

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

Transmitting

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

3.8.3 Action Explanation

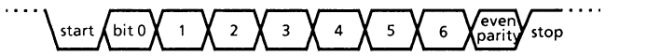
(1) Mode 1 (7-Bit UART Mode)

7-bit mode can be set by setting serial channel mode register SCMOD0 <SM10,00> to "01".

In this mode, a parity bit can be added, and the addition of a parity bit can be enabled or disabled by serial channel control register.

SCCR0 <PE0>, and even parity or odd parity is selected by SCCR0 <EVEN0> when <PE0> is set to "1" (enable).

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



← Transfer direction (transfer rate: 2400 bps at fc = 12.288 MHz)

```
        7 6 5 4 3 2 1 0
P7CR  ← - - - - - - 1 -     } Select P71 as the TxD0 pin.
P7FR  ← - - - - - - - 1
SCMOD0← X 0 - X 0 1 0 1     Set 7-bit UART mode.
SCCR0 ← X 1 1 X X X X X     Add an even parity.
BRGCR0← 0 X 1 0 0 1 0 1     Set transfer rate at 2400 bps.
TRUN  ← X X 1 - - - - -     Start the prescaler for the baud rate generator.
INTE2 ← - - - - - - 1 -     Enable INTTX0 interrupt.
SCBUF0← * * * * * * * *     Set data for transmission.
```

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

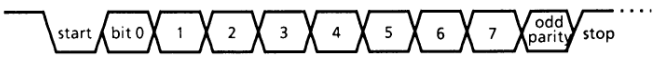
(2) Mode 2 (8-Bit UART Mode)

8-bit UART mode can be set by setting serial channel mode register SCMOD0 <SM10,00> to "10".

In this mode, a parity bit can be added, the addition of a parity bit is enabled or disabled by SCCR0 <PE0>.

SCCR0 <PE0>, and even parity or odd parity is selected by SCCR0 <EVEN0> when <PE0> is set to "1" (enable).

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



← Direction of transmission (transmission rate: 9600 bps @ fc = 12.288 MHz)

Main setting

```
        7 6 5 4 3 2 1 0
P7CR  ← - - - - - - - 1     Specify P70 (RxD0) as the input pin.
SCMOD0← - 0 1 X 1 0 0 1     Enable receiving in 8-bit UART mode.
SCCR0 ← X 0 1 X X X X X     Add an odd parity.
BRGCR0← 0 X 0 1 0 1 0 1     Set transfer rate at 9600 bps.
TRUN  ← X X 1 - - - - -     Start the prescaler for the baud rate generator.
INTE2 ← - - - - - - - 1     Enable INTRX0 interrupt.
```

INTRX0 processing

```
Acc   ← SCCR0 AND 00011100    Check for error.
if Acc ≠  0 then error
Acc   ← SCBUF                 Read the received data.
```

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

(3) Mode 3 (9-Bit UART Mode)

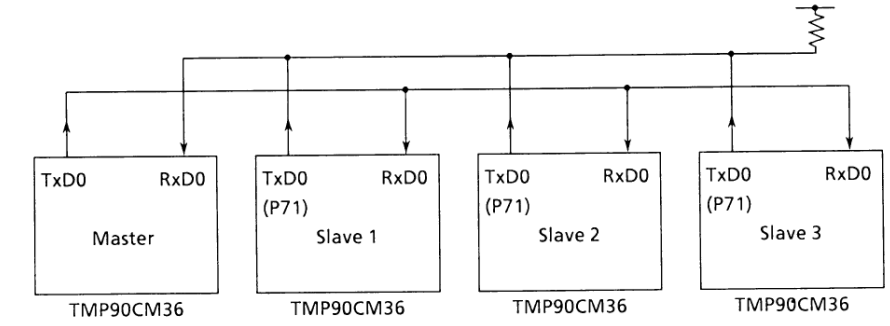
9-bit UART mode can be specified by setting SCMOD0 <SM10,00> to “11”. In this mode, parity bit cannot be added.

For transmission, the MSB (9th bit) is written in SCMOD0 <TB80>, while in receiving it is stored in SCCR0 <RB80> . For writing or reading the buffer, the MSB is read or written first then SCBUF0.

Wake-up function

In 9-bit UART mode, the wake-up function of slave controllers is enabled by setting SCMOD0 <WU0> to “1”.

The interrupt INTRX0 occurs only when SCCR0 <RB80> = 1.



Note : TxD0 pin of the slave controllers must be in open drain output mode.

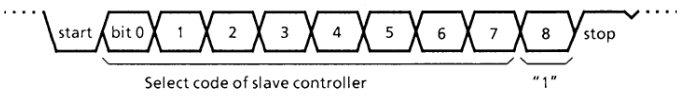
Figure 3.8 (8). Serial Link Using the Wake-up Function

Protocol

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

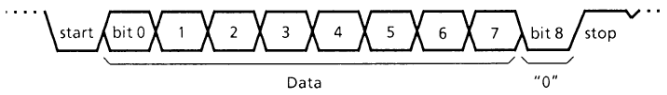
② Set the SCMOD0 <WU0> bit of each slave controller to “1” to enable data receiving.

③ The master controller transmits one-frame including the 8-bit select code for the slave controllers. The MSB (bit 8) SCMOD0 <TB80> is set to “1”.



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

⑤ The master controller transmits data to the specified slave controller whose SCMOD0 <WU0> bit is cleared to “0”. The MSB (bit 8) SCMOD0 <TB80> is set to “0”.

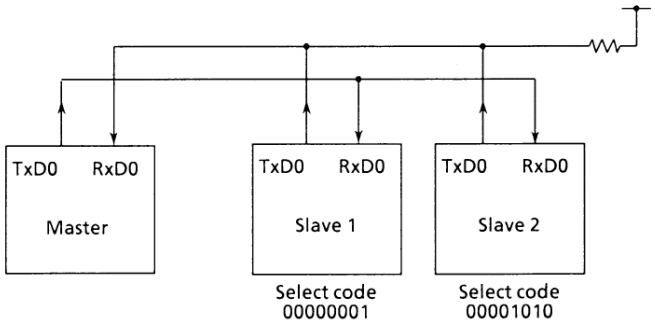


⑥ The other slave controllers (with SCMOD0 <WU0> bit remaining at “1”) ignore the receiving data because their MSBs (bit 8 or SCCR0 <RB80>) are set to “0” to disable the interrupt INTRX0.

When the WU0 bit is cleared to “0”, the interrupt INTRX0 occurs, so that the slave controller can read the receiving data.

The slave controllers (WU0 = 0) transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

Setting example: To link two slave controllers serially with the master controller, and use the internal clock ø/1 (fc/2) as the transfer clock .



- Setting the master controller

```
Main
  P7CR  ← - - - - - - - 1 0   } Select P71 as TxD0 pin and P70 as RxD0 pin.
  P7FR  ← - - - - - - - - 1   }
  INTE2 ← - - - - - - - 1 1   Enable INTRX0 and INTTX0.
  SCMOD0← 1 0 1 0 1 1 1 0     Set φ1 (fc/2) as the transmission clock in 9-bit UART mode.
  SCBUF0← 0 0 0 0 0 0 0 1     Set the select code for slave controller 1.

INTTX 0 interrupt
  SCMOD0← 0 - - - - - - -     Sets SCMOD0<TB80> to "0".
  SCBUF0← * * * * * * * *     Set data for transmission.
```

- Setting the slave controller 2

```
Main
  P7CR  ← - - - - - - - 1 0   } Select P70 as RxD0 pin and P71 as TxD0 pin (open drain
  P7FR  ← - - - - - - - 1 1   } output).
  INTE2 ← - - - - - - - 1 1   Enable INTRX0 and INTTX0.
  SCMOD0← 0 0 1 1 1 1 1 0     Set <WU0> to "1" in the 9-bit UART transmission mode
                              with transfer clock φ1 (fc/2).

INTRX0 interrupt
  Acc ← SCBUF0
  if Acc = Select code
  then SCMOD0← - - - 0 - - - -    Clear <WU0> to "0".
```

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

(1) Serial Channel (SIO1) (3.8.4 ~ 3.8.6)

3.8.4 Control Registers

The serial channel SIO1 is controlled by 4 control registers SCMOD1, SCCR1, BRGCR1, and P7FR. Transmitted and received data are stored in register SCBUF1

SCMOD1 (FFF2H)

	7	6	5	4	3	2	1	0
bit Symbol	TB81	Fixed to "0"	RXE1	WU1	SM1	SM01	SC11	SC01
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data Bit 8		1: Receive Enable	1: Wake Up Enable	00: 01: UART 7Bit 10: UART 8Bit 11: UART 9Bit		00: TO2TRG 01: BRG Mode 10: ϕ1 11: –	

Serial transmission clock

	UART mode	I/O interface
00	Timer 2 match detect signal	External clock or baud rate generator
01	Baud rate generator	
10	Internal clock ϕ1	
11	Reserved (cannot be used)	

Serial transmission mode

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

Wake-up function

	9-bit UART	Other mode
0	Interrupt when data are received.	don't care
1	Interrupt only when RB81 = 1.	

Enable receiving

0	Disable
1	Enable

Transmission data bit 8

Figure 3.8 (9). Serial Channel Mode Register (SCMOD1)

SCCR1 (FFF3H)

	7	6	5	4	3	2	1	0
bit Symbol	RB81	EVEN1	PE1	OERR1	PERR1	FERR1	SCLKC1	IOC1
Read/Write	R	R/W		R (Cleared to zero when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data Bit 8	Parity 0: Odd 1: EVEN	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK1 (↑) 1: SCLK1 (↓)	0: SCLK1 output 1: SCLK1 input

Select SCLK1 I/O

0	SCLK1 output mode
1	SCLK1 input mode

Edge selection in SCLK1 input mode

0	Transmits and receives data at the rise edge of SCLK1 (↑).
1	Transmits and receives data at the fall edge of SCLK1(↓).

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

Enable parity addition

*	0	Disable
	1	Enable

Addition/check of even parity

*	0	Odd parity
	1	Even parity

Receiving data bit 8

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

Figure 3.8 (10). Serial Channel Mode Control Register (SCCR1)

SCBUF1 (FFF4H)

	7	6	5	4	3	2	1	0	
	TB71	TB61	TB51	TB41	TB31	TB21	TB11	TB01	(Transmission)
	7	6	5	4	3	2	1	0	
	RB71	RB61	RB51	RB41	RB31	RB21	RB11	RB1	(Receiving)

Figure 3.8 (11). Serial Transmission/Receiving Registers (SCBUF1)

BRGCR1 (FFF5H)

	7	6	5	4	3	2	1	0
bit Symbol	Fixed to "0"	CTSE1	BG11	BG01	PS31	PS21	PS11	PS01
Read/Write		R/M	R/W					
After reset	0	0	0	0	0	0	0	0
Function		1: CTS Enable	00: fc/4 01: fc/16 10: fc/64 11: fc/256		Divided frequency of prescaler			

Setting of the divided frequency of baud rate generator

0000	16 divisions
0010 ≀ 1111	2 to 15 divisions
"0001" cannot be set.	

Selecting the input clock of baud rate generator

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

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

0	Disable (Ordinary times transmission possible)
1	Enable (It can't use on $\overline{CTS}$ 1 pin be cause of it become SCLK1 pin when I/O interface mode)

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

Figure 3.8 (12). Baud Rate Generator Control Registers (BRGCR1)

Port 7 Function Register

P7FR (FFB1H)

	7	6	5	4	3	2	1	0
bit Symbol	ODE2	TXDC2	SCLKC2	ODE1	TXDC1	SCLKC1	ODE0	TXDC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	P77 0: CMOS 1: Open Drain	P77 0: Port 1: TxD Output	P76 0: Port 1: SCLK Output	P74 0: CMOS 1: Open Drain	P74 0: Port 1: TxD Output	P73 0: Port 1: SCLK Output	P71 0: CMOS 1: Open Drain	P71 0: Port 1: TxD Output

Setting P71 as TxD0

0	Port output
1	TxD0 output

Setting P71 as open drain output

0	CMOS output
1	Open drain output

Setting P73 as SCLK1 output

0	Port output
1	SCLK1 output

Setting P74 as TxD output

0	Port output
1	TxD1 output

Setting P74 as open drain output

0	CMOS output
1	Open drain output

Setting P76 as SCLK2 output

0	Port output
1	SCLK2 output

Setting P77 as TxD output

0	Port output
1	TxD2 output

Setting P77 as open drain output

0	CMOS output
1	Open drain output

Figure 3.8 (13). Port 7 Function Register

3.8.5 Configuration

Figure 3.8 (14) shows the serial channel block diagram.

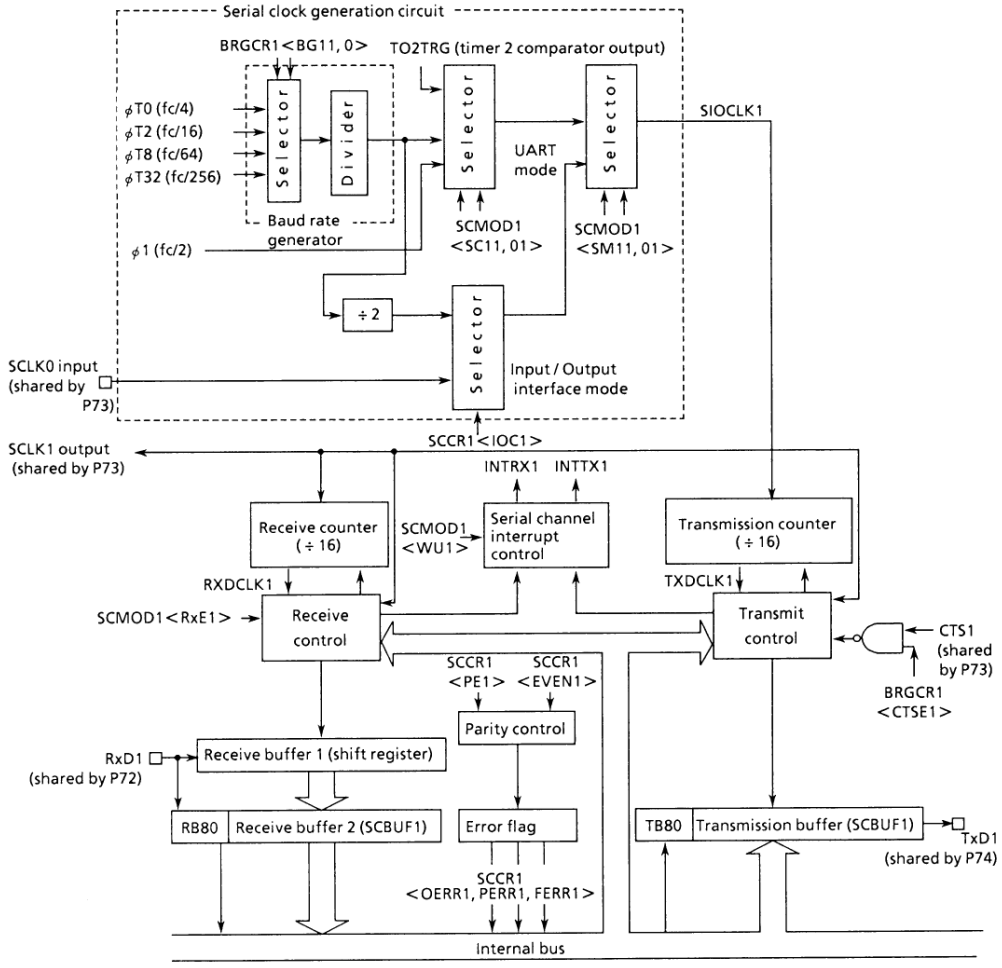


Figure 3.8 (14). Serial Channel (SIO2) Block Diagram

① Baud rate generator

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

The input clock to the baud rate generator, øT0 (fc/4), øT2 (fc/16), øT8 (fc/64), or øT32 (fc/256) is generated by the 9-bit prescaler which is shared by the timers. One of these input clocks is selected by the baud rate generator control register BRGCR1 <BG11,01>.

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

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

• UART mode

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

• I/O interface mode

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

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

øT0 = fc/4
øT2 = fc/16
øT8 = fc/64
øT32 = fc/256

Accordingly, when source clock fc is 12.288MHz, input clock øT2 (fc/16), frequency divider is 5, the transfer in UART mode becomes as follows.

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

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

Table 3.8 (1) shows an example of the transfer in UART mode.

Also with 8-bit timer 1, the serial channel (SIO1) can get transfer rate. Table 3.8 (2) shows an example of baud rate using timer 2.

Table 3.8 (3) Selection of Transfer Rate (1) (When Baud Rate Generator is Used) Unit: Kbps

Source clock (fc)	Input clock Frequency divisor	øT0 (fc/4)	øT2 (fc/16)	øT8 (fc/64)	øT32 (fc/256)
9.8304MHz	–	2457.600	614.400	153.600	38.400
–	2	76.800	19.200	4.800	1.200
–	4	38.400	9.600	2.400	0.600
–	8	19.200	4.800	1.200	0.300
–	0	9.600	2.400	0.600	0.150
12.288MHz	–	3072.000	786.000	192.000	48.000
–	5	38.400	9.600	2.400	0.600
–	A	19.200	4.800	1.200	0.300
14.7456MHz	–	3686.400	921.600	230.400	57.600
–	3	76.800	19.200	4.800	1.200
–	6	38.400	9.600	2.400	0.600
–	C	19.200	4.800	1.200	0.300

Table 3.8 (4) Selection of Transfer Rate (2) (When Timer 0 (Input Clock øT1) is Used) Unit: Kbps

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

How to calculate the transfer rate (when timer 2 is used0)

$$\text{Baud rate} = \frac{1}{\text{TREG2}} \times \frac{1}{16} \times (\text{Input clock of timer 2})$$

Input clock of timer 2
øT1 = fc/8
øT4 = fc/32
øT16 = fc/128

② Serial clock generation circuit

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

1) I/O interface mode

When in SCLK1 output mode with the setting of SCCR1 <IOC1> = “0”, the basic clock will be generated by dividing by 2 the output of the baud rate generator described before. When in SCLK1 input mode with the setting of SCCR1 <IOC1> = “1”, the rising edge or the falling edge will be detected according to the setting of SCCR1 <SCLK1> register to generate the basic clock.

2) Asynchronous communication (UART) mode

According to the setting of SCMOD1 <SC11,01>, the above baud rate generator clock, internal clock ø1 (fc/2) (312.5 Kbaud at 10MHz), or the match detect signal from timer 0 will be selected to generate the basic clock SIOCK.

③ Receiving counter

The receiving counter is a 4-bit binary counter used in asynchronous communication (UART) mode and counts up by SIOCLK1. 16 pulses of SIOCLK1 are used for receiving 1 bit of data, and the data is sampled three times at the 7th, 8th and 9th clock.

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

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

④ Receiving control

1) I/O interface mode

When in SCLK1 output mode with the setting of SCCR1 <IOC1> = "0", RxD1 signal will be sampled at the rising edge of shift clock which is output to SLCK pin.

When in SCLK1 input mode with the setting of SCCR1 <IOC1> = "1", RxD1 signal will be sampled at the rising edge or the falling edge of SCLK1 input according to the setting of SCCR1 <SCLKC1> register.

2) Asynchronous communication (UART) mode

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

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

⑤ Receiving buffer

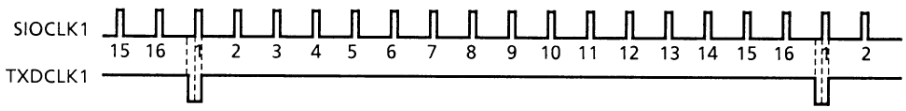
To prevent overrun from occurring, the receiving buffer has a double structure. Received data are stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data are stored in the receiving buffer 1, the stored data are transferred to another receiving buffer 2 (SCBUF1), generating an interrupt INTRX1. The CPU reads only receiving buffer 2 (SCBUF1). Even before the CPU reads the receiving buffer 2 (SCBUF1), the received data can be stored in receiving buffer 1 . However, unless the receiving buffer 2 (SCBUF1) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of receiving buffer 2 and SCCR1 <RB81> is still preserved.

The parity bit added in 8-bit UART mode and the most significant bit (MSB) in 9-bit UART mode are stored in SCCR1 <RB81>.

When in 9-bit UART, the wake-up function of the slave controllers is enabled by setting SCMOD1 <WU1> to "1", and interrupt INTRX occurs only when SCCR1 <RB81> is set to "1".

⑥ Transmission counter

Transmission counter is a 4-bit binary counter which is used in asynchronous communication (UART) mode and, like a receiving counter, counts by SIOCLK1 clock, generating TxDCLK1 every 16 clock pulses.



⑦ Transmission controller

1) I/O interface mode

In SCLK1 output mode with the setting of SCCR1 <IOC1> = "0", the data in the transmission buffer are output bit by bit to TxD1 pin at the rising edge of shift clock which is output from SLCK1 pin.

In SCLK1 input mode with the setting of SCCR1 <IOC1> = "1", the data in the transmission buffer are output bit by bit to TxD1 pin at the rising edge or falling edge of SCLK input according to the setting of SCCR1 <SCLKC1> register.

2) Asynchronous communication (UART) mode

When the transmission data are written in the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TxDCLK1, generating a transmission shift clock TxDSFT1.

⑧ Transmission buffer

Transmission buffer SCBUF1 shifts out and sends the transmission data written from the CPU from the least significant bit (LSB) in order, using transmission shift clock TxDSFT1 which is generated by the transmis-

sion control. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX1 interrupt.

⑨ Parity control circuit

When serial channel control register SCCR1 <PE1> is set to “1” , it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART or 8-bit UART mode. With SCCR1 <EVEN1> register, even (odd) parity can be selected.

For transmission, parity is automatically generated according to the data written in the transmission buffer SCBUF, and data are transmitted after being stored in SCBUF1 <TB71> when in 7-bit UART mode while in SCMOD1 <TB81> in 8-bit UART mode. <PE1> and <EVEN1> must be set before transmission data are written in the transmission buffer.

For receiving, data are shifted in the receiving buffer 1, and parity is added after the data are transferred in the receiving buffer 2 (SCBUF1), and then compared with <RB71> of SCBUF0 when in 7-bit UART mode and with SCCR1 <RB81> when in 8-bit UART mode. If they are not equal, a parity error occurs, and SCCR1 <PERR1> flag is set.

⑩ Error flag

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

1) Overrun error (SCCR1 <OERR1>)

If all bits of the next data are received in receiving buffer 1 while valid data are still stored in receiving buffer 2 (SCBUF1), an overrun error will occur.

2) Parity error (SCCR1 <PERR1>)

The parity generated for the data shifted in receiving buffer 2 (SCBUF1) is compared with the parity bit received from the RxD1 pin. If they are not equal, a parity error occurs.

3) Framing error (SCCR1 <FERR1>)

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

⑪ Generation Timing

1) UART mode

Receiving

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

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

Transmitting

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

3.8.6 Action Explanation

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

This mode is used to increase the number of I/O pins of TMP90CM36 for transmitting or receiving data to or from the external shifter register.

This mode includes SCLK1 output mode to output synchronous clock SCLK1 and SCLK1 input mode to input external synchronous clock SCLK1.

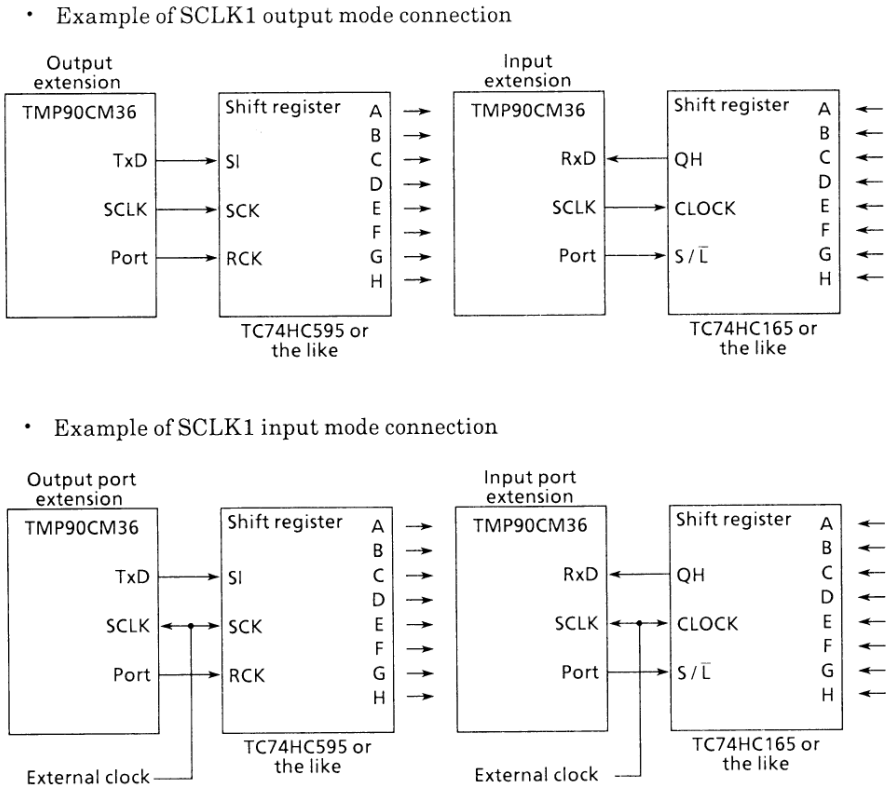


Figure 3.8 (15). I/O Interface Mode

① Transmission

In SCLK1 output mode, 8-bit data and synchronous clock are output from TxD1 pin and SLCK1 pin, respectively, each time the CPU writes data in the transmission buffer. When all data is output, IRF2 <IRFTX1> will be set to generate INTTX interrupt.

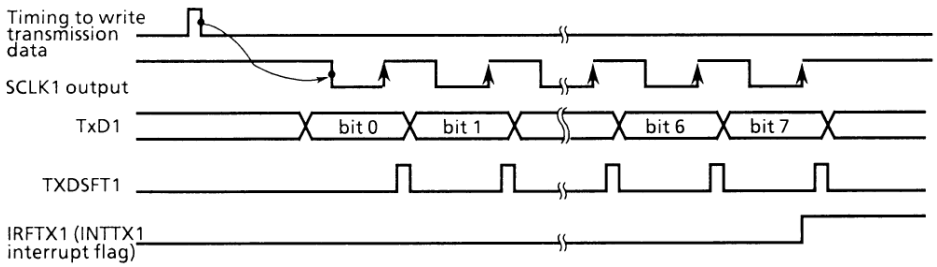


Figure 3.8 (16). Transmitting Operation in I/O Interface Mode (SCLK1 Output Mode)

In SCLK1 output mode, 8-bit data are output from TxD1 pin when SLCK1 input becomes active while data are written in the transmission buffer by CPU. When all data are output, <IRFTX1> will be set to generate INTTX1 interrupt.

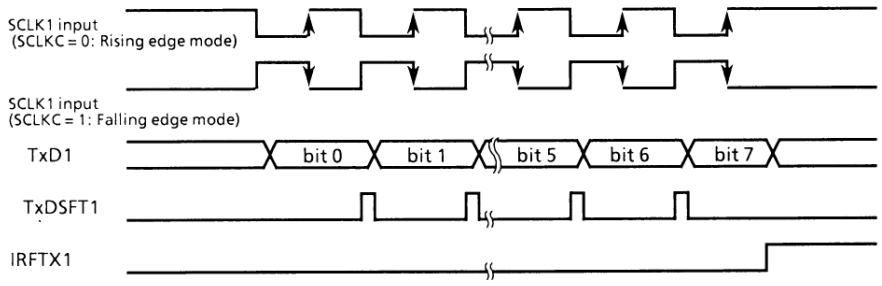


Figure 3.8 (17). Transmitting Operation in I/O Interface Mode (SCLK1 Input Mode)

② Receiving

In SCLK1 output mode, received data are read by the CPU, and synchronous clock is SCLK1 pin and the next data are shifted in the receiving buffer 1 whenever the receive interrupt flag IRF2 <IRFRX1> is cleared. When 8-bit data are received, the data will be transferred in the receiving buffer 2 (SCBUF1), and <IRFRX1> will be set again to generate INTRX1 interrupt.

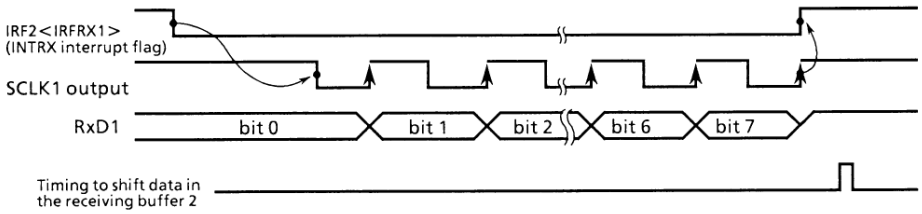


Figure 3.8 (18). Receiving Operation in I/O Interface Mode (SLCK1 Output Mode)

In SCLK1 input mode, received data are read by the CPU, and the next data is shifted in the receiving buffer 1 when SCLK1 input becomes active while the receive interrupt flag <IRFRX1> is cleared. When 8-bit data is received, the data will be transferred in the receiving buffer 2 (SCBUF1), and <IRFRX1> will be set again to generate INTRX1 interrupt.

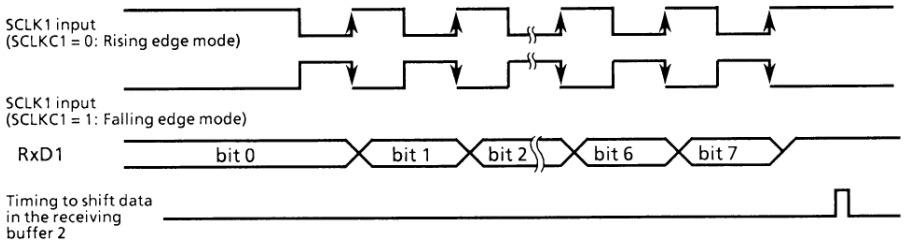


Figure 3.8 (19). Receiving Operation in I/O Interface Mode (SLCK Input Mode)

For data reception, the system must be placed in the receive enable state (SCMOD1 <RXE1> = “1”)

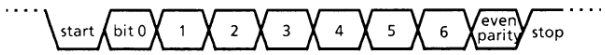
(2) Mode 1 (7-bit UART Mode)

7-bit mode can be set by setting serial channel mode register SCMOD1 <SM11,01> to "01".

In this mode, a parity bit can be added, and the addition of a parity bit can be enabled or disabled by serial channel control register.

SCCR1 <PE1>, and even parity or odd parity is selected by SCCR1 <EVEN1> when <PE1> is set to "1" (enable).

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



← Transfer direction (transfer rate: 2400 bps at fc = 12.288 MHz)

```
        7 6 5 4 3 2 1 0
P7CR  ← - - - 1 - - - -   } Select P74 as the TxD1 pin.
P7FR  ← - - - - 1 - - -   }
SCMOD1← X 0 - X 0 1 0 1   Set 7-bit UART mode.
SCCR1 ← X 1 1 X X X X X   Add an even parity.
BRGCR1← 0 X 1 0 0 1 0 1   Set transfer rate at 2400 bps.
TRUN  ← X X 1 - - - - -   Start the prescaler for the baud rate generator.
INTEL ← - - - - 1 - - -   Enable INTTX1 interrupt.
SCBUF1← * * * * * * * *   Set data for transmission.
```

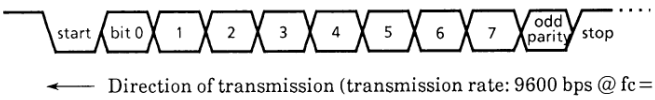
(Note) X ; Don't care - ; No change

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

8-bit UART mode can be specified by setting SCMOD1 <SM11,01> to “10”.

In this mode, a parity bit can be added, the addition of a parity bit is enabled or disabled by SCCR1 <PE1>, and even parity or odd parity is selected by SCCR1 <EVEN1> when <PE1> is set to “1” (enable).

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



← Direction of transmission (transmission rate: 9600 bps @ fc = 12.288 MHz)

Main setting

```
        7 6 5 4 3 2 1 0
P7CR  ← - - - - - 1 - -     Specify P72 (RxD1) as the input pin.
SCMOD1← - 0 1 X 1 0 0 1     Enable receiving in 8-bit UART mode.
SCCR1 ← X 0 1 X X X X X     Add an odd parity.
BRGCR1← 0 X 0 1 0 1 0 1     Set transfer rate at 9600 bps.
TRUN  ← X X 1 - - - - -     Start the prescaler for the baud rate generator.
INTE2 ← - - - - 1 - - 1     Enable INTRX1 interrupt.
```

INTRX1 processing

```
Acc   ← SCCR1 AND 00011100    Check for error.
if Acc ≠  0 then error
Acc   ← SCBUF1                Read the received data.
```

(Note) X ; don’t care - ; no change

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

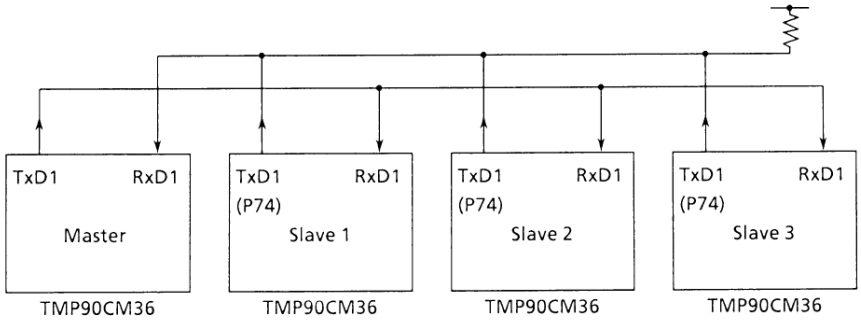
The 9-bit UART mode can be specified by setting SCMOD1 <SM11,01> to “11”. In this mode, parity bit cannot be added.

For transmission, the MSB (9th bit) is written in SCMOD1 <TB81>, while in receiving it is stored in SCCR1 <RB81> . For writing or reading the buffer, the MSB is read or written first then SCBUF1.

Wake-up function

In 9-bit UART mode, the wake-up function of slave controllers is enabled by setting SCMOD1 <WU1> to “1”.

The interrupt INTRX1 occurs only when SCCR1 <RB81> = 1.



Note : TxD1 pin of the slave controllers must be in open drain output mode.

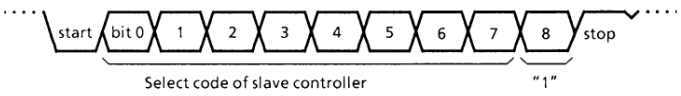
Figure 3.8 (20). Serial Link Using Wake-up Function

Protocol

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

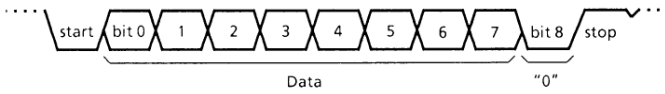
② Set the SCMOD1 <WU1> bit of each slave controller to “1” to enable data receiving.

③ The master controller transmits one-frame, including the 8-bit select code of the slave controllers. The MSB (bit 8) SCMOD1 <TB81> is set to “1”.



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

⑤ The master controller transmits data to the specified slave controller whose SCMOD1 <WU1> bit is cleared to “0”. The MSB (bit 8) SCMOD1 <TB81> is set to “0”.

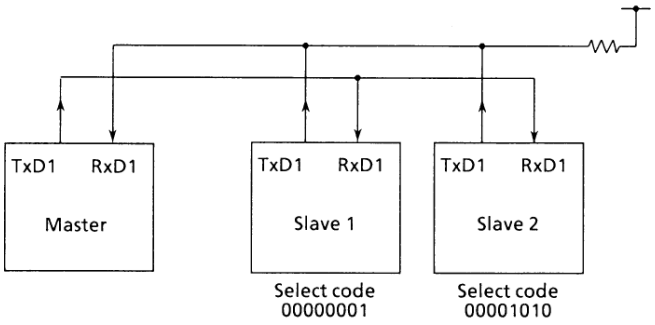


⑥ The other slave controllers (with SCMOD1 <WU1> bit remaining at “1”) ignore the receiving data because their MSBs (bit 8 or SCCR1 <RB81>) are set to “0” to disable the interrupt INTRX1.

When the WU1 bit is cleared to “0”, the interrupt INTRX1 occurs, so that the slave controller can read the receiving data.

The slave controllers (WU1 = 0) transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

Setting example: To link two slave controllers serially with the master controller, and use the internal clock ø/1 (fc/2) as the transfer clock .



- Setting the master controller

Main

```
P7CR  ← - - - 1 - 0 - -   } Select P74 as TxD1 pin and P72 as RxD1 pin.
P7FR  ← - - - - 1 - - -   }
INTE2 ← - - - - 1 1 - -   Enable INTRX1 and INTTX1.
SCMOD1← 1 0 1 0 1 1 1 0   Set φ1 (fc/2) as the transmission clock in 9-bit UART mode.
SCBUF1← 0 0 0 0 0 0 0 1   Set the select code for slave controller 1.
```

INTTX1 interrupt

```
SCMOD1← 0 - - - - - - -   Sets SCMOD1<TB81> to "0".
SCBUF1← * * * * * * * *   Set data for transmission.
```

- Setting the slave controller 2

Main

```
P7CR  ← - - - 1 - 0 - -   } Select P72 as RxD1 pin and P74 as TxD1 pin (open drain
P7FR  ← - - - 1 1 - - -   } output).
INTE2 ← - - - - 1 1 - -   Enable INTRX1 and INTTX1.
SCMOD1← 0 0 1 1 1 1 1 0   Set <WU1> to "1" in the 9-bit UART transmission mode
                          with transfer clock φ1 (fc/2).
```

INTRX1 interrupt

```
Acc  ← SCBUF1
if Acc = Select code
then SCMOD1 ← - - - 0 - - - -   Clear <WU1> to "0".
```

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

3.8.7 Configuration

The serial channels are connected to external circuits through three-pin serial ports: SCLK2 (P76), TXD2 (P77) and RXD2 (P75).

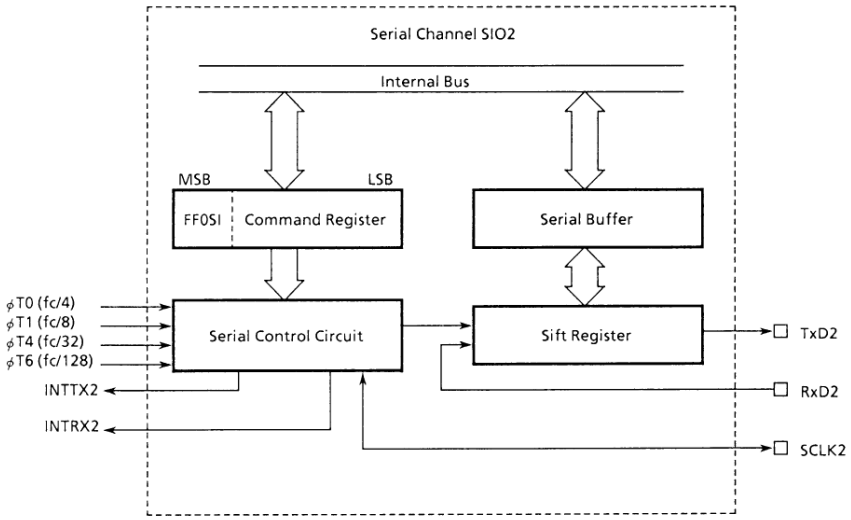


Figure 3.8 (21). Block Diagram of Serial Channels (SIO2)

Serial clock SIO2 pulses make the following selections through the serial channel mode register SCMOD2.

① Clock Source Selection

<SCLKS2> selects either an internal or external clock as the clock source.

a. (Internal clock)

SCMOD2 <CLK1, CLK0> selects the speed of either øT1 (fc/4), øT1 (fc/8), øT4 (fc/32), or øT16 (fc/128) serial clock. The serial clock pulse is externally output from the SCLK2 pin.
The serial clock automatically stops after it ends the "1-frame" serial operation. It waits until next serial operation.

b. (External clock)

<SCLK2> uses the clock pulse externally supplied to the SCLK2 pin as the serial clock pulse.

② Shift Edge Selection

a. Rising edge shift

Data shifts on the serial clock pulses's rising edge (falls at the SCLK2 pin).

b. Falling edge shift

Data shifts on the serial clock pulses's falling edge (rises at the SCLK2 pin or no falling edge shift in send mode)

3.8.8 Explanation of Operations

The send, receive and simultaneous send-receive modes for SCMOD2 <SMD1, SMD0>.

(1) Send Mode

The first send data is written into buffer registers SCBUF after the send mode is set in the command register. (Data will not be written into the buffers if the command register is not in send mode.) Then, storing "1" into serial transfer control registers SCMOD2 <SIOE> starts the send. As the send starts, the send data is synchronized with serial clock pulses and sequentially output from the TxD2 pin on the LSB side.

At the same time, the send data is transferred from the buffer registers to the shift registers. Since the buffer registers are empty, buffer empty interrupt INTTX2 is generated to request the next send data.

When the interrupt service program writes the next send data into the buffer register, the interrupt request signal isn't cleared to "0".

(Internal clock pulse)

In the internal clock operations, if all data is sent and no subsequent data is stored in the register, the serial clock pulse stops and a wait begins.

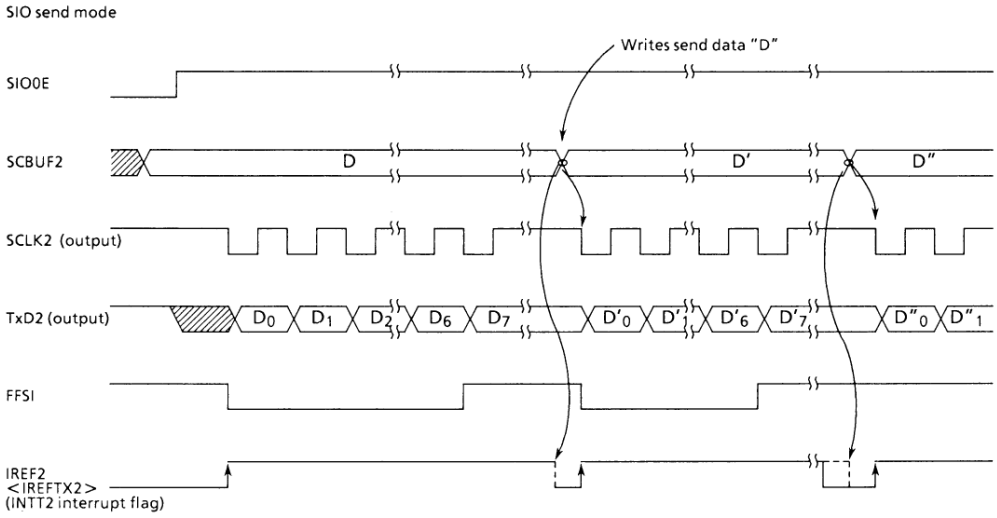
(External clock pulses)

In the internal clock operation, data must be stored in the buffer registers before the next data shift operation begins. The transfer speed in an interrupt service program is determined by the maximum delay time from the interrupt request generation to buffer register data write.

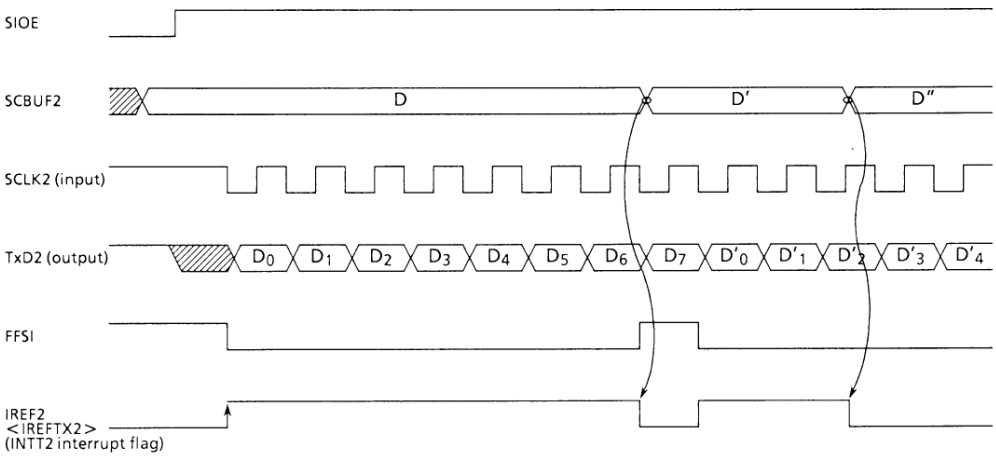
To end a send, the buffer empty interrupt service program disables (clears to "0") serial transfer control register SCMOD2 <SIO0E> instead of writing the next send data. When serial transfer control is disabled, the serial transfer ends when the send data now being shifted out is finished being sent.

The end of send can be determined by the status of serial transfer monitor flag SCMOD2 <FFSI>. In the external clock operation, the serial transfer control register SCMOD2 <SIOE> must be disabled before starting the next send data shift operation.

If the serial transfer control register SCMOD2 <SIOE> is not disabled before the shift operation begins, operations stop after sending the next send data (dummy).



(a) Internal clock operation (with wait operation) in send mode



(b) Internal clock operation (with wait operation) in send mode

Figure 3.8 (22). Chart of Serial Channel 0 Send Mode Timing

(2) Receive Mode

Setting the command register to receive mode, then setting serial serial transfer control SCMOD2 <SIOE> to enable makes receive possible. Shift data is synchronized with serial clock pulses and fetched from the RxD2 pin. When data is fetched, it is transferred from the shift register to the buffer register and the buffer-full interrupt INTRX2 is generated to request a read of receive data.

When the interrupt service program read the next receive data from the buffer register, the interrupt request signal is cleared. The following data continues to be fetched after the interrupt is generated.

After the interrupt request is cleared, data is transferred from the shift register to the buffer register when data is fetched.

(Internal clock pulses)

In the internal clock operation, if the previous receive data has not been read from the buffer regfister after the next data is fetched, the serial clock stops and waits until the previous data is read.

(External clock pulses)

In the external operation, shift operations are synchronized with externally supplied clock pulses. The data is read before the next receive data is transferred into the buffer register. If the previous data has not been read, the receive data will not be transferred into the buffer registers and all subsequently input receive data will be cancelled. The maximum transfer speed of the external clock operation is determined by the maximum delay time from interrupt request generation to receive data read.

Rising and falling edge shifts can be selected in the receive mode. Because data is fetched on the serial clock pulses's rising edge in the rising edge shift, the first shift data must already be input to the RxD2 pin when the initial serial clock pulses are applied at transfer start.

(3) Send-Receive Mode

The first send data is written into buffer registers SCBUF2 after the send-receive mode is set by the command register. Setting the serial transfer control register SCMOD2 <SIOE> to 1 enables receiving or sending data. Send data is output from the TxD2 pin on the rising edge of the serial clock pulse, while receive data is fetched from the RxD2 pin on the falling edge of the serial clock pulse.

When data is fetched, data is transferred from the shift registers to the buffer registers and buffer-full interrupt INTRX2 is genrated to request receive data read.

When the interrupt service program reads the next receive data from the buffer register, the interrupt request signal is cleared.

(Internal clock pulses)

In the internal clock operation, a wait begins until the received data is read and the next send data is written.

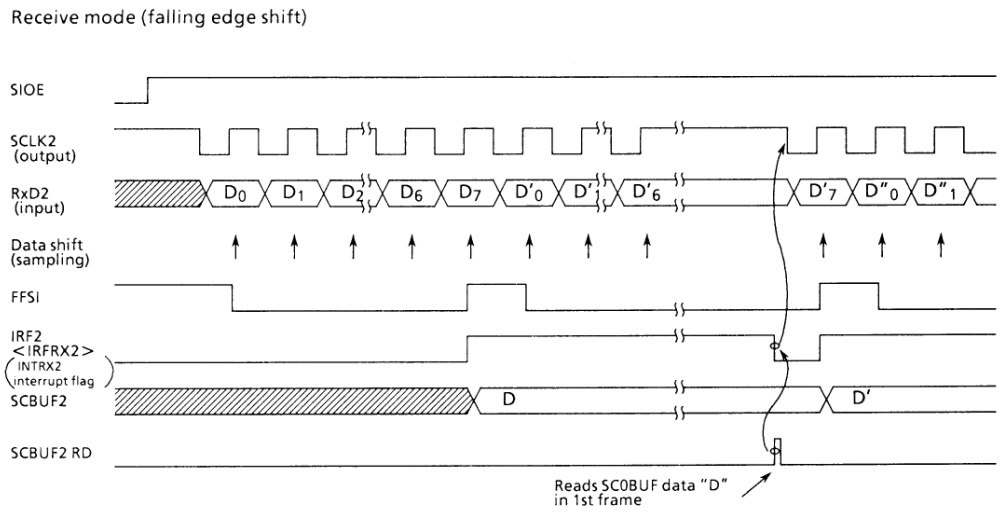
(External clock pulses)

In the external clock operation, the receive data must be read and the next send data written before starting the next shift operation, because the shift operation is synchronized with the external supplied clock pulses. The maximum transfer speed of the external clock operation is determined by the maximum delay time from interrupt request generation to send data fetch and receive data write.

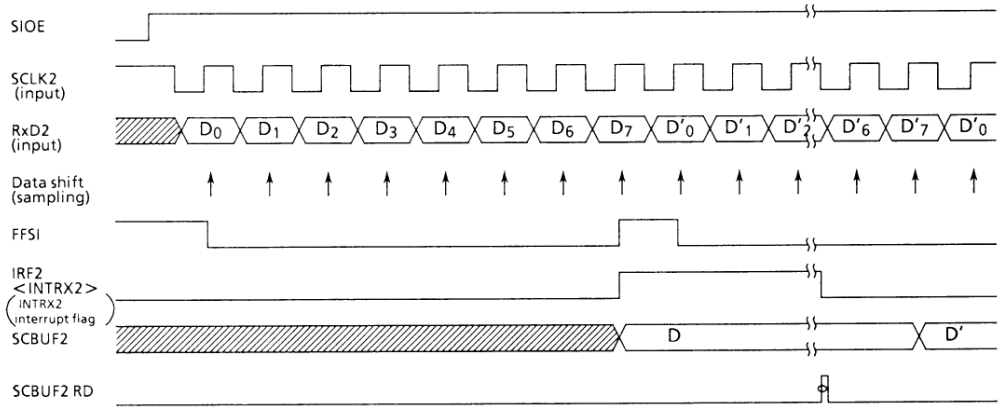
Because the same buffer registers are used for send and receive, always ensure that send data is written after 8 bits of receive data are fetched.

To end send-receive, disable the serial transfer control register. When the serial transfer control register is disabled, send-receive ends afetr receive data is organized and transferred to the buffer register.

The program checks the end of send-receive by reading serial transfer monitor flags SCMOD2 <FFSI>.

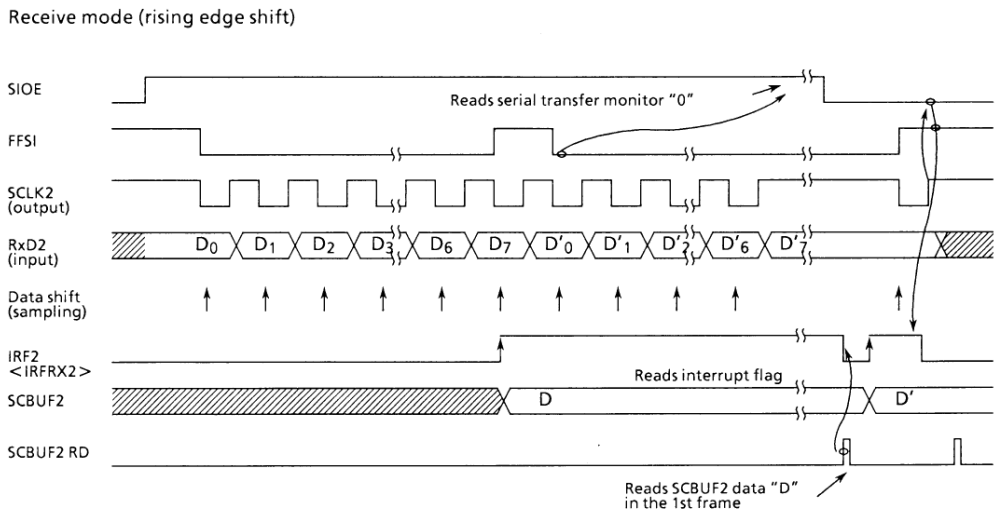


(a) Receive mode internal clock operation (with falling edge shift / wait)

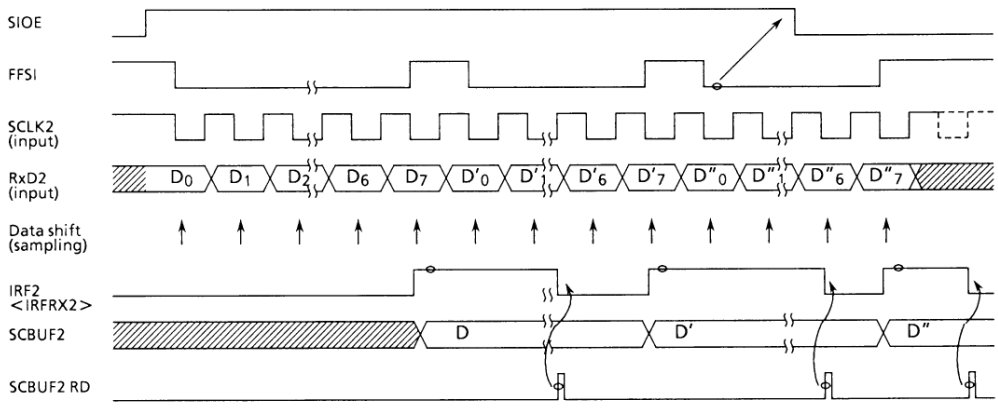


(b) Receive mode external clock operation (falling edge shift)

Figure 3.8 (23) - 1. Chart of Serial Channel ø Send-Receive Mode (Falling Edge Shift) Timing



(a) Receive mode internal clock operation (with rising edge shift / wait)



(b) Receive mode external clock operation (rising edge shift)

Figure 3.8 (23) - 2. Chart of Serial Channel ø Send-Receive Mode (Falling Edge Shift) Timing

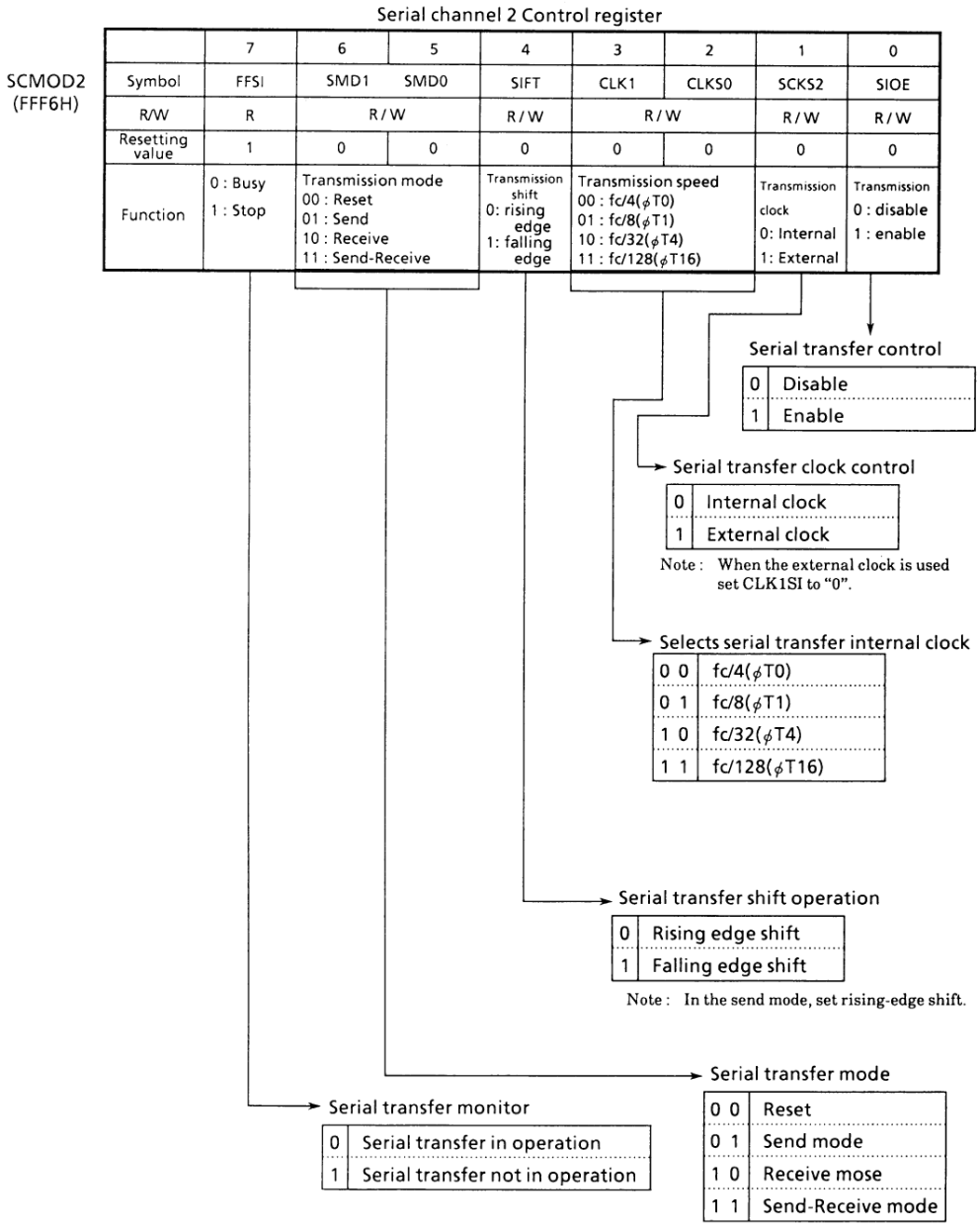


Figure 3.8 (24) - 1. Serial Channel Control Register

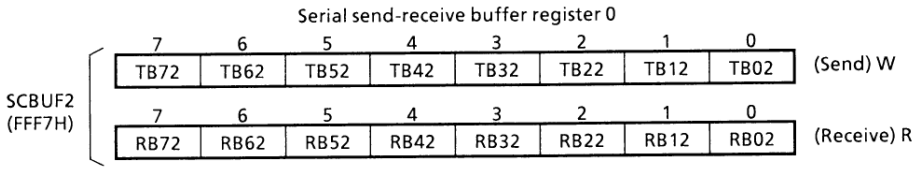


Figure 3.8 (24) - 2. Serial Channel Buffer Registers

3.9 Time Base Counter (TBC)

The TMP90CM36 has a 20-bit time base counter. This time base counter can select from the fundamental clock (fc) divided by 2, it divided by 3 or external input clock (EXIN).

The various outputs of this time base timer are used as reference signals for 24-bit capture and timing pulse generator (TPG).

The contents of the time base counter can be read with registers TBCD0 and TBCD1.

Interrupts can be generated from the TBC11 ~ TBC20 output signals of the time base counter.

When an INT TB interrupt has been generated, it is possible to determine whether the interrupt generation destination is INTTBC1 or INTTBC2 by reading CFREG <FTBC2,1>.

Table 3.9 (1) Time Base Counter and Cycle

TBC	1	2	3	4	5	6	7	8	9	10
(sec)	2^2 / fc	2^3 / fc	2^4 / fc	2^5 / fc	2^6 / fc	2^7 / fc	2^8 / fc	2^9 / fc	2^{10} / fc	2^{11} / fc
@fc = 12.5 MHz (μsec)	0.32	0.64	1.28	2.56	5.12	10.24	20.48	40.96	81.92	163.84

11	12	13	14	15	16	17	18	19	20
2^{12} / fc	2^{13} / fc	2^{14} / fc	2^{15} / fc	2^{16} / fc	2^{17} / fc	2^{18} / fc	2^{19} / fc	2^{20} / fc	2^{21} / fc
327.68	655.36	1310.72	2621.44	5242.88	10485.76	20971.52	41943.04	83886.08	167772.16

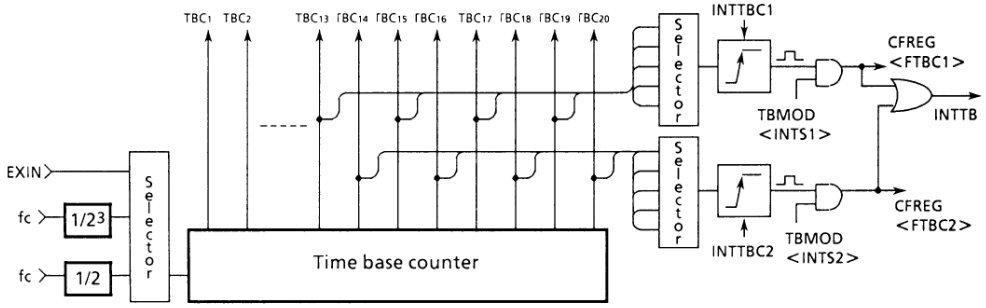


Figure 3.9 (1). Time Base Counter (TBC) Block Diagram

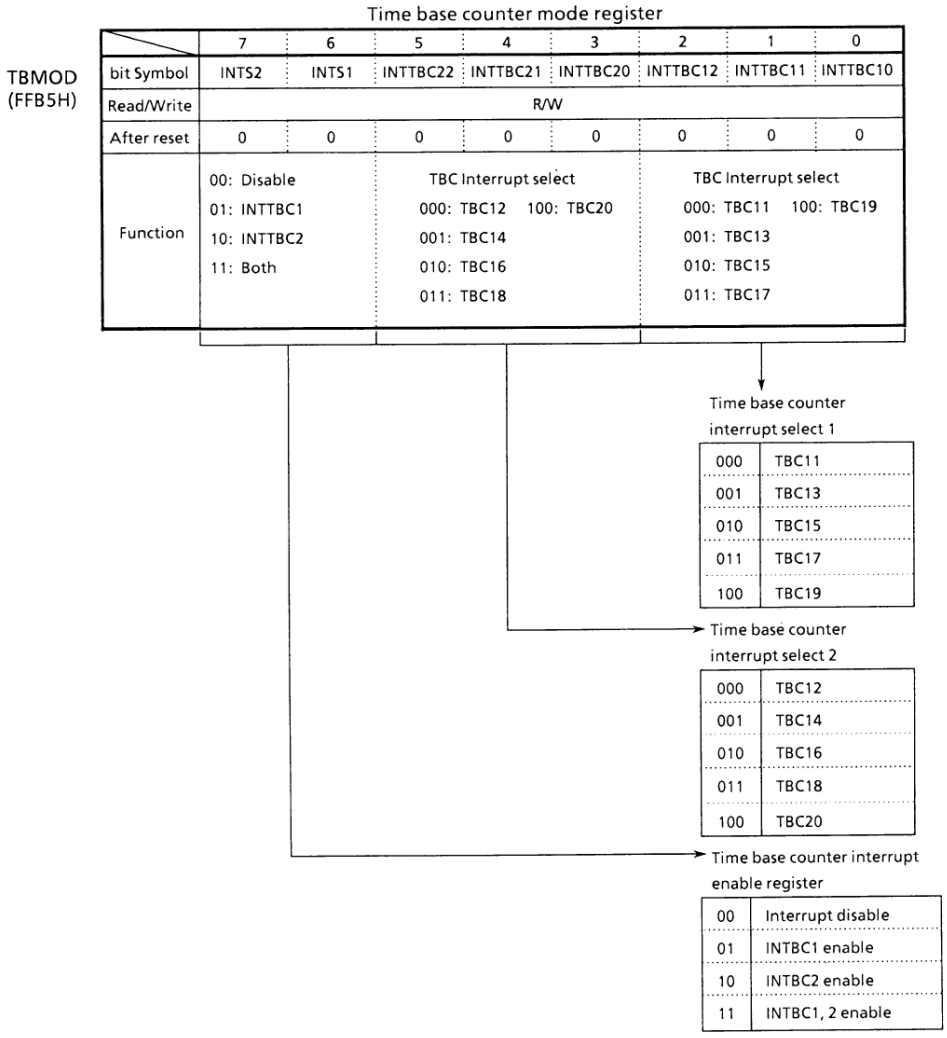


Figure 3.9 (2). Time Base Counter Mode Register (TBMOD)

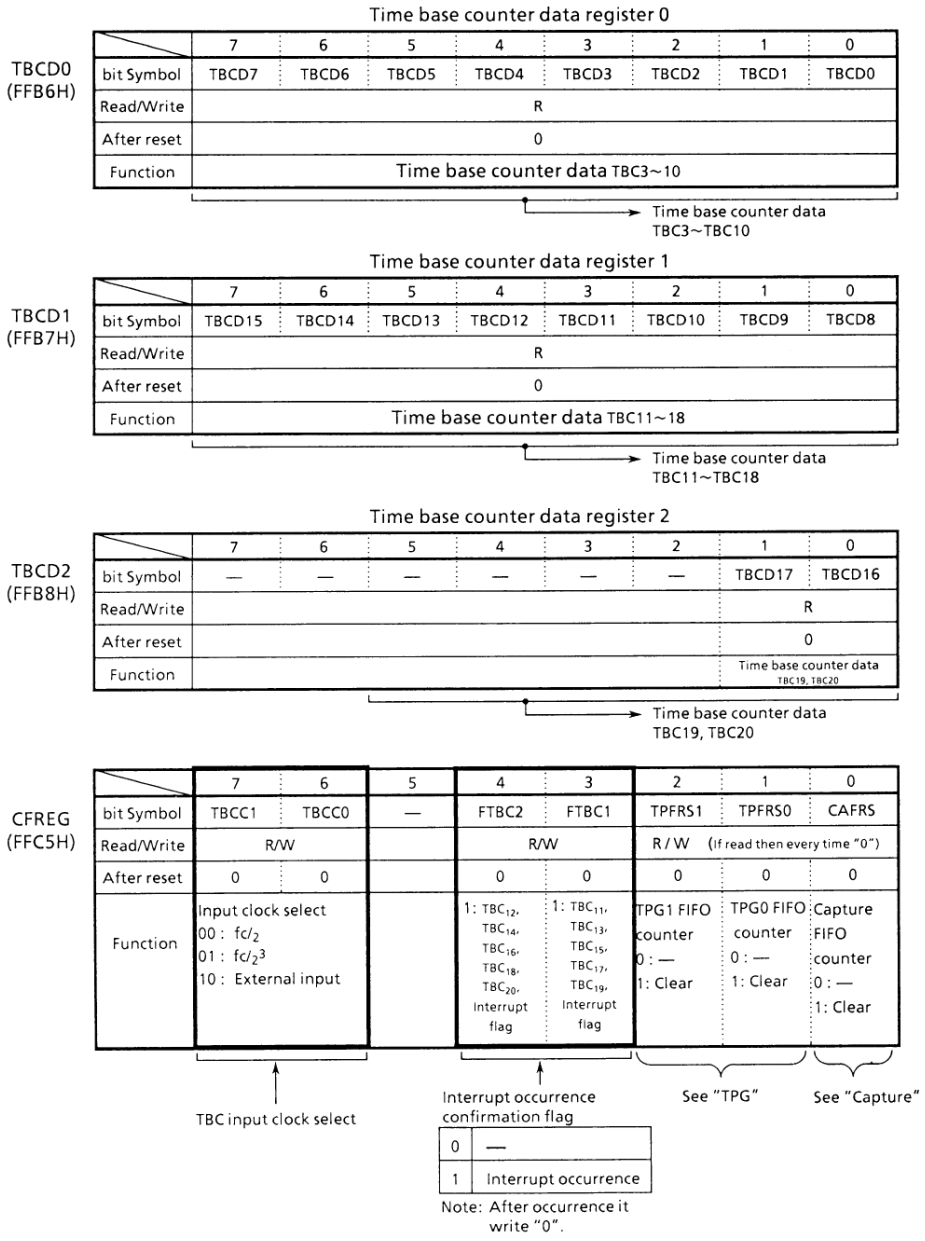


Figure 3.9 (3). Time Base Counter Related Register

3.10 Timing Pulse Generators (TPG)

The TMP90CM36 has a built-in 2 channels 24-bit x 4-stage timing pulse generation circuit with FIFO (TPG0, TPG1) that are used to control the various signals and mechanical parts.

Timing pulse generators (TPG) generate timing pulses that are synchronized with the time base counter (TBC) and which have an accuracy of 640ns (@ fc - 12.5MHz).

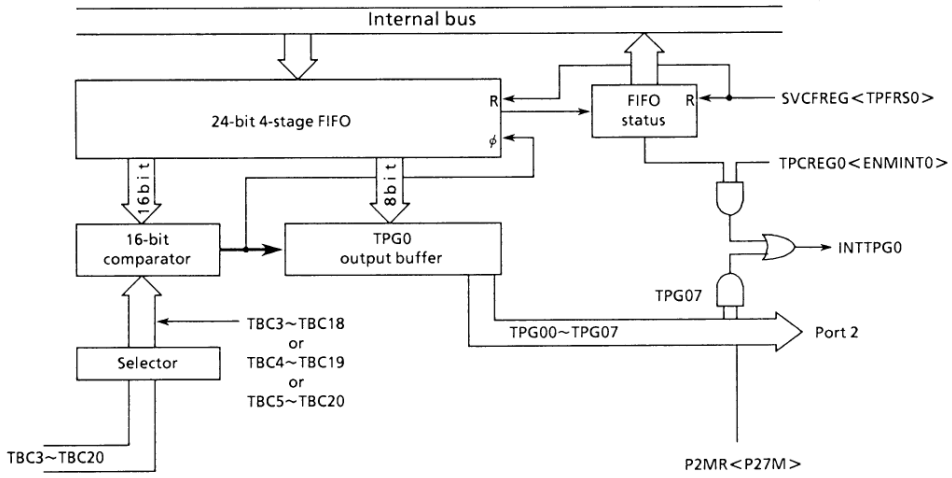


Figure 3.10 (1). TBG0 Block Diagram

3.10.1 Operation

© TPG0

1) Data

TPG0 has a total of 2 bits and 4 stages of FIFO: 6 bits in the output data register TPGREG0 and 16 bits in the comparator data registers TPGD00 and TPGD01.

The value written to the output data register is output when the values written to the comparator data registers and the TBC3 - TBC18, the TBC4 ~ TBC19 or the TBC5 ~ TBC20 values match.

Writing data to the TPGREG0 byte of the data registers increments the FIFO address. Thus, write data to memory addresses TPGD00, TPGD01, and TPGREG0, in that order.

2) FIFO

TPG0 has 4-stage FIFO (First In First Out) circuit. This FIFO generates an interrupt signal when the data becomes empty.

This interrupt signal is not generated after a reset or TP reset. Once data have been written to the data register and FIFO has been incremented, an interrupt is generated when data written to the output data register is output and FIFO becomes empty.

The interrupt is controlled with TPCREG0 <ENMINT0>.

3) TP Reset

The CFREG <TPFRS0> register resets TPG0. Writing "1" to this register resets the following TPG0 circuits.

① FIFO address counter

② FIFO status

③ Empty interrupt flag

Note: The TPG0 output buffer is not reset; therefore, the previous TP value is retained.
This buffer is cleared to "0", however, by system resets.

© TPG1

TPG1 operate the same TPG0.

3.10.2 Control Register

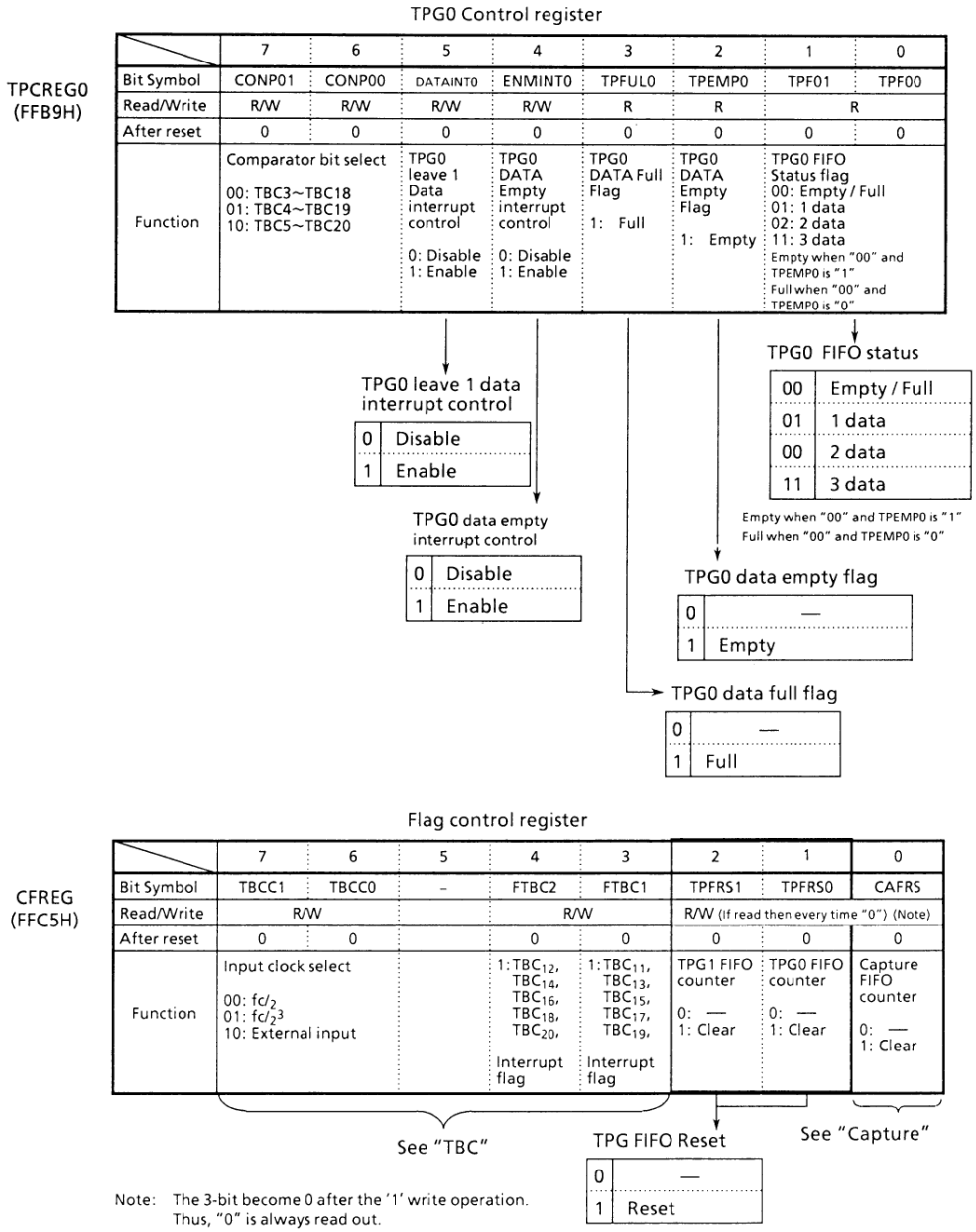


Figure 3.10 (2) - 1. TPG0, TGP1 Related Registers

TPG1 Control register

TPCREG1 (FFBDH)

	7	6	5	4	3	2	1	0
Bit Symbol	CONP11	CONP10	DATAINT1	ENMINT1	TPFUL1	TPEMP1	TPF11	TPF10
Read/Write	R/W	R/W	R/W	R/W	R	R	R	
After reset	0	0	0	0	0	0	0	0
Function	Comparator bit select 00: TBC3~TBC18 01: TBC4~TBC19 10: TBC5~TBC20		TPG1 leave 1 Data interrupt control 0: Disable 1: Enable	TPG1 DATA Empty interrupt control 0: Disable 1: Enable	TPG1 DATA Full Flag 1: Full	TPG1 DATA Empty Flag 1: Empty	TPG1 FIFO Status flag 00: Empty / Full 01: 1 data 02: 2 data 11: 3 data Empty when "00" and TPEMP0 is "1" Full when "00" and TPEMP0 is "0"	

TPG1 leave 1 data interrupt control

0	Disable
1	Enable

TPG1 data empty interrupt control

0	Disable
1	Enable

TPG1 FIFO status

00	Empty / Full
01	1 data
00	2 data
11	3 data

Empty when "00" and TPEMP0 is "1"
Full when "00" and TPEMP0 is "0"

TPG1 data empty flag

0	—
1	Empty

TPG1 data full flag

0	—
1	Full

Figure 3.10 (2) - 2. TPG0, TGP1 Related Registers

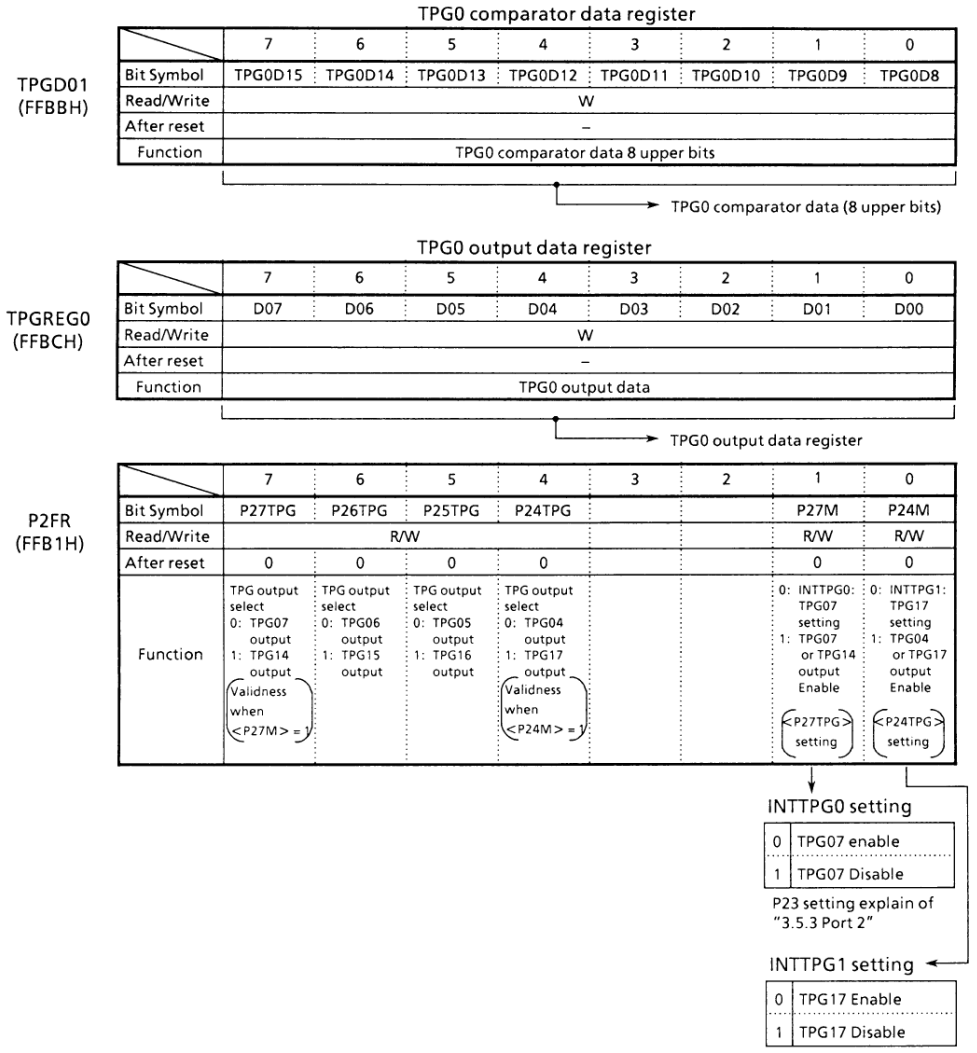


Figure 3.10 (2) - 3. TPG0, TGP1 Related Registers

TPG1 comparator data register

TPGD10 (FFBEH)

	7	6	5	4	3	2	1	0
Bit Symbol	TPG1D7	TPG1D6	TPG1D5	TPG1D4	TPG1D3	TPG1D2	TPG1D1	TPG1D0
Read/Write	W							
After reset	–							
Function	TPG1 comparator data 8 lower bits							

→ TPG1 comparator data (8 lower bits)

TPG1 comparator data register

TPGD11 (FFBFH)

	7	6	5	4	3	2	1	0
Bit Symbol	TPG1D15	TPG1D14	TPG1D13	TPG1D12	TPG1D11	TPG1D10	TPG1D9	TPG1D8
Read/Write	W							
After reset	–							
Function	TPG1 comparator data 8 upper bits							

→ TPG1 comparator data (8 upper bits)

TPG1 output data register

TPGREG1 (FFC0H)

	7	6	5	4	3	2	1	0
Bit Symbol	D17	D16	D15	D14	D13	D12	D11	D10
Read/Write	W							
After reset	–							
Function	TPG1 output data							

→ TPG1 output data register

Figure 3.10 (2) - 4. TPG0, TGP1 Related Registers

3.11 Capture Circuit

The TMP90CM36 has a RAM with 24-bits, 4-step FIFO to simplify various time measurements. The time (20-bit data from TBC1 - TBC20) is written in real time together with the input data in RAM by the servo input signals.

The latch is operated and INTCAP interrupts to the CPU are generated bu the input signals.

Figure 3.11 (1) is a block diagram of the capture circuit.

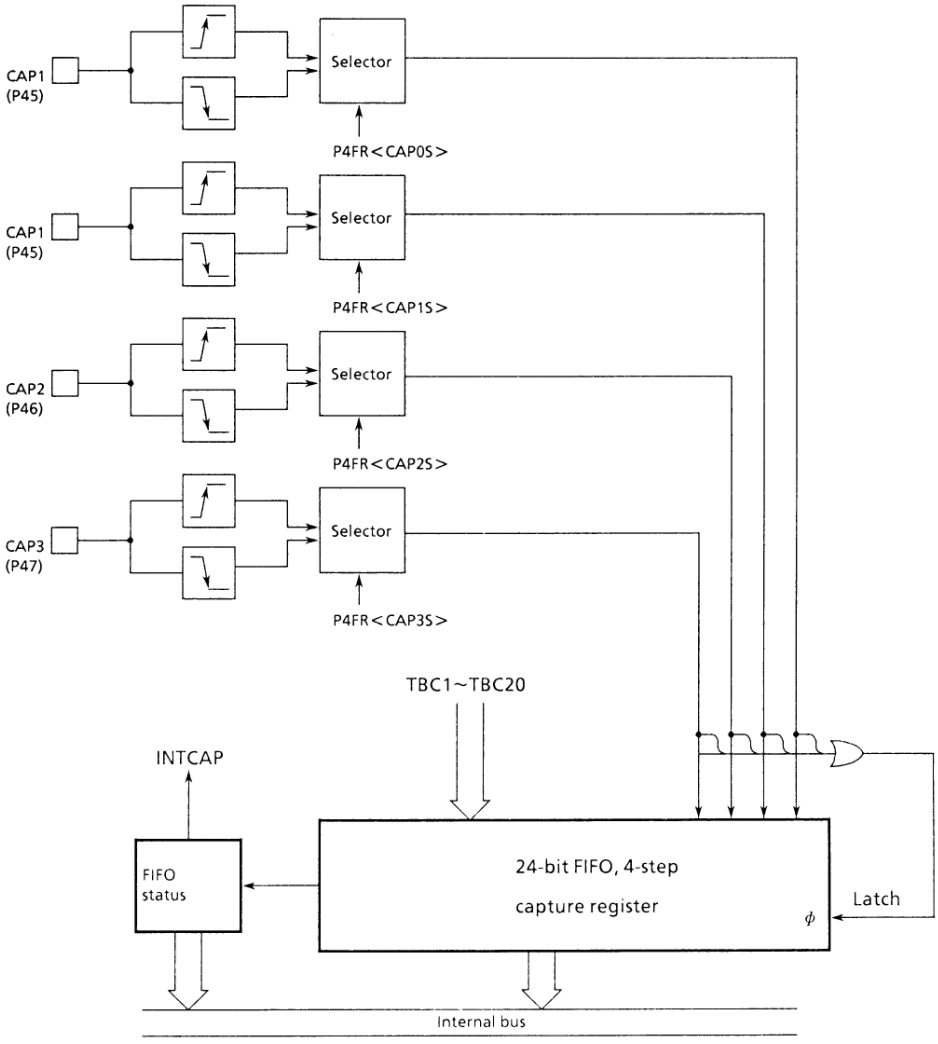


Figure 3.11 (1). Block Diagram of Capture Circuit

3.11.1 Explanation of Operations

1) Capture

There are 4 input sources for the capture circuit: CAP0, CAP1, CAP2, CAP3 (VP) signals. A total 24 bits, the 4 input signal bits and the 20-bit values from the time base counter (TBC1 - TBC20) to the 3 bytes at CAPREG0 ~ 2 are latched using the input signal edge (Select the rising-edge or the falling-edge).

The 24-bit data is obtained by reading the data* at the addresses in the sequence CAPREG0, CAPREG1 and CAPREG2.

Note: *This address must always be read last because reading the data at address CAPREG2 shifts the FIFO address.

2) FIFO

Because this capture circuit uses the FIFO (First In First Out) method, it always reads the data that is latched first. If the 4-step FIFO contents are full, the capture operation is disabled by the input signal. The FIFO status register (CAPFST) will be FFH when this happens.

When the CAPFST is 00H, FFH is read out when capture data is read out. Always read the capture data out when the FIFO status register is not “0” or after the INTCAP interrupt is generated.

3) CAP Reset

The capture circuit has a software reset in addition to its system reset.

Writing “1” into the CFPREG < CAFRS> resets the following circuits:

① FIFO address counter

② FIFO status

3.11.2 Control Registers

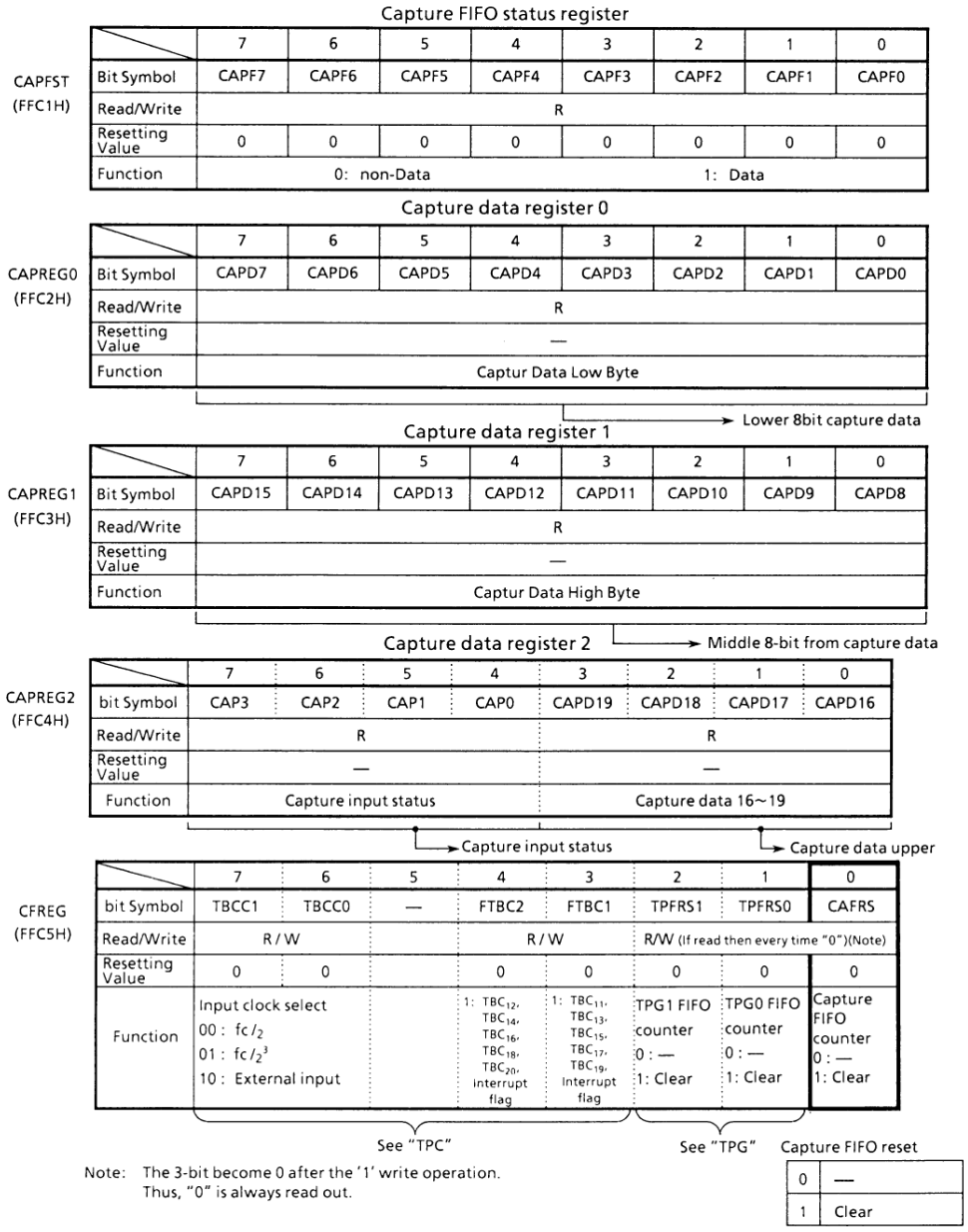


Figure 3.11 (2). Capture Related Register

3.12 PWM (Pulse Width Modulation) Output

The TMP90CM36 has 2 built-in channels for outputting pulse width modulation (PWM). D/A conversion output can be obtained by attaching an external low-pass filter for simple motor control.

One cycle in 12-bit resolution PWM output is TM = 212/(fc/2) [seconds]. Upper 7-bit PWM data latch data control pulse width for the carrier frequency in time TS (TS = TM/32)[seconds]. When the upper 7-bit data is n (n = 0 to 128), the low level pulse width is n*t (t = 2/(fc/2)) with TS as the cycle.

The lower 5-bit data controls the applied pulses of width t in “32” time periods TS (i) (i = 0 - 31) within the TM cycle. The low level pulse width is (n + 1) *t in the period in which applied pulses are output.

$$\left(\begin{array}{l}(\text{when } fc = 16.0\text{MHz}) \\ \quad T_M \fallingdotseq 1.953\text{kHz} \\ \quad T_S \fallingdotseq 62.5\text{kHz}\end{array}\right)$$

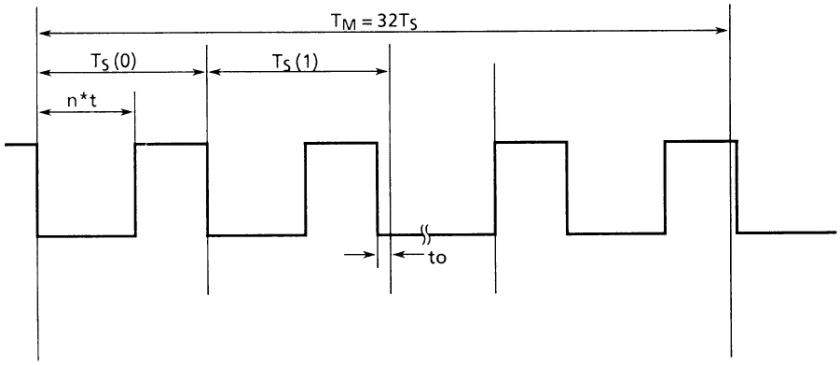


Figure 3.12 (1)

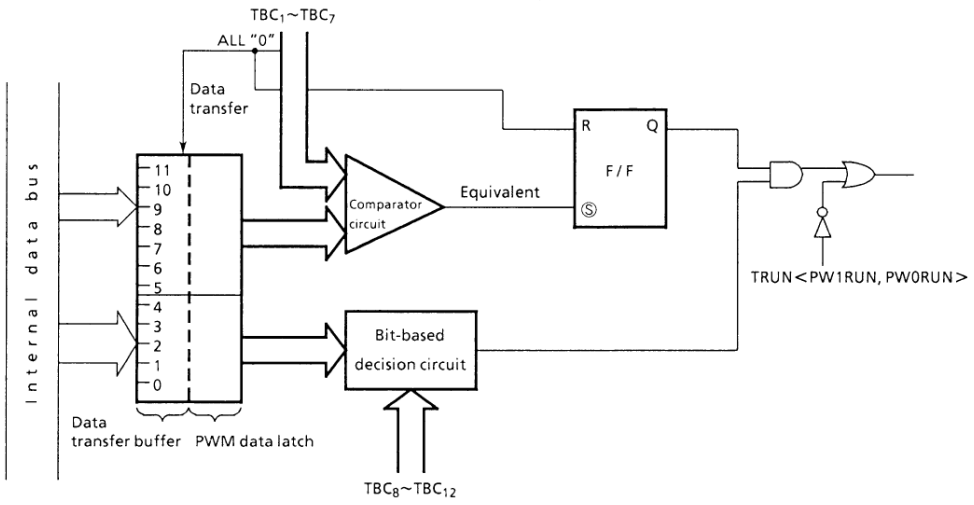


Figure 3.12 (2). Block Diagram of PW0 and PW1

3.12.1 Control Register

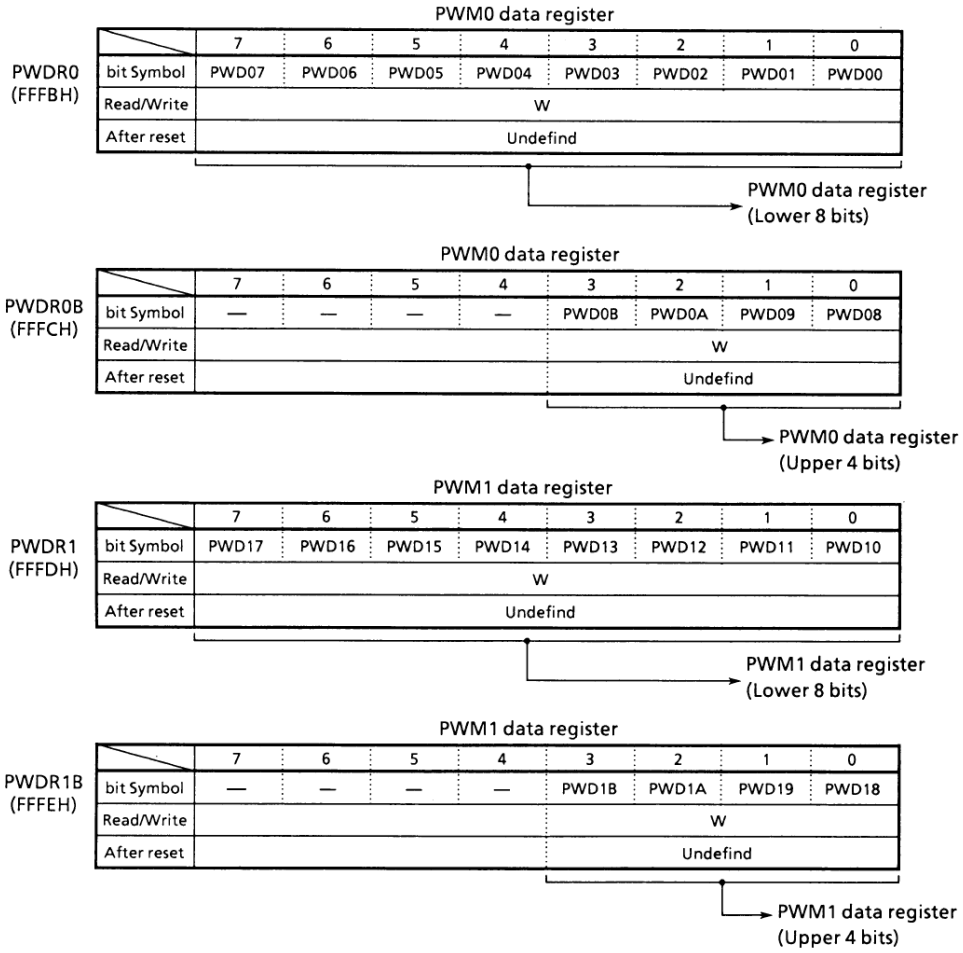


Figure 3.12 (3) - 1. PWM Related Register

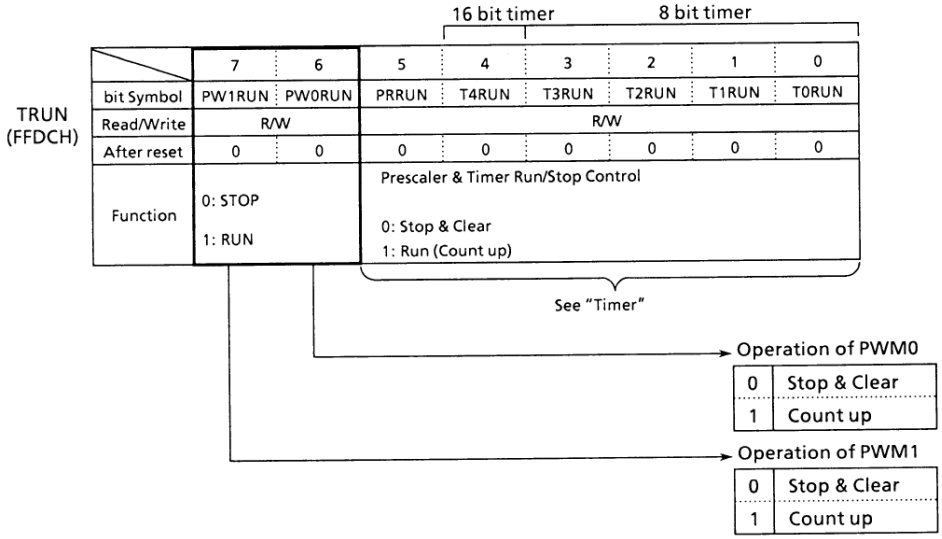


Figure 3.12 (3) - 2. PWM Related Register

3.13 Analog/Digital Converter

TMP90CM36 contains a high-speed, high-accuracy analog/digital converter (A/D converter) with 8-channel analog input that features 8-bit sequential comparison.

Figure 3.13 (1) shows the block diagram of the A/D converter. 8-channel analog input pins (AN7 to AN0) are shared by input-only port P6 and so can be used as input port.

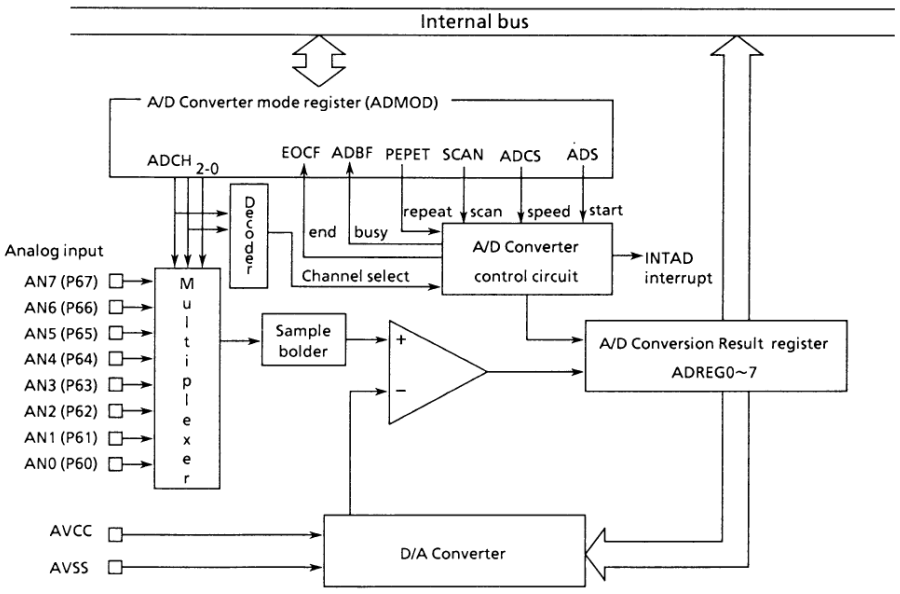


Figure 3.13 (1). Block Diagram of A/D Converter

3.13.1 Control Registers

ADMOD (FFD7H)

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

Analog input channel select (ADCH2 = 0)

	Nomal	Scan mode
00	AN0	AN0
01	AN1	AN0→AN1
10	AN2	AN0→AN1→AN2
11	AN3	AN0→AN1→AN2→AN3

Analog input channel select (ADCH2 = 1)

	Nomal	Scan mode
00	AN4	AN4
01	AN5	AN4→AN5
10	AN6	AN4→AN5→AN6
11	AN7	AN4→AN5→AN6→AN7

A/D Conversion start

0	——
1	Start A/D conversion

Note : Always "0"When data is read

A/D conversion speed selection

0	High speed conversion mode (95 states)
1	Low speed conversion mode (190 states)

Specification of A/D scan mode

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

Selecting A/D repeat mode

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

A/D conversion busy flag

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

A/D conversion end flag

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

Figure 3.13 (2). A/D Conversion Mode Register (ADMOD)

ADREG4 (FFD8H)

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	R							
After reset	Undefined							
Function	A/D conversion rsult of "AN0" or "AN4" is stored.							

A/D conversion result register for AN0 (Read Only)

ADREG15 (FFD9H)

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	R							
After reset	Undefined							
Function	A/D conversion rsult of "AN1" or "AN5" is stored.							

conversion result register for AN1 (Read Only)

ADREG26 (FFDAH)

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	R							
After reset	Undefined							
Function	A/D conversion rsult of "AN2" or "AN6" is stored.							

conversion result register for AN2 (Read Only)

ADREG37 (FFDBH)

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	R							
After reset	Undefined							
Function	A/D conversion rsult of "AN3" or "AN7" is stored.							

A/D conversion result register for AN3 (Read Only)

Figure 3.13 (3). A/D Conversion Result Register (ADREG0 ~ 3)

ADCH (FFD6H)

	7	6	5	4	3	2	1	0
bit Symbol	—	—	—	—	—	—	—	ADCH2
Read/Write								R/W
After reset								0
Function								0 : AN0~3 1 : AN4~7

Analog input channel select scan mode

	Nomal	Scan mode
0	AN0~AN3	AN0→AN1→AN2→AN3
1	AN4~AN7	AN4→AN5→AN6→AN7

Figure 3.13 (4). A/D Converter Channel Select Register

3.13. 2 Operation

(1) Analog Reference Voltage

High analog reference voltage is applied to the AVCC pin, and the low analog voltage is applied to AVSS pin.

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

(2) Analog Input Channels

Analog input channel is selected by ADMOD <ADCH1,0>, ADMOD <ADCH2>. However, which channel to select depends on the operation mode of the A/D converter.

In fixed analog input mode, one channel is selected by ADMOD <ADCH1,0>, ADMOD <ADCH2> among three pins: AN0 to AN3.

In analog input channel scan mode, the number of channels to be scanned from AN0 is specified by ADMOD <ADCH1,0>, ADMOD <ADCH2>, such as AN0 ⇨ AN1, AN0 ⇨ AN1 ⇨ AN2, and AN0 ⇨ AN1 ⇨ AN2 ⇨ AN3 or the number of channels to be scanned from AN4 is such as AN4 ⇨ AN5, AN4 ⇨ AN5 ⇨ AN6, AN4 ⇨ AN5 ⇨ AN6 ⇨ AN7.

When reset, A/D conversion channel register will be initialized to ADMOD <ADCH1,0> = 00, ADCH <ADCH2> = 0 so that AN0 pin will be selected.

The pins which are not used as analog input channel can be used as ordinary input port P5.

(3) Starting A/D Conversion

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

(4) A/D Conversion Mode

Both fixed A/D conversion channel mode and A/D conversion channel scan mode have two conversion modes, i.e., single and repeat conversion modes.

In fixed channel repeat mode, conversion of specified one channel is executed repeatedly.

In scan repeat mode, scanning from AN0 ···⇨ AN3 or from AN4, ···⇨ AN7 is executed repeatedly.

A/D conversion mode is selected by ADMOD <REPET, SCAN>.

(5) A/D Conversion Speed Selection

There are two A/D conversion speed modes: high speed mode and low speed mode. The selection is executed by ADMOD <ADCS> register.

When reset, ADMOD <ADCS> will be initialized to “0”, so that high speed conversion mode will be selected.

Main setting

```
INTE0 ← - - - 1 - - - -
ADMOD ← X X 0 0 0 1 1 1
ADCH  ← X X X X X X X 0
```

Enable INTAD.
Specify AN3 pin as an analog input channel and starts A/D conversion in high speed mode.

INTAD routine

```
A       ← ADREG37
(FF10H)← A
```

Read the value of ADREG37 into the accumulator and store the value of the accumulator in the memory at FF10H.

② When the analog input voltage of AN0~AN2 pins is kept A/D converted in high speed conversion channel scan repeat mode

```
INTE0 ← X - 0 - - - - -
ADCH  ← X X X X X X X 0
ADMOD ← X X 1 1 0 1 1 0
```

Disable INTAD.
Start the A/D conversion of analog input channels AN0~AN2 in the high-speed scan repeat mode.

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

(6) A/D Conversion End and Interrupt

- A/D conversion single mode

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

Interrupt requesting flip-flop is cleared only by resetting operation or reading the A/D conversion result storing register and cannot be cleared by instruction.

- A/D conversion repeat mode

For both fixed conversion mode and conversion channel scan mode, INTAD will be disabled when in repeat mode. Always leave the INTE0 <ADIS> flag at "0".

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

(7) Storing the A/D Conversion Result

The results of A/D conversion are stored ADREG04 to ADREG37 registers for each channel. In repeat mode, the registers are updated whenever conversion ends.

ADREG04 to ADREG37 are read-only registers.

(8) Reading the A/D Conversion Result

The results of A/D conversion are stored ADREG04 to ADREG37 registers. When the contents of one of ADREG0 to ADREG3 registers are read, ADMOD <EOCF> will be cleared to "0".

Setting example: ① When the analog input voltage of the AN3 pin pin is A/D converted and the results are read in the memory at FF10H by A/D interrupt INTAD routine.

3.14 8-Bit Power Supply Voltage Model D/A Converter

The TMP90CM36 contains 8-bit resolution D/A voltage with the following merit.

- TMP90CM36 contains 2 channel in the 8-bit resolution D/A converter (R-2R formula)
- Analog voltage V which done output is decided setting values in register DAREG0, DAREG1.

Figure 3.14 (1) shows the block diagram of D/A converter.

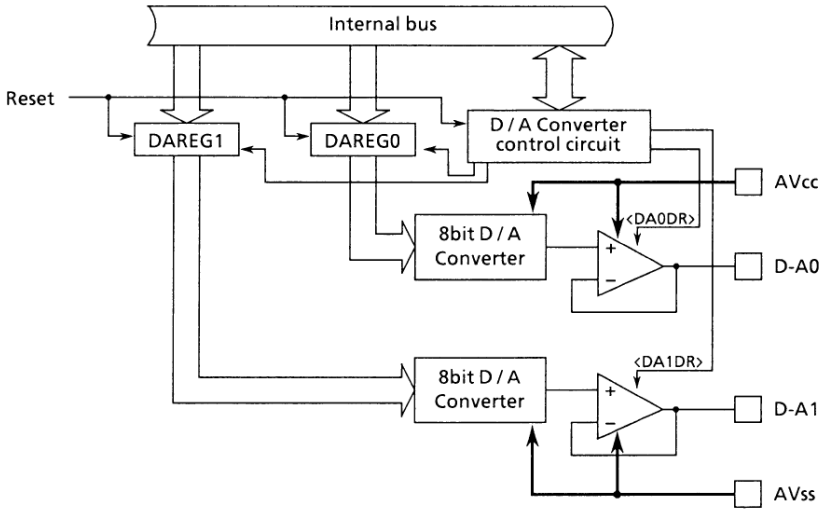


Figure 3.14 (1). D/A Converter Block Diagram

D / A Converter drive register

		7	6	5	4	3	2	1	0
DADRV (FFF8H)	bit Symbol	—	—	—	—	—	—	DA1DR	DA0DR
	Read/Write							R / W	
	After reset							0	
	Function							0 : 0V output 1 : Register values conversion	

D / A OUTPUT

0	0V output
1	Register values conversion output

D / A Converter conversion register 0

		7	6	5	4	3	2	1	0
DAREG0 (FFF9H)	bit Symbol	—							
	Read/Write	W							
	After reset	0							
	Function	Register write : D/A conversion start → D/A-0 output							

D / A Converter conversion register 1

		7	6	5	4	3	2	1	0
DAREG1 (FFFAH)	bit Symbol	—							
	Read/Write	W							
	After reset	0							
	Function	Register write : D/A conversion start → D/A-1 output							

This register become D/A converter digital input data.
Relations of register values and output voltage.
Output V = AVCC × N/256 (N = register values)

Note : After release in HALT order, it is output of voltage values in formerly HALT. Also HALT mode except stop mode continues to be output of conversion voltage in register values without relation HALT order.

Figure 3.14 (2). D/A Converter Related Register

3.14.1 Operation

If built-in D/A converter is D/A converter drive register DADRV <DA1DR, DA0DR> to “1”, D/A converter conversion register DAREG1, DAREG0 changes values from digital to analog, and then it is output in conversion voltage from D-A1, D-A0 pin. Relate input data with output voltage shows Figure 3.14 (2).

Reset operations is reset <DA1DR, DA0DR> to “0”, from D-A1, D-A0 pin is output “0V” also both DAREG1 and DAREG0 is reset “00H”, therefore, if it set DADRV to “1” after reset, it will output “AVCC/4” (Explanation of Figure 3.14). When D/A converter is in use, register write DADRV to “1” the using channel. Future, if it write input data in the DAREG, it is output applicability analog values.

Moreover, if it is practice HALT order after nomination in the STOP mode (WDMOD <HALTM1, 0> = 0,1), from D-A0, D-A1 pin without relation values DADRV, DAREG is output “0V”.

3.15 Watchdog Timer (Looping Detection Timer)

The purpose of the watchdog timer (WDT) is to detect the start of CPU misoperation due to noise, etc., and bring it back to normal.

3.15.1 Configuration

The TMP90CM36F multiplexes the watchdog timer output ($\overline{\text{WDTOUT}}$) and P80 (pin 30). P80 (output port) is switched to the $\overline{\text{WDTOUT}}$ pin and $\overline{\text{RESET}}$ is returned inside the chip by setting bit WDMOD <RESCR> = "1" of the watchdog timer mode register at address #FFD0H to "1".

Figure 3.15 (1) shows the WDT block diagram.

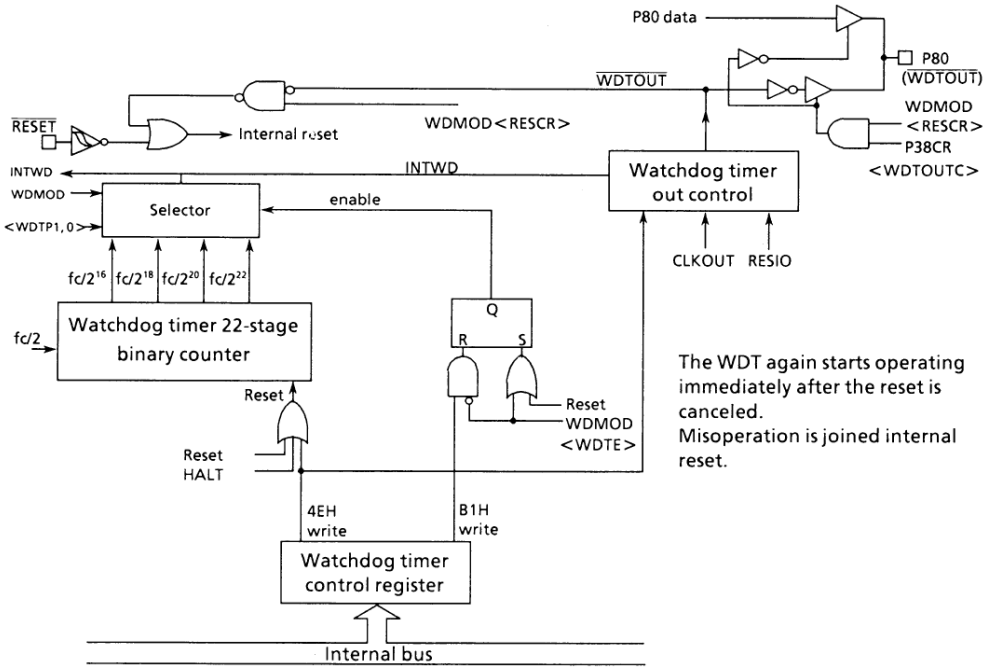


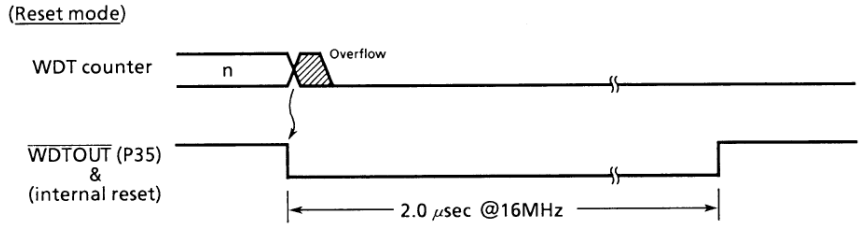
Figure 3.15 (1). Watchdog Timer Block Diagram

The watchdog timer is a 22-stage binary counter that uses (fc/2) as the input clock.

The binary counter outputs are 2^{16}/fc, 2^{18}/fc, 2^{20}/fc and 2^{22}/fc. One of these outputs is used for watchdog timer output $\overline{\text{WDTOUT}}$.

$\overline{\text{WDTOUT}}$ outputs "0" to reset the peripheral devices when the watchdog timer overflows.

$\overline{\text{WDTOUT}}$ also is connected to RESET inside the TMP90CM36. In this case, $\overline{\text{WDTOUT}}$ outputs "0" in a 32/fxtal = 2.0μsec (fxtal = 16MHz) cycle and simultaneously resets the TMP90CM36.



3.15.2 Control Registers

The watchdog timer (WDT) is controlled by two control registers (WDMODE and WDCR).

(1) Watchdog Timer Mode Register (WDMOD

① Watchdog timer detection time setting (WDTP)

This is a 2-bit flag used to set the watchdog timer interrupt time for looping (runaway) detection. This flag is initialized to WDMOD <WDTP0, 1> = 00 by resets, which results in a value of 216/fc [sec]. (The number of states is approximately 32.768.)

② Watchdog timer enable/disable control (WDTE)

This bit is initialized to WDTE = 1 by resets, which enables the watchdog timer.

To disable the watchdog timer, it is only necessary to clear this bit to "0" and write the disable code (B1H) to the WDCR register. It is difficult for the watchdog timer to be disabled by looping.

To disable the watchdog timer after it has been enabled, it is only necessary to write "1" to the <WDTE> bit.

③ Watchdog timer out reset connection (RESCR)

This flag is used to set whether or not the TMP90CM36 will be reset when looping is detected and whether or not to output $\overline{\text{WDTOUT}}$.

<RESCR> is set to "1" by reset operations; therefore, pin 30 is set as the $\overline{\text{WDTOUT}}$ pin and connected internally to the RESET pin.

Pin 30 can be set as either the $\overline{\text{WDTOUT}}$ pin or port pin by overwriting <RESCR>. However, caution is required because a redundant structure is used to prevent misoperation. The <RESCR> bit is linked to the P38CR <WDTOUTC> therefore, it is always necessary to write "1" the <WDTOUTC> when the <RESCR> bit is overwritten.

The <RESCR> bit is set only after "1" is written to the <WDTOUTC> and then either "0" or "1" is written to the RESCR bit.

Writing to the <RESCR> bit automatically clears the <WDTOUTC> to "0"; therefore, when resetting the <RESCR> bit, again write "1" to the <WDTOUTC>.

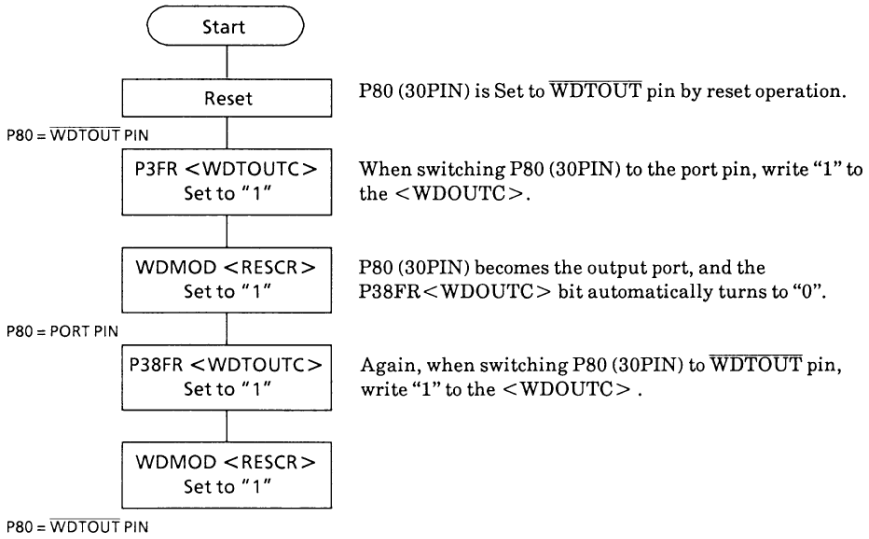


Figure 3.15 (2). Flowchart of P80/$\overline{\text{WDTOUT}}$ Pin Switching

WDMOD (FFECH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRIVE
Read/Write	R/W							
Resetting Value	1	0	0	0	0	0	0	0
Function	1: WDT Enable	WDT Detection time 00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warming Up time 0: 2^{14}/fc 1: 2^{16}/fc	Standby mode 00: RUN 01: STOP 10: IDLE 11: –		Watchdog timer out control 1: Pin 30 is set as the WDTOUT pin and is conntected internally to the RESET pin.	1: STOP Drive the pin even in the STOP mode.

Note : Write a data to this register (WDMOD) and, it can need the correct value from this register.

See "3.4.4 STOP mode."

Watchdog timer output control

0	P80 (30PIN) is set as the output pin.
1	$\overline{WDTOUT}$ is connected to $\overline{RESET}$ and P80 (pin 30) is set as the $\overline{WDTOUT}$ output pin.

Standby mode selection using the HALT instruction

00	RUN mode
01	STOP mode
10	IDLE mode
11	–

Warming up time selection when the STOP mode is reset

0	2^{14}/fc (approx. 1.3ms : @12.5MHz)
1	2^{16}/fc (approx. 5.2ms : @12.5MHz)

Watchdog timer detection time selection

00	2^{16}/fc (approx. 5.2ms : @12.5MHz)
01	2^{18}/fc (approx. 20.9ms : @12.5MHz)
10	2^{20}/fc (approx. 83.8ms : @12.5MHz)
11	2^{22}/fc (approx. 335.5ms : @12.5MHz)

Watchdog timer disable/enable control

0	Disable
1	Enable

Note: To disable the watchdog timer, it is necessary to also write the disable code to the $\overline{WDTCR}$ register. Disable is not possible by writing to this register alone.

Figure 3.15 (3). Watchdog Timer Mode Register

(2) Watchdog Timer Control Register (WDCR)

This register enables and disables the watchdog timer, and clears the binary counter.

- Disable control

The watch timer is disabled by clearing WDMOD <WDTE> to "0" and then writing the disable code (B1H) to the WDCR register.

```
WDMOD ← 0 - - - - - X X     Clears <WDTE> to "0"
WDCR  ← 1 0 1 1 0 0 0 1     Writes the disable code (B1H)
```

- Enable control

Set WDMOD <WDTE> to "1".

- Binary counter clear control

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

The binary counter is cleared and restarts counting when the clear code (4EH) is written to WDCR.

WDCR (FFEDH)

	7	6	5	4	3	2	1	0
bit Symbol	–							
Read/Write	W							
Resetting Value	–							
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 3.15 (4). Watchdog Timer Control Register

3.15.3 Operation

The watchdog timer is a timer that outputs "0" level from the watchdog timer output pin ($\overline{\text{WDTOUT}}$) after the detection time set with WDMOD <WDRP1, 0> elapses. The watchdog timer binary counter is cleared to "0" before an overflow occurs. If the CPU misoperates (loops) due to noise, etc., the binary counter will overflow unless the parity counter clear instruction is executed. The CPU can be returned to normal operation by resetting internally. A reset can be applied to both the TMP90CM36 and CPU by connecting the $\overline{\text{WDTOUT}}$ pin to the RESET pins of the peripheral devices.

The watchdog timer again starts operating immediately after the reset is canceled.

The watchdog timer stops during the IDLE mode and STOP mode and operates during the RUN mode. The watchdog timer can also be disabled when entering the RUN mode.

Example: ① Clears the binary counter.

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

② Sets the watchdog timer detection time to 2^{18}/fc

```
WDMOD ← 1 0 1 - - - X X
```

③ Disables 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 Sets the IDLE2 mode.

```
WDMOD ← 0 - - - 1 1 X X   Disables WDT and sets the IDLE mode.
WDCR  ← 1 0 1 1 0 0 0 1   Sets the standby mode.
```

Excute HALT instruction.

⑤ Sets the STOP mode (warmup time = 2^{16}/fc)

```
WDMOD ← - - - 1 0 1 X X   Sets the STOP mode.
```

Excute HALT instruction. Sets the standby mode.

4. Electrical Characteristics

TMP90CM36F/TMP90CM36T

4.1 Maximum Ratings

Symbol	Item	Rating	Unit
V_{CC}	Power supply voltage	-0.5 ~ + 7	V
V_{IN}	Input voltage	-0.5 ~ V_{CC} + 0.5	V
Σ_{IOL}	Output current (Total)	100	mA
Σ_{IOH}	Output current (Total)	-70	mA
P_D	Power dissipation (Ta = 85°C)	F 500	mW
		T 600	
T_{SOLDER}	Soldering temperature (10sec)	260	
T_{STG}	Storage temperature	-65 ~ 150	°C
T_{OPR}	Operating temperature	-20 ~ 70	°C

4.2 DC Electrical Characteristics

V_{CC} = 5V ± 10%, TA = -20 ~ 70°C (1 ~ 16MHz)
Typical Values are for TA = 25°C, V_{CC} = 5V.

Symbol	Item	Min	Max	Unit	Conditions
V_{IL}	Input Low Voltage (P0)	-0.3	0.8	V	–
V_{IL1}	P1, P2, P3, P4 , P5, P5, P6, P7	-0.3	$0.3V_{CC}$	V	–
V_{IL2}	$\overline{RESET}$, P81 (INT0), P82 ($\overline{STBY}$)	-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, P2, P3, P4 , P5, P5, P6, P7	$0.7V_{CC}$	V_{CC} + 0.3	V	–
V_{IH2}	$\overline{RESET}$, P81 (INT0), P82 ($\overline{STBY}$)	$0.75V_{CC}$	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_{DAR}	Darlington Drive Current (8 I/O Pins max)	-1.0	-3.5	mA	V_{EXT} = 1.5V R_{EXT} = 1.1kΩ
I_{LI}	Input Leakage Current	0.02 (Typ)	±5	μA	0.0 ≤ Vin ≤ V_{CC}
I_{LO}	Output Leakage Current	0.05 (Typ)	±10	μA	0.2 ≤ Vin ≤ V_{CC} - 0.2
I_{CC}	Operating Current (RUN) Idle	35 (Typ) 1.5 (Typ)	50 5	mA mA	t_{osc} = 16MHz
	STOP (TA = -20 ~ 70°C) STOP (TA = 0 ~ 50°C)	0.2 (Typ)	40 10	μA μA	0.2 ≤ Vin ≤ V_{CC} - 0.2
V_{STOP}	Power Down Voltage of (@STOP) (RAM back up)	2.0	6.0	V	V_{IL1} = 0.2Vcc, V_{IL2} = 0.8Vcc
R_{RST}	$\overline{RESET}$ Pull Up Register	50	150	KΩ	–
CIO	Pin Capacitance	–	10	pF	testfreq = 1MHz
V_{TH}	Schmitt width ($\overline{RESET}$, P81, P82)	0.4	1.0 (Typ)	V	–

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

4.3 AC Electrical Characteristics

V_{CC} = 5V ± 10% TA = -20 ~ 70°C (1 ~ 16MHz)

Symbol	Item	Variable		12.5MHz Clock		16MHz Clock		Unit
		Min	Max	Min	Max	Min	Max	
t_{OSC}	Oscillation cycle (= X)	62.5	1000	80	–	62.5	–	ns
t_{AL}	A0 ~ A7 effective address→ALE fall	0.5x - 15	–	25	–	16	–	ns
t_{LA}	ALE fall→A0 ~ A7 hold	0.5x - 15	–	25	–	16	–	ns
t_{LL}	ALE pulse width	x - 40	–	40	–	23	–	ns
t_{LC}	ALE fall→$\overline{RD}$/$\overline{WR}$ fall	0.5x - 30	–	10	–	1	–	ns
t_{CL}	$\overline{RD}$/$\overline{WR}$ rise→ALE rise	0.5x - 20	–	20	–	11	–	ns
t_{ACL}	A0 ~ A7 effective address→$\overline{RD}$/$\overline{WR}$ fall	x - 25	–	55	–	38	–	ns
t_{ACH}	Upper effective address→$\overline{RD}$/$\overline{WR}$ fall	1.5x - 50	–	70	–	44	–	ns
t_{CA}	$\overline{RD}$/$\overline{WR}$ fall→Upper address hold	0.5x - 20	–	20	–	11	–	ns
t_{ADL}	A0 ~ A7 effective address→Effective data input	–	3.0x - 35	–	205	–	153	ns
t_{ADH}	Upper effective address→Effective data input	–	3.5x - 55	–	225	–	164	ns
t_{RD}	$\overline{RD}$ fall→Effective data input	–	2.0x - 50	–	110	–	75	ns
t_{RR}	$\overline{RD}$ Pulse width	2.0x - 40	–	120	–	85	–	ns
t_{HR}	$\overline{RD}$ rise→Data hold	0	–	0	–	0	–	ns
t_{RAE}	$\overline{RD}$ rise→Address enable	x - 15	–	65	–	48	–	ns
t_{WW}	$\overline{WR}$ pulse width	2.0x - 40	–	120	–	85	–	ns
t_{DW}	Effective data→$\overline{WR}$ rise	2.0x - 50	–	100	–	65	–	ns
t_{WD}	$\overline{WR}$ rise→Effective data hold	0.5x - 10	–	30	–	21	–	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 (D0 ~ D7)
  High 0.8V_{CC}/Low 0.2V_{CC} (excluding AD0 ~ AD7)

4.4 A/D Conversion Characteristics

V_{CC} = 5V ± 10% TA = -20 ~ 70°C
f = 1 ~ 16MHz

Symbol	Parameter	Condition	Min	Max	Unit
AVCC	Analog reference voltage	Vcc - 1.5	Vcc	Vcc	V
A_{GND}	Analog reference voltage	Vss	Vss	Vss	
V_{AIN}	Analog input voltage range	Vss	–	Vcc	
I_{REF}	Analog reference voltage power supply current	–	0.5	1.0	mA
Error (Quantize error of ±0.5 LSB not included)	Total error (TA = 25°C, Vcc = V_{REF} = 5.0V)	–	–	1.0	LSB
	Total error	–	–	2.5	

4.5 Timer/Counter Input Clock (TI2, TI4)

V_{CC} = 5V ± 10% TA = -20 ~ 70°C
f = 1 ~ 16MHz

Symbol	Item	Variable		12.5MHz Clock		16MHz Clock		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	Clock cycle	8x + 100	–	740	–	600	–	ns
t_{VCKL}	Low clock pulse width	4x + 40	–	360	–	290	–	ns
t_{VCKH}	High clock pulse width	4x + 40	–	360	–	290	–	ns

4.6 Interrupt Operation

V_{CC} = 5V ± 10% TA = -20 ~ 70°C
f = 1 ~ 16MHz

Symbol	Item	Variable		12.5MHz Clock		16MHz Clock		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	INT0 Low level pulse width	4x	–	320	–	250	–	ns
t_{INTAH}	INT0 High level pulse width 	4x	–	320	–	250	–	ns
t_{INTBL}	INT1, INT2 Low level pulse width 	8x + 100	–	740	–	600	–	ns
t_{INTBH}	INT1, INT2 High level pulse width 	8x + 100	–	740	–	600	–	ns

4.7 D/A Conversion Characteristics (VCC = 5V, VSS = AVSS = 0V)

V_{CC} = 5V ± 10% TA = -20 ~ 70°C
f = 1 ~ 16MHz

Symbol	Item	Min	Typ	Max	Unit
–	Analysis ability	–	–	8	Bits
–	Absoluteness accuracy (VCC = AVCC = 5V)	–	–	1.0	%
t_{SU}	Establishment time	–	–	3	μs
R_O	Output resistance	1	2	4	kΩ
V_{AVSS}	Analog power supply voltage	–	0	–	V
V_{DAVREF}	Analog power supply voltage	4	–	V_{CC}	V
I_{DAVREF}	Reference power supply input current	0	2.5	5	mA

4.8 Serial Channel SIO1 Timing - I/O Interface Mode
(1) SCLK1 Input Mode

V_{CC} = 5V ± 10% TA = -20 ~ 70°C
f = 1 ~ 16MHz

Symbol	Item	Variable		12.5MHz Clock		16MHz Clock		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK1 cycle	16x	–	1.28	–	1	–	μs
t_{OSS}	Output data →Rising edge of SCLK	t_{SCY}/2 - 5x - 50	–	190	–	137	–	ns
t_{OHS}	SCLK1 rising edge→Output data hold	5x - 100	–	300	–	212	–	ns
t_{HSR}	SCLK1 rising edge→Input data hold	0	–	0	–	0	–	ns
t_{SRD}	SCLK1 rising edge→ Effective data input	–	t_{SCY} - 5x - 100	–	780	–	587	ns

(2) SCLK1 Output Mode

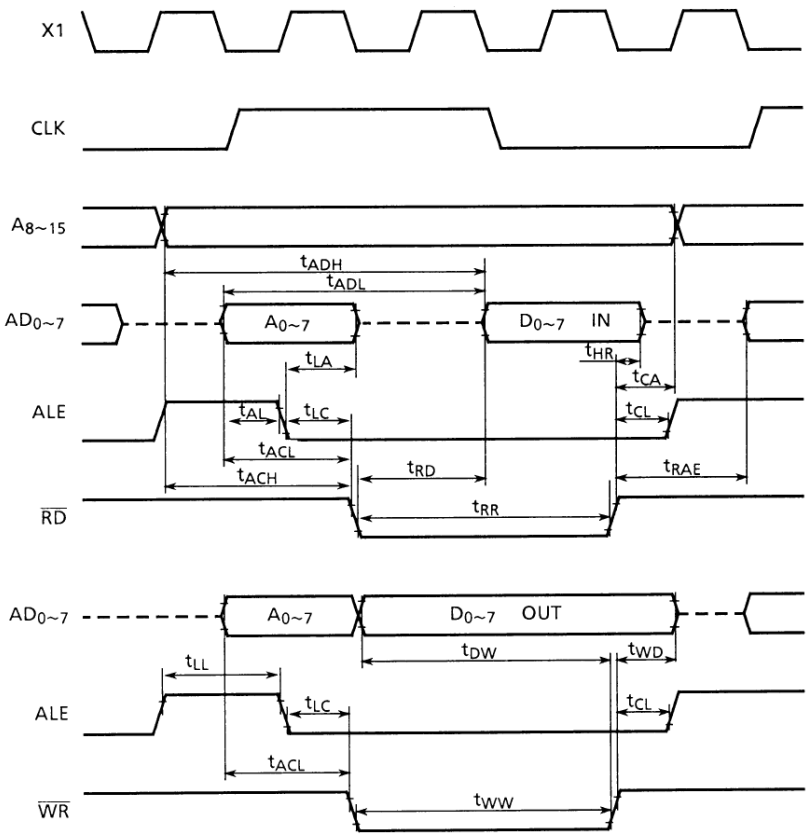
Symbol	Parameter	Variable		12.5MHz Clock		16MHz Clock		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle (programmable)	16x	8192x	1.28	655.4	1	512	μs
t_{OSS}	Output data setup→SCLK rising edge	t_{SCY} - 2x - 50	–	970	–	725	–	ns
t_{OHS}	SCLK rising edge→Output data hold	2x - 80	–	80	–	45	–	ns
t_{HSR}	SCLK rising edge→Input data hold	0	–	0	–	0	–	ns
t_{SRD}	SCLK rising edge→ Effective data input	–	t_{SCY} - 2x - 150	–	970	–	725	ns

4.9 Serial Channel SIO2 Timing

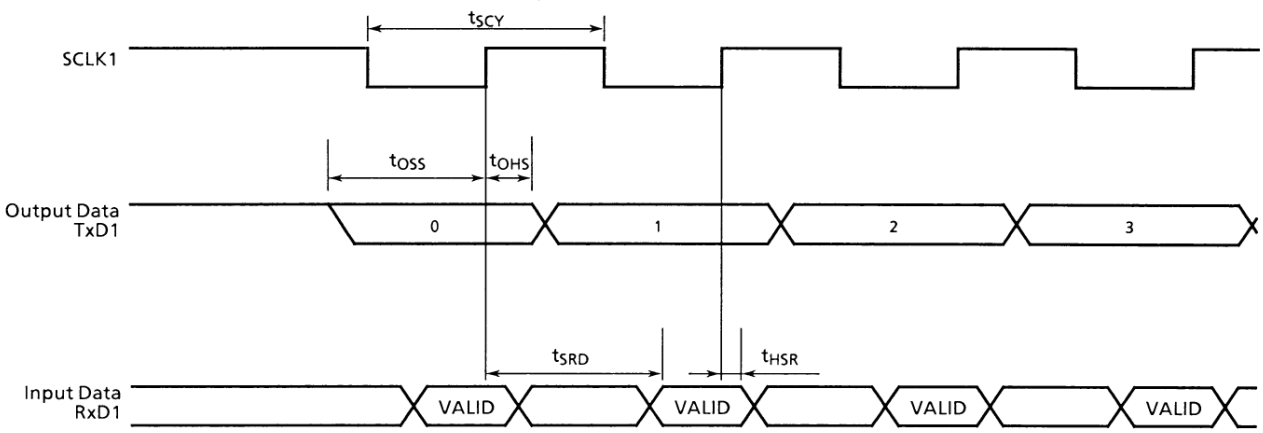
V_{CC} = 5V ± 10% TA = -20 ~ 70°C
f = 1 ~ 16MHz

Symbol	Item	Condition	10MHz Clock		VariableClock		Unit
			Min	Max	Min	Max	
t_{SCR}	Serial port clock cycle time	Internal	800	12800	8x	128x	ns
		External	1600	–	16x	–	
t_{SCL}	SCLK2 Low width	Internal	*	*	*	*	ns
		External	*	*	*	*	
t_{SCH}	SCLK2 High width	Internal	*	*	*	*	ns
		External	*	*	*	*	
t_{SKDO}	SCLK2 → TXD2 (Output data) delay time	Internal	*	–	*	–	ns
		External	*	–	*	–	
t_{SRD}	SCLK2 Rising edge to input data valid	Internal	*	–	*	–	ns
		External	*	–	*	–	
t_{HSR}	Input data hold after SCLK2 rising edge	Internal	*	–	*	–	ns
		External	*	–	*	–	

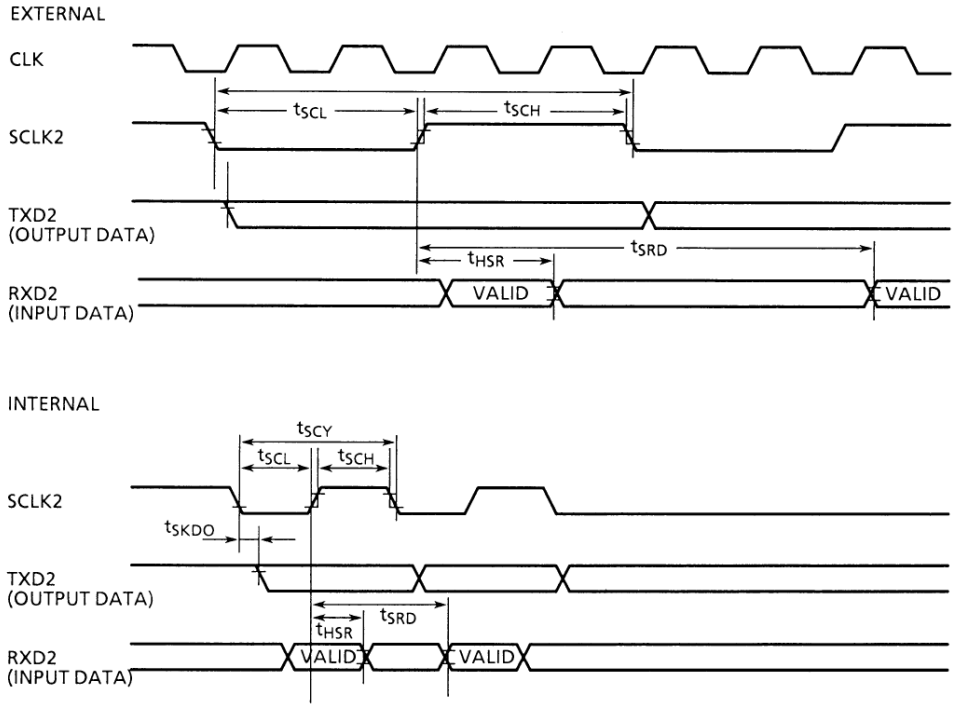
4.10 Timing Chart



4.11 Serial Channel SIO1 I/O Interface Mode Timing Chart



4.12 Serial Channel SIO2 Timing Chart



5. Special Function Register List

The special function registers (SFR) are the input/output ports, peripheral control registers. These SFR are assigned to 96-byte address areas from 0FFA0H to 0FFFFH.

(1) Input/Output Port
(2) Input/Output Port Control
(3) Timer/Event Counter Control
(4) A/D Converter Control
(5) Interrupt Control
(6) HDMA Control
(7) WDT Control
(8) Serial Channel Control
(9) Time Base Counter Control
(10) Timing Pulse Generation Control
(11) Capture Control
(12) D/A Converter
(13) PWM Control

Table configuration

Symbol	Name	Address	7	6			1	0	
									→ bit Symbol
									→ Read / Write
									→ initial value after reset
									→ Remarks

TMP90CM36 Special Function Register List

Address	Symbol	Address	Symbol	Address	Symbol
FFA0	P0	FFB0	P5FR	FFC0	TPGREG1
FFA1	P0CR	FFB1	P7FR	FFC1	CAPFST
FFA2	P1	FFB2	P2FR	FFC2	CAPREG0
FFA3	P1CR	FFB3	P4FR	FFC3	CAPREG1
FFA4	P2	FFB4		FFC4	CAPREG2
FFA5	P2CR	FFB5	TBM0D	FFC5	CFREG
FFA6	P3	FFB6	TBCD0	FFC6	TREG0
FFA7	P38CR	FFB7	TBCD1	FFC7	TREG1
FFA8	P4	FFB8	TBCD2	FFC8	TREG2
FFA9	P4CR	FFB9	TPCREG0	FFC9	TREG3
FFAA	P5	FFBA	TPGD00	FFCA	T01MOD
FFAB	P5CR	FFBB	TPGD01	FFCB	T23MOD
FFAC	P6	FFBC	TPGREG0	FFCC	TFFCR
FFAD	P7	FFBD	TPCREG1	FFCD	TRDC
FFAE	P7CR	FFBE	TPGD10	FFCE	TRUN
FFAF	P8	FFBF	TPGD11	FFCF	CAP1L/TREG4L

Address	Symbol	Address	Symbol	Address	Symbol
FFD0	CAP1H/TREG4H	FFE0	IRF0	FFF0	SUBUF0
FFD1	CAP2L/TREG5L	FFE1	IRF1	FFF1	BRGCR0
FFD2	CAP2M/TREG5H	FFE2	IRF2	FFF2	SCMOD1
FFD3	T4MOD	FFE3	INTE0	FFF3	SCCR1
FFD4	T4FFCR	FFE4	INTE1	FFF4	SCBUF1
FFD5	(Reserved)	FFE5	INTE2	FFF5	BRGCR1
FFD6	ADCH	FFE6	(Reserved)	FFF6	SCMOD2
FFD7	ADMOD	FFE7	(Reserved)	FFF7	SCBUF2
FFD8	ADREG04	FFE8	DMAV0	FFF8	DADRV
FFD9	ADREG15	FFE9	(Reserved)	FFF9	DAREG0
FFDA	ADREG26	FFEA	(Reserved)	FFFA	DAREG1
FFDB	ADREG37	FFEB	DMAV1	FFFB	PWDR0
FFDC		FFEC	WDMOD	FFFC	PWDR0B
FFDD		FFED	WDCR	FFFD	PWDR1
FFDE		FFEE	SCMOD0	FFFE	PWDR1B
FFDF	(Reserved)	FFEF	SCCR0	FFFF	

(1) I/O Port

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
P0	Port 0	FFA0H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Input mode							
P1	Port 1	FFA2H	P17	P16	P15	P14	P13	P12	P11	P10*
			R/W							
			Input mode							
P2	Port 2	FFA4H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Input mode							
P3	Port 3	FFA6H	—	—	—	—	P33	P32	P31	P30
							R/W			
							0	—	—	1
P4	Port 4	FFA8H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			Input mode							
P5	Port 5	FFAAH	P57	P56	P55	P54	P53	P52	P51	P50
			R/W							
			Input mode							
P6	Port 6	FFACH	P67	P66	P65	P64	P63	P62	P61	P60
			R							
			Input only							
			Shared with analog input pin (AN0~AN7)							
P7	Port 7	FFADH	P77	P76	P75	P74	P73	P72	P71	P70
			R/W							
			Input mode							
P8	Port 8	FFAFH	—	—	—	—	P83	P82	P81	P80
							R/W			
							0	—	—	1

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

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
P0CR	Port 0 Control Register	FFA1H	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
		(RMW disable)	0							
			0: IN			1: OUT (I/O selected bit by bit)				
P1CR	Port 1 Control Register	FFA3H	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
		(RMW disable)	0	0	0	0	0	0	0	0
			0: IN			1: OUT (I/O selected bit by bit)				
P2CR	Port 2 Control Register	FFA5H	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
		(RMW disable)	0	0	0	0	0	0	0	0
			0: IN			1: OUT (I/O selected bit by bit)				
P2FR	Port 2 Function Register	FFB2H	P27TPG	P26TPG	P25TPG	P24TPG	—	—	P27M	P24M
			R/W						R/W	R/W
			0	0	0	0	0	0	0	0
			TPG output select 0: TPG07 output 1: TPG14 output (Validness when <P27M> = 1)	TPG output select 0: TPG06 output 1: TPG15 output	TPG output select 0: TPG05 output 1: TPG16 output	TPG output select 0: TPG04 output 1: TPG17 output (Validness when <P24M> = 1)			0: Set TPG17 to INTTPG1 1: Enable TPG04 or TPG17 (<P27TPG> setting)	0: Set TPG17 to INTTPG1 1: Enable TPG04 or TPG17 (<P24TPG> setting)
P38CR	Port 3/8 Control Register	FFA7H	ALEE	STBYS	PWM1S	PWM0S	P33C	P32C	WDTOUTC	RDE
			R/W	W	R/W	R/W	W	W	W	R/W
				0	0	0	0	0	0	0
		(RMW disable)	ALE control 0: Port 1: ALE	P82 control 0: Port 1: $\overline{STBY}$	P33 control 1	P32 control	0: IN 1: OUT (I/O selected bit by bit)		P80 control 0: Disable 1: Enable	RD control 1: Always $\overline{RD}$ output (When <EXT> = 1)

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

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

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
P4CR	Port 4 Control Register	FFA9H	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
		(RMW disable)	0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P4FR	Port 4 Function Register	FFB3H	CAP3E	CAP2E	CAP1E	CAP0E	CAP3S	CAP2S	CAP1S	CAP0S
			R/W				R/W			
			0	0	0	0	0	0	0	0
			Capture input control 0: Disable 1: Enable				Capture edge detection control 0: ↑ rise 1: ↓ fall			
P5CR	Port 5 Control Register	FFABH	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
			W							
		(RMW disable)	0	0	0	0	0	0	0	0
			0: IN 1: OUT (Select I/O on bit basis)							
P5FR	Port 5 Function Register	FFB0H	EXINE	—	WAITC1	WAITC0	TO5S	TO4S	TO3S	TO1S
			R/W		R/W	R/W	R/W			
			0		0	0	0	0	0	0
			P50 control 0: Port 1: EXIN		Wait control 00: 2 State Wait 01: Normal Wait 10: Non Wait 11: —		P52 control 0: Port 1: TO5	P51 control 0: Port 1: TO4	P57 control 0: Port 1: TO3	P55 control 0: Port 1: TO1
P7CR	Port 7 Control Register	FFAEH	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
			W							
		(RMW disable)	0	0	0	0	0	0	0	0
			0: IN 1: OUT (Select I/O on bit basis)							
P7FR	Port 7 Function Register	FFB1H	ODE2	TXDC2	SCLKC2	ODE1	TXDC1	SCLKC1	ODE0	TxDC0
			R/W							
			0	0	0	0	0	0	0	0
			P77 0: CMOS 1: Open Drain	P77 control 0: Port 1: TxD output	P76 control 0: Port 1: SCLK output	P74 0: CMOS 1: Open Drain	P74 control 0: Port 1: TxD output	P73 control 0: Port 1: SCLK output	P71 control 0: CMOS 1: Open Drain	P71 control 0: Port 1: TxD output

(3) Timer/Event Counter Control (1/3)

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TREG0	8Bit Timer Reg. 0	FFC6H (RMW disable)	—							
			W							
			Undefined							
TREG1	8Bit Timer Reg. 1	FFC7H (RMW disable)	—							
			W							
			Undefined							
TREG2	8Bit Timer Reg. 2	FFC8H (RMW disable)	—							
			W							
			Undefined							
TREG3	8Bit Timer Reg. 3	FFC9H (RMW disable)	—							
			W							
			Undefined							
T01MOD	Timer0, 1 Mode Register	FFCAH	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W		R/W		R/W		R/W	
			0	0	0	0	0	0	0	0
			00: 8Bit Timer 01: 16Bit Timer 10: 8Bit PPG 11: 8Bit PWM		00: — 01: 2^6-1 PWM 10: 2^7-1 Cycle 11: 2^8-1		00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: — 01: ϕT1 10: ϕT4 11: ϕT16	
T23MOD	Timer2, 3 Mode Register	FFCBH	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W		R/W		R/W		R/W	
			0	0	0	0	0	0	0	0
			00: 8Bit Timer 01: 16Bit Timer 10: 8Bit PPG 11: 8Bit PWM		00: — 01: 2^6-1 10: 2^7-1 11: 2^8-1		00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI2 01: ϕT1 10: ϕT4 11: ϕT16	
TFFCR	8 bit Timer Flip-Flop Control Register	FFCCH	FF3C1	FF3C0	FF3IE	FF3IS	FF1C1	FF1C0	FF1IE	FF1IS
			W		R/W		W		R/W	
			—		0	0	—		0	0
			00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		1: TFF3 Invert Enable	0: Inverts by 8-bit Timer 2 1: Inverts by Timer 3	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		1: TFF1 Invert Enable	0: Inverts by 8-bit Timer 0 1: Inverts by Timer 1

(3) Timer/Event Counter Control (2/3)

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TRDC	Timer Reg. Double Buffer Control Register	FFCDH						TR4DE	TR2DE	TR0DE
								R/W		
								0	0	0
								0: Doble Boffer Disable 1: Doble Boffer Enable		
TRUN	Timer Run Control Tegister	FFCEH	—	—	PRRUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
					R/W					
					0	0	0	0	0	0
					Prescaler & Timer Run / Stop Control 0: Stop & Clear 1: Run (Count up)					
CAP1L	Capture Reg. 1	FFCFH	—							
			R							
			Undefined							
CAP1H		FFD0H	—							
			R							
			Undefined							
CAP2L	Capture Reg. 2	FFD1H	—							
			R							
			Undefined							
CAP2H		FFD2H	—							
			R							
			Undefined							
TREG4L	16Bit Timer Reg. 4	FFCFH (RMW disable)	—							
			W							
			Undefined							
TREG4H		FFD0H (RMW disable)	—							
			W							
			Undefined							
TREG5L	16Bit Timer Reg. 5	FFD1H (RMW disable)	—							
			W							
			Undefined							
TREG5H		FFD2H (RMW disable)	—							
			W							
			Undefined							

(3) Timer/Event Counter Control (3/3)

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
T4MOD	16 bit Timer Mode Register	FFD3H	CAP2T5	EQ5T5	CAP1IN	CAPM1	CAPM0	CLE	T4CLK1	T4CLK0
			R/W		W	R/W		R/W	R/W	
			0	0	1	0	0	0	0	0
			TFF5 inversion trigger 0: Disable 1: Enable		0: Soft-Capture 1: Don't care	Capture timing 00: Disable INT1 rise edge 01: TI4↑ TI5↑ INT1 rise edge 10: TI4↑ TI4↓ INT1 fall edge 11: TFF1↑ TFF1↓ INT1 rise edge		1: UC16 Clear Enable	Timer 4 Clock 00: TI4 01: φT1 (8/fc) 10: φT4 (32/fc) 11: φT16 (128/fc)	
T4FFCR	16 bit Timer F/F Control Register	FFD4H	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFFC1	TFF4C0
			W		R/W				W	
			—	—	0	0	0	0	—	—
			00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care ※ Always set at "11" when read out.		TFF4 inversion trigger 0: Disable Trigger 1: Enable Trigger				00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: Don't care ※ Always set at "11" when read out.	
					Invert when capturing VC value into CAP2	Invert when capturing VC value into CAP1	Invert when matching VC with TREG5	Invert when matching VC with TREG5		

(4) A/D Converter Control

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
ADCH	A/D Control Select Register	FFD6H	—	—	—	—	—	—	—	ADCH2
										R/W
										0
										0: AN0~3 1: AN4~7
ADMOD	A/D Converter Mode Register	FFD7H	EOCF	ADBF	REPET	SCAN	ADCS	ADS	ADCH1	ADCH0
			R		R/W	R/W	R/W		R/W	
			0	0	0	0	0	0	0	0
			1: END	1: BUSY	1: Repeat Mode Set	1: Scan Mode Set	1: Low speed mode	1: START	Analog Input Channel Select	
ADREG04	A/D Result Register CH0, 4	FFD8H								
			R							
			Undefined							
			A/D conversion result of channel 0 or channel 4 is stored							
ADREG15	A/D Result Register CH1, 5	FFD9H								
			R							
			Undefined							
			A/D conversion result of channel 1 or channel 5 is stored							
ADREG26	A/D Result Register CH2, 6	FFDAH								
			R							
			Undefined							
			A/D conversion result of channel 2 or channel 6 is stored							
ADREG37	A/D Result Register CH3, 7	FFDBH								
			R							
			Undefined							
			A/D conversion result of channel 3 or channel 7 is stored							

(5) Interrupt Control

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
IRF0	Interrupt Regist Flang & IRF Clear	FFE0H (RMW disable)	IRF3	IRF2	IRF1	IRFAD	IRFTPG1	IRFTPG0	IRFCAP	IRF0
			R (Only IRF clear code can be used to write)							
			0	0	0	0	0	0	0	0
			Interrupt Request Flag 1: Interrupt being requested (IRF is cleared to "0" by writing IRF clear code)							
IRF1		FFE1H (RMW disable)	EXF	—	IRFT5	IRFT4	IRFT3	IRFT2	IRFT1	IRFT0
			R		R					
			Undefined		0	0	0	0	0	0
			EXX instruction Reverses each time executed		1: Interrupt being requested					
IRF2		FFE2H (RMW disable)	EXT	—	IRFTX2	IRFRX2	IRFTX1	IRFRX1	IRFTX0	IRFRX0
			W		R					
			0		0	0	0	0	0	0
			P1 control 0: I/O Port 1: Address $\overline{RD}$, $\overline{WR}$		1: Interrupt being requested					
INTE0	Interrupt Enable Mask Register	FFE3H	IE3	IE2	IE1	IEAD	IETPG1	IETPG0	IECAP	IE0
			R/W							
			0	0	0	0	0	0	0	0
			0: Disable 1: Enable							
INTE1		FFE4H	—	—	IET5	IET4	IET3	IET2	IET1	IET0
			R/W							
			0	0	0	0	0	0	0	0
			0: Disable 1: Enable							
INTE2		FFE5H	—	EDGE	IETX2	IERX2	IETX1	IERX1	IETX0	IERX0
			R/W	R/W						
			0	0	0	0	0	0	0	0
			0: Disable 1: Enable							

(6) HDA Control

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAV0	HDMA Vecter Register CH0	FFE8H	DV07	DV06	DV05	DV04	DV03	DV02	DV01	DV00
			W							
			0	0	0	0	0	0	0	0
DMAV1	HDMA Vecter Register CH1	FFEBH	DV17	DV16	DV15	DV14	DV13	DV12	DV11	DV10
			W							
			0	0	0	0	0	0	0	0

(7) WDT Control

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	Watch Dog Timer Mode Register	FFECH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRIVE
			R/W			R/W			R/W	R/W
			1	0	0	0	0	0	1	0
			1: WDT Enable	WDT dctecting Time 00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warming Up mode 0: 2^{14}/fc 1: 2^{16}/fc	Standby mode 00: RUN 01: STOP 10: IDLE 11: —		Watchdog timer out control 1: P80 is set as the WDTOUT pin and is conntected internally to the RESET pin.	1: STOP Drive the pin even in the STOP mode
WDCR	Watch Dog Timer Control Register	FFEDH (RWM disable)	—							
			W							
			—							
			B1H: WDT Disable Code 4EH: WDT Clear Code							

(8) Serial Channel Control (1/2)

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
SCMOD0	Serial Channal 0 Mode Register	FFEEH	TB80	Fixed to "0"	RXE0	WU0	SM0	SM00	SC10	SC00
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission bit-8 data		1: Receive Enable	1: Wake Up Enable	00: — 01: UART 7Bit 10: UART 8Bit 11: UART 9Bit		00: TO0TRG 01: BRG Mode 10: ϕ1 11: —	
SCCR0	Serial Channal 0 Control Register	FFEFH	RB80	EVEN0	PE0	OERR0	PERR0	FERR0	—	—
			R	R/W		R (Cleared to "0" by reading)				
			Undefined	0	0	0	0	0		
			Bit of receiving data	Parity 0: Odd 1: EVEN	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing		
SCBUF0	Serial Channal 0 Buffer Register	FFF0H	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
			R (Receiving) / W (Transmission)							
			Undefined							
BRGCR0	Serial Channal 0 Baud Rate Generater Control	FFF1H	Fixed to "0"	R/W	BG10	BG00	PS30	PS20	PS10	PS00
				R/W	R/W					
			0	0	0	0	0	0	0	0
				1: CTS Enable	00: fc/4 01: fc/16 10: fc/64 11: fc/256		Divided frequency from prescaler			
SCMOD1	Serial Channel 1 Mode Register	FFF2H	TB81	Fixed to "0"	RXE1	WU1	SM1	SM01	SC11	SC01
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission bit-8 data		1: Receive Enable	1: Wake Up Enable	00: 1/O Interface 01: UART 7Bit 10: UART 8Bit 11: UART 9Bit		00: TO2TRG 01: BRG Mode 10: ϕ1 11: —	
SCCR1	Serial Channel 1 Control Register	FFF3H	RB81	EVEN1	PE1	OERR1	PERR1	FERR1	SCLKC1	IOC1
			R	R/W		R (Cleared to "0" by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Bit of receiving data	Parity 0: Odd 1: EVEN	1: Parity Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK1 (↑) 1: SCLK1 (↓)	0: SCLK1 output 1: SCLK1 input

(8) Serial Channel Control (2/2)

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
SCBUF1	Serial Channel 1 Buffer Register	0FFF4H (RMW disable)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
			R (Receiving) / W (Transmission)							
			Undefined							
BRGCR1	Serial Channel 1 Baud Rate Generater Control	FFF5H	Fixed to "0"		BG11	BG01	PS31	PS21	PS11	PS01
			R/W							
			0		0	0	0	0	0	0
					00: fc/4 01: fc/16 10: fc/64 11: fc/256		Divided frequency from prescaler			
SCMOD2	Serial Channel 2 Mode Register	FFF6H	FFSI	SMD1	SMD0	SIFT	CLK1	CLKS0	SCKS2	SIOE
			R	R/W		R/W	R/W		R/W	R/W
			1	0	0	0	0	0	0	0
			0: Busy 1: Stop	Transmission mode 00: RESET 01: trsnsmission 10: receives 11: trsnsmission and receives		Transfer Sift 0: Rising edge 1: Falling edge	Transfer spead 00: fc/4 (ϕT0) 01: fc/8 (ϕT1) 10: fc/32 (ϕT4) 11: fc/128 (ϕT16)		Transfer clock 0: Internal clock 1: External clock	Transfer control 0: Disable 1: Enable
SCBUF2	Serial Channel 2 Buffer Register	FFF7H (RMW disable)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
			R (Receiving) / W (Transmission)							
			Undefined							

(9) Time Base Counter Control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TBMOD	Time Base Counter Mode Register	FFB5H	INTS2	INTS1	INTTBC22	INTTBC21	INTTBC20	INTTBC12	INTTBC11	INTTBC10
			R/W							
			0	0	0	0	0	0	0	0
			00: Disable 01: INTTBC1 10: INTTBC2 11: Both		TBC Interrupt select 000: TBC12 001: TBC14 010: TBC16 011: TBC18		100: TBC20		TBC Interrupt select 000: TBC11 001: TBC13 010: TBC15 011: TBC17	100: TBC19
TBCD0	Time Base Counter Data Register 0	FFB6H	TBCD7	TBCD6	TBCD5	TBCD4	TBCD3	TBCD2	TBCD1	TBCD0
			R							
			0							
			Time base counter data TBCD3~10							
TBCD1	Time Base Counter Data Register 1	FFB7H	TBCD15	TBCD14	TBCD13	TBCD12	TBCD11	TBCD10	TBCD9	TBCD8
			R							
			0							
			Time base counter data TBCD11~18							
TBCD2	Time Base Counter Data Register 2	FFB8H	—	—	—	—	—	—	—	—
									R	
									0	
									Time base counter Data TBC19, TBC20	

(10) Timing Pulse Generators Control (1/2)

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TPCREG0	TPG0 Control Register	FFB9H	CONP01	CONP00		ENMINT0	TPFUL0	TPEMP0	TPF01	TPF00
			R/W	R/W		R/W	R	R	R	
			0	0		0	0	0	0	0
			Comparator bit select 00:TBC3~TBC18 01:TBC4~TBC19 10:TBC5~TBC20		0: Disable 1: Enable	TPG0 DATA Empty interrupt control 0: Disable 1: Enable	TPG0 DATA Full Flag 1: Full	TPG0 DATA Empty Flag 1: Empty	TPG0 FIFO status flag 00: Empty/fall 01: 1 Data 02: 2 Data 11: 3 Data Empty when "00" and TPEMP0 is "1" Full when "00" and TPEMP0 is "0"	
TPGD00	TPG0 Conparis-on Data Reg.0	FFBAH	TPG0D7	TPG0D6	TPG0D5	TPG0D4	TPG0D3	TPG0D2	TPG0D1	TPG0D0
			W							
			—							
			TPG0 comparator data 8 lower bits							
TPGD01	TPG0 Conparis-on Data Reg.1	FFBBH	TPG0D15	TPG0D14	TPG0D13	TPG0D12	TPG0D11	TPG0D10	TPG0D9	TPG0D8
			W							
			—							
			TPG0 comparator data 8 upper bits							
TPGREG0	TPG0 Out Data Register	FFBDH	D07	D06	D05	D04	D03	D02	D01	D00
			W							
			—							
			TPG0 output data							

(10) Timing Pulse Generators Control (2/2)

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
TPCREG1	TPG1 Control Register	FFBDH	CONP11	CONP10		ENMINT1	TPFUL1	TPEMP1	TPF11	TPF10
			R/W	R/W		R/W	R	R	R	
			0	0		0	0	0	0	0
			Comparator bit select 00:TBC3~TBC18 01:TBC4~TBC19 10:TBC5~TBC20		0: Disable 1: Enable	TPG0 DATA Empty interrupt control 0: Disable 1: Enable	TPG1 DATA Full Flag 1: Full	TPG1 DATA Empty Flag 1: Empty	TPG0 FIFO status flag 00: Empty/fall 01: 1 Data 02: 2 Data 11: 3 Data Empty when "00" and TPEMP0 is "1" Full when "00" and TPEMP0 is "0"	
TPGD10	TPG1 Conparis-on Data Reg.0	FFBEH	TPG1D7	TPG1D6	TPG1D5	TPG1D4	TPG1D3	TPG1D2	TPG1D1	TPG1D0
			W							
			—							
			TPG1 comparator data 8 lower bits							
TPGD11	TPG1 Conparis-on Data Reg.1	FFBFH	TPG1D15	TPG1D14	TPG1D13	TPG1D12	TPG1D11	TPG1D10	TPG1D9	TPG1D8
			W							
			—							
			TPG1 comparator data 8 upper bits							
TPGREG1	TPG1 Out Data Register	FFC0H	D17	D16	D15	D14	D13	D12	D11	D10
			W							
			—							
			TPG1 output data							

(11) Capture Control

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
CAPFST	Capture FIFO Status Register	FFC1H					CAPF3	CAPF2	CAPF1	CAPF0
			R							
							0	0	0	0
			0: No data 1: Data							
CAPREG0	Capture Data 0	FFC2H	CAPD7	CAPD6	CAPD5	CAPD4	CAPD3	CAPD2	CAPD1	CAPD0
			R							
			—							
			Capture data 0~7							
CAPREG1	Capture Data 1	FFC3H	CAPD15	CAPD14	CAPD13	CAPD12	CAPD11	CAPD10	CAPD9	CAPD8
			R							
			—							
			Capture data 8~15							
CAPREG2	Capture Data 2	FFC4H	CAP3	CAP2	CAP1	CAP0	CAPD19	CAPD18	CAPD17	CAPD16
			R				R			
			—				—			
			Capture Input status				Capture data 16~19			
CFREG	Flag Control Register	FFC5H	TBCC1	TBCC0	—	FTBC2	FTBC1	TPFRS1	TPFRS0	CAFRS
			R/W			R/W		R/W (注)		
			0	0		0	0	0	0	0
			Input Clock select 00: 01: 10: External clock			1: TBC_{12}, TBC_{14}, TBC_{16}, TBC_{18}, TBC_{20}, interrupt Flag	1: TBC_{11}, TBC_{13}, TBC_{15}, TBC_{17}, TBC_{19}, interrupt Flag	TPG1 FIFO counter 0: — 1: Clear	TPG0 FIFO counter 0: — 1: Clear	Capture FIFO counter 0: — 1: Clear

(12) D/A Converter

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
DADRV	D/A Drive Register	FFF8H	—	—	—	—	—	—	DA1DR	DA0DR
			R/W							
			0							
			0: 0V output 1: register value conversion output							
DAREG0	D/A Convers-ion Reg.0	FFF9H	—							
			W							
			0							
			D/A conversion started with register write, output to DA-0.							
DAREG1	D/A Convers-ion Reg.1	FFFAH	—							
			W							
			0							
			D/A conversion started with register write, output to DA-1.							

(13) PWM Control

			MSB							LSB
Symbol	Name	Address	7	6	5	4	3	2	1	0
PWDR0	PWM Data Reg.0	FFFBH	PWD07	PWD06	PWD05	PWD04	PWD03	PWD02	PWD01	PWD00
			W							
			Undefined							
PWDR0B	PWM Data Reg.0B	FFFCH	—	—	—	—	PWD0B	PWD0A	PWD09	PWD08
			W							
			Undefined							
PWDR1	PWM Data Reg.1	FFFDH	PWD17	PWD16	PWD15	PWD14	PWD13	PWD12	PWD11	PWD10
			W							
			Undefined							
PWDR1B	PWM Data Reg.1B	FFFEH	—	—	—	—	PWD1B	PWD1A	PWD19	PWD18
			W							
			Undefined							