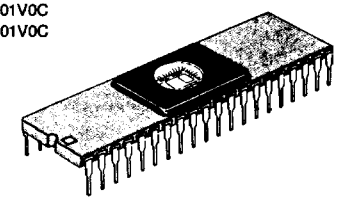
HD63701V0, HD637A01V0
HD637B01V0

(Limiting Supplies. For Development Only.)

The HD63701V0 is an 8-bit CMOS single-chip microcomputer unit, pin compatible with th HD6301V1. 4kB EPROM, 192 bytes RAM, Serial Communication Interface (SCI), parallel I/O ports and multi function timer are incorporated in the HD63701V0. It is bus compatible with HMCS6800. Execution time of key instructions are improved and several new instructions are added to increase system throughout. The HD63701V0 can be expanded up to 65k words. Like the HMCS6800 family, I/O level is TTL compatible with +5.0V single power supply. As HD63701V0 is fabricated by the advanced CMOS process technology, power dissipation is extremely reduced. In addition to that, HD63701V0 has Sleep Mode and Standby Mode at lower power dissipation mode. Therefore flexible low power consumption application is possible.

On chip EPROM can be programmed and erased by the same procedure as that of 27C256 or 27256.

HD63701V0C
HD637A01V0C
HD637B01V0C



(DC-40)

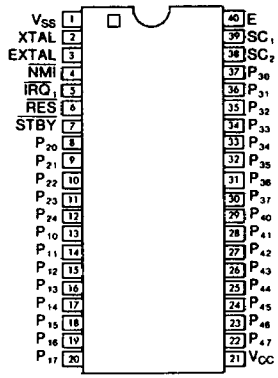
■ FEATURES

- Instruction Set Compatible with HD6301 Family
- Abundant On-Chip Functions
  4kB EPROM, 192 Bytes RAM, 29 Parallel I/O Lines, 2 Lines of Data Strobe, 16-bit Timer, Serial Communication Interface
- Low Power Consumption Mode: Sleep Mode, Standby Mode
- Minimum Instruction Execution Time
  1μs (f = 1MHz), 0.67μs (f = 1.5MHz), 0.5μs (f = 2MHz)
- Bit Manipulation, Bit Test Instruction
- Protection from System Upset: Address Trap, Op-Code Trap
- Up to 65k Words Address Space
- Wide Operation Range
  f = 0.1 to 2.0MHz (V_{CC} = 5V ± 10%)

■ TYPE OF PRODUCTS

TYPE No.	BUS TIMING
HD63701V0	1.0 MHz
HD637A01V0	1.5 HMz
HD637B01V0	2.0 HMz

■ PIN ARRANGEMENT

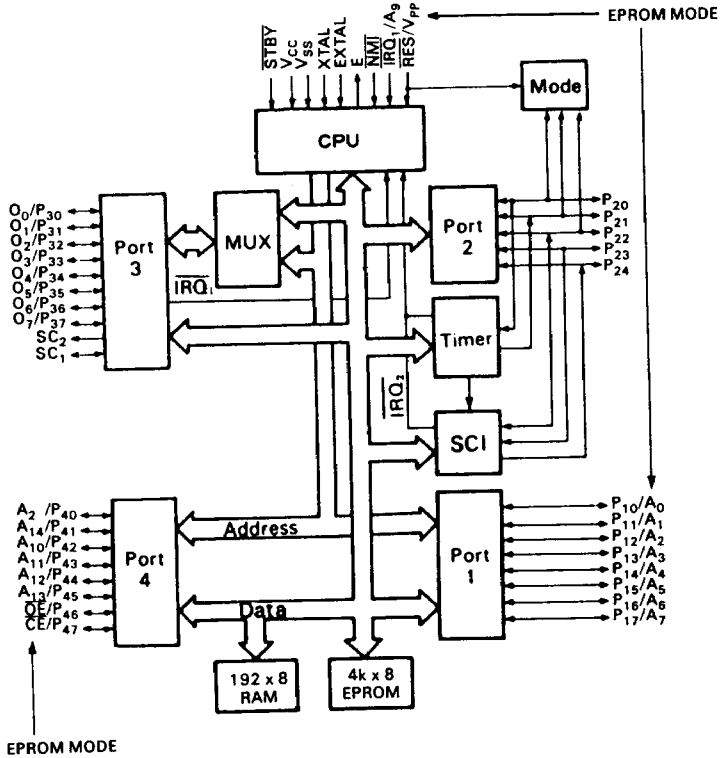


(Top View)

■ PROGRAM DEVELOPMENT SUPPORT TOOLS

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

■ **BLOCK DIAGRAM**



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 ~ +7.0	V
Input Voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	−55 ~ +125	°C
Program Voltage	V_{PP}	−0.3 ~ +13.0	V

(NOTE) This product has protection circuits in input pin 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}: $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$.

■ MCU ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V ± 10%, f = 0.1 ~ 2.0 MHz, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min.	typ.	max.	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}−0.5	—	V_{CC} +0.3	V
	EXTAL			V_{CC} × 0.7			
	Other Inputs			2.0	—		
Input "Low" Voltage	All inputs	V_{IL}		−0.3	—	0.8	V
Input Leakage Current	$\overline{RES}$	$\|I_{in}\|$	V_{in} = 0.5 ~ V_{CC}−0.5V	—	—	10.0	μA
	$\overline{NMI}$, $\overline{IRQ}$, $\overline{STBY}$					1.0	
Three State (off-state) Leakage Current	$P_{10} \sim P_{17}$, $P_{20} \sim P_{24}$, $P_{30} \sim P_{37}$, $P_{40} \sim P_{47}$, $\overline{IS3}$	$\|I_{TSI}\|$	V_{in} = 0.5 ~ V_{CC}−0.5V	—	—	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = −200μA	2.4	—	—	V
			I_{OH} = −10μA	V_{CC}−0.7	—	—	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	—	—	0.55	V
Input Capacitance	$\overline{RES}$	C_{in}	V_{in} = OV, f = 1.0MHz, Ta = 25°C			50	pF
	All Outputs			—	—	12.5	
Standby Current	Non Operation	I_{CC}		—	2.0	15.0	μA
Current Dissipation*		I_{CC}	Operating (f = 1MHz**)	—	5.0	10.0	mA
			Sleeping (f = 1MHz**)	—	1.0	2.0	
RAM Standby Voltage		V_{RAM}		2.0	—	—	V

* V_{IH} min. = V_{CC}−0.8V, V_{IL} max. = 0.8V (All output pins have 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 formulas:

typ. value (f = X MHz) = typ. value (f = 1MHz) × X

max. value (f = X MHz) = max. value (f = 1MHz) × X

(both the sleeping and operating)

● **AC CHARACTERISTICS (V_{CC} = 5.0V ± 10%, f = 0.1 ~ 2.0 MHz, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)**

BUS TIMING

Item		Symbol	Test Condition	HD63701V0			HD637A01V0			HD637B01V0			Unit
				min.	typ.	max.	min.	typ.	max.	min.	typ.	max.	
Cycle Time		t_{cyc}	Fig. 1 Fig. 2	1		10	0.666		10	0.5		10	μs
Address Strobe Pulse Width "High"		PW_{ASH}*		220		—	150		—	110		—	ns
Address Strobe Rise Time		t_{ASr}		—		25	—		25	—		25	ns
Address Strobe Fall Time		t_{ASf}		—		25	—		25	—		25	ns
Address Strobe Delay Time		t_{ASD}*		60		—	40		—	20		—	ns
Enable Rise Time		t_{Er}		—		20	—		20	—		20	ns
Enable Fall Time		t_{Ef}*		—		20	—		20	—		20	ns
Enable Pulse Width "High" Level		PW_{EH}*		450		—	300		—	220		—	ns
Enable Pulse Width "Low" Level		PW_{EL}*		450		—	300		—	220		—	ns
Address Strobe to Enable Delay Time		t_{ASED}*		60		—	40		—	20		—	ns
Address Delay Time		t_{AD1}*		—		250	—		190	—		160	ns
		t_{AD2}*		—		250	—		190	—		160	ns
Address Delay Time for Latch		t_{ADL}*		—		250	—		190	—		160	ns
Data Set-up Time	Write	t_{DSW}*		230		—	150		—	100		—	ns
	Read	t_{DSR}		80		—	60		—	50		—	ns
Data Hold Time	Read	t_{HR}		0		—	0		—	0		—	ns
	Write	t_{HW}*		60		—	40		—	30		—	ns
Address Set-up Time for Latch		t_{ASL}*		60		—	40		—	20		—	ns
Address Hold Time for Latch		t_{ASH}*		30		—	20		—	20		—	ns
Address Hold Time		t_{AH}*		60		—	40		—	30		—	
A_0 ~ A_7 Set-up Time Before E		t_{ASM}*		200		—	110		—	60		—	ns
Peripheral Read Access Time	Non-Multiplexed Bus	T_{ACCN}*		—		650	—		395	—		270	ns
	Multiplexed Bus	T_{ACCM}*		—		650	—		395	—		270	ns
Oscillator Stabilization Time		t_{RC}	Fig. 10	20		—	20		—	20		—	ms
Processor Control Set-up Time		t_{PCS}	Fig. 11	200		—	200		—	200		—	ns

* These timings change depend on t_{cyc}. The values in the table are those when t_{cyc} is minimum (at maximum operating frequency).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD63701V0			HD637A01V0			HD637B01V0			Unit
				min.	typ.	max.	min.	typ.	max.	min.	typ.	max.	
Peripheral Data Set-up Time	Port 1, 2, 3, 4	t_{PDSU}	Fig. 3	200		—	200		—	200		—	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4	t_{PDH}	Fig. 3	200		—	200		—	200		—	ns
Delay Time, Enable Positive Transition to OS3 Negative Transition		t_{OSD1}	Fig. 5	—		300	—		300	—		300	ns
Delay Time, Enable Positive Transition to OS3 Positive Transition		t_{OSD2}	Fig. 5	—		300	—		300	—		300	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*, 3, 4	t_{PWD}	Fig. 4	—		300	—		300	—		300	ns
Input Strobe Pulse Width		t_{PWIS}	Fig. 6	200		—	200		—	200		—	ns
Input Strobe Rise Time		t_{ISr}	Fig. 6	—		50	—		50	—		50	ns
Input Strobe Fall Time		t_{ISf}	Fig. 6	—		50	—		50	—		50	ns
Input Data Hold Time	Port 3	t_{IH}	Fig. 6	150		—	150		—	150		—	ns
Input Data Set-up Time	Port 3	t_{IS}	Fig. 6	0		—	0		—	0		—	ns

* Except P_{21}.

TIMER, SCI TIMING

Item	Symbol	Test Condition	HD63701V0			HD637A01V0			HD637B01V0			Unit
			min.	typ.	max.	min.	typ.	max.	min.	typ.	max.	
Timer Input Pulse Width	t_{PWT}	Fig. 7	2.0		—	2.0		—	2.0		—	t_{cyc}
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}		—		400	—		400	—		400	ns
SCI Input Clock Cycle	t_{Scyc}		2.0		—	2.0		—	2.0		—	t_{cyc}
SCI Input Clock Pulse Width	PWSCK		0.4		0.6	0.4		0.6	0.4		0.6	t_{Scyc}

MODE PROGRAMMING

Item	Symbol	Test Condition	HD63701V0			HD637A01V0			HD637B01V0			Unit
			min.	typ.	max.	min.	typ.	max.	min.	typ.	max.	
$\overline{RES}$	PW_{RSTL}	Fig. 8	3		—	3		—	3		—	t_{cyc}
Mode Programming Set-up Time	t_{MPS}		2		—	2		—	2		—	t_{cyc}
Mode Programming Hold Time	t_{MPH}		150		—	150		—	150		—	ns

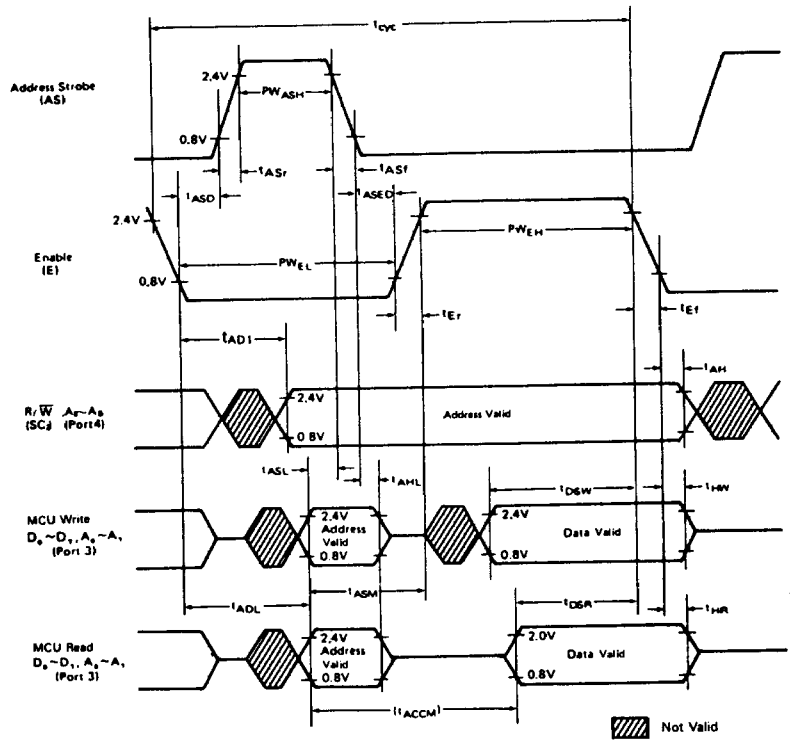


Figure 1 Expanded Multiplexed Bus Timing

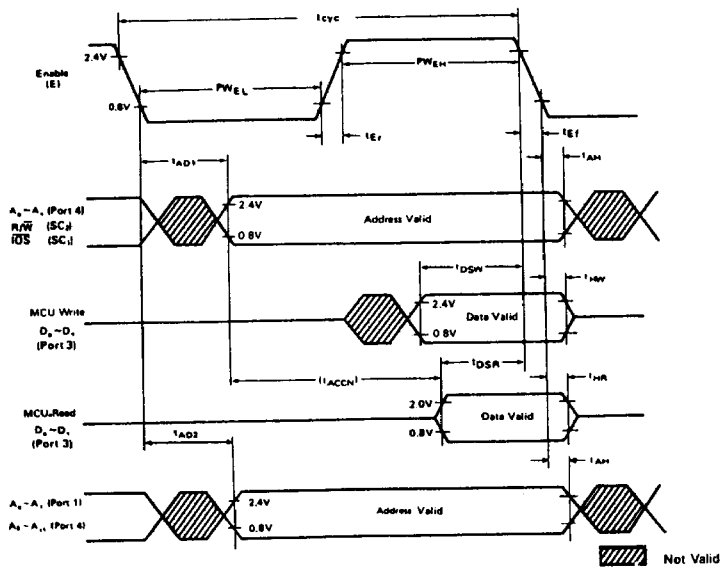
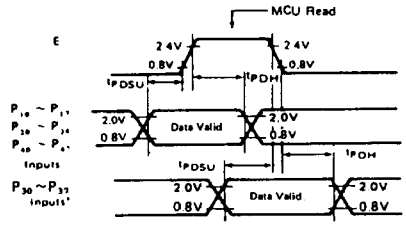
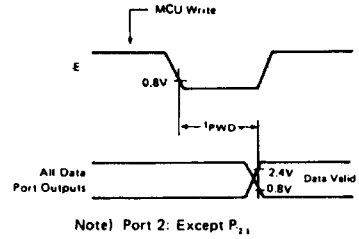


Figure 2 Expanded Non-Multiplexed Bus Timing



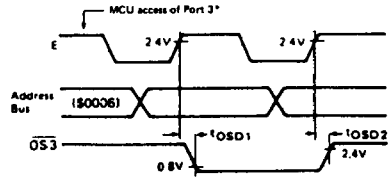
*Port 3 Non-Latched Operation

Figure 3 Port Data Set-up and Hold Times (MCU Read)



Note) Port 2: Except P_{21}

Figure 4 Port Data Delay Times (MCU Write)



*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

Figure 5 Port 3 Output Strobe Timing (Single Chip Mode)

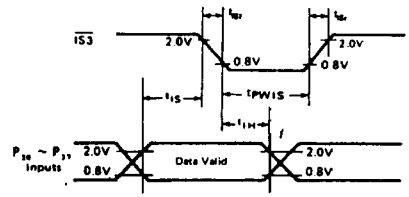


Figure 6 Port 3 Latch Timing (Single Chip Mode)

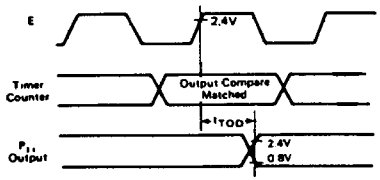


Figure 7 Timer Output Timing

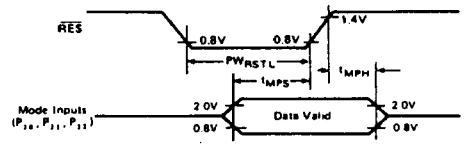
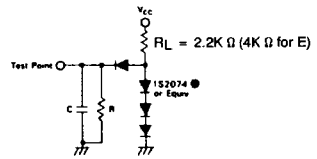


Figure 8 Mode Programming Timing



C = 90pF for P_{10}~P_{17}, P_{30}~P_{37}, P_{40}~P_{47}, SC_1, SC_2, E
= 30pF for P_{20}~P_{24}

R = 12kΩ

Figure 9 Bus Timing Test Loads (TTL Load)

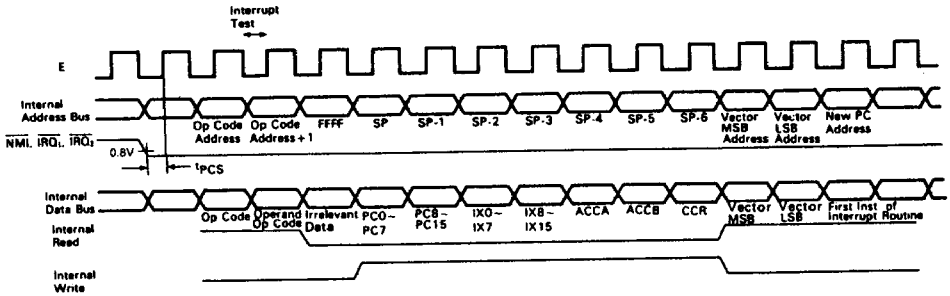


Figure 10 Interrupt Sequence

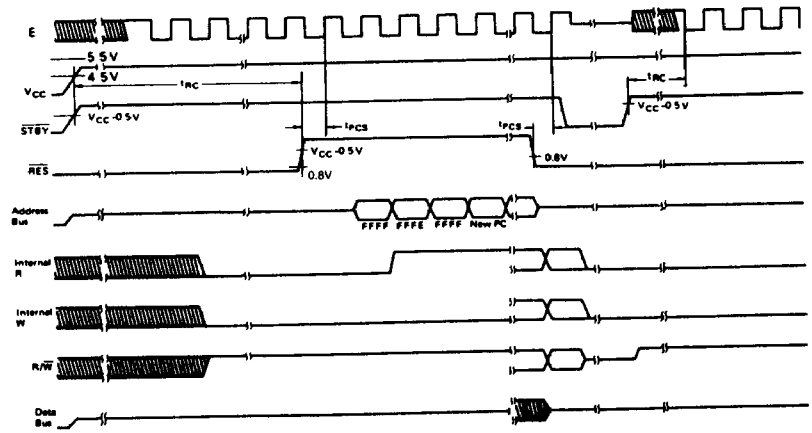


Figure 11 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

These two pins are used for power supply and GND. Recommended power supply voltage is 5V±10%.

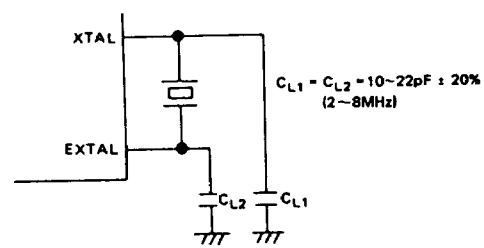
• XTAL, EXTAL

These two pins are connected with parallel resonant fundamental crystal, AT cut. For instance, in order to obtain the system clock 1MHz, a 4MHz resonant fundamental crystal is used because the devide by 4 circuitry is included. EXTAL accepts an external clock input of duty 50% (±5%) to drive, then internal clock is a quarter the frequency of an external clock. External driving frequency will be less than 4 times as maximum internal clock. For external driving, XTAL pin should be open. An example of connection circuit is shown in Fig. 12.

XTAL and EXTAL pins are also used to place the MCU in the EPROM mode. Refer to "PROGRAMMING THE HD63701V0 EPROM" section for details.

AT Cut Parallel Resonance Crystal

Frequency	(MHz)	2.5	4.0	6.0	8.0
Rs max	(Ω)	500	120	80	60
Co max	(pF)	7.0			



(a) Crystal Interface

(b) External Clock

Figure 12 Connection Circuit

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

This pin is used to place the MCU in the Standby mode. If this goes to "Low" level, the oscillation stops, the internal clock is tied to V_{SS} or V_{CC} and the MCU is reset. In order to retain information in RAM during standby, write "0" into RAM enable bit (RAME). RAME is bit 6 of the RAM Control Register at address $0014. This disables the RAM, so the contents of RAM is guaranteed. For details of the standby mode, see the Standby section.

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

This input is used to reset the MCU. $\overline{RES}$ must be held "Low" for at least 20ms when the power starts up.

To reset the MCU during system operation, it must be held "Low" for at least 3 system clock cycles. When $\overline{RES}$ goes "Low", all address buses become "High-impedance" and it continues while $\overline{RES}$ is "Low". If $\overline{RES}$ goes to "High", CPU does the following.

(1) I/O Port 2 bits, 2,1,0 are latched into bits PC2, PC1, PC0 of program control register.
(2) The contents of the two Start Addresses, $FFFE, $FFFF are brought to the program counter, from which program starts (see Table 1).
(3) The interrupt mask bit is set. In order to have the CPU recognize the maskable interrupts $\overline{IRQ_1}$ and $\overline{IRQ_2}$, clear it before those are used.

When $\overline{RES}$ goes "Low", I/O Ports become "High-impedance" immediately independent of clock state, while the internal portions of MCU are reset synchronously with clock.

- **Enable (E)**

This output pin supplies system clock. Output is a single-phase, TTL compatible and 1/4 of the crystal oscillation frequency. It can drive one TTL load and 90pF.

- **Non maskable Interrupt ($\overline{NMI}$)**

When the falling edge of the input signal of this pin is recognized, NMI sequence starts. The current instruction is continued to complete, even if $\overline{NMI}$ signal is detected. Interrupt mask bit in Condition Code Register has no effect on $\overline{NMI}$ detection. In response to NMI interrupt, the information of Program Counter, Index Register, Accumulators, and Condition Code Register are stored on the stack. On completion of this sequence, vectoring address $FFFC and $FFFD are generated to load the contents to the program counter. Then the CPU branch to a non maskable interrupt service routine.

- **Interrupt Request ($\overline{IRQ_1}/A_9$)**

This pin function as IRQ_1 pin in the Normal operation mode, bit in the EPROM mode it functions as address input pin (A_9).

$\overline{IRQ_1}$ is the level-sensitive pin which request an interrupt sequence to the CPU.

When $\overline{IRQ_1}$ goes "Low", the CPU waits until it completes the current instruction that is being executed. Then, if the interrupt mask bit in Condition Code Register is not set, CPU begins interrupt sequence; otherwise, interrupt request is neglected.

Once the sequence has started, the information of Program Counter, Index Register, Accumulators, Condition Code Register are stored on the stack. Then the CPU sets the interrupt mask bit so that no further maskable interrupts may be responded.

At the end of the cycle, the CPU generates 16 bit vectoring addresses indicating memory addresses $FFF8 and $FFF9, and load the contents to the Program Counter, then branch to an interrupt service routine.

The Internal Interrupt will generate signal (IRQ_2) which is quite the same as IRQ_1 except that it will use the vector address $FFF0 to $FFF7.

When $\overline{IRQ_1}$ and $\overline{IRQ_2}$ are generated at the same time, the former precede the latter. Interrupt Mask Bit in the condition code register, if being set, will keep the both interrupts off.

$\overline{IRQ_1}$ has no internal latch. Therefore, if $\overline{IRQ_1}$ is removed during suspension, that $\overline{IRQ_1}$ is ignored.

Table 1 Interrupt Vector Memory Map

	Vector MSB	Vector LSB	Interrupt
Highest Priority	FFFE	FFFF	RESET
	FFEE	FFEF	TRAP
	FFFC	FFFD	NMI
	FFFA	FFFB	SWI (Software Interrupt)
	FFF8	FFF9	IRQ_1 (IS3)
	FFF6	FFF7	ICF (Timer Input Capture)
	FFF4	FFF5	OCF (Timer Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SCI (RDRF+ORFE+TDRE)

On occurrence of Address error or Op-code error, TRAP interrupt is invoked. This interrupt has priority next to RES. Regardless of the Interrupt Mask Bit condition, the CPU will start an interrupt sequence. The vector for this interrupt will be $FFEE, $FFEF.

The following pins are available only in single chip mode.

- **Input Strobe ($\overline{IS3}$) (SC_1)**

This signal controls IS3 interrupt and the latch of Port 3. When the falling edge of this signal is detected, the flag of Port 3 Control Status Register is set.

For detailed explanation of Port 3 Control Status Register, see the I/O PORT 3 CONTROL STATUS REGISTER section.

- **Output Strobe ($\overline{OS3}$) (SC_2)**

This signal is used to send a strobe to an external device, indicating effective data is on the I/O pins. The timing chart for Output Strobe are shown in Figure 5.

The following pins are available for Expanded Modes.

- **Read/Write (R/$\overline{W}$) (SC_2)**

This TTL compatible output signal indicates peripheral and memory devices whether CPU is in Read ("High"), or in Write ("Low"). The normal stand-by state is Read ("High"). It can drive one TTL load and 90pF.

- **I/O Strobe ($\overline{IOS}$) (SC_1)**

In expanded non multiplexed mode 5 of operation, $\overline{IOS}$ goes to "Low" only when A_9 through A_{15} are "0" and A_8 is "1". This allows external access up to 256 addresses from $0100 to $01FF in memory. The timing chart is shown in Figure 2.

- **Address Strobe (AS) (SC_1)**

In the expanded multiplexed mode, address strobe signal appears at this pin. It is used to latch the lower 8 bits addresses multiplexed with data at Port 3. The 8-bit latch is controlled by address strobe as shown in Figure 18. Thereby, I/O Port 3 can become data bus during E pulse. The timing chart of this signal is shown in Figure 1.

Address Strobe (AS) is sent out even if the internal address area is accessed.

■ PORTS

HD63701V0 has four I/O Ports (three 8-bit ports and one 5-bit

port). Each port has an independent write-only data direction register to program individual I/O pins for input or output.*

When the bit of associated Data Direction Register is "1". I/O pin is programmed for output, if "0", then programmed for an input.

There are four ports: Port 1, Port 2, Port 3, and Port 4. Addresses of each port and associated Data Direction Registers are shown in Table 2.

* Only one exception is bit 1 of Port 2 which becomes either a data input or a timer output. It cannot be used as an output port.

RES does not affect I/O port Data Register. Therefore, just after RES, Data Register is uncertain. Data Direction Registers are reset.

Table 2 Port and Data Direction Register Address

Port	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001
I/O Port 3	$0006	$0004
I/O Port 4	$0007	$0005

- **I/O Port 1**

This is an 8-bit port, each bit being defined individually as input or outputs by associated Data Direction Register. The 8-bit output buffers have three-state capability, maintaining in high impedance state when they are used for input.

These are TTL compatible and can drive one TTL load and 90pF. After the MCU has been reset, all I/O lines of Port 1 are configured as inputs in all modes except mode 1.

In all modes except the expanded non multiplexed mode (Mode 1) and EPROM mode, port 1 always function as parallel I/O. In mode 1, port 1 will be an output line for lower address (A_0 to A_7). In the EPROM mode, port 1 is lower address (A_0 to A_7) input for EPROM.

- **I/O Port 2**

This port has five lines, whose I/O direction depends on its data direction register. The 5-bit output buffers have three-state capability, going high impedance state when used as inputs.

Port 2 is TTL compatible and can drive one TTL and 30pF. After the MCU has been reset, I/O lines are configured as inputs. Three pins of Port 2 (pins P_{20}, P_{21}, P_{22}) are used to program the mode of operation during reset. The values of these three pins during reset are latched into the upper 3 bits (bit 7, 6 and 5) of Port 2 Data Register, which is explained in the MODE SELECTION section.

In all modes, Port 2 can be configured as I/O lines. This port also provides access to the Serial I/O and the Timer. However, note that bit 1 (P_{21}) is the only pin restricted to data input or Timer output.

- **I/O Port 3**

This is an 8-bit port which can be configured as I/O lines, a data bus, or an address bus multiplexed with data bus. Its function depends on hardware operation mode programmed by the user using 3 bits of Port 2 during Reset. Port 3 as a data bus is bi-directional. For an input from peripherals, regular TTL level must be supplied, that is greater than 2.0V for a logic "1" and less than 0.8V for a logic "0". This TTL compatible three-state buffer can drive one TTL load and 90pF. In the expanded Modes, data direction register will be inhibited after Reset and data direction will depend on the state of the R/W line. Function of Port 3 is shown below.

Single Chip Mode (Mode 7)

Parallel Inputs/Outputs as programmed by its corresponding Data Direction Register.

There are two control lines associated with this port in this mode, an input strobe ($\overline{IS3}$) and an output strobe ($\overline{OS3}$), both being used for handshaking. They are controlled by I/O Port 3 Control/Status Register. Function of these two control lines of Port 3 are summarized as follows:

(1) Port 3 input data can be latched using $\overline{IS3}$ (SC_1) as a input strobe signal.
(2) $\overline{OS3}$ can be generated by CPU read or write to Port 3's data register.
(3) $\overline{IRQ_1}$ interrupt can be generated by an $\overline{IS3}$ falling edge.

Port 3 strobe and latch timing is shown in Figs. 5 and 6 respectively.

I/O Port 3 Control/Staus Register is explained as follows:

I/O Port 3 Control/Status Register

	7	6	5	4	3	2	1	0
		$\overline{IS3}$	x	OSS	LATCH	x	-x	x
$000F	$\overline{IS3}$ FLAG	$\overline{IS3}$ $\overline{IRQ_1}$ ENABLE			ENABLE			

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 LATCH ENABLE.

Bit 3 is used to control the input latch of Port 3. If the bit is set at "1", the input data on Port 3 is latched by the falling edge of IS3. The latch is released by the MCU read to Port 3; now new data can be latched again by IS3 falling edge. Bit 3 is cleared by a reset. If this bit is "0", IS3 does not affect I/O Port 3 latch operation.

Bit 4 OSS (Output Strobe Select)

This bit identifies the cause of output strobe generation: a write operation or read operation to I/O Port 3. When the bit is cleared, the strobe will be generated by a read operation to Port 3. When the bit is not cleared, the strobe will be generated by a write operation. Bit 4 is cleared by a reset.

Bit 5 Not used.
Bit $\overline{IS3}$ $\overline{IRQ_1}$ ENABLE.

If this bit is set, IRQ_1 interrupt by IS3 Flag is enabled. Otherwise the interrupt is disabled. The bit is cleared by a reset.

Bit 7 $\overline{IS3}$ FLAG.

Bit 7 is a read-only bit which is set by the falling edge of $\overline{IS3}$ (SC_1). It is cleared by a read of the Control/Status Register followed by a read/write of I/O Port 3. The bit is cleared by reset.

Expanded Non Multiplexed Mode (mode 1, 5)

In this mode. Port 3 becomes data bus. (D_0 to D_7)

Expanded Multiplexed Mode (mode 0, 2, 6)

Port 3 becomes both the data bus (D_0 to D_7) and lower bits of the address bus (A_0 to A_7). An address strobe output is "High" while the address is on the port.

EPROM Mode

In this mode, Port 3 is data inputs and outputs of EPROM (O_0 to O_7).

- **I/O Port 4**

This is an 8-bit port that becomes either I/O or address outputs depending on the selected operation mode. Each line is TTL compatible and can drive one TTL load and 90pF. Function of Port 4 for each mode is explained below.

Single Chip Mode (Mode 7): Parallel Inputs/Outputs as programmed by its associated data direction register.

Expanded Non Multiplexed Mode (Mode 5): In this mode, Port 4 becomes the lower address line (A_0 to A_7) by writing "1"s on the data direction register. After reset, this port becomes inputs. In order to use these pins as addresses, they should be programmed as outputs.

When all of the eight bits are not required as addresss, the remaining lines can be used as I/O lines (Inputs only).

Expanded Non Multiplexed Mode (Mode 1): In this mode, Port 4 becomes output for upper order address lines (A_8 to A_{15}) regardless of the value of the direction register.

Expanded Multiplexed Mode (Mode 6): In this mode, Port 4 becomes the upper address lines (A_8 to A_{15}). After reset, this port becomes inputs. In order to use these pins as addresses, they should be programmed as outputs. When all of the eight bits are not required, the remaining lines can be used as I/O lines (input only).
Expanded Multiplexed Mode (Mode 0, 2): In this mode, Port 4 becomes output for upper order address lines (A_8 to A_{15}) regardless of the value of data direction register.

EPROM Mode

In this mode, Port 4 functions as an input line for EPROM address, $\overline{CE}$ and $\overline{OE}$.

The relation between each mode and I/O Port 1 to 4 is summarized in Table 3.

■ MODE SELECTION

The MCU operation mode after the reset must be determined by the user wiring the P_{20}, P_{21} and P_{22} pins externally. These three pins are lower order bits; I/O 0, I/O 1, I/O 2 of Port 2. They are latched into the control bits PC0, PC1, PC2 of I/O Port 2 register when reset goes "High". I/O Port 2 Register is shown below.

Port 2 DATA REGISTER

	7	6	5	4	3	2	1	0
$0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0

An example of external hardware used for Mode Selection is shown in Fig. 13. The HD14053B is used to separate the peripheral device from the MCU during reset. It is necessary if the data may conflict between peripheral device and Mode generation circuit.

No mode can be changed through software because the bits 5, 6, and 7 of Port 2 Data Register are read-only. The mode selection of the HD63701V0 is shown in Table 4.

The HD63701V0 operates in three basic modes: (1) Single Chip Mode; (2) Expanded Multiplexed Mode (compatible with the HMCS6800 peripheral family), (3) Expanded Non Multiplexed Mode (compatible with HMCS6800 peripheral family).

For EPROM mode, refer to "Programming the HD63701V0 EPROM" section.

- **Single Chip Mode (Mode 7)**

In the Single Chip Mode, all ports will become I/O. This is shown in Figure 15. In this mode, SC_1, SC_2 pins are configured for control lines of Port 3 and can be used as input strobe ($\overline{IS3}$) and output strobe ($\overline{OS3}$) for data handshaking.

- **Expanded Multiplexed Mode (Mode 0, 2, 6)**

In this mode, Port 4 is configured for I/O (inputs only) or address lines. The ata bus and the lower order address bus are multiplexed in Port 3 and can be separated by the Address Strobe.

Port 2 is configured for 5 parallel I/O or Serial I/O, or Timer, or any combination thereof. Port 1 is configured for 8 parallel I/O. In this mode, HD63701V0 is expandable up to 65k words (See Fig. 16).

- **Expanded Non Multiplexed Mode (Mode 1, 5)**

In this mode, the HD63701V0 can directly address HMCS6800 peripherals without address latch. In mode 5, Port 3 becomes a data bus. Port 4 becomes A_0 to A_7 address bus or partial address bus and I/O (inputs only). Port 2 is configured for a parallel I/O, Serial I/O, Timer or any combination thereof.

Port 1 is configured as a parallel I/O only.

In this mode, HD63701V0 is expandable to 256 locations.

In mode 1, Port 3 becomes a data bus and Port 1 becomes A_0 to A_7 address bus, and Port 4 becomes A_8 to A_{15} address bus.

In this mode, the HD63701V0 is expandable to 65k word with no address latch. (See Fig. 17)

- **Lower Order Address Bus Latch**

Because the data bus is multiplexed with the lower order address bus in Port 3 in the expanded multiplexed mode, address bits must be latched. It requires the 74LS373 Transparent octal D-type to latch the LSB. Latch connection of the HD63701V0 is shown in Figure 18.

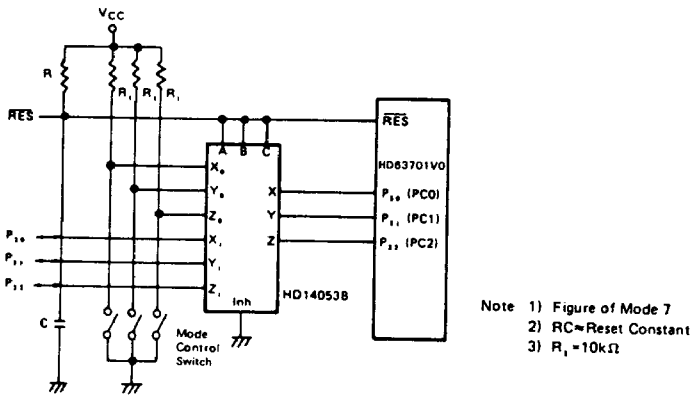
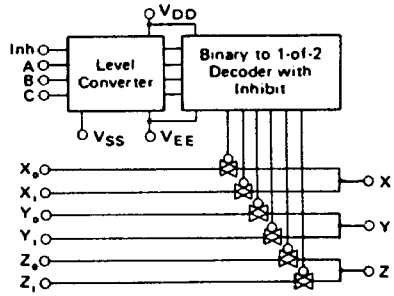


Figure 13 Recommended Circuit for Mode Selection

HITACHI
Hitachi America, Ltd. • Hitachi Plaza • 2000 Sierra Point Pkwy. • Brisbane, CA 94005-1819 • (415) 589-8300



Truth Table

Control Input				On Switch		
Inhibit	Select					
	C	B	A	HD14053B		
0	0	0	0	Z_0	Y_0	X_0
0	0	0	1	Z_0	Y_0	X_1
0	0	1	0	Z_0	Y_1	X_0
0	0	1	1	Z_0	Y_1	X_1
0	1	0	0	Z_1	Y_0	X_0
0	1	0	1	Z_1	Y_0	X_1
0	1	1	0	Z_1	Y_1	X_0
0	1	1	1	Z_1	Y_1	X_1
1	X	X	X	—		

Figure 14 HD14053B Multiplexers/De-Multiplexers

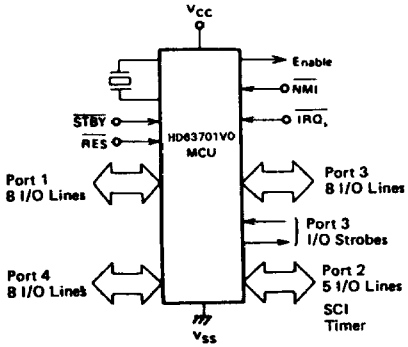


Figure 15 HD63701V0 MCU Single-Chip Mode

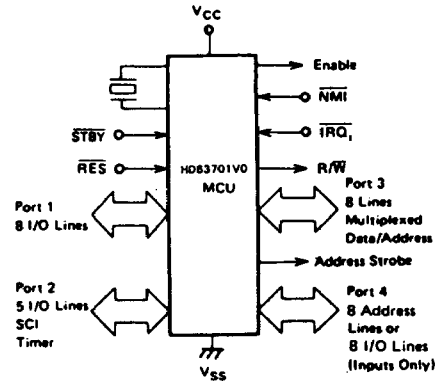
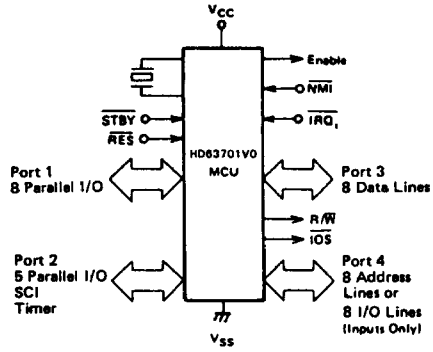
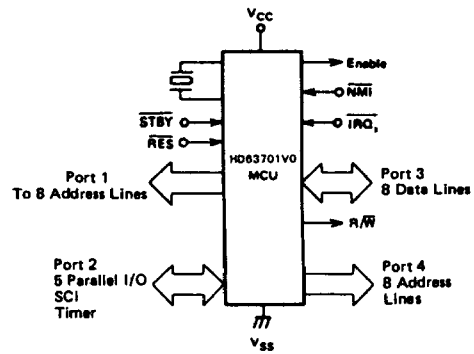


Figure 16 HD63701V0 MCU Expanded Multiplexed Mode

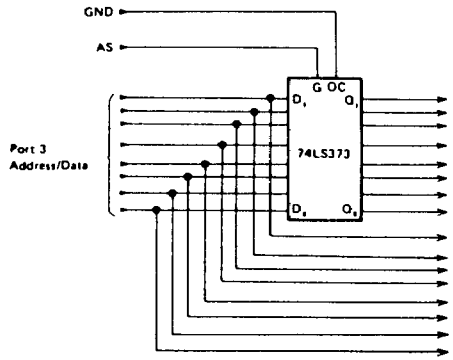


(a) Mode 5



(b) Mode 1

Figure 17 HD63701V0 MCU Expanded Non-Multiplexed Mode



Function Table

Output Control (OC)	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 18 Latch Connection

- **Summary of Mode and MCU Signal**

This section gives a description of the MCU signals for the various modes. SC_1 and SC_2 are signals which vary while the mode.

Table 3 Feature of each mode and lines

MODE			PORT 1 Eight Lines	PORT 2 Five Lines	PORT 3 Eight Lines	PORT 4 Eight Lines	SC_1	SC_2	$\overline{RES}/V_{PP}$	IRQ_1/A_9
MCU Mode	Single Chip		I/O	I/O	I/O	I/O	$\overline{IS3}$ (I)	$\overline{OS3}$ (O)	$\overline{RES}$	$\overline{IRQ_1}$
	Expanded Mux		I/O	I/O	Address Bus (A_0~A_7 Data Bus (D_0~D_7)	Address Bus* (A_8~A_{15})	AS (O)	R/$\overline{W}$ (O)	$\overline{RES}$	$\overline{IRQ_1}$
	Expanded Non-Mux	Mode 5	I/O	I/O	Data Bus (D_0~D_7)	Address Bus* (A_0~A_7)	$\overline{IOS}$ (O)	R/$\overline{W}$ (O)	$\overline{RES}$	$\overline{IRQ_1}$
		Mode 1	Address Bus (A_0~A_7)	I/O	Data Bus (D_0~D_7)	Address Bus (A_8~A_{15})	Not Used	R/$\overline{W}$ (O)	$\overline{RES}$	$\overline{IRQ_1}$
EPROM Mode			Address Bus (A_0~A_7)	Mode Input (P_{20}, P_{21}, P_{22})	Data Bus (D_0~D_7)	Address Bus $\overline{OE}$, $\overline{CE}$ (A_8, A_{10}~A_{14})	Not Used	Not Used	V_{PP}	A_9

* These lines can be substituted for I/O (Input Only) (except Mode 0, 2)

I = Input $\overline{IS3}$ = Input Strobe SC = Strobe Control
O = Output $\overline{OS3}$ = Output Strobe AS = Address Strobe
R/$\overline{W}$ = Read/Write $\overline{IOS}$ = I/O Select

Table 4 Mode Selection Summary

Mode	P_{12} (PC2)	P_{11} (PC1)	P_{10} (PC0)	EPROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(3)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(3)	Non-Multiplexed/Partial Decode
4	H	L	L	–	–	–	–	Not Used
3	L	H	H	–	–	–	–	Not Used
2	L	H	L	E(1)	I	E	MUX	Multiplexed
1	L	L	H	E(1)	I	E	NMUX	Non-Multiplexed
0	L	L	L	I	I	I(2)	MUX	Multiplexed Test

LEGEND:
I – Internal
E – External
MUX – Multiplexed
NMUX – Non-Multiplexed
L – Logic "0"
H – Logic "1"

(NOTES)
1) Internal ROM is disabled.
2) Reset vector is external for 3 or 4 cycles after RES goes "high".
3) Idle lines of Port 4 address outputs can be assigned to Input Port.

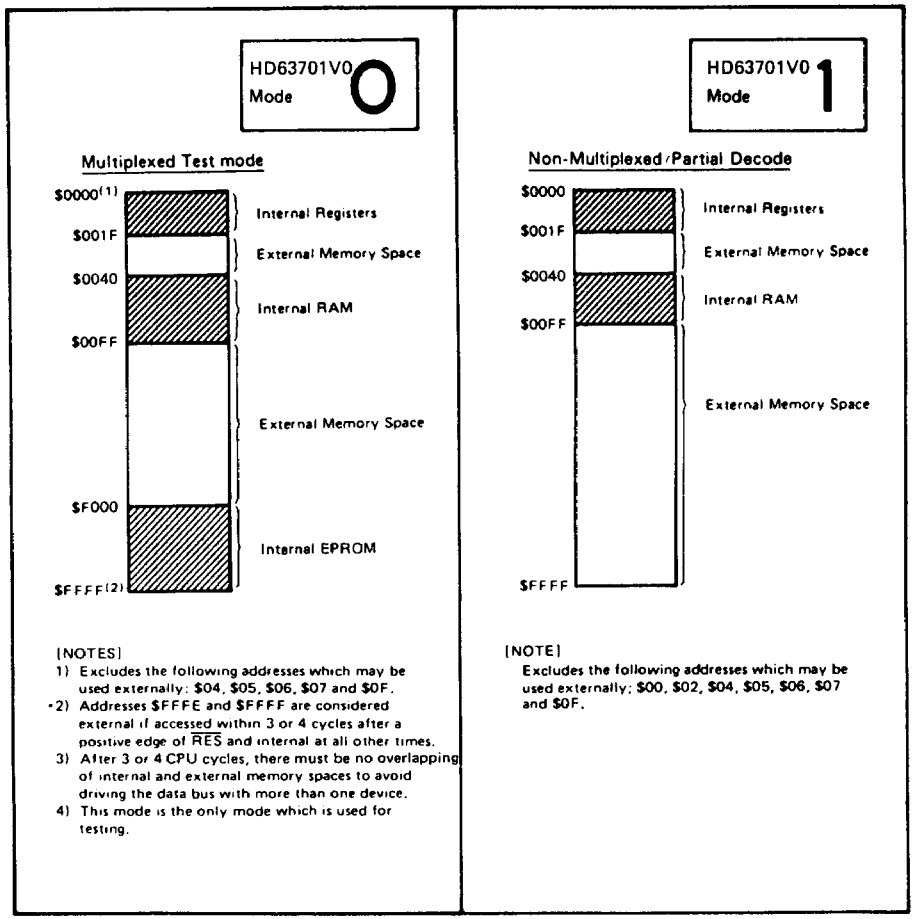
■ MEMORY MAP

The MCU can provide up to 65k byte address space depending on the operating mode. Fig. 19 shows a memory map for each operating mode. The first 32 locations of each map are for the MCU's internal register only, as shown in Table 5.

Table 5 Internal Register

Register	Address	Read, Write*4/Initial Value after Reset							
		7	6	5	4	3	2	1	0
Port 1 Data Direction Register	$00*1	W							
		$00							
Port 2 Data Direction Register	$01	W							
		$00							
Port 1 Data Register	$02*1	R/W*5							
		Uncertain							
Port 2 Data Register/Mode Register	$03	R*6			R/W*5				
		P_{22}	P_{21}	P_{20}	Uncertain				
Port 3 Data Direction Register	$04*2	W							
		$00							
Port 4 Data Direction Register	$05*3	W							
		$00							
Port 3 Data Register	$06*2	R/W*5							
		Uncertain							
Port 4 Data Register	$07*3	R/W*5							
		Uncertain							
Timer Control and Status Register	$08	R	R	R	R/W	R/W	R/W	R/W	R/W
		0	0	0	0	0	0	0	0
Counter (High Byte)	$09	R/W							
		$00							
Counter (Low Byte)	$0A	R/W							
		$00							
Output Compare Register (High Byte)	$0B	R/W							
		$FF							
Output Compare Register (Low Byte)	$0C	R/W							
		$FF							
Input Capture Register (High Byte)	$0D	R							
		$00							
Input Capture Register (Low Byte)	$0E	R							
		$00							
Port 3 Control and Status Register	$0F*2	R	R/W	Not Used	R/W	R/W	Not Used		
		0	0	1	0	0	1	1	1
Rate and Mode Control Register	$10	Not used				W	W	W	W
		1	1	1	1	0	0	0	0
Transmit/Receive Control and Status Register	$11	R	R	R	R/W	R/W	R/W	R/W	R/W
		0	0	1	0	0	0	0	0
Receive Data Register	$12	R							
		$00							
Transmit Data Register	$13	W							
		$00							
RAM Control Register	$14	R/W	R/W	Not Used					
		*7	1	1	1	1	1	1	1
Reserved	$15~$1F								

*1 External address in mode 1.
*2 External address in mode 0, 1, 2, 6. Cannot be accesed in mode 5.
*3 External address in mode 0, 1, 2.
*4 R: Read only, W: Write only, R/W: Read and write.
*5 When the CPU reads these addresses, the data on I/O pins, not the contents of data registers, are read into the CPU. (Regarding Port 3, refer to "I/O Port 3" section for detail.)
*6 Refer to "MODE SELECTION" section.
*7 Refer to "Standby Mode" section.



(to be continued)

Figure 19 HD63701V0 Memory Maps

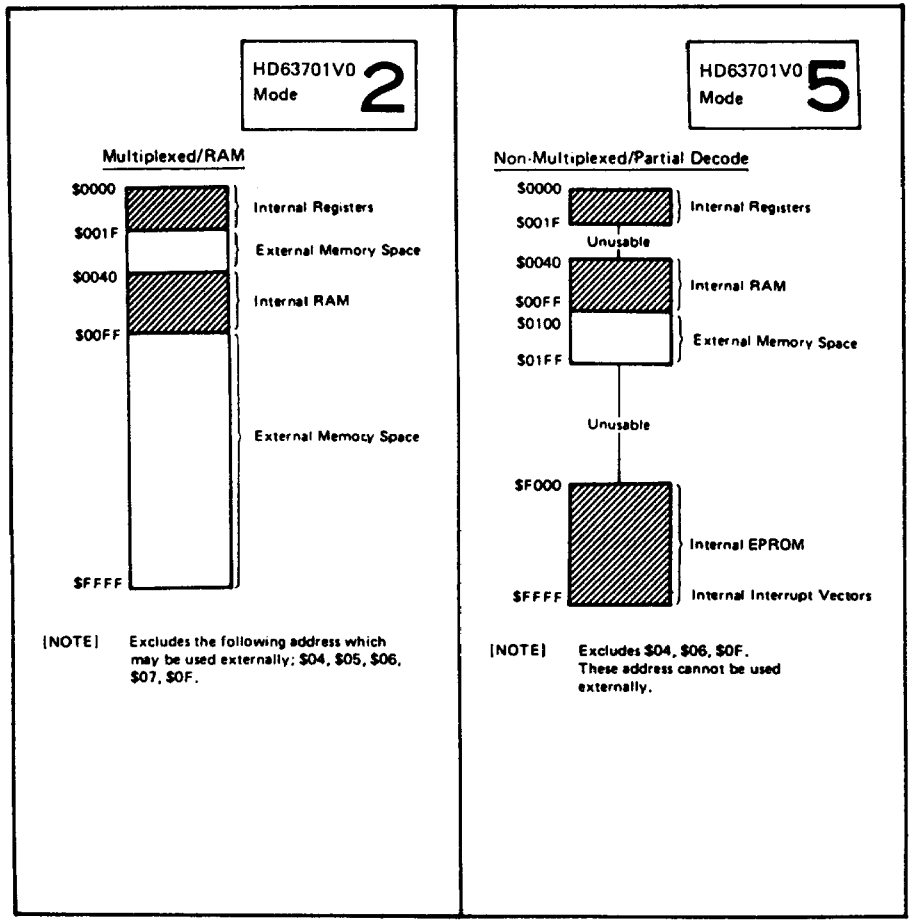


Figure 19 HD63701V0 Memory Maps

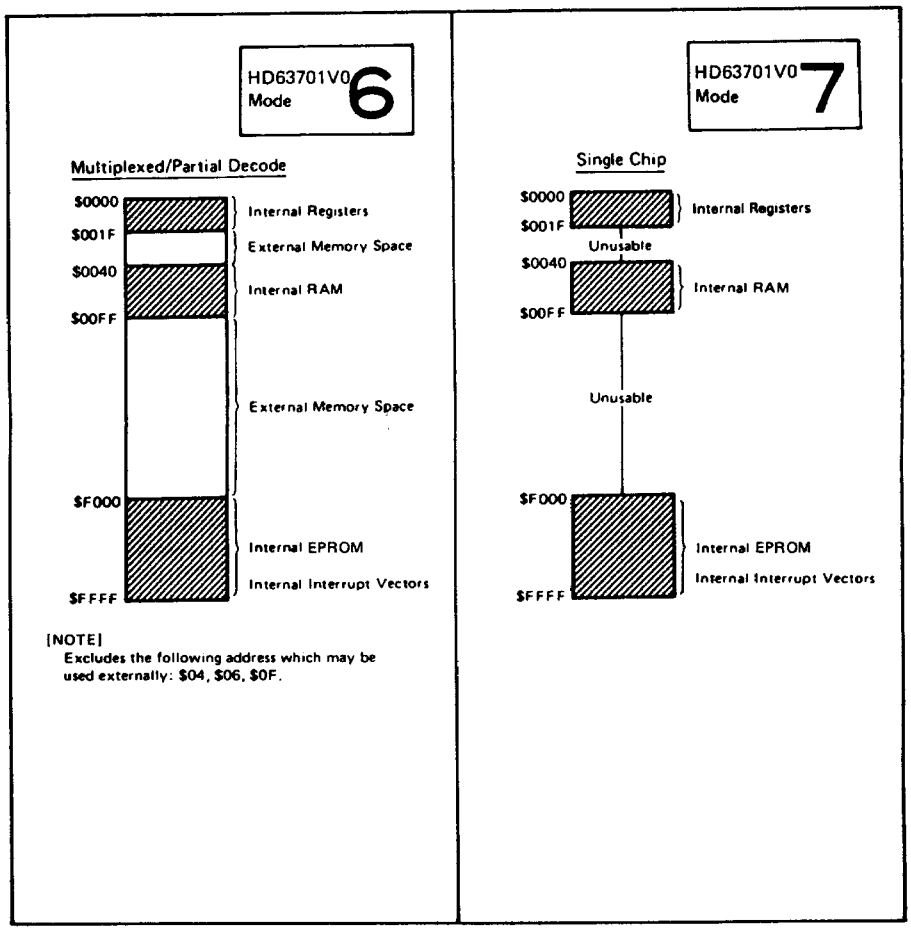


Figure 19 HD63701V0 Memory Maps

■ PROGRAMMABLE TIMER

The HD63701V0 contains 16-bit programmable timer which may be used to make measurement of input waveform. In addition to that it can generate an output waveform by itself. For both input and output waveform, the pulse width may vary from a few microseconds to several seconds.

- an 8-bit control and status register
- a 16-bit free running counter
- a 16-bit output compare register, and
- a 16-bit input capture register

A block diagram of the timer is shown in Figure 20.

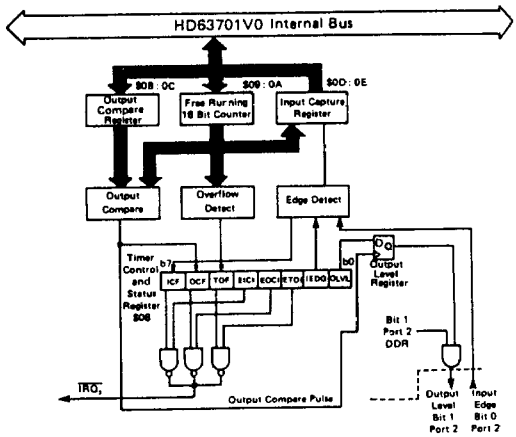


Figure 20 Programmable Timer Block Diagram

• Free Running Counter ($0009:$000A)

The key element in the programmable timer is a 16-bit free running counter, that is driven by an E (Enable) clock to increment its values. The counter value will be read out by the CPU software at any time with no effects on the counter. Reset will clear the counter.

When the MSB of this counter is read, the LSB is stored in temporary latch. The data is fetched from this latch by the subsequent read of LSB. Thus consistent double byte data can be read from the counter.

When the CPU writes arbitrary data to the MSB ($09), the value of $FFF8 is being pre-set to the counter ($09, $0A) regardless of the write data value. Then the CPU writes arbitrary data to the LSB ($0A), the data is set to the "Low" byte of the counter, at the same time, the data preceedingly written in the MSB ($09) is set to "High" byte of the counter.

When the data is written to this counter, a double byte store instruction (ex. STD) must be used. If only the MSB of counter is written, the counter is set to $FFF8.

The counter value written to the counter using the double byte store instruction is shown in Figure 21.

To write to the counter may disturb serial operations, so it should be inhibited during using the SCI in internal clock mode.

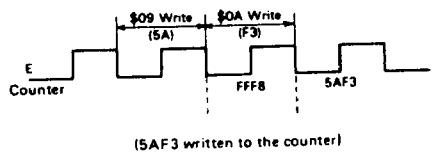


Figure 21 Counter Write Timing

• Output Compare Register ($000B:$000C)

This is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly being compared with current value of the free running counter.

When the contents match the value of the free running counter, a flag (OCF) in the timer control/status register (TCSR) is set and the current value of an output level Bit (OLVL) in the TCSR is transferred to Port 2 bit 1. When bit 1 of the Port 2 data direction register is "1" (output), the OLVL value will appear on the bit 1 of Port 2. Then, the value of output Compare Register and Output level bit may be changed for the next compare.

The output compare register is set to $FFFF during reset.

The compare function is inhibited at the cycle of writing to the high byte of the output compare register and at the cycle just after that to ensure valid compare. It is also inhibited in same manner at writing to the free running counter.

• Input Capture Register ($000D:$000E)

The input capture register is a 16-bit read-only register used to hold the current value of free running counter when the proper transition of an external input signal occurs.

The input transition change required to trigger the counter transfer is controlled by the input edge bit (IEDG).

To allow the external input signal to go in the edge detect unit, the bit of the Data Direction Register corresponding to bit 0 of Port 2 must have been cleared (to zero).

To insure input capture in all cases, the width of an input pulse requires at least 2 Enable cycles.

• Timer Control/Status Register (TCSR) ($0008)

This is an 8-bit register. All 8 bits are readable and the lower 5-bit may be written. The upper 3 bits are read-only, indicating the timer status information as is shown below.

(1) A proper transition has been detected on the input pin (ICF).
(2) A match has been found between the value in the free running counter and the output compare register (OCF).
(3) When counting up to $0000 (TOF).

Each flag has an individual enable bit in TCSR which determines whether or not an interrupt request may occur (IRQ_2). If the I-bit in Condition Code Register has been cleared, a priority vectored address occurs correspondinging to each flag. A description of each bit is as follows.

Timer Control/Status Register

7	6	5	4	3	2	1		
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

Bit 0 **OLVL (Output Level)**; When a match is found in the value between the counter and the output compare register, this bit is transferred to the Port 2 bit 1. If the DDR corresponding to Port 2 bit 1 is set "1", the value will appear on the output pin of Port 2 bit 1.

Bit 1 **IEDG (Input Edge)**; This bit control which transition of an input of Port 2 bit 0 will trigger the data transfer from the counter to the input capture register. The DDR corresponding to Port 2 bit 0 must be cleared in advance of using this function.
When IEDG = 0, trigger takes place on a negative edge ("High" to "Low" transition). When IEDG = 1, trigger takes place on a positive edge ("Low" to "High" transition).

Bit 2 **ETOI (Enable Timer Overflow Interrupt)**; When set, this bit enables TOF interrupt to generate the interrupt request (IRQ_2). When cleared, the interrupt is inhibited.

Bit 3 **EOCI (Enable Output Compare Interrupt)**; When set, this bit enables OCF interrupt to generate the interrupt request (IRQ_2). When cleared, the interrupt is

inhibited.

Bit 4 EICI (Enable Input Capture Interrupt); When set, this bit enables ICF interrupt to generate the interrupt request (IRQ_2). When cleared, the interrupt is inhibited.

Bit 5 TOF (Timer Over Flow Flag); This read-only bit is set at the transition of $FFFF to $0000 of the counter. It is cleared by CPU read of TCSR (with TOF set) followed by an CPU read of the counter ($0009).

Bit 6 OCF (Output Compare Flag); This read-only bit is set when a match is found in the value between the output compare register and the counter. It is cleared by a read of TCSR (with OCF set) followed by an CPU write to the output compare register ($000B or $000C).

Bit 7 ICF (Input Capture Flag); The read-only bit is set by a proper transition on the input, and is cleared by a read of TCSR (with ICF set) followed by an CPU read of Input Capture Register ($000D).

Reset will clear each bit of Timer Control and Status Register.

■ SERIAL COMMUNICATION INTERFACE

The HD63701V0 contains a full-duplex asynchronous Serial Communication Interface (SCI). SCI may select the several kinds of the data rate. It consists of a transmitter and a receiver which operate independently but with the same data format and the same data rate. Both the transmitter and receiver communicate with the CPU via the data bus and with the outside world through Port 2 bit 2, 3 and 4. Description of hardware, software and register is as follows.

• Wake-Up Feature

In typical multiprocessor applications the software protocol will usually have the designated address at the initial byte of the message. The purpose of Wake-Up feature is to have the non-selected MCU neglect the remainder of the message. Thus the non-selected MCU can inhibit the all further interrupt process until the next message begins.

Wake-Up feature is re-enabled by a ten consecutive "1"s which indicates an idle transmit line. Therefore software protocol must put an idle period between the messages and must prevent it within the message.

With this hardware feature, the non-selected MCU is reenabled or ("wake-up") by the next message.

• Programmable Options

The HD63701V0 has the following programmable features.

* data format; standard mark/space (NRZ)
* clock source; external or internal
* baud rate; one of 4 rates per given E clock frequency or 1/8 of external clock
* wake-up feature; enabled or disabled
* interrupt requests; enabled or masked individually for transmitter and receiver
* clock output; internal clock enabled or disabled to Port 2 bit 2
* Port 2 (bits 3, 4); dedicated or not dedicated to serial I/O individually

• Serial Communication Hardware

The serial communications hardware is controlled by 4 registers as shown in Figure 22. The registers include:

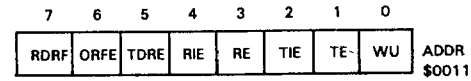
* an 8-bit control/status register
* a 4-bit rate/mode control register (write-only)
* an 8-bit read-only receive data register
* an 8-bit write-only transmit data register

Besides these 4 registers, SCI utilizes Port 2 bit 3 (input) and bit 4 (output). Port 2 bit 2 can be used when an option is selected for the internal-clock-out or the external-clock-in.

• Transmit/Receive Control Status Register (TRCSR)

TRCS Register consists of 8 bits which all may be read while only bits 0 to 4 may be written. The register are explained below.

Transmit/Receive Control Status Register



Bit 0 WU (Wake Up); Set by software and cleared by hardware on receipt of ten consecutive "1"s. While this bit is "1", RDRF and ORFE flags are not set even if data are received or errors are detected. Therefore received data are ignored. It should be noted that RE flag must have already been set in advance of WU flag's set.

Bit 1 TE (Transmit Enable); This bit enables transmitter. When this bit is set, bit 4 of Port 2 DDR is also forced to be set. It remains set even if TE is cleared. Preamble of ten consecutive "1"s is transmitted just after this bit is set, and then transmitter becomes ready to send data.

If this bit is cleared, the transmitter is disabled and SCI affects nothing on Port 2 bit 4.

Bit 2 TIE (Transmit Interrupt Enable); When this bit is set, TDRE (bit 5) causes an IRQ_2 interrupt. When cleared TDRE interrupt is masked.

Bit 3 RE (Receive Enable); When set, Port 2 bit 3 can be used as an input of receive regardless of DDR value for this bit. When cleared, the receiver is disabled.

Bit 4 RIE (Receive Interrupt Enable); When this bit is set, RDRF (bit 7) or ORFE (bit 6) cause an IRQ_2 interrupt. When cleared, this interrupt is masked.

Bit 5 TDRE (Transmit Data Register Empty); When the data is transferred from the Transmit Data Register to Output Shift Register, this bit is set by hardware. The bit is cleared by reading the status register followed by writing the next new data into the Transmit Data Register. TDRE is initialized to 1 by RES.

Bit 6 ORFE (Over Run Framing Error); When overrun or framing error occurs (receive only), this bit is set by hardware. Over Run Error occurs if the attempt is made to transfer the new byte to the receive data register while the RDRF is "1" Framing Error occurs when the bit counter is not synchronized with the boundary of the byte in the receiving bit stream. When Framing Error is detected, RDRF is not set. Therefore Framing Error can be distinguished from Overrun Error. That is, if ORFE is "1" and RDRF is "1", Overrun Error is detected. Otherwise Framing Error occurs. The bit is cleared by reading the status register followed by reading the receive data register, or by Reset.

When overrun error occurs, the received data is not transferred to the RDR, but when framing error occurs, it is transferred.

Bit 7 RDRF (Receive Data Register Full); This bit is set by hardware when the data is transferred from the receive shift register to the receive data register. It is cleared by reading the status register followed by reading the receive data register, or by Reset.

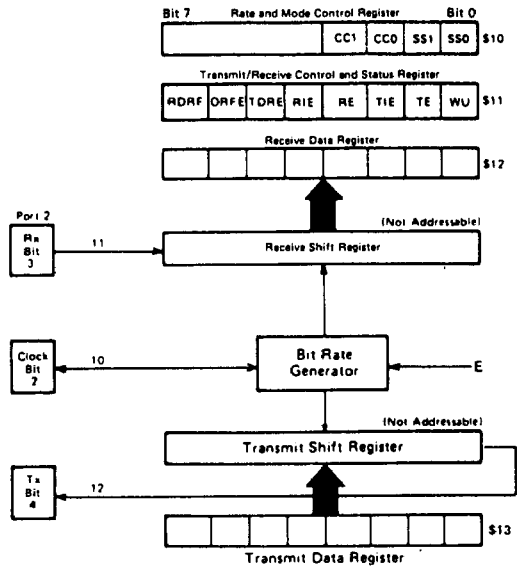


Figure 22 Serial I/O Register

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	ADDR $0010

Transfer Rate / Mode Control Register

Table 6 SCI Bit Times and Transfer Rates

	XTAL	2.4576 MHz	4.0 MHz	4.9152MHz
SS1 : SS0	E	614.4 kHz	1.0 MHz	1.2288MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13 μs/76,800Baud
0 1	E ÷ 128	208μs/4,800 Baud	128 μs/7812.5 Baud	104.2μs/ 9,600Baud
1 0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud	833.3μs/ 1,200Baud
1 1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud	3.333ms/ 300Baud

Table 7 SCI Format and Clock Source Control

CC1: CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	–	–
0 1	NRZ	Internal	Not Used***	**	**
1 0	NRZ	Internal	Output*	**	**
1 1	NRZ	External	Input	**	**

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS.
Bit 4 is used for serial output if TE = "1" in TRCS.
*** This pin can be used as I/O port.

- **Transfer rate/Mode Control Register (RMCR)**
  * Baud rate * data format * clock source
  * Port 2 bit 2 feature

It is 4-bit write-only register, cleared by Reset. The 4 bits are considered as a pair of 2-bit fields. The lower 2 bits control the bit rate of internal clock while the upper 2 bits control the format and the clock select logic.

Bit 0 SS0
Bit 1 SS1 — Speed Select

These bits select the Baud rate for the internal clock. The rates selectable are function of E clock frequency of the CPU. Table 6 lists the available Baud Rates.

Bit 2 CC0
Bit 3 CC9 — Clock Control/Format Select

They control the data format and the clock select logic. Table 7 defines the bit field.

- **Internal Generated Clock**

If the user wish to use externally an internal clock of the SCI, the following requirements should be noted.

* CC1, CC0 must be set to "10".
* The maximum clock rate must be E/16.
* The clock rate is equal to the bit rate.
* The values of RE and TE have no effect.

- **Externally Generated Clock**

If the user wish to supply an external clock to the SCI, the following requirements should be noted.

* The CC1, CC0 must be set to "11" (See Table 7).
* The external clock must be set to 8 times of the desired baud rate.
* The maximum external clock frequency is E/2 clock.

- **Serial Operations**

The SCI hardware must be initialized by the software before operation. The sequence will be normally as follows.

* Writing the desired operation control bits of the Rate and Mode Control Register.
* Writing the desired operation control bits of the TRCS register.

If Port 2 bit 3, 4 are used for serial I/O, TE, RE bits may be kept set. When TE, RE bit are cleared during SCI operation, and subsequently set again, it should be noted that TE, RE must be kept "0" for at least one bit time of the current baud rate. If TE, RE are set again within one bit time, there may be the case where the initializing of internal function for transmitter and receiver does not take place correctly.

- **Transmit Operation**

Data transmission is enabled by the TE bit in the TRCS register. When set, the output of the transmit shift register is connected with Port 2 bit 4 which is unconditionally configured as an output.

After Reset, the user should initialize both the RMC register and the TRCS register for desired operation. Setting the TE bit causes a transmission of ten-bit preamble of "1"s. Following the preamble, internal synchronization is established and the transmitter is ready to operate. Then either of the following states exists.

(1) If the transmit data register is empty (TDRE = 1), the consecutive "1"s are transmitted indicating an idle states.
(2) If the data has been loaded into the Transmit Data Register (TDRE = 0), it is transferred to the output shift register and data transmission begins.

During the data transfer, the start bit ("0") is first transferred. Next the 8-bit data (beginning at bit 0) and inally the top bit ("1"). When the contents of the Transmit Data Register is transferred to the output shift register, the hardware sets the TDRE flag bit: If the CPU fails to respond to the flag within the proper time. TDRE is kept set and then a continuous string of 1's is sent until the data is supplied to the data register.

- **Receive Operation**

The receive operation is enabled by the RE bit. The serial input is connected with Port 2 bit 3. The receiver operation is determined by the contents of the TRCS and RMC register. The received bit stream is synchronized by the first "0" (start bit). During 10-bit time, the data is strobed approximately at the center of each bit. If the tenth bit is not "1" (stop bit), the system assumes a framing error and the ORFE is set.

If the tenth bit is "1", the data is transferred to the receive data register, and the RDRF flag is set. If the tenth bit of the next data is received and still RDRF is preserved set, then ORFE is set indicating that an overrun error has occurred.

After the CPU read of the status register as a response to RDRF flag or ORFE flag, followed by the CPU read of the receive data register, RDRF or ORFE will be cleared.

■ RAM CONTROL REGISTER

The register assigned to the address $0014 gives a status information about standby RAM.

RAM Control Register

	7	6	5	4	3	2	1	0
$0014	STBY PWR	RAME	x	x	x	x	x	x

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 RAM Enable.

Using this control bit, the user can disable the RAM, RAM Enable bit is set on the positive edge of $\overline{RES}$ and RAM is enabled. The program can write "1" or "0". If RAME is cleared, the RAM address becomes external address and the CPU may read the data from the outside memory.

Bit 7 Standby Bit

This bit can be read or written by the user program. It is cleared when the V_{CC} voltage is removed. Normally this bit is set by the program before going into stand-by mode. When the CPU recovers from stand-by mode, this bit should be checked. If it is "1", the data of the RAM is retained during stand-by and it is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD63701V0 has an upward object code compatible with the HD6801 to utilize all instruction sets of the HMCS6800. The execution time of the key instruction is reduced to increase the system through-put. In addition, the bit operation instruction, the exchange instruction between the index and the accumulator, the sleep instruction are added. This section describes:

* CPU programming model (See Fig. 23)
* Addressing modes
* Accumulator and memory manipulation instructions (See Table 8)
* New instructions
* Index register and stack manipulation instructions (See Table 9)
* Jump and branch instructions (See Table 10)
* Condition code register manipulation instructions (See Table 11)
* Op-code map (See Table 12)
* Cycle-by-Cycle Operation (See Table 13)

- **CPU Programming Model**

The programming model for the HD63701V0 is shown in Figure 23. The double accumulator is physically the same as the accumulator A concatenated with the accumulator B, so that the contents of

A and B is changed with executing operation of an accumulator D.

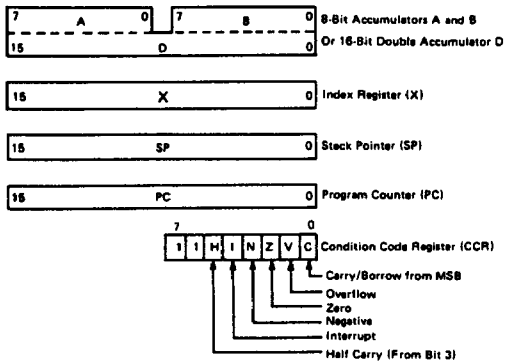


Figure 23 CPU Programming Model

- **CPU Addressing Modes**

The HD63701V0 has seven address modes which depend on both of the instruction type and the code. The address mode for every instruction is shown along with execution time given in terms of machine cycles (Table 8 to 12). When the clock frequency is 4 MHz, the machine cycles will be microseconds.

Accumulator (ACCX) Addressing

Only the accumulator (A or B) is addressed. Either accumulator A or B is specified by one-byte instructions.

Immediate Addressing

In this mode, the operand is stored in the second byte of the instruction except that the operand in LDS and LDX, etc. are stored in the second and the third byte. These are two or three-byte instructions.

Direct Addressing

In this mode, the second byte of instruction indicates the address where the operand is stored. Direct addressing allows the user to directly address the lowest 256 Bytes in the machine locations zero through 255. Improved execution times are achieved by storing data in these locations. For system configuration, it is recommended that these locations should be RAM and be utilized preferably for user's data realm. These are two-byte instructions except the AIM, OIM, EIM and TIM which have three-byte.

Extended Addressing

In this mode, the second byte indicates the upper 8 bits addresses where the operand is stored, while the third byte indicates the lower 8 bits. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the Index Register. For each of AIM, OIM, EIM and TIM instructions, the contents of the third byte are added to the lower 8 bits in the Index Register. In addition, the resulting "carry" is added to the upper 8 bits in the Index Register. The result is used for addressing memory. Because the modified address is held in the Temporary Address Register, there is no change to the Index Register. These are two-byte instructions but AIM, OIM, EIM, TIM have three-byte.

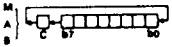
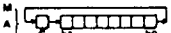
Implied Addressing

In this mode, the instruction itself gives the address; stack pointer, index register, etc. These are 1-byte instructions.

Relative Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the program counter. The resulting carry or borrow is added to the upper 8 bits. This helps the user to address the data within a range of -126 to $+129$ bytes of the current execution instruction. These are two-byte instructions.

Table 8 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	↕	•	↕	↕	↕	↕
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 → A	•	•	↕	↕	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	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	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	•	•	↕	↕	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	•	•	↕	↕	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 x B → A : B	•	•	•	•	•	⑪
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	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull 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				M / A / B 	•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3				M / A / B 	•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

Note) Condition Code Register will be explained in Note of Table 11.

(to be continued)

Table 8 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
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑤	↕
	ASLA													48	1	1	A	•	•	↕	↕	⑥	↕
	ASLB													58	1	1	B	•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	ACC A/ ACC B	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	1	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	1	1	B	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	1	1	A	•	•	R	↕	⑥	↕
	LSRB													54	1	1	B	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1	ACC A/ ACC B	•	•	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	B0	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	•

Note) Condition Code Register will be explained in Note of Table 11.

- **New Instructions**

In addition to the HD6801 Instruction Set, the HD63701V0 has the following new instructions:

AIM ..... (M) · (IMM) → (M)
Evaluates the AND of the immediate data and the memory, places the result in the memory.

OIM ..... (M) + (IMM) → (M)
Evaluates the OR of the immediate data and the memory, places the result in the memory.

EIM ..... (M) + (IMM) → (M)
Evaluates the EOR of the immediate data and the contents of memory, places the result in memory.

TIM ..... (M) · (IMM)
Evaluates the AND of the immediate data and the memory, changes the flag of associated condition code register.

Each instruction has three bytes; the first is op-code, the second is immediate data, the third is address modifier.

XGDX ..... (ACCD) ↔ (IX)
Exchanges the contents of double accumulator and the index register.

SLP ..... The CPU is brought to the sleep mode. For sleep mode, see the "sleep mode" section.

Table 9 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	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X − M M + 1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
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_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	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_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD↔IX	•	•	•	•	•	•

Note) Condition Code Register will be explained in Note of Table 11.

Table 10 Jump, Branch Instruction

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 ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ 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		⑧					
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

Note) *WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 11.

Table 11 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 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	•
Set 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	——— 16 ———					
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

[NOTE 1] 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 9? (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 of ACCB?

[NOTE 2] CLI instructions and interrupt.
If interrupt mask-bit is set (I="1") and interrupt is requested ($\overline{IRQ_1}$ = "0" or $\overline{IRQ_2}$ = "0"), and then CLI instruction is executed, the CPU responds as follows.

1 the next instruction of CLI is one-machine cycle instruction.
Subsequent two instructions are executed before the interrupt is responded.
That is, the next and the next of the next instruction are executed.
2 the next instruction of CLI is two-machine cycle (or more) instruction.
Only the next instruction is executed and then the CPU jump to the interrupt routine.
Even if TAP instruction is used, instead of CLI, the same thing occurs.

Table 12 OP-Code Map

OP CODE HI	LO	0000 0	0001 1	0010 2	0011 3	ACC A 0100 4	ACC B 0101 5	IND 0110 6	EXT / DIR 0111 7	ACCA or SP IMM 1000 8	DIR 1001 9	IND 1010 A	EXT 1011 B	ACCB or X IMM 1100 C	DIR 1101 D	IND 1110 E	EXT 1111 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				ADDD				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	

UNDEFINED OP CODE

* Only for instructions of AIM, OIM, EIM, TIM

• **Instruction Execution Cycles**

In the HMCS6800 series, the execution cycle of each instruction is the number of cycles between the start of the current instruction fetch and just before the start of the subsequent instruction fetch.

The HD63701V0 uses a mechanism of the pipeline control for the instruction fetch and the subsequent instruction fetch is performed during the current instruction being executed.

Therefore, the method to count instruction cycles used in the HMCS6800 series cannot be applied to the instruction cycles such as MULT, PULL, DAA and XGDX in the HD63701V0.

Table 13 provides the information about the relationship among each data on the Address Bus, Data Bus, and R/W status in cycle-by-cycle basis during the execution of each instruction.

Table 13 Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMMEDIATE					
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	2	1	Op Code Address+1	1	Operand Data
		2	Op Code Address+2	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	3	1	Op Code Address+1	1	Operand Data (MSB)
		2	Op Code Address+2	1	Operand Data (LSB)
		3	Op Code Address+3	1	Next Op Code
DIRECT					
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	3	1	Op Code Address+1	1	Address of Operand (LSB)
		2	Address of Operand	1	Operand Data
		3	Op Code Address+2	1	Next Op Code
STA	3	1	Op Code Address+1	1	Destination Address
		2	Destination Address	0	Accumulator Data
		3	Op Code Address+2	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	4	1	Op Code Address+1	1	Address of Operand (LSB)
		2	Address of Operand	1	Operand Data (MSB)
		3	Address of Operand+1	1	Operand Data (LSB)
		4	Op Code Address+2	1	Next Op Code
STD STS STX	4	1	Op Code Address+1	1	Destination Address (LSB)
		2	Destination Address	0	Register Data (MSB)
		3	Destination Address+1	0	Register Data (LSB)
		4	Op Code Address+2	1	Next Op Code
JSR	5	1	Op Code Address+1	1	Jump Address (LSB)
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Jump Address	1	First Subroutine Op Code
TIM	4	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Op Code Address+3	1	Next Op Code
AIM EIM OIM	6	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	Address of Operand	0	New Operand Data
		6	Op Code Address+3	1	Next Op Code

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

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

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

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

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMPLIED					
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	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Accumulator Data
		4	Op Code Address+1	1	Next Op Code
PULX	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Index Register (LSB)
		4	Stack Pointer−1	0	Index Register (MSB)
		5	Op Code Address+1	1	Next Op Code
RTS	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Return Address (MSB)
		4	Stack Pointer+2	1	Return Address (LSB)
		5	Return Address	1	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	FFFF	1	Restart Address (LSB)
		5	FFFF	1	Restart Address (LSB)
		6	FFFF	1	Restart Address (LSB)
		7	FFFF	1	Restart Address (LSB)

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

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

– Continued –

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
RELATIVE					
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	Branch Offset
		2	FFFF	1	Restart Address (LSB)
		3	Branch Address·····Test="1" Op Code Address+1···Test="0"	1	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Branch Address	1	First Op Code of Subroutine

■ LOW POWER CONSUMPTION MODE

The HD63701V0 has two low power consumption modes; sleep and standby mode.

• Sleep Mode

On execution of SLP instruction, the MCU is brought to the sleep mode. In the sleep mode, the CPU sleeps (the CPU clock becomes inactive), but the contents of the registers in the CPU are retained. In this mode, the peripherals of CPU will remain active. So the operations such as transmit and receive of the SCI data and counter may keep in operation. In this mode, the power consumption is reduced to about 1/6 the value of a normal operation.

The escape from this mode can be done by interrupt, $\overline{RES}$, $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode (This will be mentioned later). When interrupt is requested to the CPU and accepted, the sleep mode is released, then the CPU is brought in the operation mode and jumps to the interrupt routine. When the CPU has masked the interrupt, after recovering from the sleep mode, the next instruction of SLP starts to execute. However, in such a case that the timer interrupt is inhibited on the timer side, the sleep mode cannot be released due to the absence of the interrupt request to the CPU. Fig. 24 shows the timing chart of sleep instruction.

This sleep mode is available to reduce an average power consumption in the applications of the HD63701V0 which may not be always running.

The conditions of clock pins, input pins and E clock pin are the same as those of operation. Refer to Table 15 for the other pin conditions.

• Standby Mode

Bringing $\overline{STBY}$ "Low", the CPU becomes reset and all clocks of the HD63701V0 become inactive. It goes into the standby mode. This mode remarkably reduces the power consumptions of the HD63701V0.

In the standby mode, if the HD63701V0 is continuously supplied with power, the contents of RAM is retained. The standby mode should escape by the reset start. The following is the typical application of this mode.

First, NMI routine stacks the MCU's internal information and the contents of SP in RAM, disables RAME bit of RAM control register, sets the Standby bit, and then goes into the standby mode. If the Standby bit keeps set on reset start, it means that the power has been kept during standby mode and the contents of RAM is normally guaranteed. The system recovery may be possible by returning SP and bringing into the condition before the standby mode has started. The timing relation for each line in this application is shown in Figure 25.

In this mode, $\overline{RES}$, $\overline{NMI}$, $\overline{IRQ}$ pins are not active, XTAL is in "1" output and all the other pins are in high impedance. When external clock is used the input level to EXTAL should be "High" (= V_{CC} is desirable) to reduce the standby current. If EXTAL is "Low" the standby current is increased by approximately 5 μA.

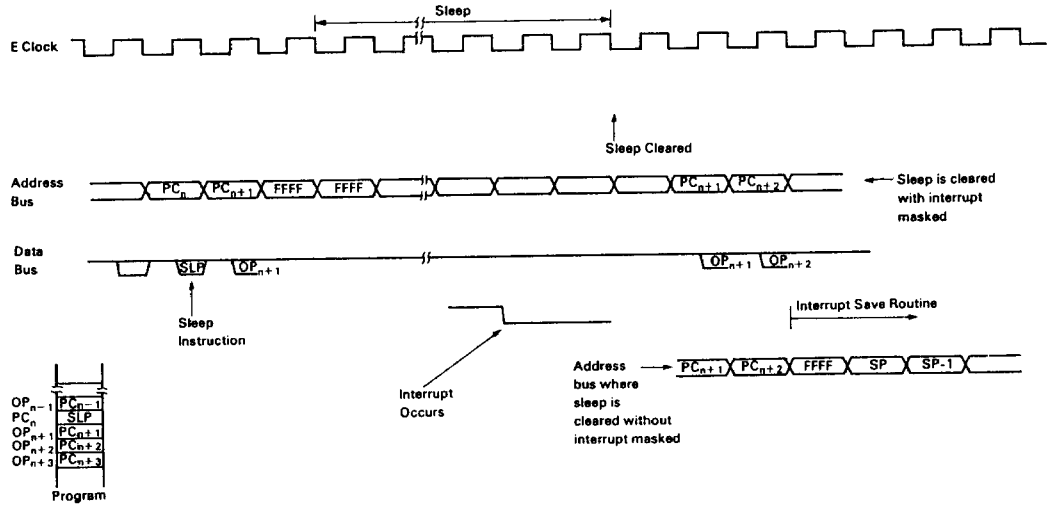


Figure 24 Sleep Instruction Timing Chart

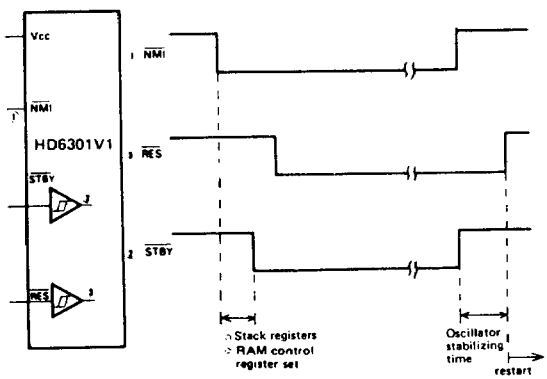


Figure 25 Standby Mode Timing

■ ERROR PROCESSING

When the HD63701V0 fetches an undefined instruction or fetches an instruction from unusable memory area, it generates the highest priority internal interrupt, that may protect from system upset due to noise or a program error.

- **Op-Code Error**

Fetching an undefined op-code, the HD63701V0 will stack the CPU register as in the case of a normal interrupt and vector to the TRAP ($FFEE, $FFEF), that has a second highest priority (RES is the highest).

- **Address Error**

When an instruction is fetched from other than a resident EP-ROM, RAM, or an external memory area, the CPU starts the same interrupt as op-code error. In the case which the instruction is fetched from external memory area and that area is not usable, the address error cannot be detected.

The addresses which cause address error in particular mode are shown in Table 14.

Table 14 Address Error

Mode	0	1	2	5	6	7
Address	$0000 ↓ $0001F	$0000 ↓ $001F	$0000 ↓ $001F	$0000 ↓ $003F $0200 ↓ $EFFF	$0000 ↓ $001F	$0000 ↓ $003F $0100 ↓ $EFFF

This feature is applicable only to the instruction fetch, not to normal read/write of data accessing.

System Flow chart of HD63701V0 is shown in Fig. 26.

Transitions among the active mode, sleep mode, standby mode and reset are shown in Fig. 27.

Figures 28, 29, 30 and 31 shows a system configuration.

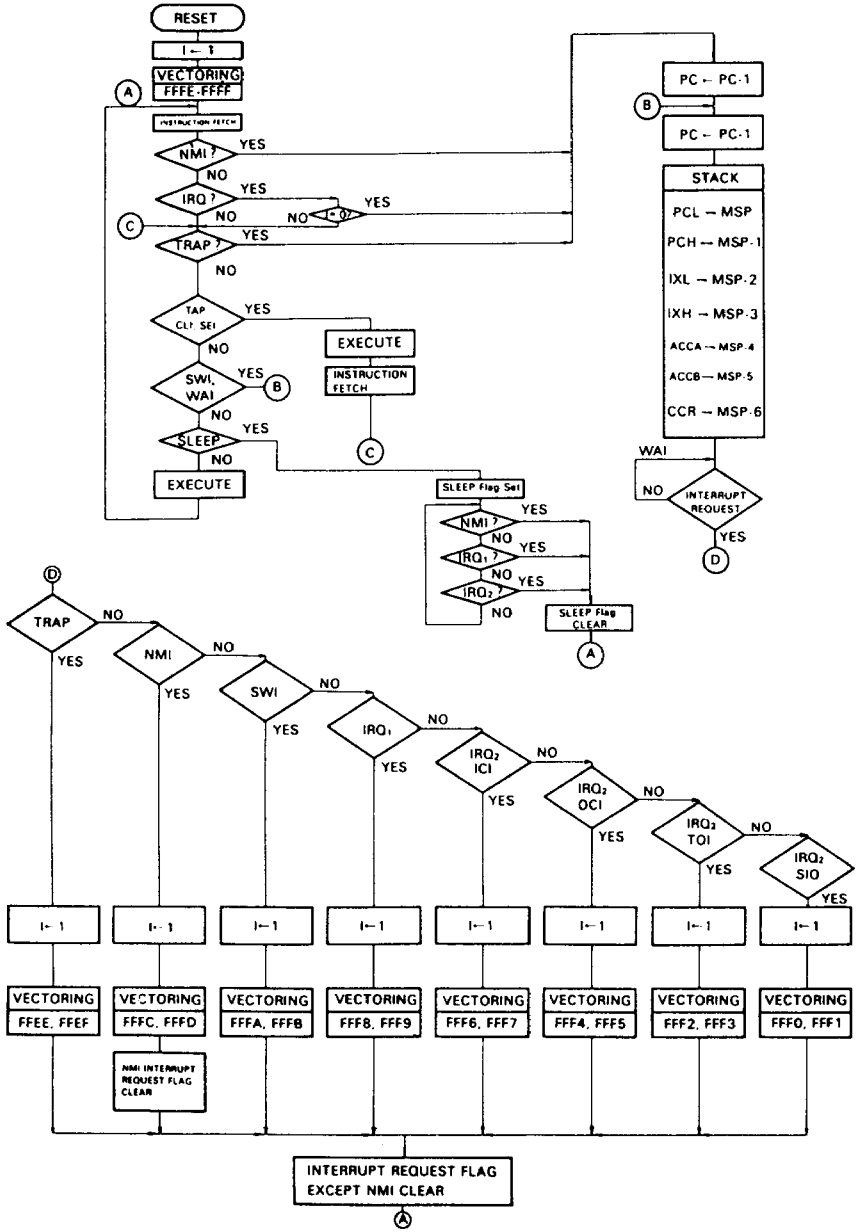


Figure 26 HD63701V0 System Flow Chart

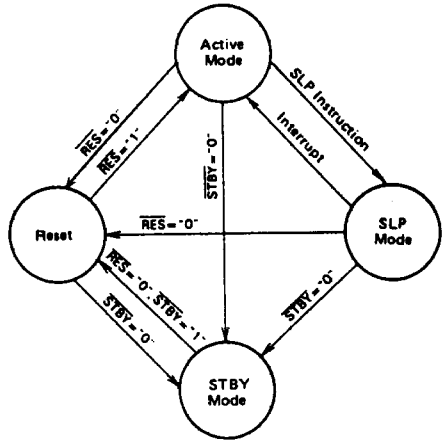


Figure 27 Transitions among Active Mode, Standby Mode, Sleep Mode, and Reset

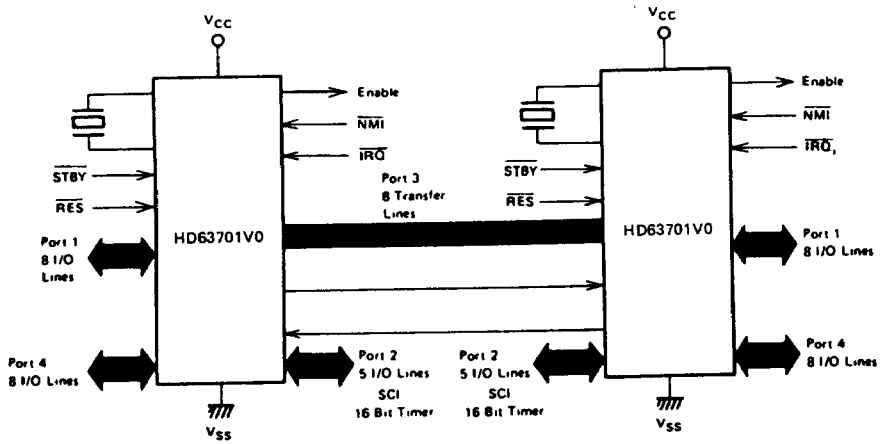


Figure 28 HD63701V0 MCU Single-Chip Dual Processor Configuration

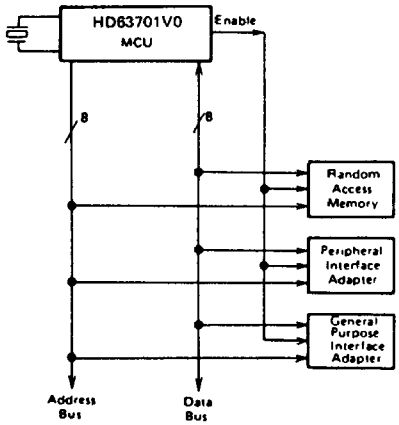


Figure 29 HD63701V0 MCU Expanded Non-Multiplexed Mode (Mode 5)

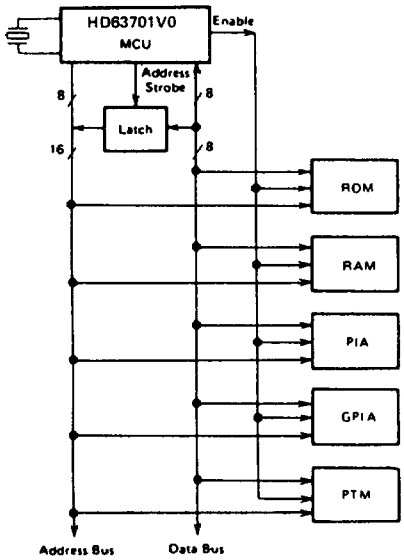


Figure 30 HD63701V0 MCU Expanded Multiplexed Mode

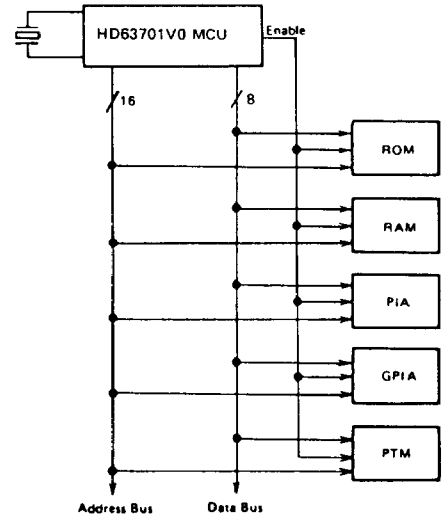


Figure 31 HD63701V0 MCU Expanded Non-Multiplexed Mode (Mode 1)

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

There is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD63701V0 as possible and insert bypass capacitor between V_{CC} line and GND for V_{CC} stabilization to minimize the start up time.

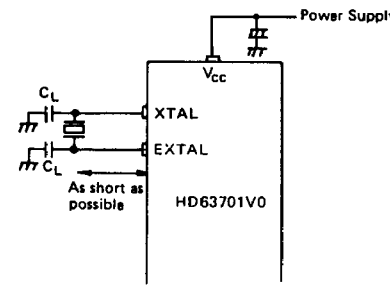


Figure 32 Precaution to the board design of oscillation circuit

Table 15 Pin Condition in Sleep Mode

Pin \ Mode		0	1	2	5	6	7
Port 1 P_{10}~P_{17}	Function	I/O Port	Lower Address Bus	I/O Port	←	←	←
	Condition	Keep the condition just before sleep	Output "1"	Keep the condition just before sleep	←	←	←
Port 2 P_{20}~P_{24}	Function	I/O Port	←	←	←	←	←
	Condition	Keep the condition just before sleep	←	←	←	←	←
Port 3 P_{30}~P_{37}	Function	$\overline{E}$: Lower Address Bus E: Data Bus	Data Bus	$\overline{E}$: Lower Address Bus E: Data Bus	Data Bus	$\overline{E}$: Lower Address Bus E: Data Bus	I/O Port
	Condition	$\overline{E}$: Output "1" E: High Impedance	High Impedance	$\overline{E}$: Output "1" E: High Impedance	High Impedance	$\overline{E}$: Output "1" E: High Impedance	Keep the condition just before sleep
Port 4 P_{40}~P_{47}	Function	Upper Address	←	←	Lower Address Bus or Input Port	Upper Address Bus or Input Port	I/O Port
	Condition	Output "1"	←	←	Address Bus: Output "1" Port: Keep the condition just before sleep	←	Keep the condition just before sleep
SC_2		Output "1" (Read Condition)	←	←	←	←	Output "1"
SC_1		Output Address Strobe	←	←	Output "1"	Output Address Strobe	Input Pin

Table 16 Pin Condition during RESET

pin \ mode	0, 2, 6	1	5	7
Port 1 P_{10} ~ P_{17}	high impedance (input)	←	←	←
Port 2 P_{20} ~ P_{24}	high impedance (input)	←	←	←
Port 3 P_{30} ~ P_{37}	$\overline{E}$: "1" output E: high impedance	high impedance	←	←
Port 4 P_{40} ~ P_{47}	high impedance (input)	←	←	←
SC_2 (R/$\overline{W}$)	"1" output (Read)	←	←	"1" output
SC_1 (AS)	$\overline{E}$: "1" output E: "0" output	←	"1" output	high impedance (input)

■ PROGRAMMING THE HD63701V0 EPROM

To program the on chip EPROM, it is necessary to operate the HD63701V0 in EPROM mode. The HD63701V0 becomes EPROM mode by setting $\overline{STBY}$ = Low, P_{20} = High, P_{21} = High, P_{22} = Low, $\overline{NMI}$ = Low, EXTAL = High and XTAL = Low, as shown in Fig. 33 and Fig. 34.

In EPROM mode, the on chip EPROM can be programmed by the same procedure as that of 27C256 or 27256. So it can be programmed by EPROM writer for 27C256 or 27256.

The memory map of EPROM mode is shown in Fig. 35. This mode has 32k byte address space ($0000 to $7FFF), and internal EPROM is located at the top ($0000 to $0FFF). When using a EPROM writer for 27C256 (27256), the data to be written into the HD63701V0 must be located at $0000 through $0FFF on the buffer RAM of EPROM writer.

Write operation to addresses $1000 through $7FFF does not affect the HD63701V0, and read data from these addresses are always $FF.

• Program/Verify

The HD63701V0 EPROM can utilizes the high-performance programming method. This method achieves faster programming time without any voltage stress to the device nor deterioration in reliability of programming data. The basic flow chart and timing diagram are shown in Fig. 36 and Fig. 37.

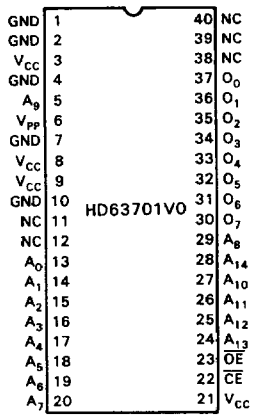


Figure 33 HD63701V0 EPROM Mode Pin Assignment

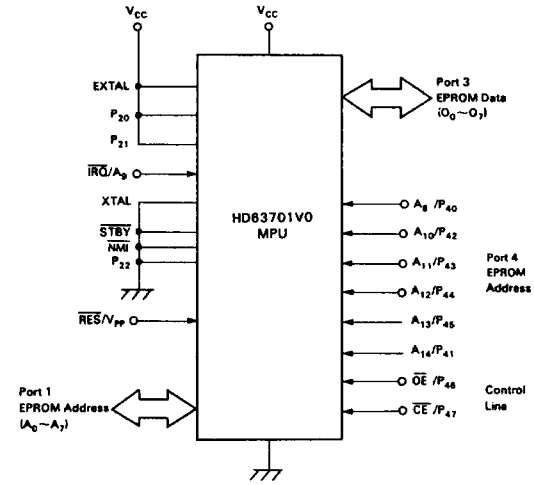


Figure 34 EPROM Mode

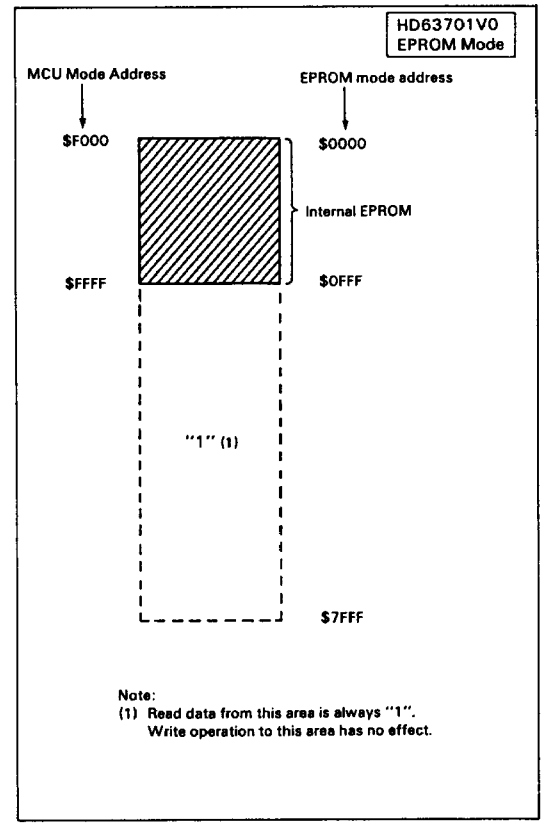


Figure 35 EPROM Mode Memory Map

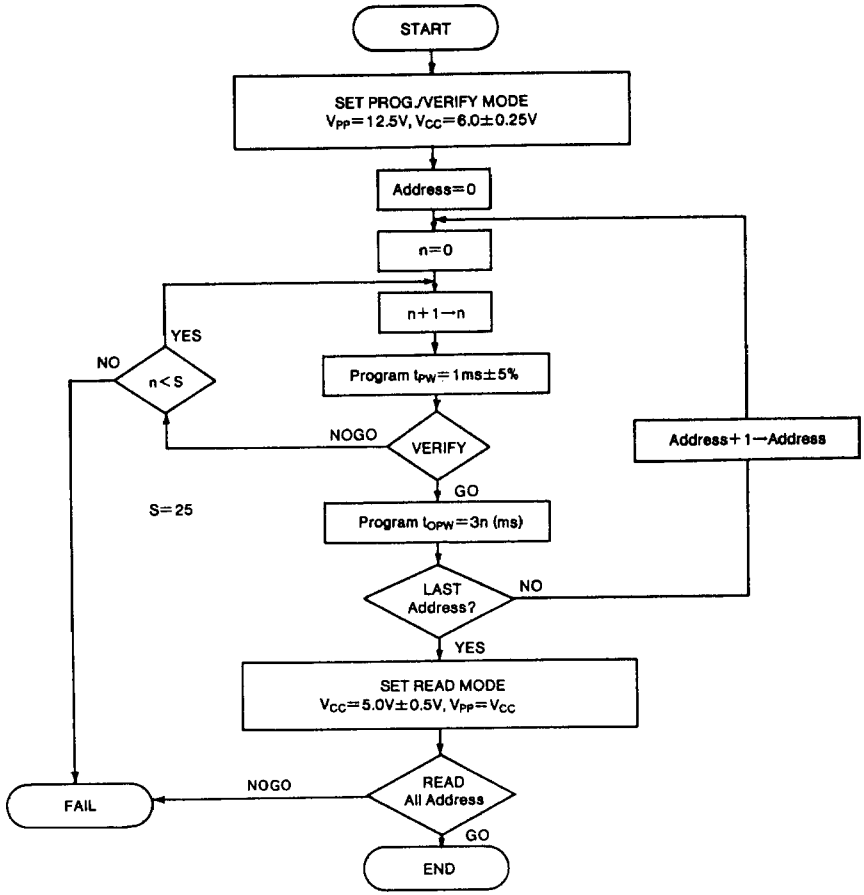


Figure 36 High Performance Programming Flowchart

Table 17 Mode Select

Mode \ Pin	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{CC}	Output
Read	L	L	V_{CC}	V_{CC}	D_{out}
Output Disable	L	H	V_{CC}	V_{CC}	High Z
High Performance Program	L	H	V_{PP}	V_{CC}	D_{in}
Verify	H	L	V_{PP}	V_{CC}	D_{out}
Program Inhibit	H	H	V_{PP}	V_{CC}	High Z

■ PROGRAMMING ELECTRICAL CHARACTERISTICS

• **DC CHARACTERISTICS (V_{CC} = 6.0V±0.25V, V_{PP} = 12.5V±0.3V, V_{SS} = 0V, Ta = 25°C±5°C unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	O_0~O_7, A_0~A_{11}, OE, CE	V_{IH}		2.2	–	V_{CC}+0.3	V
Input "Low" Voltage	O_0~O_7, A_0~A_{11}, OE, CE	V_{IL}		−0.3	–	0.8	V
Output "High" Voltage	O_0~O_7	V_{OH}	I_{OH} = −200μA	2.4	–	–	V
Output "Low" Voltage	O_0~O_7	V_{OL}	I_{OL} = 1.6mA	–	–	.45	V
Input Leakage Current	O_0~O_7, A_0~A_{11}, OE, CE	I_{LI}	V_{IN} = 5.25V/0.5V	–	–	2	μA
V_{CC} Current		I_{CC}		–	–	30	mA
V_{PP} Current		I_{PP}		–	–	30	mA

• **AC CHARACTERISTICS (V_{CC} = 6.0V±0.25V, V_{PP} = 12.5V±0.3V, V_{SS} = 0V, Ta = 25°C±5°C unless otherwise noted.)**

Item	Symbol	Test condition	min	ty	max	Unit
Address Set-up Time	t_{AS}		2	–	–	μs
$\overline{CE}$ Set-up Time	t_{OES}		2	–	–	μs
Data Set-up Time	t_{DS}		2	–	–	μs
Address Hold Time	t_{AH}		0	–	–	μs
Data Hold Time	t_{DH}		2	–	–	μs
Data Output Disable Time	t_{DF}	Fig.37*1			130	ns
V_{PP} Set-up Time	t_{VPS}		2	–	–	μs
Program Pulse Width (High Speed Writing)	t_{PW}		0.95	1.0	1.05	ms
Program Pulse Width	t_{OPW}		2.85	–	78.75	ms
V_{CC} Set-up Time	t_{VCS}		2	–	–	μs
Data Output Delay Time	t_{OE}		0	–	150	ns

*1 Input Pulse Level = 0.8~2.2V
Input Rise Time/Fall Time ≦ 20ns.
Tuning Reference Level Input: 1.0V, 2.0V.
Output: 0.8V, 2.0V.

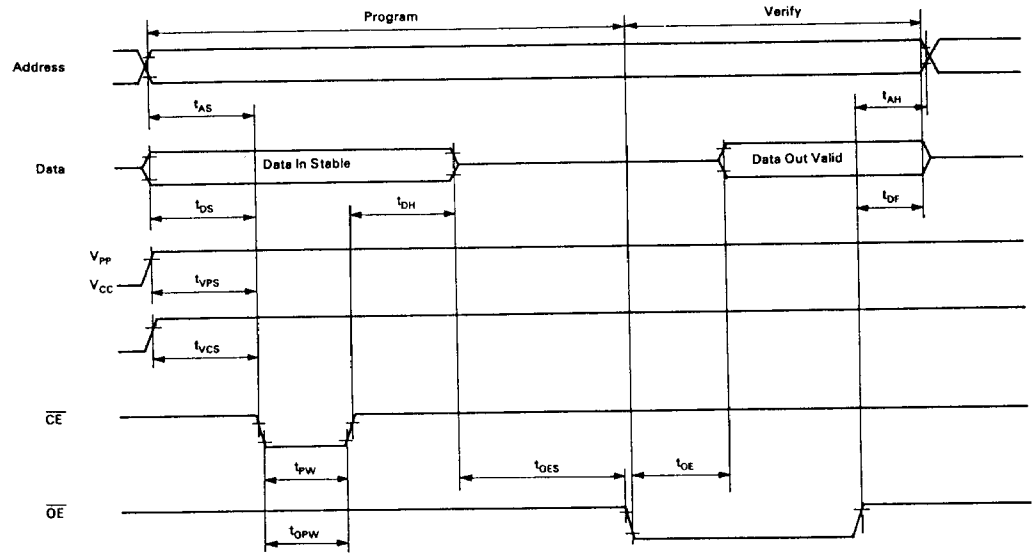


Figure 37 EPROM Program/Verify Timing

■ READ OPERATION

• DC CHARACTERISTICS ($Ta = 25°C \pm 5°C$, $V_{CC} = 5V \pm 10\%$, $V_{PP} = V_{CC}$)

Item	Symbol	Test Condition	min	typ	max	Unit
Input High Voltage	V_{IH}		2.2	–	$V_{CC}+0.3$	V
Input Low Voltage	V_{IL}		−0.3	–	0.8	V
Output High Voltage	V_{OH}	$I_{OH} = -200\mu A$	2.4	–	–	V
Output Low Voltage	V_{OL}	$I_{IL} = 1.6mA$	–	–	0.5	V
V_{CC} Current	I_{CC}	$\overline{CE} = V_{IL}$	–	–	30	mA
V_{PP} Current	I_{PP}	$V_{PP} = V_{CC}$	–	–	1.0	mA

• AC CHARACTERISTICS ($Ta = 25°C \pm 5°C$, $V_{CC} = 5V \pm 10\%$, $V_{PP} = V_{CC}$)

Item	Symbol	Test Condition		min	typ	Unit
Address to Output Delay	t_{ACC}	$\overline{CE} = \overline{OE} = V_{IL}$	Fig.38*2	–	300	ns
$\overline{CE}$ to Output Delay	t_{CE}	$\overline{OE} = V_{IL}$		–	300	ns
$\overline{OE}$ to Output Delay	t_{OE}	$\overline{CE} = V_{IL}$		–	120	ns
$\overline{OE}$ to Output Float	t_{DF}	$\overline{CE} = V_{IL}$		0	105	ns
Address to Output Hold	t_{OH}	$\overline{CE} = \overline{OE} = V_{IL}$		0	–	ns

*2 Input Pulse Level = 0.8–2.2V
Input Rise Time/Fall Time ≦ 20ns.
Output Load = I_{TTL} Gate + 90pF
Timing Reference Level Input: 1.0V, 2.0V.
Output: 0.8V, 2.0V.

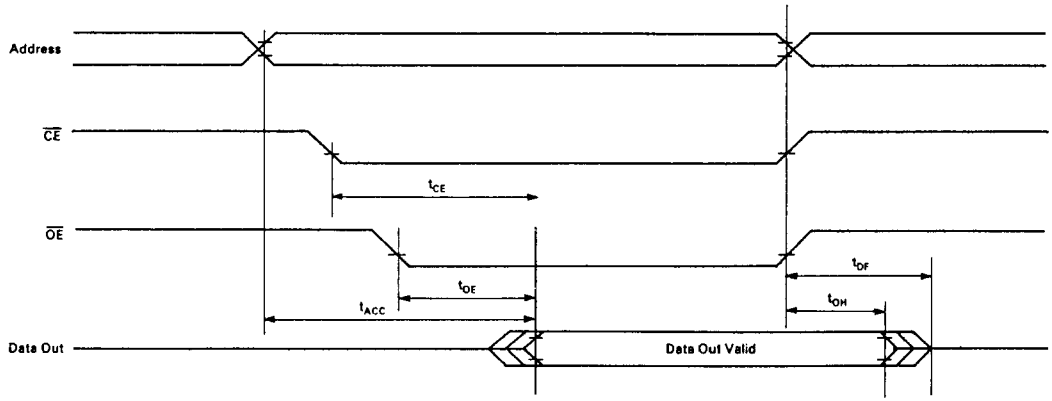


Figure 38 EPROM Read Timing

- **Erasure**

Erasure of HD63701V0 EPROM is performed by exposure to ultraviolet light of 2537Å and all the data are changed to "1" after this erasure procedure. The minimum integrated dose (i.e. UV intensity × exposure time) for erasure is 15W·sec/cm². this condition is attained by exposing a device to an ultraviolet lamp of 1200 μW/cm² for 20 to 30 minutes. The HD63701V0 should be placed about one inch away from UV lamp tubes.

[Precautions on using EPROM On-chip Single-chip Microcomputer]

(1) If the MCU is exposed to strong light especially a fluorescent lamp or the sunlight, EPROM data may be erased or the MCU may malfunction by photocurrent. Therefore, after programming, it is suggested that applications which expose the LSI to ambient light may require an opaque label over the window.
(2) Do not rub the window with materials like plastics, or do not touch a charged body on the window. Electrostatic charge may adversely affect the functionality of the LSI. A conductive opaque label, suggested above, is effective on distributing charge equally.
(3) If the glass window is stained, erasure time will be extended. Remove stains from the window with a solvent which has no influence on the package like alcohol. Don't rub the window hard but wipe out softly.

■ DIFFERENCES BETWEEN HD6301V1, HD6303R, HD63P01M1, AND HD63701V0

Basically, HD63701V0 is compatible with HD6301V1 but it has several different functions and characteristics. Table 17 shows the difference.

■ DIFFERENCES BETWEEN HD6301V1, HD6303R, HD63P01M1, AND HD63701V0

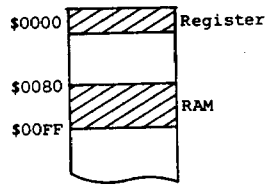
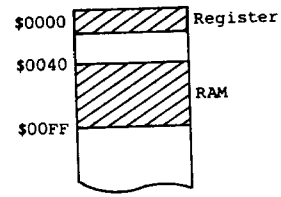
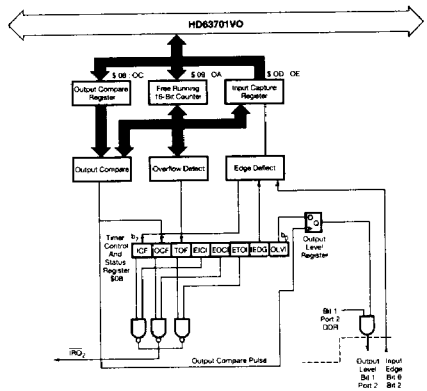
<table>
<tr><th colspan="2">Item</th><th colspan="2">HD6301V</th><th>HD63701V0</th></tr>
<tr><td rowspan="5">Function</td><td>RAM</td><td colspan="2">RAM Size: 128-byte
Address: $0080–$00FF
 
</td><td>RAM Size: 192-byte
Address: $0040–$00FF
 
</td></tr>
<tr><td>Operation Mode</td><td colspan="2">Mode 4: Expanded Multiplexed Mode = Mode 2</td><td>HD63701V0 does not have Mode 4</td></tr>
<tr><td>Timer</td><td colspan="2">After providing supply voltage, output level is undefined (0 or 1) unless the contents of the Output Compare Register matches with those of the Free Running Counter. The Output Level Register is not initialized by reset.
 

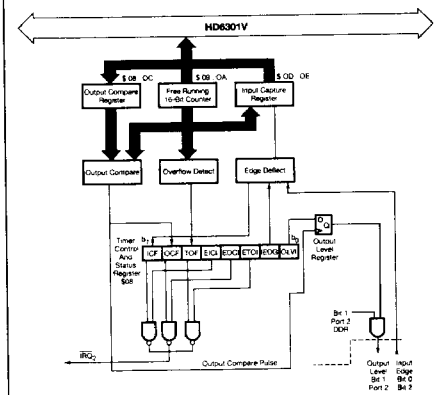
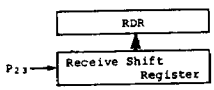
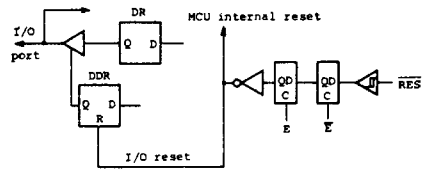
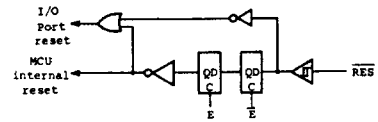
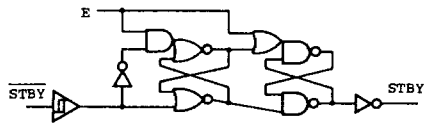
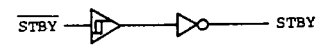
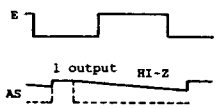
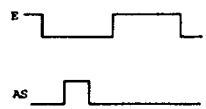
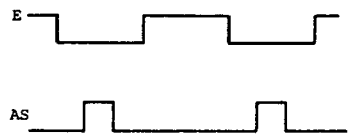
Figure 20 Programmable Timer Block Diagram</td><td>The Output Level Register is initialized to 0 by reset.
 

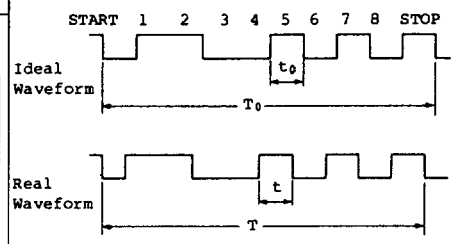
Figure 20 Programmable Timer Block Diagram</td></tr>
<tr><td>SCI</td><td>HD6301V1, HD6303R, HD63P01M1
When framing error occurs, receive data is not transferred from the Receive Shift Register to Receive Data Register (RDR).
 
</td><td>HD6303R1
Receive data is transferred from Receive Shift Register to RDR even if framing error occurs.</td><td>Receive data is transferred from Receive Shift Register to RDR even if framing error occurs.</td></tr>
</table>

■ DIFFERENCES BETWEEN HD6301V1, HD6303R, HD63P01M1, AND HD63701V0 (Continued)

<table>
<tr><th colspan="2">Item</th><th>HD6301V</th><th>HD63701V0</th></tr>
<tr><td rowspan="4">Function</td><td>Port Reset</td><td>The DDR of port is reset synchronously with E clock. I/O state is undefined from providing power supply till oscillation start (max. 20ms).
 
</td><td>The DDR of port is reset asynchronously with E clock. CPU enters into high impedance state (input state) by bringing RES Low.
Reset release and MCU internal reset is performed synchronously with E clock.
 
</td></tr>
<tr><td>Standby Mode</td><td>STBY signal is latched synchronously with E clock.
 
</td><td>STBY signal is latched asynchronously with E clock. CPU enters into standby state by bringing STBY low.
 
</td></tr>
<tr><td>AS (Address Strobe)</td><td>HD63P01M1
 

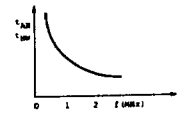
In Expanded Multiplexed Mode (mode 0, 2, 4 or 6), AS becomes high impedance state for a half E clock cycle during reset. Therefore, I/O Port 3 functions as data bus during reset.</td><td>HD6301V1, HD6303R, HD6303R1
 

During reset, AS functions normally.</td><td> 

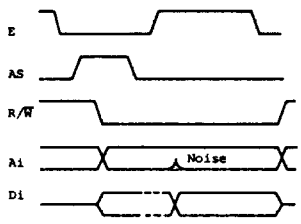
During reset, AS functions normally.</td></tr>
<tr><td>SCI Receive Margin</td><td>HD6301V1, HD6303R, HD6303R1
The SCI receive margin is shown below.
<table><tr><td>Bit distortion tolerance $(t-t_0)/t_0$</td><td>±37.5%</td></tr><tr><td>Character distortion tolerance $(T-T_0)/T_0$</td><td>+3.5%
−2.5%</td></tr></table></td><td>HD63P01M1
The SCI receive margin is shown below.
<table><tr><td>Bit distortion tolerance $(t-t_0)/t_0$</td><td>±25%</td></tr><tr><td>Character distortion tolerance $(T-T_0)/T_0$</td><td>±3.75%</td></tr></table></td><td>The SCI receive margin is shown below.
 

<table><tr><td>Bit distortion tolerance $(t-t_0)/t_0$</td><td>±37.5%</td></tr><tr><td>Character distortion tolerance $(T-T_0)/T_0$</td><td>±3.75%</td></tr></table></td></tr>
</table>

■ **DIFFERENCES BETWEEN HD6301V1, HD6303R, HD63P01M1, AND HD63701V0 (Continued)**

<table>
<tr><th colspan="2" rowspan="2">Item</th><th colspan="2">HD6301V</th><th rowspan="2">HD63701V0</th></tr>
<tr><th>HD6301V1, HD6303R, HD6303R1</th><th>HD63P01M1</th></tr>
<tr><td rowspan="4">Function</td><td>Supply Voltage</td><td>V_{CC} = 5V ± 10%
(f = 0.1 ~ 2 MHz)
V_{CC} = 3 ~ 6V
(f = 0.1 ~ 0.5 MHz)</td><td>V_{CC} = 5V ± 10%
(f = 0.1 ~ 1 MHz)</td><td>V_{CC} = 5V ± 10% (f = 0.1 ~ 2 MHz)</td></tr>
<tr><td>Address/Data Hold Time (t_{AH}, t_{HW})</td><td colspan="2">t_{AH} = 20 ns min.
t_{HW} = 20 ns min.
t_{AH} and t_{HW} are constant independently of operating frequency.</td><td>t_{AH}, t_{HW} = 60 ns (f = 1 MHz)
= 40 ns (f = 1.5 MHz)
= 30 ns (f = 2 MHz)
t_{AH} and t_{HW} are proportion to 1/f. (f = operating frequency)
  </td></tr>
<tr><td rowspan="5">Specification</td><td>Address Delay Time</td><td colspan="2">(1) t_{AD1} and t_{AD2} are constant independently of operating frequency. In HD63B01V (B version of HD6301V), t_{AD1} and t_{AD2} are 160 ns max. at 0.1 MHz through 2 MHz operation.
(2) t_{ADL} is related to operating frequency. (t_{ADL} is in proportion to 1/f. f = operating frequency)</td><td>t_{AD1}, t_{AD2} and t_{ADL} are related to operating frequency (They are in proportion to 1/f. f = operating frequency). Therefore, if HD637B01V operates at lower operating frequency, t_{AD1}, t_{AD2} and t_{ADL} will become 160 ns or more. t_{AD1}, t_{AD2} and t_{ADL} are calculated as follows.
t_{AD} (f MHz) ≒ 250 ns (1 MHz) × 1/f (MHz)</td></tr>
<tr><td>I_{in} and C_{in} of $\overline{RES}$</td><td colspan="2">I_{in} = 1.0 μA max., C_{in} = 12.5 pF max.</td><td>I_{in} = 10 μA max. C_{in} = 50 pF max. Since $\overline{RES}$ is multiplexed with V_{PP}, C_{in} and I_{in} are larger than those of HD6301V.</td></tr>
<tr><td>Load Capacitance of E</td><td colspan="2">2 – LSTTL + 40pF
I_{OL} = 0.8 mA, I_{OH} = –200 μA</td><td>1 – TTL + 90pF
I_{OL} = 1.6 mA, I_{OH} = –200 μA</td></tr>
<tr><td>Load Capacitance of Port 1</td><td colspan="2">1 – TTL + 30pF</td><td>1 – TTL + 90pF</td></tr>
<tr><td>Spec. of Crystal Oscillator</td><td colspan="2">Spec.
R_s = 60Ω max.</td><td>Spec.
<table><tr><td>Clock frequency (MHz)</td><td>2.5</td><td>4.0</td><td>6.0</td><td>8.0</td></tr><tr><td>Rs max. (Ω)</td><td>500</td><td>120</td><td>80</td><td>60</td></tr></table></td></tr>
<tr><td>Storage Temperature</td><td colspan="2">T_{stg} = –55 – +150°C</td><td>T_{stg} = –55 – +125°C</td></tr>
</table>

■ **DIFFERENCES BETWEEN HD6301V1, HD6303R, HD63P01M1, AND HD63701V0 (Continued)**

	Item	HD6301V		HD63701V0
Function	GND Noise	HD6301V1, HD6303R	HD6303R1, HD63P01M1	Noise is reduced by 50%.
		 If load capacitance in each data line and GND impedance are large, noise may appear on address bus during MCU write cycle and data won't be written into RAM correctly. The noise is caused by GND impedance which becomes large when large transient current flows into GND at High to Low transition of data line.	Noise is reduced by 33%.	
	Miscellaneous	Chip design and manufacturing process of the HD6301V differ from those of the HD63701V0. Therefore, actual spec. and margin are different between the HD6301V and the HD63701V0. Please carefully examine your system before applying HD6301V or HD63701V0 to your system.		

■ **PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT**

As shown in Fig. 39, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD63701V0 as possible.

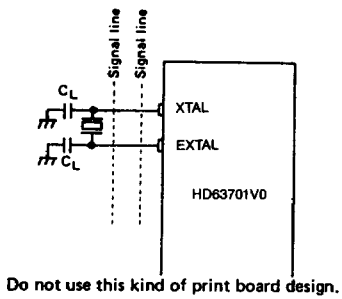


Figure 39 Precaution to the board design of oscillation circuit

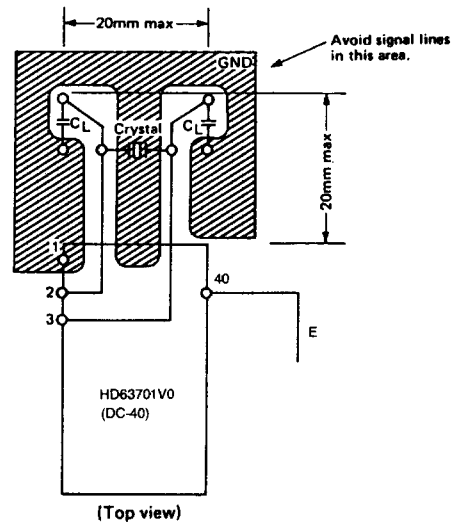


Figure 40 Example of Oscillation Circuits in Board Design

■ 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 41.

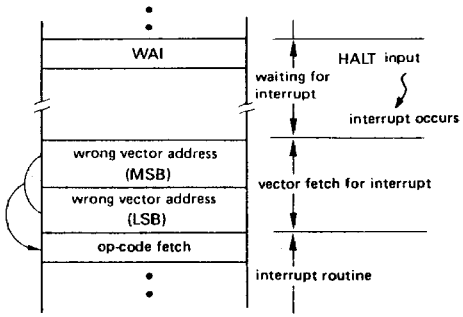


Figure 41 MAC function during WAI

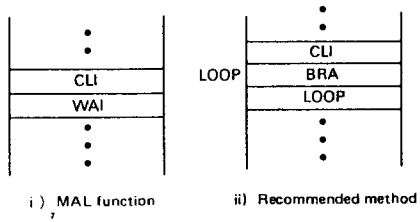


Figure 42 Program to wait for interrupt

■ 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 a 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

$\overline{RES}$ 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 $\overline{RES}$ signal is input to the LSI in synchronism with the internal clock ϕ (shown in Figure 43.)

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 oscillator stabilization time.

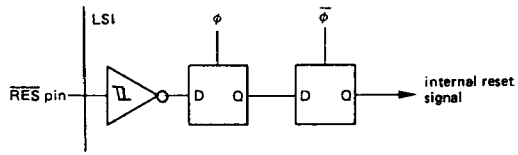


Figure 43 $\overline{RES}$ circuit