

# HD63701Y0, HD637A01Y0, HD637B01Y0

(Limiting Supplies. For Development Only.)

## Description

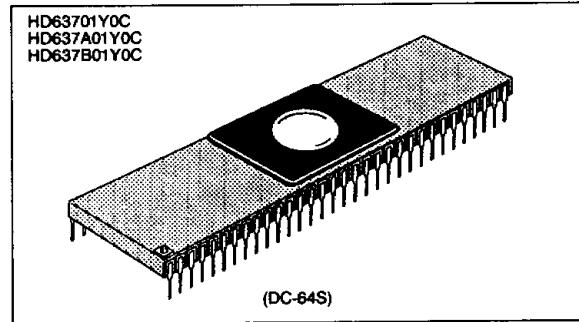
The HD63701Y0 is a CMOS 8-bit single-chip micro-computer unit with 16k-byte EPROM, programmed by the same method as the standard 27256 EPROM. It is available in a ceramic package. The user-programmable on-chip EPROM reduces lead time between software development and production. The ceramic package with window is used in the debugging development stage and small volume production. The ceramic package is a hermetically sealed 64-pin shrink DIP with quartz window that allows for EPROM erasure in the same way as a 27256 EPROM.

## FEATURES

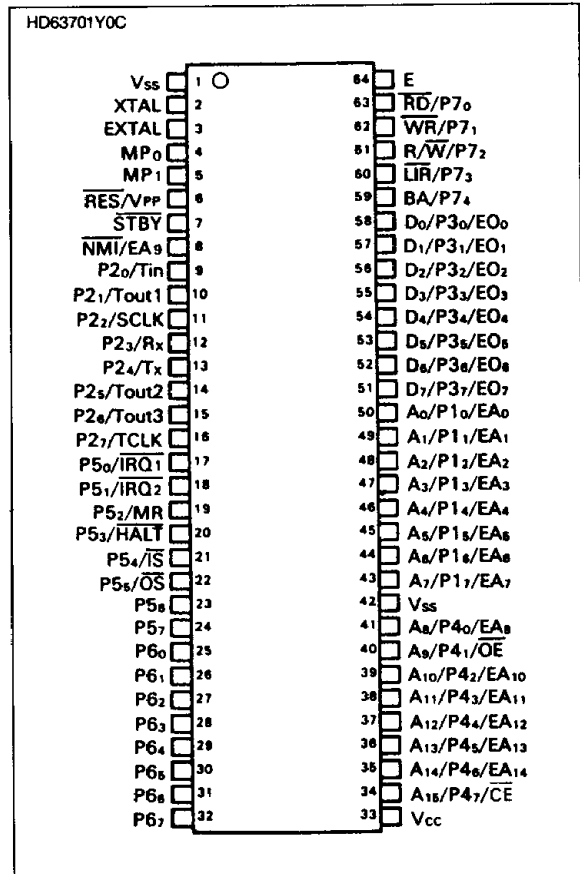
- Instruction set compatible with the HD6301Y0
- 16k bytes EPROM
- 256 bytes RAM
- 53 parallel I/O lines
- Parallel handshake interface (port 6)
- Darlington transistor direct drive lines (port 2) and port 6)
- 16-bit programmable timer
- 8-bit reloadable timer
- Serial communications interface (SCI)
- Three kinds of memory-ready function for low-speed memory access
- Halt function
- Error detection function (address error, opcode error)
- Interrupts
  - 3 external
  - 7 internal
- MCU operation modes
  - Mode 1: expanded mode (internal ROM inhibited)
  - Mode 2: expanded mode (internal ROM valid)
  - Mode 3: single-chip mode
- EPROM programming mode
- Address space up to 65k bytes
- Low power modes
  - Sleep mode
  - Standby mode
- Minimum instruction time 0.5  $\mu$ s ( $f = 2$  MHz)

## PROGRAM DEVELOPMENT SUPPORT TOOLS

- Cross assembler and C compiler software for IBM PC and compatibles
- In circuit emulator for use with IBM PCs and compatibles
- Programming socket adapter for programming the EPROM-on-chip device.



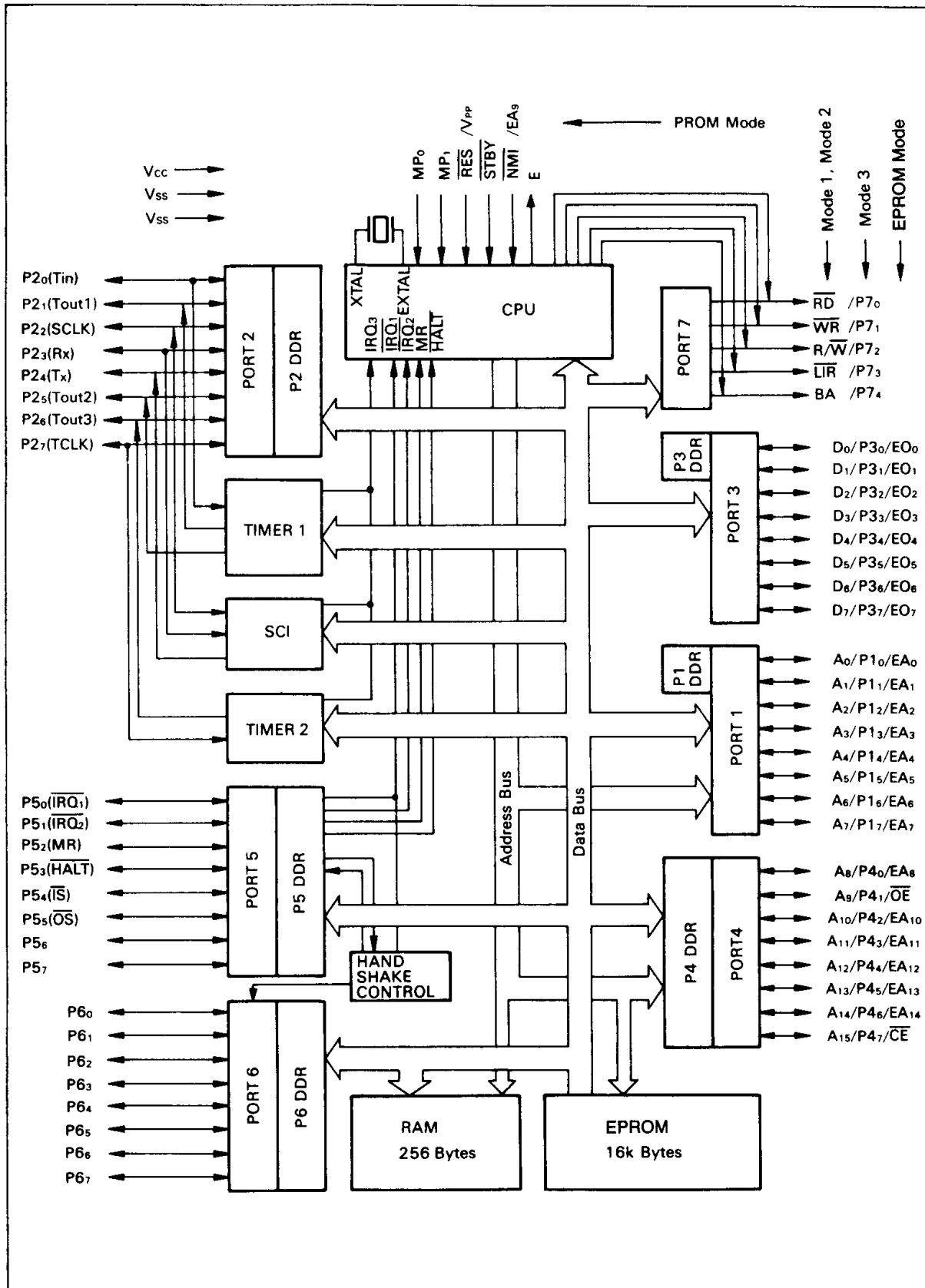
## PIN ARRANGEMENT



Part No.	Clock Freq. (MHz)	Package
HD63701Y0C	0.1 to 1.0	64-pin ceramic shrink DIP (DC-64S)
HD637A01Y0C	0.1 to 1.5	
HD637B01Y0C	0.1 to 2.0	



Block Diagram



# HD63701Y0, HD637A01Y0, HD637B01Y0

## Pin Description

Pin No.	Mode 1, Mode 2		Mode 3		EPROM Mode	
DC-64S	Pin Name	Function	Pin Name	Function	Pin Name	Function
1	V <sub>SS</sub>	Ground	V <sub>SS</sub>	Ground	V <sub>SS</sub>	Ground
2	XTAL	Crystal connection	XTAL	Crystal connection		
3	EXTAL	Crystal or external clock connection	EXTAL	Crystal or external clock connection		
4	MP <sub>0</sub>	Mode select inputs	MP <sub>0</sub>	Mode select inputs		
5	MP <sub>1</sub>		MP <sub>1</sub>			
6	RES	Reset input	RES	Reset input	V <sub>PP</sub>	EPROM Programming voltage
7	STBY	Standby mode input	STBY	Standby mode input		
8	NMI	Nonmaskable interrupt	NMI	Nonmaskable interrupt	EA <sub>9</sub>	Address bus, bit 9
9	P2 <sub>0</sub> /T <sub>in</sub>	Port 2, bit 0/ Timer 1 input	P2 <sub>0</sub> /T <sub>in</sub>	Port 2, bit 0/ Timer 1 input		
10	P2 <sub>1</sub> /T <sub>out 1</sub>	Port 2, bit 1/ Timer 1 output 1	P2 <sub>1</sub> /T <sub>out 1</sub>	Port 2, bit 1/ Timer 1 output 1		
11	P2 <sub>2</sub> /SCLK	Port 2, bit 2/ SCI clock	P2 <sub>2</sub> /SCLK	Port 2, bit 2/ SCI clock		
12	P2 <sub>3</sub> /R <sub>x</sub>	Port 2, bit 3/ SCI receive input	P2 <sub>3</sub> /R <sub>x</sub>	Port 2, bit 3/ SCI receive input		
13	P2 <sub>4</sub> /T <sub>x</sub>	Port 2, bit 4/ SCI transmit output	P2 <sub>4</sub> /T <sub>x</sub>	Port 2, bit 4/ SCI transmit output		
14	P2 <sub>5</sub> /T <sub>out 2</sub>	Port 2, bit 5/ Timer 1 output 2	P2 <sub>5</sub> /T <sub>out 2</sub>	Port 2, bit 5/ Timer 1 output 2		
15	P2 <sub>6</sub> /T <sub>out 3</sub>	Port 2, bit 6/ Timer 2 output 3	P2 <sub>6</sub> /T <sub>out 3</sub>	Port 2, bit 6/ Timer 2 output 3		
16	P2 <sub>7</sub> /TCLK	Port 2, bit 7/ Timer 2 clock	P2 <sub>7</sub> /TCLK	Port 2, bit 7/ Timer 2 clock		
17	P5 <sub>0</sub> /IRQ <sub>1</sub>	Port 5, bit 0/ Interrupt input 1	P5 <sub>0</sub> /IRQ <sub>1</sub>	Port 5, bit 0/ Interrupt input 1		
18	P5 <sub>1</sub> /IRQ <sub>2</sub>	Port 5, bit 1/ Interrupt input 2	P5 <sub>1</sub> /IRQ <sub>2</sub>	Port 5, bit 1/ Interrupt input 2		
19	P5 <sub>2</sub> /MR	Port 5, bit 2/ Memory ready input	P5 <sub>2</sub>	Port 5, bit 2		
20	P5 <sub>3</sub> /HALT	Port 5, bit 3/ Halt input	P5 <sub>3</sub>	Port 5, bit 3		
21	P5 <sub>4</sub> /IS	Port 5, bit 4/ Input strobe	P5 <sub>4</sub> /IS	Port 5, bit 4/ Input strobe		
22	P5 <sub>5</sub> /OS	Port 5, bit 5/ Output strobe	P5 <sub>5</sub> /OS	Port 5, bit 5/ Output strobe		
23	P5 <sub>6</sub>	Port 5,	P5 <sub>6</sub>	Port 5,		
24	P5 <sub>7</sub>	bits 6 and 7	P5 <sub>7</sub>	bits 6 and 7		

(continued)



Pin Description (Cont)

Pin No.	Mode 1, Mode 2		Mode 3		EPROM Mode	
	Pin Name	Function	Pin Name	Function	Pin Name	Function
DC-64S						
25	P6 <sub>0</sub>	Port 6, bits 0-7	P6 <sub>0</sub>	Port 6, bits 0-7		
26	P6 <sub>1</sub>		P6 <sub>1</sub>			
27	P6 <sub>2</sub>		P6 <sub>2</sub>			
28	P6 <sub>3</sub>		P6 <sub>3</sub>			
29	P6 <sub>4</sub>		P6 <sub>4</sub>			
30	P6 <sub>5</sub>		P6 <sub>5</sub>			
31	P6 <sub>6</sub>		P6 <sub>6</sub>			
32	P6 <sub>7</sub>		P6 <sub>7</sub>			
33	V <sub>CC</sub>	+5 V power supply	V <sub>CC</sub>	+5 V power supply	V <sub>CC</sub>	+5 V power supply
34	A <sub>15</sub>	Address bus,	P4 <sub>7</sub>	Port 4, bits 7-0	CE	Chip enable
35	A <sub>14</sub>	bits 15-8	P4 <sub>6</sub>		EA <sub>14</sub>	Address bus,
36	A <sub>13</sub>		P4 <sub>5</sub>		EA <sub>13</sub>	bits 14-10
37	A <sub>12</sub>		P4 <sub>4</sub>		EA <sub>12</sub>	
38	A <sub>11</sub>		P4 <sub>3</sub>		EA <sub>11</sub>	
39	A <sub>10</sub>		P4 <sub>2</sub>		EA <sub>10</sub>	
40	A <sub>9</sub>		P4 <sub>1</sub>		OE	Output enable
41	A <sub>8</sub>		P4 <sub>0</sub>		EA <sub>8</sub>	Address bus, bit 8
42	V <sub>SS</sub>	Ground	V <sub>SS</sub>	Ground	V <sub>SS</sub>	Ground
43	A <sub>7</sub>	Address bus,	P1 <sub>7</sub>	Port 1, bits 7-0	EA <sub>7</sub>	Address bus,
44	A <sub>6</sub>	bits 7-0	P1 <sub>6</sub>		EA <sub>6</sub>	bits 7-0
45	A <sub>5</sub>		P1 <sub>5</sub>		EA <sub>5</sub>	
46	A <sub>4</sub>		P1 <sub>4</sub>		EA <sub>4</sub>	
47	A <sub>3</sub>		P1 <sub>3</sub>		EA <sub>3</sub>	
48	A <sub>2</sub>		P1 <sub>2</sub>		EA <sub>2</sub>	
49	A <sub>1</sub>		P1 <sub>1</sub>		EA <sub>1</sub>	
50	A <sub>0</sub>		P1 <sub>0</sub>		EA <sub>0</sub>	
51	D <sub>7</sub>	Data bus,	P3 <sub>7</sub>	Port 3, bits 7-0	EO <sub>7</sub>	Data bus,
52	D <sub>6</sub>	bits 7-0	P3 <sub>6</sub>		EO <sub>6</sub>	bits 7-0
53	D <sub>5</sub>		P3 <sub>5</sub>		EO <sub>5</sub>	
54	D <sub>4</sub>		P3 <sub>4</sub>		EO <sub>4</sub>	
55	D <sub>3</sub>		P3 <sub>3</sub>		EO <sub>3</sub>	
56	D <sub>2</sub>		P3 <sub>2</sub>		EO <sub>2</sub>	
57	D <sub>1</sub>		P3 <sub>1</sub>		EO <sub>1</sub>	
58	D <sub>0</sub>		P3 <sub>0</sub>		EO <sub>0</sub>	
59	BA	Bus available output	P7 <sub>4</sub>	Port 7, bits 4-0		
60	LIR	Load instruction register output	P7 <sub>3</sub>			
61	R/W	Read/Write output	P7 <sub>2</sub>			
62	WR	Write output	P7 <sub>1</sub>			
63	RD	Read output	P7 <sub>0</sub>			
64	E	External clock output	E	External clock output		

**Pin Function**

**Power (V<sub>CC</sub>, V<sub>SS</sub>)**

V<sub>CC</sub> and V<sub>SS</sub> are the power supply pins. Apply +5V ±10% to V<sub>CC</sub>. Connect the V<sub>SS</sub> pins to ground.

**Clock (XTAL, EXTAL)**

XTAL and EXTAL connected to an AT-cut parallel resonant crystal supply the system clock. The chip has a divide-by-four circuit. For example, if a 4 MHz crystal is used, the system clock will be 1 MHz.

Figure 1 is an example of the crystal oscillator connection. The crystal and C<sub>L1</sub> and C<sub>L2</sub> should be located as close as possible to the XTAL and EXTAL pins. No line must cross the lines between the crystal oscillator and the XTAL and EXTAL pins.

The EXTAL pin can be driven by an external clock with a 45% to 55% duty cycle. The LSI divides the external clock frequency by four. The external clock should therefore be less than four times the maximum clock frequency. When using an external clock, the XTAL pin should be left open.

**Enable Clock (E)**

E provides a system clock to external circuits. Its frequency is one-fourth that of the crystal oscillator or external clock.

**Reset ( $\overline{\text{RES}}$ )**

This pin resets the MCU's internal state and initiates a startup procedure.

**Nonmaskable Interrupt ( $\overline{\text{NMI}}$ )**

When CPU detects a falling edge at the  $\overline{\text{NMI}}$  input, it begins the internal nonmaskable interrupt sequence. The instruction being executed when the NMI is detected will proceed to completion. The interrupt mask bit of the condition code register does not affect the nonmaskable interrupt.

**Interrupt Requests ( $\overline{\text{IRQ}}_1, \overline{\text{IRQ}}_2$ )**

The interrupt requests are level-sensitive inputs which request an internal interrupt sequence from the CPU.

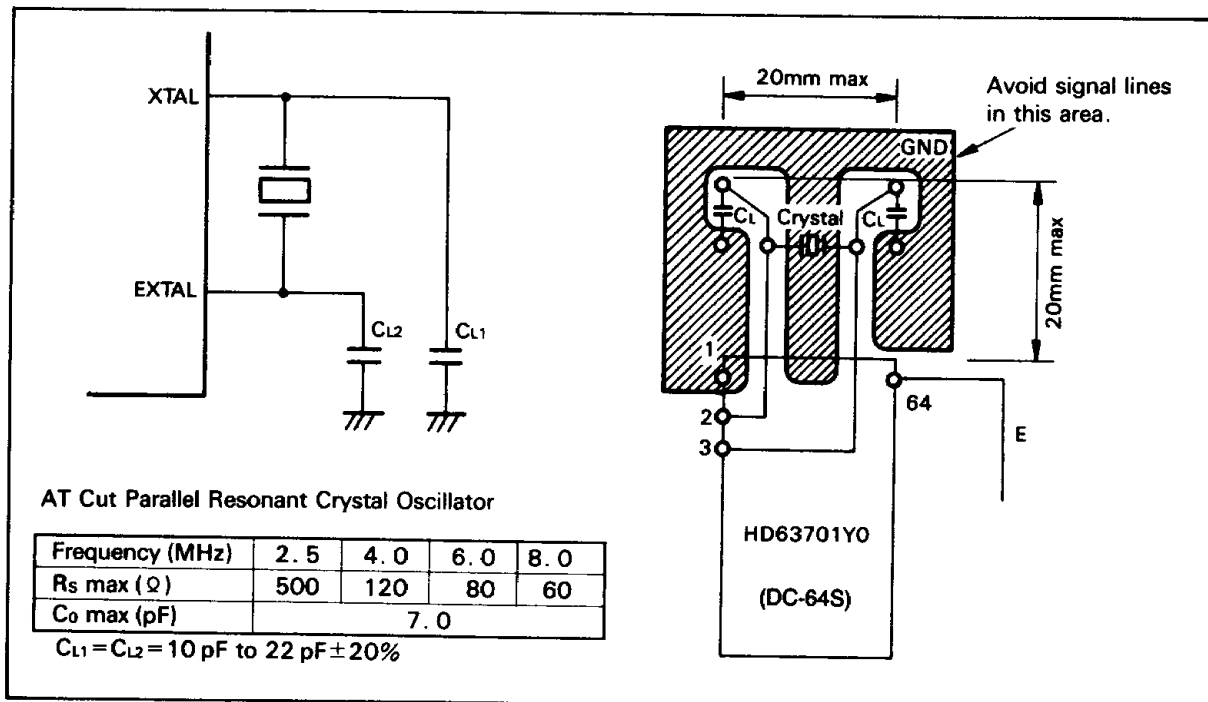


Figure 1. Recommended Crystal Oscillator Connection



**Mode Program ( $\overline{MP}_0$ ,  $\overline{MP}_1$ )**

These pins determine the MCU operation mode (expanded modes 1 or 2, or single-chip mode 3).

**Standby ( $\overline{STBY}$ )**

The  $\overline{STBY}$  pin puts the MCU in standby mode.

**Halt ( $\overline{HALT}$ )**

The halt control input stops instruction execution and release the buses. When  $\overline{HALT}$  switches low, the CPU finishes the current instruction, then stops and enters the halt state.

**Memory Ready ( $\overline{MR}$ )**

The memory ready control input lengthens the system clock's high period to allow access to low-speed memory.

**Read and Write ( $\overline{RD}$ ,  $\overline{WR}$ )**

The read and write outputs show active low outputs to peripherals or memories when the CPU is reading or writing.

**Read/Write ( $\overline{R/W}$ )**

The read/write signal is high when the CPU is reading, and low when it is writing to peripherals or memory.

**Load Instruction Register ( $\overline{LIR}$ )**

The  $\overline{LIR}$  output is low when the CPU executes an opcode fetch cycle.

**Bus Available ( $\overline{BA}$ )**

The bus available output control signal goes high when the CPU accepts  $\overline{HALT}$  and releases the buses. It is normally low.

**Input Handshake, Output Handshake ( $\overline{IS}$ ,  $\overline{OS}$ )**

$\overline{IS}$  and  $\overline{OS}$  are the handshake interface input and output.

**Timer 1 Input ( $\overline{Tin}$ )**

$\overline{Tin}$  is the external input-capture pin for timer 1.

**Timer 1 Outputs 1 and 2 ( $\overline{Tout1}$ ,  $\overline{Tout2}$ )**

$\overline{Tout1}$  and  $\overline{Tout2}$  are the outputs of timer 1's output compare registers.

**Timer 2 Output ( $\overline{Tout3}$ )**

$\overline{Tout3}$  is the output for timer 2.

**Timer 2 Clock Input ( $\overline{TCLK}$ )**

$\overline{TCLK}$  is timer 2's external clock input.

**SCI Clock ( $\overline{SCLK}$ )**

$\overline{SCLK}$  is the serial communication interface (SCI) I/O pin.

**Receive, Transmit ( $\overline{Rx}$ ,  $\overline{Tx}$ )**

$\overline{Rx}$  and  $\overline{Tx}$  are the SCI receive input and transmit output.

**Port 1 ( $\overline{P1_0-P1_7}$ )**

In mode 3 (single-chip mode), these pins are an 8-bit I/O port.

**Port 2 ( $\overline{P2_0-P2_7}$ )**

Port 2 is an 8-bit I/O port.

**Port 3 ( $\overline{P3_0-P3_7}$ )**

In mode 3 (single-chip mode), these pins are an 8-bit I/O port.

**Port 4 ( $\overline{P4_0-P4_7}$ )**

In mode 3 (single-chip mode), these pins are an 8-bit I/O port.

**Port 5 ( $\overline{P5_0-P5_7}$ )**

Port 5 is an 8-bit I/O port.

**Port 6 ( $\overline{P6_0-P6_7}$ )**

Port 6 is an 8-bit I/O port.

**Port 7 ( $\overline{P7_0-P7_4}$ )**

In mode 3 (single-chip mode), port 7 is a 5-bit output port.

**Address Bus ( $\overline{A_0-A_{15}}$ )**

In modes 1 and 2 (expanded modes), these pins are the 16-bit address bus.

**Data Bus ( $\overline{D_0-D_7}$ )**

In modes 1 and 2 (expanded modes), these pins are the 8-bit data bus.

**Chip Enable ( $\overline{CE}$ )**

The chip enable input enables EPROM programming and verifying. When this signal is low, the EPROM is enabled.

---

## HD63701Y0, HD637A01Y0, HD637B01Y0

---

### **Program Voltage ( $V_{PP}$ )**

$V_{PP}$  is the input for the program voltage ( $12.5V \pm 0.3V$ ) for programming the EPROM.

### **Output Enable ( $\overline{OE}$ )**

$\overline{OE}$  is the control for data verification output.

### **EPROM Address Bus ( $EA_0$ - $EA_{14}$ )**

In the PROM programming mode,  $EA_0$ - $EA_{14}$  are the PROM address bus.

### **EPROM Data Bus ( $EA_0$ - $EA_7$ )**

In the PROM programming mode,  $EO_0$ - $EO_7$  are the PROM data bus.

**Functional Description**

**CPU**

**Registers** : Figure 2 is the HD63701Y0 register programming model. The double accumulator D consists of accumulators A and B, so using accumulator D destroys the contents of A and B.

**CPU Operation Modes** : When active, the CPU fetches an instruction from a memory location and executes it. This sequence starts when reset is

cancelled, and repeats if not affected by a special instruction (such as SWI, RTI, WAI, or SLP), or control signal (such as NMI,  $\overline{IRQ_1}$ ,  $\overline{IRQ_2}$ ,  $\overline{IRQ_3}$ , HALT, or  $\overline{STBY}$ ). Other than the active mode, the CPU can be in the reset mode, the halt mode, or either of the low power dissipation modes, sleep or standby. Figure 3 shows the operation mode transitions, and figure 4 is a flow chart of the transition control. Table 1 shows the CPU operating states and port states.

**Table 1. CPU Operation and Port States**

Port	Mode	Reset	STBY <sup>4</sup>	HALT <sup>3</sup>	Sleep
Port 1 (A <sub>0</sub> to A <sub>7</sub> )	Mode 1, 2	High	High Z	High Z	High
	Mode 3	High Z			Keep
Port 2	Mode 1, 2	High Z	High Z	Keep	Keep
	Mode 3				
Port 3 (D <sub>0</sub> to D <sub>7</sub> )	Mode 1, 2	High Z	High Z	High Z	High Z
	Mode 3				Keep
Port 4 (A <sub>8</sub> to A <sub>15</sub> )	Mode 1	High	High Z	High Z	High
	Mode 2	High Z			Note 5
	Mode 3				Keep
Port 5	Mode 1, 2	High Z	High Z	Keep	Keep
	Mode 3				
Port 6	Mode 1, 2	High Z	High Z	Keep	Keep
	Mode 3				
Port 7	Mode 1, 2	Note 1	High Z	Note 2	Note 1
	Mode 3	High Z			Keep

- Notes :
1.  $\overline{RD}$ ,  $\overline{WR}$ , R/W,  $\overline{LIR}$  = High, BA = Low
  2.  $\overline{RD}$ ,  $\overline{WR}$ , R/W = High Z,  $\overline{LIR}$ , BA = High
  3. HALT is unacceptable in mode 3.
  4. E pin goes to high impedance state.
  5. Address output pin = High  
Input port = High Z
  6. Keep: The output port is retained, and the input port goes to the high impedance state.



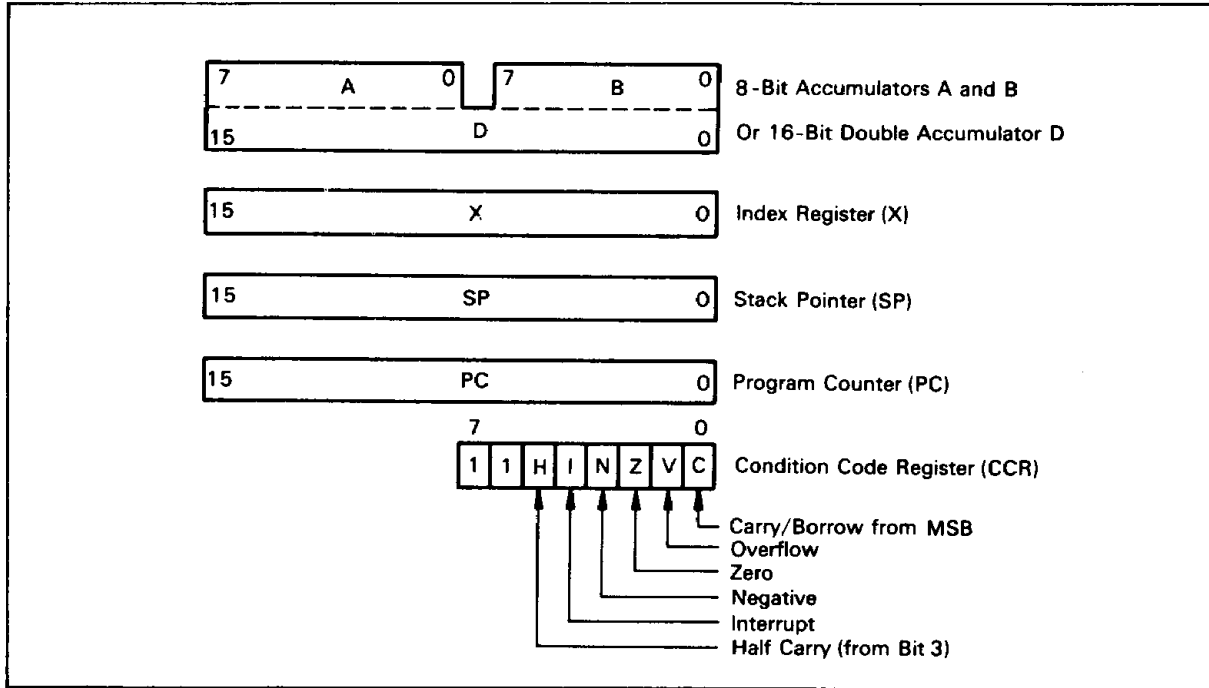


Figure 2. CPU Registers

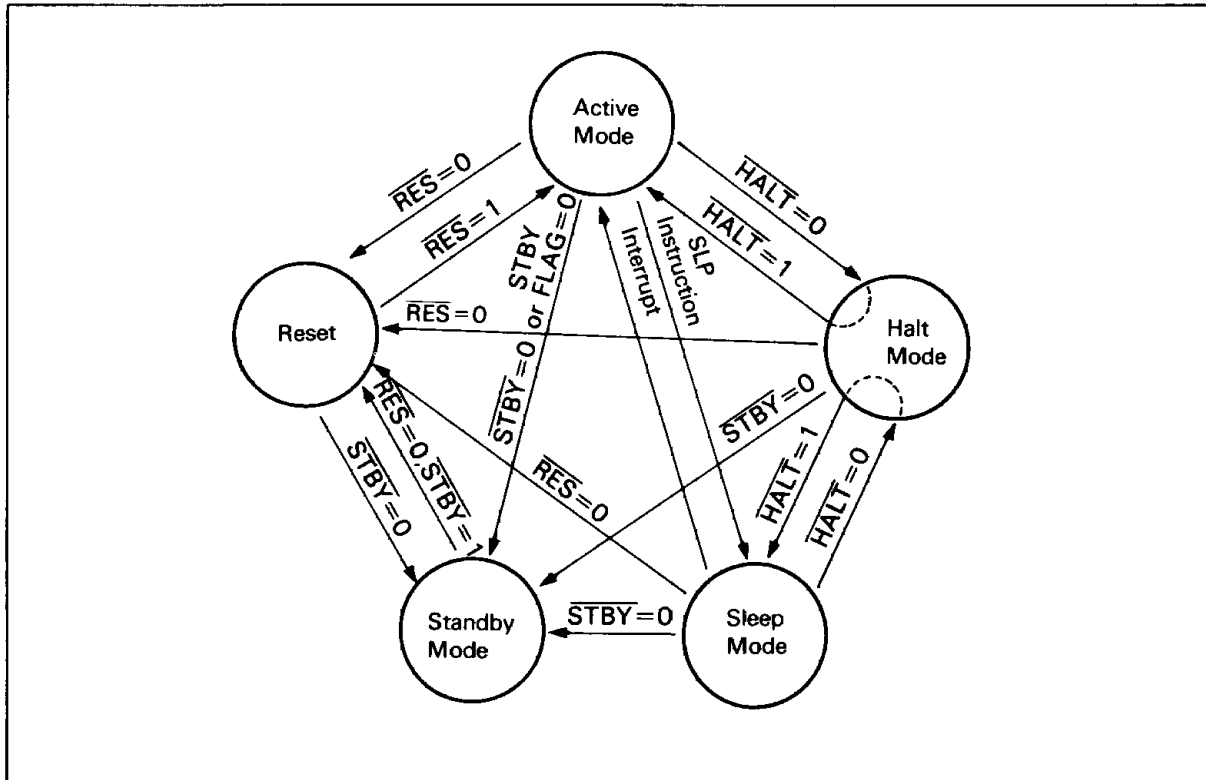


Figure 3. CPU Operation Mode Transitions



# HD63701Y0, HD637A01Y0, HD637B01Y0

**Reset Mode** : To reset the MCU, hold  $\overline{\text{RES}}$  low for at least 3 clock cycles during operation, or for at

least 20 ms during power-up. When  $\overline{\text{RES}}$  is brought high, the MCU begins the reset procedure :

**Table 2. Registers at Reset**

Address	Register	Abbreviation	R/W <sup>2</sup>	Value at Reset <sup>3</sup>
00	Port 1 DDR (Data Direction Register)	P1DDR	W	\$FE
01	Port 2 DDR	P2DDR	W	\$00
02	Port 1	PORT1	R/W	Indefinite
03	Port 2	PORT2	R/W	Indefinite
04	Port 3 DDR	P3DDR	W	\$FE
05	Port 4 DDR	P4DDR	W	\$00
06	Port 3	PORT3	R/W	Indefinite
07	Port 4	PORT4	R/W	Indefinite
08	Timer Control/Status Register 1	TCSR1	R/W	\$00
09	Free Running Counter (MSB)	FRCH	R/W	\$00
0A	Free Running Counter (LSB)	FRCL	R/W	\$00
0B	Output Compare Register 1 (MSB)	OCR1H	R/W	\$FF
0C	Output Compare Register 1 (LSB)	OCR1L	R/W	\$FF
0D	Input Capture Register (MSB)	ICRH	R	\$00
0E	Input Capture Register (LSB)	ICRL	R	\$00
0F	Timer Control/Status Register 2	TCSR2	R/W	\$10
10	Rate/Mode Control Register	RMCR	R/W	\$C0
11	T <sub>X</sub> /R <sub>X</sub> Control Status Register 1	TRCSR1	R/W	\$20
12	Receive Data Register	RDR	R	\$00
13	Transmit Data Register	TDR	W	Indefinite
14	RAM/Port 5 Control Register	RP5CR	R/W	\$F8 or \$78
15	Port 5	PORT5	R/W	Indefinite
16	Port 6 DDR	P6DDR	W	\$00
17	Port 6	PORT6	R/W	Indefinite
18	Port 7	PORT7	R/W	Indefinite
19	Output Compare Register 2 (MSB)	OCR2H	R/W	\$FF
1A	Output Compare Register 2 (LSB)	OCR2L	R/W	\$FF
1B	Timer Control/Status Register 3	TCSR3	R/W	\$20
1C	Time Constand Register	TCONR	W	\$FF
1D	Timer 2 Up Counter	T2CNT	R/W	\$00
1E	T <sub>X</sub> /R <sub>X</sub> Control Status Register 2	TRCSR2	R/W	\$28
1F	Reserved (Note 1)	TSTREG	—	—
20	Port 5 DDR	P5DDR	W	\$00
21	Port 6 Control/Status Register	P6CSR	R/W	\$07
22	Reserved	—	—	—
23	Reserved	—	—	—
24	Reserved	—	—	—
25	Reserved	—	—	—
26	Reserved	—	—	—
27	Reserved	—	—	—

- Notes :
1. Test register. Don't access this register.
  2. R : Read-only register, W : Write-only register, R/W : Read/write register.
  3. Unused bits set to 1.



1. Latch the value of the port mode program pins ( $MP_0, MP_1$ )
2. Initialize the internal registers (table 2)
3. Set the interrupt mask bit
4. Put the contents of the last two addresses ( $\$FFFE, \$FFFF$ ) into the program counter (start address) and start the program from this address.

**Halt Mode** : When the  $\overline{HALT}$  signal is low, the CPU will be in halt mode. In halt mode, the CPU does not execute instructions, and address and data buses are released.

- Internal clocks do not stop
- Peripherals (timer, SCI, etc) function
- Interrupts are accepted

Table 1 shows the state of each port. Halt mode is released when  $\overline{HALT}$  goes high. A  $\overline{RES}$  or  $\overline{STBY}$  signal also releases the halt mode and put the CPU into reset or standby modes, respectively.

**Sleep Mode** : The SLP instruction causes the CPU to stop operation and go into sleep mode. In sleep mode, power dissipation is one-fifth of that in the active mode.

- Internal clocks stop
- Register contents are retained
- Peripherals function
- Interrupts are accepted

Table 1 shows the state of each port. Sleep mode is released by a  $\overline{RES}$ ,  $\overline{STBY}$ , or interrupt. After being released, the CPU goes to reset, standby, or interrupt, respectively.

**Standby Mode** : The MCU goes to standby mode when the  $\overline{STBY}$  signal goes low, or when the STBY flag in the RAM/port 5 control register (RP5CR : \$0014) is cleared by software (figure 5). In standby mode, power dissipation is reduced to several  $\mu A$ .

- All clocks stop
- MCU goes to reset state
- RAM contents are retained

Table 1 shows the state of each port. Reset releases standby mode.

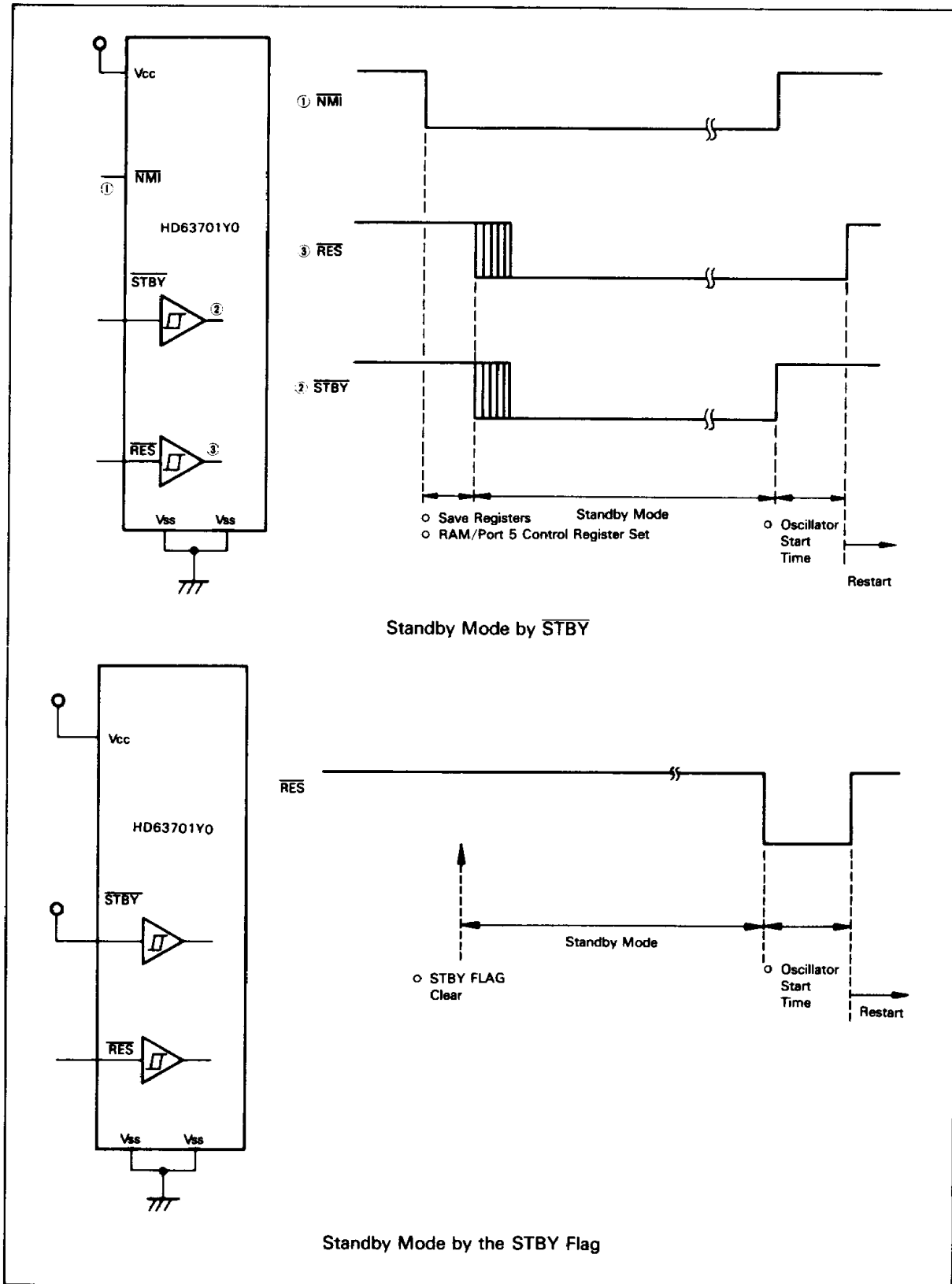


Figure 5. Standby Mode Timing

**Interrupts**

The HD63701Y0 provides 3 external and 7 internal interrupts (figure 6, table 3). At an interrupt request, the CPU will complete the current instruction before beginning the interrupt sequence.

The MCU pushes the program counter, index register, accumulator, and condition code register onto the stack. Then the CPU sets the interrupt mask bit. Finally, it fetches the interrupt's vector (table 4) and branches to the interrupt routine that begins at the vector address.

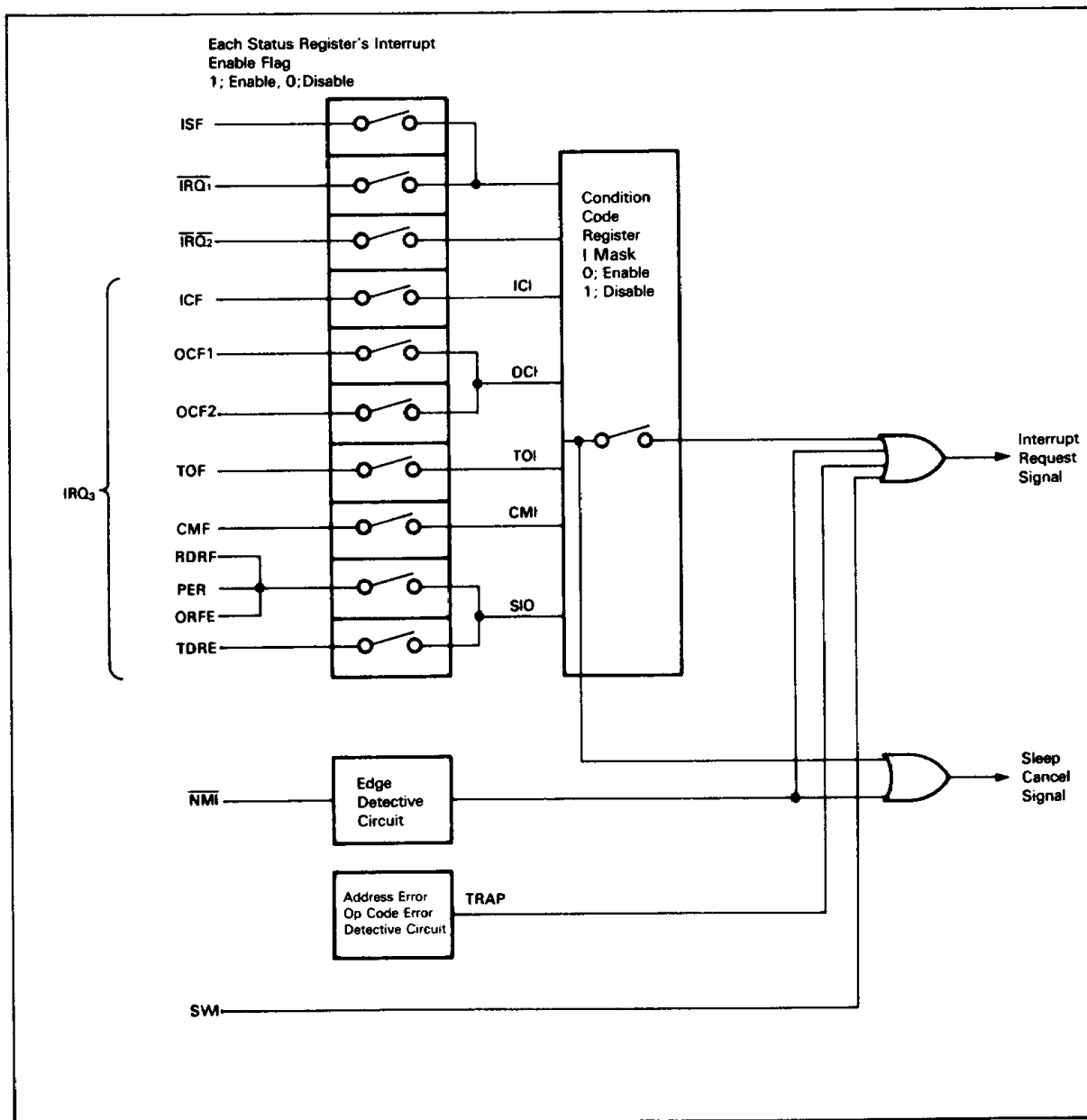


Figure 6. Interrupt Block Diagram

**Table 3. Interrupt Sources**

Type	Interrupt	Symbol	Description
External	Nonmaskable interrupt	NMI	Generated by falling edge of $\overline{\text{NMI}}$ signal. Accepted regardless of interrupt mask bit.
	Interrupt request 1, 2	$\overline{\text{IRQ}}_1$ , $\overline{\text{IRQ}}_2$	Generated by low level at $\overline{\text{IRQ}}_1$ , $\overline{\text{IRQ}}_2$ pins.
	Port 6 input strobe	ISF	Generated by IS
Soft	Software interrupt	SWI	Generated by interrupt instruction
	Trap	TRAP	Generated by fetching undefined instruction or request
Timer	Timer 1 input capture	ICI	The timer and serial interrupts generate the internal interrupt $\overline{\text{IRQ}}_3$
	Timer 1 output compare	OCI	
	Timer 1 overflow	TOI	
	Timer 2 counter match	CMI	
Serial	Serial-Interrupt	SCI	

**Table 4. Interrupt Vectors**

Priority	Vector		Interrupt	kind	
	MSB	LSB			
Highest	FFFE	FFFF	$\overline{\text{RES}}$	—	
↑	FFEE	FFEF	TRAP	Soft	
	FFFC	FFFD	NMI	External	
	FFFA	FFFB	SWI (Software interrupt)	Soft	
	FFF8	FFF9	$\overline{\text{IRQ}}_1$ , ISF (Port 6 input strobe)	External	
	FFF6	FFF7	ICI (Timer 1 input capture)	Timer	
	FFF4	FFF5	OCI (Timer 1 output compare 1, 2)	Timer	
	FFF2	FFF3	TOI (Timer 1 overflow)	Timer	
	FFEC	FFED	CMI (Timer 2 counter match)	Timer	
	FFEA	FFEB	$\overline{\text{IRQ}}_2$	External	
	Lowest	FFF0	FFF1	SIO (RDRF + ORFE + TDRE + PER)	Serial

**Memory Ready**

When MR is high, the system clock operates normally. But when MR is low, the high period will be lengthened depending on its low time in integral multiples of its cycle time. It can be lengthened up to 9  $\mu$ s. The one of three memory ready functions can be chosen by the MRE and AMRE bits in the RAM/port 5 control register (figure 7). See RAM/port 5 control register for details.

During internal address or invalid memory access, MR is prohibited internally from decreasing operation speed. Even in the halt state, MR can lengthen the high period of the system clock to allow peripheral devices to access low-speed memories. MR is also used as P5<sub>2</sub>.

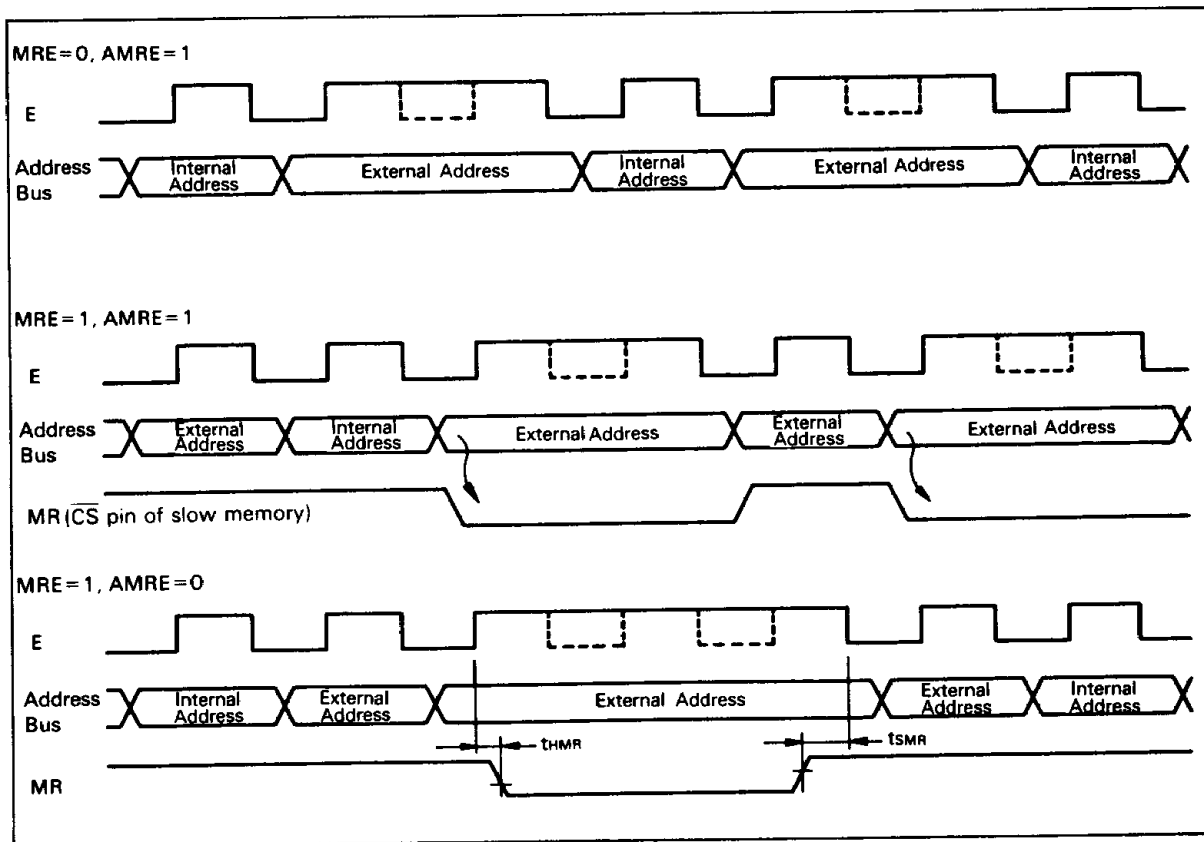
**Trap**

The CPU generates an internal interrupt with the highest priority, TRAP, when it tries to fetch an undefined instruction or address. At a TRAP, the CPU pushes the internal registers onto the stack and restarts from \$FFEE, \$FFEF.

The TRAP prevents the system from failing due to noise or program error. Which addresses are undefined depend on the port mode (table 5).

**Table 5. Undefined Address Space**

Mode 1	Mode 2	Mode 3
\$0000 to \$0027	\$0000 to \$0027	\$0000 to \$003F, \$0140 to \$BFFF



**Figure 7. Memory Ready Timing**



**Port Mode Selection**

The mode program pins, MP<sub>0</sub> and MP<sub>1</sub>, determine the operation mode of the MCU as shown in table 6. These modes allow for either external memory access (expanded modes) or more I/O ports (single-chip mode). Table 7 and figure 8 show how the ports are used in the 3 modes.

**Mode 1 (Expanded)** : In mode 1, port 3 is the data bus. Port 1 is the lower address bus and port 4 is the upper address bus. These buses interface directly with the HMCS6800 buses. Port 7 is used for control

signals. In mode 1, on-chip ROM is disabled, and external address space is expandable up to 65k bytes.

**Mode 2 (Expanded)** : In mode 2, on-chip ROM is available. External address space is expandable to 48k dytes. Port 3 is the data bus. Port 1 is the lower address bus and port 4 can be either the upper address bus or an input port.

**Mode 3 (Single Chip)** : In mode 3, all ports are available for I/O. No external memory can be accessed.

**Table 6. Mode Selection**

Mode	MP <sub>1</sub>	MP <sub>0</sub>	ROM	RAM	Interrupt Vector
1 (Expanded)	Low	High	Ext	Int*	Ext
2 (Expanded)	High	Low	Int	Int*	Int
3 (Single chip)	High	High	Int	Int	Int

Note : \*External RAM can be addressed by clearing the RAME bit in RP5CR (\$0014).

**Table 7. Ports in Each Mode**

Port	Mode 1	Mode 2	Mode 3
1	Address Bus (A <sub>0</sub> —A <sub>7</sub> )	Address Bus (A <sub>0</sub> —A <sub>7</sub> )	I/O Port
2	I/O Port	I/O Port	I/O Port
3	Data Bus (D <sub>0</sub> —D <sub>7</sub> )	Data Bus (D <sub>0</sub> —D <sub>7</sub> )	I/O Port
4	Address Bus (A <sub>8</sub> —A <sub>15</sub> )	I/O Port or Address Bus (A <sub>8</sub> —A <sub>15</sub> )	I/O Port
5	I/O Port	I/O Port	I/O Port
6	I/O Port	I/O Port	I/O Port
7	RD, WR, R/W, LIR, BA	RD, WR, R/W, LIR, BA	Output Port

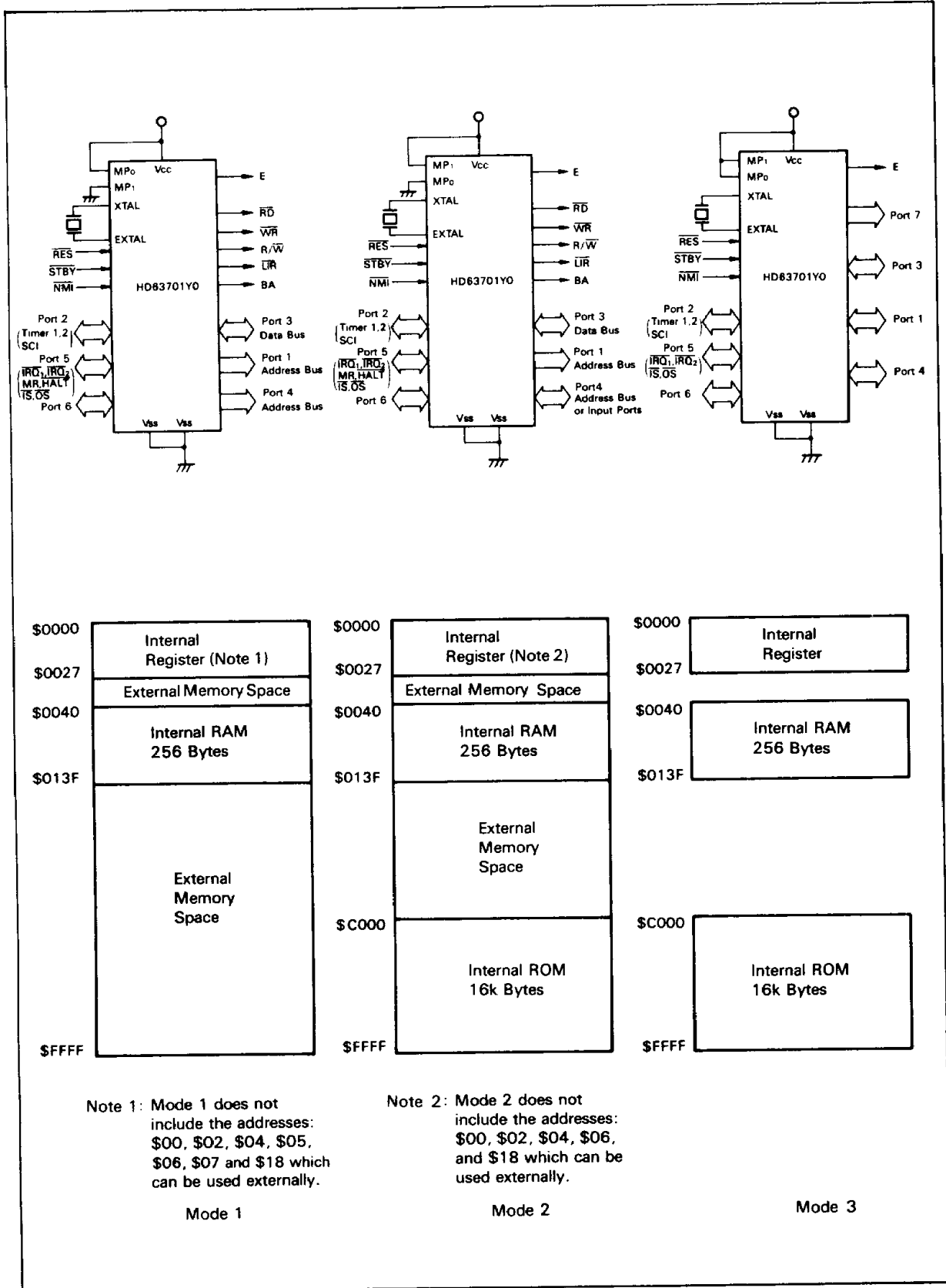


Figure 8. Operation Mode Pin Function and Memory Map



**Ports**

The HD63701Y0 provides six 8-bit I/O ports and one 5-bit output-only port. Each I/O port has a data direction register (DDR) which controls the direction of the port (0=input, 1=output). The DDRs are cleared at reset, and the I/O ports all become input ports.

**Port 1** : Port 1 is an 8-bit I/O port (figure 9). The LSB of the DDR (\$0000) selects the data direction of the whole port (Figure 10). In the expanded modes (1 and 2) port 1 is the lower address bus (A<sub>7</sub>-A<sub>0</sub>). Port 1 can drive one TTL load and 90 pF capacitance.

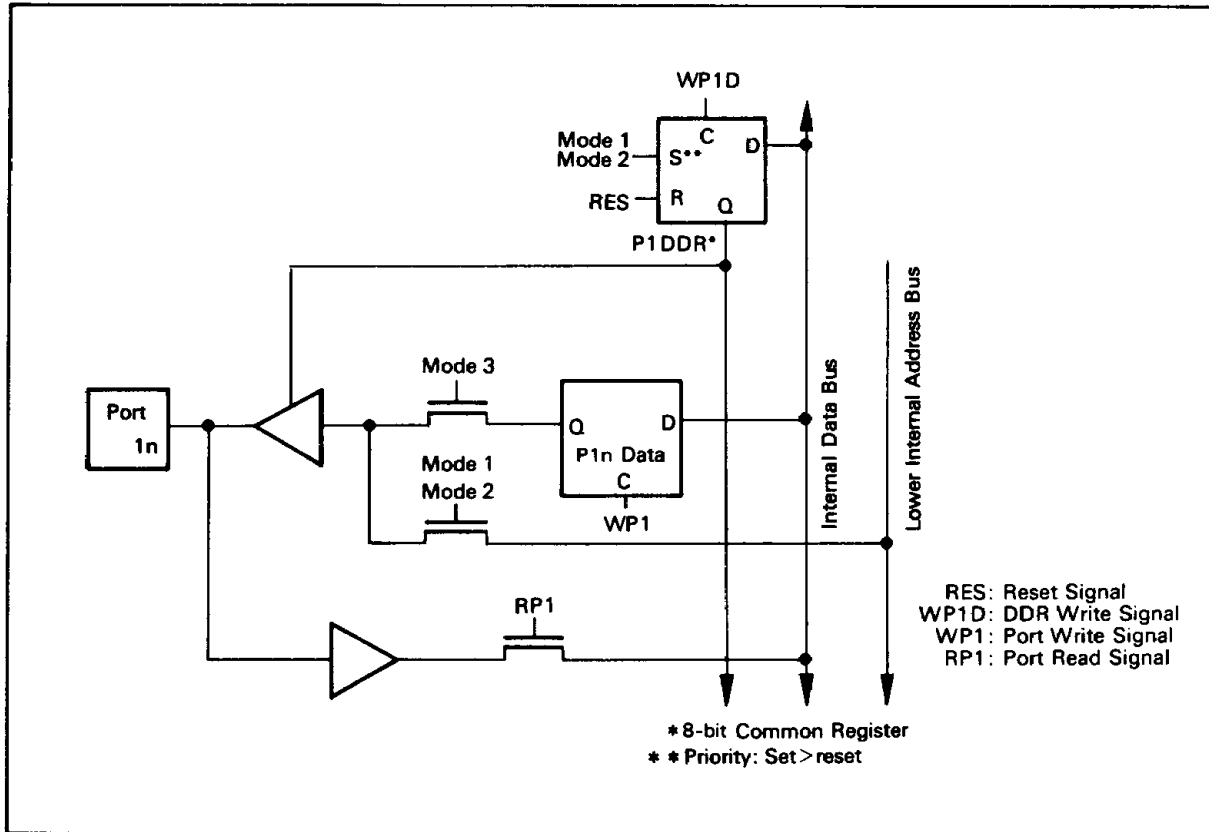


Figure 9. Port 1 Block Diagram

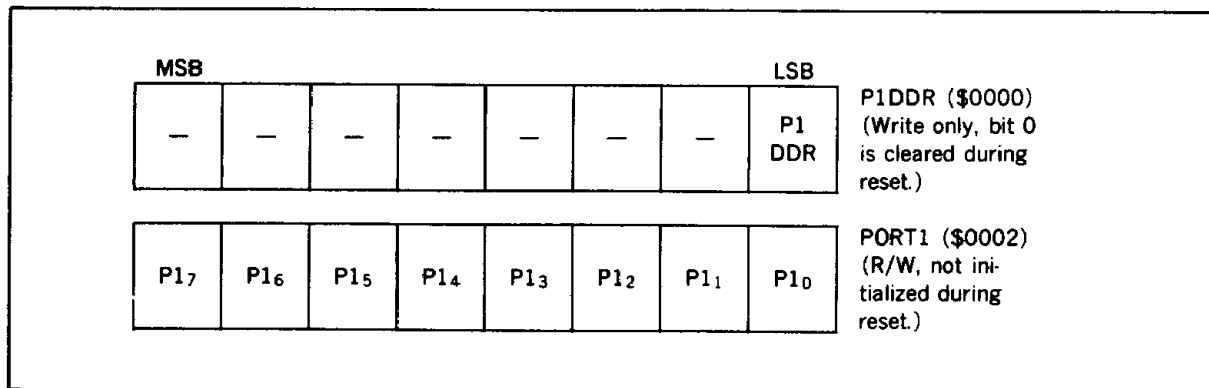


Figure 10. Port 1 Register and Data Direction Register

**Port 2** : Port 2 is an 8-bit I/O port (figure 11). Each bit of the DDR (\$0001) defines the data direction of the corresponding bit of port 2 (figure 12). Port 2 can drive one TTL load and 30 pF capacitance. It can produce 1 mA when  $V_{out} = 1.5 V$

to directly drive the base of a Darlington transistor.

Port 2 pins are also used as I/O pins by timers 1, 2, and the SCI (table 8). The pin functions are controlled by registers in timers 1, 2, and the SCI.

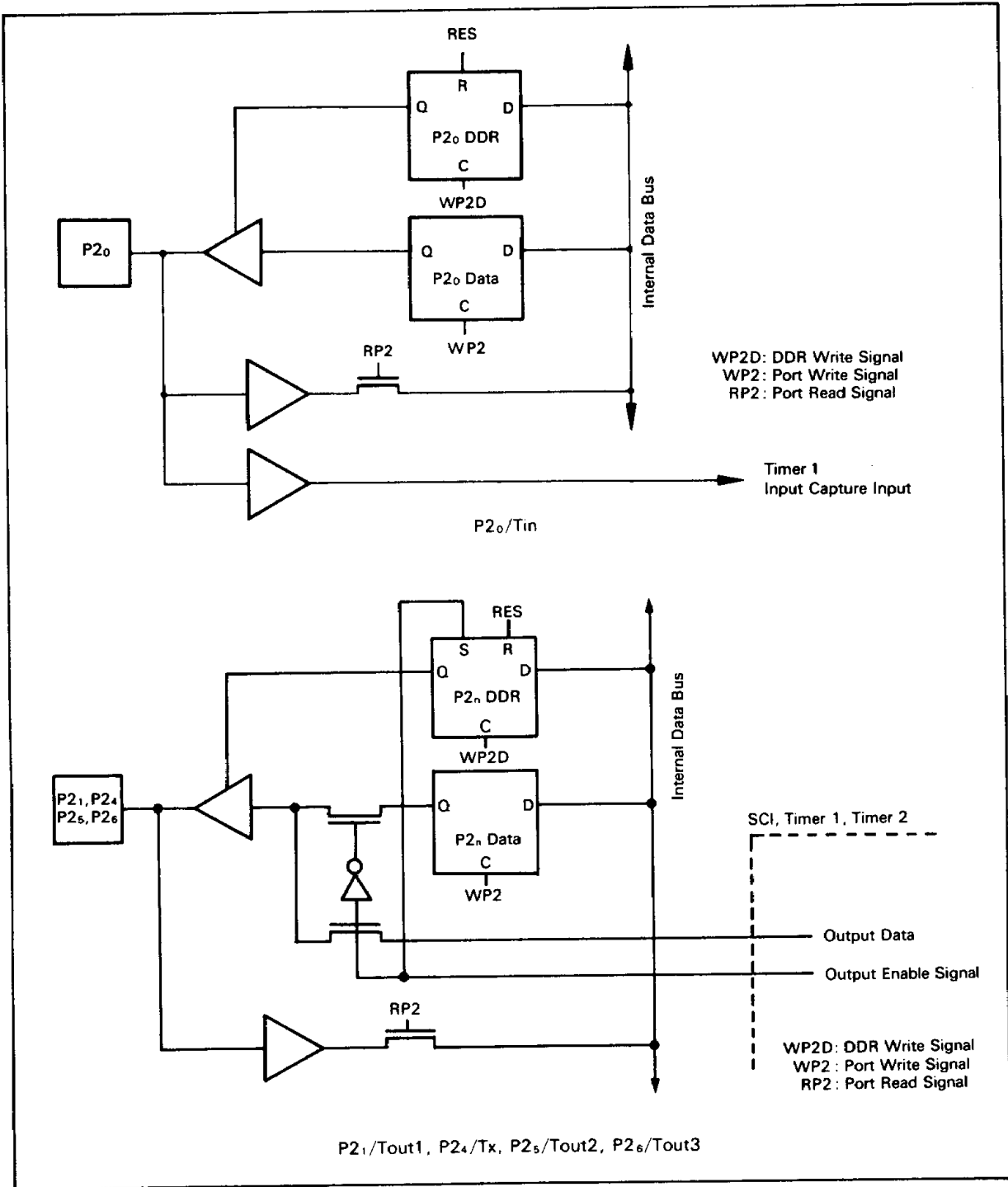
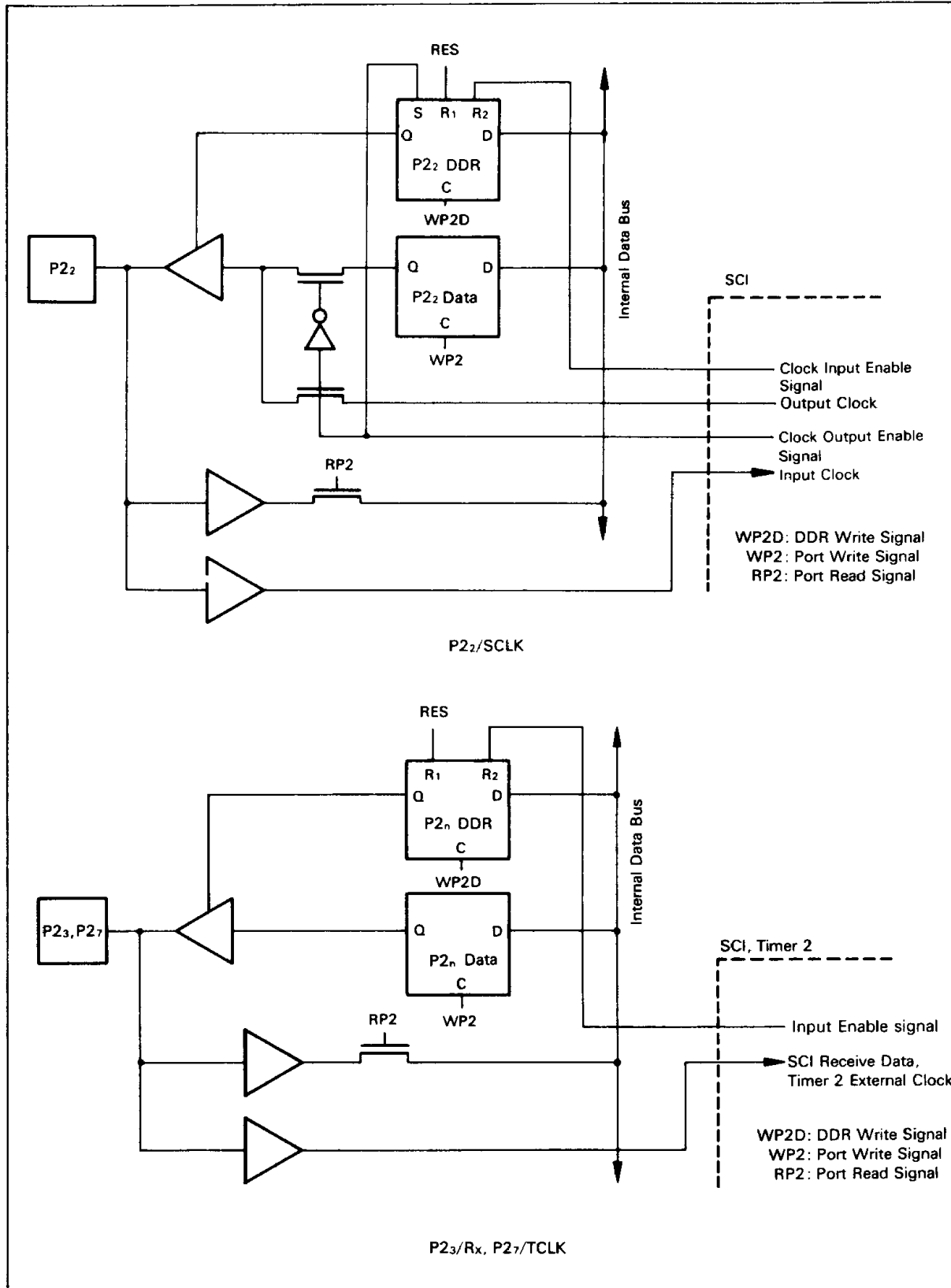


Figure 11. Port 2 Block Diagram

(continued)



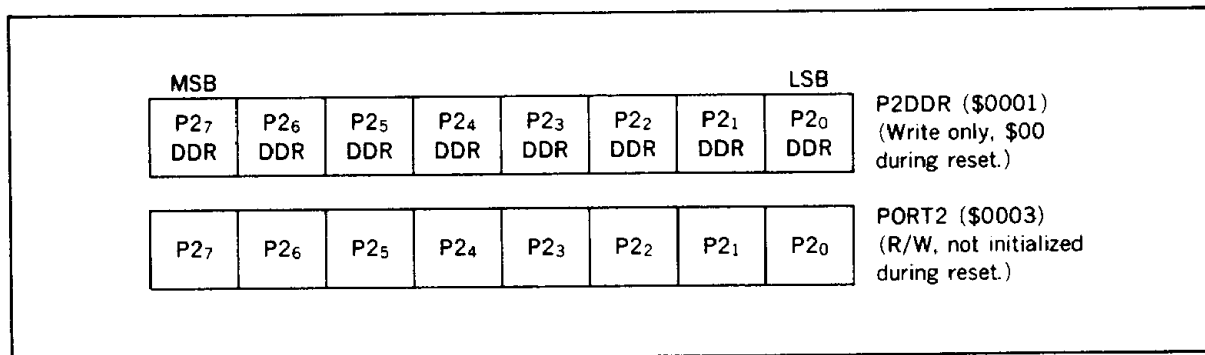


Figure 12. Port 2 Register and Data Direction Register

Table 8. Port 2 Pin Functions

Port 2 Pin	Alternate Function	Description
P2 <sub>0</sub>	Tin	Timer 1 input
P2 <sub>1</sub>	Tout1	Timer 1 output 1
P2 <sub>2</sub>	SCLK	SCI clock
P2 <sub>3</sub>	Rx	SCI receive input
P2 <sub>4</sub>	Tx	SCI transmit output
P2 <sub>5</sub>	Tout2	Timer 1 output 2
P2 <sub>6</sub>	Tout3	Timer 2 output 3
P2 <sub>7</sub>	TCLK	Timer 2 clock

# HD63701Y0, HD637A01Y0, HD637B01Y0

**Port 3** : Port 3 is an 8-bit I/O port (figure 13). The LSB of the DDR (\$0004) selects the data direction of the whole port (figure 14). In the expanded

modes (1 and 2) port 3 is the lower data bus (D<sub>7</sub>-D<sub>0</sub>). Port 3 can drive one TTL load and 90 pF capacitance.

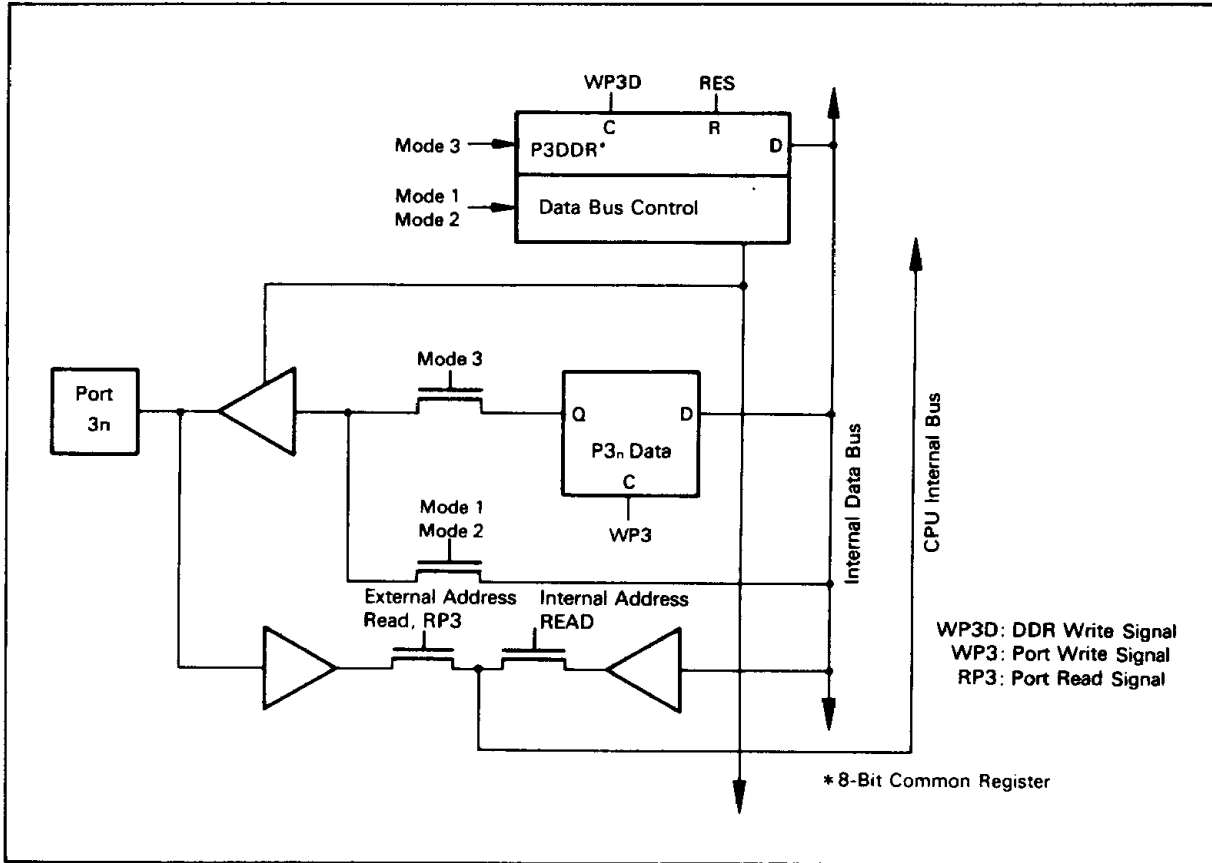


Figure 13. Port 3 Block Diagram

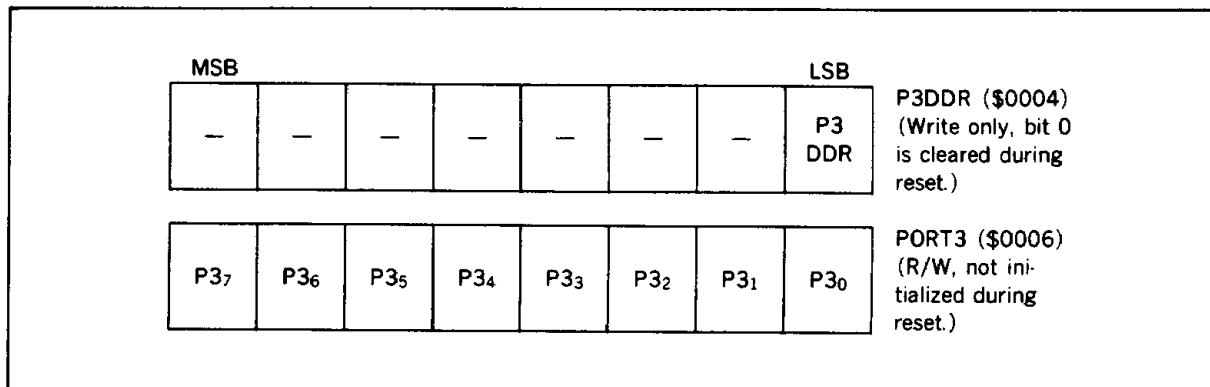


Figure 14. Port 3 Register and Data Direction Register



**Port 4** : Port 4 is an 8-bit I/O port (figure 15). Each bit of the DDR (\$0005) defines the data direction of the corresponding bit of port 4 (figure 16). In the expanded modes (1 and 2), port 4 is the upper address bus (A<sub>15</sub>-A<sub>8</sub>). In mode 1 (expanded mode with no external ROM), the DDR is set

automatically and port 4 outputs addresses. In mode 2 (expanded mode with external ROM), the DDR must be set to 1 for port 4 to function as the address bus. Pins that are not needed for the address bus can be used as input pins. Port 4 can drive one TTL load and 90 pF capacitance.

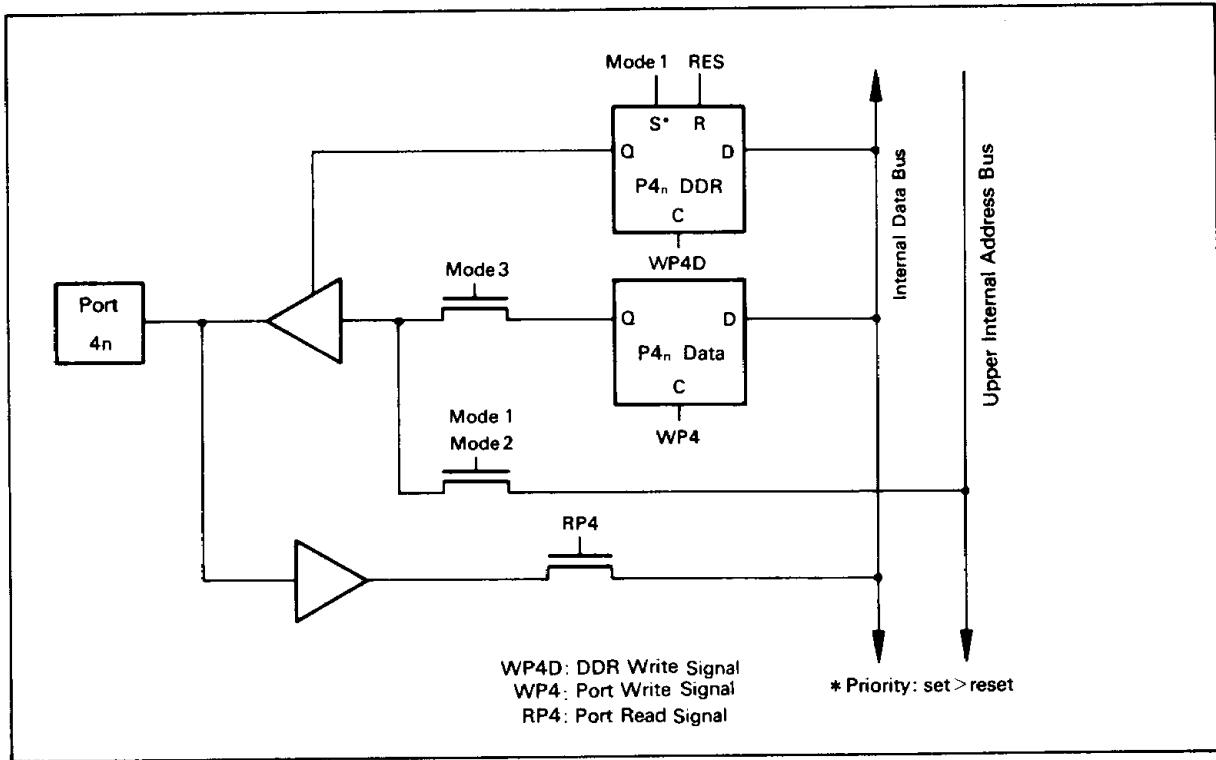


Figure 15. Port 4 Block Diagram

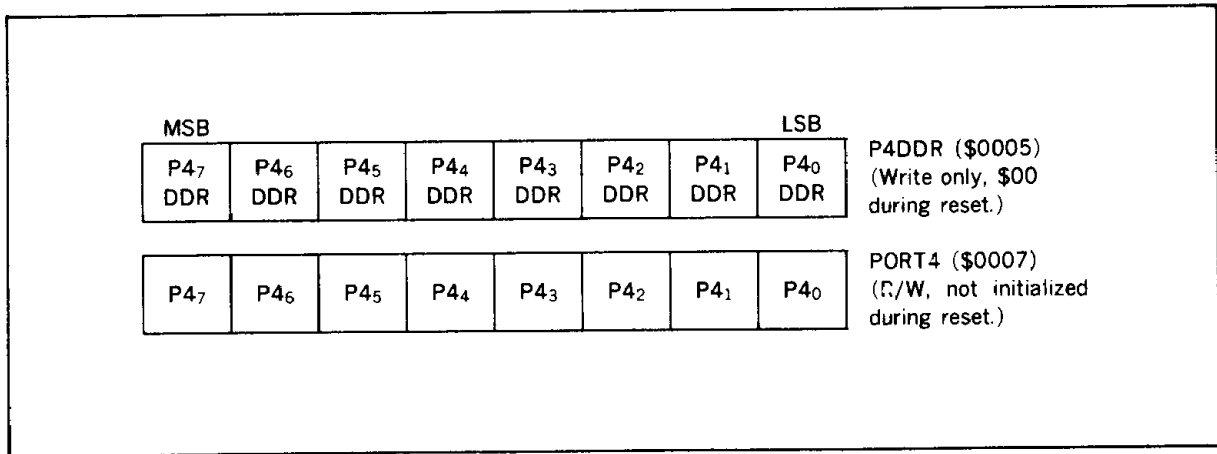


Figure 16. Port 4 Register and Data Direction Register



**Port 5** : Port 5 is an 8-bit I/O port (figure 17). Each bit of the DDR (\$0020) defines the data direction of the corresponding bit of port 5 (figure 18). Port 5 can drive one TTL load and 30 pF capacitance.

P5<sub>5</sub>-P5<sub>0</sub> are also used as control pins (table 9). The function of these pins is determined by the RAM/port 5 control register (RP5CR), except for P5<sub>4</sub>/ $\overline{IS}$  and P5<sub>5</sub>/ $\overline{OS}$ , which are controlled by the port 6 control/status register (P6CSR).

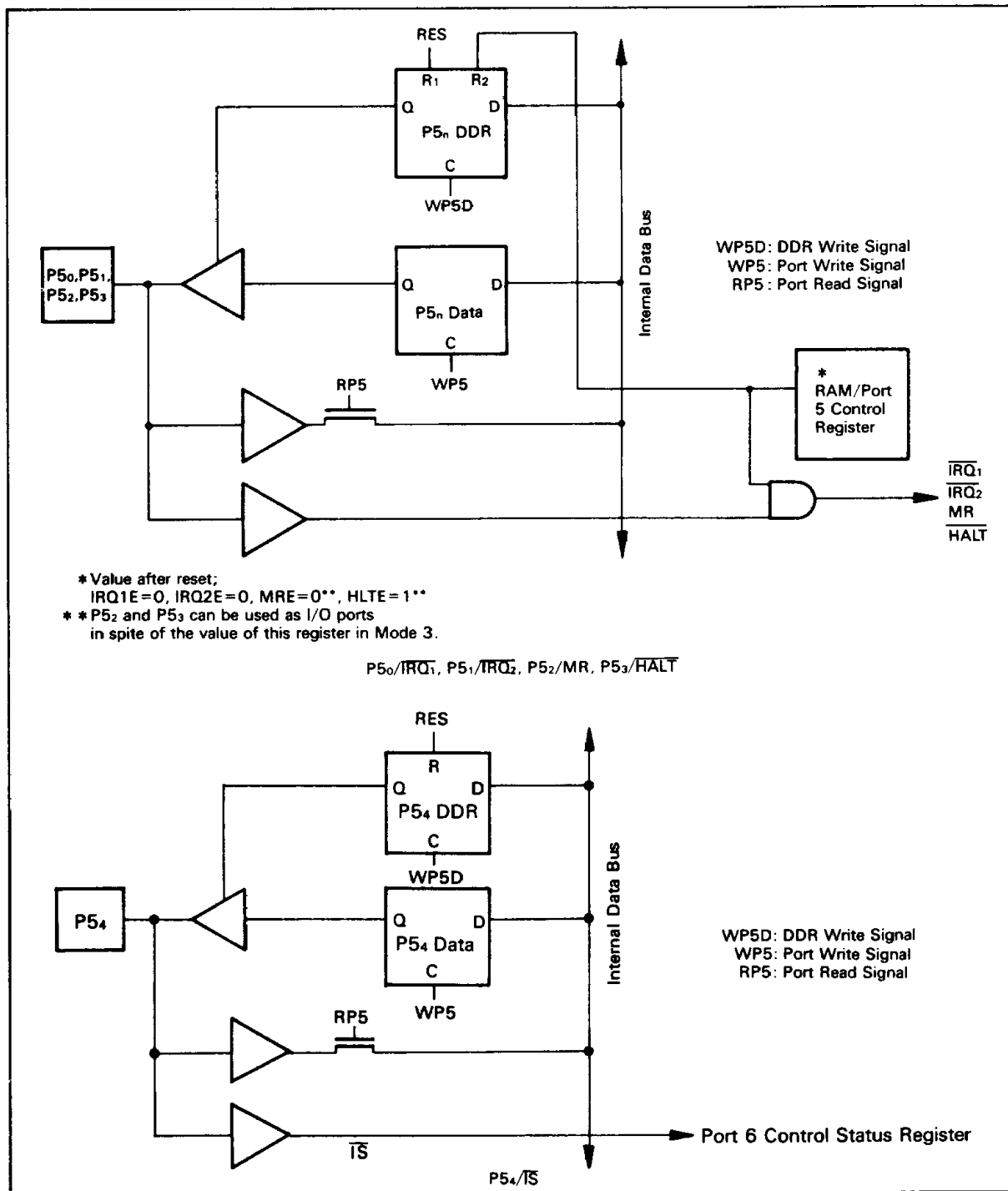


Figure 17. Port 5 Block Diagram

(continued)

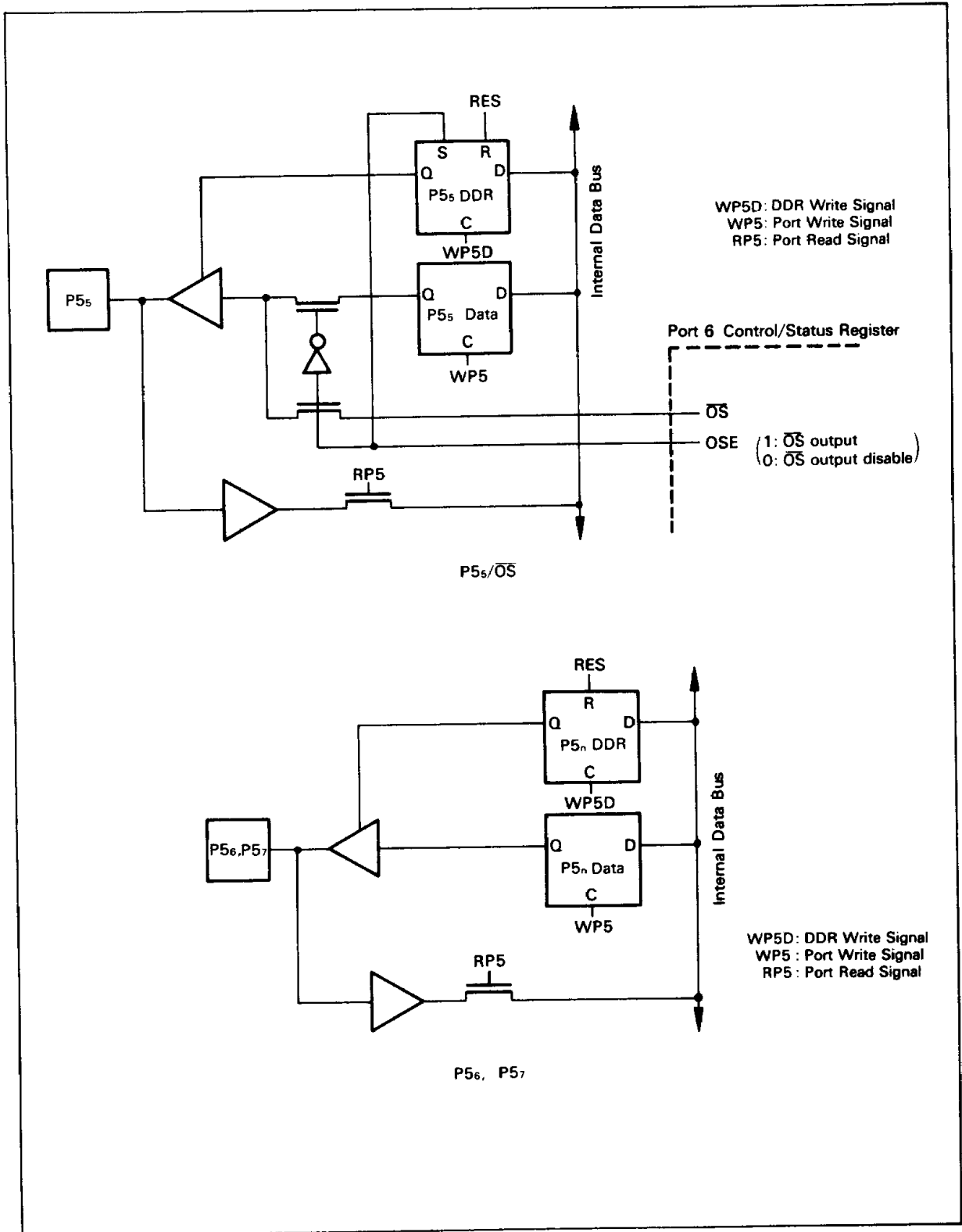


Figure 17. Port 5 Block Diagram (Cont)

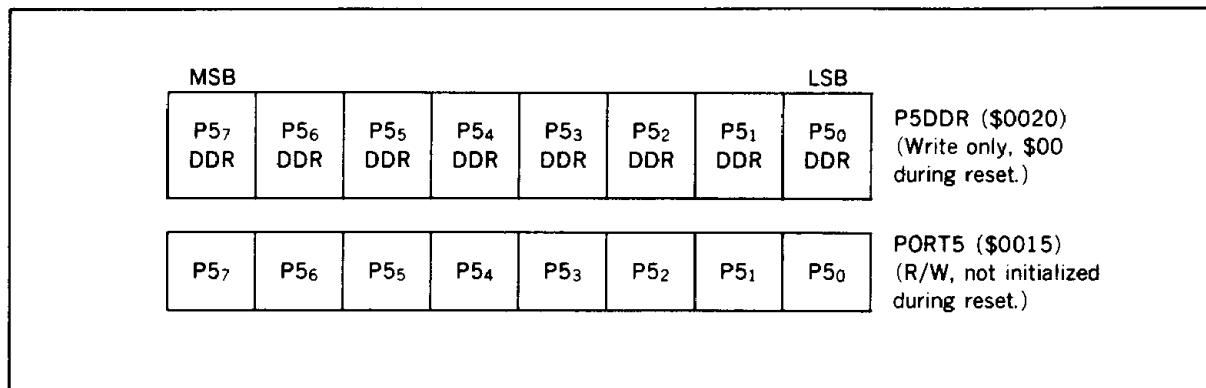


Figure 18. Port 5 Register and Data Direction Register

Table 9. Port 5 Pin Functions

Port 5 Pin	Alternate Function	Description
P5 <sub>0</sub>	$\overline{IRQ_1}$	Interrupt input 1
P5 <sub>1</sub>	$\overline{IRQ_2}$	Interrupt input 2
P5 <sub>2</sub>	MR	Memory ready input
P5 <sub>3</sub>	$\overline{HALT}$	Halt input
P5 <sub>4</sub>	$\overline{IS}$	Input strobe
P5 <sub>5</sub>	$\overline{OS}$	Output strobe

**RAM/Port 5 Control Register** : The RAM/port 5 control register (RP5CR: \$0014) controls onchip RAM and port 5 (figure 19).

- **IRQ<sub>1</sub>E, IRQ<sub>2</sub>E** : Setting **IRQ<sub>1</sub>E** and **IRQ<sub>2</sub>E** to 1 selects P5<sub>0</sub> and P5<sub>1</sub> as the  $\overline{\text{IRQ}}_1$  and  $\overline{\text{IRQ}}_2$  interrupt inputs. These bits are cleared at reset.
- **MRE, AMRE** : When **MRE** or **AMRE** is set to 1, P5<sub>2</sub> becomes the MR input. When both are 0, memory ready is inhibited (table 10). In mode 3, memory ready is always inhibited, regardless of these bits. **MRE** is cleared at reset, **AMRE** is set to 1.
- **HLTE** : When **HLTE** is set to 1, P5<sub>3</sub> becomes the  $\overline{\text{HALT}}$  input. When 0,  $\overline{\text{HALT}}$  is inhibited. In mode 3,  $\overline{\text{HALT}}$  is always inhibited, regardless of **HLTE**. This bit is set to 1 at reset.
- **STBY FLAG** : Clearing **STBY FLAG** by software puts the MCU into standby mode. This flag is set to 1 at reset, so reset cancels the standby mode. If the **STBY** pin is low, this flag cannot be cleared.
- **RAME** : When **RAME** is set to 1, on-chip RAM is enabled. When 0, it is disabled. **RAME** is set to 1 at reset. This bit should be set to 0 before going into standby state to protect on-chip RAM data.
- **STBY PWR** : When  $V_{CC}$  is not provided in standby mode, **STBY PWR** is cleared. If **STBY PWR** is set before the MCU goes to standby, and remains set after standby,  $V_{CC}$  was continuously supplied, and the contents of on-chip RAM are valid.

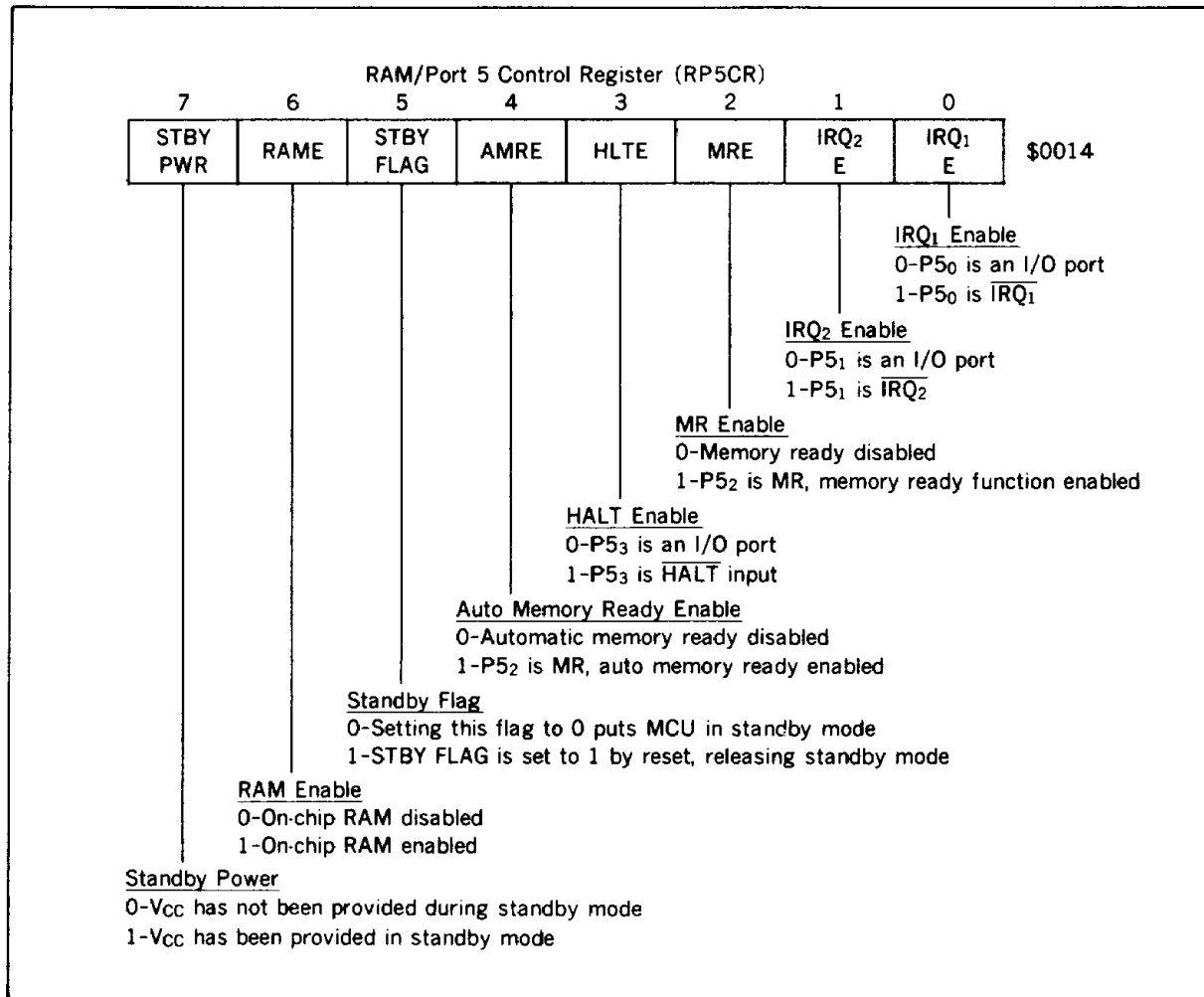


Figure 19. RAM/Port 5 Control Register

Table 10. Memory Ready Function

MRE	AMRE	Function
0	0	Memory ready inhibited.
0	1	Auto memory ready. When the CPU accesses the external address regardless of MR, E clock automatically stays high one-cycle longer. This state is retained during reset.
1	0	Memory ready. MR pin controls E clock high time.
1	1	When the CPU accesses the external address space with the P5 <sub>2</sub> (MR) pin low the auto memory ready operates. This function useful if there is both high-speed memory and slow memory outside. Input $\overline{\text{CS}}$ signal of slow memory to MR pin.

**Port 6** : Port 6 is an 8-bit I/O port (figure 20). Each bit of the DDR (\$0016) defines the data direction of the corresponding bit of port 6 (figure 21). Port 5 can drive one TTL load and 30 pF capacitance. In addition, it can drive the base of Darlington transistors directly.

Port 6 can function as a parallel handshake interface under the control of the port 6 control/status register (P6CSR : \$0021). Port 6 has a data latch for input data (IS LATCH).

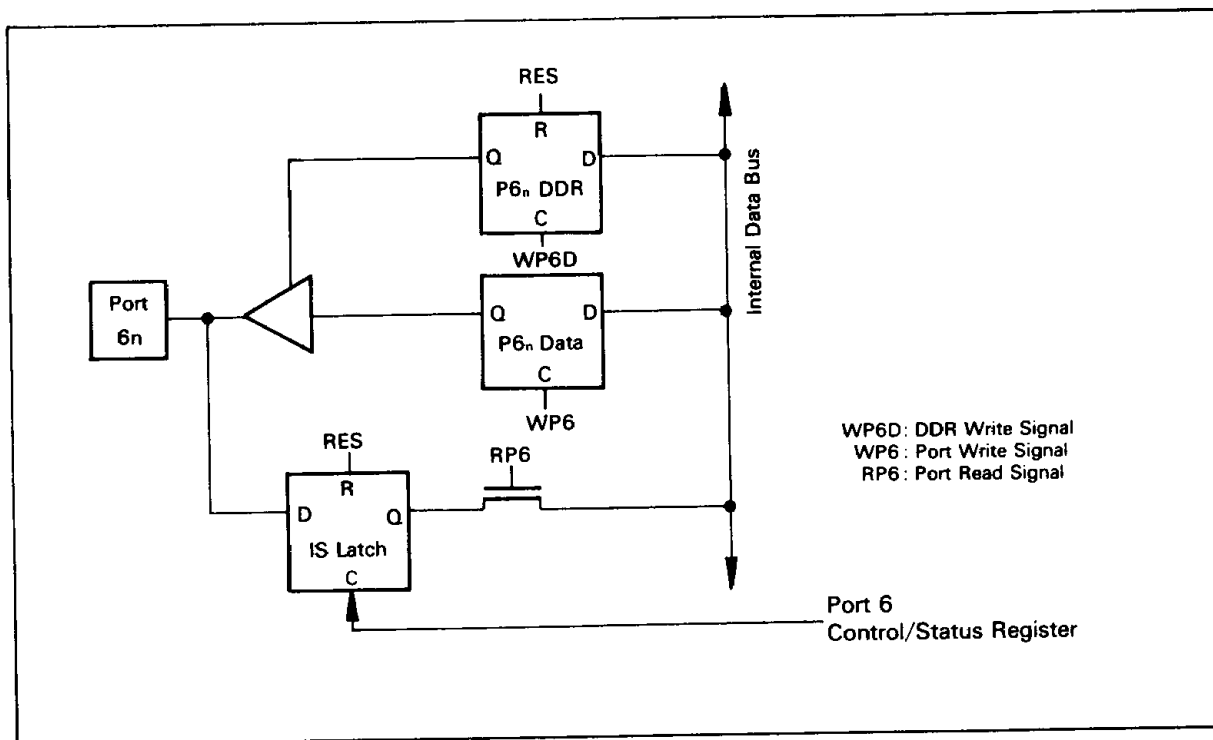


Figure 20. Port 6 Block Diagram

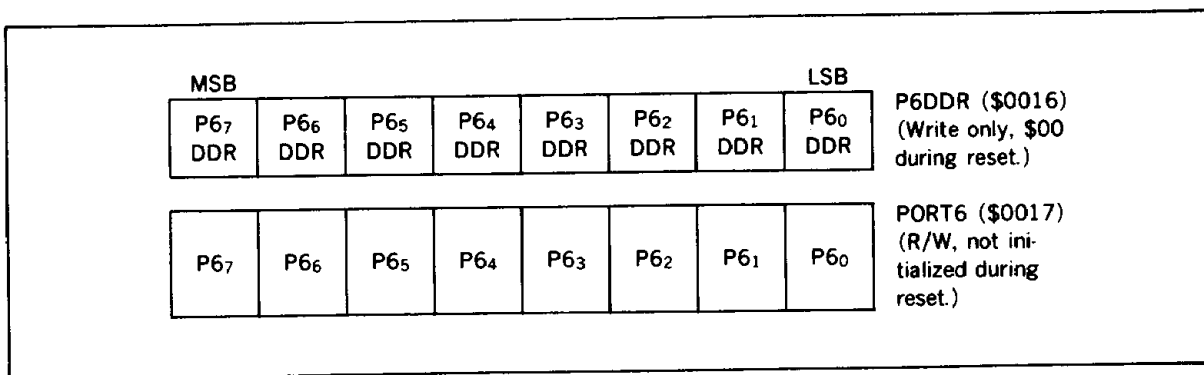


Figure 21. Port 6 Register and Data Direction Register

**Port 6 Control/Status Register** : The port 6 control/status register (P6CSR: \$0021) controls and holds the status of the port 6 handshake interface (figure 22). The handshake interface functions as follows.

1. Latches the data input at port 6 on the falling edge of  $\overline{IS}$  ( $P5_4$ ).
2. Outputs  $\overline{OS}$  ( $P5_5$ ) when reading or writing to port 6.
3. When IS FLAG is set by the falling edge of IS, an interrupt occurs (figure 23).

● **LATCH ENABLE** : The LATCH ENABLE bit controls the port 6 input latch (IS LATCH). When it is set, the input data at port 6 will be latched in at the falling edge of  $\overline{IS}$  ( $P5_4$ ). Reading port 6 clears the latch. If LATCH ENABLE is 0, the input latch is disabled, and  $P5_4$  acts as an ordinary I/O port. LATCH ENABLE is cleared at reset.

● **OSS** : When OSS is set, writing to port 6 initiates an output strobe signal ( $\overline{OS}/P5_5$ ). When OSS is cleared, reading port 6 initiates an  $\overline{OS}$ . OSS is cleared at reset.

● **OSE** : When OSE is set,  $P5_5$  is the output strobe.  $\overline{OS}$ . When cleared, it is a normal I/O port.

● **IS IRQ<sub>1</sub> ENABLE** : When IS IRQ<sub>1</sub> ENABLE is set, IS FLAG set causes an  $\overline{IRQ}_1$  interrupt. When cleared, IS FLAG does not cause an interrupt. This bit is cleared at reset.

● **IS FLAG** : The IS FLAG is set by the falling edge of  $\overline{IS}$ . It is a read-only flag. It is cleared by reading or writing to port 6 after reading the P6CSR. IS FLAG is cleared during reset.

Table 11 shows the conditions that set and reset the port 6 control/status register flags.

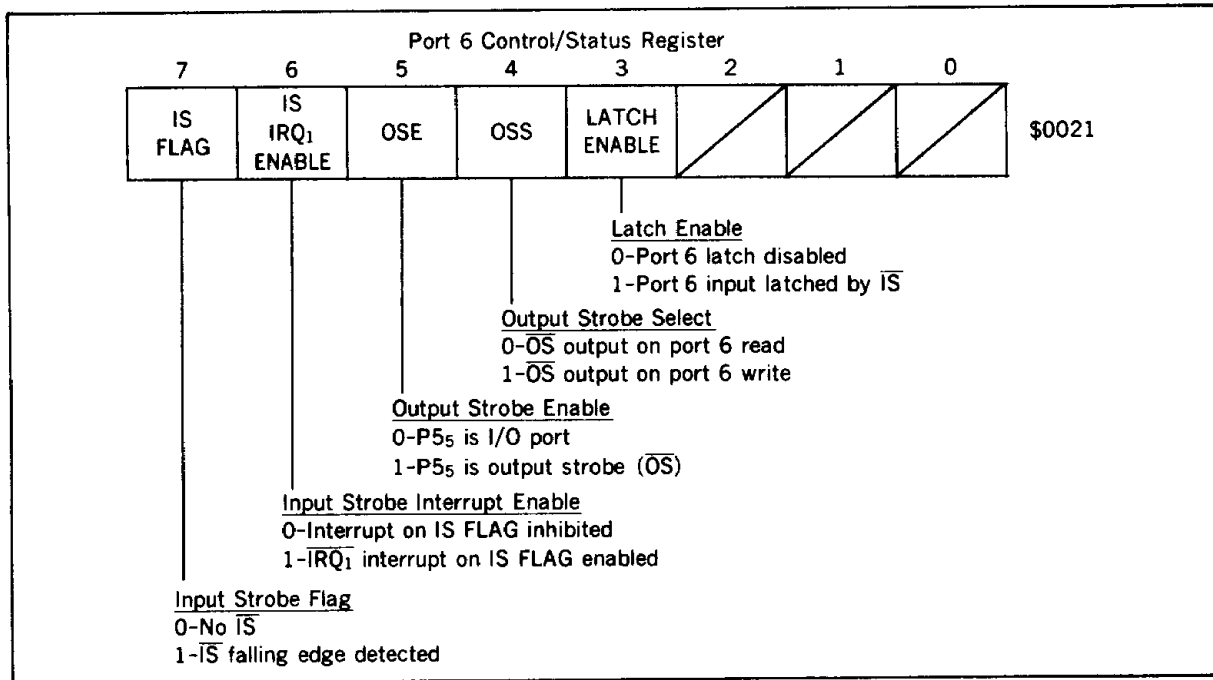


Figure 22. Port 6 Control/Status Register

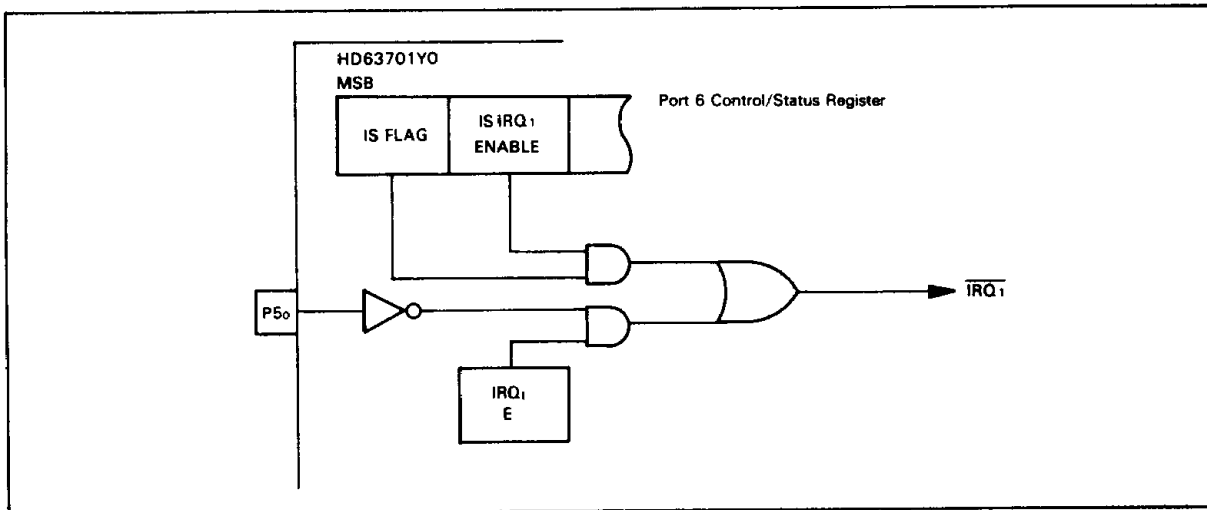


Figure 23. Input Strobe Interrupt Block Diagram

Table 11. Port 6 Control Status Register Status Flags Set and Reset Conditions

Flag	Set Condition	Clear Condition
IS FLAG	Falling edge input to P5 <sub>4</sub> ( $\overline{IS}$ )	<ul style="list-style-type: none"> <li>Read the P6CSR then read or write the port 6, when IS FLAG=1</li> <li><math>\overline{RES}=0</math></li> </ul>
ICF	FRC→ICR by rising or falling edge input to P2 <sub>0</sub> . (Selected by IEDG)	<ul style="list-style-type: none"> <li>Read the TCSR1 or TCSR2 then ICRH, when ICF=1</li> <li><math>\overline{RES}=0</math></li> </ul>



# HD63701Y0, HD637A01Y0, HD637B01Y0

**Port 7** : Port 7 is a 5-bit output only port (figures 24, 25). In the expanded modes (1 and 2), port 7 outputs control signals from the CPU. Port 7 goes

to high-impedance state during reset. Port 7 can drive one TTL load and 30 pF capacitance.

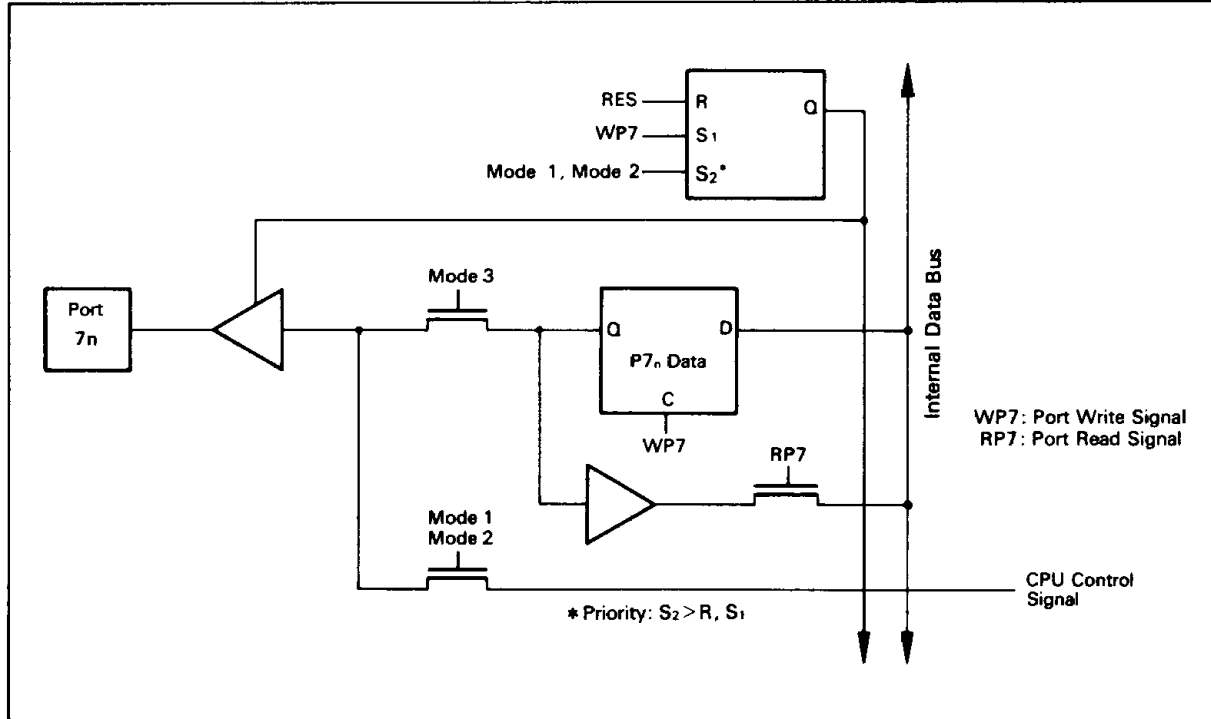


Figure 24. Port 7 Block Diagram

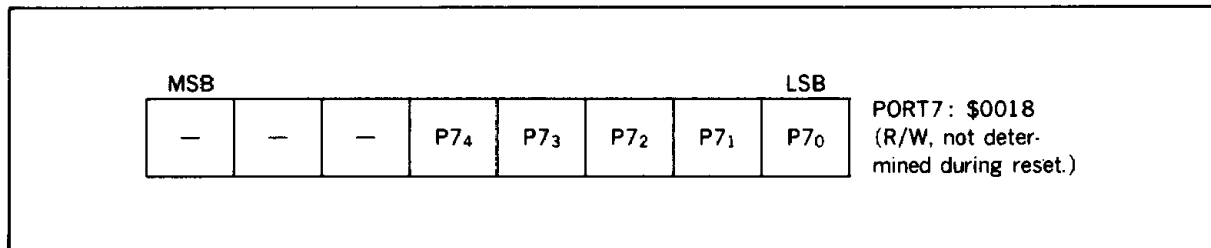


Figure 25. Port 7 Register

**16-Bit Programmable Timer (Timer 1)**

Timer 1 can simultaneously measure an input waveform and generate two independent output waveforms. The pulse widths of both input and outputs can vary from microseconds to seconds.

Timer 1 (figure 26) is configured as follows:

**Registers :**

- Control/status registers 1 and 2
- Free-running counter
- Output compare registers 1 and 2
- Input capture register

**Timer I/O pins :**

- Tin (P2<sub>0</sub>), input
- Tout1 (P2<sub>1</sub>), output
- Tout2 (P2<sub>2</sub>), output

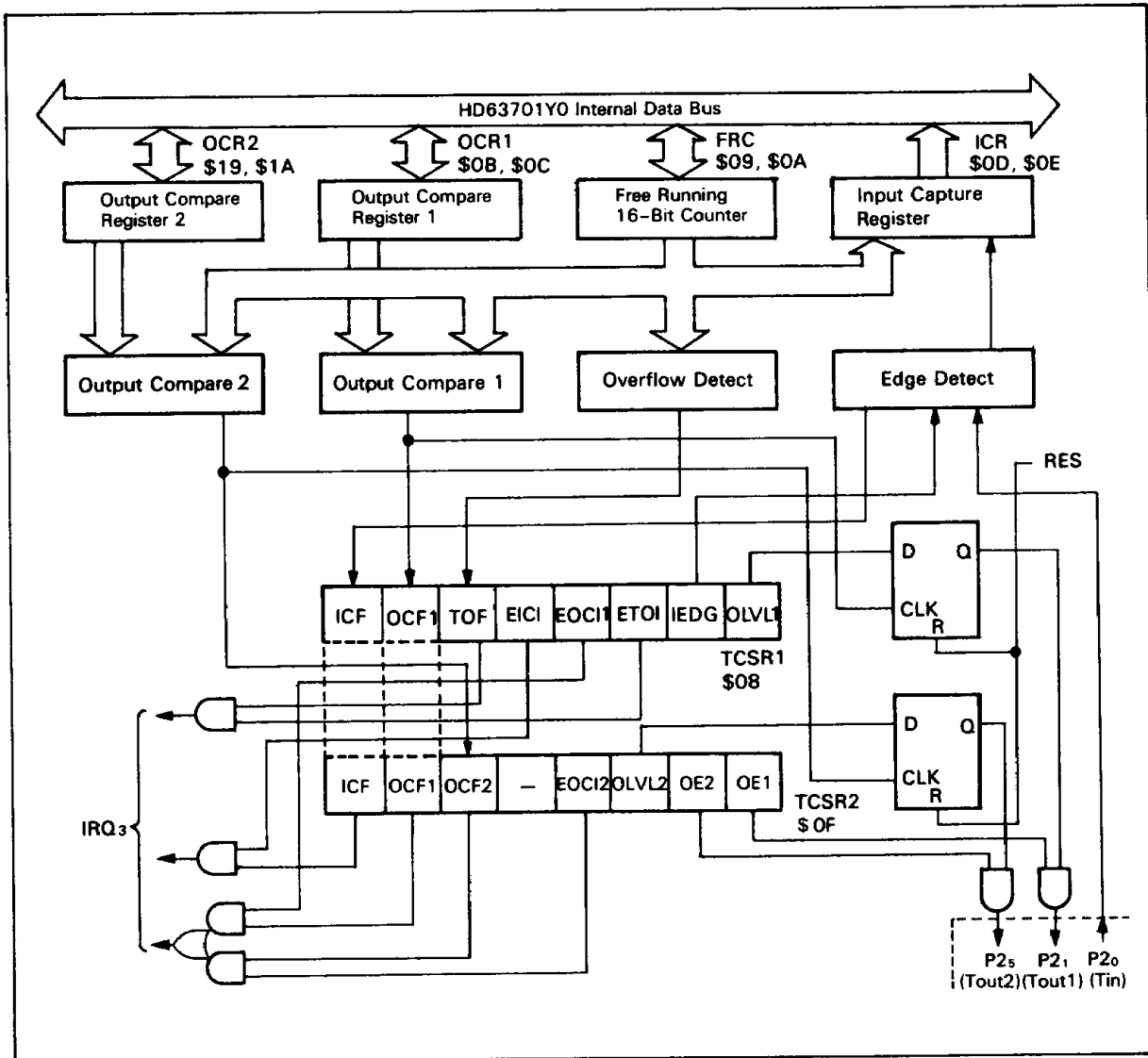


Figure 26. Timer 1 Block Diagram

**Free-Running Counter :** The free-running counter (FRC: \$0009, \$000A) is a 16-bit free-running counter incremented by the system clock. Its value can be read without affecting its operation. The FRC is cleared at reset.

**Output Compare Registers :** The output compare registers (OCR1: \$000B, \$000C, OCR2: \$0019, \$001A) are 16-bit read/write registers that control the output waveforms. The data in the FRC is always being compared to the OCRs. When the data matches, an output compare flag (OCF) is set in the corresponding timer control/status register (TCSR).

**Input Capture Register :** The input capture register (ICR: \$000D, \$000E) is a 16-bit read-only register which stores the FRC's value when a P2<sub>0</sub> transition causes an input capture pulse. IEDG of TCSR1 determines which transition causes the input capture. ICR is cleared at reset.

**Timer Control/Status Register 1 :** The timer control/status register 1 (TCSR1: \$0008; figure 27) is an eight bit register. All bits can be read and the lower five can be written. The upper three are read-only status bits. TCSR1 is cleared at reset.

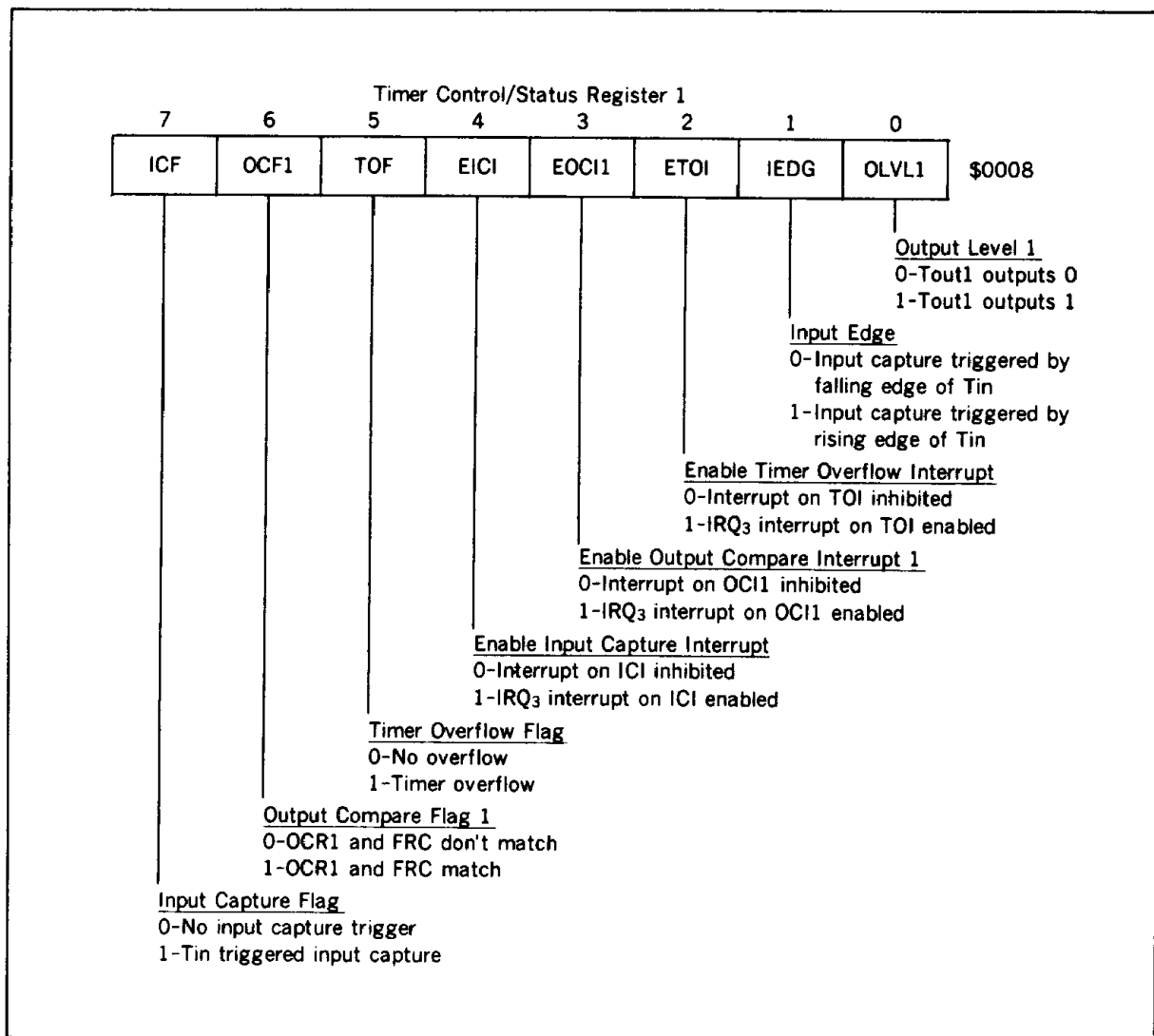


Figure 27. Timer Control/Status Register 1

- **OLVL1**: If OE1 of TCSR2 is set, the value of OLVL1 will appear at Tout1 (P2<sub>1</sub>) when a match occurs between the FRC and OCR1.
- **IEDG**: IEDG determines which edge of Tin/P2<sub>0</sub> will trigger data transfer from the counter to the ICR. IEDG=0 selects the falling edge (high to low transition), IEDG=1 selects the rising edge (low to high transition).
- **ETOI**: When ETOI is set to 1, timer overflow will cause internal interrupt IRQ<sub>3</sub>. When it is cleared, the interrupt is inhibited.
- **EOCI1**: When EOCI1 is set to 1, a counter match with OCI1 will cause an IRQ<sub>3</sub> interrupt. When it is cleared, the interrupt is inhibited.
- **EICI**: When EICI is set to 1, an input capture signal (Tin) will cause an interrupt IRQ<sub>3</sub>. When it is cleared, the interrupt is inhibited.
- **TOF**: The read-only flag TOF is set to 1 when the counter increments from \$FFFF to \$0000. It is

cleared when the CPU reads the FRC's upper byte (\$0009) after it reads the TCSR1 with TOF=1.

- **OCF1**: The read-only flag OCF1 is set to 1 when a match occurs between OCR1 and the FRC. It is cleared when the CPU writes to the OCR1 (\$000B or \$000C) after it reads the TCSR1 or TCSR2 with OCF1 or OCF2=1. OCF1 is also available at TCSR2, bit 6.
- **ICF**: The read-only flag ICF1 is set to 1 when Tin/P2<sub>0</sub> makes the transition defined by IEDG and the FRC is transferred to the ICR. It is cleared when the CPU reads the upper byte or the ICR after it reads the TCSR1 or TCSR2 with ICF=1. ICF is also available at TCSR2, bit 7.

**Timer Control/Status Register 2**: The timer control/status register 2 (TCSR2: \$000F; figure 28) is a seven bit register. All bits can be read and the lower four can be written. The upper three are read-only status bits. All bits are cleared at reset, except bit 4, which isn't used.

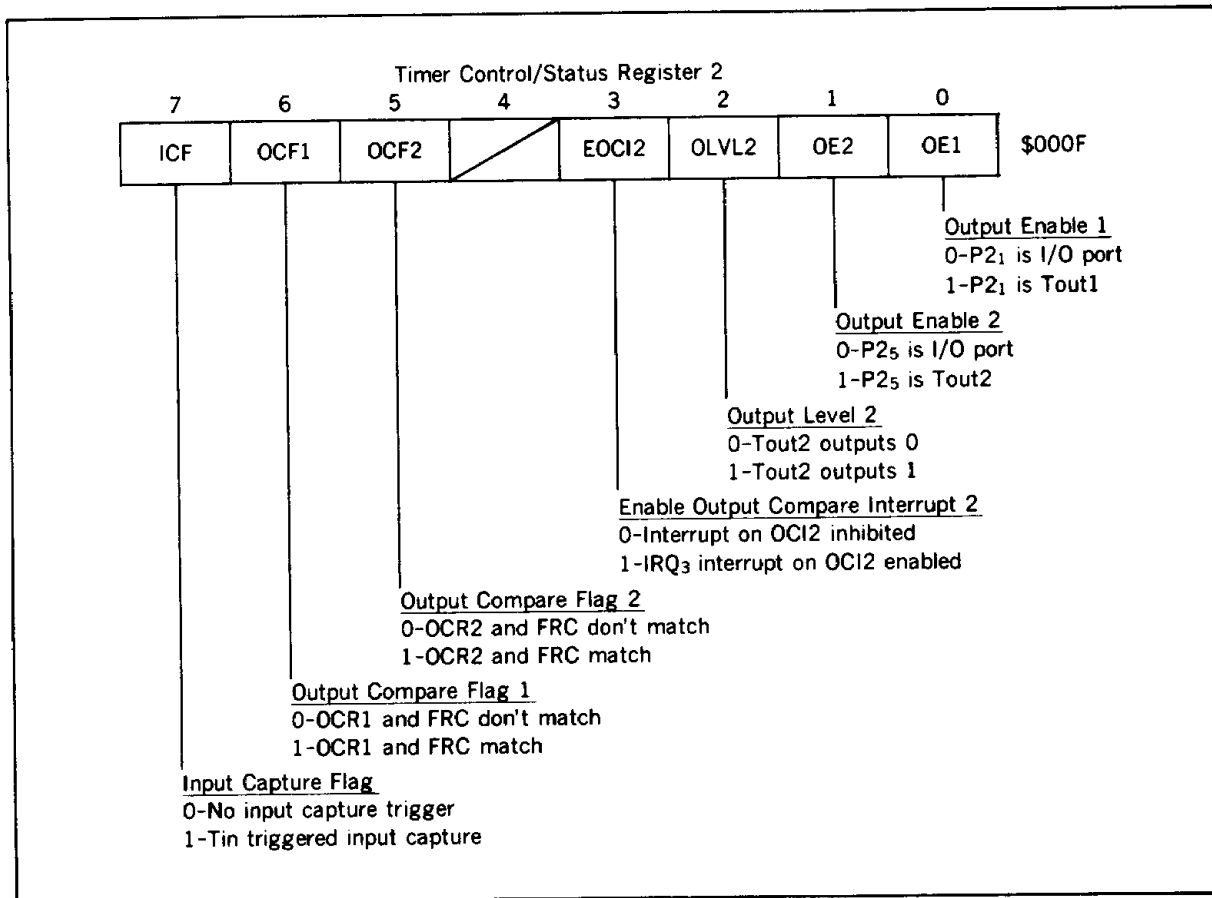


Figure 28. Timer Control/Status Register 2

- **OE1** : When OE1 is set, OLVL1 will be output to Tout1/P2<sub>1</sub> when there is a match between the FRC and OCR1. When OE1 is 0, P2<sub>1</sub> will be an I/O port.
- **OE2** : When OE2 is set, OLVL2 will be output to Tout2/P2<sub>5</sub> when there is a match between the FRC and OCR2. When OE1 is 0, P2<sub>5</sub> will be an I/O port.
- **OLVL2** : If OE2 of the TCSR2 is set, the value of OLVL2 will appear at Tout2 (P2<sub>5</sub>) when a match occurs between the FRC and OCR2.
- **EOCI2** : When EOCI2 is set, a counter match with OCI2 will cause an interrupt IRQ<sub>3</sub>. When it is cleared, the interrupt is inhibited.
- **OCF2** : The read-only flag OCF2 is set when a match occurs between OCR2 and the FRC. It is cleared when the CPU writes to OCR2 (\$0019 or \$001A) after it reads the TCSR2 with OCF2=1.
- **OCF1, ICF** : OCF1, ICF in the TCSR<sub>2</sub> are the

same as in the TCSR1. They can be addressed at either register.

Table 12 shows the conditions that set and reset the timer 1 flags.

**8-Bit Reloadable Timer (Timer 2)**

In addition to timer 1, the HD63701Y0 has an 8-bit reloadable timer, which can count external events. Timer 2 has one output, so together with timer 1, the HD63701Y0 can output three independent waveforms.

Timer 2 (figure 29) is configured as follows :

Registers :

- Control/status register (7 bits)
- Upcounter (8 bits)
- Time constant register (8 bits)

Timer I/O pins :

- Tout3 (P2<sub>6</sub>), output
- TCLK (P2<sub>7</sub>), input

**Table 12. Timer 1 Status Flags Set and Reset Conditions**

<b>Flag</b>	<b>Set Condition</b>	<b>Clear Condition</b>
OCF1	OCR1 = FRC	<ul style="list-style-type: none"> <li>• Read the TCSR1 or TCSR2 then write to the OCR1H or OCR1L, when OCF1 = 1</li> <li>• <math>\overline{RES} = 0</math></li> </ul>
OCF2	OCR2 = FRC	<ul style="list-style-type: none"> <li>• Read the TCSR2 then write to the OCR2H or OCR2L, when OCF2 = 1</li> <li>• <math>\overline{RES} = 0</math></li> </ul>
TOF	FRC = \$FFFF + 1 cycle	<ul style="list-style-type: none"> <li>• Read the TCSR1 then FRCH, when TOF = 1</li> <li>• <math>\overline{RES} = 0</math></li> </ul>

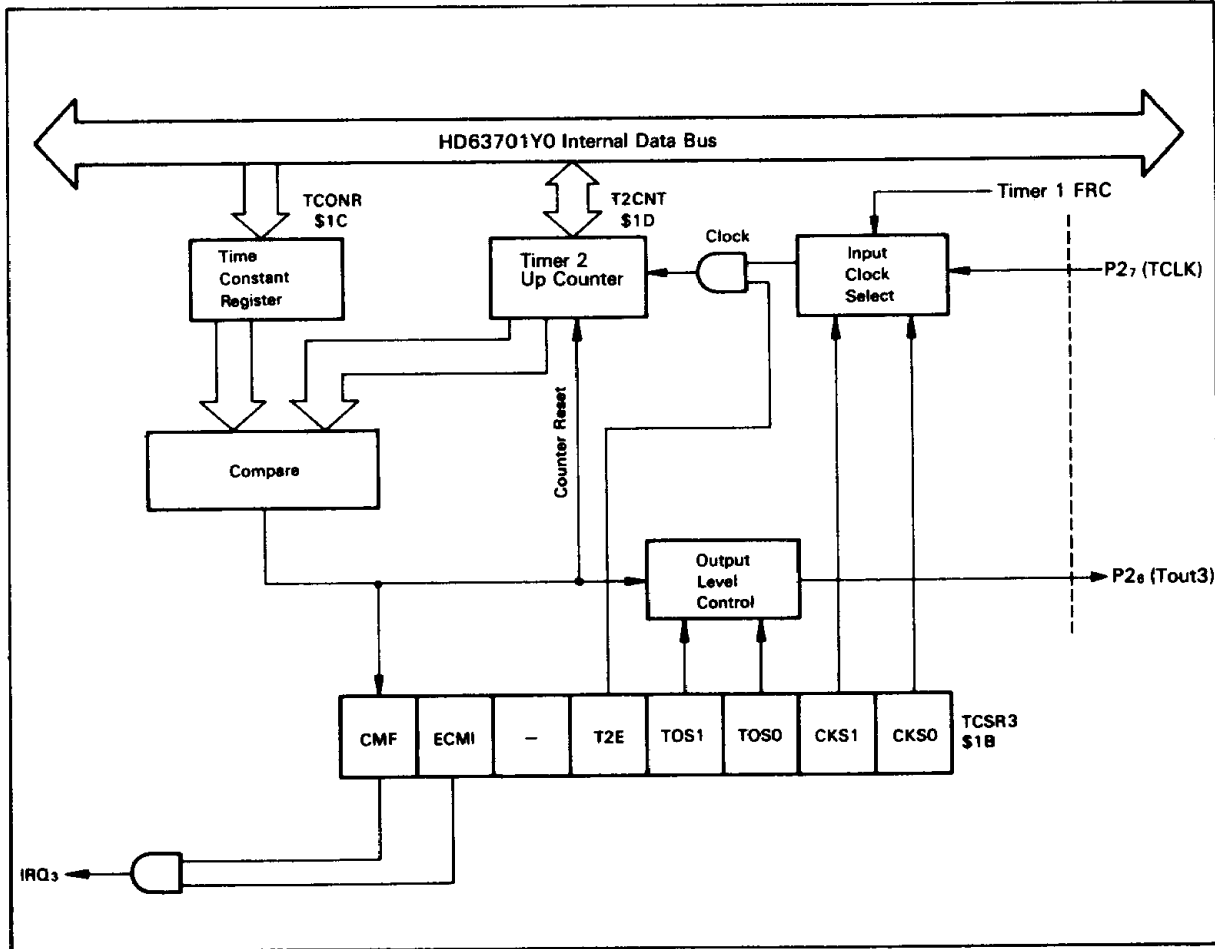


Figure 29. Timer 2 Block Diagram

**Timer 2 Upcounter :** The 8-bit upcounter (T2CNT: \$001D) counts the clock specified by CKS0 and CKS1 of the TCSR3. The CPU can read the value of the T2CNT at any time without affecting its operation. In addition, any value can be written to it at any time, even when counting.

It is cleared when a match occurs between the T2CNT and the TCONR, or at reset.

**Time Constant Register :** The 8-bit, write-only

time constant register (TCONR: \$001C) is constantly compared to the T2CNT. A match sets the counter match flag (CMF) of the timer control/status register 3 (TCSR3). It is set to \$FF at reset.

**Timer Control/Status Register 3 :** Timer control/status register 3 (TCSR3: \$001B; figure 30) is a 7-bit register. All bits can be read and all bits except for bit 7, CMF, can be written to. CMF can only be cleared, not set. The TCSR3 is cleared at reset.

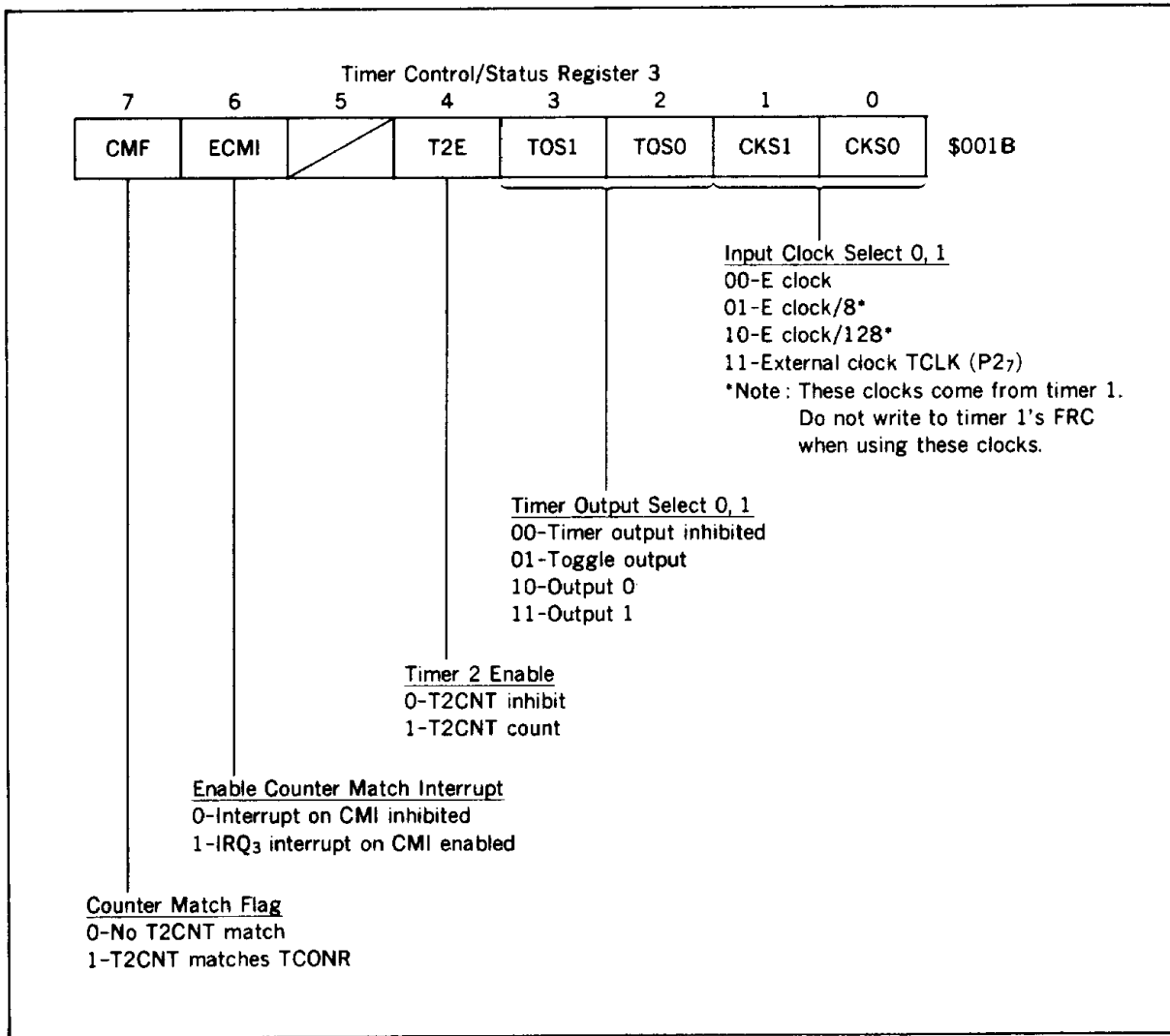


Figure 30. Timer Control/Status Register 3

- **CKS0, CKS1**: CKS0 and CKS1 select timer 2's input clock as shown in figure 30. When an external clock is selected, P2<sub>7</sub> will be the clock input automatically.
- **TOS0, TOS1**: TOS0 and TOS1 select the output that occurs on a counter match as shown in figure 30.
- **T2E**: When T2E is set, a clock is input to the T2CNT. When it is 0, the clock is inhibited.
- **ECMI**: When ECMI is set, a match between the T2CNT and the TCONR will cause an interrupt IRQ<sub>3</sub>. When it is cleared, the interrupt is inhibited.
- **CMF**: CMF is set by a match between the T2CNT and the TCONR. It is cleared by writing a 0 to it when it is set to 1. You cannot write a 1 to CMF.

Table 13 shows the conditions that set and reset the timer 2 flags.

### Serial Communication Interface

The serial communications interface (SCI) operates in two modes: asynchronous with NRZ encoding, and synchronous.

The SCI (figure 31) is configured as follows:

Registers:

- Transmit/receive control/status registers 1 and 2 (TRCSR1, TRCSR2)
- Rate/mode control register (RMCR)
- Receive data register (RDR)
- Receive shift register
- Transmit data register (TDR)
- Transmit shift register

SCI I/O pins:

- SCLK (P2<sub>2</sub>), input/output
- R<sub>x</sub> (P2<sub>3</sub>), input
- T<sub>x</sub> (P2<sub>4</sub>), output

**Asynchronous Mode**: The asynchronous mode has eight transfer frame formats with 7 or 8 data bits, 1 or 2 stop bits and parity or no parity (figure 32).

Setting TE in the TRCSR1 enables transmission, making P2<sub>4</sub> the serial output Tx regardless of the direction set in port 2's DDR. To transmit data, set the desired format in the RMCR and the TRCSR2. When TE is set, transmission can begin after a preamble for internal synchronization, consisting of one frame with all 1s for the data. At this stage, if the TDR is empty (TDRE=1), consecutive 1s are output to indicate the idle state. If the TDR contains data, it is sent to the transmit shift register and transmitted.

During data transmission, the SCI transmits a 0 start bit first, then the 7-or 8-bit data, starting with bit 0. When PEN of the TRCSR2=1, it sends the even or odd parity bit as selected by EOP. Lastly, the SCI sends one or two stop bits of 1.

When the TDR is empty, hardware sets the TDRE flag. If the CPU doesn't respond to the flag before the next data transfer to the transmit shift register should take place, a 1 is sent instead of the 0 start bit, and continues to transmit 1s (marking) until the CPU puts data in the TDR.

Setting the RE bit of the TRCSR1 enables reception, making P2<sub>3</sub> the serial input R<sub>x</sub>. The TRCSR2 and the RMCR specify reception operation. The SCI uses the first 0 received as a start bit, and synchronizes the receive bit flow. Each following bit will be strobed in the middle.

If a stop bit is not 1, the SCI assumes a framing error, and sets ORFE. Then the received data is transferred to the RDR so the CPU can read it. This makes it possible to detect a line break. ORFE is cleared when the CPU reads the RDR after reading the TRCSR.

When PEN is set to 1, the SCI checks the parity bit. If the parity bit does not match the parity selected by EOP, the SCI set PER to indicate a parity error. The data can be read after a parity error, as with the framing error.

**Table 13. Timer 2 Status Flag Set and Reset Conditions**

Flag	Set Condition	Clear Condition
CMF	T2CNT = TCONR	<ul style="list-style-type: none"> <li>• Write 0 to CMF, when CMF = 1</li> <li>• RES = 0</li> </ul>



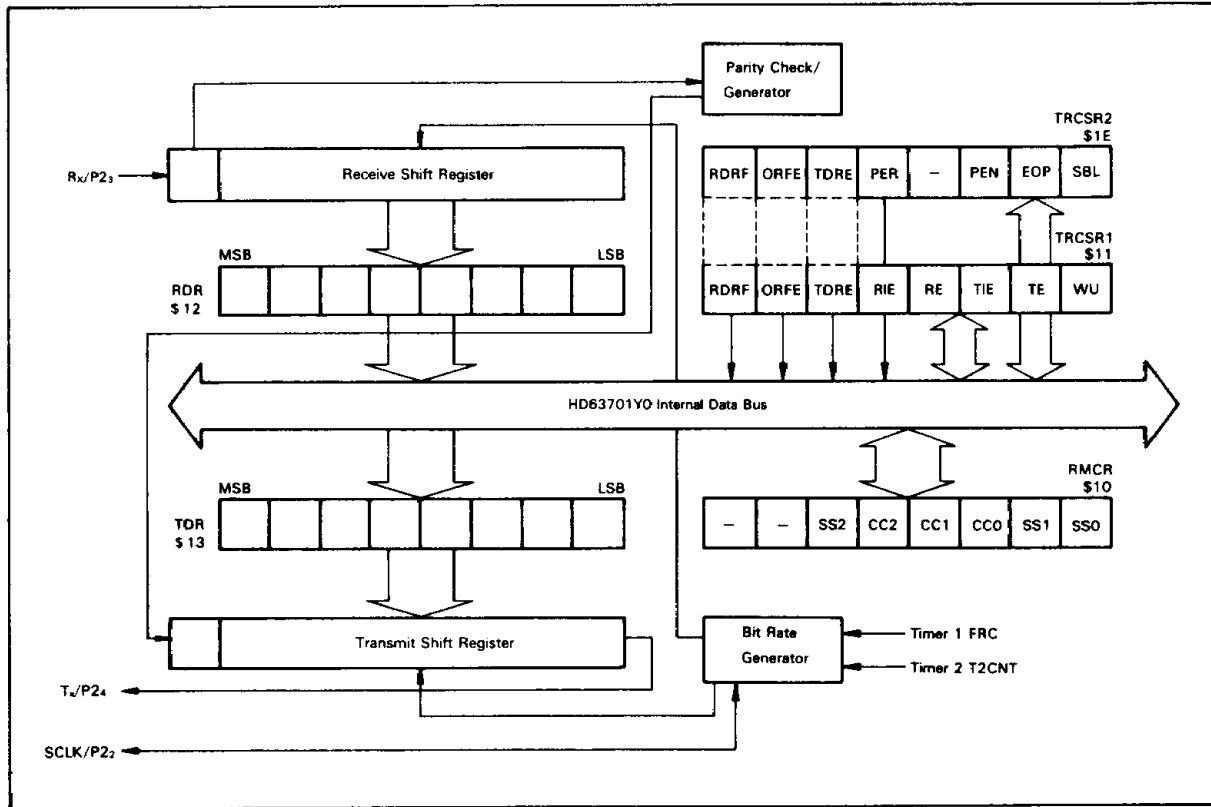


Figure 31. SCI Block Diagram

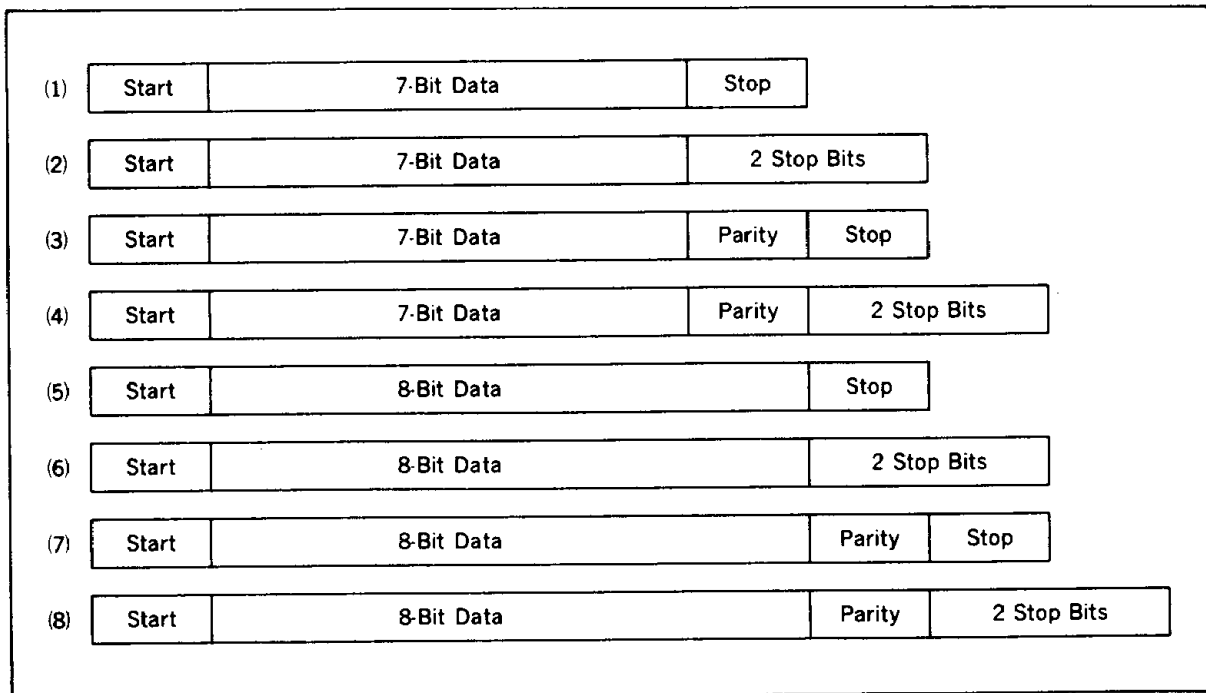


Figure 32. Asynchronous Formats

The SCI sets the RDRF flag if the frame is received without framing or parity errors. The CPU can get the received data by reading the RDR after reading the TRCSR, which clears RDRF. If RDRF is still set when the SCI is receiving the stop bit of the next frame, the SCI sets the ORFE flag to indicate an overflow error. In 7-bit data format, bit 8 is 0.

The clock source is selected by CC1 and CC0. If they are 10, the internal bit rate clock is output at P2<sub>2</sub>, regardless of TE or RE. If they are 11, an external TTL-compatible clock must be connected to P2<sub>2</sub> at 16 times (16×) the desired bit rate, but not greater than E.

**Synchronous Mode** : In the clocked synchronous mode, data is transmitted synchronously with a clock pulse. Since the HD63701Y0 has an independent transmitter and receiver, full duplex operation is available, but only in asynchronous mode. In synchronous mode, P2<sub>2</sub> is the only clock pin, so simultaneous transmission and reception is impossible. TE and RE should therefore never be set to 1 at the same time. Figure 33 shows the clock and data format in synchronous mode.

Setting TE in TRCSR1 enables transmission, making P2, the serial output Tx regardless of the direction set in port 2's DDR. To transmit data, set the desired format in RMCR and TRCSR2. When an external clock is selected and TDRE is 0, the SCI

transmits data from Tx, synchronized with 8 clock pulses input at SCLK. If clock output is selected, the SCI outputs transmit data and clock pulses.

The SCI transmits data starting with bit 0. TDRE is set when the transmit shift register is empty. After eight clock pulses, external clock pulses are ignored.

Setting the RE bit of the TRCSR1 enables reception, making P2<sub>3</sub> the serial input Rx. Reception operation is specified by the TRCSR2 and the RMCR.

If external clock input is selected, eight external clock pulses and synchronized data are input at Rx and SCLK. The SCI receives the data in the receive shift register by this clock, and sets the RDRF flag after it receives the eighth data bit. More than eight external clock pulses are ignored. When RDRF is cleared, the SCI starts receiving the next data immediately, so RDRF should be cleared when SCLK is high.

When clock output is selected, eight clocks are output from SCLK when RE is set. The external transmitter should synchronize to these clocks. Reception of the first byte of data sets RDRF. Clearing RDRF causes reception to continue by transmitting eight more clocks.

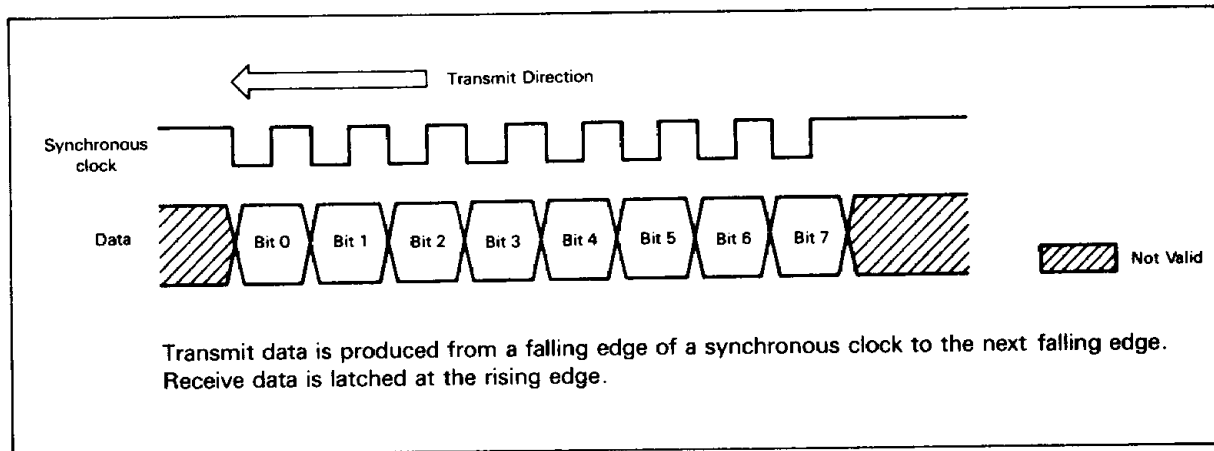


Figure 33. Synchronous Format

**Transmit/Receive Control/Status Register 1**  
 1: Transmit/receive control/status register 1  
 (TRCSR1: \$0011; figure 34) has eight readable

bits, four of which can also be written to. It is initialized to \$20 at reset.

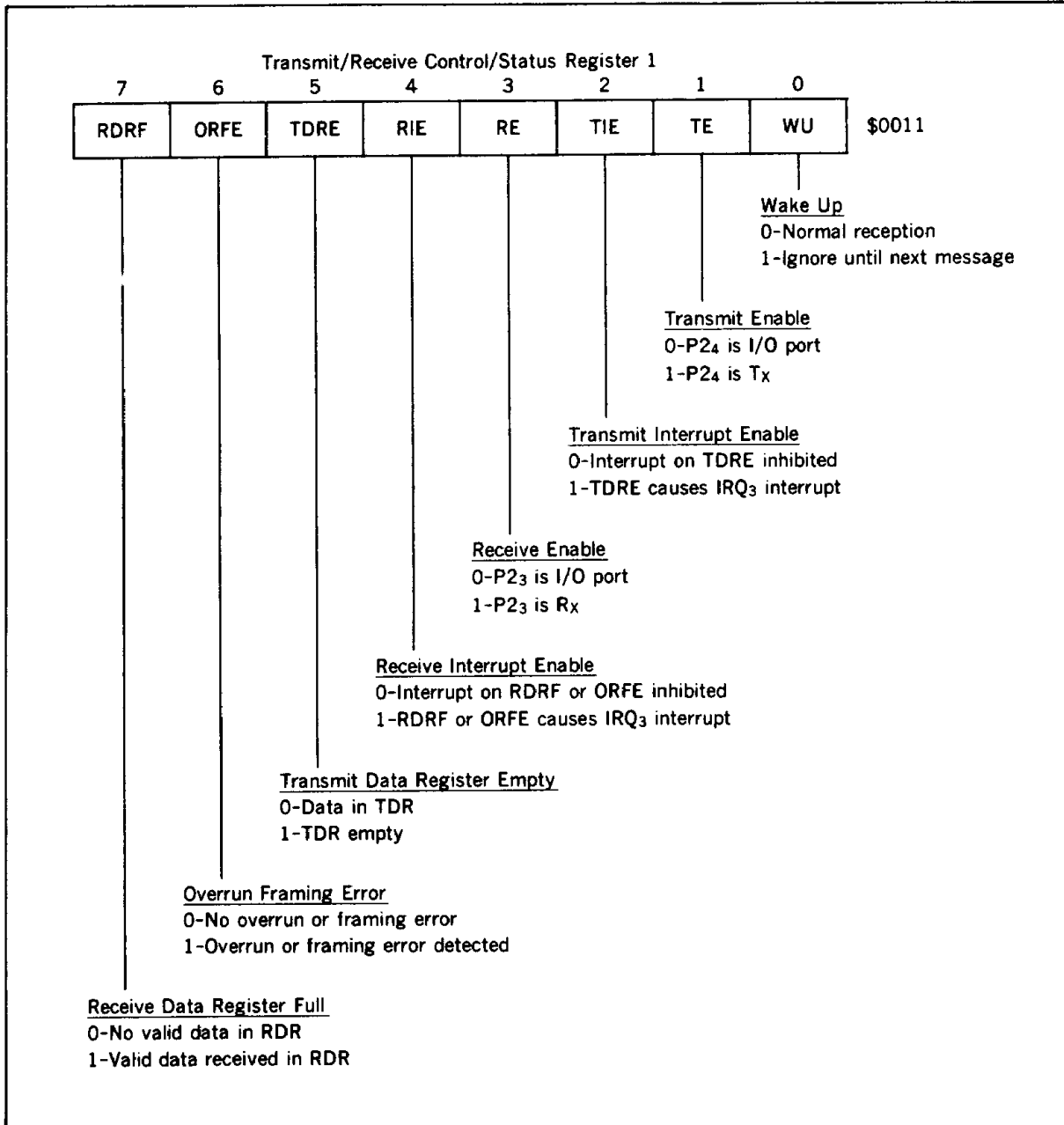


Figure 34. Transmit/Receive Control/Status Register 1

- **WU**: WU is the asynchronous mode wake-up function. When WU is set, the SCI stops receiving data until the next message. The reception of a frame of consecutive 1s wakes the SCI up, clears WU and starts reception. The RE flag should be set before setting WU. The wake-up function is not available in synchronous mode.
- **TE**: When TE is set, transmit data will appear at Tx/P2<sub>4</sub>. It is preceded by a one-frame preamble in asynchronous mode, in synchronous mode it appears immediately.
- **TIE**: When TIE is set, TDRE set will cause an IRQ<sub>3</sub> interrupt. When it is cleared, the interrupt is inhibited. TIE is cleared at reset.
- **RE**: When RE is set, a signal is input at Rx/P2<sub>3</sub>. When RE is cleared, P2<sub>2</sub> can be used as an I/O port.
- **RIE**: When RIE is set, RDRF or ORFE set will cause an IRQ<sub>3</sub> interrupt. When it is cleared, the interrupt is inhibited. RIE is cleared at reset.
- **TDRE**: The SCI sets TDRE when the TDR is transferred to the transmit shift register in the asynchronous mode, leaving the TDR empty. It is set in the synchronous mode when the transmit shift register is empty. TDRE is cleared by reading the TRCSR1 or the TRCSR2 and writing new data to the TDR while TDRE=1. TDRE is set to 1 at reset. The TDRE should be cleared in the transmit state after TE is set.
- **ORFE**: The SCI sets ORFE when an overrun or framing error occurs during data receive. ORFE is cleared by reading the TRCSR1 or the TRCSR2 and the RDR when ORFE=1. ORFE is cleared at reset.
- **RDRF**: The SCI sets RDRF when data is received normally and transferred from the receive shift register to the RDR. It is cleared by reading the TRCSR1 or the TRCSR2 and the RDR when RDRF=1. RDRF is cleared at reset.

**Transmit/Receive Control/Status Register 2** : Transmit/receive control/status register 2 (TRCSR2: \$001E; figure 35) has seven readable bits, the lower three of which can also be written to.

- **SBL** : When SBL is 0, 1 stop bit is selected for asynchronous mode. When SBL is 1, 2 stop bits are used. SBL is cleared at reset.
- **EOP** : EOP selects even or odd parity in asynchronous mode if PEN=1. If EOP is 1, the SCI uses odd parity. If it is 0, the SCI uses even parity.
- **PEN** : If PEN is 1, the SCI generates and checks

parity bits in the asynchronous mode. If it is 0, it uses no parity.

The above 3 bits (SBL, EOP, PEN) do not affect the SCI operation in the clocked synchronous mode.

- **PER** : PER is set when a parity error occurs. It is cleared by reading the RDR after reading the TRCSR2 with PER=1.
- **TDRE, ORFE, RDRF** : TDRE, ORFE, RDRF are the same as in the TRCSR1. These bits can be accessed at either address.

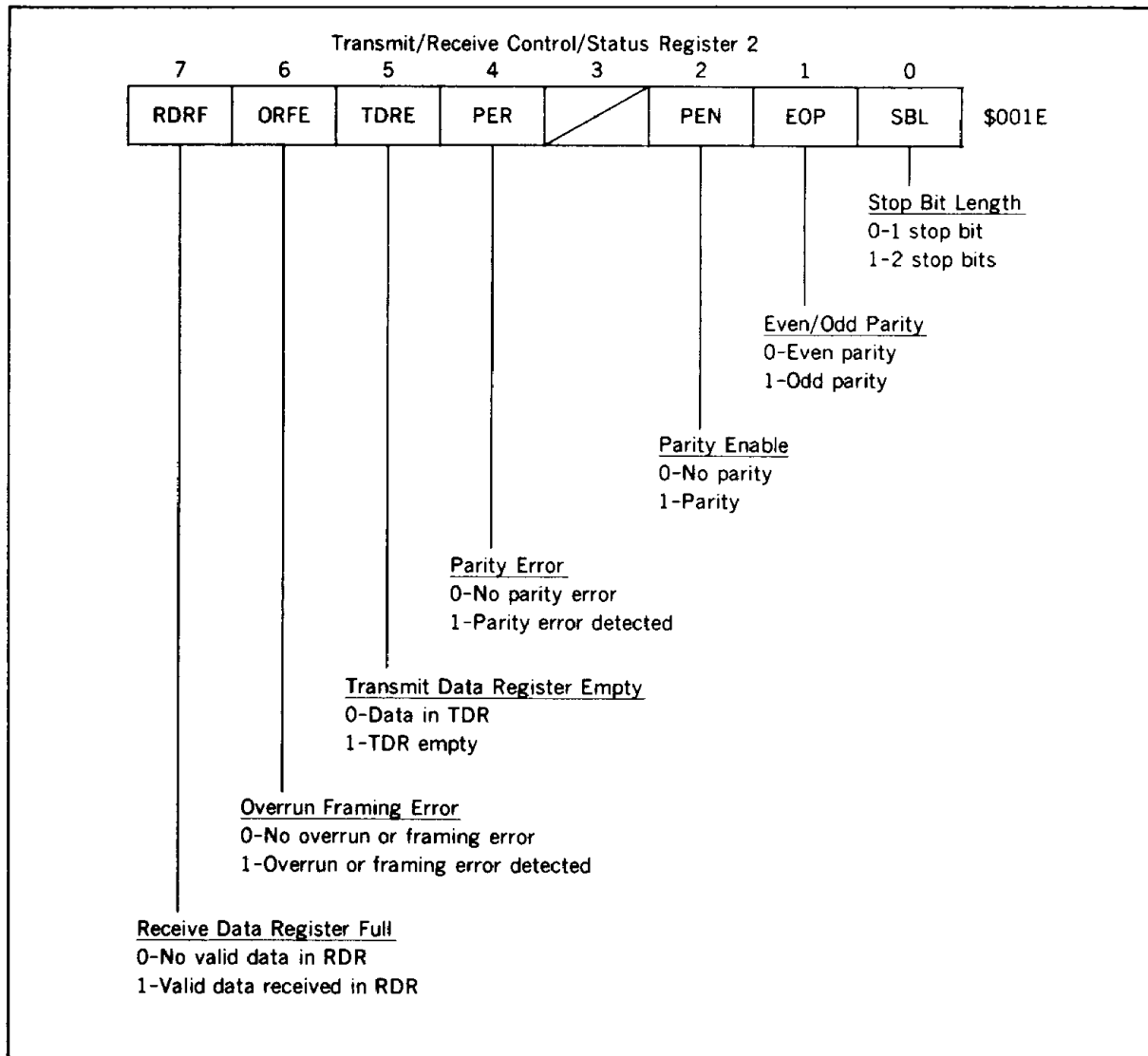


Figure 35. Transmit/Receive Control/Status Register 2

**Transfer Rate/Mode Control Register** : The 6-bit transfer rate/mode control register (RMCR: \$0010; figure 36) controls the following SCI functions.

- Baud rate
- Clock source
- Operation mode
- Data format
- P<sub>2</sub>/SCLK function

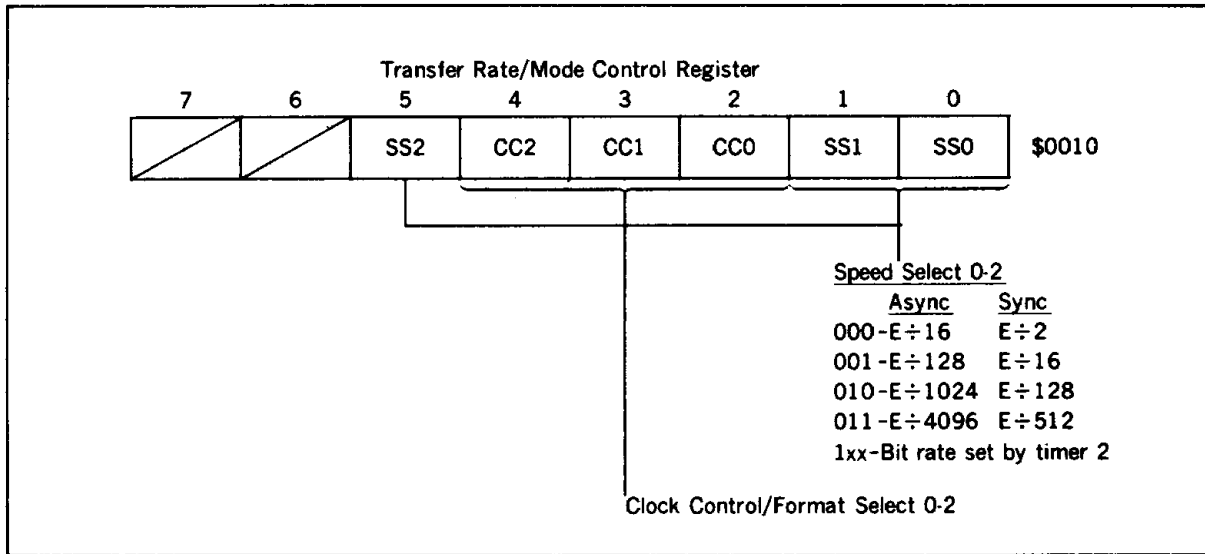
All bits are read/write. Bits 0-6 are cleared at reset.

- SS0-SS2: SS0-SS2 select the transfer rate as shown in tables 14 and 15.

- CC0-CC2: CC0-CC2 control the data format and clock source as shown in table 16. CC0-CC2 are cleared at reset, putting the SCI into clock synchronous mode with external clock. This makes P<sub>2</sub> into the serial clock input. To use P<sub>2</sub> as an I/O port after reset, set CC1 and CC0 to 0, 1.

When using the SCI with an internal clock, do not write to the timer/counter which is the clock source for the SCI.

Table 17 shows the conditions that set and reset the SCI flags.



**Figure 36. Transfer Rate/Mode Control Register**

# HD63701Y0, HD637A01Y0, HD637B01Y0

**Table 14. SCI Bit Times and Transfer Rates**

**(1) Asynchronous Mode**

SS2	SS1	SS0	XTAL	2.4576 MHz	4.0 MHz	4.9152 MHz
			E	614.4 kHz	1.0 MHz	1.2288 MHz
0	0	0	E ÷ 16	26 μs/38400 Baud	16 μs/62500 Baud	13 μs/76800 Baud
0	0	1	E ÷ 128	208 μs/4800 Baud	128 μs/781.25 Baud	104.2 μs/9600 Baud
0	1	0	E ÷ 1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1200 Baud
0	1	1	E ÷ 4096	6.67 ms/150 Baud	4.096 ms/244.1 Baud	3.333 ms/300 Baud
1	—	—	—	*	*	*

\*When SS2 is 1, timer 2 provides SCI clocks. The baud rate is shown as follows with the TCONR as N.

$$\text{Baud Rate} = \frac{f}{32(N+1)}$$

f: input clock frequency to the timer 2 counter  
N=0-255

**(2) Clocked Synchronous Mode\***

SS2	SS1	SS0	XTAL	4.0 MHz	6.0 MHz	8.0 MHz
			E	1.0 MHz	1.5 MHz	2.0 MHz
0	0	0	E ÷ 2	2 μs/bit	1.33 μs/bit	1 μs/bit
0	0	1	E ÷ 16	16 μs/bit	10.7 μs/bit	8 μs/bit
0	1	0	E ÷ 128	128 μs/bit	85.3 μs/bit	64 μs/bit
0	1	1	E ÷ 512	512 μs/bit	341 μs/bit	256 μs/bit
1	—	—	—	**	**	**

\*Bit rates in the case of internal clock operation. In the case of external clock operation, the external clock is operatable up to DC to 1/2 system clock.

\*\*The bit rate is shown as follows with the TCONR as N.

$$\text{Bit Rate}(\mu\text{s/bit}) = \frac{4(N+1)}{f}$$

f: input clock frequency to the timer 2 counter  
N=0-255

**Table 15. Baud Rate and TCONR Example**

Baud Rate(Baud)	XTAL				
	2.4576 MHz	3.6864 MHz	4.0 MHz	4.9152 MHz	8.0 MHz
110	21*	32*	35*	43*	70*
150	127	191	207	255	51*
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	—	7	12
9600	1	2	—	3	—
19200	0	—	—	1	—
38400	—	—	—	0	—

\*E/8 clock is input to the timer 2 up counter. E clock for all others.



**Table 16. SCI Format and Clock Source**

<b>CC2</b>	<b>CC1</b>	<b>CC0</b>	<b>Mode</b>	<b>Transmit Format</b>	<b>Clock Source</b>
0	0	0	Clock Sync	8-bit data	Ext
0	0	1	Async	8-bit data	Int
0	1	0	Async	8-bit data	Int
0	1	1	Async	8-bit data	Ext
1	0	0	Clock Sync	8-bit data	Int
1	0	1	Async	7-bit data	Int
1	1	0	Async	7-bit data	Int
1	1	1	Async	7-bit data	Ext

**Table 17. SCI Status Flags Set and Reset Conditions**

<b>Flag</b>	<b>Set Condition</b>	<b>Clear Condition</b>
RDRF	Receive shift register→RDR	<ul style="list-style-type: none"> <li>• Read the TRCSR1 or TRCSR2 then RDR, when RDRF=1</li> <li>• <math>\overline{RES}=0</math></li> </ul>
ORFE	<ul style="list-style-type: none"> <li>• Framing error (Asynchronous mode) Stop bit=0</li> <li>• Overrun error (Asynchronous mode) Receive shift register→RDR when RDRF=1</li> </ul>	<ul style="list-style-type: none"> <li>• Read the TRCSR1 or TRCSR2 then RDR, when ORFE=1</li> <li>• <math>\overline{RES}=0</math></li> </ul>
TDRE	<ul style="list-style-type: none"> <li>• Asynchronous mode TDR→Transmit shift register</li> <li>• Clocked synchronous mode Transmit shift register is "empty"</li> <li>• <math>\overline{RES}=0</math></li> </ul>	<ul style="list-style-type: none"> <li>• Read the TRCSR1 or TRCSR2 then write to the TDR, when TDRE=1</li> </ul> <p>Note: TDRE should be reset after the TE is set.</p>
PER	Parity when PEN=1	<ul style="list-style-type: none"> <li>• Read the TRCSR2 then RDR, when PER=1</li> <li>• <math>\overline{RES}=0</math></li> </ul>



**EPROM Operation**

The HD63701Y0's on-chip EPROM is programmed in the EPROM mode (figures 37 and 38). EPROM mode is set by bringing MP<sub>0</sub>, MP<sub>1</sub>, and STBY low. In PROM mode, the MCU doesn't operate. It can be

programmed like a standard 27256 EPROM using a standard EPROM programmer and a socket adapter. Table 18 lists recommended EPROM programmers and socket adapters.

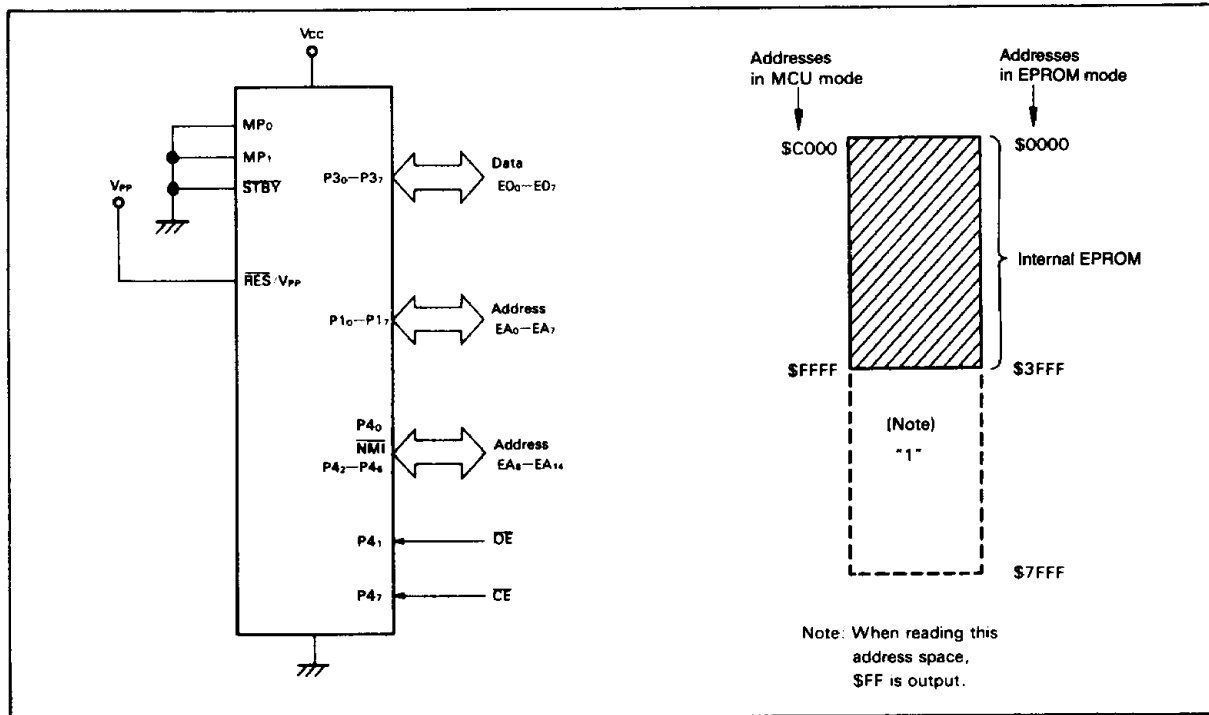


Figure 37. EPROM Mode Functional Diagram and Memory Map

Table 18. EPROM Programmers and Socket Adapters

EPROM Programmer		Socket Adapter	
Maker	Type Name	Maker	Type Name
DATA I/O	121B	Hitachi	HS31YESS11H
	22B		
	29B		
AVAL Corp	PKW-1000		HS31YESS21H

Table 19. EPROM Mode Selection

Mode	Pin			
	CE	OE	Vpp	E0 <sub>0</sub> —E0 <sub>7</sub>
Programming	Low	High	Vpp	Data input
Verify	High	Low	Vpp	Data output
Programming inhibited	High	High	Vpp	High impedance



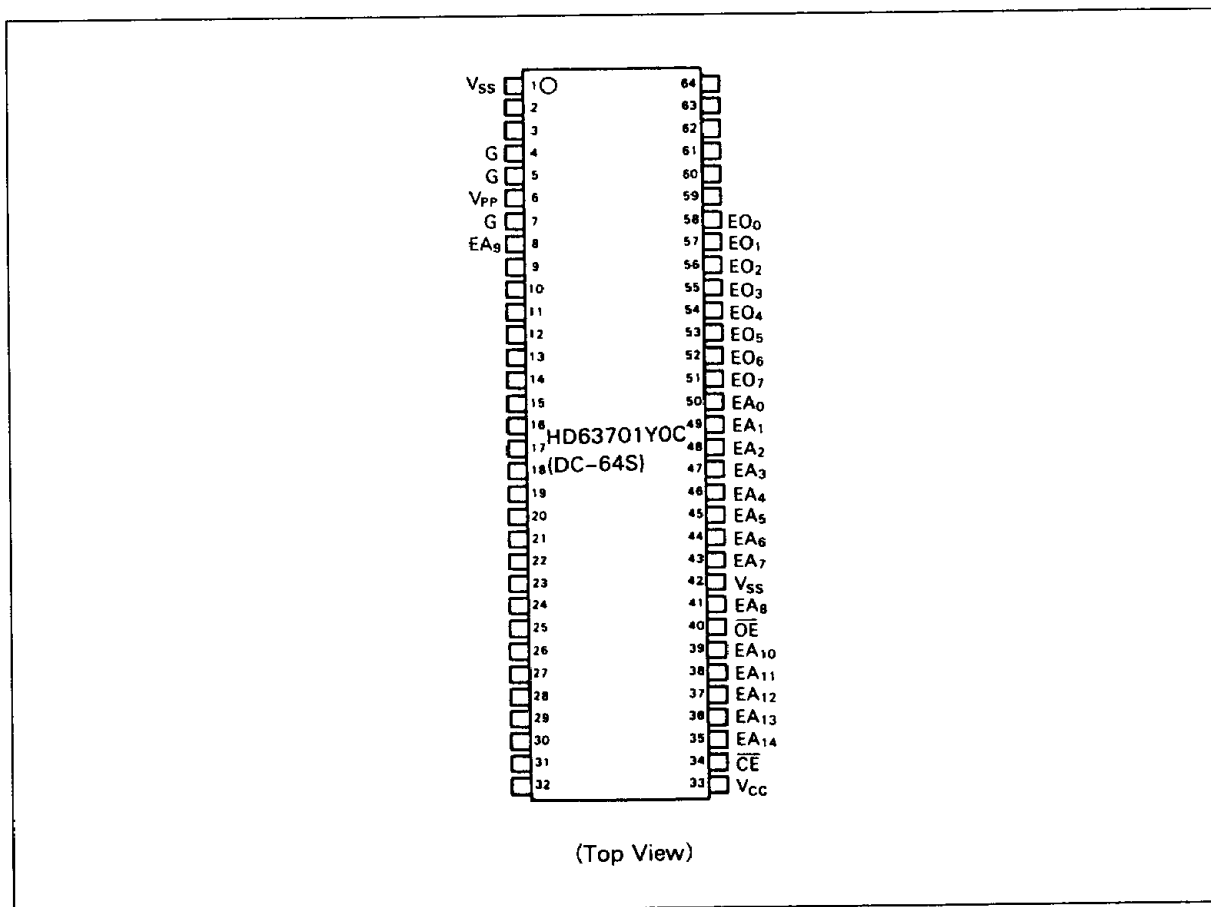


Figure 38. EPROM Mode Pin Arrangement

**Programming and Verification**

The HD63701Y0 can be high-speed programmed without causing voltage stress or affecting data reliability. Table 19 shows how programming and verification modes are selected. Figure 39 is a programming flowchart, and figure 54 is a timing chart.

Since the HD63701Y0 has a 16k byte capacity, when programming start at address \$0000 and end at \$3FFF (figure 37), and data from address \$4000 to \$7FFF should be programmed \$FF.

**Erasing**

The EPROM on HD63701Y0s in ceramic "window" packages can be erased by ultraviolet light. All erased bits become 1s.

Erasing conditions are : ultraviolet (UV) light with wavelength 2537 Å with a minimum irradiation of 15 W·s/cm<sup>2</sup>. These conditions are satisfied by exposing the LSI to a 12,000μW/cm<sup>2</sup> UV source for 15-20 minutes, at a distance of 1 inch.

**Precautions**

Note that the ceramic package HD63701Y0 can be erased and reprogrammed, but the plastic package type cannot.

If an attempt is made to access addresses of \$4000 or higher, the EPROM may not be programmed or verified correctly.

Be careful that the EPROM programmer, socket adapter and LSI match. Using the wrong programmer of socket adapter may cause an over-voltage and damage the LSI. Make sure that the LSI is firmly fixed in the socket adapter, and that the socket adapter is firmly fixed in the programmer.

The EPROM should be programmed with V<sub>PP</sub>=12.5 V. Other PROMs use 21 V. If 21 V is applied to the HD63701Y0, the LSI may be permanently damaged. 12.5 V is Intel's 27256 V<sub>PP</sub> specification.

To avoid erasure, shield the window of ceramic-package HD63701Y0s from UV light and electrostatic charges. Conductive, UV-opaque labels are available for this purpose.

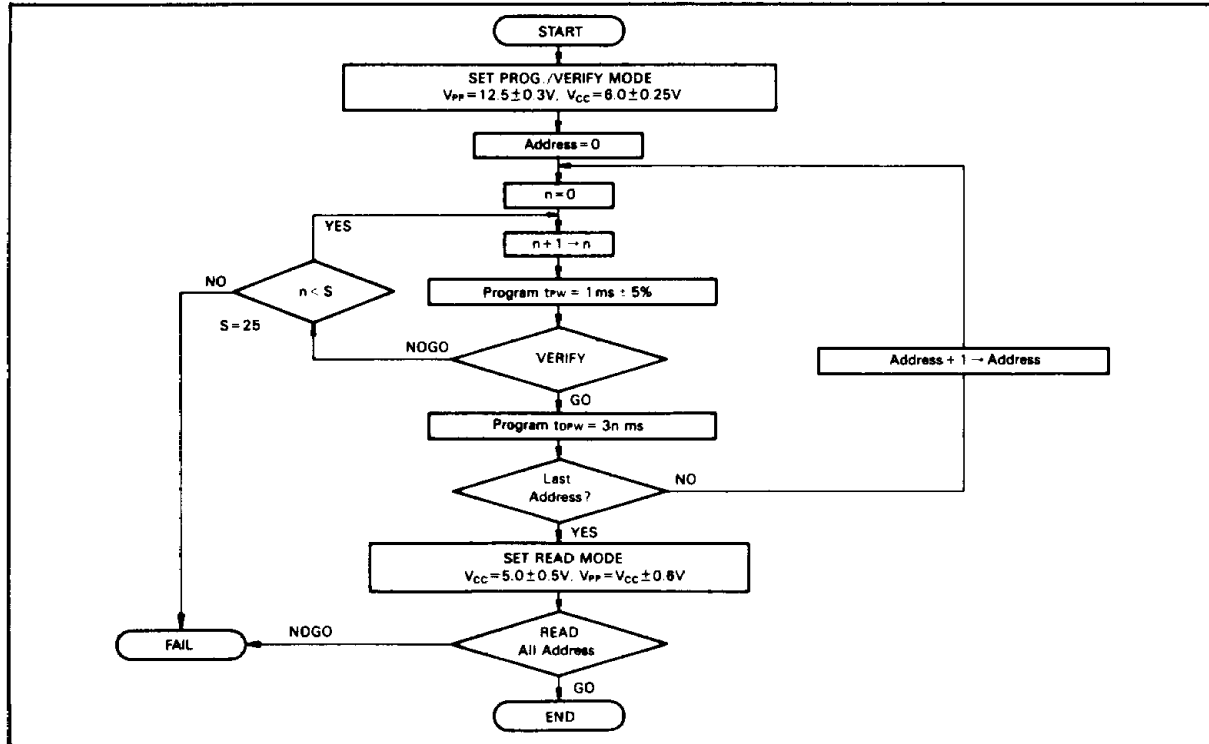


Figure 39. High-Speed Programming Flowchart

### Instruction Set

HD63701Y0 object code is upwardly compatible with the HD6801 to use the entire instruction set of the HMCS6800. It also reduces the execution times of key instructions to improve throughput. Bit manipulation, index and accumulator change, and sleep instructions have also been added.

### Addressing Modes

The HD63701Y0 provides 7 addressing modes.

**Accumulator (ACCX) Addressing:** Only one accumulator, A or B, is selected. These are one-byte instructions.

**Immediate Addressing:** The data is located in the second byte of the instruction, except for LDS and LDX, where the data is in the second and third bytes. These are two-or three-byte instructions.

**Direct Addressing:** The second byte of the instruction is the address that the data is stored at. Bytes \$00 through \$FF (0-255) can be addressed directly. Storing data in this area reduces execution time, so it is suggested that \$00-\$FF be used as user's data storage when configuring a system. These are two-byte instructions, or three bytes for AIM, OIM, EIM, or TIM.

**Extended Addressing:** The second byte is the upper 8 bits of the data's absolute address, and the third byte is the lower 8 bits. These are three-byte instructions.

**Indexed Addressing:** The lower eight bits of the index register are added to the second byte of the instruction (third byte for AIM, OIM, EIM, or TIM). The carry is added to the upper byte of the index register and the result is put in the temporary address register so the index register contents don't change. This result is the address of the data. These are two-byte instructions, or three bytes for AIM, OIM, EIM, or TIM.

**Implied Addressing:** The instruction only addresses particular operands, such as the stack pointer, or index register. These are one-byte instructions.

**Relative Addressing:** The second byte of the instruction is added to the lower eight bits of the program counter. The carry or borrow is added to the upper 8 bits. This allows addresses from -126 to +129 bytes from the current instruction to be accessed. These are two-byte instructions.

### Instruction Set Summary

Table 20 shows the general operation of each instruction. Table 21 is an opcode map. Table 22 shows the detailed operation of the instructions.

**Table 20. Instruction Set Summary**  
**Accumulator, Memory Manipulation Instructions**

Operations	Mnemonic	Addressing Modes												Boolean/ Arithmetic Operation	Condition Code Register								
		IMMED			DIRECT			INDEX			EXTEND				IMPLIED			5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#		OP	~	#	H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3			A + M → A	:	•	:	:	:	:	
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3			B + M → B	:	•	:	1	:	1	
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3			A : B + M : M + 1 → A : B	•	•	:	:	:	:	
Add Accumulators	ABA													1B	1	1	A + B → A	:	•	:	:	:	
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3			A + M + C → A	:	•	:	:	:		
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3			B + M + C → B	:	•	:	:	:		
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3			A · M → B	•	•	:	:	R	•	
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3			B · M → B	•	•	:	:	R	•	
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3			A · M	•	•	:	:	R	•	
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3			B · M	•	•	:	:	R	•	
Clear	CLR							6F	5	2	7F	5	3			00 → M	•	•	R	S	R	R	
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3			A - M	•	•	1	:	:	:	
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3			B - M	•	•	:	:	:	:	
Compare Accumulators	CBA													11	1	1	A - B	•	•	1	1	1	
Complement, 1's	COM							63	6	2	73	6	3			M → M	•	•	:	:	R	S	
	COMA													43	1	1	A → A	•	•	:	:	R	S
	COMB													53	1	1	B → B	•	•	:	:	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3			00 - M → M	•	•	1	:	1	2	
	NEGA													40	1	1	00 - A → A	•	•	:	:	1	2
	NEGB													50	1	1	00 - B → B	•	•	:	:	1	2
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	:	:	3	
Decrement	DEC							6A	6	2	7A	6	3			M - 1 → M	•	•	:	:	4	•	
	DECA													4A	1	1	A - 1 → A	•	•	:	:	4	•
	DECB													5A	1	1	B - 1 → B	•	•	:	:	4	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3			A ⊕ M → A	•	•	1	:	R	•	
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3			B ⊕ M → B	•	•	:	:	R	•	
Increment	INC							6C	6	2	7C	6	3			M + 1 → M	•	•	:	:	5	•	
	INCA													4C	1	1	A + 1 → A	•	•	:	:	5	•
	INCB													5C	1	1	B + 1 → B	•	•	:	:	5	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3			M → A	•	•	:	:	R	•	
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3			M → B	•	•	1	:	R	•	
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3			M + 1 → B, M → A	•	•	:	:	R	•	
Multiply Unsigned	MUL													3D	7	1	A × B → A : B	•	•	•	•	•	•
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3			A ∨ M → A	•	•	:	:	R	•	
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3			B ∨ M → B	•	•	1	:	R	•	
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•

(continued)



# HD63701Y0, HD637A01Y0, HD637B01Y0

## Accumulator, Memory Manipulation Instructions (Cont)

Operations	Mnemonic	Addressing Modes												Boolean/ Arithmetic Operation	Condition Code Register											
		IMMED			DIRECT			INDEX			EXTEND				IMPLIED			5	4	3	2	1	0			
		OP	~	#	OP	~	#	OP	~	#	OP	~	#		OP	~	#	H	I	N	Z	V	C			
Pul Data	PULA													32	3	1	SP+1→SP, Msp→A		●	●	●	●	●	●		
	PULB													33	3	1	SP+1→SP, Msp→B		●	●	●	●	●	●		
Rotate Left	ROL						69	6	2	79	6	3					●	●	!	!	⑥	!				
	ROLA												49	1	1		●	●	!	!	⑥	!				
	ROLB												59	1	1		●	●	!	!	⑥	!				
Rotate Right	ROR						66	6	2	76	6	3					●	●	!	!	⑥	!				
	RORA												46	1	1		●	●	!	!	⑥	!				
	RORB												56	1	1		●	●	!	!	⑥	!				
Shift Left Arithmetic	ASL						68	6	2	78	6	3					●	●	!	!	⑥	!				
	ASLA												48	1	1		●	●	!	!	⑥	!				
	ASLB												58	1	1		●	●	!	!	⑥	!				
Double Shift Left, Arithmetic	ASLD												05	1	1		●	●	!	!	⑥	!				
Shift Right Arithmetic	ASR						67	6	2	77	6	3					●	●	!	!	⑥	!				
	ASRA												47	1	1		●	●	!	!	⑥	!				
	ASRB												57	1	1		●	●	!	!	⑥	!				
Shift Right Logical	LSR						64	6	2	74	6	3					●	●	R	!	⑥	!				
	LSRA												44	1	1		●	●	R	!	⑥	!				
	LSRB												54	1	1		●	●	R	!	⑥	!				
Double Shift Right Logical	LSRD												04	1	1		●	●	R	!	⑥	!				
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A→M	●	●	!	!	R	●			
	STAB				D7	3	2	E7	4	2	F7	4	3				B→M	●	●	!	!	R	●			
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A→M B→M+1	●	●	!	!	R	●			
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A-M→A	●	●	!	!	!	!			
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B-M→B	●	●	!	!	!	!			
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A: B-M: M+1→ A: B	●	●	!	!	!	!			
Subtract Accumulators	SBA												10	1	1					A-B→A	●	●	!	!	!	!
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A-M-C→A	●	●	!	!	!	!			
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B-M-C→B	●	●	!	!	!	!			
Transfer Accumulators	TAB												16	1	1					A→B	●	●	!	!	R	●
	TBA												17	1	1					B→A	●	●	!	!	R	●
Test Zero or Minus	TST						6D	4	2	7D	4	3				M-00	●	●	!	!	R	R				
	TSTA												4D	1	1					A-00	●	●	!	!	R	R
	TSTB												5D	1	1					B-00	●	●	!	!	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	●	●	!	!	R	●			
OR Immediate	OIM				72	6	3	62	7	3							M∨IMM→M	●	●	!	!	R	●			
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	●	●	!	!	R	●			
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	●	●	!	!	R	●			



Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes												Boolean/ Arithmetic Operation	Condition Code Register								
		IMMED			DIRECT			INDEX			EXTEND				IMPLIED			5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#		OP	~	#	H	I	N	Z	V	C
Compare Index Reg	CPX	BC	3	3	9C	4	2	AC	5	2	BC	5	3			X-M:M+1	•	•	1	1	1	1	
Decrement Index Reg	DEX													09	1	1	X-1-X	•	•	•	1	•	•
Decrement Stack Pntr	DES													34	1	1	SP-1-SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X+1-X	•	•	•	1	•	•
Increment Stack Pntr	INS													31	1	1	SP+1-SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3			M-X <sub>H</sub> , (M+1)-X <sub>L</sub>	•	•	⑦	1	R	•	
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3			M-SP <sub>H</sub> , (M+1)-SP <sub>L</sub>	•	•	⑦	1	R	•	
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3			X <sub>H</sub> -M, X <sub>L</sub> -(M+1)	•	•	⑦	1	R	•	
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3			SP <sub>H</sub> -M, SP <sub>L</sub> -(M+1)	•	•	⑦	1	R	•	
Index Reg → Stack Pntr	TXS													35	1	1	X-1-SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP+1-X	•	•	•	•	•	•
Add	ABX													3A	1	1	B+X-X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X <sub>L</sub> -Msp, SP-1-SP X <sub>H</sub> -Msp, SP-1-SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP+1-SP, Msp-X <sub>H</sub> SP+1-SP, Msp-X <sub>L</sub>	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD-IX	•	•	•	•	•	•

Jump, Branch Instructions

Operations	Mnemonic	Addressing Modes												Branch Test	Condition Code Register									
		RELATIVE			DIRECT			INDEX			EXTEND				IMPLIED			5	4	3	2	1	0	
		OP	~	#	OP	~	#	OP	~	#	OP	~	#		OP	~	#	H	I	N	Z	V	C	
Branch Always	BRA	20	3	2														None	•	•	•	•	•	•
Branch Never	BRN	21	3	2														None	•	•	•	•	•	•
Branch if Carry Clear	BCC	24	3	2														C=0	•	•	•	•	•	•
Branch if Carry Set	BCS	25	3	2														C=1	•	•	•	•	•	•
Branch if =Zero	BEQ	27	3	2														Z=1	•	•	•	•	•	•
Branch if ≥Zero	BGE	2C	3	2														$N \oplus V = 0$	•	•	•	•	•	•
Branch if >Zero	BGT	2E	3	2														$Z + (N \oplus V) = 0$	•	•	•	•	•	•
Branch if Higher	BHI	22	3	2														C+Z=0	•	•	•	•	•	•
Branch if ≤Zero	BLE	2F	3	2														$Z + (N \oplus V) = 1$	•	•	•	•	•	•
Branch if Lower Or Same	BLS	23	3	2														C+Z=1	•	•	•	•	•	•
Branch if <Zero	BLT	2D	3	2														$N \oplus V = 1$	•	•	•	•	•	•
Branch if Minus	BMI	2B	3	2														N=1	•	•	•	•	•	•
Branch if Not Equal Zero	BNE	26	3	2														Z=0	•	•	•	•	•	•
Branch if Overflow Clear	BVC	28	3	2														V=0	•	•	•	•	•	•
Branch if Overflow Set	BVS	29	3	2														V=1	•	•	•	•	•	•
Branch if Plus	BPL	2A	3	2														N=0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2															•	•	•	•	•	•
Jump	JMP							6E	3	2		7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR							9D	5	2		AD	5	2		BD	6	3		•	•	•	•	•
No Operation	NOP														01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI														3B	10	1		•	•	•	•	•	•
Return From Subroutine	RTS														39	5	1		•	•	•	•	•	•
Software Interrupt	SWI														3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI														3E	9	1		•	Ⓢ	•	•	•	•
Sleep	SLP														1A	4	1		•	•	•	•	•	•

Note: \*WAI puts R/W high; Address Bus goes to FFFF; Data Bus goes to the three state.



Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code Register					
		IMPLIED				5	4	3	2	1	0
		OP	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	1	1	0→C	●	●	●	●	●	R
Clear Interrupt Mask	CLI	0E	1	1	0→I	●	R	●	●	●	●
Clear Overflow	CLV	0A	1	1	0→V	●	●	●	●	R	●
Sat Carry	SEC	0D	1	1	1→C	●	●	●	●	●	S
Set Interrupt Mask	SEI	0F	1	1	1→I	●	S	●	●	●	●
Set Overflow	SEV	0B	1	1	1→V	●	●	●	●	S	●
Accumulator A→CCR	TAP	06	1	1	A→CCR	⑩					
CCR→Accumulator A	TPA	07	1	1	CCR→A	●	●	●	●	●	●

— Description of symbols —

Legend :

- OP Operation Code (Hexadecimal)
- ~ Number of MCU Cycles
- M<sub>sp</sub> Contents of memory location pointed by Stack Pointer
- # Number of Program Bytes
- + Arithmetic plus
- Arithmetic Minus
- Boolean AND
- ⊕ Boolean Inclusive OR
- ⊕ Boolean Exclusive OR
- M Complement of M
- Transfer into
- 0 Bit=Zero
- 00 Byte=Zero

Condition Code Symbols :

- H Half-carry from bit 3 to bit 4
- I Interrupt mask
- N Negative (sign bit)
- Z Zero (byte)
- V Overflow, 2's complement
- C Carry/Borrow from/to bit 7
- R Reset Always
- S Set Always
- I Set if true after test or clear
- Not Affected

Note : Condition Code Register Notes : (Bit set if test is true and cleared otherwise)

- ① (Bit V) Test : Result=10000000?
- ② (Bit C) Test : Result≠00000000?
- ③ (Bit C) Test : BCD Character of high-order byte greater than 10? (Not cleared if previously set.)
- ④ (Bit V) Test : Operand=10000000 prior to execution?
- ⑤ (Bit V) Test : Operand=01111111 prior to execution?
- ⑥ (Bit V) Test : Set equal to N⊕C=1 after the execution of instructions
- ⑦ (Bit N) Test : Result less than zero? (Bit 15=1)
- ⑧ (All Bit) Load Condition Code Register from Stack.
- ⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
- ⑩ (All Bit) Set according to the contents of Accumulator A.
- ⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)





# HD63701Y0, HD637A01Y0, HD637B01Y0

Table 21. Opcode Map

OP CODE						ACC	ACC	IND	EXT/DIR*	ACCA or SP				ACCB or X					
		HI	0000	0001	0010	0011	A	B		IMM	DIR	IND	EXT	IMM	DIR	IND	EXT		
LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0000	0	/	SBA	BRA	TSX	NEG				SUB								0	
0001	1	NOP	CBA	BRN	INS	/	/	AIM		CMP								1	
0010	2	/	/	BHI	PULA	/	/	OIM		SBC								2	
0011	3	/	/	BLS	PULB	COM				SUBD		ADD							3
0100	4	LSRD	/	BCC	DES	LSR				AND								4	
0101	5	ASLD	/	BCS	TXS	/	/	EIM		BIT								5	
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6	
0111	7	TPA	TBA	BEQ	PSHB	ASR			/	STA		/	STA					7	
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8	
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9	
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A	
1011	B	SEV	ABA	BMI	RTI	/	/	TIM		ADD								B	
1100	C	CLC	/	BGE	PSHX	INC				CPX		LDD							C
1101	D	SEC	/	BLT	MUL	TST			BSR	JSR		/	STD					D	
1110	E	CLI	/	BGT	WAI	/	/	JMP		LDS		LDX							E
1111	F	SEI	/	BLE	SWI	CLR			/	STS		/	STX					F	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		


- Notes: 1. Undefined Opcode   
 2. \*Only each instructions of AIM, OIM, EIM, TIM



Table 22. Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W	RD	WR	LIR	Data Bus
-----------------------------	--------	---------	-------------	-----	----	----	-----	----------

**IMMEDIATE**

ADC ADD	2	1	Op Code Address+1	1	0	1	1	Operand Data
AND BIT		2	Op Code Address+2	1	0	1	0	Next Op Code
CMP EOR	3	1	Op Code Address+1	1	0	1	1	Operand Data (MSB)
LDA ORA		2	Op Code Address+2	1	0	1	1	Operand Data (LSB)
SBC SUB		3	Op Code Address+3	1	0	1	0	Next Op Code

**DIRECT**

ADC ADD	3	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
AND BIT		2	Address of Operand	1	0	1	1	Operand Data
CMP EOR		3	Op Code Address+2	1	0	1	0	Next Op Code
LDA ORA	3	1	Op Code Address+1	1	0	1	1	Destination Address
SBC SUB		2	Destination Address	0	1	0	1	Accumulator Data
		3	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX	4	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
LDD LDS		2	Address of Operand	1	0	1	1	Operand Data (MSB)
LDX SUBD		3	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		4	Op Code Address+2	1	0	1	0	Next Op Code
STD STS	4	1	Op Code Address+1	1	0	1	1	Destination Address (LSB)
STX		2	Destination Address	0	1	0	1	Register Data (MSB)
		3	Destination Address+1	0	1	0	1	Register Data (LSB)
		4	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Jump Address (LSB)
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer-1	0	1	0	1	Return Address (MSB)
		5	Jump Address	1	0	1	0	First Subroutine Op Code
TIM	4	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
AIM EIM	6	1	Op Code Address+1	1	0	1	1	Immediate Data
OIM		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code

(continued)

Table 22. Cycle-by-Cycle Operation (Cont)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W	RD	WR	LIR	Data Bus
<b>INDEXED</b>								
JMP	3	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Jump Address	1	0	1	0	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Accumulator Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD ADD	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data (MSB)
		4	IX+Offset+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Register Data (MSB)
		4	IX+Offset+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer-1	0	1	0	1	Return Address (MSB)
		5	IX+Offset	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	IX+Offset	0	1	0	1	New Operand Data
		6	Op Code Address+2	1	0	1	0	Next Op Code
TIM	5	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	IX+Offset	0	1	0	1	00
		5	Op Code Address+2	1	0	1	0	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	IX+Offset	0	1	0	1	New Operand Data
		7	Op Code Address+3	1	0	1	0	Next Op Code

(continued)



Table 22. Cycle-by-Cycle Operation (Cont)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W	RD	WR	LIR	Data Bus
<b>EXTEND</b>								
JMP	3	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	Jump Address	1	0	1	0	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Accumulator Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data (MSB)
		4	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Register Data (MSB)
		4	Destination Address+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
JSR	6	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Stack Pointer	0	1	0	1	Return Address (LSB)
		5	Stack Pointer-1	0	1	0	1	Return Address (MSB)
		6	Jump Address	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Address of Operand	0	1	0	1	00
		5	Op Code Address+3	1	0	1	0	Next Op Code

(continued)

Table 22. Cycle-by-Cycle Operation (Cont)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W	RD	WR	LIR	Data Bus
<b>IMPLIED</b>								
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	0	1	0	Next Op Code
DAA XGDX	2	1 2	Op Code Address+1 FFFF	1 1	0 1	1 1	0 1	Next Op Code Restart Address (LSB)
PULA PULB	3	1 2 3	Op Code Address+1 FFFF Stack Pointer+1	1 1 1	0 1 0	1 1 1	0 1 1	Next Op Code Restart Address (LSB) Data from Stack
PSHA PSHB	4	1 2 3 4	Op Code Address+1 FFFF Stack Pointer Op Code Address+1	1 1 0 1	0 1 1 0	1 1 0 1	1 1 1 0	Next Op Code Restart Address (LSB) Accumulator Data Next Op Code
PULX	4	1 2 3 4	Op Code Address+1 FFFF Stack Pointer+1 Stack Pointer+2	1 1 1 1	0 1 0 0	1 1 1 1	0 1 1 1	Next Op Code Restart Address (LSB) Data from Stack (MSB) Data from Stack (LSB)
PSHX	5	1 2 3 4 5	Op Code Address+1 FFFF Stack Pointer Stack Pointer-1 Op Code Address+1	1 1 0 0 1	0 1 1 1 0	1 1 0 0 1	1 1 1 1 0	Next Op Code Restart Address (LSB) Index Register (LSB) Index Register (MSB) Next Op Code
RTS	5	1 2 3 4 5	Op Code Address+1 FFFF Stack Pointer+1 Stack Pointer+2 Return Address	1 1 1 1 1	0 1 0 0 0	1 1 1 1 1	1 1 1 1 0	Next Op Code Restart Address (LSB) Return Address (MSB) Return Address (LSB) First Op Code of Return Routine
MUL	7	1 2 3 4 5 6 7	Op Code Address+1 FFFF FFFF FFFF FFFF FFFF FFFF	1 1 1 1 1 1 1	0 1 1 1 1 1 1	1 1 1 1 1 1 1	0 1 1 1 1 1 1	Next Op Code Restart Address (LSB) Restart Address (LSB) Restart Address (LSB) Restart Address (LSB) Restart Address (LSB) Restart Address (LSB) Restart Address (LSB)

(continued)



Table 22. Cycle-by-Cycle Operation (Cont)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
<b>IMPLIED</b>								
WAI	9	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer-1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer-2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer-3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer-4	0	1	0	1	Accumulator A
		8	Stack Pointer-5	0	1	0	1	Accumulator B
		9	Stack Pointer-6	0	1	0	1	Conditional Code Register
RTI	10	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Conditional Code Register
		4	Stack Pointer+2	1	0	1	1	Accumulator A
		5	Stack Pointer+3	1	0	1	1	Accumulator B
		6	Stack Pointer+4	1	0	1	1	Index Register (MSB)
		7	Stack Pointer+5	1	0	1	1	Index Register (LSB)
		8	Stack Pointer+6	1	0	1	1	Return Address (MSB)
		9	Stack Pointer+7	1	0	1	1	Return Address (LSB)
		10	Return Address	1	0	1	0	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer-1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer-2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer-3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer-4	0	1	0	1	Accumulator A
		8	Stack Pointer-5	0	1	0	1	Accumulator B
		9	Stack Pointer-6	0	1	0	1	Conditional Code Register
		10	Vector Address FFFA	1	0	1	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	0	1	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	0	1	0	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Op Code Address+1	1	0	1	0	Next Op Code
		↑		↓		↓		
		Sleep						

(continued)



# HD63701Y0, HD637A01Y0, HD637B01Y0

**Table 22. Cycle-by-Cycle Operation (Cont)**

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W	RD	WR	LIR	Data Bus
-----------------------------	--------	---------	-------------	-----	----	----	-----	----------

**RELATIVE**

BCC BCS	3	1	Op Code Address + 1	1	0	1	1	Branch Offset	
BEQ BGE		2	FFFF	1	1	1	1	Restart Address (LSB)	
BGT BHI		3	3	{ Branch Address...Test="1" { Op Code Address+2...Test="0"	1	0	1	0	First Op Code of Branch Routine Next Op Code
BLE BLS									
BLT BMT									
BNE BPL									
BRA BRN									
BVC BVS									
BSR	5	1	Op Code Address + 1	1	0	1	1	Offset	
		2	FFFF	1	1	1	1	Restart Address (LSB)	
		3	Stack Pointer	0	1	0	1	Return Address (LSB)	
		4	Stack Pointer - 1	0	1	0	1	Return Address (MSB)	
		5	Branch Address	1	0	1	0	First Op Code of Subroutine	

## The Differences Between HD63701Y0 and HD6301Y0

Item	HD63701Y0	HD6301Y0										
Input low voltage of $\overline{RES}$ , $MP_0$ , $MP_1$	$V_{IL} = 0.6 \text{ V max}$	$V_{IL} = 0.8 \text{ V max}$										
$I_{in}$ and $C_{in}$ of $\overline{RES}$	$I_{in} = 10 \mu\text{A max}$ $C_{in} = 65 \text{ pF max}$ $I_{in}$ and $C_{in}$ are larger than HD6301Y0 because $\overline{RES}$ is also used as $V_{PP}$ .	$I_{in} = 1.0 \mu\text{A max}$ $C_{in} = 12.5 \text{ pF max}$										
Crystal oscillator characteristics	Internal resistance of crystal oscillator $R_S$  <table border="1"> <thead> <tr> <th>Frequency (MHz)</th> <th>2.5</th> <th>4.0</th> <th>6.0</th> <th>8.0</th> </tr> </thead> <tbody> <tr> <td><math>R_S \text{ max } (\Omega)</math></td> <td>500</td> <td>120</td> <td>80</td> <td>60</td> </tr> </tbody> </table>	Frequency (MHz)	2.5	4.0	6.0	8.0	$R_S \text{ max } (\Omega)$	500	120	80	60	Internal resistance of crystal oscillator $R_S$  $R_S = 60 \Omega \text{ max}$
Frequency (MHz)	2.5	4.0	6.0	8.0								
$R_S \text{ max } (\Omega)$	500	120	80	60								
Storage temperature	$T_{stg} = -55 \text{ to } 125 \text{ }^\circ\text{C}$	$T_{stg} = -55 \text{ to } 150 \text{ }^\circ\text{C}$										
Caution	The HD63701Y0 differs from HD6301Y0 in chip design and manufacturing process. When applying the HD63701Y0 system to HD6301Y0, and HD6301Y0 system to HD63701Y0, note that characteristic values are not exactly the same even if guaranteed values are the same.											



### Absolute Maximum Ratings

Item	Symbol	Value	Unit
Supply voltage	V <sub>CC</sub>	-0.3 to +7.0	V
V <sub>PP</sub> voltage	V <sub>PP</sub>	-0.3 to +13.0	V
Input voltage	V <sub>in</sub>	-0.3 to V <sub>CC</sub> +0.3	V
Operating temperature	T <sub>opr</sub>	0 to +70	°C
Storage temperature	T <sub>stg</sub>	-55 to +125	°C

Note: This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend  $V_{in}, V_{out} : V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{CC}$ .

### Electrical Characteristics

#### DC Characteristics

(V<sub>CC</sub>=5.0 V ± 10%, f=0.1 to 2.0 MHz, V<sub>SS</sub>=0 V, T<sub>a</sub>=0 to +70 °C, unless otherwise noted.)

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Input high voltage	RES, STBY, MP <sub>0</sub> , MP <sub>1</sub>	V <sub>IH</sub>	V <sub>CC</sub> -0.5		V <sub>CC</sub> +0.3	V	
	EXTAL		V <sub>CC</sub> ×0.7		V <sub>CC</sub> +0.3	V	
	Other inputs		2.0		V <sub>CC</sub> +0.3	V	
Input low voltage	RES, MP <sub>0</sub> , MP <sub>1</sub> , SCLK(P2) <sup>3</sup>	V <sub>IL</sub>	-0.3		0.6	V	
	All other inputs		-0.3		0.8	V	
Input leakage current	RES	I <sub>in</sub>			10.0	μA	V <sub>in</sub> =0.5 to V <sub>CC</sub> -0.5 V
	NMI, STBY, MP <sub>0</sub> , MP <sub>1</sub>				1.0	μA	
Three state leakage current	Ports 1, 2, 3, 4, 5, 6, 7	I <sub>TSI</sub>			1.0	μA	V <sub>in</sub> =0.5 to V <sub>CC</sub> -0.5 V
Output high voltage		V <sub>OH</sub>	2.4			V	I <sub>OH</sub> =-200 μA
			V <sub>CC</sub> -0.7			V	I <sub>OH</sub> =-10 μA
Output low voltage		V <sub>OL</sub>			0.4	V	I <sub>OL</sub> =1.6 mA
Darlington drive current	Ports 2, 6	-I <sub>OH</sub>	1.0		10.0	mA	V <sub>out</sub> =1.5 V
Input capacitance	RES	C <sub>in</sub>			65	pF	V <sub>in</sub> =0 V, f=1 MHz, T <sub>a</sub> =25°C
	All other inputs				12.5	pF	
Standby current	Not operating	I <sub>STB</sub>		3.0	15.0	μA	
Current dissipation <sup>1</sup>		I <sub>SLP</sub>		1.5	3.0	mA	Sleeping (f=1 MHz <sup>2</sup> )
				2.3	4.5	mA	Sleeping (f=1.5 MHz <sup>2</sup> )
				3.0	6.0	mA	Sleeping (f=2 MHz <sup>2</sup> )
		I <sub>CC</sub>		7.0	10.0	mA	Operating (f=1 MHz <sup>2</sup> )
				10.5	15.0	mA	Operating (f=1.5 MHz <sup>2</sup> )
				14.0	20.0	mA	Operating (f=2 MHz <sup>2</sup> )
RAM standby voltage		V <sub>RAM</sub>	2.0			V	

Notes :

- V<sub>IH</sub> min=V<sub>CC</sub>-1.0V, V<sub>IL</sub> max=0.8V (All output terminals are at no load.)
- Current dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about current dissipations at x MHz operation are decided according to the following formula:  
 typ. value (f=x MHz) = typ. value (f=1 MHz) × x  
 max. value (f=x MHz) = max. value (f=1 MHz) × x  
 (both the sleeping and operating)
- Only serial clock use.





# HD63701Y0, HD637A01Y0, HD637B01Y0

## AC Characteristics

( $V_{CC}=5.0V \pm 10\%$ ,  $f=0.1$  to  $2.0$  MHz,  $V_{SS}=0V$ ,  $T_a=0$  to  $+70$  °C, unless otherwise noted.)

## Bus Timing

Item	Symbol	HD63701Y0			HD637A01Y0			HD637B01Y0			Unit	Test Condition
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
Cycle time	$t_{cyc}$	1		10	0.666		10	0.5		10	$\mu s$	Fig. 40
Enable rise time	$t_{Er}$			25			25			25	ns	
Enable fall time	$t_{Ef}$			25			25			25	ns	
Enable pulse width high level <sup>1</sup>	$PW_{EH}$	450			300			220			ns	
Enable pulse width low level <sup>1</sup>	$PW_{EL}$	450			300			220			ns	
Address, R/ $\bar{W}$ delay time <sup>1</sup>	$t_{AD}$			250			190			160	ns	
Data delay time (Write)	$t_{DDW}$			200			160			120	ns	
Data set-up time (Read)	$t_{DSR}$	80			70			60			ns	
Address, R/ $\bar{W}$ hold time <sup>1</sup>	$t_{AH}$	80			50			40			ns	
Data hold time (Write) <sup>1</sup>	$t_{HW}$	80			50			40			ns	
Data hold time (Read)	$t_{HR}$	0			0			0			ns	
$\overline{RD}$ , $\overline{WR}$ pulse width <sup>1</sup>	$PW_{RW}$	450			300			220			ns	
$\overline{RD}$ , $\overline{WR}$ delay time	$t_{RWD}$			40			40			40	ns	
$\overline{RD}$ , $\overline{WR}$ hold time	$t_{HRW}$			20			20			20	ns	
$\overline{LIR}$ delay time	$t_{DLR}$			200			160			120	ns	
$\overline{LIR}$ hold time	$t_{HLR}$	10			10			10			ns	
MR set-up time <sup>1</sup>	$t_{SMR}$	400			280			230			ns	Fig. 41
MR hold time <sup>1</sup>	$t_{HMR}$			100			70			50	ns	
E clock pulse width at MR	$PW_{EMR}$			9			9			9	$\mu s$	
Processor control set-up time	$t_{PCS}$	200			200			200			ns	Figs. 42, 52, 53
Processor control rise time	$t_{PCr}$			100			100			100	ns	Figs. 41, 42
Processor control fall time	$t_{PCf}$			100			100			100	ns	
BA delay time	$t_{BA}$			250			190			160	ns	Fig. 42
Oscillator stabilization time	$t_{RC}$	20			20			20			ms	Fig. 53
Reset pulse width	$PW_{RST}$	3			3			3			$t_{cyc}$	

Note: 1. These timings change in approximate proportion to  $t_{cyc}$ . The figures in this characteristics represent those when  $t_{cyc}$  is minimum (=in the highest speed operation).



Peripheral Port Timing

Item	Symbol	HD63701Y0			HD637A01Y0			HD637B01Y0			Unit	Test Condition
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
Peripheral data set-up time (Ports 1, 2, 3, 4, 5, 6)	t <sub>PDSU</sub>	200			200			200			ns	Fig. 44
Peripheral data hold time (Ports 1, 2, 3, 4, 5, 6)	t <sub>PDH</sub>	200			200			200			ns	
Delay time (From enable fall edge to peripheral output) (Ports 1, 2, 3, 4, 5, 6, 7)	t <sub>PWD</sub>			300			300			300	ns	Fig. 45
Input strobe pulse width	t <sub>PWIS</sub>	200			200			200			ns	Fig. 49
Input data hold time (Port 6)	t <sub>IH</sub>	150			150			150			ns	
Input data set-up time (Port 6)	t <sub>IS</sub>	100			100			100			ns	
Output strobe delay time	t <sub>OSD1</sub> t <sub>OSD2</sub>			200			200			200	ns	Fig. 50

Timer, SCI Timing

Item	Symbol	HD63701Y0			HD637A01Y0			HD637B01Y0			Unit	Test Condition
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
Timer 1 input pulse width	t <sub>PWT</sub>	2.0			2.0			2.0			t <sub>cyc</sub>	Fig. 48
Delay time (enable positive transition to timer output)	t <sub>TOD</sub>			400			400			400	ns	Figs. 46, 47
SCI input clock cycle (Async. mode)	t <sub>Scyc</sub>	1.0			1.0			1.0			t <sub>cyc</sub>	Fig. 48
SCI input clock cycle (Clock sync.)		2.0			2.0			2.0			t <sub>cyc</sub>	Fig. 43
SCI transmit data delay time (Clock sync. mode)	t <sub>TXD</sub>			220			220			220	ns	Fig. 43
SCI receive data set-up time (Clock sync. mode)	t <sub>SRX</sub>	260			260			260			ns	
SCI receive data hold time (Clock sync. mode)	t <sub>HRX</sub>	100			100			100			ns	
SCI input clock pulse width	t <sub>PWCK</sub>	0.4		0.6	0.4		0.6	0.4		0.6	t <sub>Scyc</sub>	Fig. 48
Timer 2 input clock cycle	t <sub>cyc</sub>	2.0			2.0			2.0			t <sub>cyc</sub>	
Timer 2 input clock pulse width	t <sub>PWCK</sub>	200			200			200			ns	
Timer 1 - 2, SCI input clock rise time	t <sub>CKr</sub>			100			100			100	ns	
Timer 1 - 2, SCI input clock fall time	t <sub>CKf</sub>			100			100			100	ns	

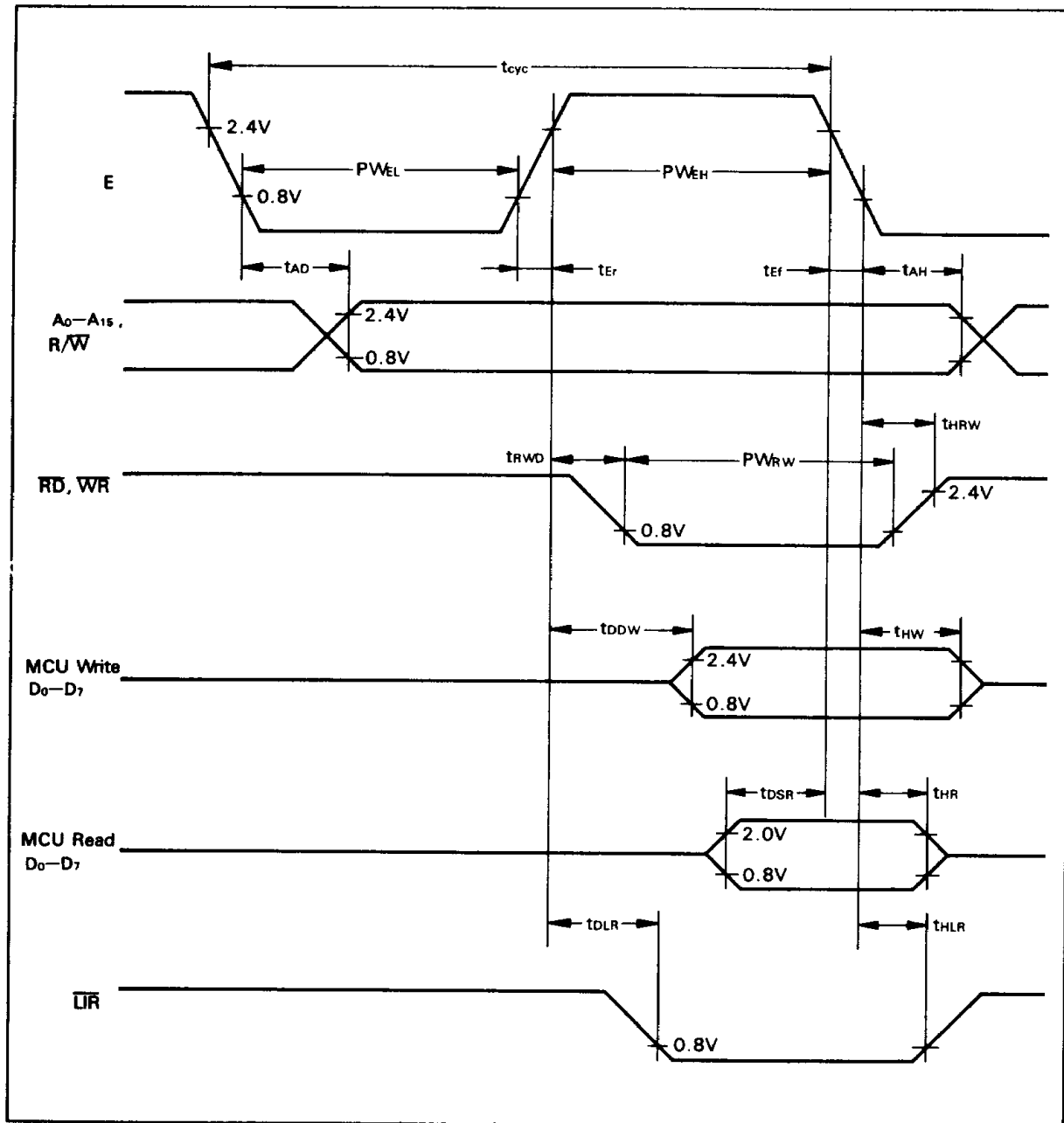


Figure 40. Mode 1, Mode 2 Bus Timing

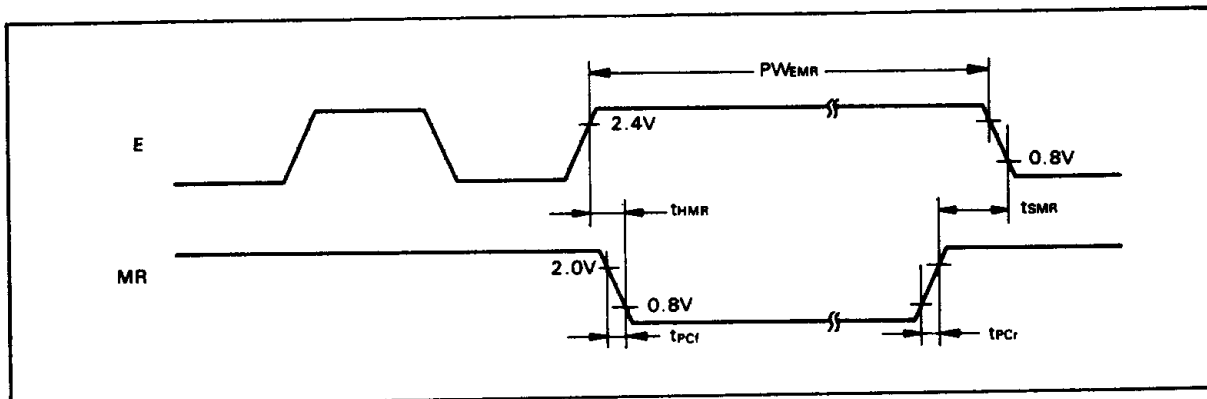


Figure 41. Memory Ready and E Clock Timing

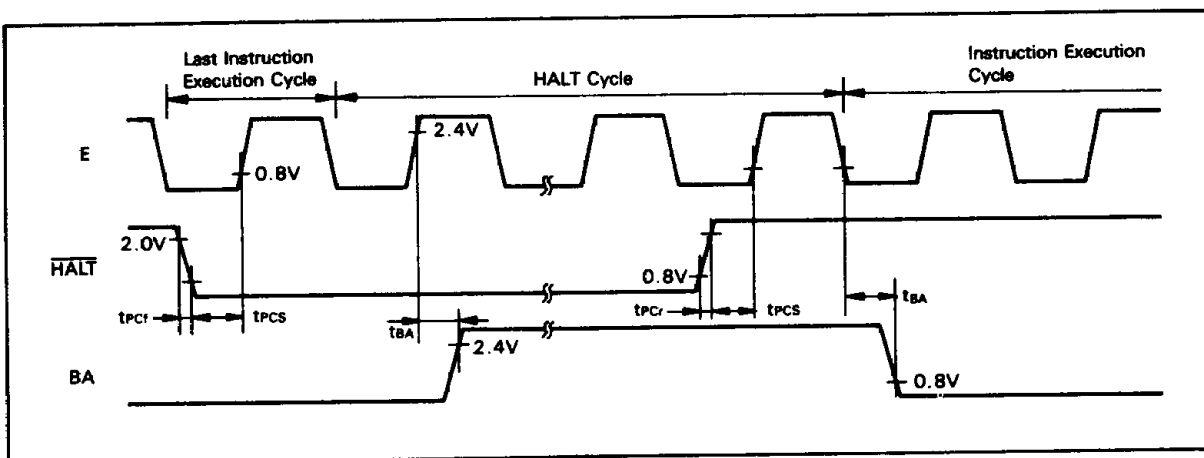


Figure 42. HALT and BA Timing

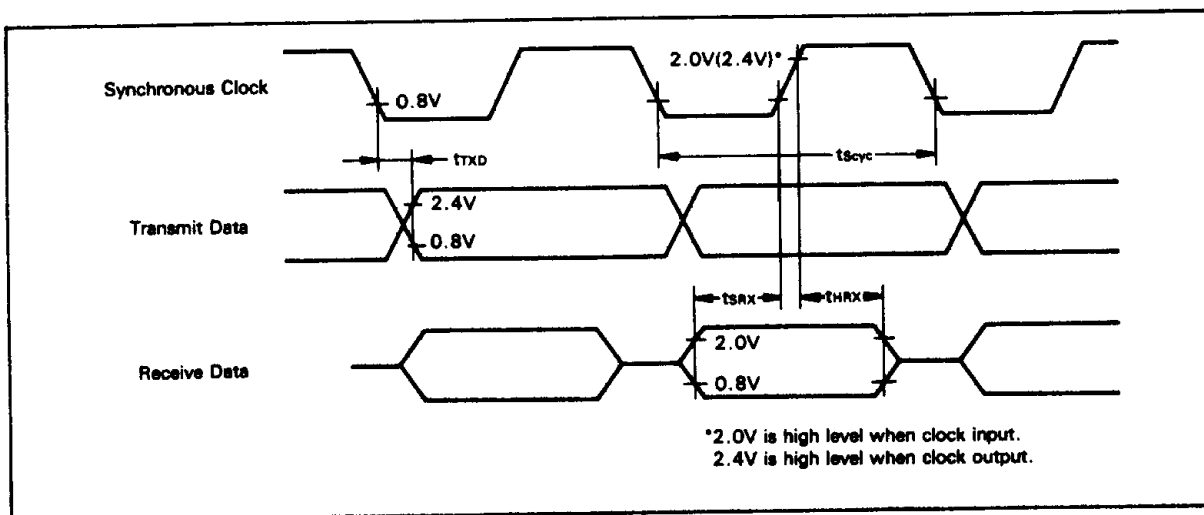


Figure 43. SCI Clocked Synchronous Timing

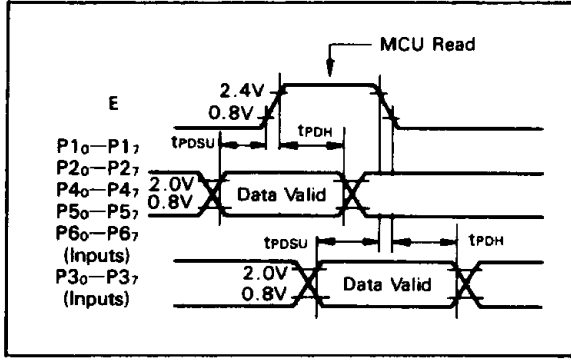


Figure 44. Port Data Set-up and Hold Times (MCU Read)

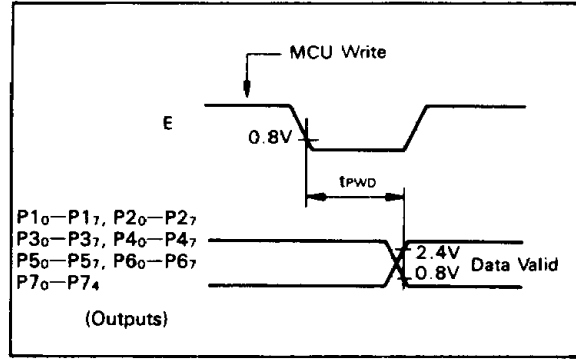


Figure 45. Port Data Delay Times (MCU Write)

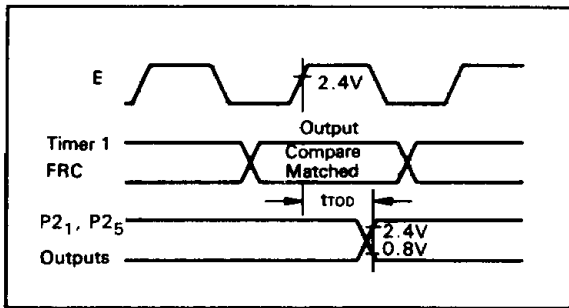


Figure 46. Timer 1 Output Timing

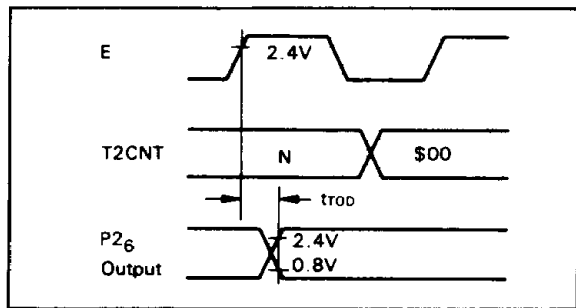


Figure 47. Timer 2 Output Timing

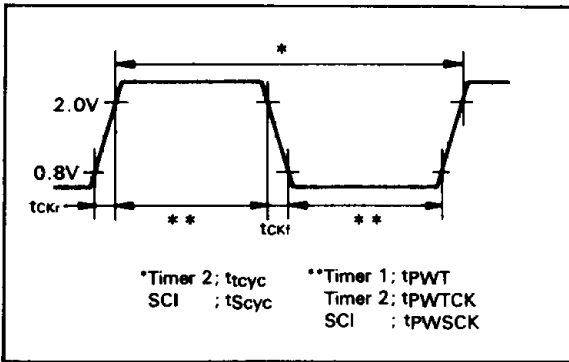


Figure 48. Timer 1 - 2, SCI Input Clock Timing

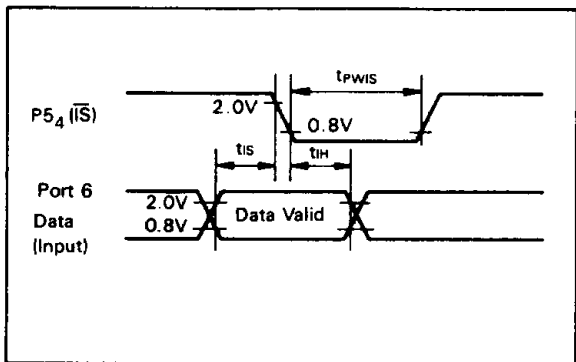


Figure 49. Port 6 Input Latch Timing

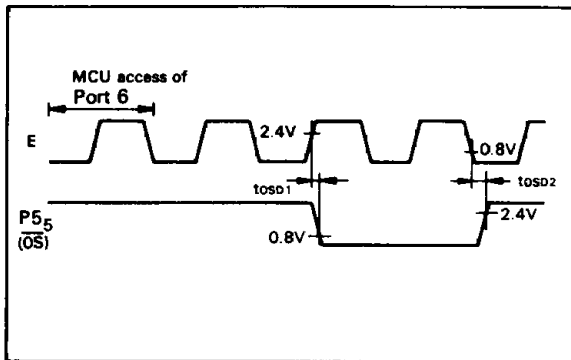


Figure 50. Output Strobe Timing

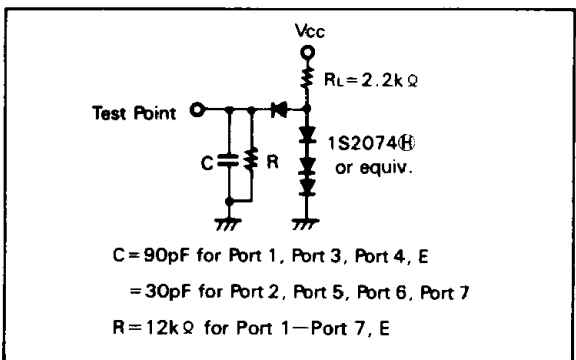


Figure 51. Bus Timing Test Loads (TTL Load)

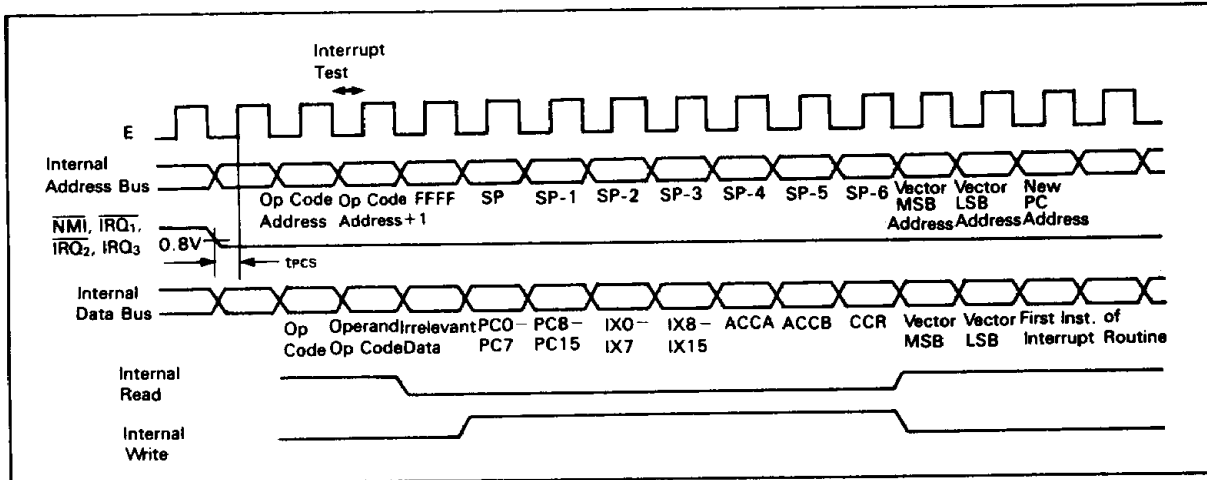


Figure 52. Interrupt Sequence

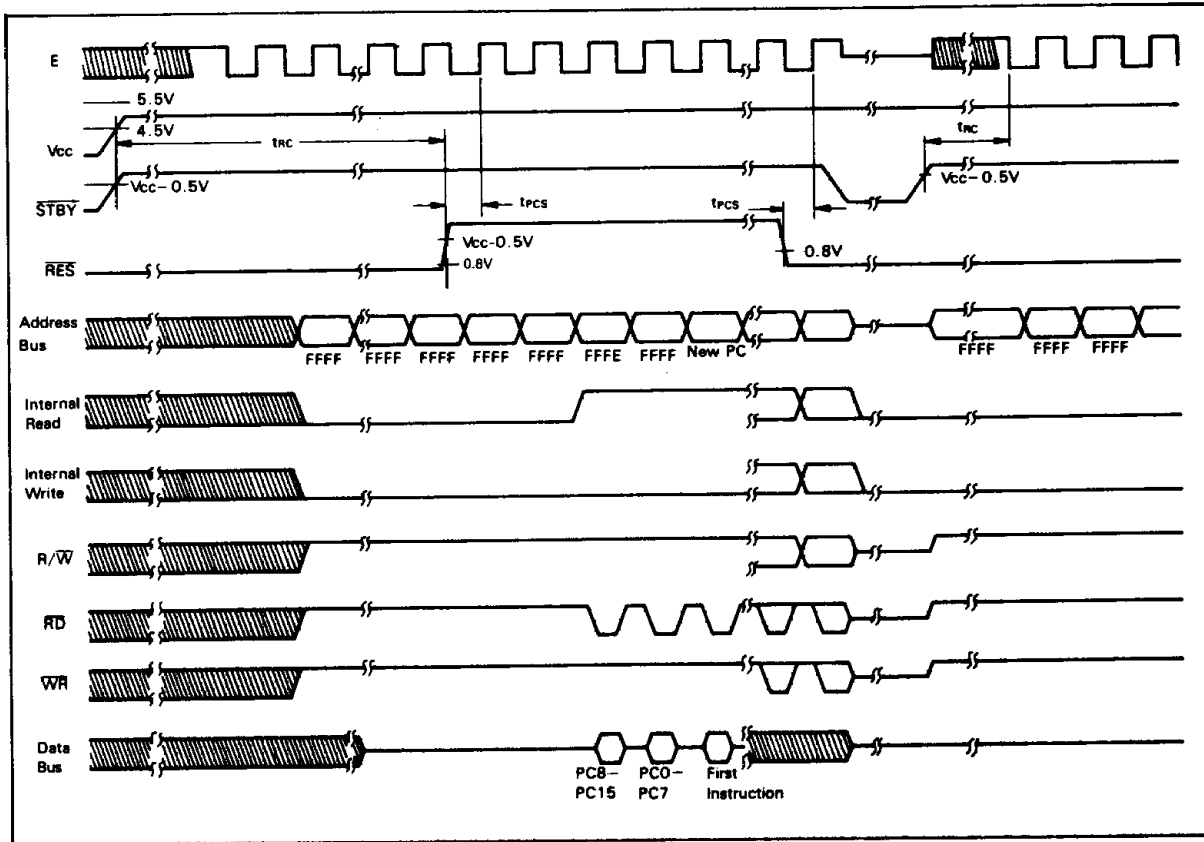


Figure 53. Reset Timing

**Programming Electrical Characteristics**

**DC Characteristics**

( $V_{CC}=6V \pm 0.25V$ ,  $V_{PP}=12.5V \pm 0.3V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C \pm 5^\circ C$ , unless otherwise notes.)

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Input high voltage	$O_0-O_7, A_0-A_{14}, \overline{OE}, \overline{CE}$	$V_{IH}$	2.2	—	$V_{CC}+0.3$	V
Input low voltage	$O_0-O_7, A_0-A_{14}, \overline{OE}, \overline{CE}$	$V_{IL}$	-0.3	—	0.8	V
Output high voltage	$O_0-O_7$	$V_{OH}$	2.4	—	—	V $I_{OH}=-200\mu A$
Output low voltage	$O_0-O_7$	$V_{OL}$	—	—	0.45	V $I_{OL}=1.6mA$
Input leakage current	$O_0-O_7, A_0-A_{14}, \overline{OE}, \overline{CE}$	$ I_{L} $	—	—	2	$\mu A$ $V_{in}=5.25V/0.5V$
$V_{CC}$ current		$I_{CC}$	—	—	30	mA
$V_{PP}$ current		$I_{PP}$	—	—	40	mA

**AC Characteristics**

( $V_{CC}=6V \pm 0.25V$ ,  $V_{PP}=12.5V \pm 0.3V$ ,  $T_a=25^\circ C \pm 5^\circ C$ , unless otherwise noted.)

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Address set-up time	$t_{AS}$	2	—	—	$\mu S$	Fig. 54*
$\overline{OE}$ set-up time	$t_{OES}$	2	—	—	$\mu S$	
Data set-up time	$t_{DS}$	2	—	—	$\mu S$	
Address hold time	$t_{AH}$	0	—	—	$\mu S$	
Data hold time	$t_{DH}$	2	—	—	$\mu S$	
Output disable delay time	$t_{DF}$	—	—	130	ns	
$V_{PP}$ set-up time	$t_{VPS}$	2	—	—	$\mu S$	
Program pulse width	$t_{PW}$	0.95	1.0	1.05	ms	
$\overline{CE}$ pulse width when overprogramming	$t_{OPW}$	2.85	—	78.75	ms	
$V_{CC}$ set-up time	$t_{VCS}$	2	—	—	$\mu S$	
Data output delay time	$t_{OE}$	0	—	500	ns	

Note: \* Input Pulse level 0.8~2.2V  
 Input rising/falling time  $\leq 20ns$   
 Timing reference level { input : 1.0V, 2.0V  
 output : 0.8V, 2.0V

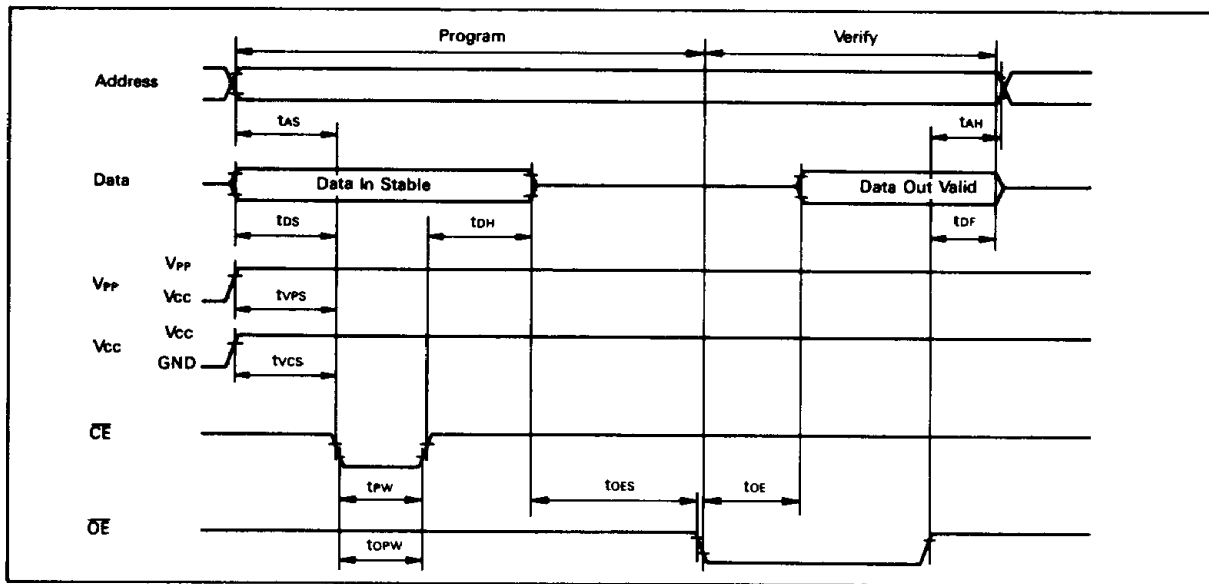
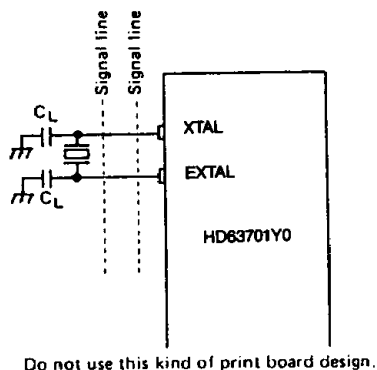


Figure 54. EPROM Programming/Verify timing



**Warning Concerning the Board of Oscillation Circuit**

When designing a board, note that crosstalk may disturb the normal oscillation if signal lines are placed near the oscillation circuit as shown in Figure 55. Place the crystal and  $C_L$  as close to the HD63701Y0 as possible.



Do not use this kind of print board design.

Figure 55 Warning concerning board design of oscillation circuit

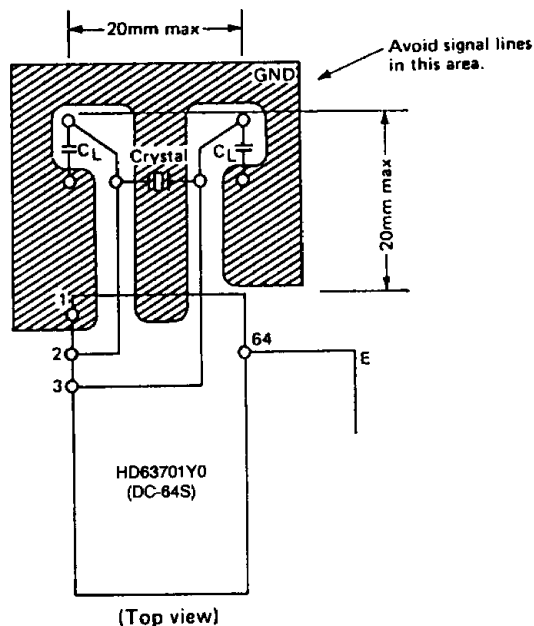


Figure 56 Example of Oscillation Circuits in Board Design

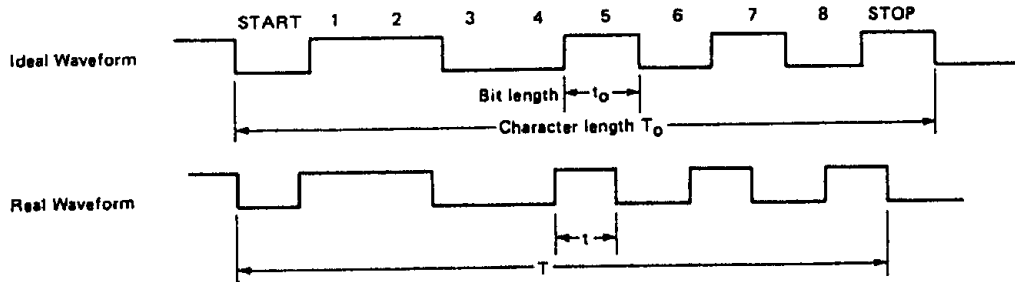
**Receive Margin of the SCI**

Receive margin of the SCI contained in the HD63701Y0 is shown in Table 22.

Note: SCI = Serial Communication Interface.

Table 23

Bit distortion tolerance ( $t-t_0$ ) / $t_0$	Character distortion tolerance ( $T-T_0$ ) / $T_0$
$\pm 43.7\%$	$\pm 4.37\%$





**Warning Concerning WAI Instruction**

If the HALT signal is accepted by the MCU while the WAI instruction is executing, the CPU will not operate correctly after HALT mode is canceled.

WAI is a instruction which waits for an interrupt. The corresponding interrupt routine is executed after an interrupt occurs.

However, during the execution of the WAI instruction, HALT input makes the CPU malfunction and fetch an abnormal interrupt vectoring address.

In HALT mode, the CPU operates correctly without the WAI instruction and WAI is executed correctly without HALT input. Therefore, if HALT input is necessary, make interrupts wait during the loop routine, as shown in Figure 57.

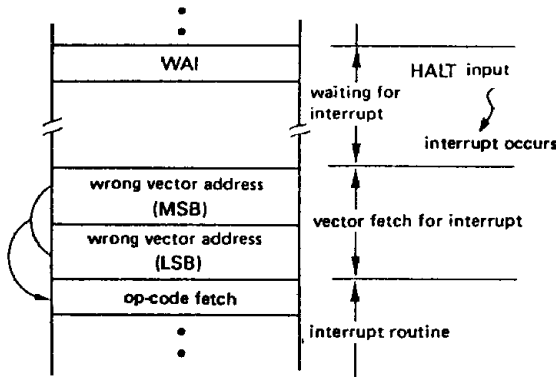


Figure 57 MAC function during WAI

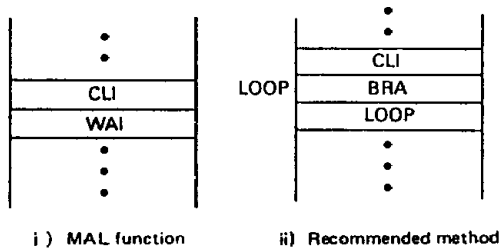


Figure 58 Program to wait for interrupt

**Write-Only Register**

When the CPU reads a write-only register, the read data is always \$FF, regardless of the value in the write-only register. Therefore, be careful of the results of instructions which read write-only register and perform an arithmetic or logical operation on its contents, such as AIM, ADD, or ROL, is executed, because the arithmetic or logical operation is always done with the data \$FF. In particular, don't use the AIM, OIM or EIM instruction to manipulate the DDR bit of PORT.

**Warning Concerning Power Start-Up**

RES must be held low for at least 20 ms when the power starts up. In this case, the internal reset function is not effective until the oscillation begins at power-on. The RES signal is input to the LSI in synchronism with the internal clock  $\phi$  (shown in Figure 59.)

Therefore, after power starts up, the LSI conditions such as its I/O ports and operating mode, are unstable. Fix the level of I/O ports by means of an external circuit to determine the level for system operation during the oscillation stabilization time.

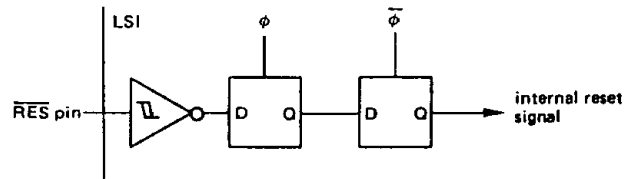


Figure 59 RES circuit