CMOS 8-BIT MICROCONTROLLERS

TMP91C642N/TMP91C642F

1. OVERVIEW AND FEATURES

The TMP91C642N is an advanced function, highly integrated 8-bit microcontroller for software servos.

In addition to I/O ports and other basic structural components, the TMP91C642N has high-speed, high-precision signal measuring circuits, dedicated PWM output circuits and high-precision timing pulse generator circuits that simplify the control of VCR systems and servo motors.

(1) Efficiently systematized instructions

163 basic instructions
Multiplication, dvision, 16-bit arithmetic operations, bit manipulation (for 1-, 4-, 8- and 16-bit processing) , instructions.

(2) Pipeline processing for high speed.

Minimum execution time 400 nsec (at 10MHz oscillation frequency)

(3) Internal ROM : 16K byte

(4) Internal RAM : 320 byte

(5) High-precision, 8-bit A/D converter (8+4 channels)

(6) General-purpose, synchronized serial interface (2 channels)

(7) 8-bit timer/counter (4 channels)

- Reloadable timer
  (one internal channel with up/down modes)
- Counter data, read/write

(8) Interrupts

11 internal and 3 external

(9) Watchdog timer

(10) Power down function (two standby modes)

(11) Servo input control pins

Capstan FG, drum FG/PG, CTL and EXT

(12) Servo control output pins

12-bit PWM output (2 channels)
8-bit PWM output (1 channel)

(13) 18-bit time base counter

(14) 24-bit time base counter capture, 8-step with FIFO.

(15) 32-bit timing pulse generator, 8-step with FIFO
(16-bit comparator data and 16-bit timing output)

(16) Composite sync (C-sync) input
- Vertical sync signal (V-sync) discriminator
- Odd-even field discriminator
- NTSC/PAL discriminator

(17) VISS/VASS detection
- CTL duty discriminator
- 16-bit VASS data latch

(18) Micro-DMA

(19) 64-pin SDIP and 64-pin QFP

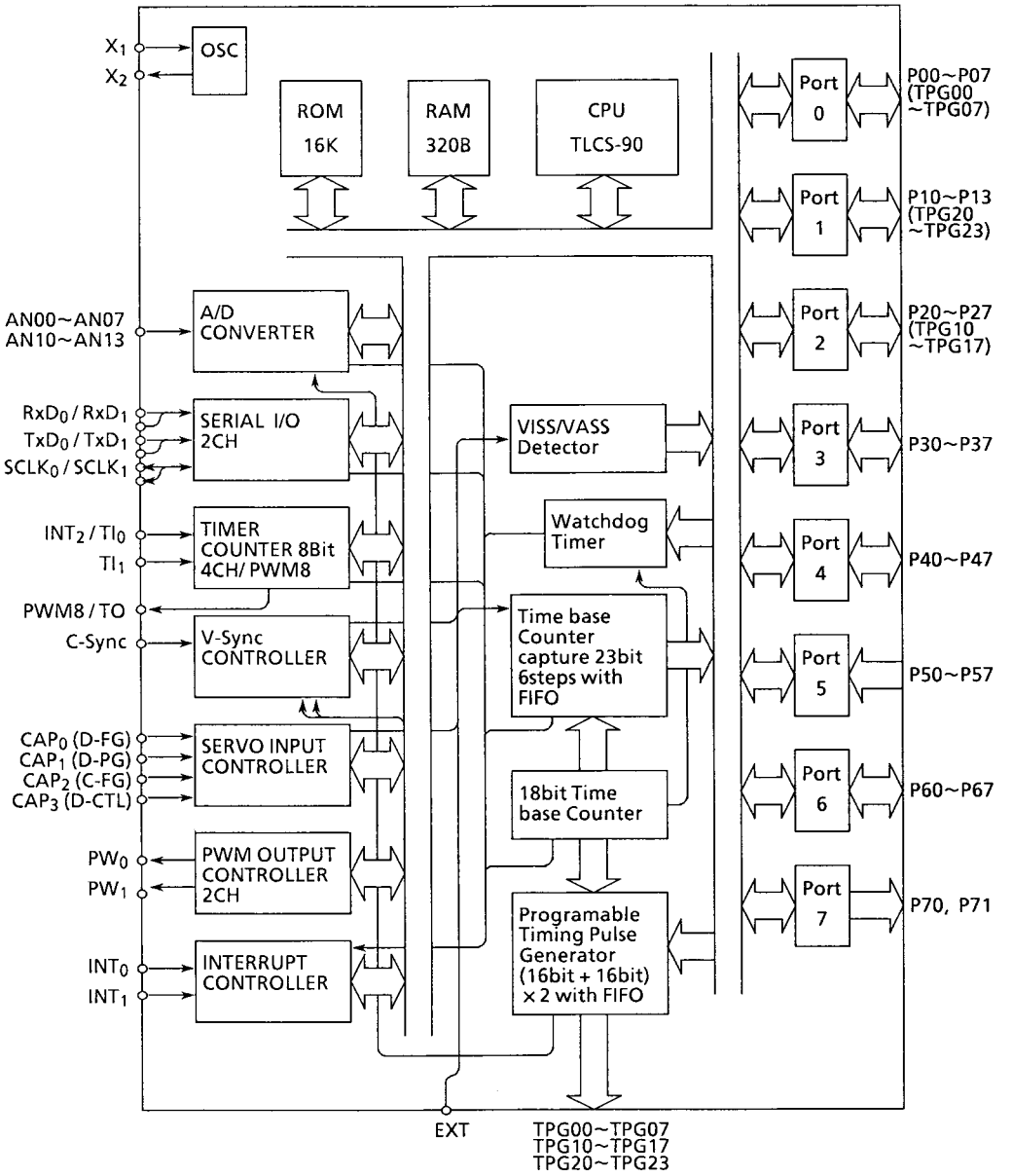


Figure 1 TMP91C642 Block Diagram

2. PIN ASSIGNMENT AND FUNCTIONS

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

2.1 Pin Assignment

Figure 2.1 shows pin assignment of the TMP91C642.

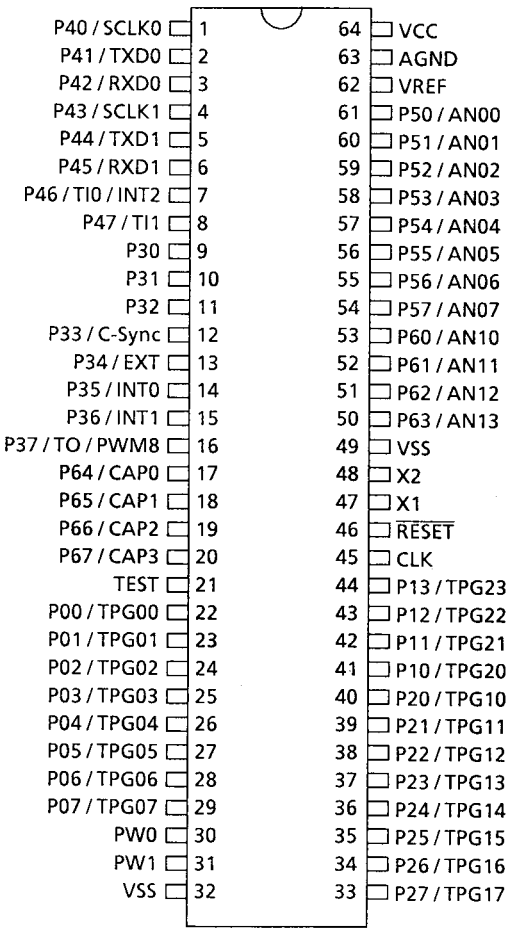


Figure 2.1 (1) Pin Assignment (Shrink Dual Inline Package)

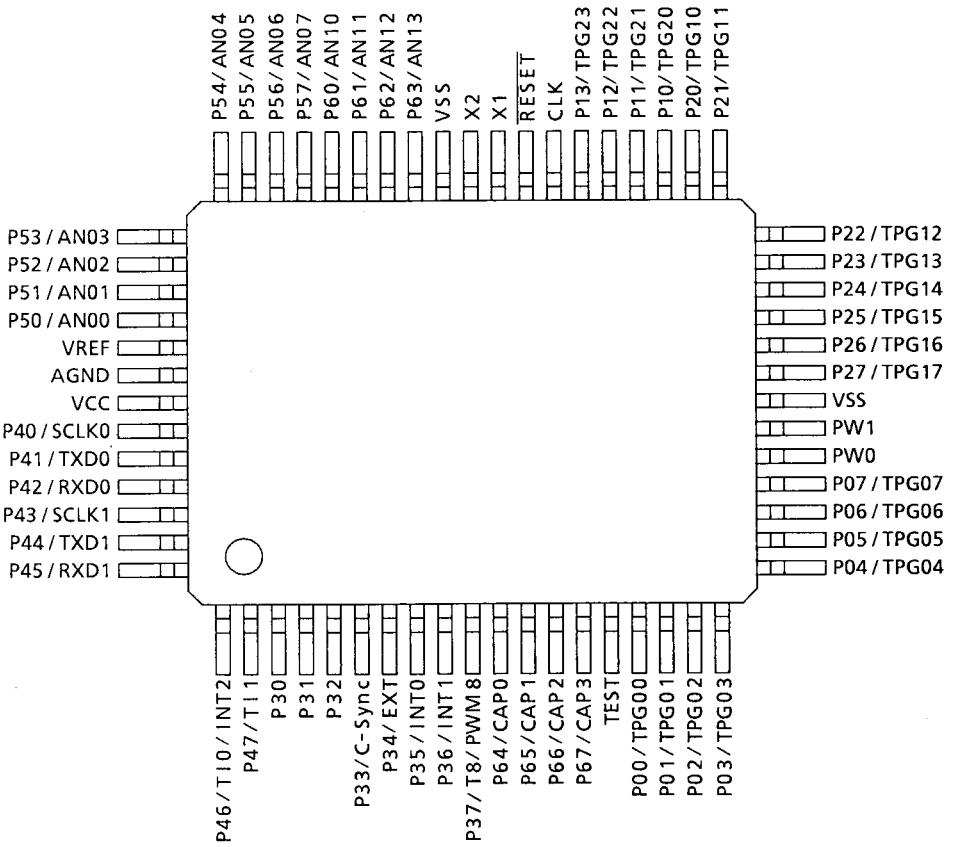


Figure 2.1 (2) Pin-Assignment (Flat Package)

2.2 Pin Names and Functions

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

Table 2.2 Pin Names and Functions (1/2)

Pin name	No. of pins	I/O 3-state	Function
P00~P07 / TPG00~ TPG07	8	I/O 3-state	Port 0 : 8-bit I/O port. Input/output can be specified in bytes.
			Timing pulse generator (TPG) output : These pins are also used for TPG0-0 to TPG0-7 output pins. *P00 only can be placed in 3-state by TPG0-0.
P10~P13 / TPG20~ TPG23	4	I/O 3-state	Port 1 : 4-bit input/output port. Input/output can be specified in bits.
			Timing pulse generator (TPG) output: These pins are also used for TPG0-8 to TPGG0-11 or TPG1-8 to TPG1-11 output pins. P10 only can be placed in 3-state by TPG0-14/TPG1-14.
P20~P27 / TPG10~ TPG27	8	I/O 3-state	Port 2 : 8-bit input/output port. Input/output can be specified in bits.
			Timing pulse generator (TPG) output : These pins are also used for TPG1-0 to TPG1-7 output pins. P20 only can be placed in 3-state by TPG0-15.
P30~P32	3	I/O	Port 3 : 3-bit input/output port. Input/output can be specified in bits.
P33 C-Sync	1	I/O	Port 3 : 1-bit input/output port. Input/output can be specified.
			Composite sync input
P34 / EXT	1	I/O	Port 3 : 1-bit input/output port. Input/output can be specified.
			Servo signal trigger input
P35 / INT0	1	Input	Port 3 : 1-bit input port
			Interrupt request pin 0 : Level and rising edge are programmable.
P36 / INT1	1	Input	Port 3 : 1-bit input port.
			Interrupt request pin : Pin for requesting interrupt of rising edge.
P37 / PWM8 / TO1	1	Output	Port 3 : 1-bit output port
			Motor control output : 8-bit PWM output
			Timer output 0 and 1 : Timer 0 and 1 output
P40 · P43 / SCLK0 · SCLK1	2	I/O	Port 4 : 1-bit input/output port. Input/output can be specified in bits.
			Serial clock input/output 0 and 1

Table 2.2 Pin Names and Functions (2/2)

Pin name	No. of pins	I/O 3-state	Function
P41 · P44 / T_XD_0 · T_XD_1	2	I/O	Port 4 : 1-bit input/output port. Input/output can be specified in bits.
			Serial send data 0 and 1
P42 · P45 / R_XD_0 · R_XD_1	2	I/O	Port 4 : 1-bit input/output port. Input/output can be specified in bits.
			Serial receive data 0 and 1
P46 / TI0 / INT2	1	I/O	Port 4 : 1-bit input/output port. Input/output can be specified.
			Timer 0 and timer 2 counter input Interrupt request pin : Pin for requesting interrupt of rising edge.
P47 / T11	1	I/O	Port 4 : 1-bit input/output port. Input/output can be specified.
			Timer 3 counter input
P50~P57 / AN00~AN07	8	Input	Port 5 : 8-bit input port.
			Analog input 0 : 8 analog inputs to A/D converter.
P60~P63 / AN10~AN13	4	I/O	Port 6 : 4-bit input/output port. Input/output can be specified in bits.
			Analog input 1 : 4 analog inputs to A/D converter.
P64~P67 / CAP_0~CAP_3	4	Input	Port 6 : 4-bit input port
			Servo signal trigger input
PWM0	1	Output	Motor control output : Output from PWM12-0.
PWM1	1	Output	Motor control output : Output from PWM12-1.
VREF	1	–	A/D converter reference voltage input
AGND	1	–	A/D converter GND pin
CLK	1	Output	Clock output : Outputs a pulse that is one-fourth the clock oscillating pulse frequency. Pulled up during reset.
TEST	1	–	Test pin (normally pulled up)
$\overline{RESET}$	1	Input	Reset : Initializes internally.
X_1 / X_2	2	I/O	Quartz crystal oscillator connector pin
V_{SS} (GND)	2	–	GND pin (0V)
V_{CC}	1	–	Power pin (+ 5V)

3. OPERATION

This chapter describes the functions and the basic operations of the TMP91C642 in every block.

3.1 CPU

The TMP91C642 contains a high-performance 8-bit CPU that offers improved instructions, addressing and execution speeds conventional 8-bit CPUs.

3.1.1 Memory Map

(1) Internal ROM

The TMP91C642 internally contains an 16K-byte ROM. The address space from 0000H to 3FFFH is provided to the ROM. The CPU starts executing a program from 0000H by resetting.

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

(2) Internal RAM

The TMP91C642 also contains a 320 byte RAM, which is allocated to the address space from FE80H to FFBFH. The CPU allows the access to a certain RAM area (FF00H to FFBFH, 192 bytes) by a short operation code (opcode) in a "direct addressing mode".

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

(3) Internal I/O

The TMP91C642 provides a 64-byte address space as an internal I/O area, whose addresses range from FFC0H to FFFFH. This I/O area can be accessed by the CPU using a short opcode in the "direct addressing mode".

Figure 3.1 (1) is a memory map indicating the areas accessible by the CPU in the respective addressing mode.

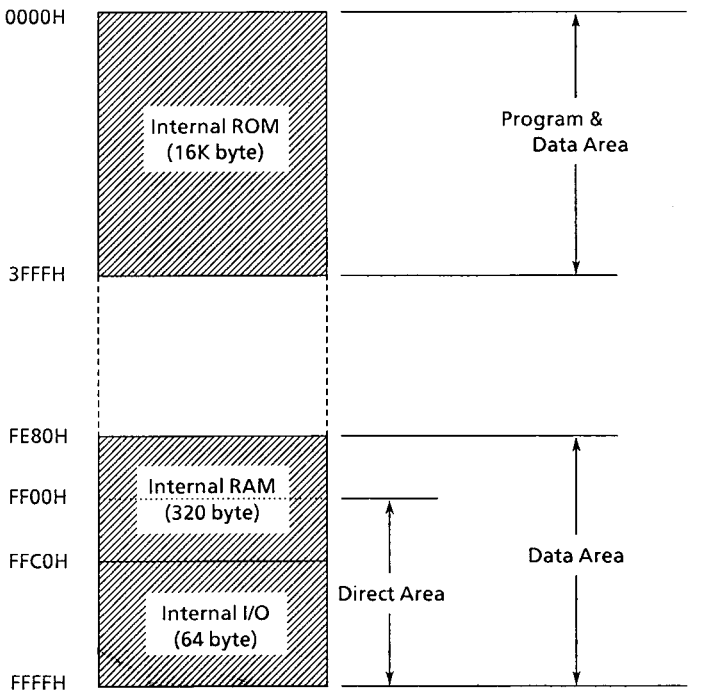


Figure 3.1 (1) Memory Map

3.1.2 Registers

Figure 3.1 (2) shows the configuration of registers.

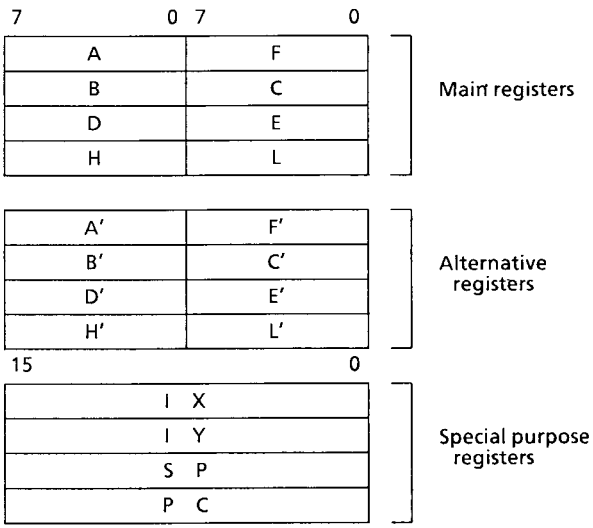


Figure 3.1 (2) Configuration of Registers

The TMP91C642 uses the main registers, the alternative registers and the special purpose registers. The main registers and the alternative registers are allowed to be exchanged of their contents by a register exchange instruction.

(1) Register A

This is an 8-bit register used mainly for 8-bit arithmetic and logic operations.

(2) Register F

This is an 8-bit register that stores the status of operation results. Configuration of register F is shown in Figure 3.1 (3).

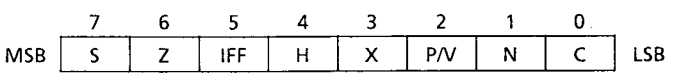


Figure 3.1 (3) Configuration of Register F

- Sign flag (S)

The sign flag is set to "1" when the arithmetic result is negative. It stores the contents of the most significant bit (MSB) of the arithmetic and logic unit (ALU).

- Zero flag (Z)

  Z flag is set to "1" when the all bits of the ALU after operation are "0".

- Parity/Overflow flag (P/V)

  This flag has two functions. One is to indicate the parity (P) resulted from a logical operation (AND, OR, or XOR). It is set to "1" when the parity of operation result is even, and "0" for odd.
  The other is to indicates the overflow (V) in an arithmetic operation (ADD, ADC, SUB, SBC, or CP). The flag is set to "1" when the result cannot be expressed by a signed integral number.
  The P/V flag selects either function according to the instruction.

- Carry flag (C)

  The flag is set to "1" if a carry or borrow has occurred on the MSB of the ALU.

- Expansion carry flag (X)

  Like the carry flag (C), it is set to "1" when the MSB of the ALU involves a carry or borrow as a result of an operation except that it applies to a wider range of instructions (e.g., INC rr).

- Half carry flag (H)

  It is set to "1" when a carry or borrow has occurred on the 4th bit of the lower side in the ALU.

- Addition/Subtraction flag (N)

  This flag is set to "1" if the executed operation is a subtraction (SUB, SBC, CP, or DEC).

- Interrupt enable flag (IFF)

  A maskable interrupt is enabled or disabled by this flag. This flag is set to "1" by an EI instruction and "0" by an DI instruction.

Note : This flag is shared with the alternative register F'.

(3) Registers B, C, D, E, H and L

All these registers have an 8-bit configuration. They function as 16-bit register pairs (concatenated BC, DE and HL) as well as independent 8-bit register. Registers B or register pair BC is also used as a counter for the loop instruction (DJNZ). Register pair HL is used for 16-bit data processing including 16-bit arithmetic/logic operations.

(4) Registers A', F', B', C', D', E', H' and L'

These registers have the same structure as the main registers (A, F, B, C, D, E, H and L). They are called alternative registers. There is no instruction that directly accesses these alternative registers, but its data can be processed by a register exchange instruction.

Following are examples of register exchange instructions that allow the exchange of data between a main register and an alternative register:

```
EX    AF, AF'
EXX
```

(5) Registers IX and IY

IX and IY are 16-bit independent registers called index registers.

These registers are used mainly for specifying memory addresses. IX and IY registers are also used for 16-bit additions.

(6) SP register

SP register is a 16-bit register called a stack pointer (SP), that stores the start address of the memory stack area (Last in, first out basis). It is decremented when a CALL or PUSH instruction is executed or an interrupt is accepted. It is incremented by execution of RET instruction or a POP instruction.

(7) PC register

This is a 16-bit register called a program counter, and stores the memory address of the next instruction to be executed.

It is initialized to 0000H when the $\overline{\text{RESET}}$ pin becomes low.

(8) Other

By executing the data exchange instruction [EXX] between a main register and an alternative register, the EXF bit (exchange flag: Bit 1 of memory address FFE6H) of the internal I/O register is inverted. This is a read-only bit, and is not initialized by resetting.

3.1.3 Addressing Modes

Eight addressing modes are available for the TMP91C642. They are used in combination with various instructions to enhance the CPU's processing capabilities.

They are: Register mode, immediate mode, register indirect mode, index mode, register index mode, extend mode, direct mode and relative mode. The first seven addressing modes are used most frequently. The relative addressing mode is only applicable to specific instructions.

(1) Register addressing mode

In the register addressing mode, the operand represents a specified register.

Example: LD A, B

The contents of Register B are loaded into Register A.

(2) Immediate addressing mode

In this mode, the operand is in the instruction.

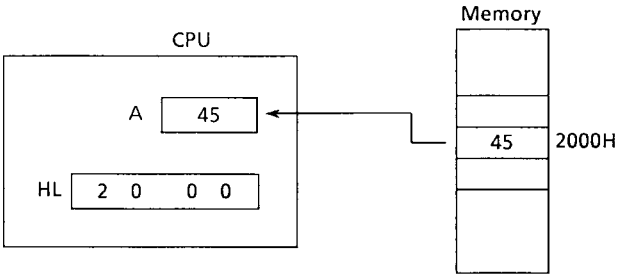
Example: LD A, 12H

Immediate data "12H" are loaded into Register A.

(3) Register indirect addressing mode

In the register indirect addressing mode, the operand is a memory address indicated by a register pair (BC, DE, HL, IX, IY or SP).

Example: LD A, (HL)

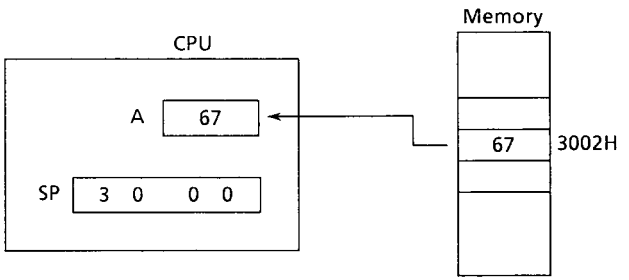


"45H" in the memory address 2000H is loaded into Register A.

(4) Index addressing mode

In the index addressing mode, the operand is a memory address specified by adding an 8-bit displacement value in the opcode to the contents of a specified register pair (IX, IY or SP).

Example: LD A, (SP+2)

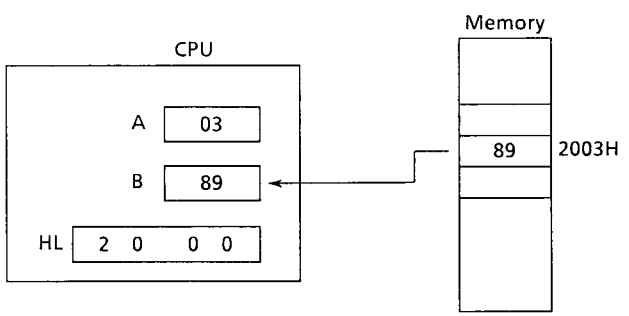


"67H" in the memory address 3002H is loaded into Register A. The displacement value ranges from −128 to +127.

(5) Register index addressing mode

In this mode, the operand is a memory address specified by adding the displacement value of Register A to the contents of register pair HL.

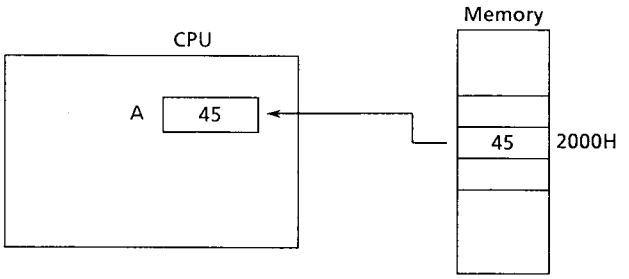
Example: LD B, (HL+A)



"89H" in the memory address 2003H is loaded into Register B. In this mode, the data in Register A are considered as 8-bit signed number, and the displacement value ranges from −128 to +127.

(6) Extended addressing mode

In this mode, the operand is accessed by 2-byte (16-bit) data in the opcode.
Example: LD A, (2000H)



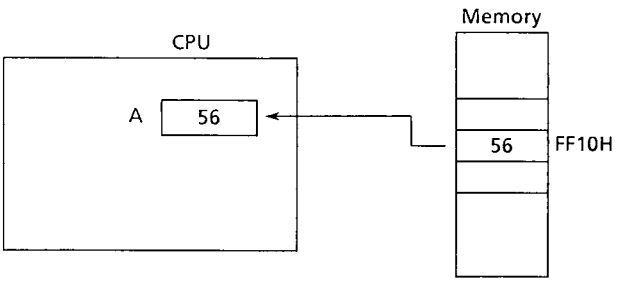
"45H" in the memory address 2000H is loaded into Register A.

(7) Direct addressing mode

The operand in this mode is a memory address from FF00H to FFFFH specified by 1-byte (8 bits) data in the opcode. Compared with the extended addressing mode, it saves both program memory and executing time. This mode allows the access to 256-byte addresses from FF00H to FFFFH

For the TMP91C642, this direct area is divided into the internal RAM (192 bytes from FF00H to FFBFH) and the internal I/O area (64 bytes from FFC0H to FFFFH).

Example: LD A, (FF10H)

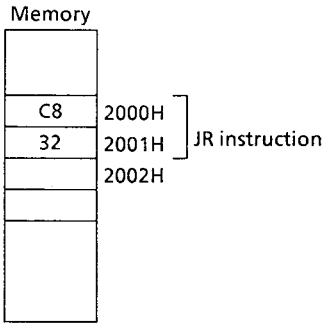


"56H" in the memory address FF10H is loaded into Register A.

(8) Relative addressing mode

In the relative addressing mode, the operand is defined as the relative address to the current address. This mode is applicable to instructions involving an 8-bit displacement value (JR and DJNZ) and a 16-bit displacement value (LDAR, JRL and CALR).

Example: JR 2034H



In this example, the program execution jumps to the address 2034H. Since the program counter is already incremented by 2 at the time of address computation, the displacement is obtained by the following formula based on the "memory address of the JR instruction + 2":

Destination address − (address of instruction being executed + 2)

In the example, the displacement "32H" is obtained by:

2034H − (2000H + 2)

In any other instructions using the relative addressing mode (DJNZ, LDAR, JRL and CALR), the displacement is always calculated based on the "address of the current instruction + 2".

3.1.4 Instructions

The TMP91C642 supports a lot of addressing modes as well as powerful instruction sets. There are 163 basic instructions as categorized into the following nine groups:

- 8-bit transfer instruction
- 16-bit transfer instruction
- Exchange, block transfer and search instructions
- 8-bit arithmetic and logical operation instruction
- Special operation and CPU control instructions
- 16-bit arithmetic and logical operation instruction
- Rotate and shift instructions
- Bit manipulation instruction
- Jump, call and return instruction

Table 3.1 (1) lists the 163 basic instructions. Table 3.1. (2) descries the mnemonics and their meaning.

(1) 8-bit transfer instruction

The 8-bit transfer instructions include those for transferring 8-bit data between registers, register and immediate address, register and memory, or memory and immediate address.

(2) 16-bit transfer instruction

The 16-bit transfer instructions include those for transferring 16-bit data between registers, register and immediate address, register and memory, and memory and immediate address, PUSH and POP instructions using the stack, and LDA (Load Address) instruction that calculates an effective address and loads its value into a register.

(3) Exchange, block transfer and search instructions

The data exchange instructions are executed to exchange 16-bit data between registers, between memory and register, or between a main register and an alternative register.

The block transfer instructions can transfer data in any memory block to other memory area.

The block search instructions are executed to find out a particular 8-bit character in a given memory block.

LDIR, LDDR, CPIR and CPDR included in the block transfer and search instructions read the current instruction each time a 1-byte memory is transferred or compared, thus making it possible to acknowledge an interrupt before reaching to the end of the block.

(4) 8-bit arithmetic and logical operation instruction

8-bit arithmetic and logical operation instructions perform 8-bit arithmetic and logical operations between Register A and another register, Register A and immediate address, Register A and memory, register and immediate address, and memory and immediate address (ADD, ADC, SUB, SBC, AND, OR, XOR and CP), or increment/ decrement the contents of register or memory by 1 (INC, DEC, INCX and DECX).

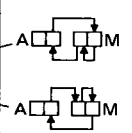
The INCX (Increment if X) instruction increments the contents of a memory specified by the operand if the X flag is "1", and does nothing if not. The DECX instruction performs the same operation except that it decrements the data.

Examples:

```
INC  IX        : Increment Register IX
LD   A, (IX)   : Load contents of memory into Register A
```

Table 3.1 (1) TMP91C642 Basic Instructions (163 types)

LD	r, r	SBC	A, r	DI		RLCA	
LD	r, n	SBC	A, n	EI		RLC	r
LD	r, mem	SBC	A, mem	SWI		RLC	mem
LD	mem, r	SBC	r, n	MUL	HL, r L×r	RRCA	
LD	mem, n	SBC	mem, n	MUL	HL, n	RRC	r
LD	rr, rr	AND	A, r	MUL	HL, mem	RRC	mem
LD	rr, nn	AND	A, n	DIV	HL, r L←Q	RLA	
LD	rr, mem	AND	A, mem	DIV	HL, n H←R	RL	r
LD	mem, rr	AND	r, n	DIV	HL, mem	RL	mem
LDW	mem, nn	AND	mem, n	ADD	HL, rr	RRA	
PUSH	qq	OR	A, r	ADD	HL, nn	RR	r
POP	qq	OR	A, n	ADD	HL, mem	RR	mem
LDA	rr, mem	OR	A, mem	ADC	HL, rr	SLAA	
EX	DE, HL	OR	r, n	ADC	HL, nn	SLA	r
EX	AF, AF'	OR	mem, n	ADC	HL, mem	SLA	mem
EXX	BC, DE, HL	XOR	A, r	SUB	HL, rr	SRAA	
EX	mem, rr	XOR	A, n	SUB	HL, nn	SRA	r
LDI	(DE)←(HL)	XOR	A, mem	SUB	HL, mem	SRA	mem
LDIR	BC←BC – 1	XOR	r, n	SBC	HL, rr	SLLA	
LDD		XOR	mem, n	SBC	HL, nn	SLL	r
LDDR		CP	A, r	SBC	HL, mem	SLL	mem
CPI	A – (HL)	CP	A, n	AND	HL, rr	SRLA	
CPIR	BC←BC – 1	CP	A, mem	AND	HL, nn	SRL	r
CPD		CP	r, n	AND	HL, mem	SRL	mem
CPDR		CP	mem, n	OR	HL, rr	RLD	mem
ADD	A, r	INC	r	OR	HL, nn	RRD	mem
ADD	A, n	INC	mem	OR	HL, mem	BIT	b, r
ADD	A, mem	DEC	r	XOR	HL, rr	BIT	b, mem
ADD	r, n	DEC	mem	XOR	HL, nn	RES	b, r
ADD	mem, n	INCX	(n)	XOR	HL, mem	RES	b, mem
ADC	A, r	DECX	(n)	CP	HL, rr	SET / TSET	b, r
ADC	A, n	DAA	A	CP	HL, nn	SET / TSET	b, mem
ADC	A, mem	CPL	A	CP	HL, mem	JP	cc, mem
ADC	r, n	NEG	A	ADD	ix, rr	JR	cc, PC + d
ADC	mem, n	LDAR	HL, PC + dd	ADD	ix, nn	JRL	PC + dd
SUB	A, r	CCF		ADD	ix, mem	CALL	cc, mem
SUB	A, n	SCF		INC	rr	CALR	PC + dd
SUB	A, mem	RCF		INCW	mem	DJNZ	[BC,]PC + d
SUB	r, n	NOP		DEC	rr	RET	cc
SUB	mem, n	HALT		DECW	mem	RETI	



mem : (n), (nn), (rr), (ix + d), (HL + A)
r : A, B, C, D, E, H, L
Q : Quotient
rr : BC, DE, HL, IX, IY, SP
ix : IX, IY, SP
R : Remainder

Table 3.1 (2) TMP91C642 Mnemonics and Their Meaning

Mnemonic	Meaning	Mnemonic	Meaning
LD	Load	MUL	Multiply
LDW	Load Word	DIV	Divide
PUSH	Push	INCW	Increment Word
POP	Pop	DECW	Decrement Word
LDA	Load Address	RLCA	Rotate Left Circular Accumulator
EX	Exchange	RLC	Rotate Left Circular
EXX	Exchange X	RRCA	Rotate Right Circular Accumulator
LDI	Load and Increment	RRC	Rotate Right Circular
LDIR	Load, Increment and Repeat	RLA	Rotate Left Accumulator
LDD	Load and Decrement	RL	Rotate Left
LDDR	Load, Decrement and Repeat	RRA	Rotate Right Accumulator
CPI	Compare and Increment	RR	Rotate Right
CPIR	Compare, Increment and Repeat	SLAA	Shift Left Arithmetic Accumulator
CPD	Compare and Decrement	SLA	Shift Left Arithmetic
CPDR	Compare, Decrement and Repeat	SRAA	Shift Right Arithmetic Accumulator
ADD	Add	SRA	Shift Right Arithmetic
ADC	Add with Carry	SLLA	Shift Left Logical Accumulator
SUB	Subtract	SLL	Shift Left Logical
SBC	Subtract with Carry	SRLA	Shift Right Logical Accumulator
AND	And	SRL	Shift Right Logical
OR	Or	RLD	Rotate Left Digit
XOR	Exclusive Or	RRD	Rotate Right Digit
CP	Compare	BIT	Bit Test
INC	Increment	RES	Reset Bit
DEC	Decrement	SET	Set Bit
INCX	Increment if X	TSET	Test and Set
DECX	Decrement if X	JP	Jump
DAA	Decimal Adjust Accumlator	JR	Jump Relative
CPL	Complement	JRL	Jump Relative Long
NEG	Negate	CALL	Call
LDAR	Load Address Relative	CALR	Call Relative
CCF	Complement Carry Flag	DJNZ	Decrement and Jump if Non Zero
SCF	Set Carry Flag	RET	Return
RCF	Reset Carry Flag	RETI	Return from Interrupt
NOP	No Operation		
HALT	Halt		
DI	Disable Interrupt		
EI	Enable Interrupt		
SWI	Software Interrupt		

(5) Special and CPU control instruction

Special instructions are used to control Register A (DAA, CPL and NEG), load the data in a relative address into a register (LDAR), control the carry flag (CCF, SCF and RCF), multiply an 8-bit data by an 8-bit data and obtain a 16-bit representation (MUL), divide a 16-bit data by an 8-bit divisor and obtain an 8-bit quotient with an 8-bit residual (DIV), and operate nothing (NOP).

If a quotient cannot be represented by an 8-bit number (outside the 0-255 range) or if the divisor is 0 (e.g., "divide 1,000 by 2" or "divide 5,000 by 0"), the overflow flag is set to "1". In this case, the quotient and residue are indefinite.

CPU control instructions are executed to suspend the CPU operation (HALT), enable/disable a maskable interrupt (EI/DI), and to execute a software interrupt (SWI).

(6) 16-bit arithmetic and logical operation instruction

The 16-bit arithmetic and logical operation instructions perform 16-bit arithmetic logical operations between the register pair HL and another register pair, the register pair HL and immediate address, and the register pair HL and memory (ADD, ADC, SUB, SBC, AND, OR, XOR and CP), perform an addition between the index registers (IX and IY) or Stack Pointer SP and a register pair, or immediate address and memory, or perform an increment/decrement (INC, INCW, DEC, and DECW) the contents of a register pair or memory by 1.

Note that "ADD HL, rr", "ADD ix, gg", "INC rr" and "DEC rr" result in a different flag status.

(7) Rotate and shift instructions

The rotate and shift instructions use 8-bit data (RLC, RRC, RL, RR, SLA, SRA, SLL and SRL) or binary-coded decimal (BCD) data (RLD and RRD).

(8) Bit manipulation instruction

The bit manipulation instructions perform testing, setting and resetting a particular bit in a register or memory (BIT, SET and RES).

A test and reset instruction (TEST) is also available for executing multiple tasks.

(9) Jump, call and return instructions

The addressing modes of jump instruction are the register indirect, index, register index, extended and the 8-bit and 16-bit relative. Note, however, that the 16-bit relative addressing mode can be used only for an unconditional jump instruction.

The addressing modes of call instruction are the 16-bit relative, the register indirect, index, register index, and extended. Again, 16-bit relative addressing mode is only applicable to an unconditional call instruction.

Return instructions contain unconditional return instruction, conditional return instruction and RETI instruction for return back from the interrupt processing.

These instructions pop up the program counter (PC) and the register pair (AF) from the stack. 16 condition codes and over 16 mnemonics are used in these instructions, because certain flags have more than one meaning (e.g. Z and EQ, NZ and NE, PE and OV, PO and NOV).

In addition to the above instructions, "DJNZ PC+d" and "DJNZ BC, PC+d" may be used to control program loops.

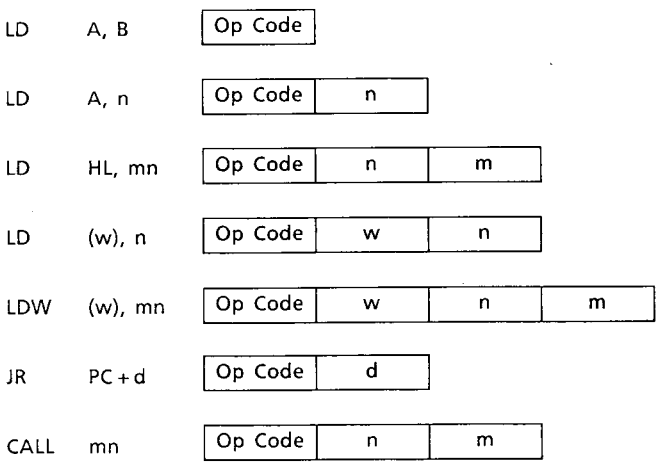
"DJNZ PC+d" decrements the contents of the Register B (8-bit) each time the instruction is executed, and executes a relative jump until it becomes zero. "DJNZ BC, PC+ d" decrements the contents of the register pair BC (16-bit), and executes a relative jump until it becomes zero.

Appendix C lists the TMP91C642 machine instructions. The table includes the instruction group, mnemonics, codes, functions, flag status and executing time.

The executing time can be calculated using the value in the "T" column, which denotes the number of states. Time for one state is equivalent to a time twice as long as the clock oscillation cycle. For example, if the clock oscillation frequency is 10MHz, the time for one state is 200 ns.

Executing "LD A, r" at the clock frequency of 10MHz requires two states, and thus takes 200ns × 2 = 400ns for the execution.

Appendix B contains code maps. The TMP91C642 supports 1-byte opcode instructions and 2-byte opcode instructions. The 1-byte opcode instruction is formatted as follows:

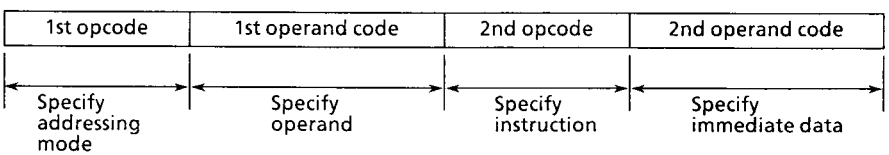


As shown in the code format, a 1-byte opcode instruction has an opcode in the first byte and operand codes in the subsequent bytes. If there are two byte operand codes, the lower operand is placed before the upper operand. If both a source and a destination are included as operand codes, the destination is placed first.

A 2-byte opcode instruction begins with the first opcode, followed by an operand code specified by the first opcode, then the second opcode and its operand code. For example,

LD	B, C	1st Op Code	2nd Op Code			
LD	B, (n)	1st Op Code	n	2nd Op Code		
LD	B, (IX + d)	1st Op Code	d	2nd Op Code		
LD	B, (mn)	1st Op Code	n	m	2nd Op Code	
ADD	B, n	1st Op Code	2nd Op Code	n		
ADD	(vw), n	1st Op Code	w	v	2nd Op Code	n

In a 2-byte opcode instruction, the position of the second opcode is determined by the first opcode. Basically, the first opcode (E0H~FEH) in 2-byte opcode instruction provides data to select the addressing mode of the operand. The first operand code that follows the first opcode serves to specify the memory addressing mode. The second operand code that follows the second opcode specifies the immediate addressing mode. Their roles can be summarized as follows:



3.2 Basic Timing

Each instruction of the TMP91C642 is executed by combination of read, write and dummy cycles. These are basic cycles that synchronize with the system clock. 1/2 of the frequency of the clock oscillation is used as the system clock; e.g., if the clock frequency is 10 MHz, the frequency of the system clock is 5MHz. The system clock cycle is also called a "state".

The TMP91C642 bus operation are basically synchronous, and each of memory read, memory write and dummy cycles is completed in two states, unless they are not requested to wait.

The "CLK" pin generates a pulse at a frequency that further halves the frequency of the system clock. This CLK signal synchronizes with the bus cycles with no wait request.

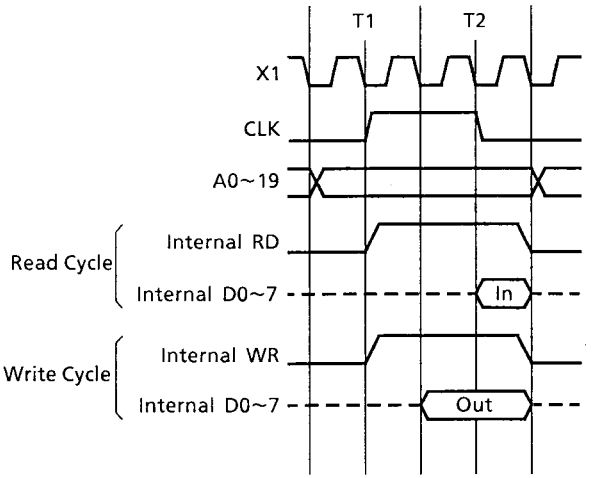


Figure 3.2 (1) Timing of Internal Memory Read/Write Cycles

3.2.1 Interrupt Acknowledge Timing

Figure 3.2 (2) shows the basic timing of interrupts being acknowledged. An interrupt request is sampled by the CPU at the falling edge of the CLK signal in the last bus cycle of each instruction. Note, however, that the sampling of a non-maskable interrupt ($\overline{NMI}$) is delayed a half the system clock cycle.

When an interrupt request is acknowledged, the CPU starts an interrupt response sequence that proceeds as follows: 1) A read cycle (In this cycle, the read data is not used in the CPU because the pipeline processing prefetches instructions. The pipeline processing is described in "3.2.6 Bus Operation for Executing Instructions".), 2) two dummy cycles (the CPU receives an interrupt vector from an internal interrupt controller), 3) output of the interrupt vector to the address bus (000H for the upper address locations A8 to A19) and read out of the dummy cycle, 4) one dummy cycle, 5) saving the contents of the program counter PC and those of the register pair AF into the stack (four write cycles), and 6) the CPU resets the interrupt enable flag IFF to "0" (to disable interrupts) and jumps to the interrupt processing routine.

If a "micro DMA processing" is specified as the interrupt, the CPU follows the sequence to be described in 3.3.2 Micro DMA processing".

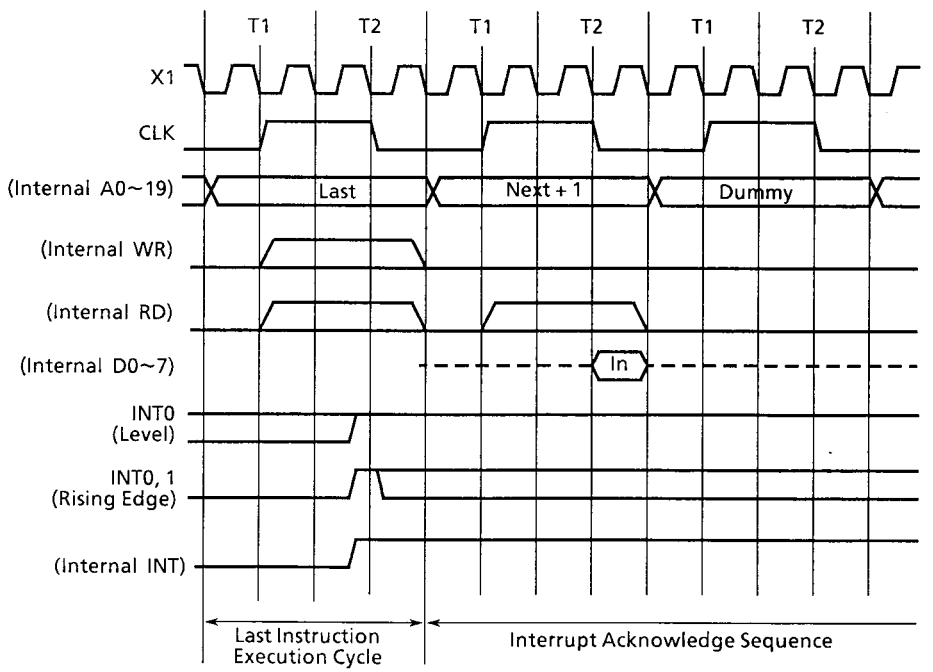


Figure 3.2 (2) Interrupt Acknoledge Timing

3.2.2 Reset

The basic timing of the reset operation is indicated in Figure 3.2 (3). In order to reset the TMP91C642, the $\overline{\text{RESET}}$ input must be maintained at the "0" level for at least ten system clock cycles (10 states). When a reset request is accepted, all I/O ports function as input ports (high impedance state). The CLK pin that always function as output ports turn to the "1" level, and the other output ports turn to the "0" level. The dedicated input ports remain unchanged. The registers of the CPU also remain unchanged. Note, however, that the program counter PC and the interrupt enable flag IFF are cleared to "0". Register A shows an undefined status.

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

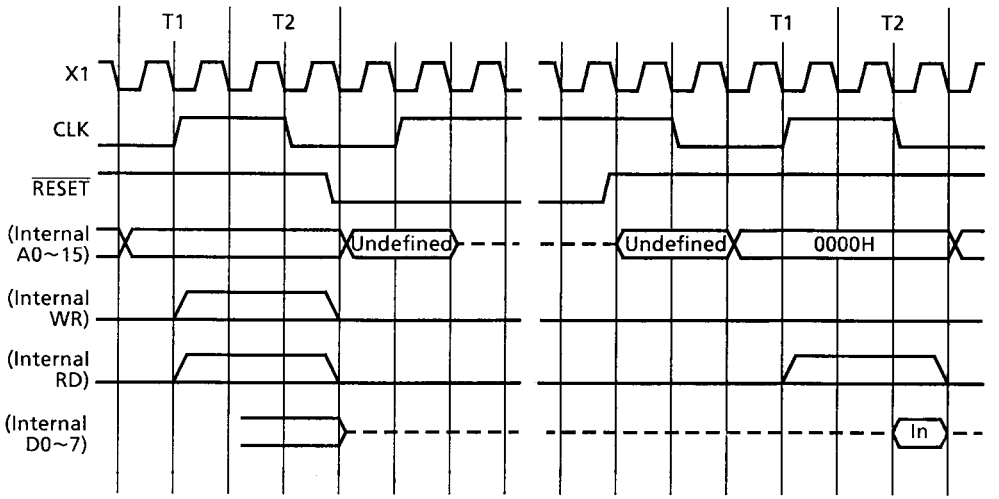


Figure 3.2 (3) Reset Timing

3.2.3 System Flowchart

Figure 3.2 (4) is a system flowchart of the TMP91C642. A normal operation repeats a loop between "Fetch instruction" and "Execute instruction".

When an interrupt is acknowledged, the CPU proceeds to "Interrupt processing". Executing the return instruction RETI makes the CPU return to the address that follows the address of the SWI instruction in the case that the SWI instruction is executed under software control.

When a HALT instruction is executed, the CPU suspends the operation until an interrupt is requested. When the interrupt is acknowledged, the CPU starts the interrupt processing. However, when a maskable interrupt is requested with the interrupt enable flag at "0" (interrupts are disabled), the CPU only releases the HALT state and starts executing an instruction that follows the HALT instruction.

For details of the interrupt processing, refer to "3.3 Interrupt Function". The timing of releasing the HALT state is described in "3.4 Standby Function".

By setting the $\overline{\text{RESET}}$ input level to "0", the CPU always returns to "$\overline{\text{RESET}}$"start position without regard to its current position in the flowchart.

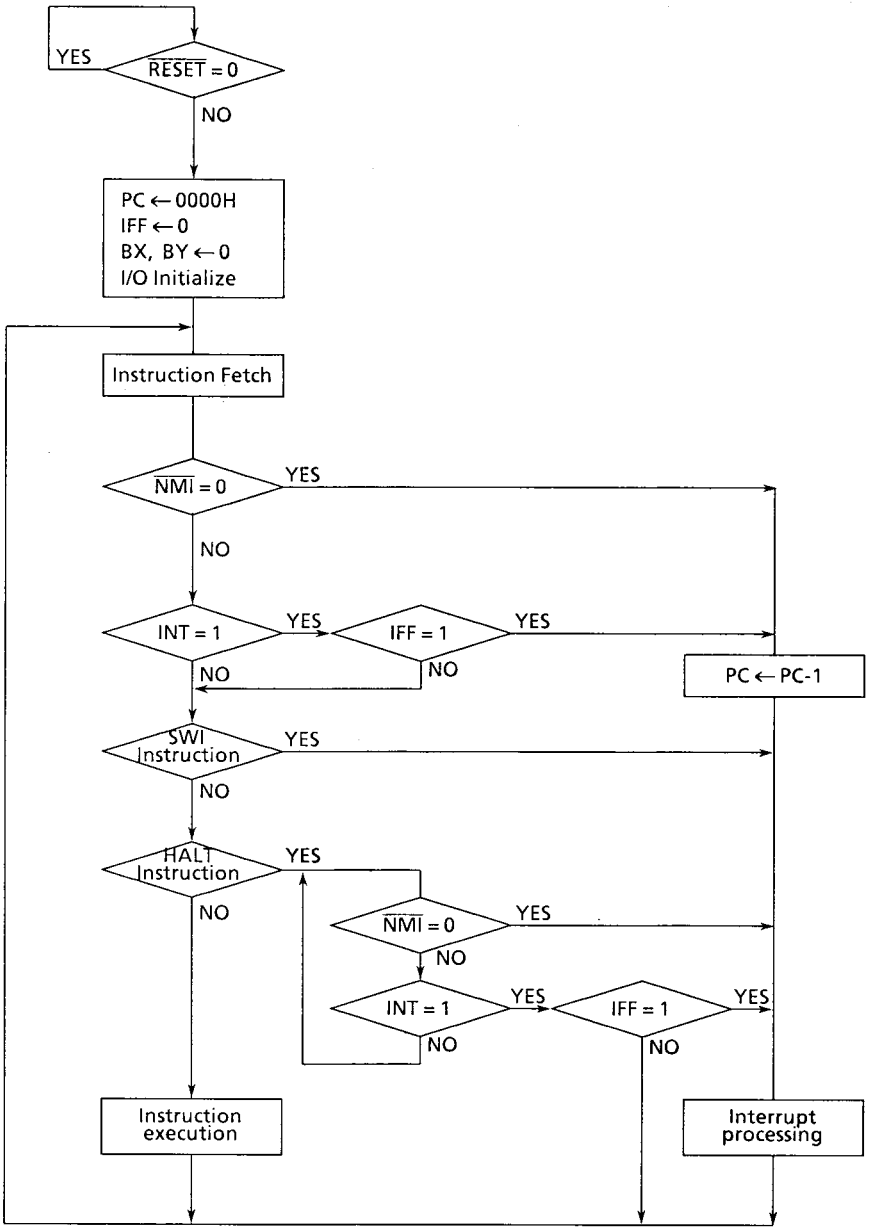


Figure 3.2 (4) TMP91C642 System Flowchart

3.2.4 Bus Operation For Executing Instructions

The TMP91C642 adopts a pipeline processing method in which it executes an instruction simultaneous with the next instruction fetch. The concept of this processing is illustrated in Figure 3.2 (5).

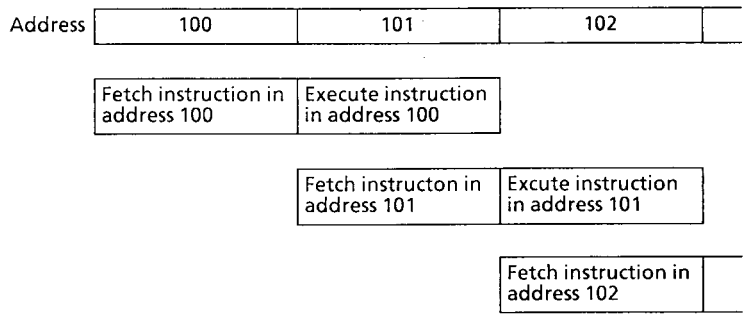


Figure 3.2 (5) Pipeline Processing

This pipeline processing allows the TMP91C642 to obtain a higher executing speed than the conventional method that fetches the next instruction after the previous instruction is executed. The bus operation for each instruction begins with "fetching a code in the address that follows the first instruction code", and not with "fetching the first instruction code". The first instruction code is fetched when the CPU is executing the previous instruction. An example of this processing is shown in Figure 3.2 (6).

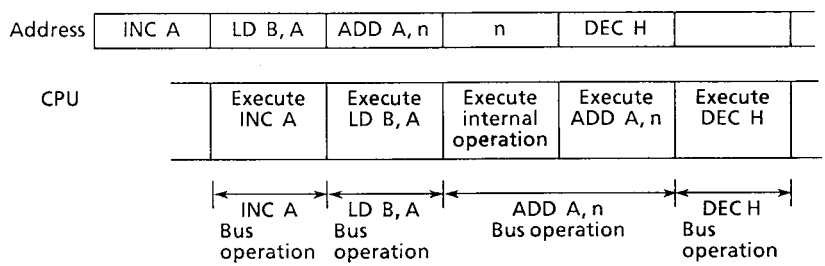


Figure 3.2 (6) Pipeline Processing and Bus Operation

3.3 Interrupt Functions

The TMP91C642 supports a general purpose interrupt processing mode and a micro DMA processing mode that enables automatic data transfer by the CPU for internal and external interrupt requests. After the reset state is released, all interrupt requests are processed in the general purpose interrupt processing mode. However, they can be processed in the micro DMA processing mode by using a DMA enable register to be described later.

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

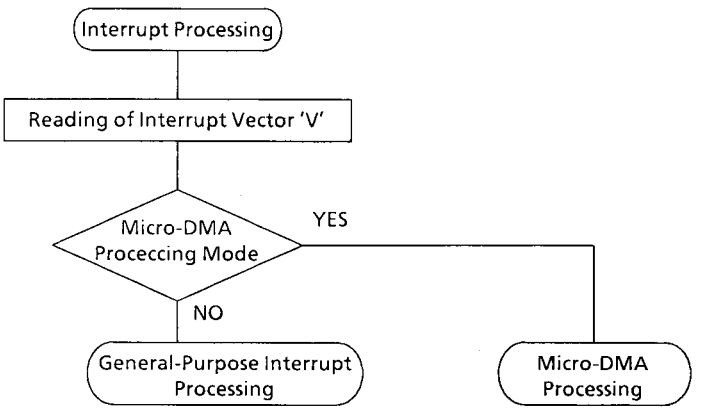


Figure 3.3 (1) Interrupt Response Flowchart

When an interrupt is requested, the source of the interrupt transmits the request to the CPU via an internal interrupt controller. The CPU starts the interrupt processing if it is a non-maskable or maskable interrupt requested in the EI state. However, a maskable interrupt requested in the DI state is ignored.

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

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

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

3.3.1 General Purpose Interrupt Processing

A General Purpose Interrupt Is Processed As Shown in Figure 3.3. (2).

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

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

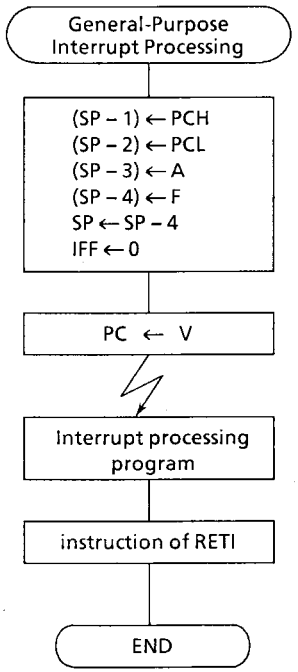


Figure 3.3 (2) General Purpose Interrupt Processing Flowchart

An interrupt processing program ends with a RETI instruction.

When this instruction is executed, the data previously stacked from the program counter PC and the register pair AF are restored.

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

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

Table 3.3 (1) lists the possible interrupt sources.

Table3.3 (1) Interrupt Sources

Priority order	Type	Interrupt sourece	Start address of general purpose interrupt processing	Start address of Mico DMA processing parameter
1	Non maskable	SWI instruction	0010H	—
2		INTWD (Watchdog)	0020H	—
3		INT0 (External input 0)	0028H	FF28H
4		INTCAP (Capture Interrupt)※1	0030H	FF30H
5		INTAD (A/D Converter)	0038H	FF38H
6		INTSIO (Serial I/O)※2	0040H	※3
7		INTT0 (Timer 0)	0048H	FF48H
8		INTT1 (Timer 1)	0050H	FF50H
9		INTT2 (Timer 2)	0058H	FF58H
10		INTT3 (Timer 3)	0060H	FF60H
11		INTTB (Devider)	0068H	FF68H
12		INT1 (External input 1)	0070H	FF70H
13		INTVA (VASS)	0078H	FF78H
13		INT2 (External input 2)	0078H	FF78H

Note ※1) C-FG, D-FG, D-PG, P-CTL, EXT, VP (C-Sync) / TPFIFO
※2) SIO_1 / SIO_2
※3) The INTSIO can not use Micro DMA processing.

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

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

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

3.3.2 Micro DMA Processing

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

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

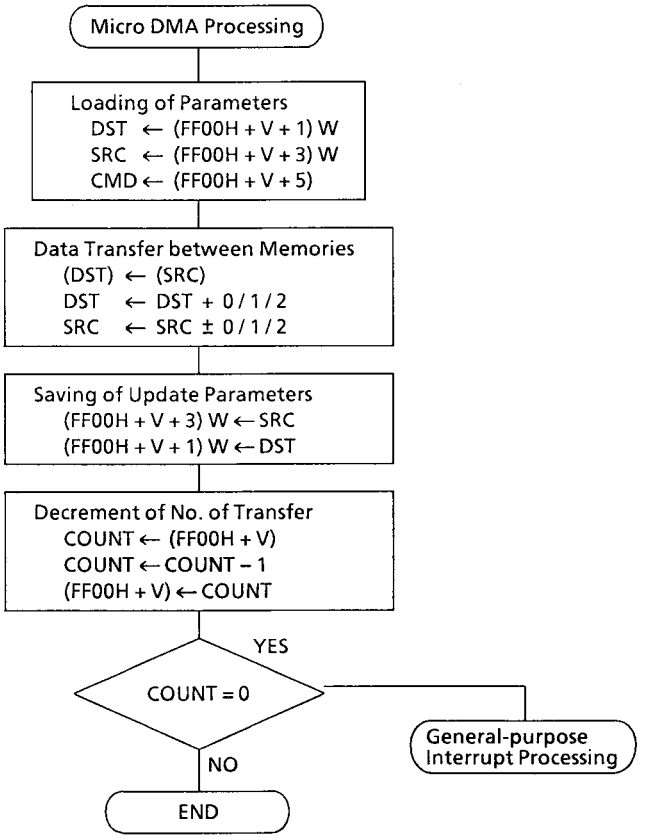


Figure 3.3 (3) Micro DMA Processing Flowchart

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

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

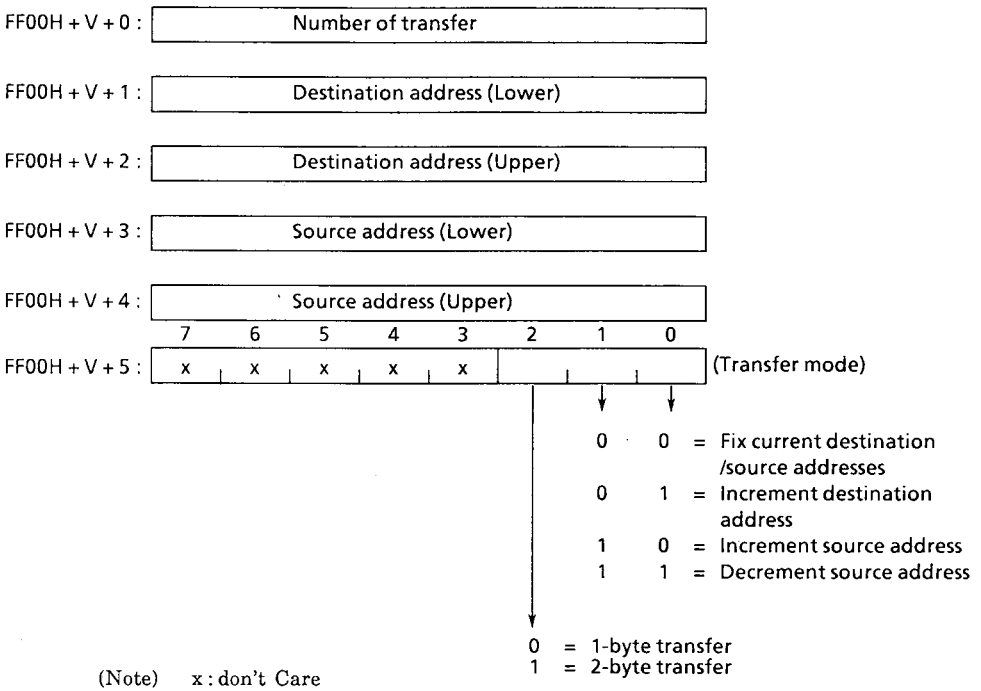


Figure 3.3 (4) Parameters for Micro DMA Processing

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

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

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

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

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

Table 3.3 (2) Addresses Updated by Micro DMA Processing

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

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

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

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

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

Figure 3.3 (5) shows the interrupt processing flowchart.

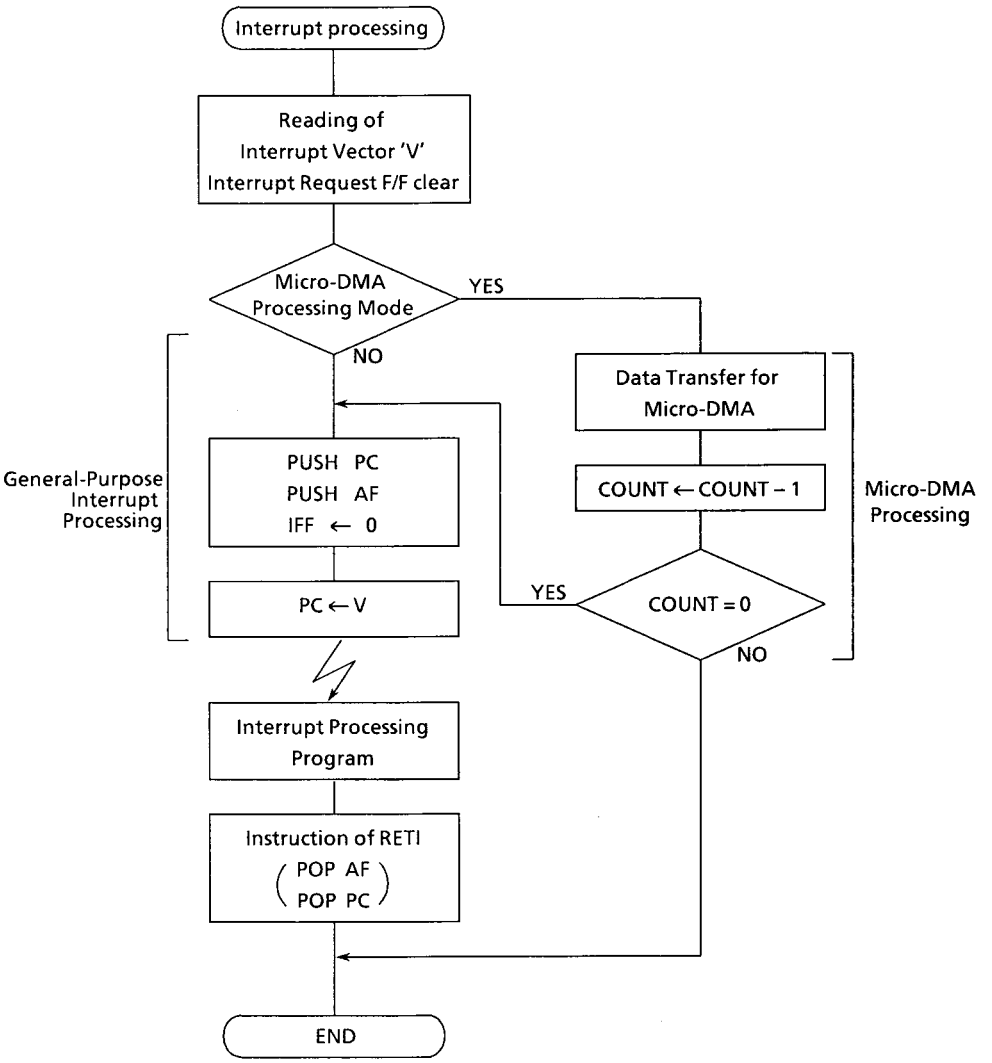


Figure 3.3 (5) Interrupt Processing Flowchart

3.3.3 Interrupt Controller

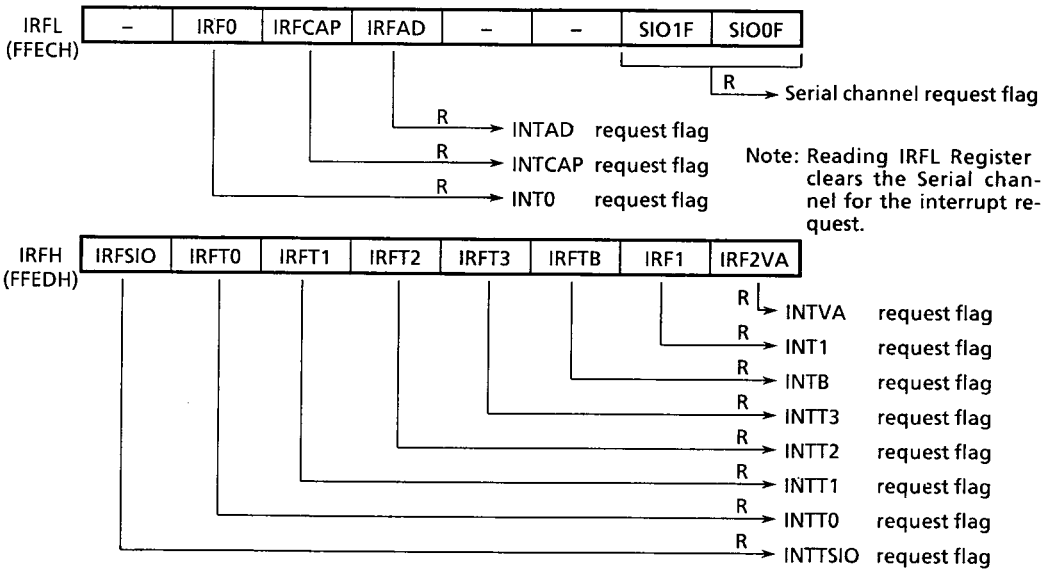
Figure 3.3 (7) outlines the interrupt circuit. The left side of this figure represents an interrupt controller, and the right side comprises the CPU's interrupt request signal circuit and HALT release signal circuit (See "3.4 Standby Function" for the HALT operation).

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

LD (FFECH), 38H/8,

The Interrupt Request Flip-flops for the interrupt channel "INT1" whose vector is 58H is reset to "0".

The status of an Interrupt Request Flip-flops is found out by reading the memory address FFECH or FFEDH. "0" denotes there is no interrupt request, and "1" denotes that an interrupt is requested. Figure 3.3 (6) illustrates the bit configuration indicating the status of Interrupt Request Flip-flops.



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

Figure 3.3 (6) Configration of Interrupt Request Flip-flops

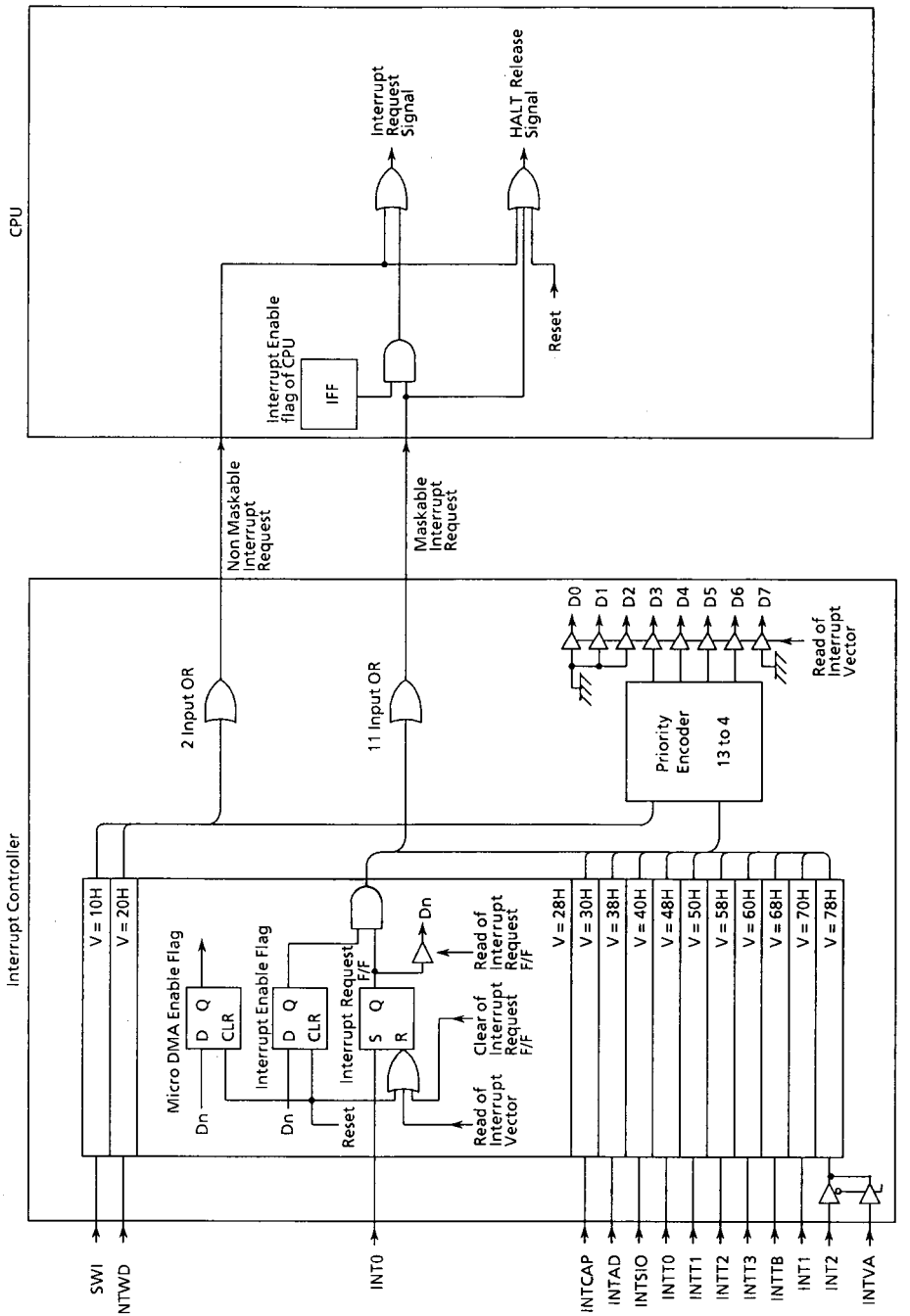


Figure 3.3 (7) Interrupt Circuit

The interrupt enable flags provided for all interrupt request channels are assigned to the memory address FFE9H or FFEAH. Setting any of these flags to "1" enables an interrupt of the respective channel. These flags are initialized to "0" by resetting.

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

Figure 3.3 (8) shows the bit configuration of the interrupt enable flags and micro DMA enable flags.

Interrupt by External input2 (INT2) and that by VASS flag (INTVA) use a common interrupt request channel. The interrupt controller first accepts INTVA after a reset. INT2 can be used by setting the "INTVA/INT2 selection bit" (INT2S: Bit 3 of memory address FFEAH) to "1".

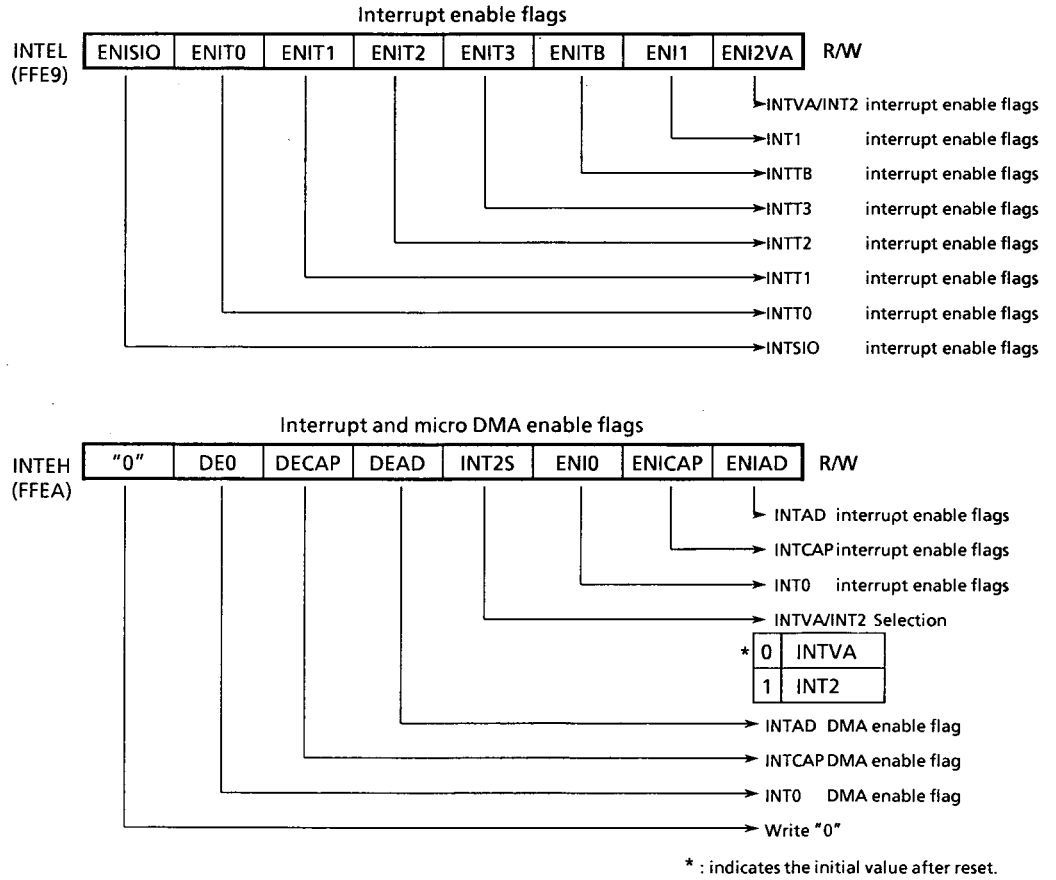


Figure 3.3 (8) Interrupt/Micro DMA Enable Flags

3.4 Standby Function

When a HALT instruction is executed, the TMP91C642 selects one of the following modes as determined by the halt mode set register:

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

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

These HALT state can be released by resetting or requesting an interrupt. Either a non-maskable or maskable interrupt with EI (enable interrupt) condition is acknowledged and interrupt processing is processed.

However the CPU executes the DI (disable interrupt) instruction, a maskable interrupt may be accepted, and the CPU starts executing the instruction following the HALT instruction.

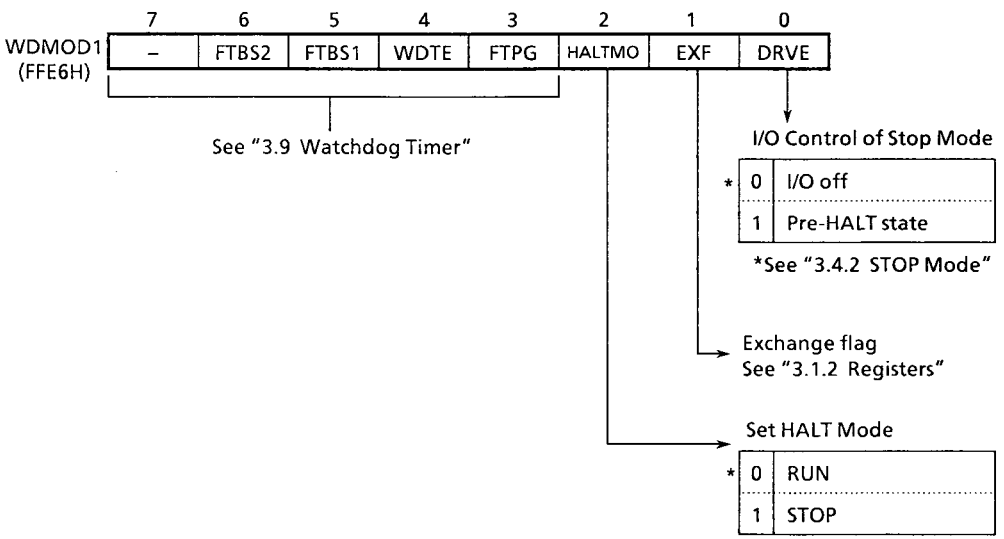


Figure 3.4 (1) HALT Mode Set Register

3.4.1 RUN Mode

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

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

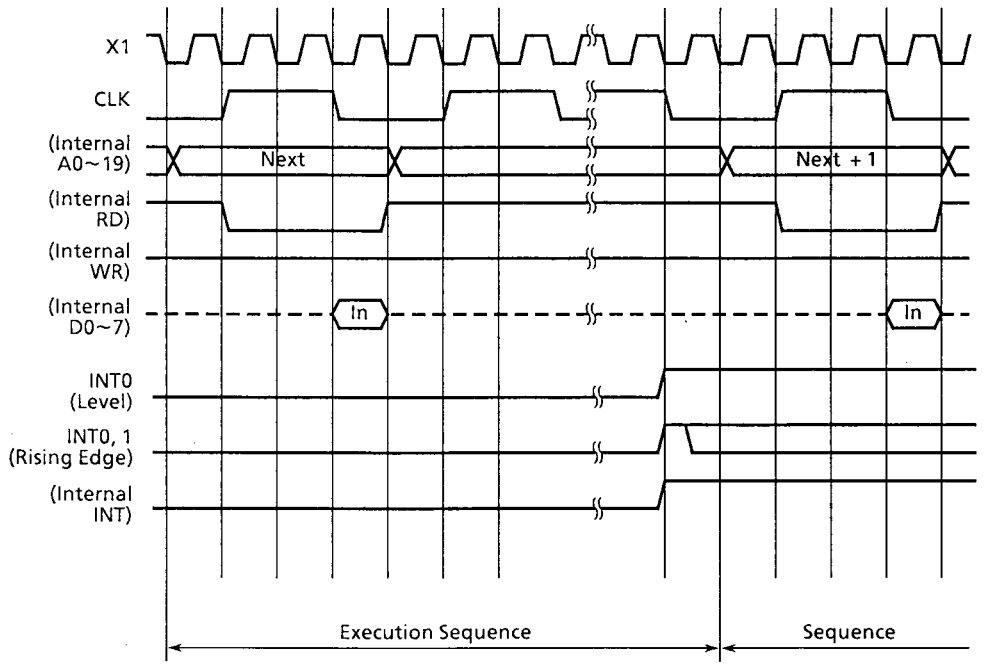


Figure 3.4 (2) Timing Chart for Releasing the HALT State by Interrupt in RUN Mode

3.4.2 STOP Mode

Figure 3.4 (3) is a timing chart for releasing the HALT state by interrupts in the STOP mode.

The STOP mode is selected to stop all internal circuits including the internal oscillator. In this mode, all pins except special ones are put in the high-impedance state, independent of the internal operation of the MCU. Table 3.4 summarizes the state of these pins in the STOP mode. Note, however, that the pre-halt state (The status prior to execution of HALT instruction) of all output pins can be retained by setting the internal I/O register DRVE (Drive enable: Bit 0 of memory address FFE6H) to "1". The content of this register is initialized to "0" by resetting.

When the CPU accepts an interrupt request, the internal oscillator is restarted immediately. However, to get the stabilized oscillation, the system clock starts its output after the time set by the warming up counter.

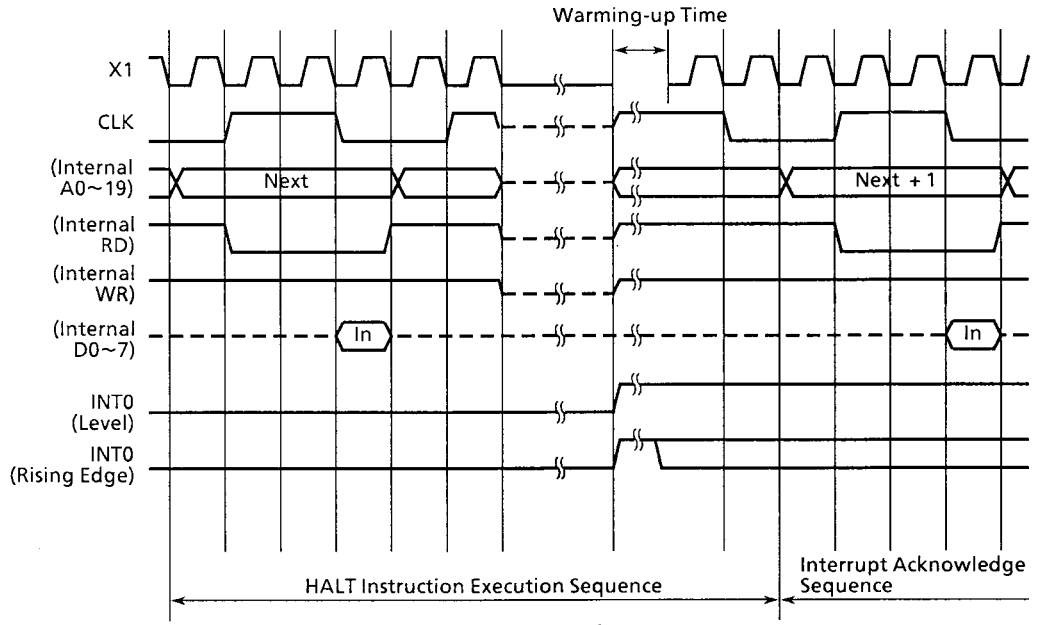


Figure 3.4 (3) Timing Chart of HALT Released by Interrupt in STOP Mode

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

Note : A warming-up time is maximum $(fc/2)^{20}$ sec.

Table 3.4 State of Pins in STOP Mode (DRIVE bit is set to "0")

Pin Name	State
P00~P07 / TPG00~07	Pre-HALT state
P10~P13 / TPG20~23	High impedance
P20~P27 / TPG10~17	High impedance
P30~P32	High impedance
P33~P34	Input
P35 / INTO	Ready for input
P36~P37	High impedance
P40~P47	High impedance
P50~P57	High impedance
P60~P63	High impedance
PW0, PW1	Pre-HALT state
CLK	High impedance
$\overline{\text{RESET}}$	Ready for input
X_1	High impedance
X_2	"1"

Note : Pre-HALT state: The status prior to the execution of HALT instruction is maintained.

3.5 Function of Ports

The TMP91C642 contains total 52 bits input/output ports. These ports function not only for the general-purpose I/O but also for the input/output of the internal CPU and I/O. Table 3.5 describes the functions of these ports.

Table 3.5 Functions of Ports

Port name	Pin name	No. of pins	Direction	Direction set unit	Pin name for internal function
Port 0	P00~P07	8	I/O	Byte	TPG00~TPG07
Port 1	P10~P13	4	I/O	Bit	TPG20~TPG23
Port 2	P20~P27	8	I/O	Bit	TPG10~TPG17
Port 3	P30~P32	3	I/O	Bit	—
	P33	1	I/O	Bit	C – SynC
	P34	1	I/O	Bit	EXT
	P35, P36	2	Input	—	INT0.INT1
	P37	1	Output	—	PWM8 / TO
Port 4	P40	1	I/O	Bit	SCLK0
	P41	1	I/O	Bit	TXD0
	P42	1	I/O	Bit	RXD0
	P43	1	I/O	Bit	SCLK1
	P44	1	I/O	Bit	TxD1
	P45	1	I/O	Bit	RxD1
	P46	1	I/O	Bit	INT2 / TI0
	P47	1	I/O	Bit	TI1
Port 5	P50~P57	8	Input	—	AN00~AN07
Port 6	P60~P63	4	I/O	Bit	AN10~AN13
	P64~P67	4	Input	—	CAP0~CAP3

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

3.5.1 Port 0 (P00~P07)

Port 0 is a general-purpose 8-bit input/output port (P0 : memory address FFC0H). Input and output can be specified by the control register (P0C : memory address FFC1H bit 0). However, P00's output is also controlled by timing pulse generator signals TPG0-15 and by the control register (P0S : memory address FFC1H bit 1) .

Reset makes all control register bits and output latch register bits "0" and enables the input mode.

In addition to its general-purpose input/output functions, Port 0 generates timing pulses TPG00~TPG07.

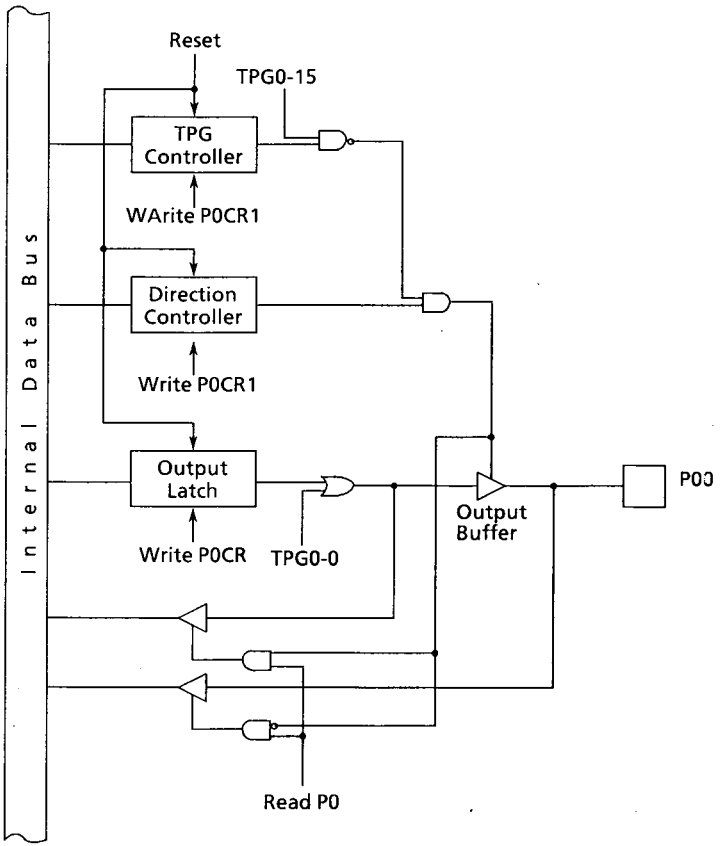
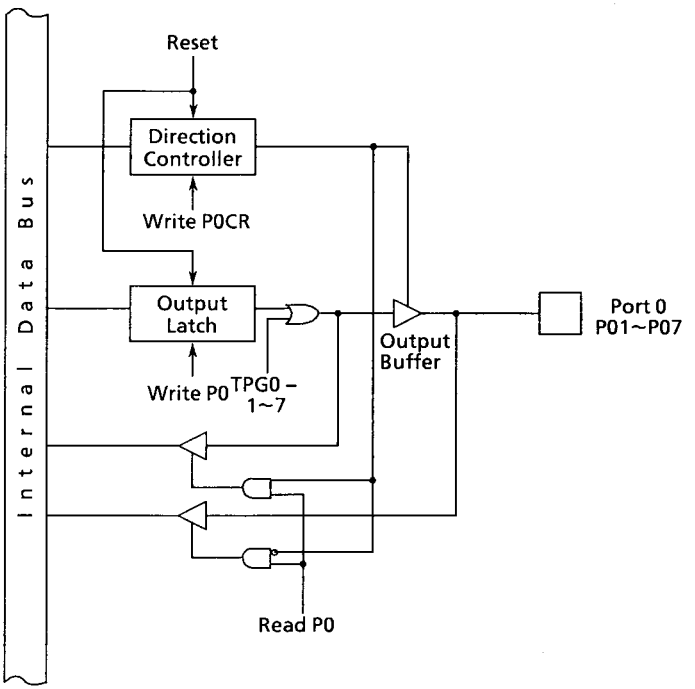
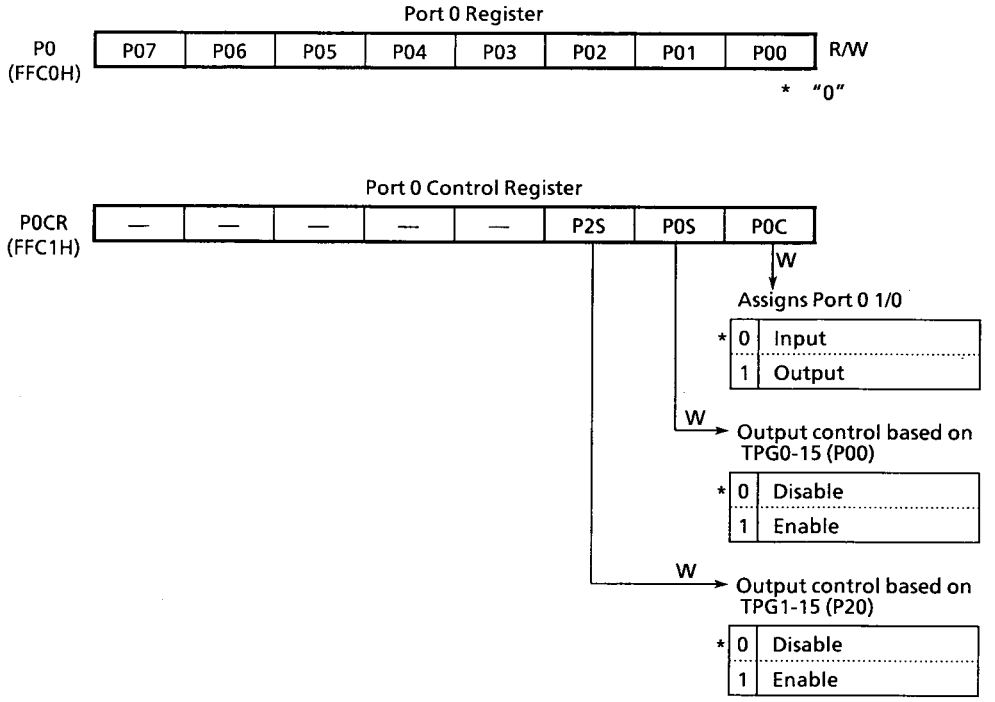


Figure 3.5 (1) -1 Port 00



Note : The structure of P04~P07 output is open drain.

Figure 3.5 (1) -2 Port 01~07



Port P00 status enabled by the Port 0 control register and TPG0-15.

	P0C	P0S	TPG0-15	P00 pin status
*	0	0	0	Input
		0	1	
		1	0	
		1	1	
	1	0	0	Output
		0	1	
		1	0	
		1	1	High impedance

* Initial value after reset

Figure 3.5 (2) Registers for Ports 0

3.5.2 Port 1 (P10~P13)

Port 1 is a general-purpose 4-bit input/output port (P1 : memory address FFC2H) . Input and output can be specified by the control register (PICR : memory address FFC3H bits 0~3) . P10 output is also controlled by timing pulse generator signals TPG0-14/ TPG1-14.

Reset makes all control register bits and output latch register bits "0" and enables the input mode.

In addition to its general-purpose input/output functions, Port 1 generates timing pulses TPG10~TPG13.

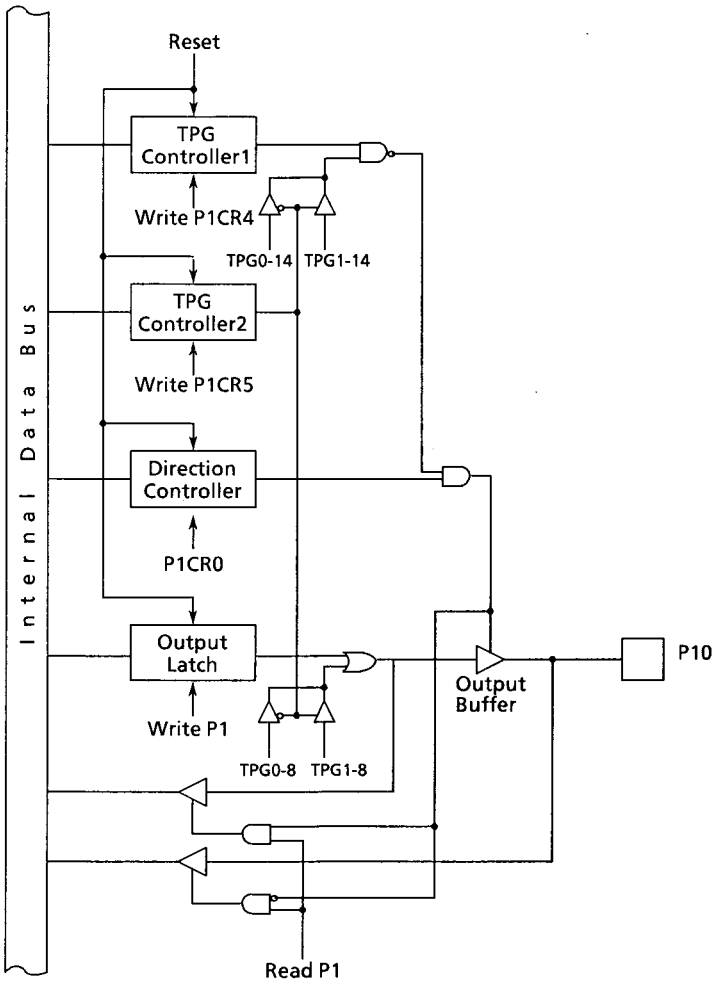


Figure 3.5 (3) -1 Port 10

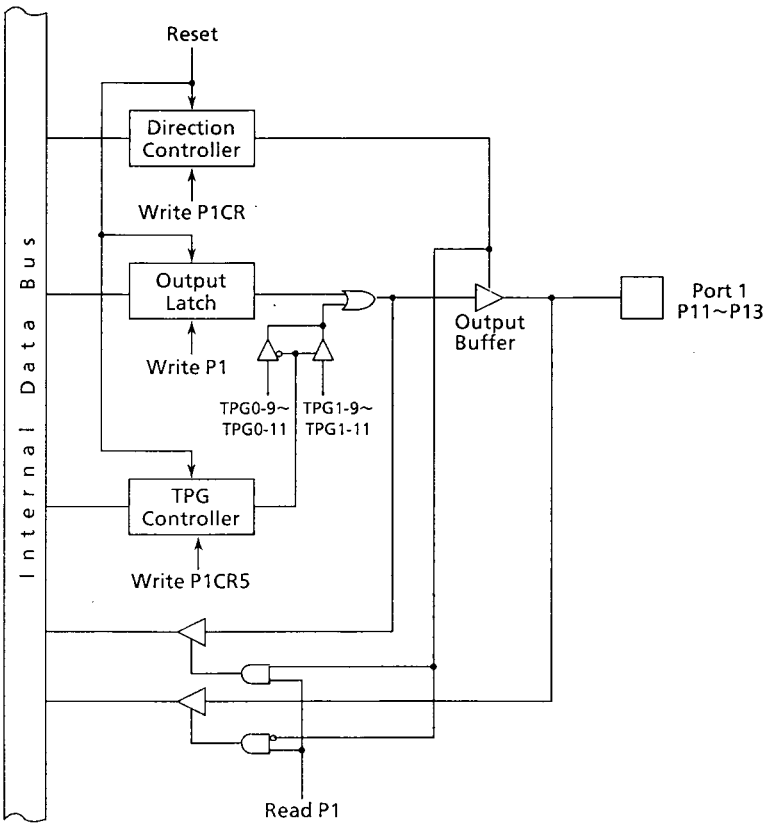
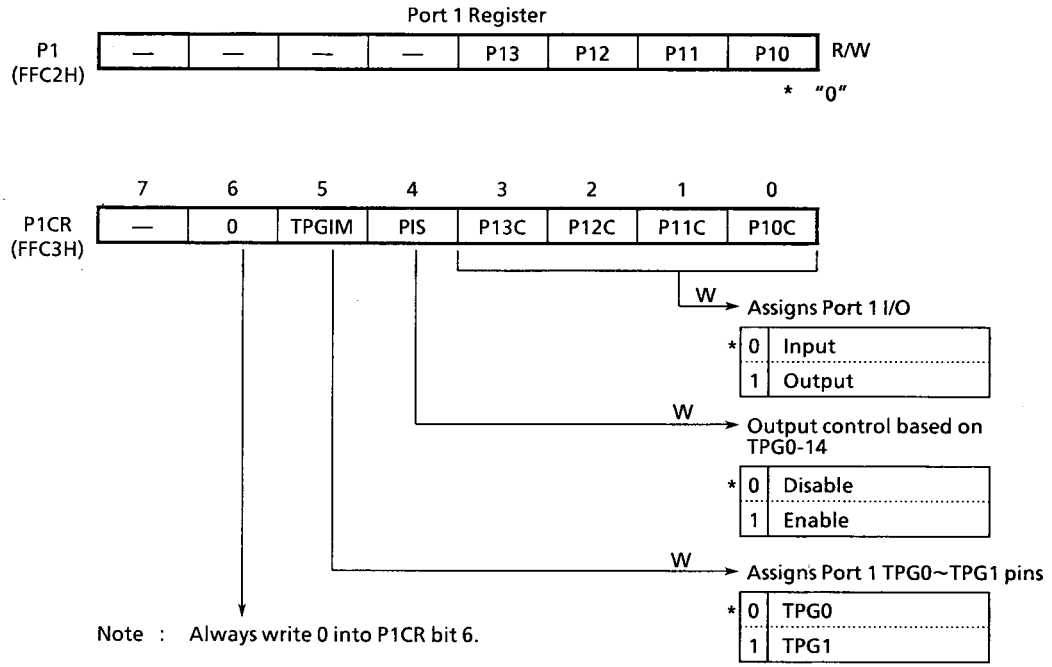


Figure 3.5 (3) -2 Port 11~13



Port P10 status enabled by the Port 1 control register P10C and by TPG0-14/TPG1-14.

	P10C	P1S	TPG0-14 / TPG1-14	P00 pin status
*	0	0	0	Input
		0	1	
		1	0	
		1	1	
	1	0	0	Output
		0	1	
		1	0	
		1	1	High impedance

* Initial value after reset

Figure 3.5 (4) Registers for Ports 1

3.5.3 Port 2 (P20~P27)

Port 2 is a general-purpose 8-bit input/output port (P2 : memory address FFC4H) . Input and output can be specified by the control register (P2CR : memory address FFC5H) . However, P20 output is also controlled by timing pulse generator signal TPG1-15.

Reset makes all control register bits and output latch register bits "0" and enables the input mode.

In addition to its general-purpose input/output functions, Port 2 generates timing pulses TPG20~TPG27.

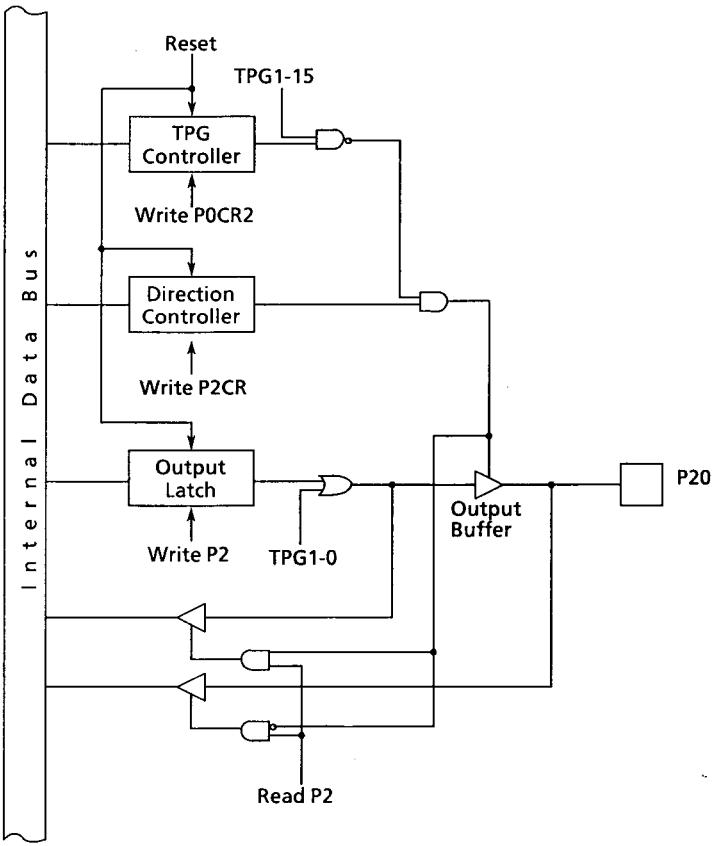


Figure 3.5 (5) -1 Port 20

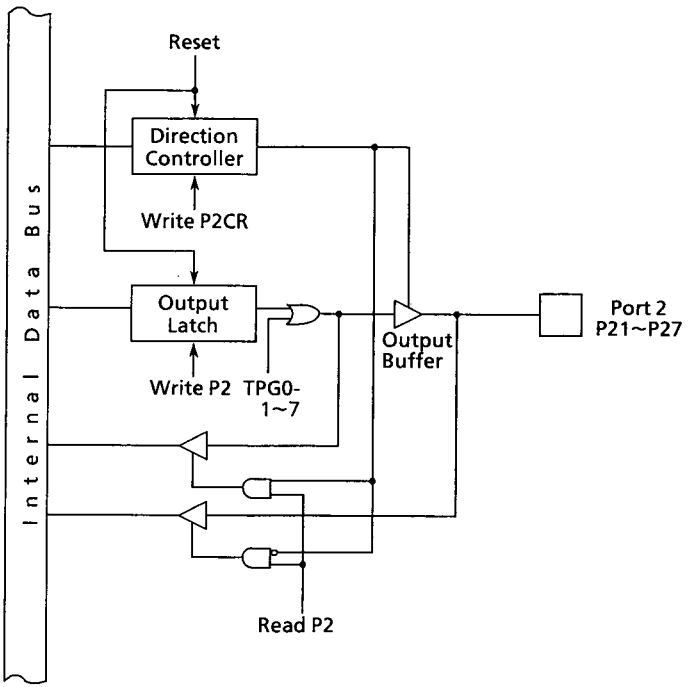
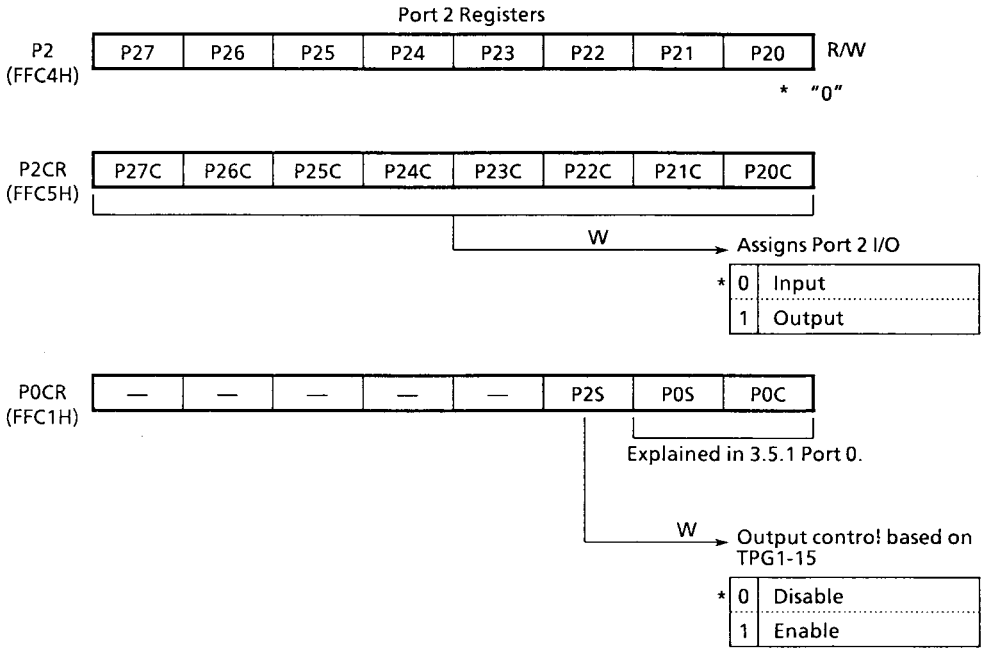


Figure 3.5 (5) -2 Port 21~27



Port P20 status enabled by Port 2 control register P20C and by TPG1-15.

	P20C	P2S	TPG1-15	P20 pin status
*	0	0	0	Input
		0	1	
		1	0	
		1	1	
	1	0	0	Output
		0	1	
		1	0	
		1	1	High impedance

* Initial value after reset

Figure 3.5 (6) Registers for Ports 2

3.5.4 Port 3 (P30~P37)

Port 3 is a general-purpose port (P3 : memory address FFC6H) with 3 bits fixed for input or output and 5 bits which can be specified for input or output.

Input/output specifiable pins P30~P34 are specified by the control register (P3CR : memory address FFC7H bits 0~4).

Reset makes all control register bits and output latch register bits "0" and places input/output pins P30~P32 and input-fixed pins P33~34 in input mode. In this mode, output-fixed pin P37 outputs "0" for PWM8 internal functions. P33 also has an input function for a built-in input signal cyclic separation circuit* ; P34 inputs servo trigger signals. P35 and P36 also function as interrupt control pins INT0 and INT1.

* For horizontal/vertical sync separation of video signals

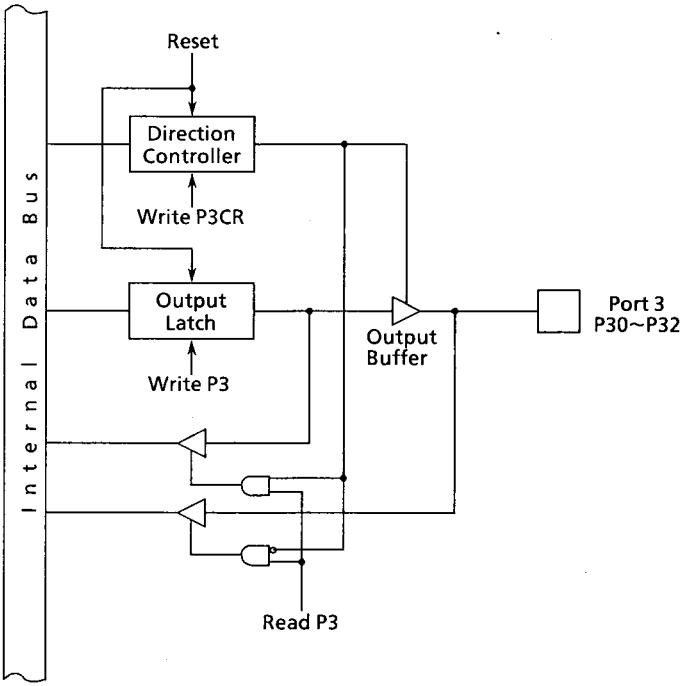


Figure 3.5 (7) -1 Port 30~32

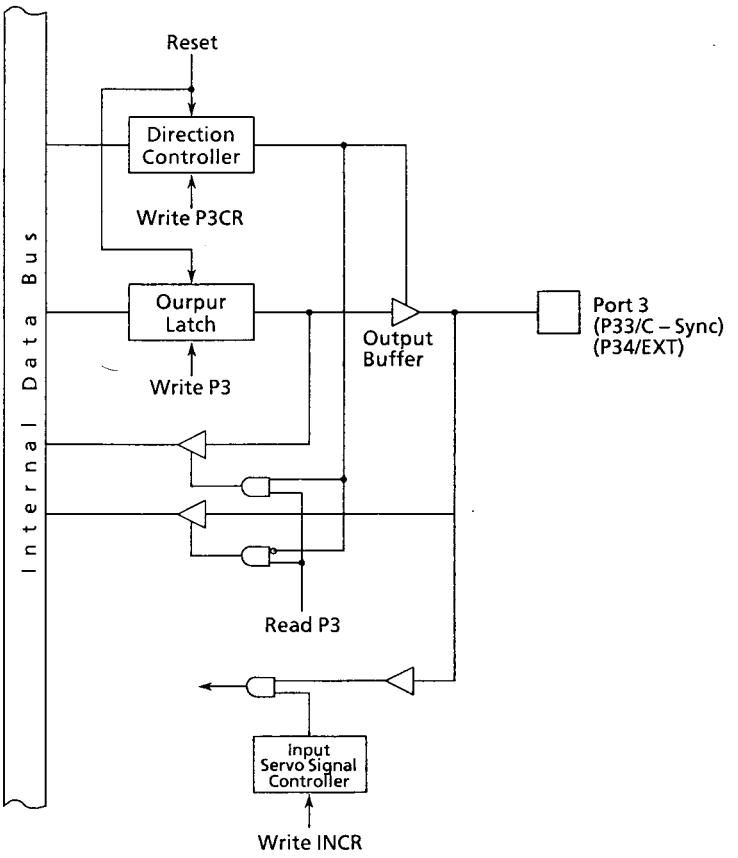


Figure 3.5 (7) -2 Port 33~34

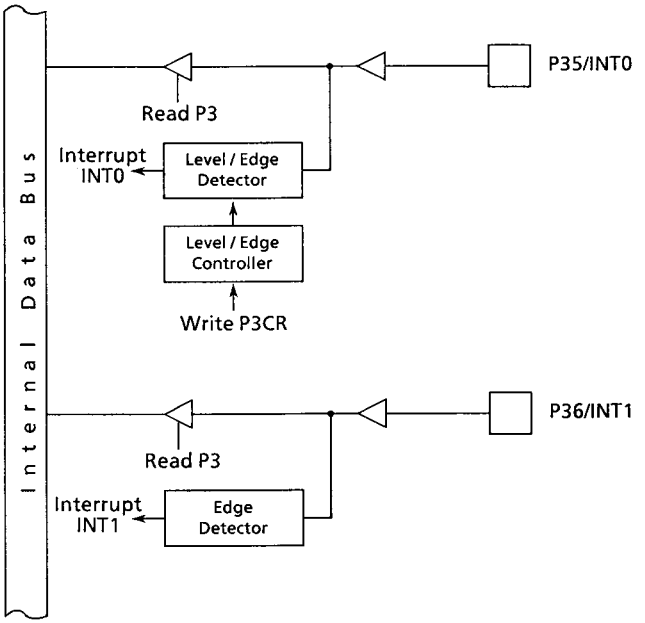


Figure 3.5(7) -3 Port 35~36

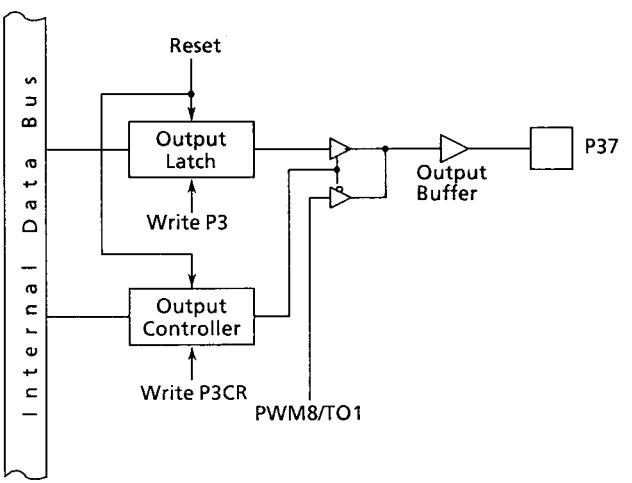
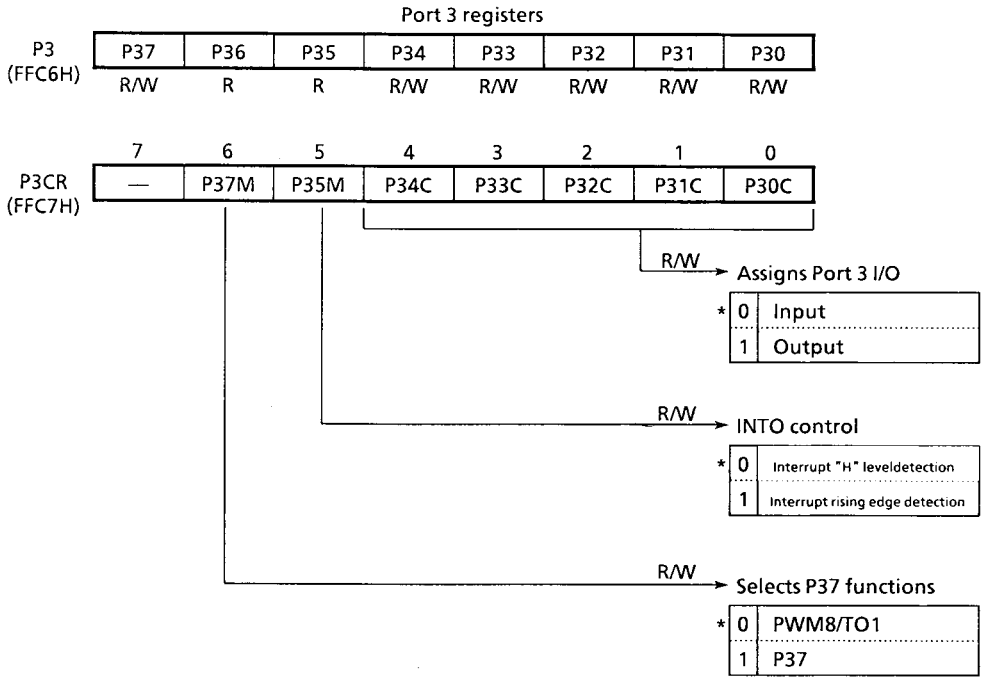


Figure 3.5 (7) -4 Port 37



* Initial value after reset

Figure 3.5 (8) Registers for Port 3

3.5.5 Port 4 (P40~P47)

Port 4 is a general-purpose port (P4 : memory address FFC8H) . Input and output can be specified by the control register (P4CR : memory address FFC9H)

Reset makes all control register bits and output latch register bits "0" and sets the input mode.

In addition to general-purpose input/output functions, Port 4 has serial channel (SIO0/SIO1) for I/O functions (SCLK0/SCLK1, TXD0/TXD1 and RXD0/RXD1) , and input functions for interrupt (INT2) INT2 and timers (TI0 and TI1) .

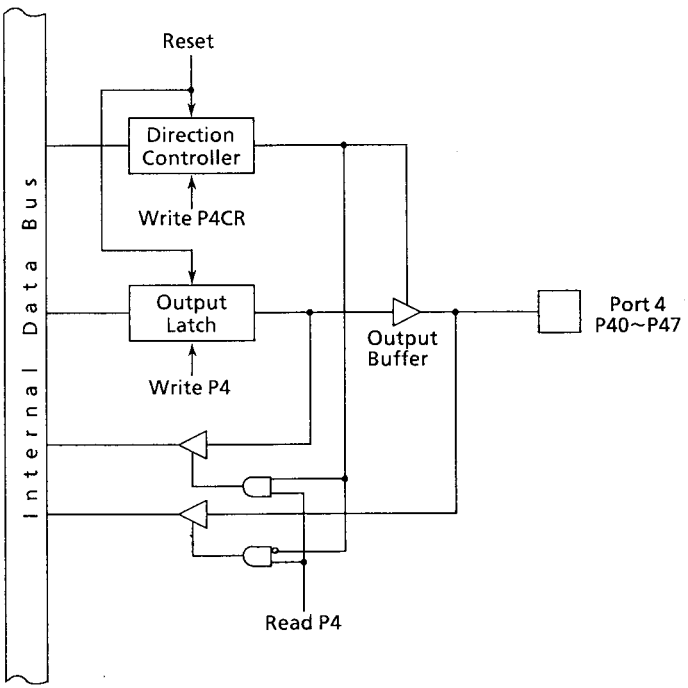


Figure 3.5 (9) Port 4

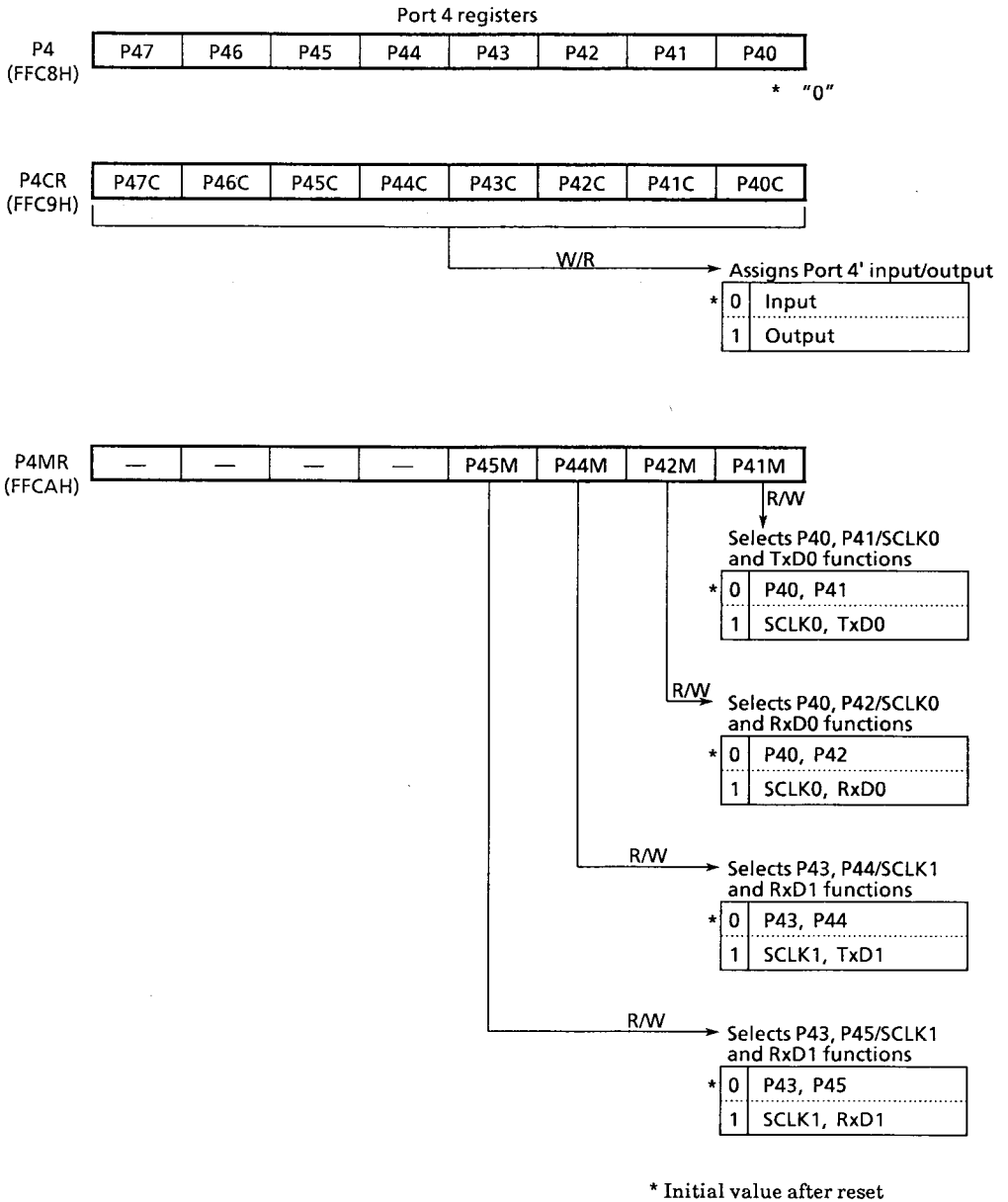


Figure 3.5 (10) Registers for Port 4

3.5.6 Port 5 (P50~P57)

Port 5 is an 8-bit input port (P5: memory address FFCBH) that is also used for analog input at pins AN00~AN07.

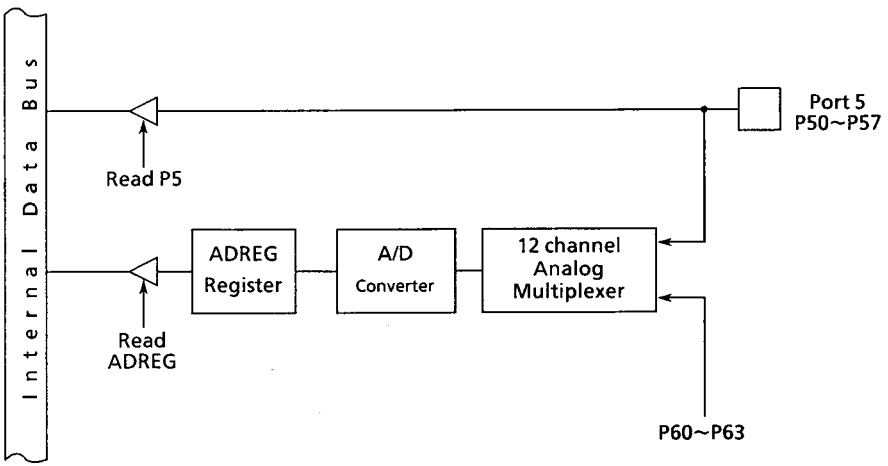


Figure 3.5 (11) Port 5

Port 5 Register

P5 (FFCBH)	P57	P56	P55	P54	P53	P52	P51	P50	R

Figure 3.5 (12) Registers for Port 5

3.5.7 Port 6 (P60~P67)

Port 6 is a general-purpose input/output port (P6 : memory address FFCCH) with 4 bits fixed for input and 4 bits which can be specified for input or output.

Input/output specifiable pins P60~P63 are specified by the control register (P6CR : memory address FFCDH bits 0~3).

Reset makes control register bits and output latch register bits "0" and places them in input mode.

In addition to its general-purpose input/output functions, Port 6 inputs A/D conversion signals AN00~AN03 and servo trigger signals CAP0~CAP3.

Note : When P60~P63 are used for A/D input (P6M : memory address FFCDH's bit 4), Port 6's input/output cannot be used.

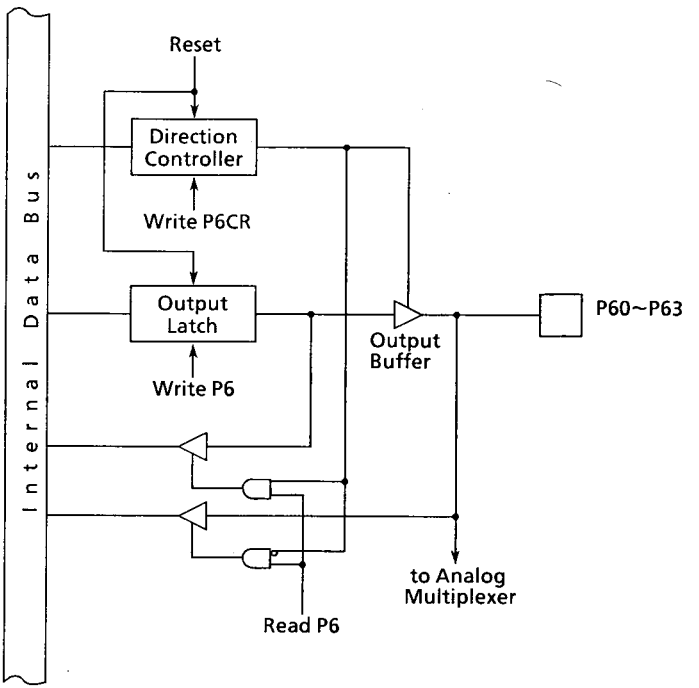


Figure 3.5 (13) -1 Port 60~63

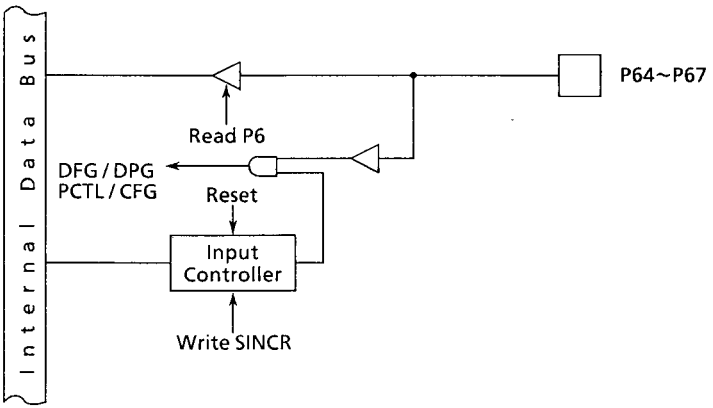


Figure 3.5 (13) -2 Port 64~67

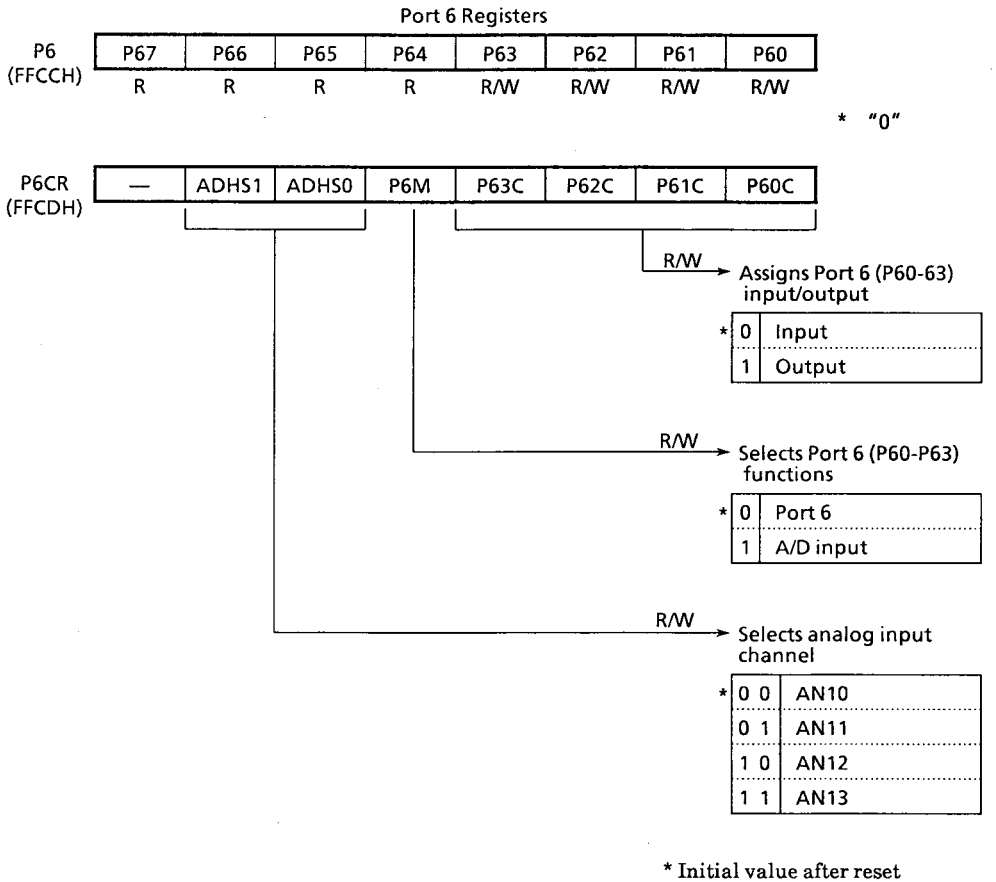


Figure 3.5 (14) Registers for Port 6

3.6 Timer

The TMP91C642 has four 8-bit timer/counters and one 18-bit timer base counter.

The four 8-bit timer/counters can operate independently or timers 0 and 1 can be cascade-connected so that the TMP91C642 can also be used as a 16-bit timer.

3.6.1 8-bit timers 0 and 1

The timer counter of the TMP91C642 timers 0 and 1 each configured from an 8-bit increment counter, 8-bit timer register and 8-bit comparator circuits. These two timer/counters can be operated independently or they can be cascade-connected for use as a 16-bit timer/counter.

The clock pulses input to each timer/counter are received by the timer base counter TBC2, TBC6 and TBC10. Timer 0 is connected to external input pin T10.

The timer/counters are cleared when the timer start control register (TRUN : memory address FFDAH bits 0 and 1) is set to "0" or when there is equivalent signal output.

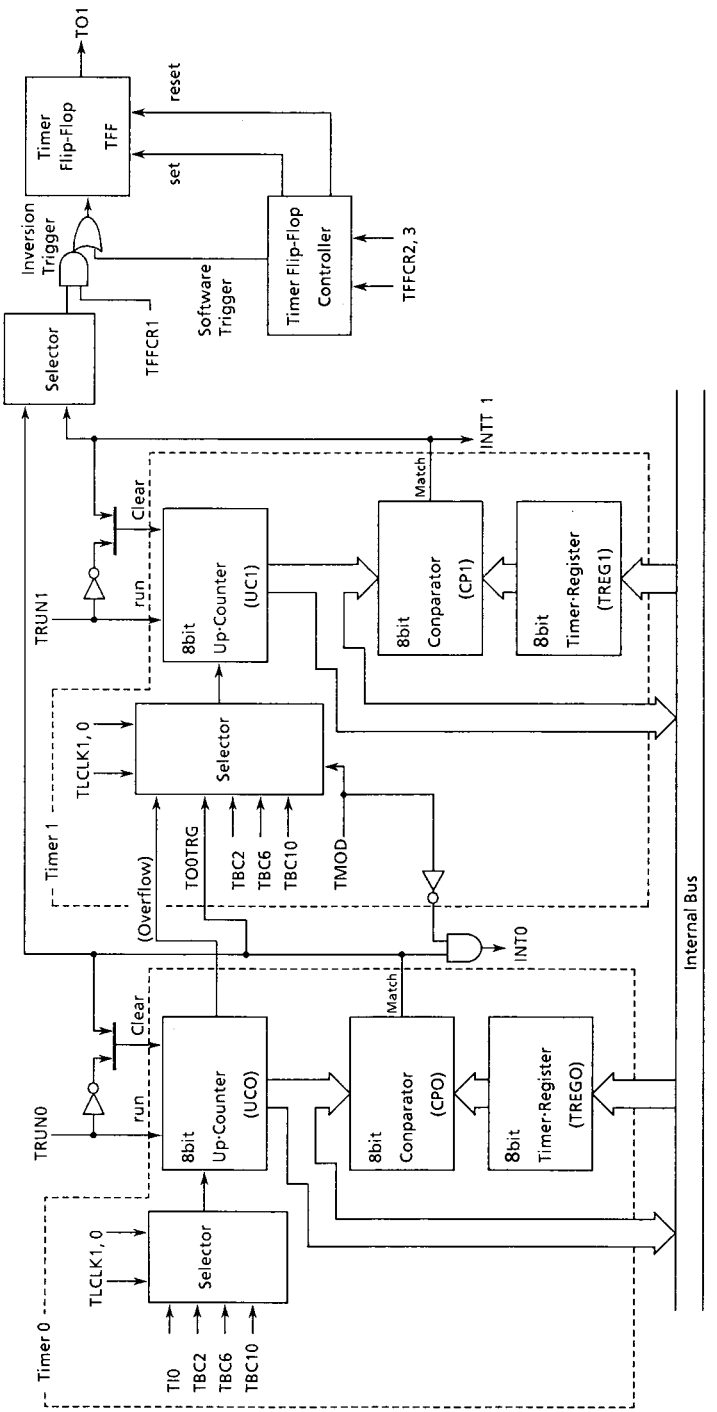
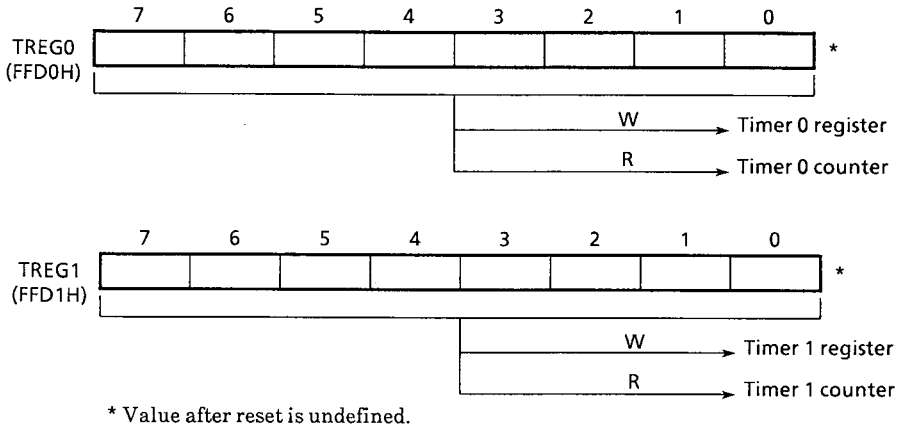
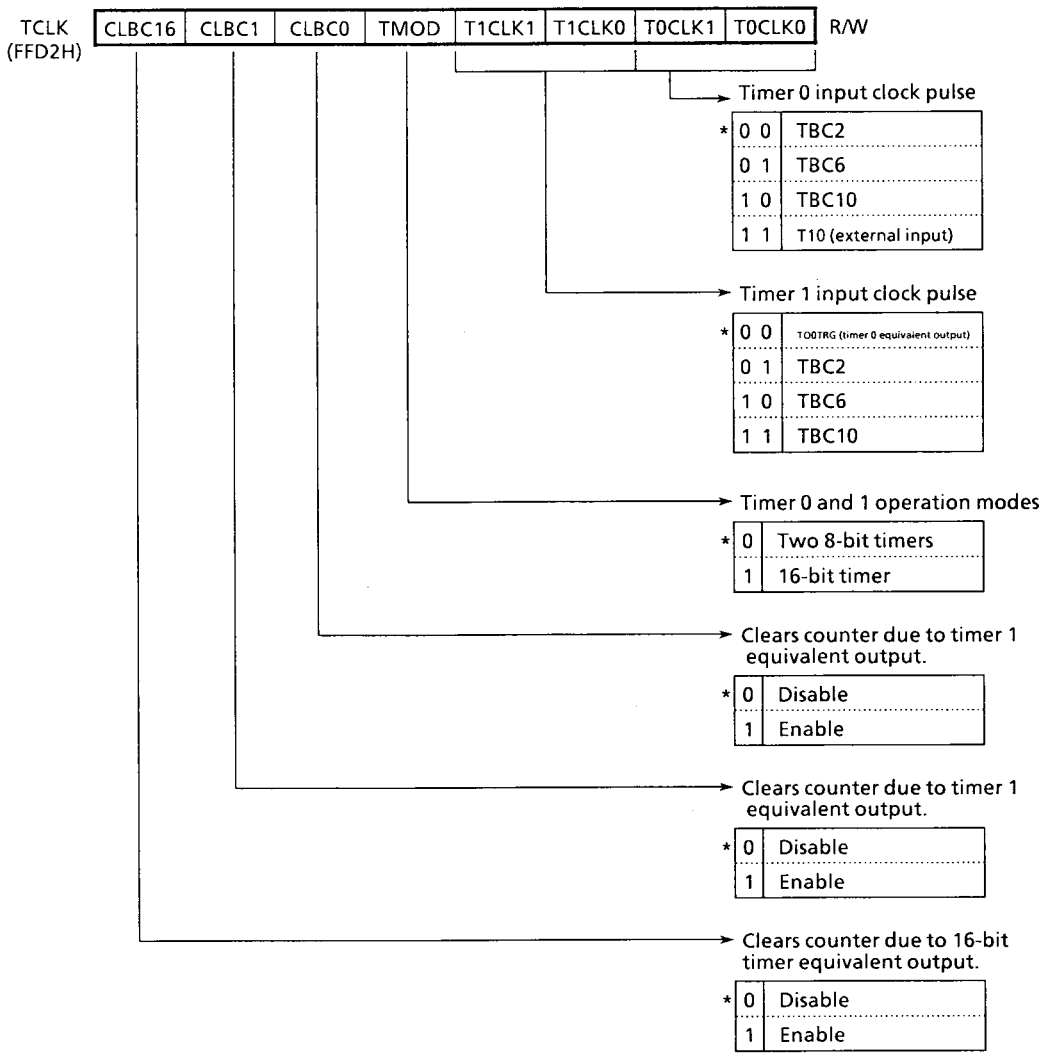


Figure 3.6 (1) Block Diagram of 8-bit Timers (Timers 0 and 1)



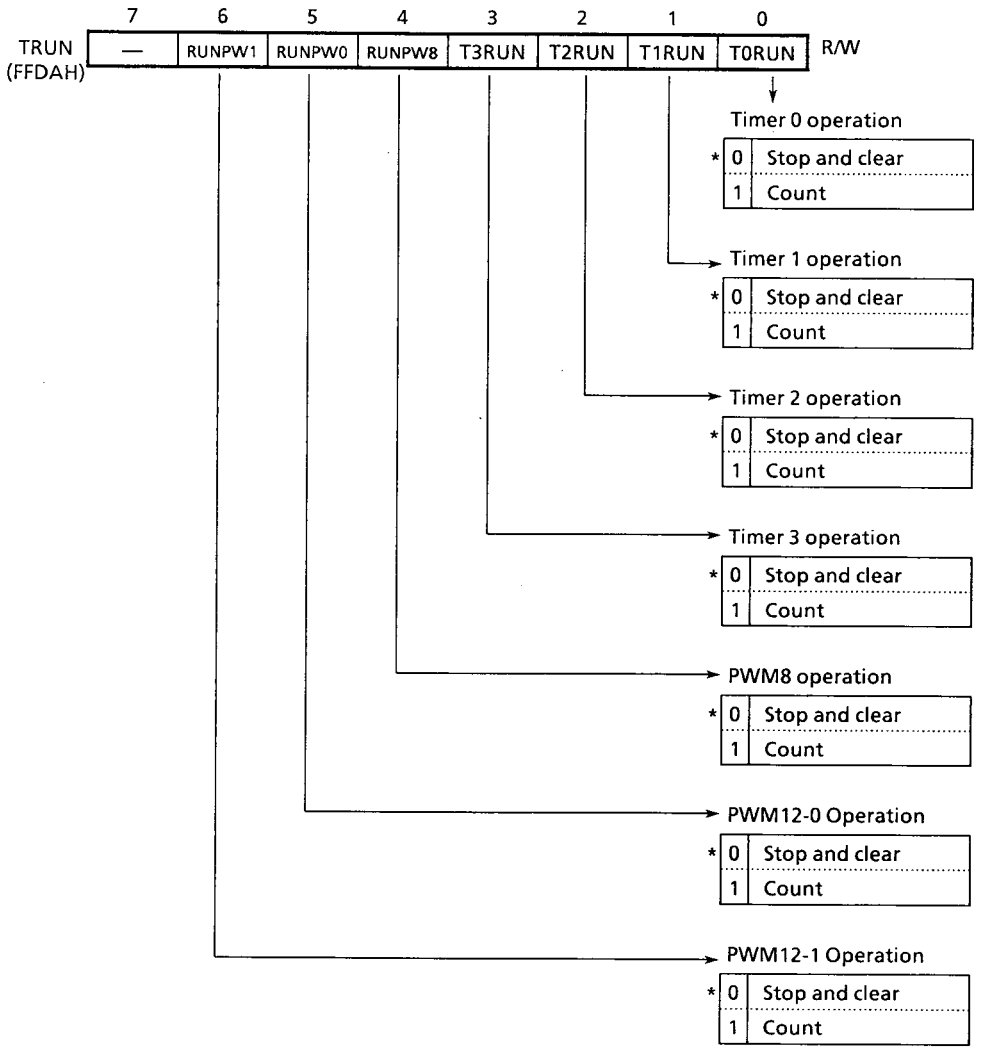
* Value after reset is undefined.

Figure 3.6 (2) -1 Timer 0 and 1 Register



* Initial value after reset

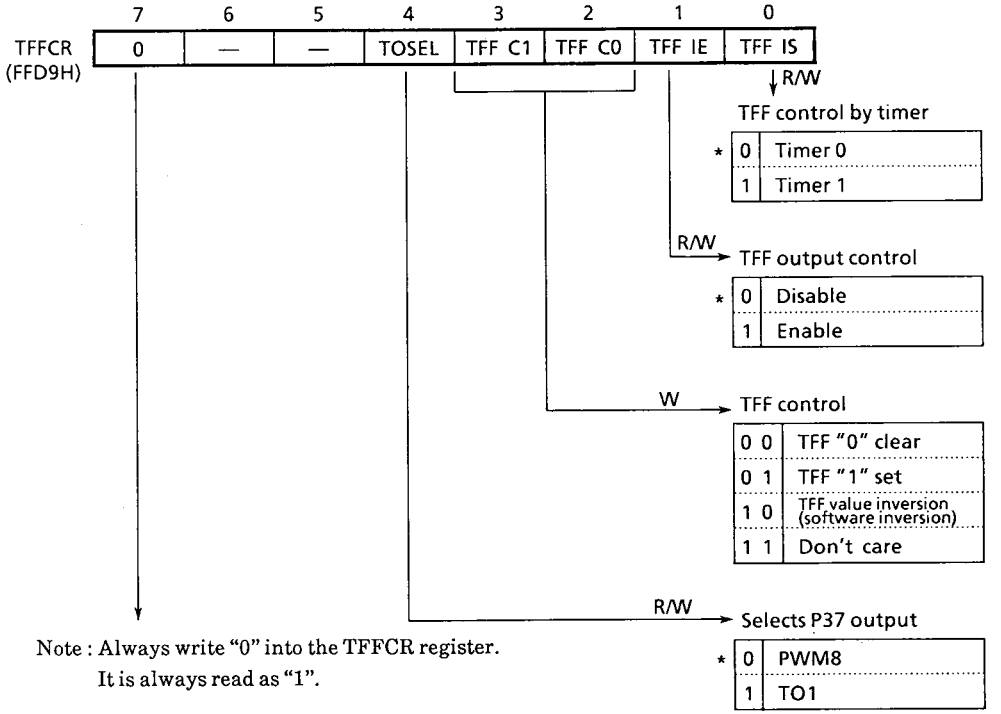
Figure 3.6 (2) -2 Timers Operation



* Initial value after reset.

Note : "T0RUN" and "T1RUN" are simultaneously set to "1" to start timer in 16-bit mode (TMOD = 1).

Figure 3.6 (2) -3 Timer Start Control Register



Note : Always write "0" into the TFFCR register.
It is always read as "1".

* Initial value after reset

Figure 3.6 (2) -4 Timer TFF Control

3.6.2 8-bit Timer 2

Timer 2 is configured from an 8-bit increment counter, timer register and comparator circuits just as is timer 0 and 1.

The clock pulses input to the timer counter are connected to the timer base counter TBC2 and TBC6 to the external input pin T10. They are also connected to the 'C-FG' input pin.

The timer/counter is cleared immediately after device reset, and when there is equivalent signal output by the timer 2 control register (TMR2CR : memory address FFD5H bits 2 and 3) or by software.

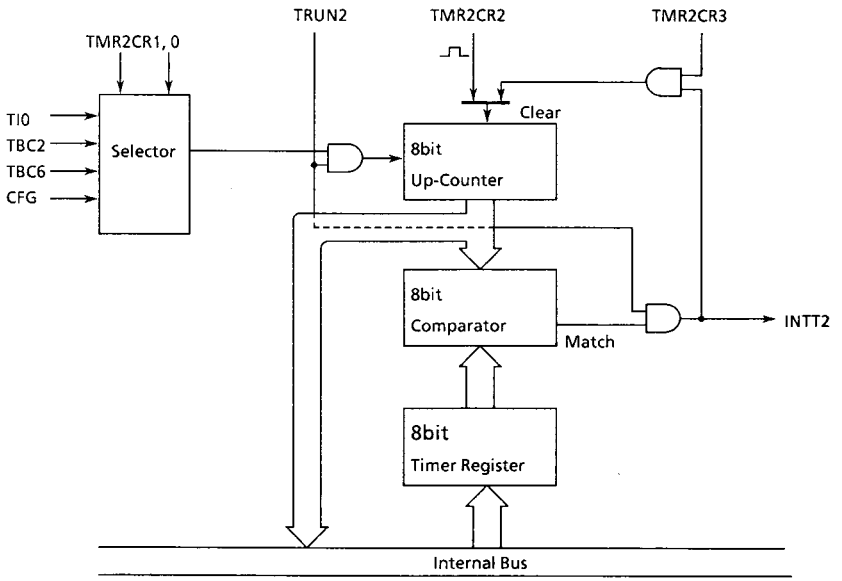
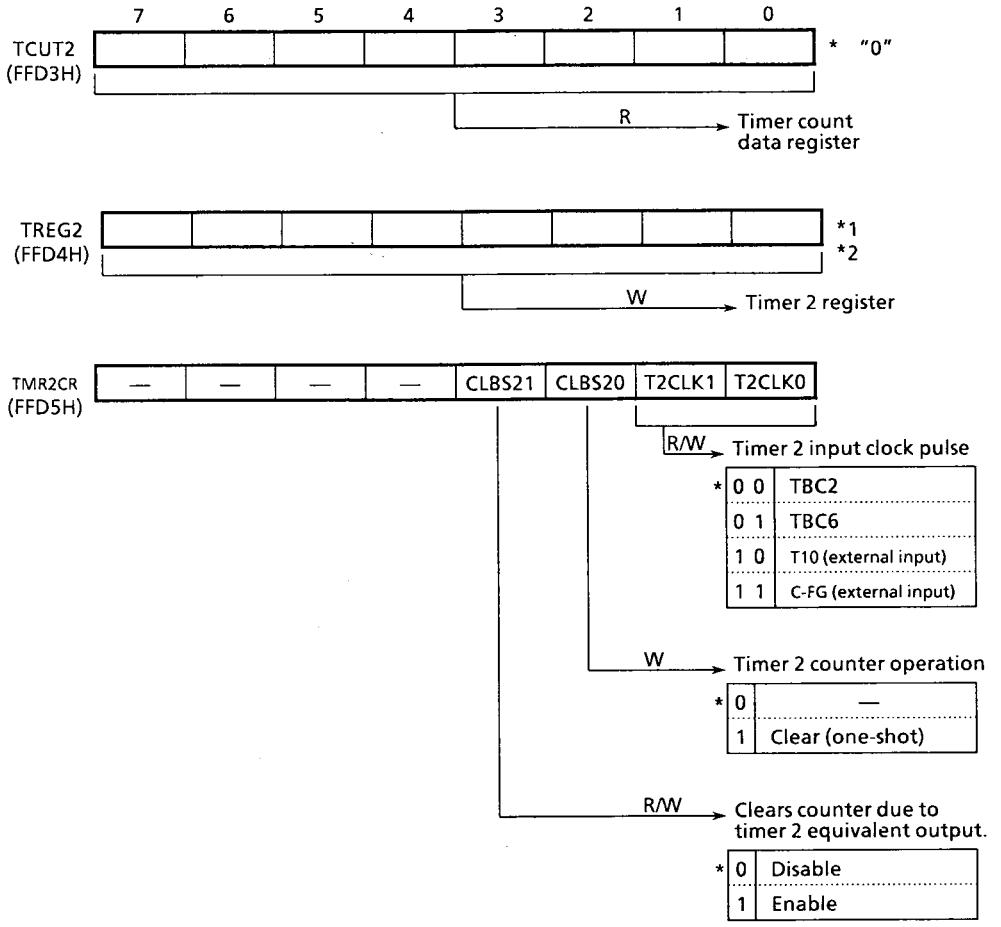


Figure 3.6 (3) Block Diagram of 8-bit Timer 2



* Initial value after reset.

*1 Register values immediately after reset are undefined.

*2 "1" is always read out when a write-only register is read.

For details on the timer start/stop, see Figure 3.6 (2) -2 Timer Start Control Register TRUN.

Figure 3.6 (4)

3.6.3 8-bit Timer 3

Timer 3 is configured from an 8-bit increment/decrement counter, 8-bit counter buffer register and 8-bit comparator circuits.

The clock pulses input to the increment-decrement counter may select either the timer base counter's TBC2 or TBC6 or the external TI1 or P-CTL input pin.

The timer/counter is cleared immediately after device reset, and when there is equivalent signal output by the timer 3 control register (TMR3CR : memory address FFD8H bits 3 and 4) or by software.

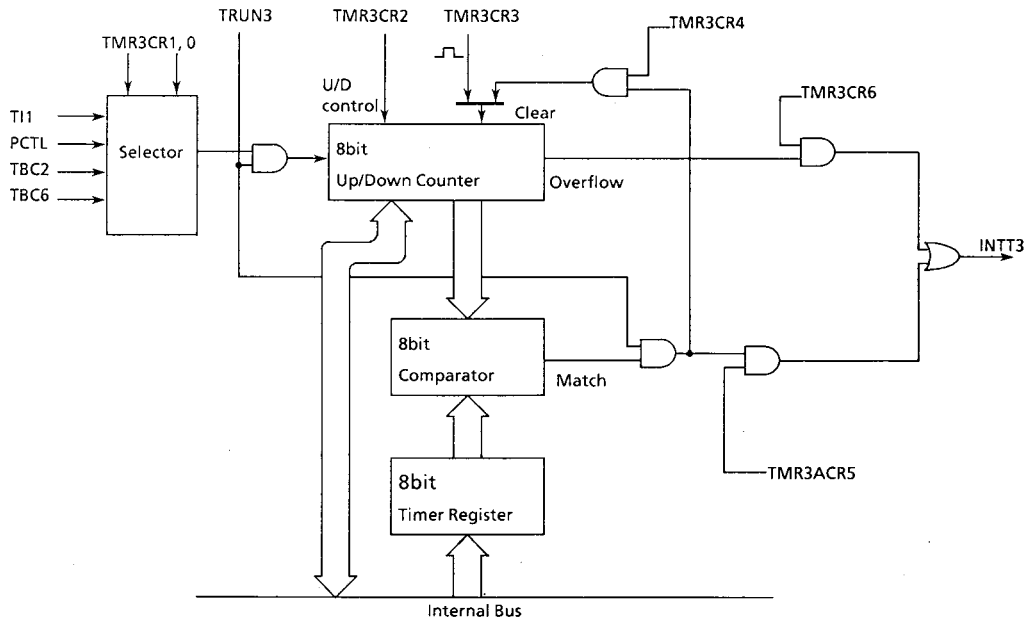
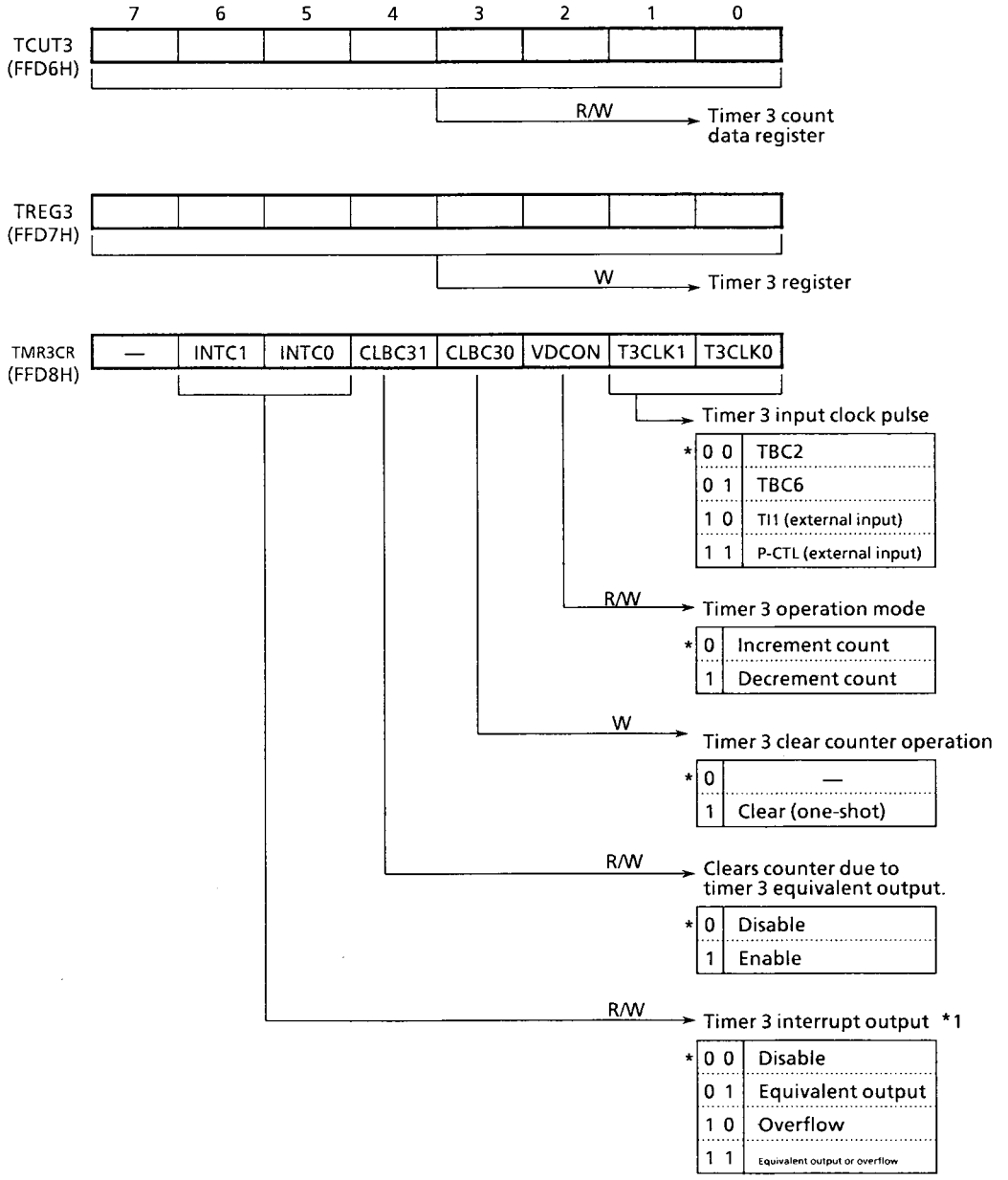


Figure 3.6 (5) Block Diagram of 8-bit Timer 3



* Initial value after reset

* Timer 3 operation must be stopped to switch timer 3 interrupt output.

Figure 3.6 (6)

3.7 Serial Channel

The TMP91C642 contains two channels as serial input/output channels for I/O expansion. The two serial channels are connected to external circuits through two three-pin serial ports: SCLK0 (P40) , TXD0 (P41) and RXD0 (P42) , or SCLK1 (P43) , TXD1 (P43) and RXD1 (P44) .

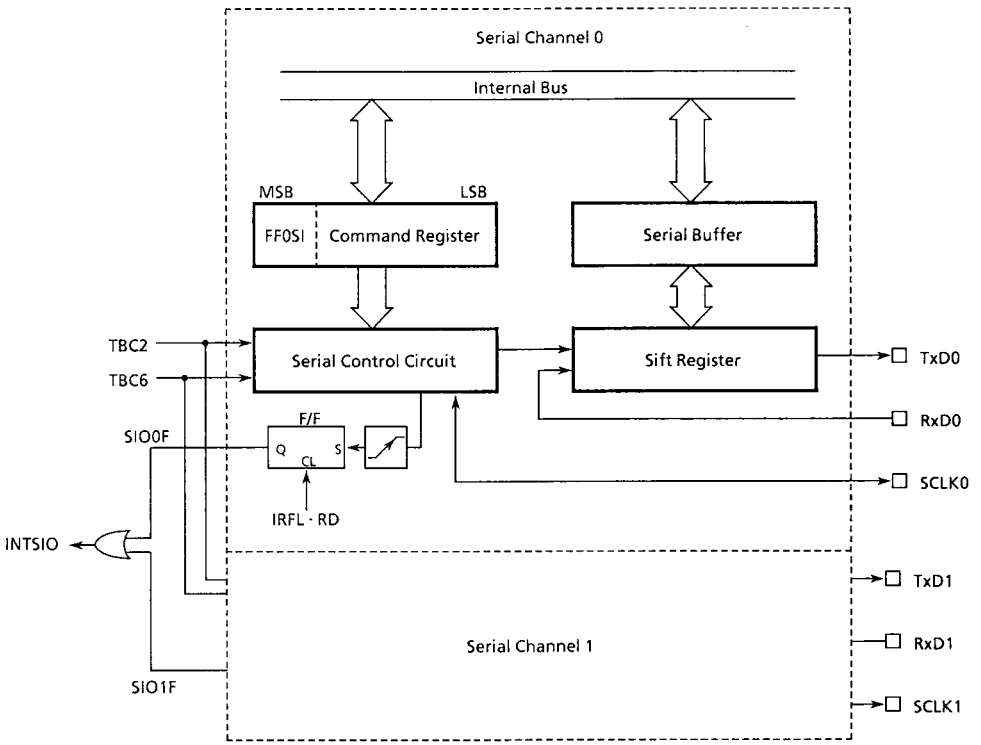


Figure 3.7 (1) Block Diagram of Serial Channels

3.7.1 Serial Clock Pulse

Serial clock pulses make the following selections through the serial channel mode register (SCMOD0 and SCMOD1 : memory addresses FFE3H and FFE5H bits 0-5) .

① Clock source selection

SCLKS0 (SCLKS1 for channel1) selects either an internal or external clock as the clock source.

a. (internal clock)

CLK0SI (CLK1SI for channel 1) selects the speed of either the TBC2 or the TBC6 serial clock. The serial clock pulse is externally output from the SCLK0 pin (the SCLK1 pin for channel 1) .

The serial clock automatically stops after it ends the '1-frame' serial operation. It waits until next serial operation.

b. (external clock)

SCLKS0 uses the clock pulse externally supplied to the SCLK0 pin (SCLK1 pin for channel 1) as the serial clock pulse.

When this pulse is used, selection of serial transfer internal clock pulses CLK0SI or CLK1SI must be set to '0'.

② Shift edge selection

a. Rising edge shift

Data shifts on the serial clock pulse's rising edge (falls at the SCLK0 or SCLK1 pin).

b. Falling edge shift

Data shifts on the serial clock pulse's falling edge (rises at the SCLK0 or No falling edge shift in send mode. SCLK1 pin) .

3.7.2 Explanation of Operations

The send, receive and simultaneous send-receive modes for SC0MOD5 and SC0MOD4 (S0MD0, S0MD1) channel 1 are selected by SC1MOD5 and SC1MOD4 (S1MD0 and S0MD1).

(1) Send mode

The first send data is written into buffer registers SC0BUF and SC1BUF 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 SIO0E and SIO1E starts the send. As the send starts, the send data is synchronized with serial clock pulses and sequentially output from the TxD 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 INTSIO0 or INTSI01 is generated to request the next send data. Channel data at interrupt generation is decided by reading the interrupt request flags, IRFL1 and IRFL0 (memory address FFECH bits 1 and 0) . These flags are cleared to "0" when they are read, so channel data must be decided at the beginning of the interrupt routine.

When the interrupt service program writes the next send data into the buffer register, the interrupt request signal is cleared to "0".

(Internal clock pulses)

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.

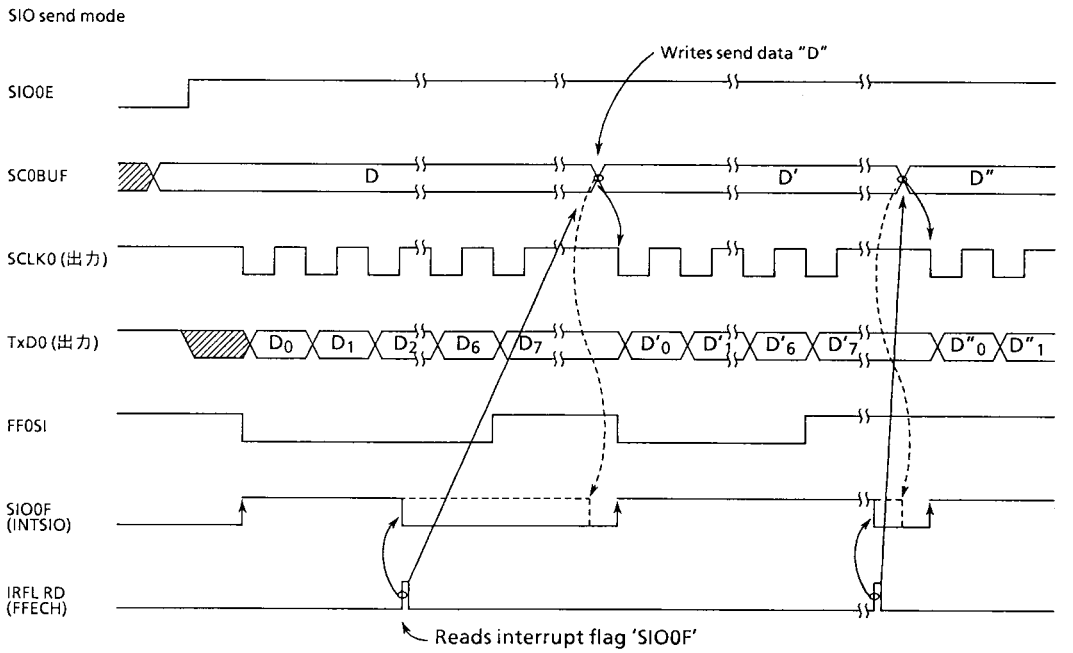
(External clock pulses)

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 interrupt request generation to buffer register data write.

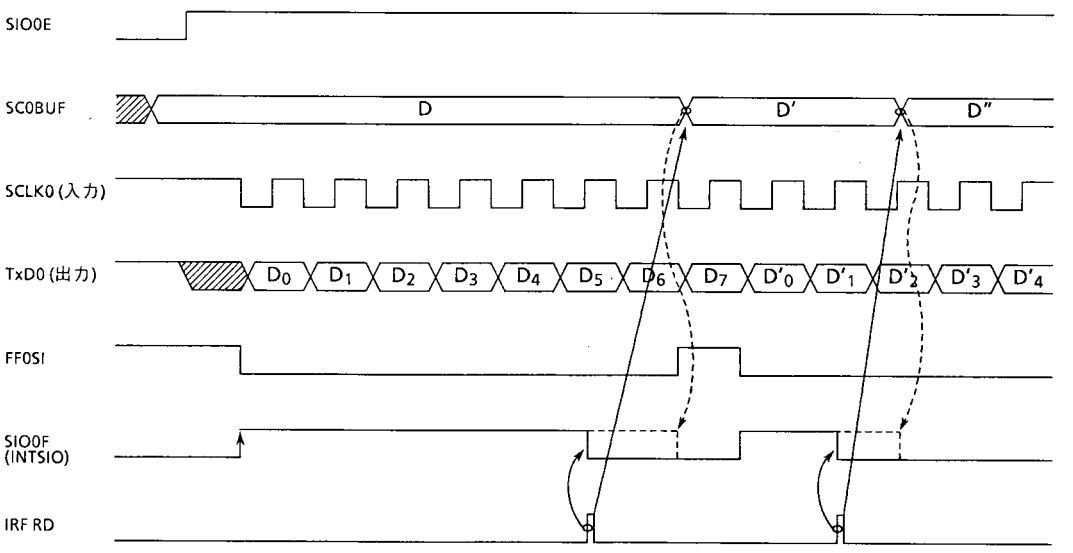
To end a send, the buffer empty interrupt service program disables (clears to "0") serial transfer control register SIO0E or SIO1E instead of writing the next send data. When serial transfer control is disabled, the serial transfer ends when the send data now being shifted out is finished being sent.

The end of send can be determined by the status of serial transfer monitor flag FF0SI or FF1SI. In the external clock operation, the serial transfer control register must be disabled before starting the next send data shift operation.

If the serial transfer control register 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 3.7 (2) Chart of Serial Channel 0 Send Mode Timing

(2) Receive mode

Setting the command register to receive mode, then settin serial transfer control to enable makes receive possible. Shift data is synchronized with serial clock pulses and fetched from the RxD pin. When data is fetched, it is transferred from the shift register to the buffer register and buffer-full interrupt INTSI00 or INTSI01 is generated to request a read of receive data. Channel data at interrupt generation is decided by reading the interrupt request flags IRFL1 and IRFL0 (memory address FFECH bits 1 and 0) .

These flags are cleared to "0" when they are read, so channel data must be decided at the beginning of the interrupt routine.

When the interrupt service program reads 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 cleard, 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 register 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 pulse's rising edge in a rising edge shift, the first shift data must already be input to the RxD 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 SC0BUF and SC1BUF after the send-receive mode is set by the command register. Setting the serial transfer control register SIO0E or SIO1E to 1 enables receiving or sending data. Send data is output from the TxD pin on the rising edge of the serial clock pulse, while receive data is fetched from the RxD 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 INTSIO0 or INTSIO1 is generated to request receive data read.

Channel data at interrupt generation is decided by reading interrupt request flags IRFL1 and IRFL0 (memory address FFECH bits 1 and 0). These flags are cleared to "0" when they are read, so channel data must be decided at the beginning of the interrupt routine.

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 receive data is read and the next send data is written.

(External clock pulses)

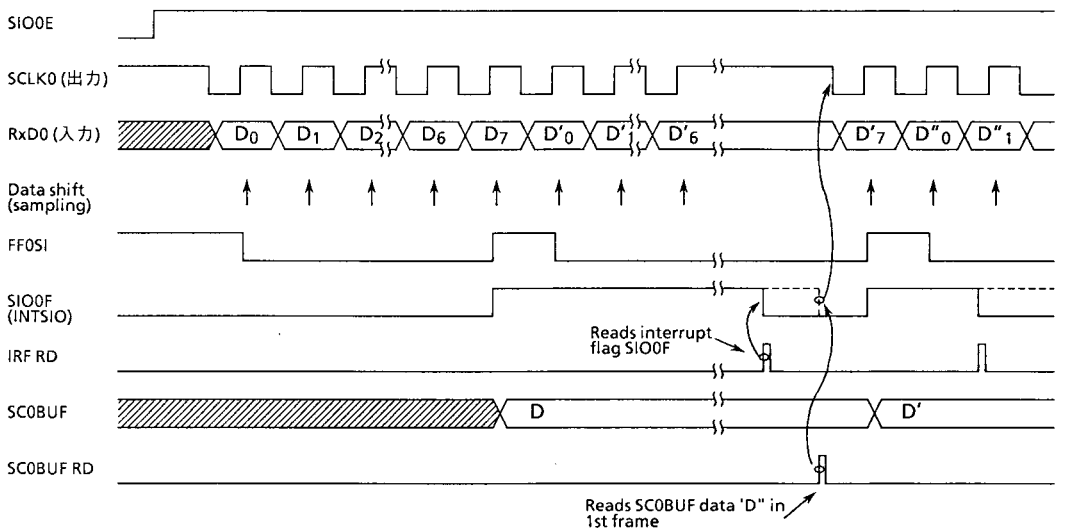
In the external clock operation, the receive data must be read and the next send data written before starting the next shift operation, because the shift operation is synchronized with external supplied clock pulses. The maximum transfer speed of the external clock operation is determined by the maximum 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.

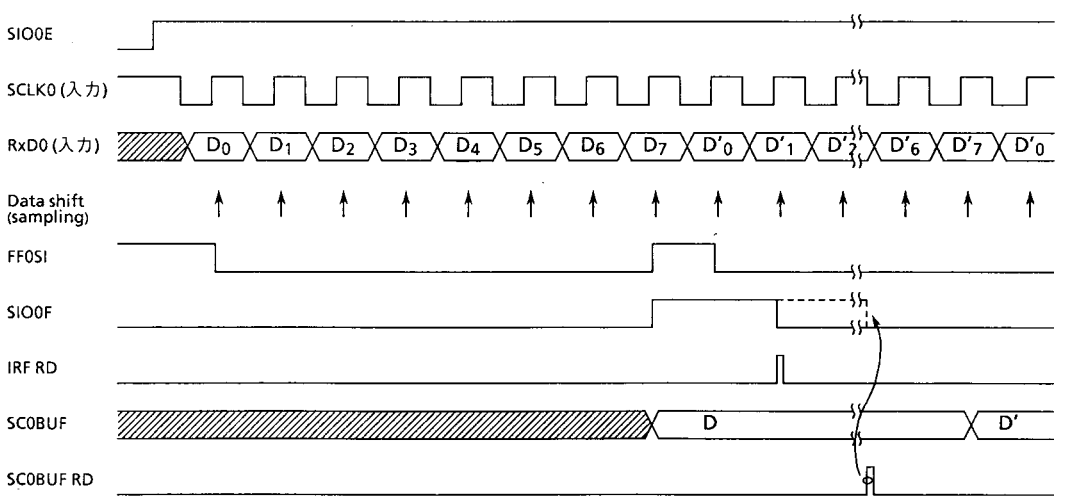
To end send-receive, disable the serial transfer control register. When the serial transfer control register is disabled, send-receive ends after receive data is organized and transferred to the buffer register.

The program checks the end of send-receive by reading serial transfer monitor flags FF0SI and FF1SI.

SIO receive mode (falling edge shift)



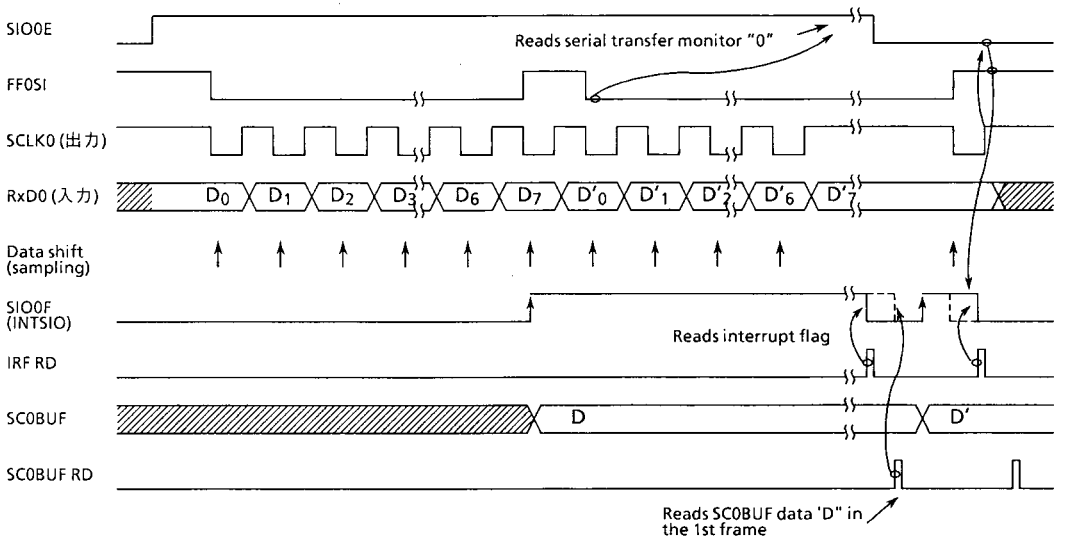
(a) Receive mode internal clock operation (with falling edge shift/wait)



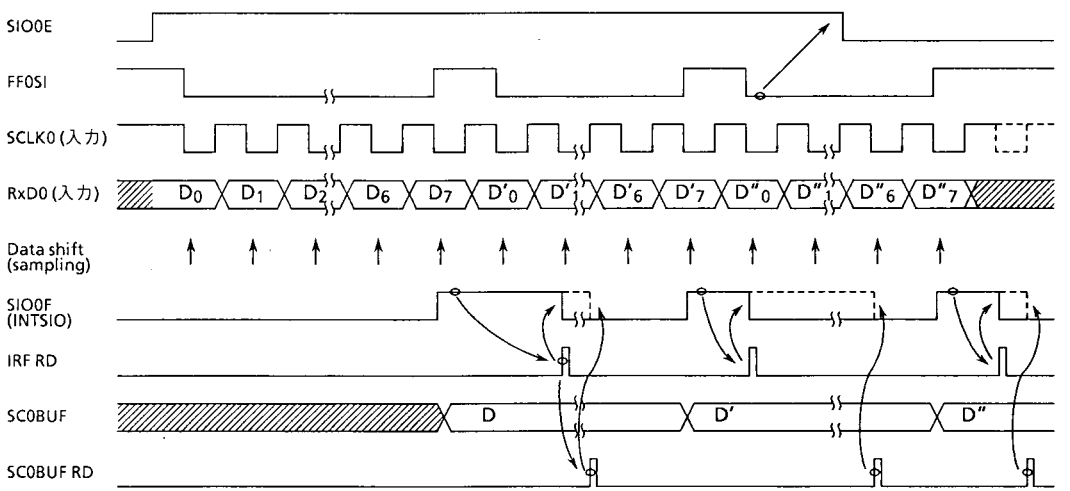
(b) Receive mode external clock operation (falling edge shift)

Figure 3.7 (3) -1 Chart of Serial Channel ϕ Send-Receive Mode (falling edge shift) Timing

SIO Receive mode (rising edge shift)



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



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

Figure 3.7 (3) -2 Chart of Serial Channel ϕ Send-Receive Mode (falling edge shift) Timing

3.7.3 Control Registers

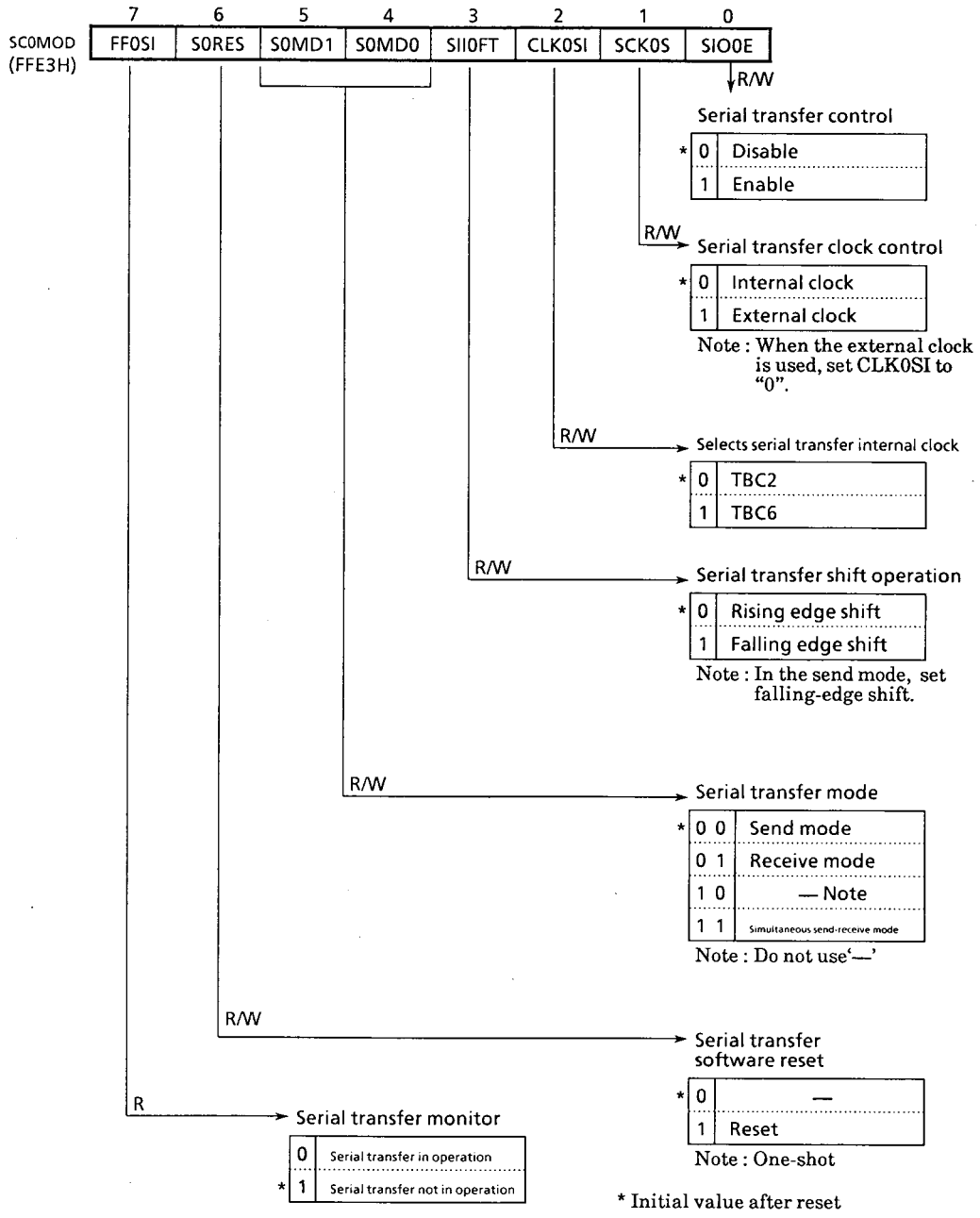


Figure 3.7 (4) -1 Serial Channel 0 Control Register

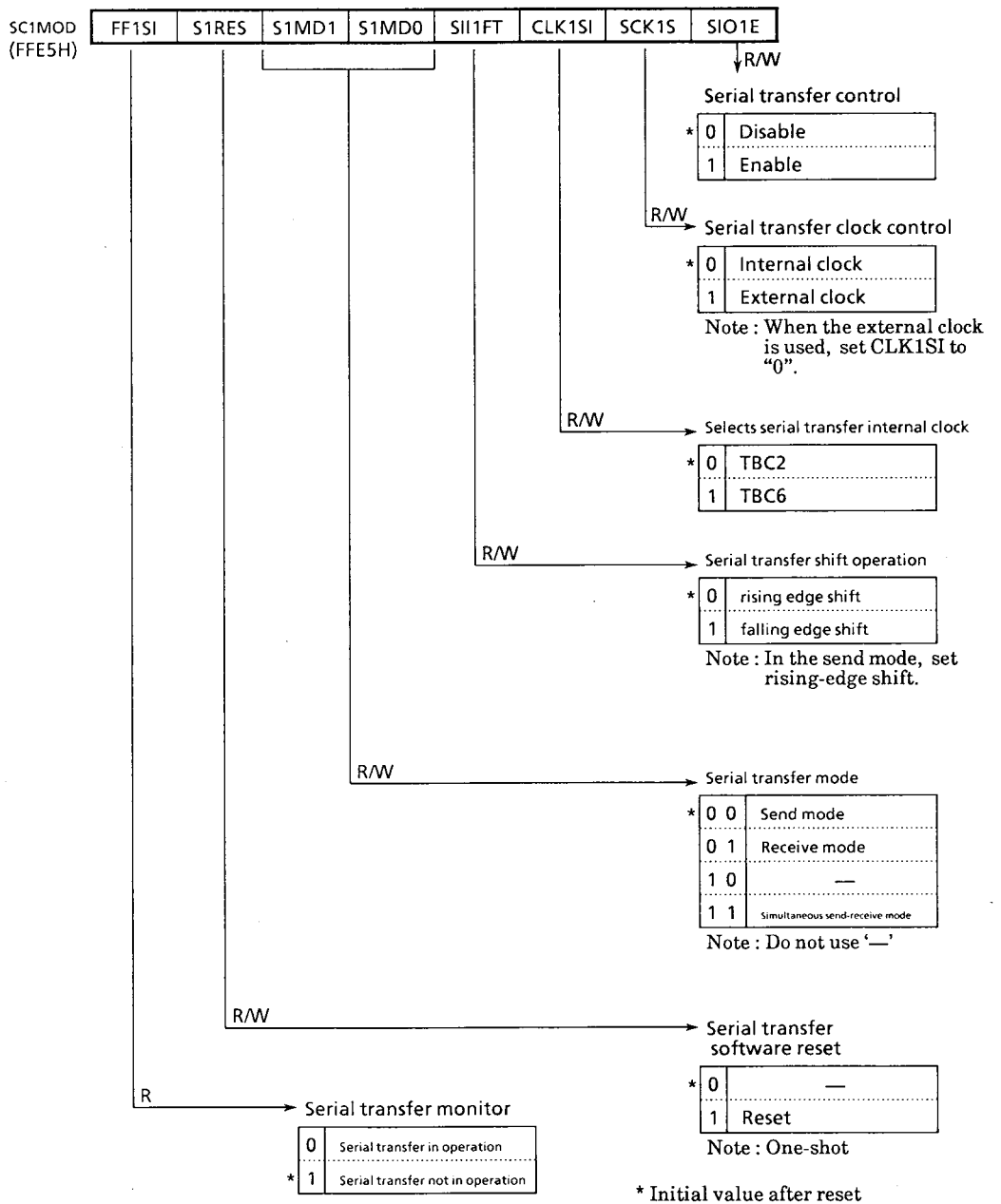


Figure 3.7 (4) -2 Serial Channel 1 Control Register

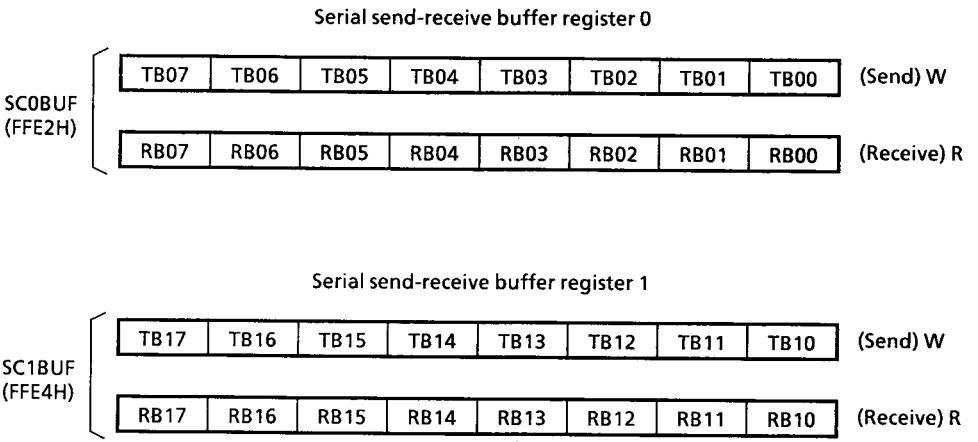


Figure 3.7 (4) -3 Serial Channel Buffer Registers

3.8 Analog/Digital (A/D) converter

The TMP91C642 built-in analog-to-digital (A/D) converter features high speed and high precision, uses an 8-bit successive approximation format, and has 12 (8+4) channel analog inputs.

The 12 channel analog input pins are also used as ports, pins P50 to P57 as input ports and pins P60 to P63 input/output ports. However, if the 4 channels AN10 to AN13 are used as ports and pins P60 to P63 are used dynamically, then AN00 to AN07 cannot be used.

The high-speed A/D conversion ends 95 states (19 microseconds @fc = 10MHz) after it starts.

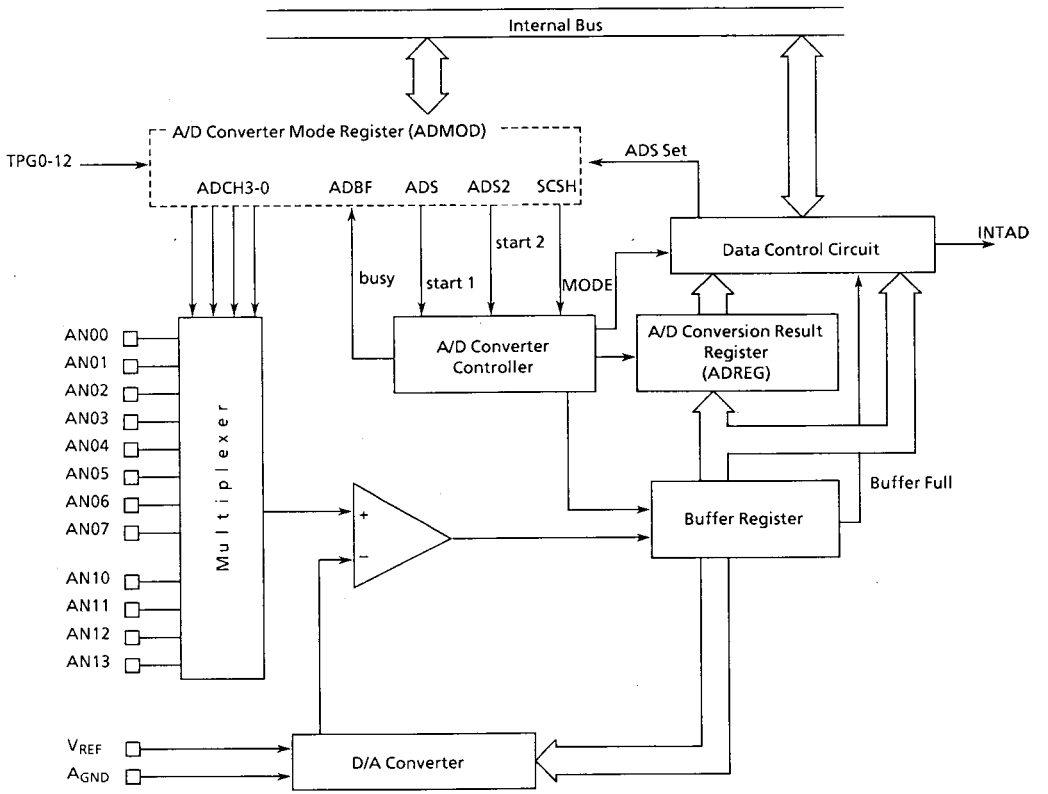


Figure 3.8 (1) Block Diagram of A/D Converter

3.8.1 Control Registers

ADMOD (FFE0H)

7	6	5	4	3	2	1	0
ADS2	ADBF	ADS	SCSH	ADCH3	ADCH2	ADCH1	ADCH0

R/W → Analog Input Channel Select

	P6M = '0'	P6M = '1'
0000	AN00	AN00
0001	AN01	AN01
0010	AN02	AN02
0011	AN03	AN03
0100	AN04	AN04
0101	AN05	AN05
0110	AN06	AN06
0111	AN07	AN07
1000	Cannot be used	AN10
1001		AN11
1010		AN12
1011		AN13
1100		Cannot be used
1101		
1110		
1111		

Note : The analog input channel when A/D conversion mode register SCSH is set to "1" is determined by the port 6 control register ADHS0 or ADHS1.

W → A/D conversion repeat control

0	Normal mode
* 1	Repeat mode

R/W → A/D conversion mode register

* 0	Normal mode
1	A/D conversion dedicated mode

Note : When the A/D conversion dedicated mode is used, the Port 6 control register bit 4 (P6M) must be set to "1". When A/D conversion values are read, SCSH is set to "0".

R → A/D conversion BUSY flag

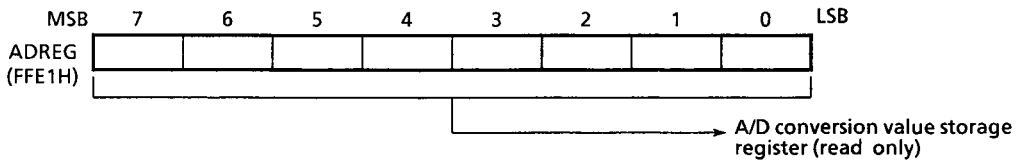
* 0	A/D conversion not in operation
1	A/D conversion in operation

R/W → A/D conversion operation control

* 0	Stop
1	Start

Note : After ADS becomes "1", do not write "1" again.

* Initial value after reset



3.8.2 Explanation of Operations

1) Analog reference voltage

High analog reference voltage is applied to the 'VREF' pin and low analog reference voltage is applied to the 'GND' pin. A/D conversion is made by dividing the reference voltage across VREF-AGND 256 times by a ladder resistor and comparing that with the analog input voltage to make a decision.

2) Analog input channel

A. Normal mode and repeat mode

To start A/D conversion, if the A/D conversion mode register (SCSH : memory address FFE0H bit 4) is set to '0', one analog input channel from the 12 channels AN00-AN07 (pins P50-P57) and AN10-AN13 (pins P60-P63) is selected by the analog input channel select register (ADCH0-ADCH2 : memory address FFE0H bits 0-3).

However, if the four channels AN10-AN13 (pins P60-P63) are being used as ports, the analog input channel is selected only from AN00-AN07. (See 3.5.7 Port 6.)

B. A/D conversion dedicated mode

When the A/D conversion dedicated mode register (SCSH: memory address FFE0H bit 4) is set to '1' (written by the the register write command or by TPG0-12), the A/D conversion input channel is selected from AN10-AN13 by the Port 6 control registers (ADHS1 and ADHS0: at memory address FFCDH bits 6 and 5). However, pins P60-P63 must be set as analog input channels (P6M=-1) at this time. (See 3.5.7 Port 6.)

3) A/D conversion start operation

A. Repeat mode

Writing "1" into ADMOD5 (ADS) starts the A/D conversion in the repeat mode because ADMOD7 (ADS2) is "1" after reset . Once "1" is written in ADS, the A/D conversion repeats and continues to store new A/D conversion values in the A/D conversion value storage register ADREG until a "0" is written into ADS2. This repeated A/D conversion is called the repeat mode.

Writing "0" into ASD2 ends the A/D conversion now in execution or ends A/D conversion when a new A/D conversion value is stored in ADREG.

B. Normal mode

Writing "0" into ADMOD7 (ADS2) followed by writing "1" into ADS starts A/D conversion in normal mode. After one A/D conversion, A/D conversion ends when a new conversion value is stored in ADREG.

Note : When restarting A/D conversion, always check that the ADBF flag is "0". Then start A/D conversion.
Do not write "1" into the ADS register again, after A/D conversion starts.
Before switching to repeat mode, normal mode or analog input channel, end A/D conversion first.

C. A/D conversion dedicated mode

If "1" is set in the ADMOD4 register during repeat mode, the A/D input channel will be switched to a pin from AN10 to AN13, A/D conversion will be performed one more time, and A/D conversion will stop after the A/D conversion in repeat mode ends. An INTAD interrupt signal is generated after A/D conversion in repeat mode only. Reading the A/D conversion value storage register twice after the end of A/D conversion (ADS is "0" or INTAD flag is "1") starts A/D conversion again in repeat mode.

To set "1" in the ADMOD4 (SCSH) register, either write "1" into address FFE0H bit 4 or set TPG0-12 to "1". However, if TPG0-12 outputs are set to "1", they must be set to "0" before the A/D conversion values are read.

4) Read out of A/D conversion values

A. Normal and repeat modes

The results of an A/D conversion are always stored in and read from the ADREG register after the A/D conversion starts (ADS=1).

If the A/D conversion dedicated mode is being used in this instance, always make sure that the A/D conversion mode register ADMOD4 (SCSH) is "0".

B. A/D conversion dedicated mode

The results of A/D conversion when the A/D conversion mode register SCSH is set to "1" during A/D conversion in repeat mode will be stored in the ADREG buffer register when an INTAD interrupt signal is generated.

For this reason, if the A/D conversion mode register ADMOD4 (SCSH) is "1", the read must take place after making sure that the INTAD flag (IRFAD : memory address FFECH bit 4) is "1".

The values that are read out first from the ADREG register are the A/D values converted in A/D conversion dedicated mode. The values read out next from the ADREG register are the A/D values converted in repeat mode, before the values were A/D converted in A/D conversion dedicated mode .

When the ADREG register is read twice, A/D conversion is performed again in repeat mode.

Setting the A/D conversion mode register

There are two methods of setting the A/D conversion register. One is to write "1" into the SCSH register, the other is to set pin TPG-12 to "1".

Note : When the SCSH register is set by TPG0-12, always set using the minimum pulse width.

3.9 D/A Conversion (Pulse Width Modulation) Output

The TMP91C642 has 3 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.

3.9.1 PW0/PW1

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.

(when fc=10MHz)

$T_M \doteqdot 1.22KHz$

$T_S \doteqdot 39.062KHz$

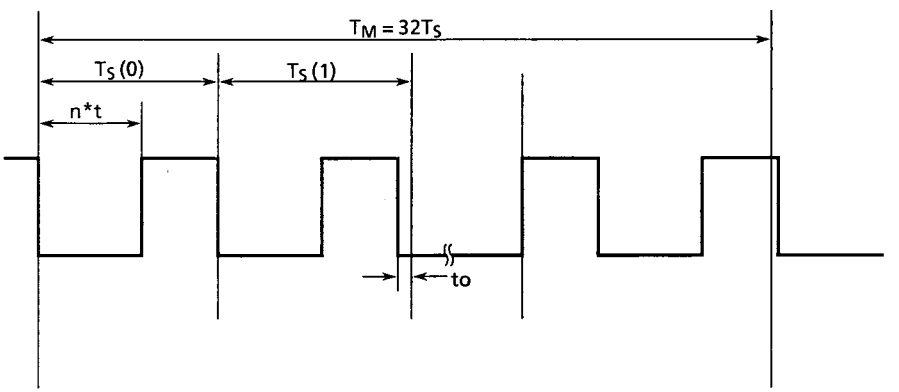


Figure 3.9 (1)

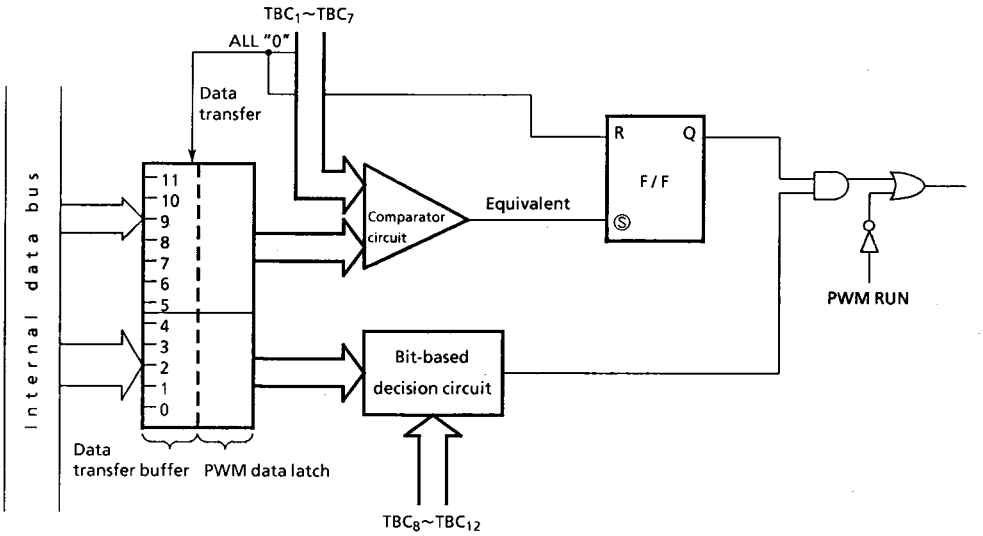


Figure 3.9 (2) Block Diagram of PW0 and PW1

3.9.2 PWM8

In 8-bit resolution PWM output, one cycle is TN = 28/(fc/2)[seconds].

When data is n (n = 0-256), the low level pulse width TN as the cycle is n*to (to = 22/(fc/2)). However, when n = 0, it is 1/(fc/2).

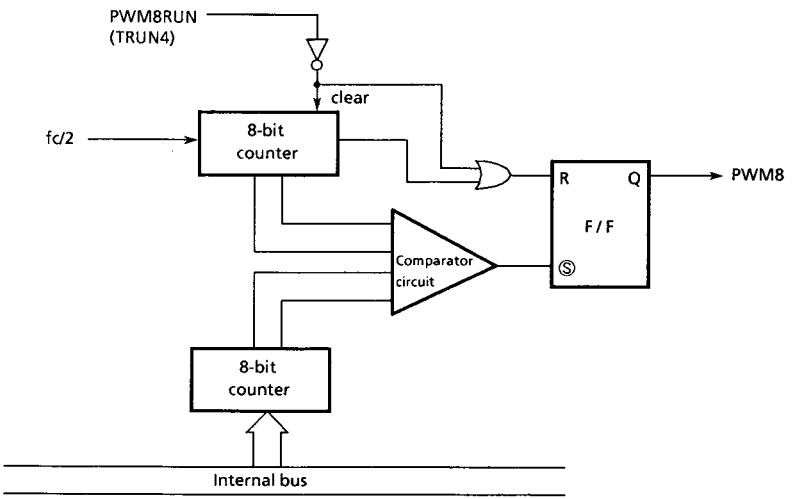
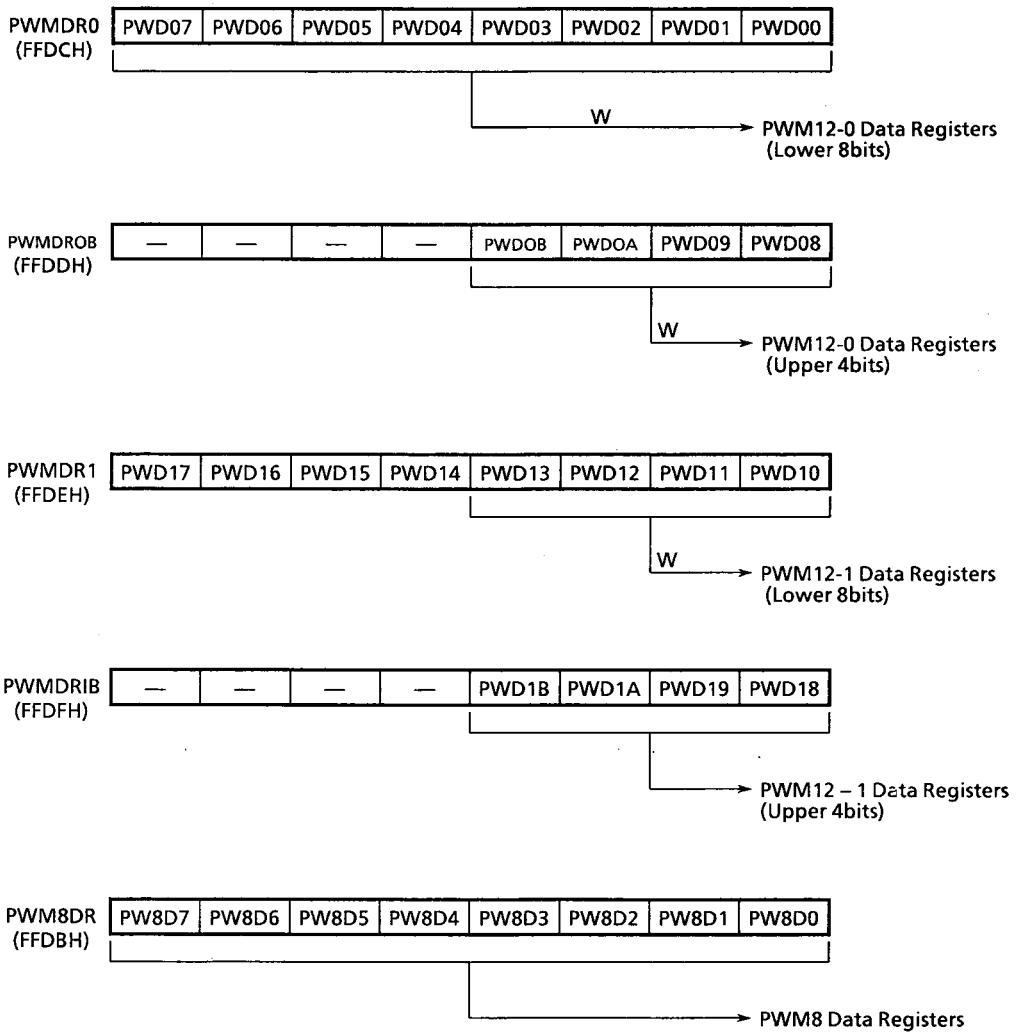


Figure 3.9 (3) Block Diagram of PWM8

3.9.3 Control Registers



3.10 Time Base Counter/Watchdog Timer

3.10.1 Time Base Counter

The TMP91C642 has an 18-bit time base counter. The time base counter inputs a clock pulse that is the frequency of the source clock pulse (fc) divided by two.

Outputs from the time base counter are the reference signals for the timer's input clock signals, 24-bit capture, and timing pulse generator.

Interrupts can be generated from signals output from time base counter outputs TBC11-TBC18.

When an INT TB interrupt is generated, the interrupt destination is determined by reading the flag WDMOD(FFE6H) bits 3, 5, and 6.

Table 3.10 Time Base Counter and Cycle

TBC	1	2	3	4	5	6	7	8	9
(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
@fc = 10MHz (μsec)	0.4	0.8	1.6	3.2	6.4	12.8	25.6	51.2	102.4

10	11	12	13	14	15	16	17	18
2^{11}/fc	2^{12}/fc	2^{13}/fc	2^{14}/fc	2^{15}/fc	2^{16}/fc	2^{17}/fc	2^{18}/fc	2^{19}/fc
204.8	409.6	819.2	1638.4	3276.8	6553.6	13107.2	26214.4	52428.8

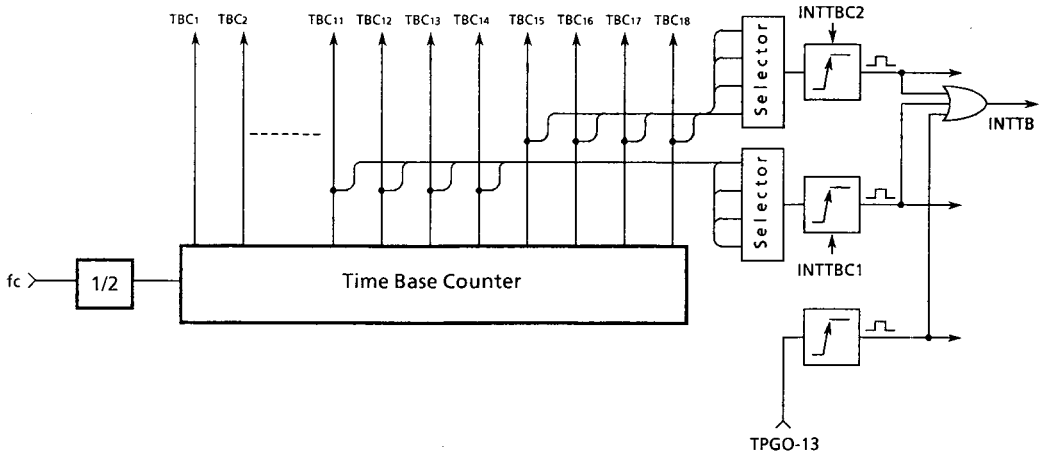


Figure 3.10 (1) Block Program of Time Base Counter (TBC)

3.10.2 Watchdog Timer (Endless Loop/Idle Detection Timer)

If noise or other factors cause the CPU to operate in error (enter an endless loop) the watchdog timer detects that fact so that the CPU can be returned to normal operating state. When an endless loop is detected, a non-maskable interrupt is generated to tell the CPU.

◇ Configuration

The watchdog timer is conFigured from a 5-step binary counter that uses TBC18 as the input clock, a flip-flop for controlling the enable/disable of watchdog timer output, and control registers.

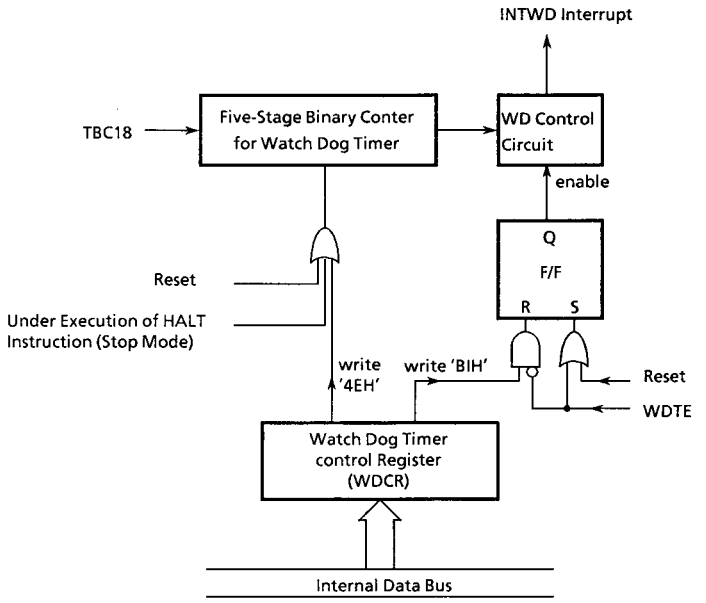


Figure 3.10 (2) Block Diagram of Watchidog Timer

◈ Control registers

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

1) Watchdog timer enable/disable control register WDTE The reset initializes

WDTE to 1, thus enabling the watchdog timer.

To disable the watch dog timer, these bits must be cleared to "0" and the disable code B1H must be written into the WDCR register.

To return from disable to enable, just set the WDTE bits to "1".

2) Watchdog timer control register WDCR

Watchdog timer disable and binary counter clear are controlled by this register.

- Disable control

The watchdog timer is disabled by writing disable code B1H into the WDCR register after WDMOD4 (WDTE) is cleared to "0".

```
WDMOD ← x x x 0 x x x x    Clears WDTE to "0".
WDCR  ← 1 0 1 1 0 0 0 1    Writes disable code B1H
```

- Writing the clear code

4EH into the WDCR register clears the binary counter to "0" and restarts the count.

```
WDCR  ← 0 1 0 0 1 1 1 0    Writes the clear code 4EH.
```

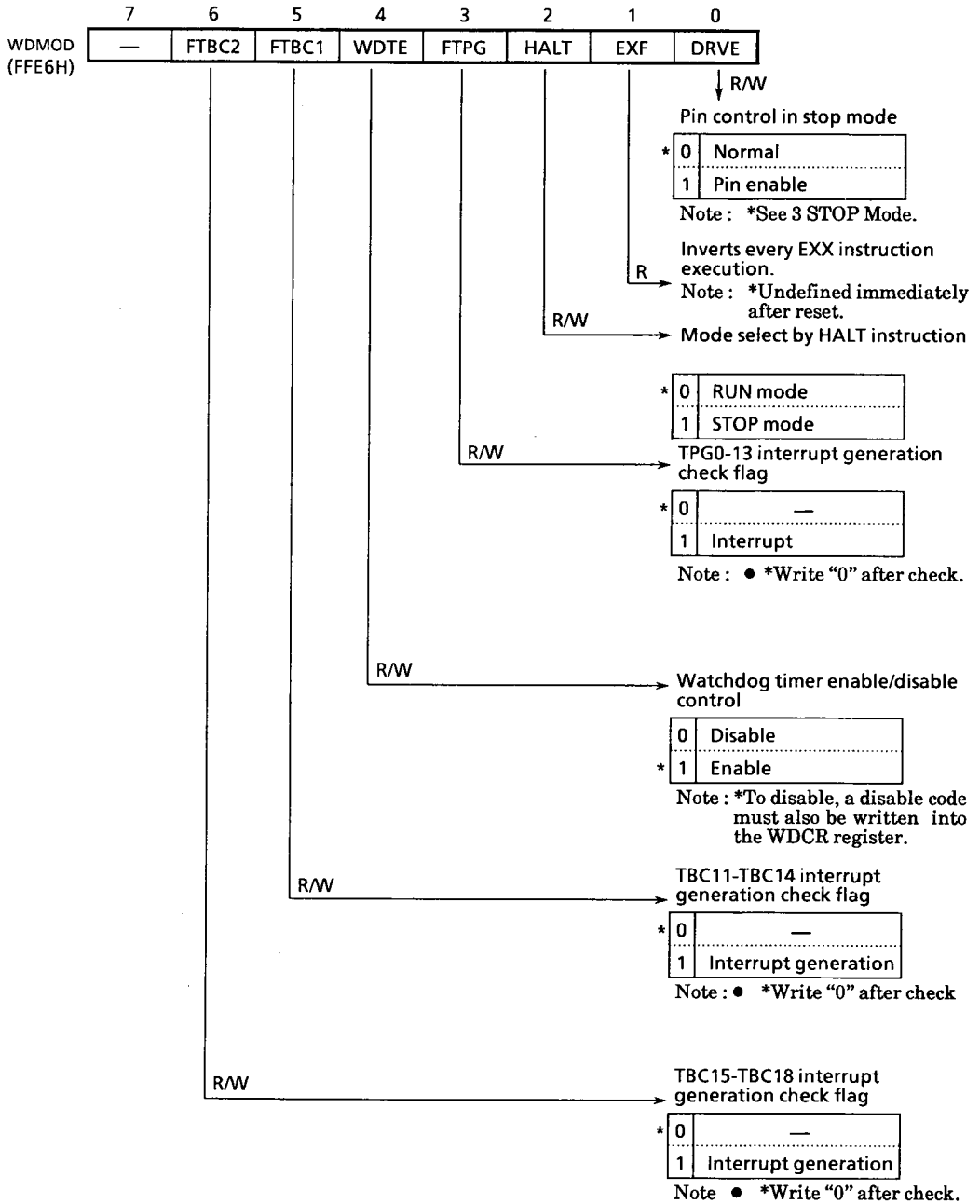

WDMOD (FFE6H)
7 6 5 4 3 2 1 0
— FTBC2 FTBC1 WDTE FTPG HALT EXF DRVE
R/W
Pin control in stop mode
* 0 Normal
1 Pin enable
Note : *See 3 STOP Mode.
R
Inverts every EXX instruction execution.
Note : *Undefined immediately after reset.
R/W
Mode select by HALT instruction
* 0 RUN mode
1 STOP mode
R/W
TPG0-13 interrupt generation check flag
* 0 —
1 Interrupt
Note : • *Write "0" after check.
R/W
Watchdog timer enable/disable control
0 Disable
* 1 Enable
Note : *To disable, a disable code must also be written into the WDCR register.
R/W
TBC11-TBC14 interrupt generation check flag
* 0 —
1 Interrupt generation
Note : • *Write "0" after check
R/W
TBC15-TBC18 interrupt generation check flag
* 0 —
1 Interrupt generation
Note • *Write "0" after check.

Figure 3.10 (3)

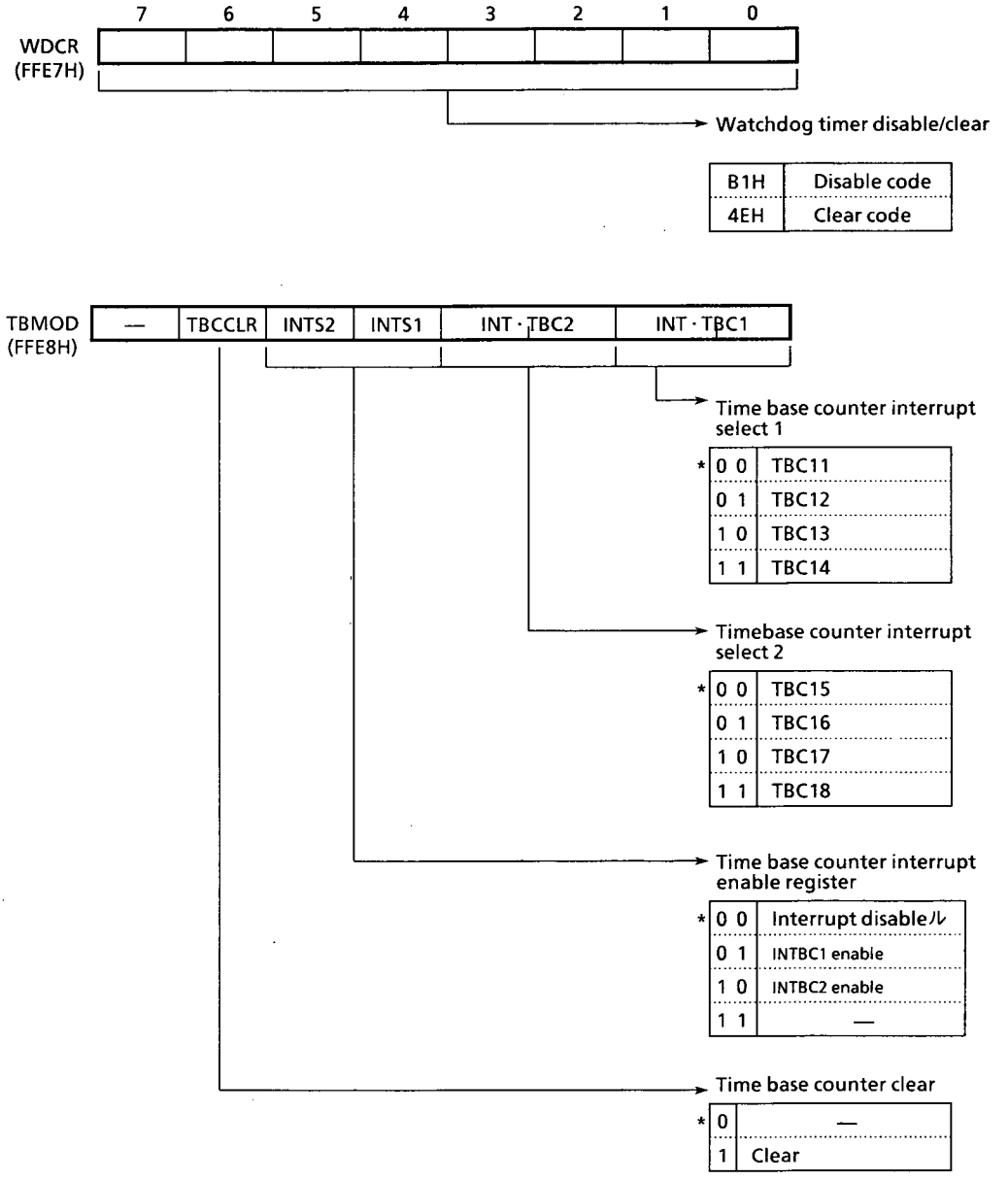

7 6 5 4 3 2 1 0
WDCR
(FFE7H)
Watchdog timer disable/clear
B1H Disable code
4EH Clear code
TBMOD
(FFE8H)
— TBCCLR INTS2 INTS1 INT · TBC2 INT · TBC1
Time base counter interrupt select 1
* 0 0 TBC11
0 1 TBC12
1 0 TBC13
1 1 TBC14
Timebase counter interrupt select 2
* 0 0 TBC15
0 1 TBC16
1 0 TBC17
1 1 TBC18
Time base counter interrupt enable register
* 0 0 Interrupt disable
0 1 INTBC1 enable
1 0 INTBC2 enable
1 1 —
Time base counter clear
* 0 —
1 Clear

Figure 3.10 (4)

3.11 Capture Circuit

The TMP91C642 has a RAM with 24-bit, 8-step FIFO to simplify various time measurements. The time (18-bit data from TBC1-TBC18) 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 by the input signals.

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

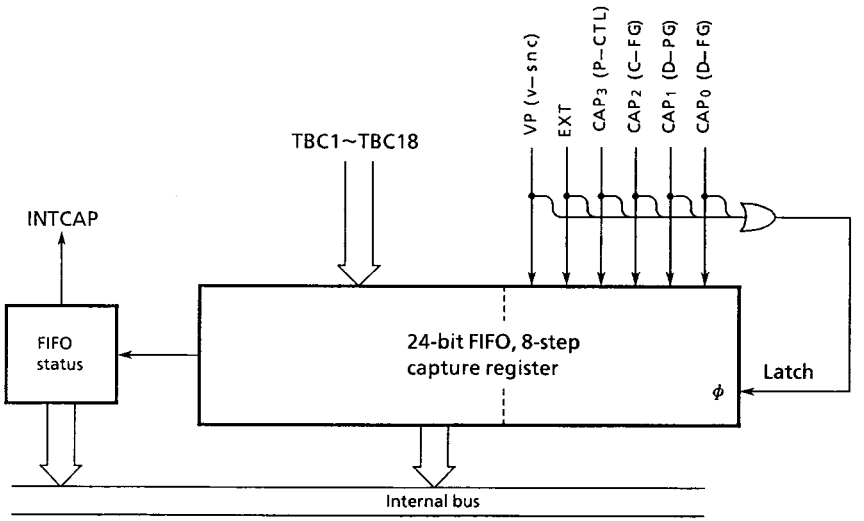


Figure 3.11(1) Block Diagram of Capture Circuit

3.11.1 Explanation Of Operations

1) Capture

There are 6 input sources for the capture circuit: CAP0(D-FG), CAP1(D-PG), CAP2(C-FG), CAP3(P-CTL) and EXT external input signals, and the V-Sync (VP) signals separated from the C-Sync signals. A total of 24 bits, the 6 input signal bits and the 18-bit values from the time base counter (TBC1-TBC18) to the 3 bytes at addresses FFFAH-FFFCH are latched using the input signal edge (see 3.13 Servo input). The latch can also be operated by setting bit 1 in the servo flag control register (SVCFREG: memory address FFFE) to "1". (The same is true of EXT input.)

The 24-bit data is obtained by reading the data* at the addresses in the sequence FFFAH, FFFBH and FFFCH.

Note : *This address must always be read last because reading the data at address FFFCH 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 8-step FIFO contents are full, the capture operation is disabled by the input signal. The FIFO status register (memory address FFFDH) will be FFH when this happens.

When the FIFO status register 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 servo flag control register (memory address FFFEH bit 0) resets the following circuits:

① FIFO address counter

② FIFO status

4) Capture interrupt

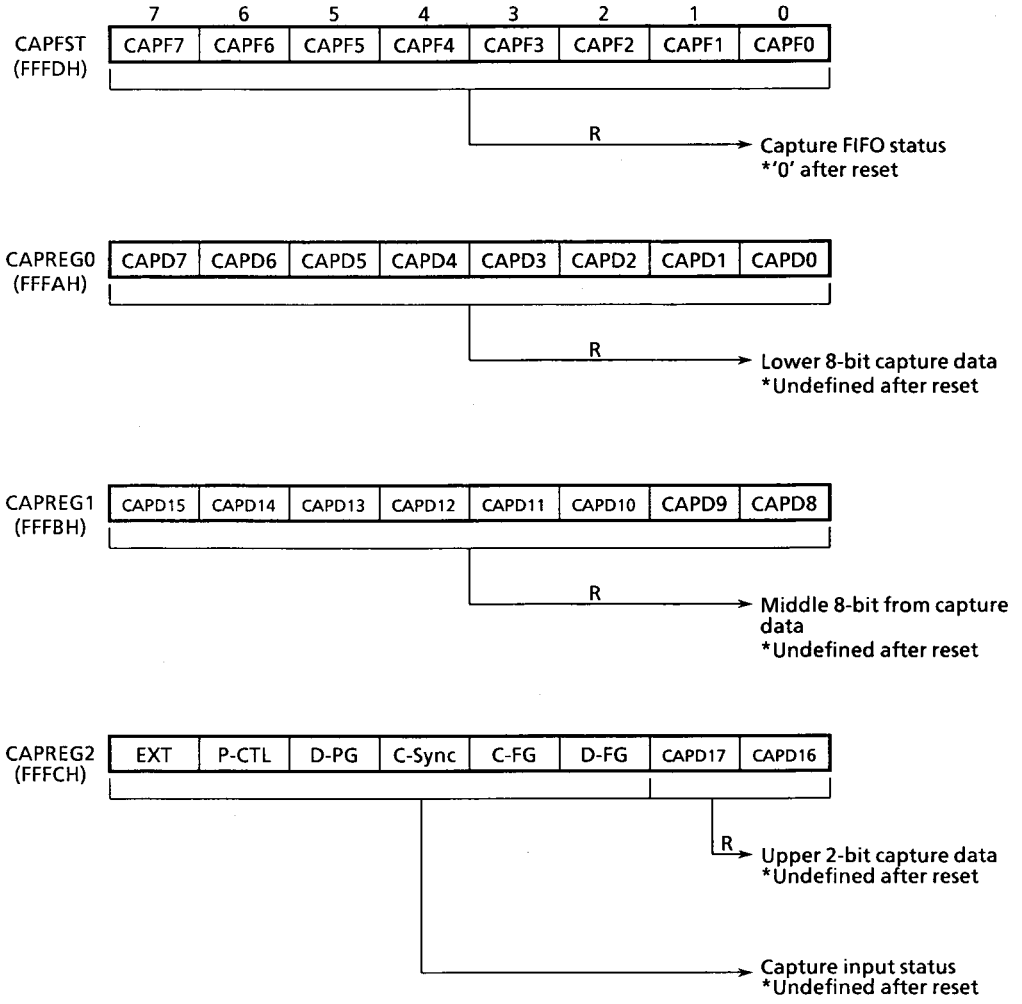
The INTCAP interrupt is generated by latch operation by the input edge. The INTCAP interrupt signal continues to be generated until FIFO status becomes "0" (data is empty).

5) Data processing

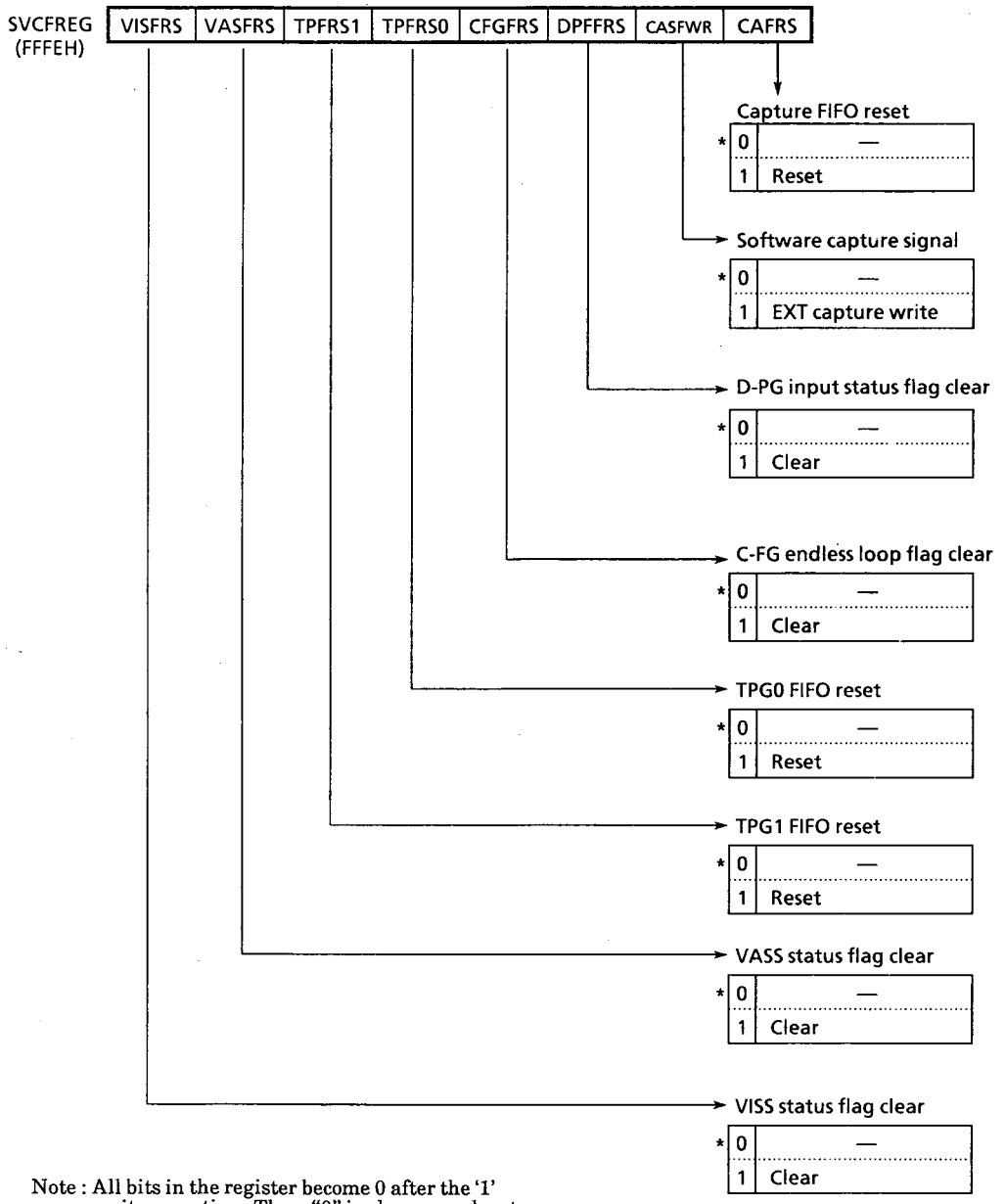
The bit of the data corresponding to the input signal is set to 1 by the upper 6 bits of the 24 bits at addresses FFFAH-FFFCH, which are latched by the input signal. Referring to this data allows the input signal to be identified.

The CPU stores the latched data in its internal RAM. When it detects the next input signal edge, it subtracts the data in RAM from the data now latched, allowing the detection of capstan motor and drum motor r.p.m. Detection precision is 200 nanoseconds (@fc = 10MHz) for extremely small quantized errors.

3.11.2 Control Registers



Note : When reading 3 capture data bytes, always read CAPREG2 last.



Note : All bits in the register become 0 after the '1' write operation. Thus, "0" is always read out.

3.12 Timing Pulse Generator (TPG)

The TMP91C642 contains two channels for a 32-bit 8-step FIFO with timing pulse generator circuit to control all signal and mechanism types used by VCRs.

The timing pulse generator (TPG) generates16-bit timing pulses, at 400nsec precision (@fc = 10MHz), that are synchronized with the time-base counter TBC).

Figure 3.12 is a block diagram of the timing pulse generator circuit.

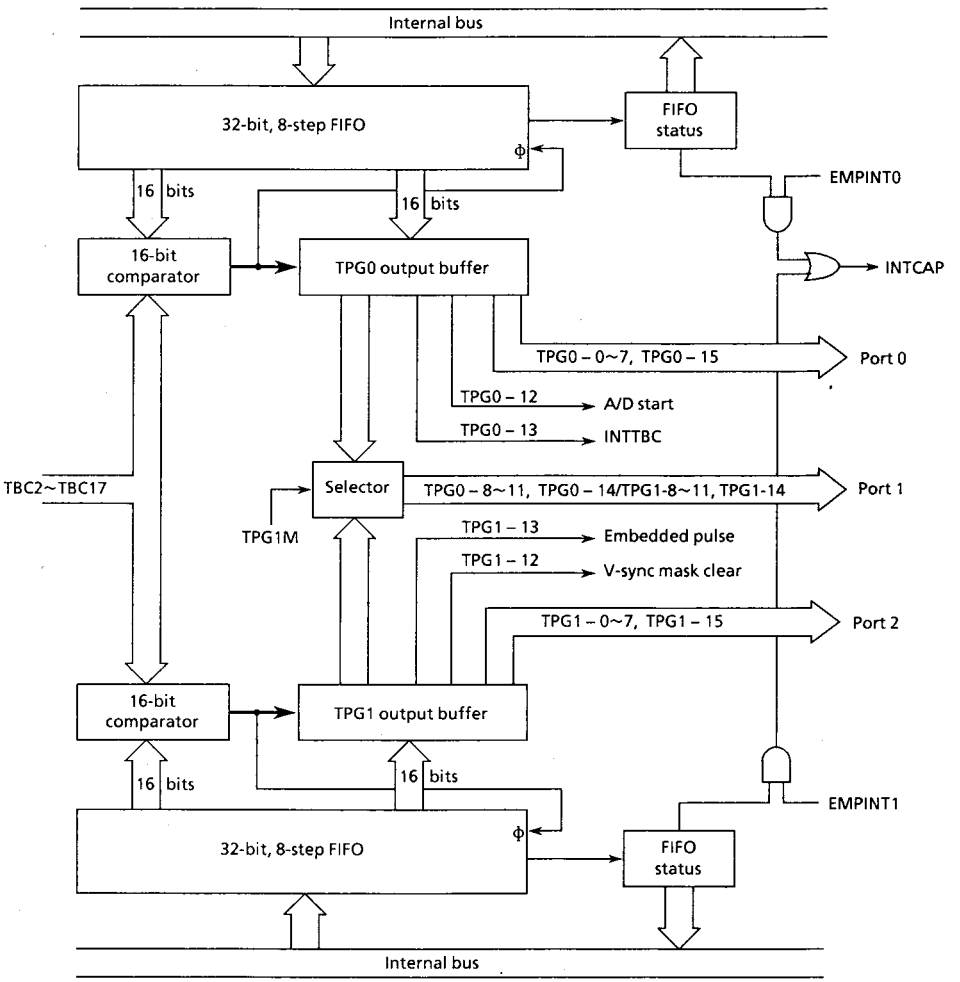


Figure 3.12 Block Diagram of Timing Pulse Generation Circuit

Table 3.12 TPG Connection

TPG No.	Output to	TPG No.	Output to
TPG0 – 0	Port 00	TPG1 – 0	Port 20
1	01	1	21
2	02	2	22
3	03	3	23
4	04	4	24
5	05	5	25
6	06	6	26
7	07	7	27
8	10	8	10
9	11	9	11
10	12	10	12
11	13	11	13
12	A/D start	12	V – sync mask clear
13	INTTBC	13	Embed V – sync (VP)
14	Port 10 (3-value control)	14	Port 10 (3-value control)
15	00 (3-value control)	15	Port 20 (3-value control)

3.12.1 Explanation of operations

1) Data

The TPG has a 32-bit 8-step FIFO, 16 bits for the output data register (memory addresses FFF4H and FFF5H) and 16 bits for the comparator data register (memory addresses FFF6H and FFF7H). When the values written in the comparator data register are the same as the values in TBC2-TBC17, the values written in the output data register are output.

Note : *When TP (memory address FFFFH bit 3) is "0", the data in the TPG0 channel is written; when TP is "1", the data in the TPG1 channel is written.

Writing the data at FFF7H, one of the four bytes of data from addresses FFF4H to FFF7H, increments FIFO addresses. Therefore when writing ordinary data, write the data in the memory address order FFF4H, FFF5H, FFF6H and FFF7H.

2) FIFO

Each TPG has an 8-step FIFO (first-in, first-out) circuit. When there is no data, the FIFO generates an interrupt signal.

This interrupt signal is not generated immediately after resets or TP resets. An interrupt is generated after data is written into the data register once, the FIFO is incremented, the out-data register is output, then the FIFO becomes empty.

The interrupt for TPG0 is controlled by TPCRE0 (memory address FFF8H bit 5) and for TPG1 by TPOCRE1 (memory address FFF9H bit 5).

Note : See 3 Interrupt Control

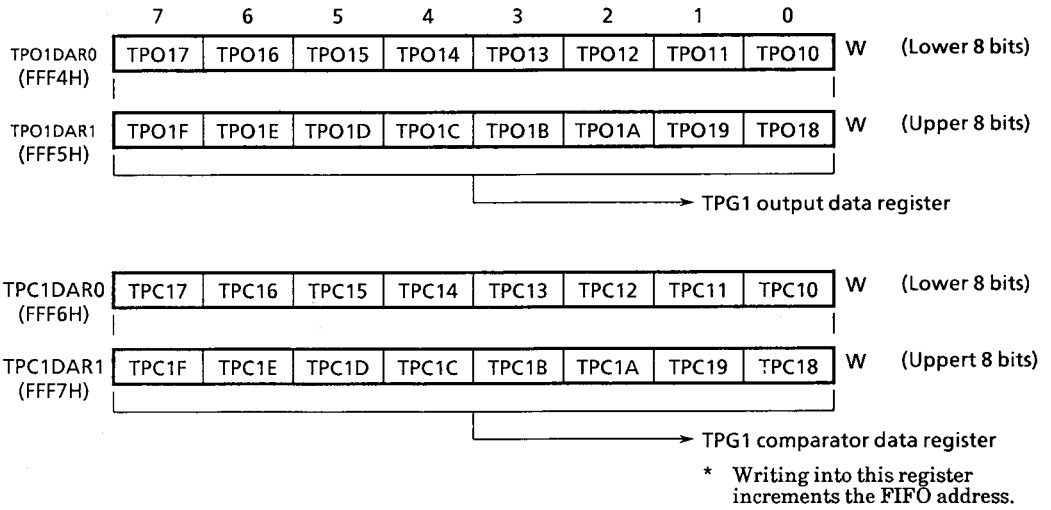
3) TP reset

There are registers TPFRS0 and TPFRS1 (memory address FFFEH bits 4 and 5) to be used for resetting the TPG. Writing "1" into these registers resets TPG0 and TPG1 following circuits.

① FIFO address counter
② FIFO status
③ Empty interrupt flag

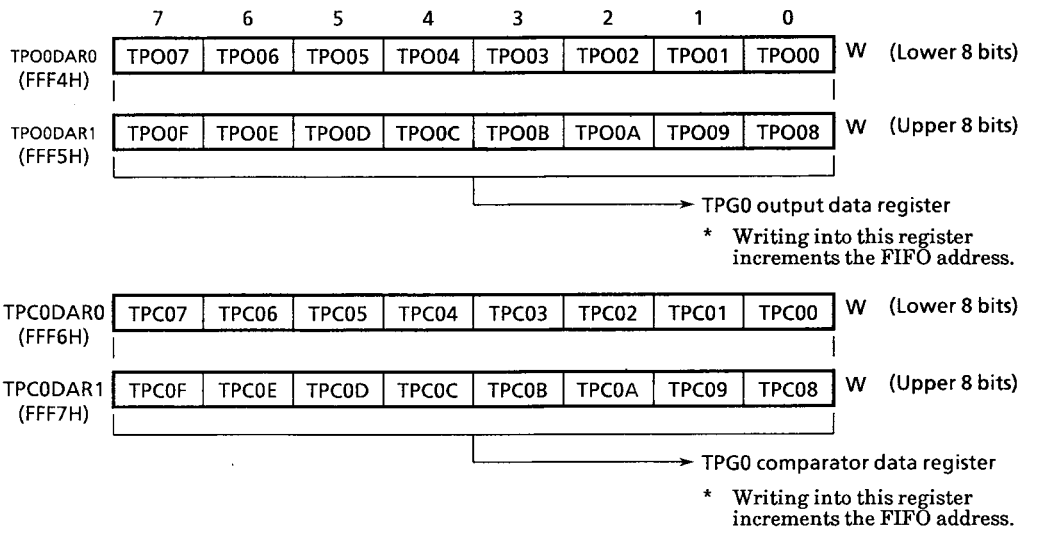
Note : Since the TPG output buffer is not reset, it holds the values stored prior to the TP reset. However, in a system reset, all bits are cleared to "0".

- When TP (memory address FFFH bit 3) is set to '0'

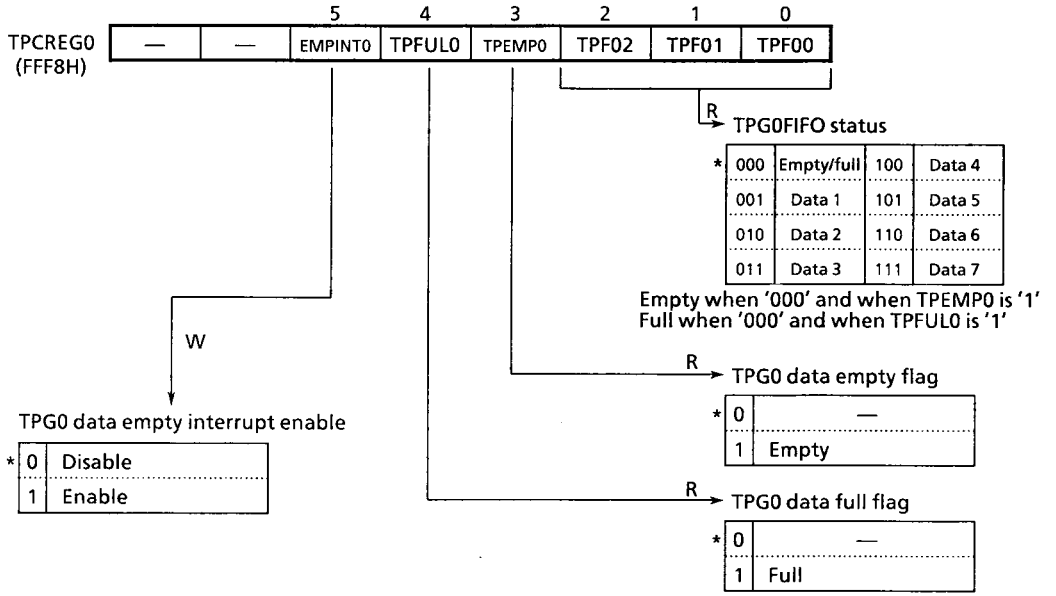


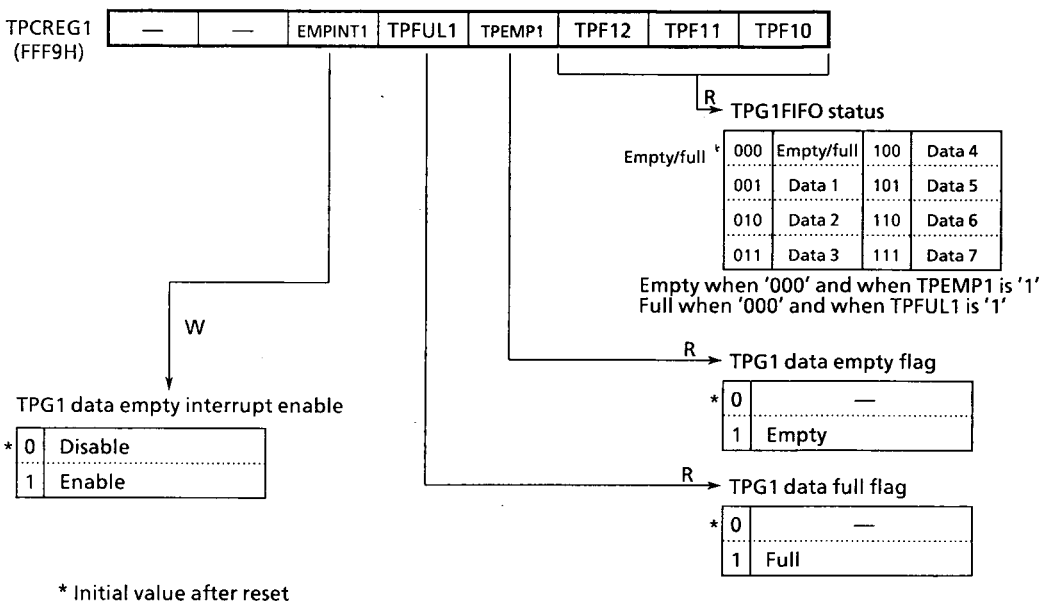
Note : Undefined after reset

- When TP (memory address FFFH bit 3) is set to '1'



Note : Undefined after reset





* Initial value after reset

3.13 Servo Input

The TMP91C642 has the following servo-input dedicated and signal processing circuits for efficient arithmetic/logic operations in software.

- C-FG programmable frequency-divider circuit
- C-FG mask counter
- P-CTL programmable frequency-divider circuit
- D-FG/DPG input processing circuit
- VISS/VASS detection circuits
- C-sync input processing circuit

3.13.1 C-FG Input Circuit

The signals used for C-FG input signals are frequency generator (FG) micro-signals from the capstan motor that are amplified by an external circuit sensor amplifier, then waveform-shaped and set to 0V-5V logic level.

The C-FG input circuit has a 1/2-frequency-divider/doubler circuit and a 5-bit programmable frequency-divider to handle features such as the capstan motor's variable speed playback (special playback). A 4-bit mask counter prevents the motor from running too fast.

The dedicated input enable register SINCR2 (memory address FFCEH bit 2) can enable or disable input signals and be used as an input port at the same time.

① C-FG signal sampling rate selector

This circuit selects input signals of 1/2, 1x and 2x the sampling rate. The selector is controlled by the dedicated input control register SSINCR (memory address FFEEH bits 3 and 4).

② C-FG programmable frequency-divider

This frequency-divider is made up from a 5-bit counter. Frequency divisions from 1 to 1/32 are assigned by the C-FG control register PCFSCR (memory address FFF2H bits 0 to 4).

③ C-FG mask counter

The mask counter masks C-FG signals that are generated within the time assigned by the program and prevents the generation of interrupts shorter than the mask time. Mask time up to 768 microseconds (@fc=10MHz) is assigned by the C-FG mask control register PCFCR1 (upper 4 bits at memory address FFF0H).

Table 3.13 (1) lists the mask control registers and mask times.

3.13.2 D-FG/D-PG Input Circuits

The TMP91C642 has a D-FG input signal 3-bit programmable frequency-divider circuit that can be synchronized with the detect edge select circuit and D-PG input so that the microprocessor can handle different kinds of drum frequency generation (FG) and pulse generation (PG) methods.

The signals used are micro-signals from the frequency generator (FG) and pulse generator (PG), both of which are attached to the drum motor. They are waveform-shaped and the 0V-5V logic level signals obtained are used.

The D-FG/D-PG detection edge is controlled by the input control register SSINCR (memory address FFEEH bits1 and 5) and the 1- to 1/8-frequency division rate is assigned by the D-FG frequency division control register PCDFCR (memory address FFF1H bits 3 to 5) for the 3-bit programmable frequency-divider circuit.

3.13.3 P-CTL Input Circuit

The signals used for P-CTL signals are 0-5V logic level signals obtained by an external circuit that amplifies and waveform-shapes the signals fetched by the control head as control pulses recorded on VCR tapes during playback.

The P-CTL input circuit is configured from a selector circuit for edge detection and a 5-bit programmable frequency-divider.

Edge detection is selected by the control register CSYNCR (memory address FFEFFH bit 5).

The 1- to 1/32-frequency division rate is assignd by the control register PCDFCR (PCTPR4-0: memory address FFF1H bits 3 to 7) for the 5-bit frequency-divider circuit.

Input control signals are also supplied to timer 3 and the P-CTL detection circuit.

Table 3.13 (1) C-FG mask

C F M S 3 2 1 0	n	Mask time (μs) @fc = 10MHz
0 0 0 0	0	∞
0 0 0 1	1	51.2
0 0 1 0	2	102.4
0 0 1 1	3	153.6
0 1 0 0	4	204.8
0 1 0 1	5	256.0
0 1 1 0	6	307.2
0 1 1 1	7	358.4
1 0 0 0	8	409.6
1 0 0 1	9	460.8
1 0 1 0	10	512.0
1 0 1 1	11	563.2
1 1 0 0	12	614.4
1 1 0 1	13	665.6
1 1 1 0	14	716.8
1 1 1 1	15	768.0

Mask Time
$= TBC_8 \times n$
$= (1/(fc/2)) \times 2^8 \times n$ (Sec)

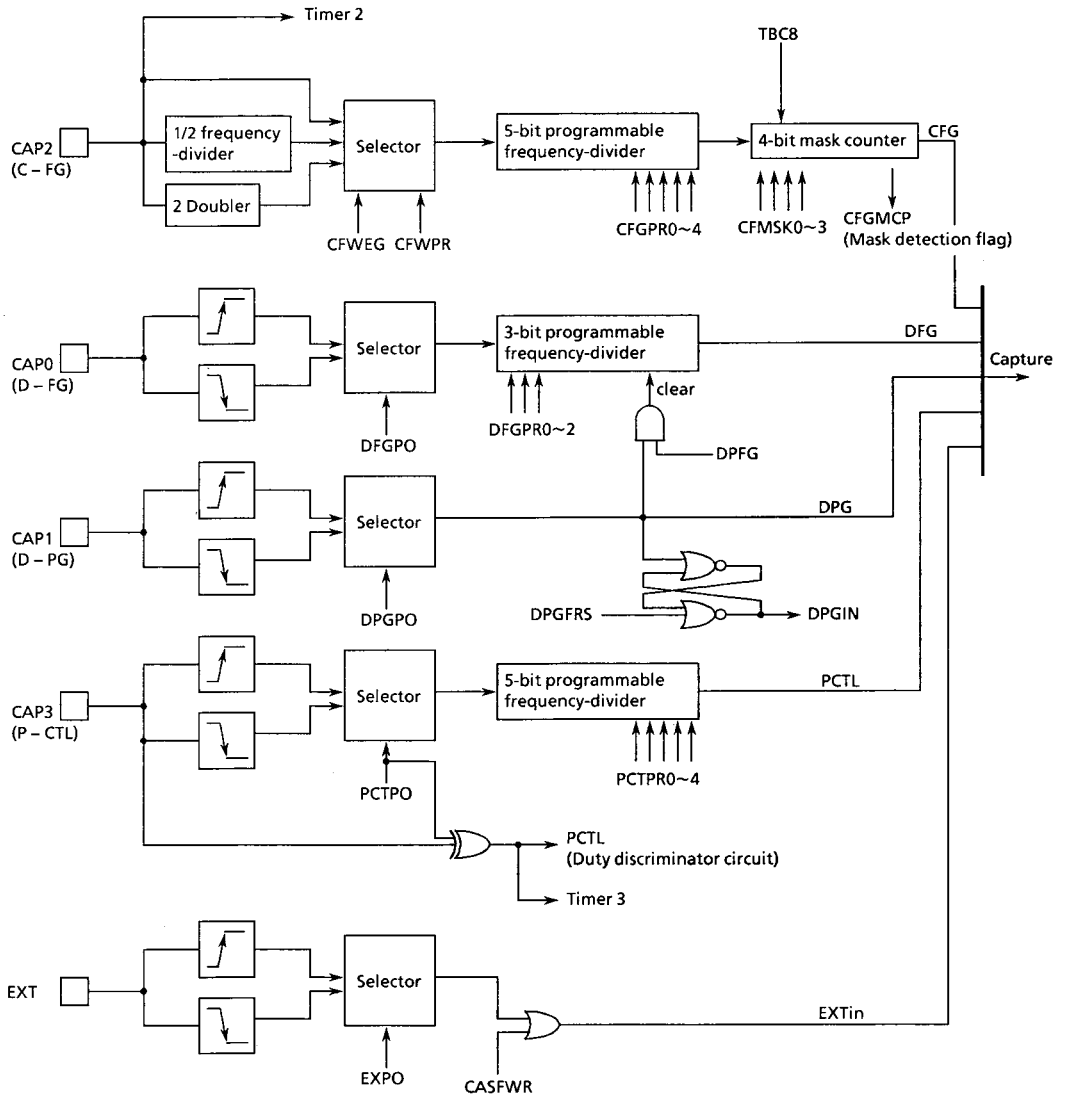


Figure 3.13 (1) Block Diagram of Servo Input

3.13.4 VISS/VASS Detection Circuit

The VISS/VASS detection circuit handles VHS control coding format. 3.13 (2) shows, it is configured from a duty discriminator circuit, code detection circuit and 16-bit data register.

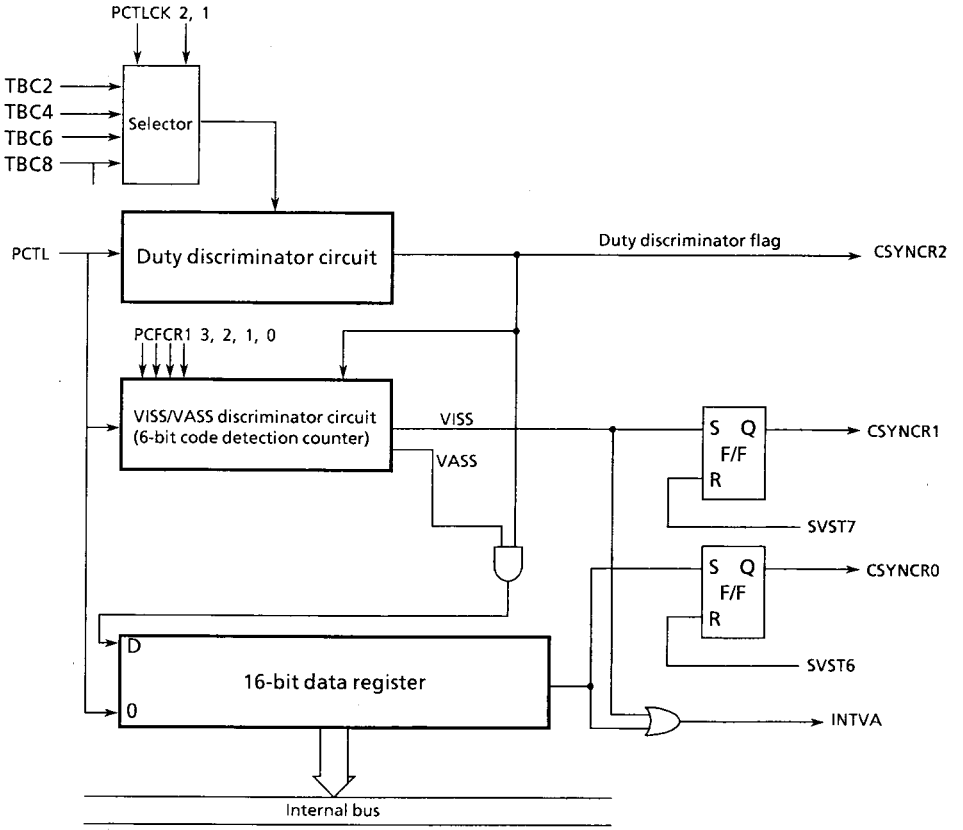


Figure 3.13 (2) Block Diagram of VISS/VASS

1. Duty Discriminator Circuit

The duty discriminator circuit measures the cycle of control signals CTL and detects the duty width. If the results of discriminating duty width are 50% or more, a '0' is set in the duty discriminator flag register CTLDIT (memory address FFEFH bit 2), if the results are 50% or less, a '1' is set in the register.

2. VISS/VASS discrimination

The following regulations have been prescribed for VISS (VHS index search system: index search code) and VASS (VHS address search system: address code).

- Index search code

Write "0" at both ends and 1 at 61 ± 3bits, for a total of 63 ± 3 bits.

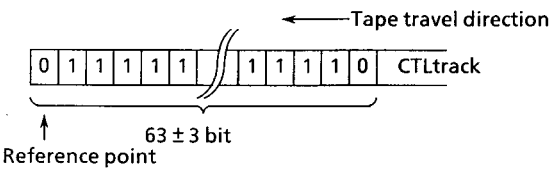


Figure 3.13 (3)-1

- Address cod

Write, three times, an 11-bit header at both ends and between them 4-bit BCD 4-digit data as shown in the diagram below.

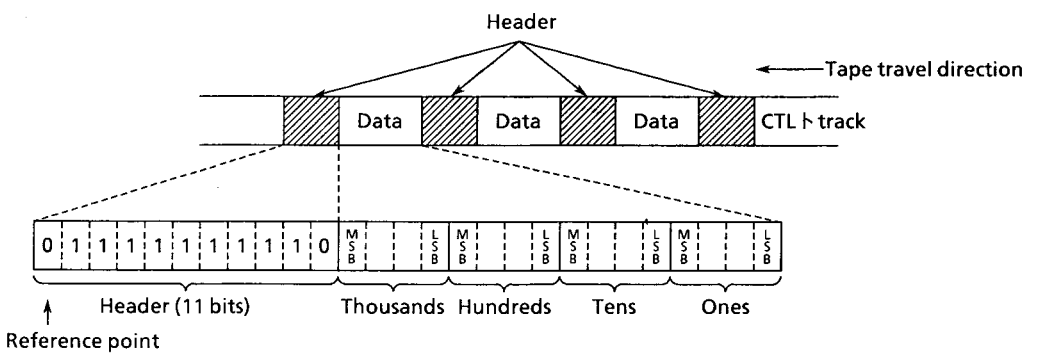


Figure 3.13 (3) -2

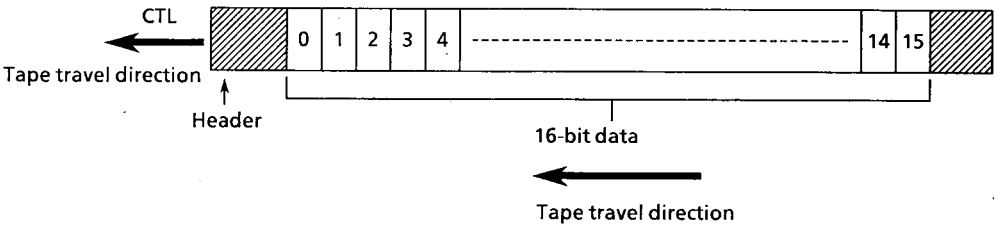
① Detection of index search code

Duty detection results consisting of a consecutive 61+ 3 "1" bits must be detected for the index search code. The TMP91C642 software assigns this detection count for the numeric count of 4 to 60 using the 4 bits of the VISS discrimination control register PCFCR1 (memory address FFFOH bits 0 to 3).

Therefore, when the index search code is detected, the VISS discrimination flag VISSF (memory address FFEFH bit 1) is set and the INTVA interrupt is generated.

② Address code

The address code recognizes that the control code starts with "0" and is followed by 9 consecutive "1"s and a "0" code, then stores the 16-bit data after the next code in the shift register. When all16 address code bits have been detected, the VASS discrimination flag VASSF (memory address FFEFH bit 0) is set and the INTVA interrupt is generated. The sequence in which data is stored in the shift register is shown below.

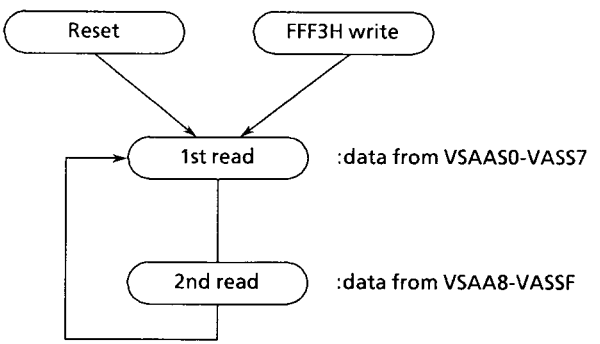


16-bit VZSS data register (memory address FFF3H)

Register	VASS7	VASS6	VASS5	VASS4	VASS3	VASS2	VASS1	VASS0
Latch data	7	6	5	4	3	2	1	0

VASSF	VASSE	VASSD	VASSC	VASSB	VASSA	VASS9	VASS8
15	14	13	12	11	10	9	8

Note : When the16-bit VASS data register is read, memory address FFF3H is read twice. Values from VASS0-VASS7 are read out the first time; values from VASS8-VASSF are read out the second time.
When the data has been read the second time then the data which was read first is read again. Note that the 16-bit transfer instruction cannot be used. If any data is used to write address FFF3H, it is specified in VASS0-VASS7.



3.13.5 C-Sync Processing Circuit

The C-sync processing circuit is configured from the V (vertical) sync separator circuit, the even number field discriminator circuit, and the 50/60 field discriminator circuit. Figure 3.13 (4) is a block diagram of this circuit.

The signals used for C-sync input are 0-5V logical level signals obtained by an external circuit that amplifies and waveform-shapes the composite sync signals in video signals from a TV or VCR.

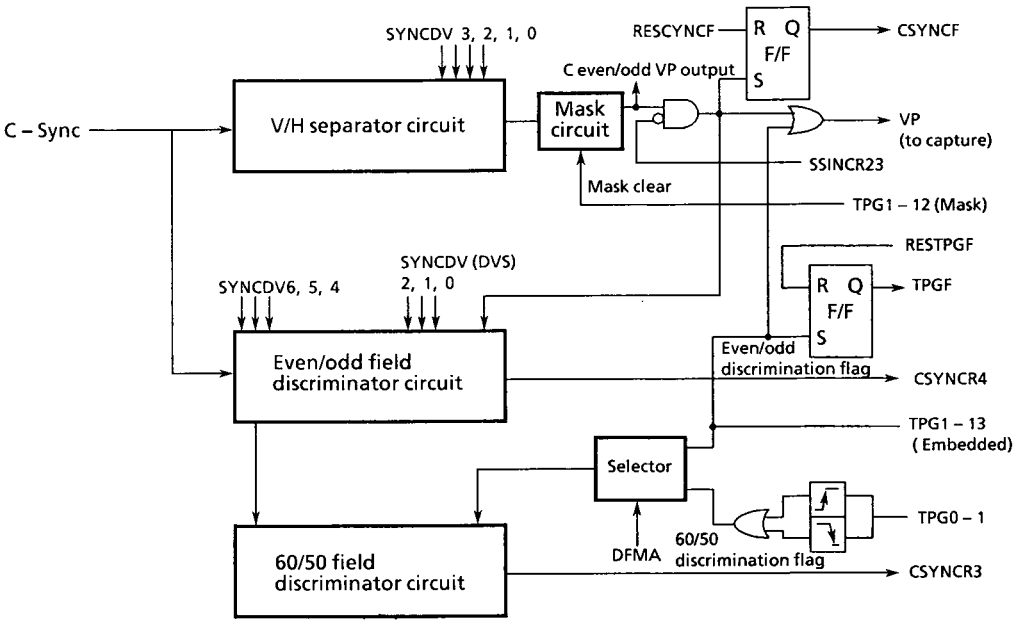


Figure 3.13 (4) Block Diagram of C-sync Processing

1. V-sync separator circuit

The V-sync separator circuit is the circuit that separates V-sync signals from C-sync input signals. The separated V-sync signals (VP) are used as reference signals for the servo during recording.

Figure 3.13 (5) is a block diagram of the circuit.

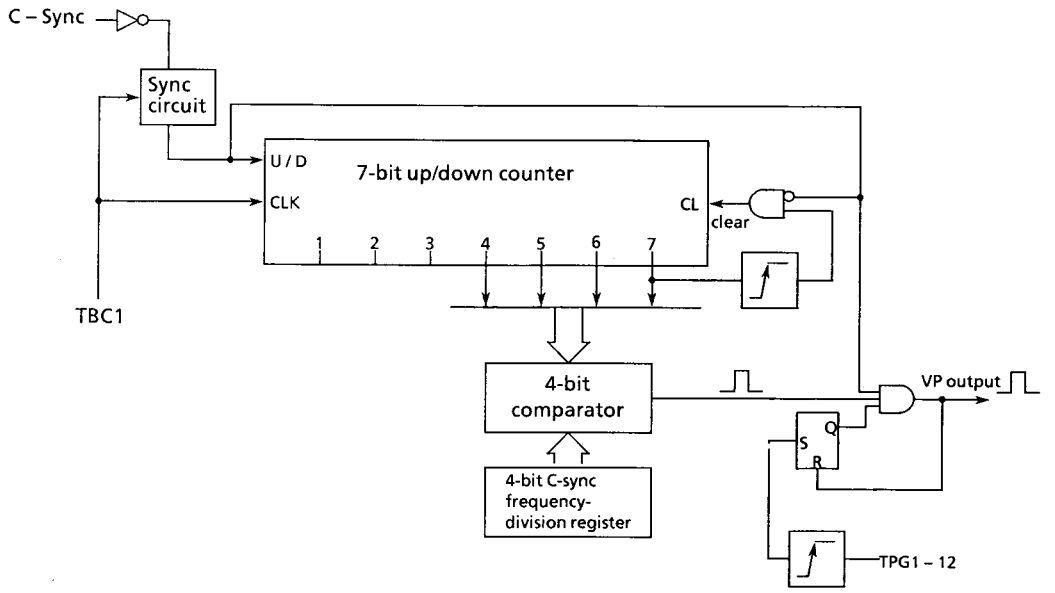


Figure 3.13 (5) Vertical/Horizontal (V/H) Separator Circuit

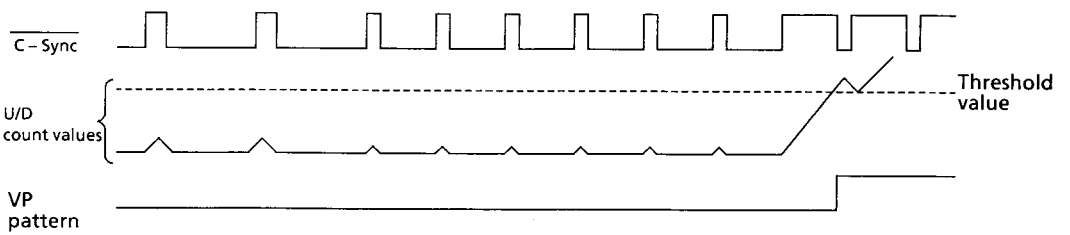


Figure 3.13 (6) V-sync Separation

2. Even/odd field discriminator circuit

The even/odd field discriminator decides the odd or even number field for the C-sync signal. The results of this decision are used to match, in frame units, the phases of playback and recording during record.

This circuit is configured from a 3-bit counter that masks C-sync signals for a fixed period, a mask register (memory address FFCFH bits 4-6@DVS:0) that determines the mask period, a 3-bit counter that measures that fixed time from V-sync signals, and a measurement period assignment register (memory address FFCFH bits 0-2 @DVS:1).

Figure 3.13 (7) is a block diagram of the even/odd field discriminator circuit.

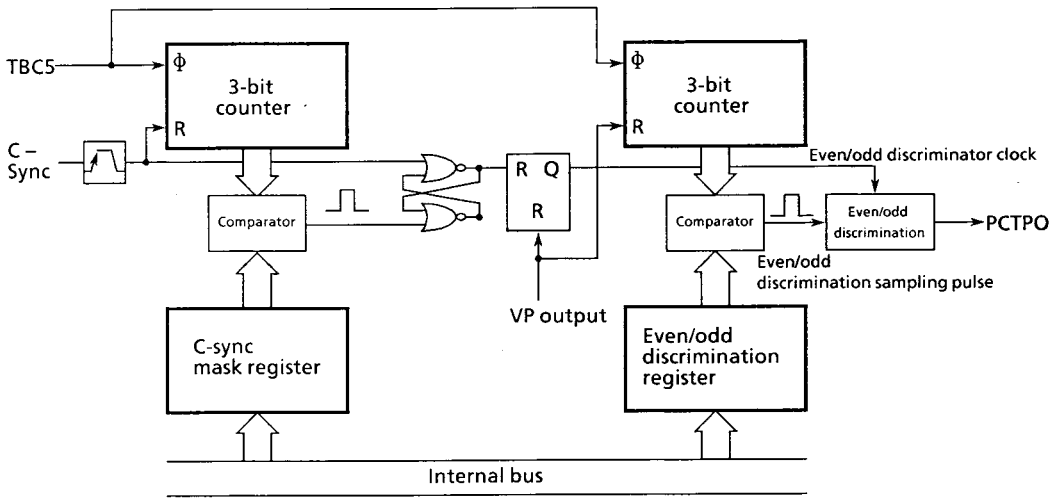


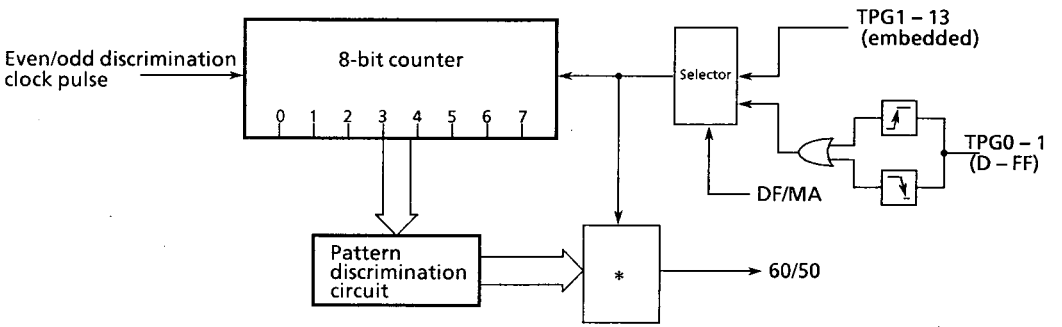
Figure 3.13 (7) Block Diagram of Even/odd Field Discriminator

3. 60/50 Field Discrimination Circuit

The 60/50 field discrimination circuit detects video signals of different format (PAL and NTSC) and switches the system's operating mode.

The 60/50 field discrimination is made by counting the number of H-sync signals in a V-sync cycle duration.

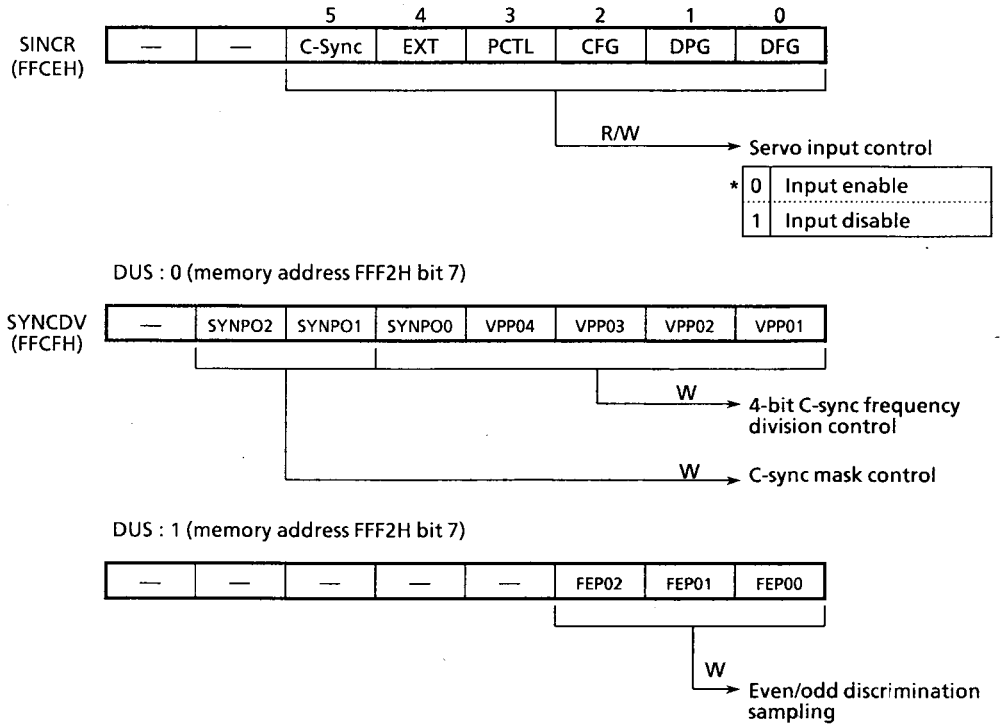
Figure 3.13 (8) diagrams the 60/50 field discrimination circuit.

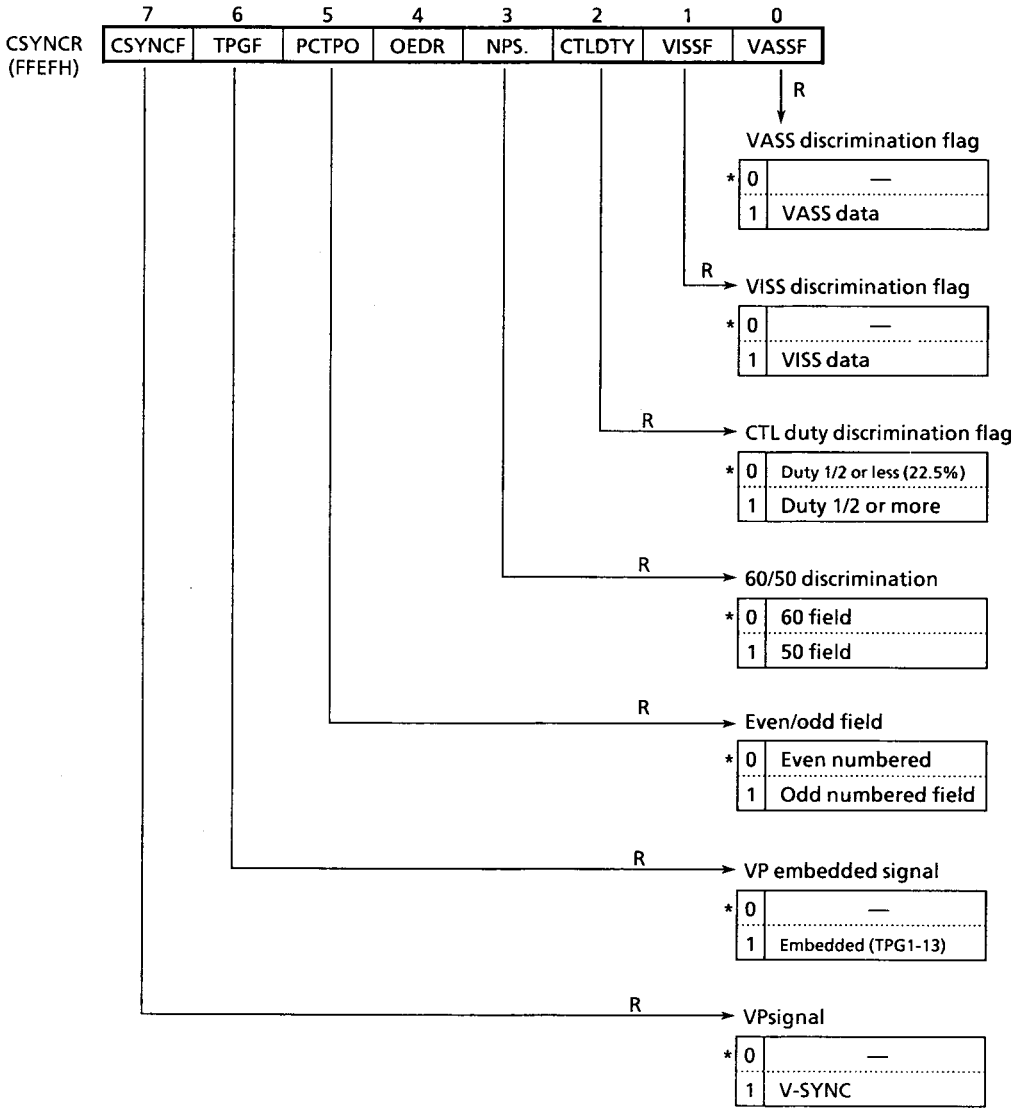


* 60/50 discrimination circuit

Figure 3.13 (8) Block Diagram of 60/50 Field Discrimination Circuit

3.13.6 Control Registers




CSYNCR
(FFEFH)
7 6 5 4 3 2 1 0
CSYNCF TPGF PCTPO OEDR NPS. CTLDTY VISSF VASSF
R
VASS discrimination flag
* 0 —
1 VASS data
R
VISS discrimination flag
* 0 —
1 VISS data
R
CTL duty discrimination flag
* 0 Duty 1/2 or less (22.5%)
1 Duty 1/2 or more
R
60/50 discrimination
* 0 60 field
1 50 field
R
Even/odd field
* 0 Even numbered
1 Odd numbered field
R
VP embedded signal
* 0 —
1 Embedded (TPG1-13)
R
VPsignal
* 0 —
1 V-SYNC

PCFCR1 (FFF0H)

7	6	5	4	3	2	1	0
CFMSK3	CFMSK2	CFMSK1	CFMSK0	VISS3	VISS2	VISS1	VISS0

W → VISS discrimination control register

W → C-FG mask control register

PCDFCR (FFF1H)

DFGPR2	DFGPR1	DFGPR0	PCTPR4	PCTPR3	PCTPR2	PCTPR1	PCTPR0

W → P-CTL frequency-division control register

W → D-FG frequency-division control register

PCDFCR (FFF2H)

DVS	PCTLCK2	PCTLCK1	CFGPR4	CFGPR3	CFGPR2	CFGPR1	CFGPR0

W → C-FG frequency-division control register

W → PCTL duty measurement clock selector

*

00	TBC2
01	TBC4
10	TBC6
11	TBC8

W → C – sync frequency-division control register select

*

0	VPPO/SYNPO
1	FEPO

Note : *See the 'SYNCDV' register (memory address FFCFH).

PCTLD (FFF3H)

	7	6	5	4	3	2	1	0
①	VASS7	VASS6	VASS5	VASS4	VASS3	VASS2	VASS1	VASS0
②	VASSF	VASSE	VASSD	VASSC	VASSB	VASSA	VASS9	VASS8

R → 16-bit VASS data register

Reads out VASS0-VASS7 for the first read.
Reads out VASS8-VASSF for the second read.
VASS0-VASS7 are specified immediately after reset.
VASS0-VASS7 are also specified when FFF3H is written into.

Note : 16-bit transfer instruction cannot be used.

SSINCR2 (FFFFH)

7	6	5	4	3	2	1	0
—	—	—	—	TP	DFMA	PB	CFGMCP

R

C-FG endless loop detection flag

*	0	—
	1	Endless loop

R / W → Vertical sync detection output disable register

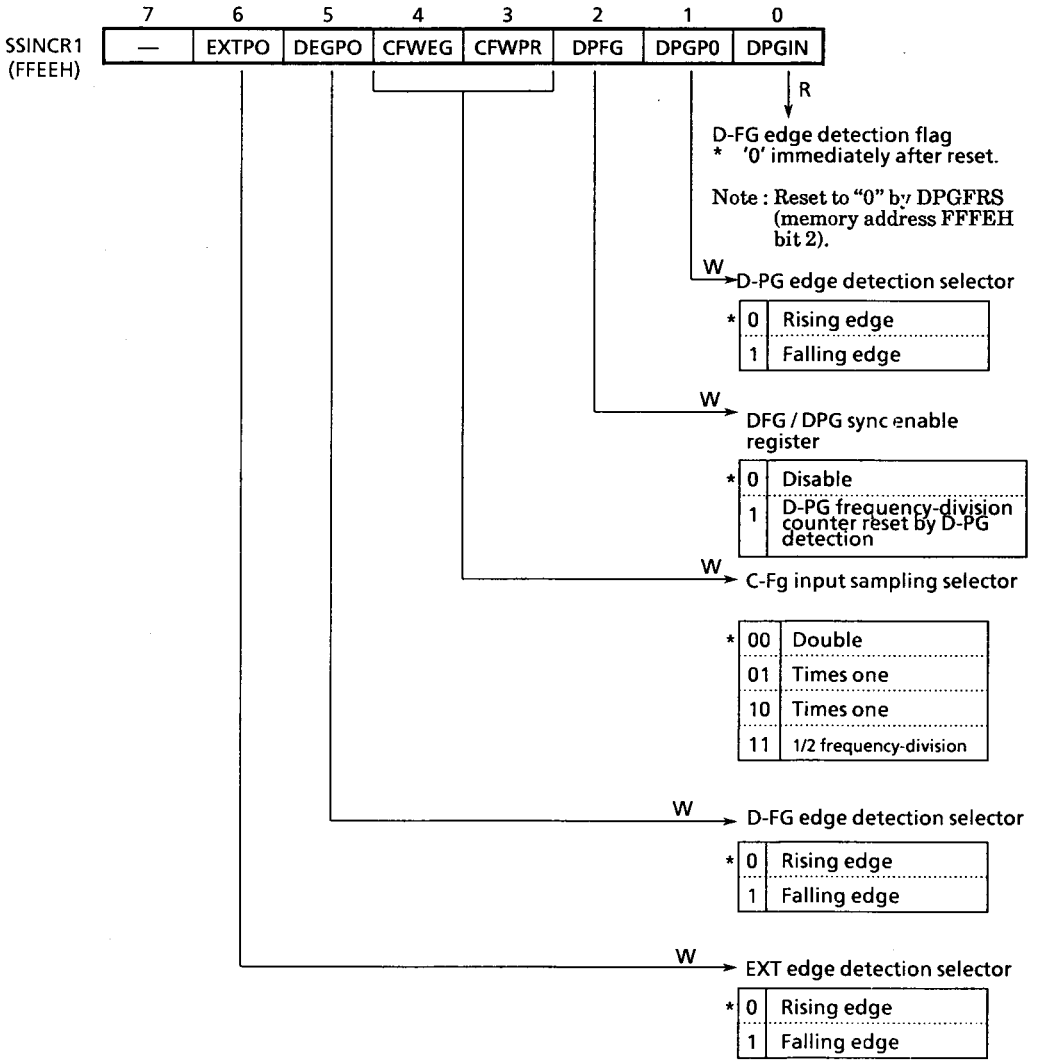
*	0	Enable
	1	Disable

R / W → Signal input for 50/60 discrimination

*	0	D-FF (video switch)
	1	Embedded signal

R / W → TPG1/TPG0 data control

*	0	TPG0
	1	TPG1


SSINCR1
(FFEEH)
7 6 5 4 3 2 1 0
— EXTPO DEGPO CFWEG CFWPR DPFG DPGP0 DPGIN
R
D-FG edge detection flag
* '0' immediately after reset.
Note : Reset to "0" by DPGFRS (memory address FFFEH bit 2).
W
D-PG edge detection selector
* 0 Rising edge
1 Falling edge
W
DFG / DPG sync enable register
* 0 Disable
1 D-PG frequency-division counter reset by D-PG detection
W
C-Fg input sampling selector
* 00 Double
01 Times one
10 Times one
11 1/2 frequency-division
W
D-FG edge detection selector
* 0 Rising edge
1 Falling edge
W
EXT edge detection selector
* 0 Rising edge
1 Falling edge

4. ELECTRICAL CHARACTERISTICS

4.1 Absolute Maximum Ratings

Parameter	Symbol	Pins	Rating	Unit
Supply voltage	V_{CC}		−0.5~+7	V
Input voltage	V_{IN}		−0.5~V_{CC}+0.5	V
Output voltage	V_{OUT1} V_{OUT2}	except Sink open drain pin Sink open drain pin	−0.5~V_{CC}+0.5 −0.5~+10	V
Output current (Per 1 pin)	I_{OUT1} I_{OUT2}	P_{20}	−20 −3	mA
Input current (Per 1 pin)	I_{IN1} I_{IN2} I_{IN3} I_{IN4}	P20 Except Sink open drain pin but include ports P37, P21~P23 P21~P23	20 30 10 2	mA
Input/Output current (Total)	ΣI_{OUT} ΣI_{IN}		−60 120	mA
Power Dissipation (T_{opr} = 70°C)	P_{DF} P_{DS}	Flat package SDIP package	500 600	mW
Soldering Temperature (Time)	T_{SOLDER}		260 (10sec)	°C
Storage Temperature	T_{STG}		−65~+150	°C
Operating Temperature	T_{opr}		−20~+70	°C

4.2 DC Characteristics

Ta = −20~70°C, V_{CC} = 5V ± 10%
Typical values are for TA = 25°C and V_{CC} = 5V

Parameter	Symbol	Min	Max	Typ.	Unit	Test Condition
Input Low Voltage	V_{IL}	−0.3	0.3V_{CC}		V	
	V_{IL1}	−0.3	0.25V_{CC}		V	Hysteresis Input
High Voltage	V_{IH}	0.7V_{CC}	V_{CC}+0.3		V	
	V_{IH1}	0.75V_{CC}	V_{CC}+0.3		V	Hysteresis Input
Output Low Voltage	V_{OL}		0.45		V	@I_{OL} = 1.6mA
Output High Voltage	V_{OH}	2.4			V	@I_{OH} = −200μA
	V_{OH1}	0.9V_{CC}			V	@I_{OH} = −20μA
Output Low Current (Sink open drain)	I_{OL1}		@V_{OL} = 0.45V	2	mA	P37
	I_{OL2}		@V_{OL} = 1.0V	TBD	mA	P_{04}~P_{07}, P_{24}~P_{27}, P_{W0}, P_{W1}
Output Low Current (P20)	I_{OLC}	10			mA	@V_{OL} = 1.0V
Output High Current (P20)	I_{OHC}		−10		mA	@V_{OH} = 3.5V
Hysteresis Voltage	V_{HS}			0.7	V	VCC = 5V, Ta = 25°C
Input Leakage Current	I_{LI1}		±10		μA	0.2≦Vin≦VCC−0.2
Output Leakage Current	I_{LO}		±20		μA	0.2≦Vin≦VCC−0.2
Operating Current	I_{CC1}		50	TBD	mA	@fc = 10MHz, Output Port '1'
	I_{CC2}			TBD		

4.3 AC Characteristics

Ta = −20~70°C V_{CC} = 5V ± 10% CL = 50pF

Symbol	Parameter	10MHz Clock		Variable Clock		Unit
		Min	Max	Min	Max	
t_{OSC}	OSC. Period = X	100		100	1000	ns
t_{CYC}	CLK Period	400		4X	4X	ns
t_{WL}	CLK Low Width	160		2X-40		ns
t_{WH}	CLK High Width	160		2X-40		ns
t_{CPW}	CLK to Data Output		200		X + 100	ns
t_{PRC}	Port Data Setup to CLK	200		2X		ns
t_{CPR}	Port Data Hold After CLK	100		X		ns
t_{CGW}	CLK to TPG Data Output		400		3X + 100	

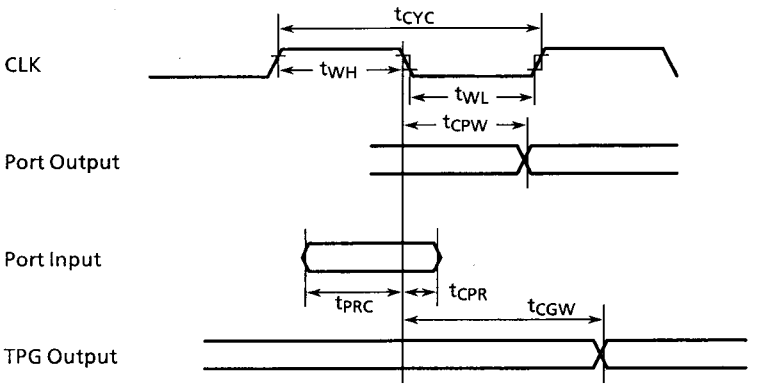
- AC output level High 2.2V/Low 0.8V
- AC input level High 0.8V_{CC}/Low 0.2V_{CC}

4.4 Serial Channel Timing

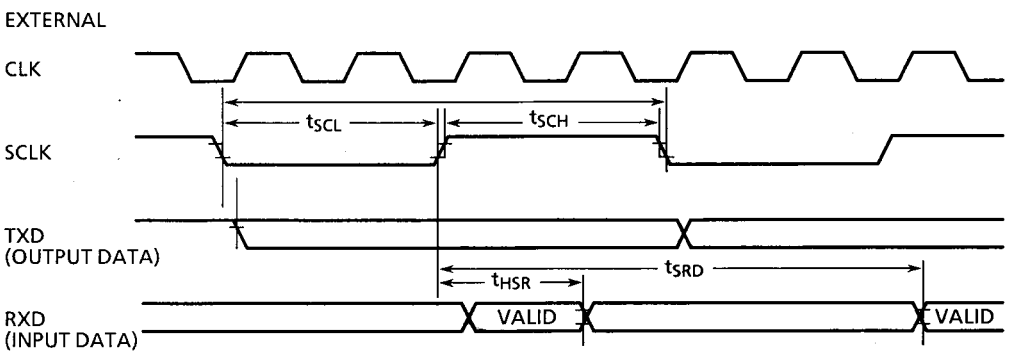
Symbol	Parameter	Condition	10MHz Clock		Variable Clock		Unit
			Min	Max	Min	Max	
t_{SCR}	Serial Port Clock Cycle Time	Internal	800	12800	8X	128X	ns
		External	1600		16X		
t_{SCL}	SCLK Low Width	Internal	*	*	*	*	ns
		External	*	*	*	*	
t_{SCH}	SCLK High Width	Internal	*	*	*	*	ns
		External	*	*	*	*	
t_{SKDO}	SCLKO → TXD (Output Data) Delay Time	Internal	*		*		ns
		External	*		*		
t_{SRD}	SCLK Rising Edge to Input DATA Valid	Internal	*		*		ns
		External	*		*		
t_{HSR}	Input Data Hold After SCLK Rising Edge	Internal	*		*		ns
		External	*		*		

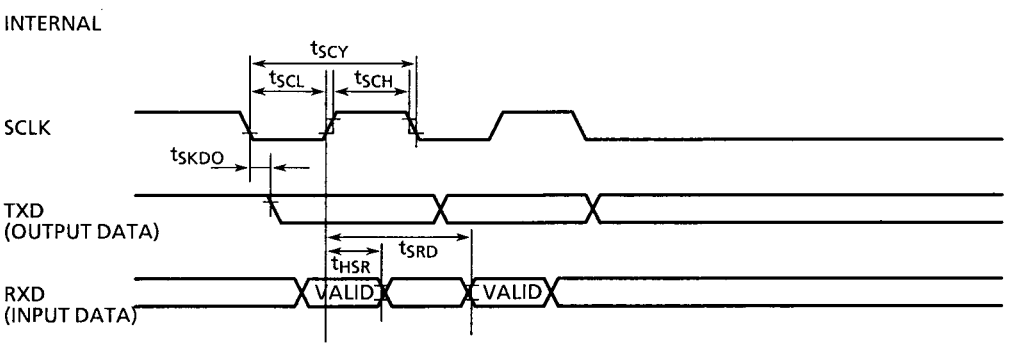
* TBD

4.5 Timing Chart



4.6 Serial Channel Timing Chart



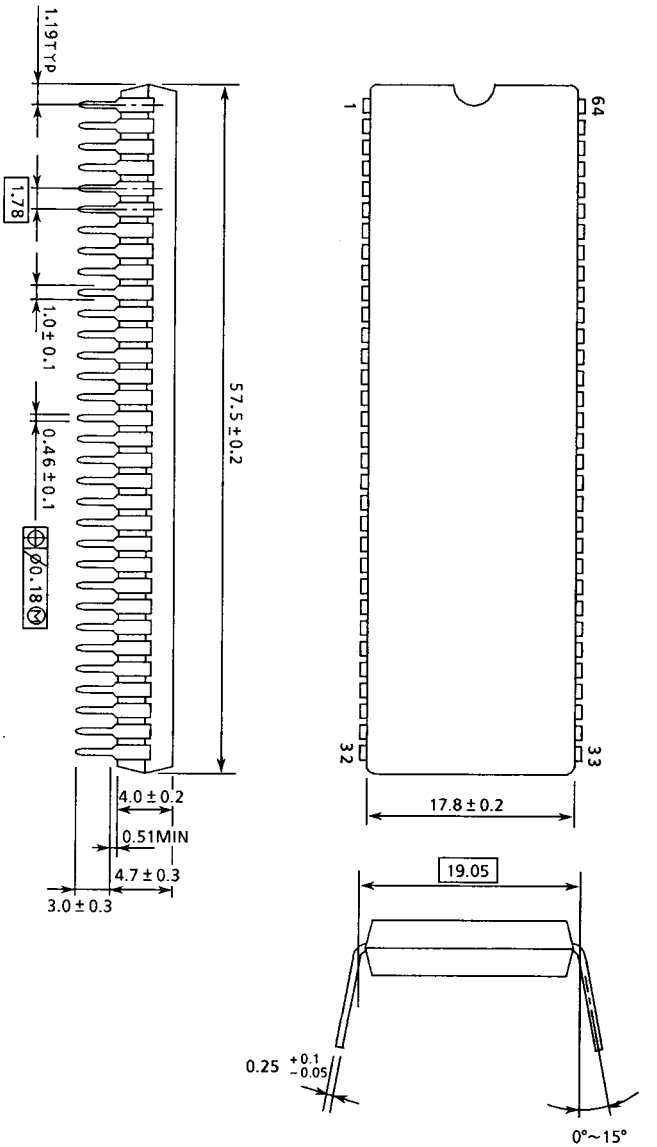


5. OUTSIDE DIMENSIONS

5.1 DIP Pacakage

SDIP64-P-750

Unit : mm

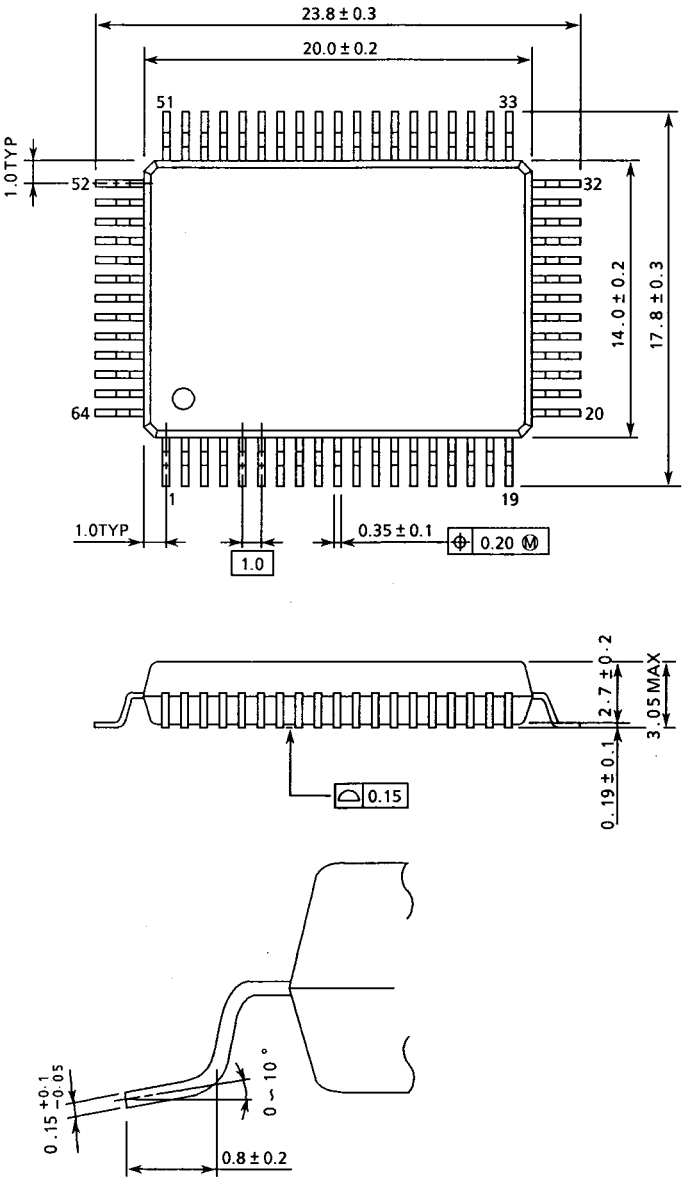


Note : Lead pitch: 1.78
Tolerance: ±0.25 to the theoretical center of each lead obtained as based on the No.1 and No.64 pins.

5.2 Falt Package

QFP64-P-1420A

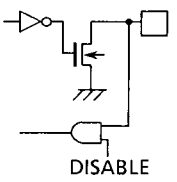
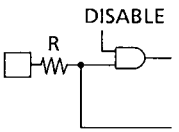
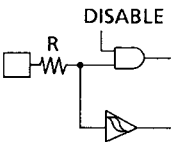
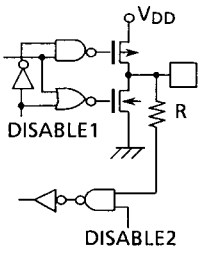
Unit : mm



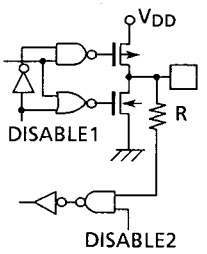
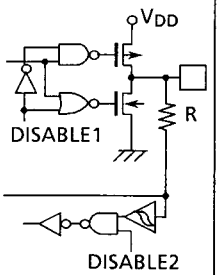
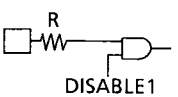
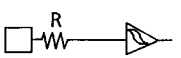
Note : The pin assignments of DIP package and flat package are different.
For writing, refer to 2.1 Pin assignment

6. I/O Port Circuits

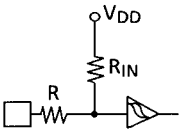
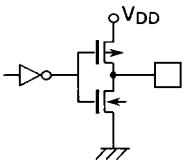
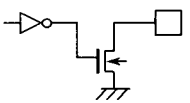
(1/3)

I/O	CIRCUIT	PORT	REMARKS
I/O		P04~P07 P21~P23, P24~P27	Output : Shink open drain output Initial "Hi-z"
Input		P50~P57	
		P64~P67	
I/O		P00, P01~P03 P10, P11~P13 P20 P30~P32 P40~P47	Tri-state I/O P00, P10, and P20 are each controlled by timing pulse generator signals.

(2/3)

I/O	CIRCUIT	PORT	REMARKS
I/O		P60~P63	Tri-state I/O
		P33~P34 P64~P67	Tri-state I/O
Input		P36	
		P35	Hysteresis input

(3/3)

I/O	CIRCUIT	PORT	REMARKS
Input	 R_{IN} = 80KΩ (TYP.) R = 1KΩ (TYP.)	$\overline{RESET}$	Hysteresis input
Output		CLK	Push-pull output
		P37 PW0, PW1	Sink open drain

7. Table of Special Function Registers

The special function registers include the I/O ports and peripheral control registers allocated to the 64-byte address from OFFCOH OFFFFH.

1) I/O port
2) I/O port control
3) Timer/PWM control
4) A/D converter control
5) Serial channel control
6) Watchdog/Timer Base Counter/Interrupt Control
7) Servo Signal input
8) P-CTL/C-Sync Control
9) TPG Control
10) Captur Control

1) I/O Port

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P0	Port 0	0FFC0H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			I/O mode							
P1	Port 1	0FFC2H					P13	P12	P11	P10
							R/W			
							Input mode			
P2	Port 2	0FFC4H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Input mode							
P3	Port 3	0FFC6H	P37	P36	P35	P34	P33	P32	P31	P30
				R		R/W				
			Output only	Input only		I/O mode				
P4	Port 4	0FFC8H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			Input mode							
P5	Port 5	0FFCBH	P57	P56	P55	P54	P53	P52	P51	P50
			R							
			Input only							
P6	Port 6	0FFCCH	P67	P66	P65	P64	P63	P62	P61	P60
			R				R/W			
			Input only				Input mode			

Note: Read/Write
R/W ; Either read or write is possible
R ; Only read is possible
W ; Only write is possible

2) I/O Port Control

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P0CR	Port0 Control Reg.	0FFC1H						P2S	P0S	P0C
								W	W	W
								0	0	0
								P20 Control 0 : I/O Output Control 1 : TPG Output Control	P00 Control 0 : I/O Output Control 1 : TPG Output Control	P00 Control 0 : In 1 : Out
P1CR	Port1 Control Reg.	0FFC3H		Note : Bit 6 of P1CR should be written "0".	TPG1M	P1S	P13C	P12C	P11C	P10C
					W	W	W			
					0	0	0			
					TPG Output Control 0 : TPG0 1 : TPG1	P10 Control 0 : I/O Output Control 1 : TPG Output Control	0 : In 1 : Out			
P2CR	Port2 Control Reg.	0FFC5H	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0							
			0 : In				1 : Out			
P3CR	Port3 Control Reg.	0FFC7H		P37M	P35M	P34C	P33C	P32C	P31C	P30C
				R/W	R/W	R/W				
				0	0	0				
				PWM / Tol Control 0 : PWM8/Tol 1 : Port	INT0 0 : level 1 : edge ↑	0 : In 1 : Out				
P4CR	Port4 Control Reg.	0FFC9H	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			R/W							
			0							
			0 : In				1 : Out			
P4MR	Port4 Mode Control Reg.	0FFCAH					P45M	P44M	P42M	P41M
							R/W	R/W	R/W	R/W
							0	0	0	0
							0 : Port	0 : Port	0 : Port	0 : Port
							1 : RxD1	1 : TxD1	1 : RxD0	1 : TxD0
P6CR	Port6 Control Reg.	0FFCDH		ADHS1	ADHS0	P6M	P63C	P62C	P61C	P60C
				R/W		R/W	R/W			
				0	0	0	0			
				AD Input Channel Control 00 : AN10 01 : AN11 10 : AN12 11 : AN13		AD Input Control 0 : Port 1 : AD Input	0 : In 1 : Out			

3) Timer/PWM Control (1/2)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TREG0 (TCUT0)	8bit Timer Reg.0	0FFD0H	–							
			W							
			Undefined							
	8bit Timer Counter Data Reg.0	0FFD0H	–							
			R							
			Undefined							
TREG1 (TCUT1)	8bit Timer Reg.1	0FFD1H	–							
			W							
			Undefined							
	8bit Timer Counter Data Reg.1	0FFD1H	–							
			R							
			Undefined							
TCUT2	8bit Timer Counter Data Reg.2	0FFD3H	–							
			R							
			Undefined							
TREG2	8bit Timer Reg.2	0FFD4H	–							
			Undefined							
			W							
TCUT3	8bit Timer Counter Data Reg.3	0FFD6H	–							
			Undefined							
			R/W							
TREG3	8bit Timer Reg.3	0FFD7H	–							
			Undefined							
			W							
TCLK (TMR0, 1CR)	Timer0, 1 Source Clock and Control Reg.	0FFD2H	CLBC16	CLBC1	CLBC0	TMOD	TICLK1	TICLK0	TOLCK1	TOCLK0
			R/W	R/W	R/W	R/W	R/W		R/W	
			0	0	0	0	0	0	0	0
			16bit Timer Counter Clear Control 0 : Disable 1 : Enable	Timer1 Counter Clear Control 0 : Disable 1 : Enable	Timer0 Counter Clear Control 0 : Disable 1 : Enable	Timer Mode 0 : 8bit 1 : 16bit	Time1 CLK-in Control 00 : TOOTRG 01 : TBC2 10 : TBC6 11 : TBC10		Time0 CLK-in Control 00 : TBC2 01 : TBC6 10 : TBC10 11 : TI0	
TFFCR	8bit Timer Flip-Flop Control Reg.	0FFD9H	Note : Should be written "0".			TOSEL	TFFC1	TFFC0	TFFIE	TFFIS
						R/W	W		R/W	
						0	–		0	0
						P37 Output Select 0 : PWM8 1 : TO1	00 : Clear TFF 01 : Set TFF 10 : Invert TFF 11 : Don't care		1 : TFF Invert Enable	0 : Rotation for Timer 0
TMR2CR	Timer 2 Control Reg.	0FFD5H					CLBC21	CLBC20	T2CLK1	T2CLK0
							R/W	W	R/W	
							0	–	0	0
							Counter Clear Control 0 : Disable 1 : Enable	Counter Soft Clear 1 : Clear	CLK-in Control 00 : TBC2 01 : TBC6 10 : TI0 11 : C-FG	

3) Timer/PWM Control (2/2)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TMR3CR	Timer3 Control Reg.	0FFD8H		INTC1	INTC0	CLBC31	CLBC30	VDCON	T3CLK1	T3CLK0
				R/W		R/W	W	R/W	R/W	
				0	0	0	–	0	0	0
				Interrupt Control 00 : – 01 : INTT3 10 : Overflow 11 : INTT3 or Overflow		Counter Clear Control 0 : Disable 1 : Enable	Counter Soft Clear 1 : Clear	Up / Down Control 0 : Down 1 : Up	CLK-in Control 00 : TBC2 01 : TBC6 10 : TII 11 : P-CTL	
TRUN	Timer & PWM Run / Stop Control	0FFDAH		RUNPW1	RUNPW0	RUNPW8	T3RUN	T2RUN	T1RUN	T0RUN
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0	0	0	0	0	0	0
				PWM12-1	PWM12-0	PWM8	Timer3	Timer2	Timer1	Timer0
				0 : STOP			1 : START			
PWMDR0	PWM Data Reg.0	0FFDCH	PWD07	PWD06	PWD05	PWD04	PWD03	PWD02	PWD01	PWD00'
			W							
			–							
			Undefined							
PWMDR0B	PWM Data Reg.0B	0FFDDH					PWD08	PWD0A	PWD09	PWD08
							W			
							–			
							Undefined			
PWMDR1	PWM Data Reg.1	0FFDEH	PWD17	PWD16	PWD15	PWD14	PWD13	PWD12	PWD11	PWD10
			W							
			–							
			Undefined							
PWMDR1B	PWM Data Reg.1B	0FFDFH					PWD1B	PWD1A	PWD19	PWD18
							W			
							–			
							Undefined			
PWM8DR	PWM8 Data Reg.	0FFDBH	PW8D7	PW8D6	PW8D5	PW8D4	PW8D3	PW8D2	PW8D1	PW8D0
			W							
			–							
			Undefined							

4) A/D Converter Control

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
ADMOD	A/D Converter mode Reg.	0FFE0H	ADS2	ADBF	ADS	SCSH	ADCH3	ADCH2	ADCH1	ADCH0
			W	R	R/W	R/W	R/W			
			1	0	0	0	0	0	0	0
				0 : Stop 1 : Busy	0 : Stop 1 : Start	0 : Normal (Repeat) 1 : Special	Analog Input Channel Select			
ADREG	A/D Result Register	0FFE1H	–							
			R							
			–							
P6CR	Port6 (A/D-in) Control Reg.	0FFCDH		ADHS1	ADHS0	P6M	P63C	P62C	P61C	P60C
				R/W		R/W	R/W			
				0	0	0	0			
				A/D Input Channel Control 00 : AN10 01 : AN11 10 : AN12 11 : AN13		AD Input Control 0 : Port 1 : AD Input	0 : In 1 : Out			

5) Serial Channel Control

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel0 Buffer Register	0FFE2H	RB07	RB06	RB05	RB04	RB03	RB02	RB01	RB00
			TB07	TB06	TB05	TB04	TB03	TB02	TB01	TB00
			R (Receiv)/W (Send)							
			Undefined							
SC0MOD	Serial Channel0 Mode Register	0FFE3H	FFOSI	SORES	SOMD1	SOMD0	SIIOFT	CLKOSI	SCKOS	SIO0E
			R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			1	0	0	0	0	0	0	0
			0 : Busy 1 : Stop	0 : – 1 : Reset	Transmission mode 00 : Send 01 : Receiv 10 : Don't Care 11 : Send-receive		Transmission Shift 0 : rising-edge 1 : falling-edge	0 : TBC2 1 : TBC6	Transmission clock 0 : Internal Clock 1 : External Clock	Transmission Control 0 : Disable 1 : Enable
SC1BUF	Serial Channel1 Buffer Register	0FFE4H	RB17	RB16	RB15	RB14	RB13	RB12	RB11	RB10
			TB17	TB16	TB15	TB14	TB13	TB12	TB11	TB10
			R (Receiv) / W (Send)							
			Undefined							
SC1MOD	Serial Channel1 Mode Register	0FFE5H	FF1SI	S1RES	S1MD1	S1MD0	SII1FT	CLK1SI	SCK1S	SIO1E
			R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			1	0	0	0	0	0	0	0
			0 : Busy 1 : Stop	0 : – 1 : Reset	Transmission mode 00 : Send 01 : Receiv 10 : Don't Care 11 : Send/Receiv		Transmission Shift 0 : rising-edge 1 : falling-edge	0 : TBC2 1 : TBC6	Transmission clock 0 : Internal Clock 1 : External Clock	Transmission Control 0 : Disable 1 : Enable

6) Watchdog/Timer Base Counter/Interrupt Control

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
WDMOD	Watch Dog Timer Mode Reg.	0FFE6H		FTBC2	FTBC1	WDTE	FTPG	HALT	EXF	DRVE
				R/W	R/W	R/W	R/W	R/W	R	R/W
				0	0	1	0	0	undefined	0
				1 : $TBC_{15\sim18}$ Interrupt Flag	1 : $TBC_{11\sim14}$ Interrupt Flag	1 : WDT Enable	1 : $TPG_{0\sim13}$ Interrupt Flag	Stand-by mode 0 : Run Mode 1 : Stop Mode	Invert each time EXX instruction is executed	1 : to drive pin in STOP mode
WDCR	Watch Dog Timer Control Reg.	0FFE7H	Watchdog timer clear disable code Register							
			W							
			–							
			B1H : WDT Disable Code 4EH : WDT Clear Code							
TBMOD	Time Bease Counter Mode Reg.	0FFE8H		TBCCLR	INTS2	INTS1	INT · TBC2		INT · TBC1	
				R/W	R/W	R/W	R/W		R/W	
				0	0	0	0	0	0	0
				TBC Clear 1 : Clear	TBC Interrupt Enable Reg. 00 : Disable 01 : INT TBC1 10 : INT TBC2 11 : INT TBC1 & INT TBC2		TBC Interrupt select 2 00 : TBC15 01 : TBC16 10 : TBC17 11 : TBC18		TBC Interrupt select 1 00 : TBC11 01 : TBC12 10 : TBC13 11 : TBC14	
INTEL	Interrupt Enable Mask Reg.	0FFE9H	ENISI0	ENIT0	ENIT1	ENIT2	ENIT3	ENITB	ENI1	ENI2VA
			R/W							
			0	0	0	0	0	0	0	0
			1 : Enable				0 : Disable			
INTEH (DMAEL)	Micro DMA Enable Reg.	0FFEAH	Note : Should be written "0".	DE0	DECAP	DEAD	INT2S	ENI0	ENICAP	ENIAD
				R/W			R/W			
				0	0	0	0	0	0	0
				1 : Enable 0 : Disable			1 : Enable 0 : Disable			
DMAEH	Micro DMA Enable Reg.	0FFEBH	0	DET0	DET1	DET2	DET3	DETB	DE1	DE2VA
			R/W							
			0	0	0	0	0	0	0	0
			1 : Enable				0 : Disable			
IREL	Interrupt Request Flag	0FFECH		IRF0	IRFCAP	IRFAD			SIO2F	SIO1F
				R					R	R
				0	0	0			0	0
				1 : Interrupt being requested					1 : SIO1 Request	1 : SIO0 Request
IRFH		0FFEDH	IRFSIO	IRFT0	IRFT1	IRFT2	IRFT3	IRFTE	IRF1	IRF2VA
			R							
			0	0	0	0	0	0	0	0
			1 : Interrupt being requested							

7) Servo Signal Input

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
SINCR	Servo Input Control Reg.	0FFCEH			C-Sync	EXT	P-CTL	C-FG	D-PG	D-FG
					R/W	R/W	R/W	R/W	R/W	R/W
					0	0	0	0	0	0
			Servo Input Signal 0 : Enable 1 : Disable							
SSINCR1	Servo Signal Control Reg.1	0FFEEH		EXTPO	DFGPO	CFWEG	CFWPR	DPFG	DPGO	DPGIN
				W	W	W	W	W	W	R
				0	0	0	0	0	0	0
				EXT Edge Detection 0 : ↑ 1 : ↓	DFG Edge Detection 0 : ↑ 1 : ↓	CFG frequency-divider Selector 0 : 1 : 1/2	CFG Edge Detection 0 : ↑ ↓ 1 : ↑	DFG/DPG asynchronous Reg. 0 : Disable 1 : asynchronous	DPG Edge Detection 0 : ↑ 1 : ↓	DFG Edge Detect Flag
SSINCR2	Servo Signal Control Reg.2	0FFFFH					TP	DFMA	PB	CFGMCP
							R/W	R/W	R/W	R
							0	0	0	0
							TPG0/TPG1 Reg. Control 0 : TPG0 1 : TPG1	*1	*2	CFG Run away Flag
CSYNCR	(C-Sync Control Reg.)	0FFEFH	CSYNCF	TPGF	PCTPO	OEDR	NPS	CTLDTY	VISSFL	VASSFL
			R	R	W	R	R	R	R	R
			0	0	0	0	0	0	0	0
			V-Sync Detection Flag 1 : V-Sync	quasi-V Detection Flag 1:embedded (TRG1-13)	PCTL Signal Control 0 : Forward 1 : Reverse	Odd / Even 0 : Even 1 : Odd	60 / 50 0 : 60 1 : 50	CTL Duty 0 : 22.5% 1 : 60%	VISS Flag 1 : VISS data	VASS Flag 1 : VASS data
PCFCR1	C-FG Mask Control Reg.	0FFF0H	CFMSK3	CFMSK2	CFMSK1	CFMSK0	VISS3	VISS2	VISS1	VISS0
			W				W			
			0	0	0	0	0	0	0	0
			C-FG Mask Time Control				VISS discrimination control			
PCDFCR	D-FG /P-CTL Control Reg.	0FFF1H	DFGPR2	DFGPR1	DFGPR0	PCTPR4	PCTPR3	PCTPR2	PCTPR1	PCTPR0
			W			W				
			0	0	0	0	0	0	0	0
			C-FG 3bit Prescaler Control			P-CTL 5bit Prescaler Control				
PCFSCR	C-FG Control Reg.	0FFF2H	DVS	PCTLCK2	PCTLCK1	CFGPR4	CFGPR3	CFGPR2	CFGPR1	CFGPR0
			W	W		W				
			0	0	0	0	0	0	0	0
			SYNCDV Reg. Select 0 : VPD. SYNC 1 : FEP	CTL Duty Detection Input Clock 00 : TBC2 10 : TBC6 01 : TBC4 11 : TBC8		C-FG 5bit Prescaler Control				
SYNCDV	C-Sync Input Control Reg.	FFCF (DVS = 0)		SYNP02	SYNP01	SYNP00	VPP04	VPP03	VPP02	VPP01
				W			W			
				0	0	0	0	0	0	0
				C-Sync Mask Control			C-Sync frequency-divider control			
	Even/Odd Sompling Control Reg.	(DVS = 1)						FEP02	FEP01	FEP00
								W		
								0	0	0
								Even / Odd judgement		

*1 : 50/60 field discriminator Input Control, 0 : D-FF1 1 : Embeded

*2 : Vertical sync detector Output Control, 0 : Enable 1 : Desable

8) PCTL/C-Sync Control

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
PCFCR1	VISS Control Reg.	0FFF0H	CFMSK3	CFMSK2	CFMSK1	CFMSK0	VISS3	VISS2	VISS1	VISS0
			W				W			
			0	0	0	0	0	0	0	0
			C-FG Mask Time Control				VISS discrimination control			
PCFSCR	C-Sync Control Reg.	0FFF2H	DVS	PCTLCK2	PCTLCK1	CFGPR4	CFGPR3	CFGPR2	CFGPR1	CFGPR0
			W	W			W			
			0	0	0	0	0	0	0	0
			SYNCDV Reg. Select 0 : VPP, SYNC 1 : FEP	CTL Duty Detection Input Clock 00 : TBC2 10 : TBC6 01 : TBC4 11 : TBC8		C-FG 5bit Prescaler Control				
PCTLD	VASS Data Reg.	0FFF3H	VASS7	VASS6	VASS5	VASS4	VASS3	VASS2	VASS1	VASS0
		①	R							
			0	0	0	0	0	0	0	0
			VASS Data Low Byte							
			VASSF	VASSE	VASSD	VASSC	VASSB	VASSA	VASS9	VASS8
		②	R							
			0	0	0	0	0	0	0	0
			VASS Data High Byte							
CSYNCR	C-Sync Control Reg.	0FFEEH	CSYNCF	TPGF	PCTPO	OEDR	NPS	CTLDTY	VISSFL	VASSFL
			R	R	W	R	R	R	R	R
			0	0	0	0	0	0	0	0
			V-Sync Detection Flag 1 : V – Sync	quasi-V Detection Flag 1:Embedded (TRG1-13)	PCTL Signal Control 0 : Forward 1 : Reverse	Odd/Even 0 : Even 1 : Odd	60/50 0 : 60 1 : 50	CTL Duty 0 : 22.5% 1 : 60%	VISS Flag 1 : VISS data	VASS Flag 1 : VASS data
SSINCR2	C-Sync Control Reg.	0FFFFH					TP	DFMA	PB	CFGMCP
							R/W	R/W	R/W	R
							0	0	0	0
							TPG0/TPG1 Reg. Control 0 : TPG0 1 : TPG1	*1	*2	CFG Runaway Flag

*1 : 50/60 field discriminator Input Control, 0 : D-FF1 1 : Embeded

*2 : Vertical sync detector Output Control, 0 : Enable 1 : Disable

9) TPG Control (1/2)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TPCREG0	T.P.G.0 Control Reg.	0FFF8H			EMPINT0	TPFUL0	TPEMP0	TPF02	TPF01	TPF00
					W	W	W	R		
					0	0	0	0	0	0
					TPG0 Data Empty Interrupt Control Reg. 0 : Disable 1 : Enable	TPG0 Data Full Flag 1 : FULL	TPG0 Data Empty Flag 1 : Empty	TPG0 FIFO Status Flag. 000 Empty / full 100 Data 4 001 Data 1 101 Data 5 010 Data 2 110 Data 6 011 Data 3 111 Data 7 Empty when '000' and when TPEMP1 is "1" Full when '000' and when TPFUL is "1"		
TPCREG1	T.P.G.1 Control Reg.	0FFF9H			EMPINT1	TPFUL1	TPEMP1	TPF12	TPF11	TPF10
					W	R	R	R		
					-	0	0	0	0	0
					TPG1 Data Empty Interrupt Control Reg. 0 : Disable 1 : Enable	TPG1 Data Full Flag 1 : FULL	TPG1 Data Empty Flag 1 : Empty	TPG1 FIFO Status Flag. 000 Empty / full 100 Data 4 001 Data 1 101 Data 5 010 Data 2 110 Data 6 011 Data 3 111 Data 7 Empty when '000' and when TPEMP1 is "1" Full when '000' and when TPFUL is "1"		
TPCDAR0	TPG Comparison Data Reg.0	0FFF6H (TPF = 1)	TPC17	TPC16	TPC15	TPC14	TPC13	TPC12	TPC11	TPC10
			W							
			-							
			TPG1-0~TPG1-7 Comparator Data Register							
		(TPF = 0)	TPC07	TPC06	TPC05	TPC04	TPC03	TPC02	TPC01	TPC00
			W							
			-							
			TPG0-0~TPG0-7 Comparator Data Register							
TPCDAR1	TPG Comparison Data Reg.1	0FFF7H (TPF = 1)	TPC1F	TPC1E	TPC1D	TPC1C	TPC1B	TPC1A	TPC19	TPC18
			W							
			-							
			TPG1-8~TPG1-15 Comparator Data Register							
		(TPF = 0)	TPC0F	TPC0E	TPC0D	TPC0C	TPC0B	TPC0A	TPC09	TPC08
			W							
			-							
			TPG0-8~TPG0-15 Comparator Data Register							
TPODAR0	TPG Out Data Reg.0	0FFF4H (TPF = 1)	TPO17	TPO16	TPO15	TPO14	TPO13	TPO12	TPO11	TPO10
			W							
			-							
			TPG1-0~TPG1-7 Output Data Register							
		(TPF = 0)	TPO07	TPO06	TPO05	TPO04	TPO03	TPO02	TPO01	TPO00
			W							
			-							
			TPG0-0~TPG0-7 Output Data Register							

9) TPG Control (2/2)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TPODAR1	TPG Out Data Reg.1	0FFF5H (TPF = 1)	TPO1F	TPO1E	TPO1D	TPO1C	TPO1B	TPO1A	TPO19	TPO18
			W							
			–							
			TPG1-8~TPG1-15 Output Data Register							
		(TPF = 0)	TPO0F	TPO0E	TPO0D	TPO0C	TPO0B	TPO0A	TPO09	TPO08
			W							
			–							
			TPG0-8~TPG0-15 Output Data Register							
SVCFREG	Servo Flag Control Reg.	0FFFEH	VISFRS	VASFRS	TPFRS1	TPFRS0	CFGFRS	DPGFRS	CASFWR	CAFRS
			R/W (Read is always '0')							
			0	0	0	0	0	0	0	0
			VISS Status Flag 0: – 1: Clear	VASS Status Flag 0: – 1: Clear	TPG1 FIFO Counter 0: – 1: Reset	TPG2 FIFO Counter 0: – 1: Reset	C-FG Runaway Flag 0: – 1: Clear	D-FG Input Status Flag 0: – 1: Clear	Softwear Captur (EXT) 0: – 1: Captur	Captur FIFO Counter 0: – 1: Reset

10) Captur Control

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
CAPFST	Captur FIFO Status Reg.	0FFFDH	CAPF7	CAPF6	CAPF5	CAPF4	CAPF3	CAPF2	CAPF1	CAPF0
			R							
			0	0	0	0	0	0	0	0
			0 : non-Data				1 : Data			
CAPREG0	Captur Data 0	0FFFAH	CAPD7	CAPD6	CAPD5	CAPD4	CAPD3	CAPD2	CAPD1	CAPD0
			R							
			–							
			Captur Data Low Byte							
CAPREG1	Captur Data 1	0FFFBH	CAPD15	CAPD14	CAPD13	CAPD12	CAPD11	CAPD10	CAPD9	CAPD8
			R							
			–							
			Captur Data High Byte							
CAPREG2	Captur Data 2	0FFFCH	EXT	P-CTL	D-PG	C-SYNC	C-FG	D-FG	CAPD17	CAPD16
			R				R			
			–				–			
			Captur Input Status				Captur Data			
SVCFREG	Servo Flag Control Reg.	0FFFEH	VISFRS	VASFRS	TPFRS1	TPFRS0	CFGFRS	DPGFRS	CASFWR	CAFRS
			R/W (Read is always '0')							
			0	0	0	0	0	0	0	0
			VISS Status Flag 0: – 1: Clear	VASS Status Flag 0: – 1: Clear	TPG1 FIFO Counter 0: – 1: Reset	TPG0 FIFO Counter 0: – 1: Reset	C-FG Runaway Flag 0: – 1: Clear	D-PG Input Status Flag 0: – 1: Clear	Software Captur (EXT) 0: – 1: Captur	Captur FIFO Counter 0: – 1: Reset