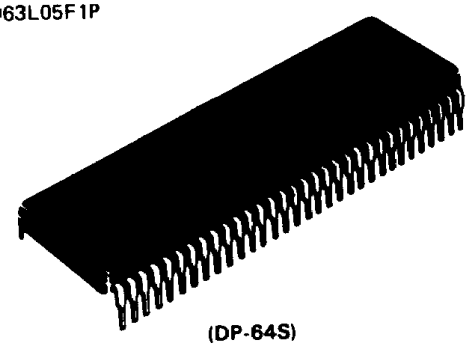
HD63L05F1
CMOS MCU (Microcomputer Unit)

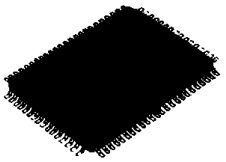
The HD63L05F1 is a CMOS single-chip microcomputer suitable for low-voltage and low-current operation. Having CPU functions similar to those of the HMCS6800 family, the HD63L05F1 is equipped with a 4k bytes ROM, 96 bytes RAM, I/O, timer, 8 bits A/D, and LCD (7 x 7 segments max.) drivers, all on one chip.

HD63L05F1P



(DP-64S)

HD63L05F1F



(FP-80)

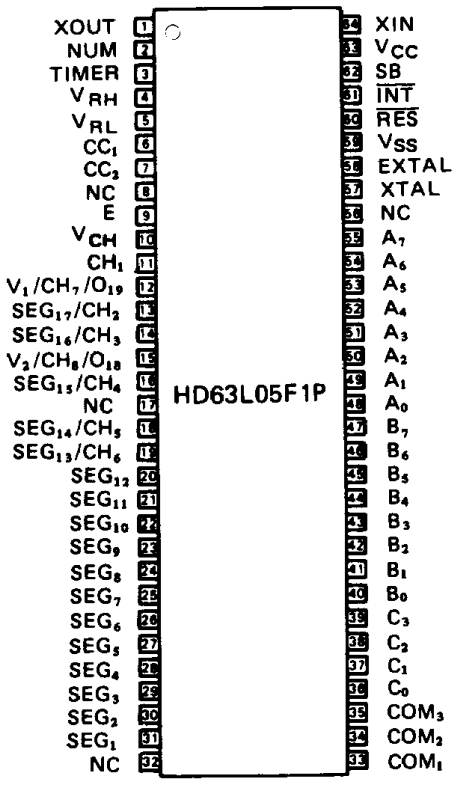
■ HARDWARE FEATURES

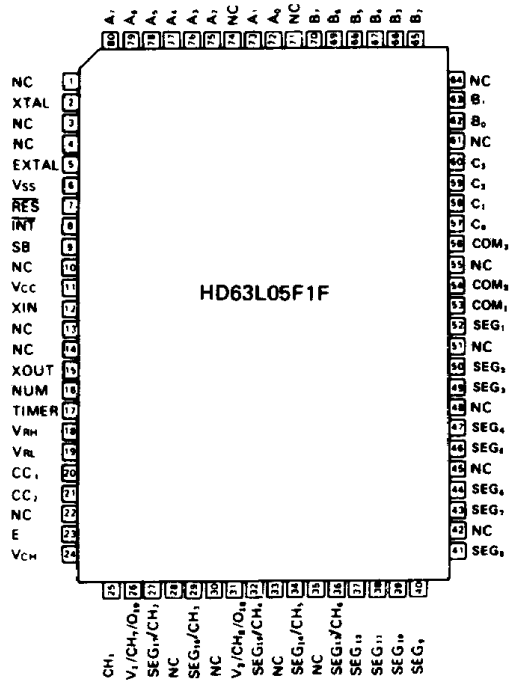
- 3V Power Supply
- 8-Bit Architecture
- Built-in 4k Bytes ROM (Mask ROM)
- Built-in 96 Bytes RAM
- 20 Parallel I/O Ports
- Built-in 7 x 7 Segments LCD Driver Capability
- Built-in 8-Bit Timer
- Built-in 8-Bit A/D Converter
- Program Halt Function for Low Power Dissipation
- Stand-by Input Terminal for Data Holding

■ SOFTWARE FEATURES

- An Instruction Set Similar to That of The HMCS6800 Family (Compatible with The HD6805S)
- HMCS6800 Family Software Development System is Applicable

■ PIN ARRANGEMENT (Top View)

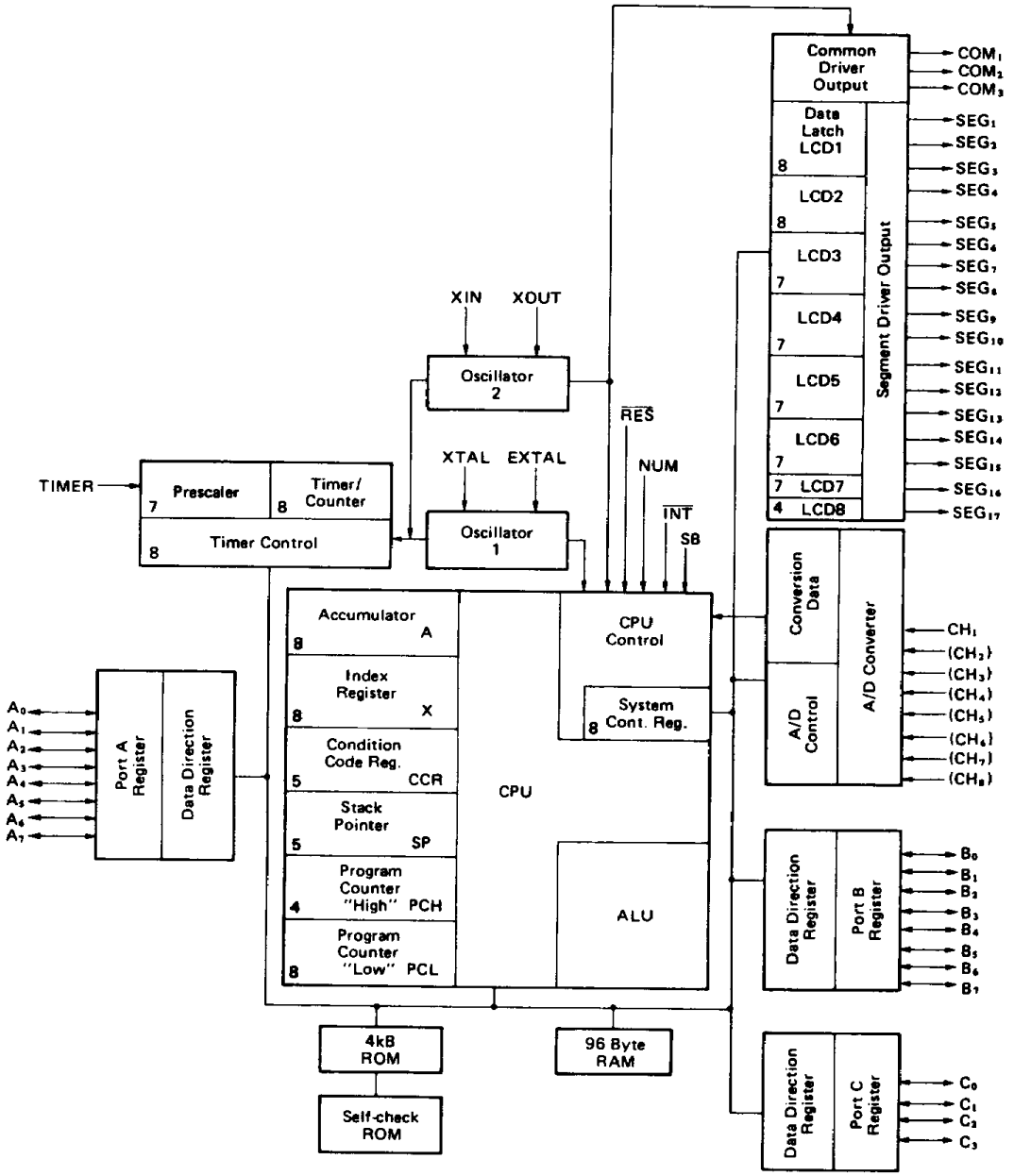




■ PROGRAM DEVELOPMENT SUPPORT TOOLS

- Cross assembler software for use with IBM PCs and compatibles
- In circuit emulator for use with IBM PCs and compatibles

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +5.5	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Output Voltage	V_{out}	-0.3 ~ V_{CC}+0.3	V
Operating Temparature	T_{opr}	-20 ~ +75	°C
Storage Temparature	T_{stg}	-55 ~ +125	°C

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded.
Normal operation should be under recommended operating conditions.
If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS (V_{CC} = 3.0V ±0.8V, V_{SS} = 0V, T_a = -20 ~ +75°C, typ means typical value at V_{CC} = 3.0V, unless otherwise noted.)

• DC CHARACTERISTICS

Item			Symbol	Test Condition	min	typ	max	Unit
Input "High" Level Voltage	XTAL, XIN		V_{IH}	Connect C_L = 0.5μF to V_{CH}	V_{CC}-0.3	–	V_{CC}	V
	RES, INT, SB				0.5V_{CC}+0.9	–	V_{CC}	V
	TIMER				0.8V_{CC}	–	V_{CC}	V
	NUM (Normal Mode)				V_{CC}-0.2	–	V_{CC}	V
Input "Low" Level Voltage	XTAL, XIN		V_{IL}	Connect C_L = 0.5μF to V_{CH}	V_{CC}-2.1	–	V_{CC}-1.8	V
	RES, INT, SB				V_{SS}	–	0.2V_{CC}	V
	TIMER				V_{SS}	–	0.2V_{CC}	V
	NUM (Test Mode)				V_{SS}	–	0.2	V
Self Check Input Voltage	NUM (Self Check Mode)		V_{IM}		0.5V_{CC}-0.2	–	0.5V_{CC}+0.2	V
Input Pull-Up Current	RES (INT: Mask Option) NUM		$-I_{R1}$	V_{CC} = 3.0V, V_{in} = 0V	3	15	30	μA
Input Leakage Current	TIMER, SB		$\|I_{IN}\|$	V_{in} = 0V ~ V_{CC}	–	–	1.0	μA
Current Dissipation	Crystal* Oscillation	During System Operation	I_{CC1}	f = 400kHz No load. Tested after setting up the internal status by self check.	–	100	200	μA
		At Halt			–	40	80	μA
		At Standby			–	2	5	μA
		At A/D Operation			–	200	600	μA
	RC* Oscillation	During System Operation	I_{CC2}	R = 100kΩ No load. Tested after setting up the internal status by self check.	–	120	200	μA
		At Halt			–	60**	100**	μA
		At Standby			–	2	5	μA
		At A/D Operation			–	220	600	μA
Output "Low" Level Voltage	E		V_{OL}	I_{OL} = 30μA	–	–	0.3	V

* Depends on the mask-option.
** 60μA → 30μA and 100μA → 60μA when OSC1 is stopped by Halt.

- **AC CHARACTERISTICS (V_{CC} = 3.0V ± 0.8V, V_{SS} = 0V, Ta = −20 ~ 75°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Operating Clock Frequency		f_{cl}		100	400	500	kHz
Cycle Time		t_{cyc}		8	10	40	μs
Oscillation Frequency * (Resistor Option)		f_{OSCR}	R = 100kΩ ±1%	300	400	500	kHz
External Clock Duty		Duty		45	50	55	%
Oscillation Start Time * (Crystal Option)		t_{OSCf}	C_D = 10pF ±20%, R_S = 1kΩ max	–	–	150	ms
Oscillation Start Time * (Resistor Option)		t_{OSCR}	R = 100kΩ ±1%, Connect C_L = 0.5μF to V_{CH}	–	–	2	ms
Oscillation Start Time (32kHz) *		t_{OSC1}	C_G = 10pF ±20%, R_S = 20kΩ max	–	–	1	s
Internal Capacitance of Oscillator	EXTAL	C_D		–	10	–	pF
	XOUT			–	10	–	pF
Delay Time of Oscillation Delay Time *		t_{DLY}	Selected by mask option	0	–	1	s
Reset Delay Time		t_{RLH}	External Capacitance = 2.2μF	200	–	–	ms
$\overline{RES}$ Pulse Width		t_{RWL}	With 32kHz OSC	48	–	–	μs
			Without 32kHz OSC	$1.5t_{cyc}$ + 1	–	–	μs
$\overline{INT}$ Pulse Width *		t_{IWL}	Without 32kHz OSC	t_{cyc} + 1	–	–	μs
			With 32kHz OSC	32	–	–	μs
TIMER Pulse Width		t_{TWL}	In the case of counter	t_{cyc} + 1	–	–	μs

* Depends on mask-option.

- **PORT CHARACTERISTICS (V_{CC} = 3.0V ± 0.8V, V_{SS} = 0V, Ta = −20 ~ 75°C, unless otherwise noted.)**

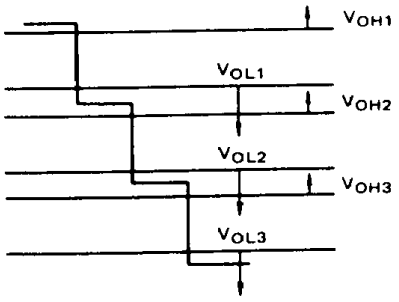
Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Level Voltage *	Port A, B, C	V_{OH}	CMOS Output, I_{OH} = −100μA	V_{CC}−0.3	–	–	V
	Port A, B, C		Key Load CMOS Output I_{OH} = −10μA	V_{CC}−0.3	–	–	V
Output "Low" Level Voltage	Port A, B, C	V_{OL}	I_{OL} = 100μA	–	–	0.3	V
Input "High" Level Voltage	Port A, B, C	V_{IH}		0.8V_{CC}	–	V_{CC}	V
Input "Low" Level Voltage	Port A, B, C	V_{IL}		V_{SS}	–	0.2V_{CC}	V
Input Leakage Current	Port A, B, C	$\|I_{IN}\|$	V_{in} = 0V ~ V_{CC}	–	–	1.0	μA
Input Pull-Up Current *	Port A, B, C	$-I_{R2}$	V_{CC} = 3.0V, V_{in} = 0V	4	20	40	μA

* Depends on mask-option.

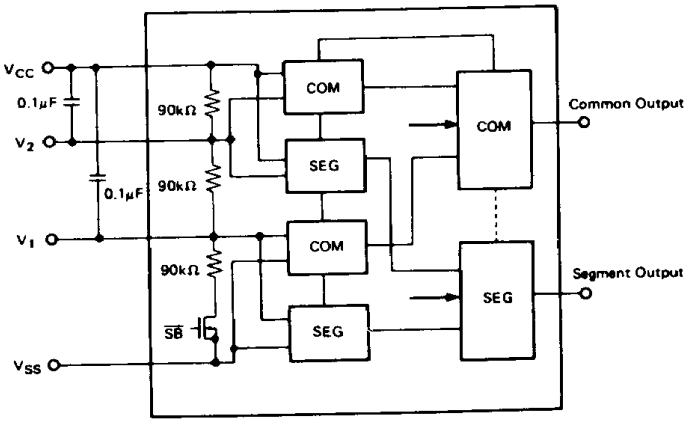
- **LCD DRIVER OUTPUT CHARACTERISTICS (V_{CC} = 3.0V, V_{SS} = 0V, T_a = -20 ~ +75°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Level Voltage	Segment	V_{OH1}	V_1 = 1.00V, V_2 = 2.00V I_{OH} = -1μA	2.8	–	–	V
		V_{OH2}		1.8	–	–	V
		V_{OH3}		0.8	–	–	V
Output "Low" Level Voltage	Segment	V_{OL1}	V_1 = 1.00V, V_2 = 2.00V I_{OL} = 1μA	–	–	2.2	V
		V_{OL2}		–	–	1.2	V
		V_{OL3}		–	–	0.2	V
Output "High" Level Voltage	Common	V_{OH1}	V_1 = 1.00V, V_2 = 2.00V I_{OH} = -5μA	2.8	–	–	V
		V_{OH2}		1.8	–	–	V
		V_{OH3}		0.8	–	–	V
Output "Low" Level Voltage	Common	V_{OL1}	V_1 = 1.00V, V_2 = 2.00V I_{OL} = 5μA	–	–	2.2	V
		V_{OL2}		–	–	1.2	V
		V_{OL3}		–	–	0.2	V
Dividing Resistor		R_{LCD}	Tested between V_1 and V_2	45	90	180	kΩ
Output "High" Level Voltage*	Segment	V_{OH}	In the case of Output Port, I_{OH} = -30μA	V_{CC}-0.3	–	–	V
Output "Low" Level Voltage*	Segment	V_{OL}	In the case of Output Port, I_{OL} = 30μA	–	–	0.3	V

* Depends on mask-option.



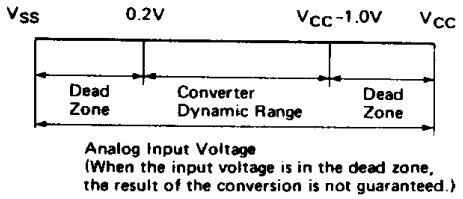
Output Level of SEG and COM



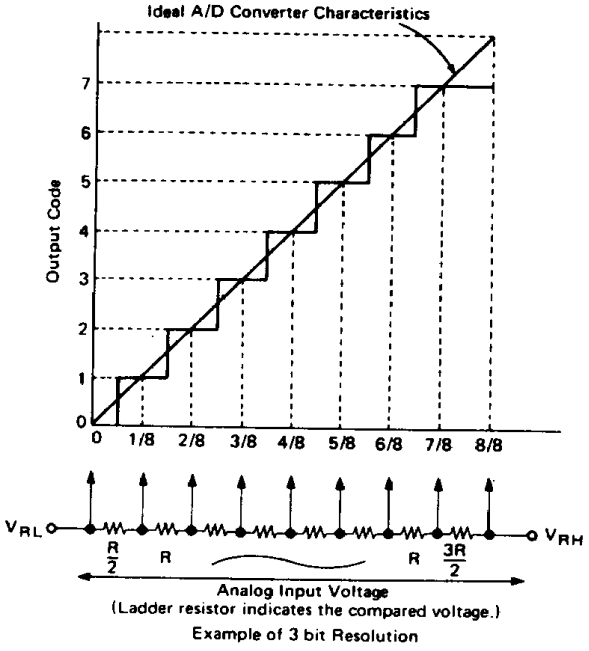
Power Supply Circuit for LCD Display

• A/D CONVERTER CHARACTERISTICS (V_{CC} = 3.0V, V_{SS} = 0V, T_a = -20°C ~ +75°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Conversion Accuracy	Resolution			–	–	8	bit
	Absolute Accuracy		$V_{RL} = 0.2V < V_{in} < V_{RH} = 2.0V$	-2	–	+2	LSB
Reference Voltage	"High" Side	V_{RH}		–	–	V_{CC}	V
	"Low" Side	V_{RL}		V_{SS}	–	–	V
	$V_{RH} - V_{RL}$	ΔV_{REF}		1.8	–	–	V
Input Voltage Range	Input Range	V_{IN}		V_{RL}	–	V_{RH}	V
	Input Dynamic Range	V_{DYN}		0.2	–	V_{CC}-1.0	V
Ladder Resistor ($V_{RH} - V_{RL}$)		R_{HL}		40	80	160	kΩ
Conversion Time		t_{CNV}		2	–	. 4	ms
Programmable Voltage Comparison	Judge Error		$V_{RL} = 0.2V < V_{in} < V_{RH} = 2.0V$	-4	–	+4	LSB
	Judge Time	t_{CMP}		–	–	60	μs



Dynamic Range of the Comparator



Example of 3 bit Resolution

■ SIGNALS

The input and output signals of the MCU are described in the following:

- **V_{CC}, V_{SS}**
  Power is applied to the MCU at these two terminals. V_{CC} is a positive power input port and V_{SS} is grounded.

- **$\overline{INT}$**
  This terminal is used to envoke an external interruption to the MCU. For details, see the information given under the title, "Interruptions" (↓ Negative going edge type).

- **XTAL, EXTAL**
  These are control input ports to the built-in clock circuit. A crystal or a resistor is connected to each of them depending on the degree of stability of the internal oscillation. For the method of using the input terminals, see the information, "Internal Oscillator Option"

- **XIN, XOUT**
  These terminals are connected to a crystal for the oscillator on the time base. A clock operation is possible by using a 32.768kHz crystal. For details, see "Internal Oscillator Option".

- **TIMER**
  An external input terminal at which the internal timer is counted down. For details, see the information, "Timer".

- **$\overline{RES}$**
  Used to reset the MCU. For details, see "Reset".

- **STANDBY (SB)**
  An external input terminal used to stop the MCU and hold data. For details, see "Internal Oscillator Option".

- **A/D Input Terminals ($CH_1 \sim CH_8$)**
  Input terminals for analog voltages needed for A/D conversion. These may also be used as level check inputs under program control. For details, see the information, "A/D Converter".

- **V_{RH}, V_{RL}**
  Reference voltages for A/D conversion are applied to these two terminals. For details, see "A/D Converter"

- **CC_1, CC_2**
  These are not intended for user applications. Open them.

- **NUM**
  This is not intended for user applications. Connect it to V_{CC}.

- **Input/Output Terminals ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_3$)**
  Each of these 20 terminals consists of two 8 bits ports and one 4 bits ports. It may be used as an input or output under program control of the data direction register. For details, see "Input/Output".

- **Liquid Crystal Driver Terminals ($COM_1 \sim COM_3$, $SEG_1 \sim SEG_{17}$)**
  $COM_1 \sim COM_3$ are for driving common electrodes, while $SEG_1 \sim SEG_{17}$ are for driving segments. $SEG_1 \sim SEG_{17}$ can be used as outputs by mask-option and $SEG_{13} \sim SEG_{17}$ can be used as analog inputs for A/D converter by mask-option.

Mixing segment driver with output ports is not available in mask-option.

- **V_1, V_2**
  These are terminals for LCD driver. V_1 and V_2 are connected to V_{CC} via capacitors (0.1μF each). These two terminals can be used as output or analog inputs by mask-option when segments are used as output ports.

- **V_{CH}**
  Output terminal from internal voltage regulator. A capacitor (0.5μF) is connected between V_{CH} and V_{CC}. Don't draw current from this terminal.

- **E**
  System clock output (cycle clock 100kHz typ.)
  This NMOS open-drain output stays at "Low" level when the MCU is in halt mode, standby mode or reset.

■ MEMORY

The MCU memory is configured as shown in Figure 1. During the processing of an interrupt, the contents of the MCU resisters are pushed onto the stack in the order shown in Figure 2. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will cause only the program counter (PCH, PCL) contents to be pushed onto the stack.

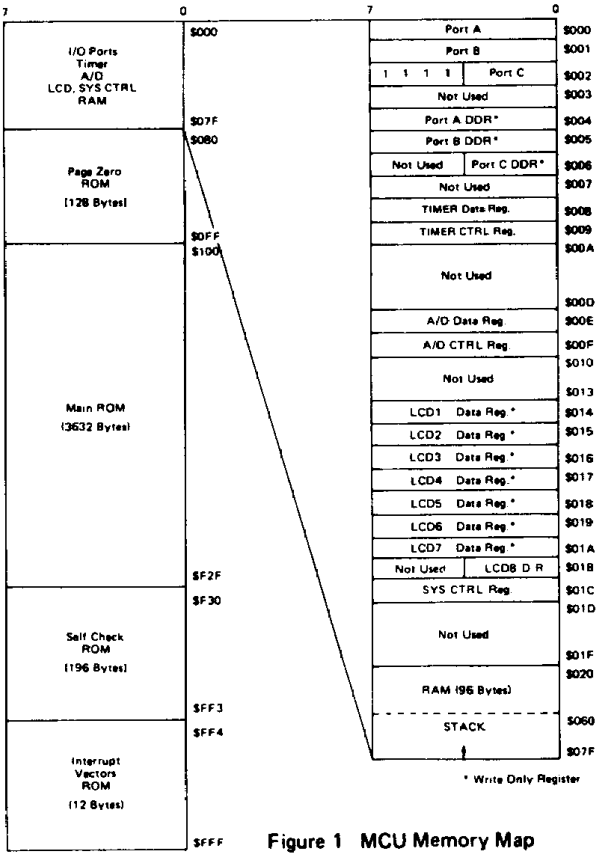


Figure 1 MCU Memory Map

(Cautions)

It is not possible to change the contents of the Write Only Register (For example, the Data Direction Register of the I/O port) of the HD63L05F1 by applying the Read/Modify/Write instructions, BSET, or BCLR.

For preventing the system from wild running, don't read the Not Used area of the memory map.

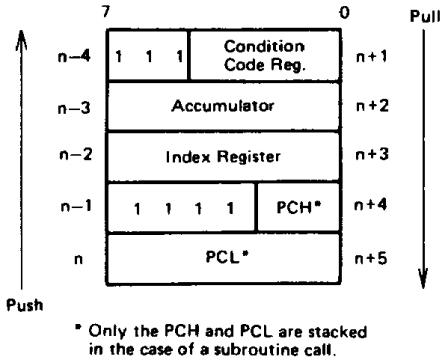


Figure 2 Interruption Stack Sequence

■ REGISTER

The CPU has five registers that can be operated by the programmer. They are shown in Figure 3.

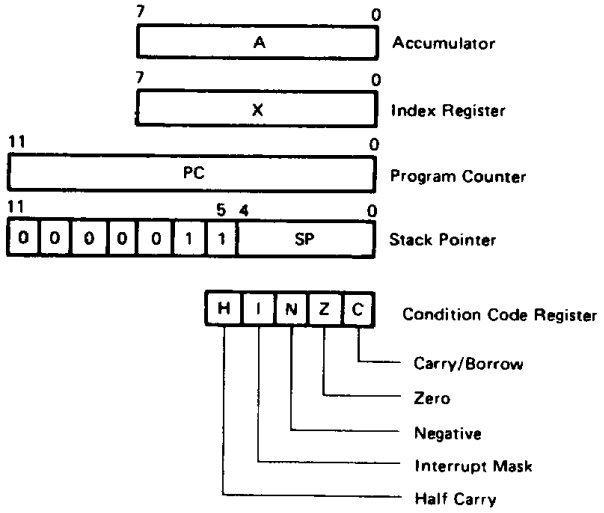


Figure 3 Programming Model

• Accumulator (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

• Index Register (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations and data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage register.

• Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The most significant bits of the stack pointer are permanently set to 0000011. During a MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 which allow the programmer to use 15 levels of subroutine calls.

• Condition Code Register (CC)

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

Half Carry (H)

Used during arithmetic operations (ADD and ADC) to indicate that a carry occurred between bit 3 and bit 4.

Interrupt (I)

This bit is set to mask the internal interrupts and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set, it is latched and will be processed as soon as the interrupt bit is reset.

(Note)

CLI (clear interrupt mask bit) is used to allow the interruption from the instruction after next. SEI (set interrupt mask bit) masks the interruption from next instruction.

Negative (N)

Used to indicate that the result of the last arithmetic, logical of data manipulation was negative (bit 7 in result equal to logical one).

Zero (Z)

Used to indicate that the result of the last arithmetic, logical of data manipulation was zero.

Carry/Borrow (C)

Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instruction, shifts, and rotates.

■ SYSTEM CONTROL REGISTER

Apart from the registers for program operation explained above, there is a register that controls system operation. Its configuration is shown in Figure 4.

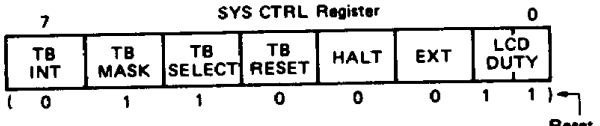


Figure 4 System Control Register Configuration

A Time Base counter is built in the MCU to generate two kinds of time base interrupts (1 second cycle and 1/16 second cycle). Clock signal to this counter is provided from the OSC1 or OSC2 depending on the mask-option. This counter is a frequency divider behind the 32 kHz oscillator.

- **Time Base Interrupt Request Flag (TB INT)**

Stores an interrupt request from the time base which is selected by the TB select bit and is cleared by system reset or by software. If the TB MASK bit or I (Interrupt bit in the CCR) is set, the interrupt request is not acknowledged. Only logical "0" can be written into this bit by software.

- **Time Base Interrupt Mask (TB MASK)**

If this bit is set, an interrupt request from the time base is not acknowledged.

- **Time Base Select Bit (TB SELECT)**

This bit selects the time base. In logical "1", an interrupt from the 1-second cycle time base is acknowledged. In logical "0", 1/16 second cycle time base is acknowledged.

- **Time Base Reset Bit (TB RESET)**

This bit resets the frequency divider behind the 32kHz oscillator. When this bit is set, one shot reset pulse is generated by the hardware. Then it resets the frequency divider and after that, the frequency divider restarts. The CPU always reads this bit as logical "0"

Since the frequency divider also provides the system clocks to the A/D converter and LCD drivers etc., writing "1" to "TB RESET" bit during execution of A/D converter and TIMER (when ϕ_{32k} is selected) causes different data from the correct result and writing "1" to this bit causes flicker of the LCD display.

- **Halt (HALT)**

Used to halt the CPU. When this bit is set, the registers are saved onto the stack in the same sequence as interrupt processing. After all registers have been saved, the CPU halts and is wait-for-interrupt state.

If this bit is reset by an external interrupt or an internal interrupt, the CPU restarts operating. By using the Halt function with Time Base Interrupt, the CPU can operate intermittently itself.

- **EXT**

When the form of output port is selected by DUTY selecting bit and the mask-option, ϕWRITE is available at the specified terminal (SEG_1 to O_{19}) according to the designation of pin location. ϕWRITE clock can be got on every writing data into LCD register 1 and be used as the write clock in the case of transferring data of LCD register 1 to the outside. Normally, EXT must be reset.

- **Duty Select Bit (LCD DUTY)**

The LCD drive signal is based on 1/3 bias – 1/3 duty. However, there are switching circuits built in for static drive signal and output ports. For details, see the information given in "LCD Circuit".

(Note)

The EXT bit and the LCD DUTY bits have to be initialized in 1 milli second from the start of CPU operation when the static drive signal or output port is selected.

■ TIMER

The MCU timer circuitry is shown in Figure 5. The 8-bit counter is loaded under program control and counts down toward zero as soon as the clock input is applied. When the timer reaches zero the timer interrupt request bit (bit 7) in the timer control register is set. The CPU responds to this interrupt by saving the present CPU state in the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The interrupt bit (I bit) in the condition code register will also prevent a timer interrupt from being processed.

The clock input to the timer can be from an external source applied to the TIMER input terminal (active at negative edge) or it can be the internal signal (ϕ_2 or ϕ_{32k}). When the internal clock signal is used as the source, the clock input is gated by the input applied to the TIMER input terminal; this permits easy measurement of its pulse width. ϕ_2 is provided from OSC1 and the frequency is 1/4 of OSC1. ϕ_{32k} is provided from OSC1 (the frequency is 1/12 of OSC1) or OSC2 (32.768 kHz crystal) depending on the mask-option. If the OSC1 continues to oscillate during the halt mode, 32.768 kHz crystal is selected as the clock source or external clock is applied, the timer can be active in the halt mode. Note that the timer operation is asynchronous to the CPU when the mask-option which the OSC1 stops oscillating in the halt mode is selected.

A 7-bit prescaler is provided to extend the timing interval up to a maximum of 128 counts before being applied to the timer. The number of prescaling counts can be program controlled by the lower 3 bits within the TIMER CTRL register. The timer continues to count past zero and its present count can be monitored at any time by monitoring the TIMER Data register. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At the time of resetting, the prescaler and the counter are all initialized to logical "1". The timer interrupt request bit is cleared and the timer interrupt mask bit is set. The timer interrupt request bit (bit 7 of TIMER CTRL Register) is set to logical "1" when timer count reaches zero, and is cleared by program or by system reset. Only logical "0" can be written into this bit by program. The bit 6 of Timer Control Register is writable by program. Both of these bits can be read by CPU.

■ RESETS

The MCU can be reset either by initial power-up or by the external reset input ($\overline{RES}$). All the I/O ports are initialized to Input mode (DDRs are cleared) during reset.

Upon power-up, a minimum of 150 milliseconds is needed before allowing the reset input to go "High". This time allows the internal oscillator (OSC1) to stabilize. Connecting a capacitor to the $\overline{RES}$ input as shown in Figure 8 will provide sufficient delay.

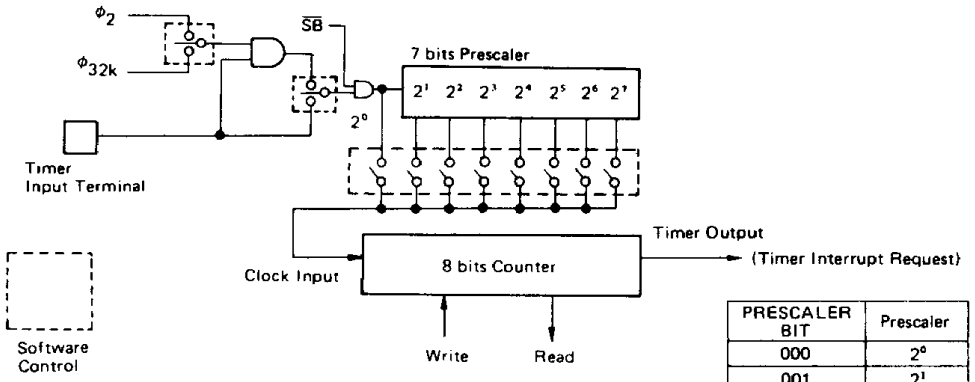


Figure 5 Timer Block Diagram

PRESCALER BIT	Prescaler
000	2^0
001	2^1
010	2^2
011	2^3
100	2^4
101	2^5
110	2^6
111	2^7

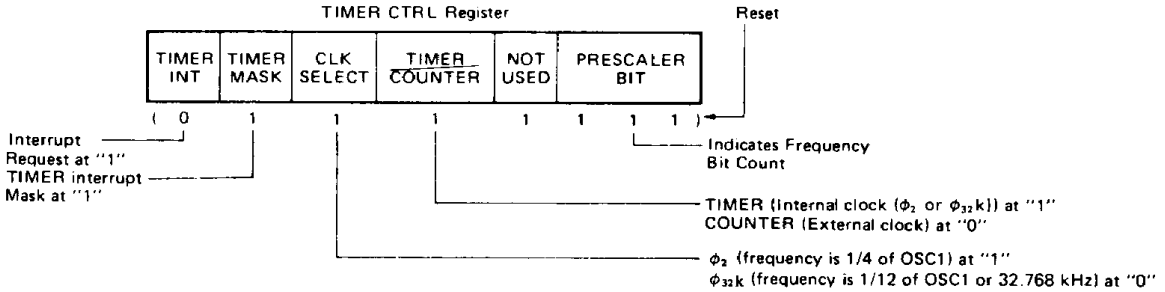


Figure 6 Timer Control Register Configuration

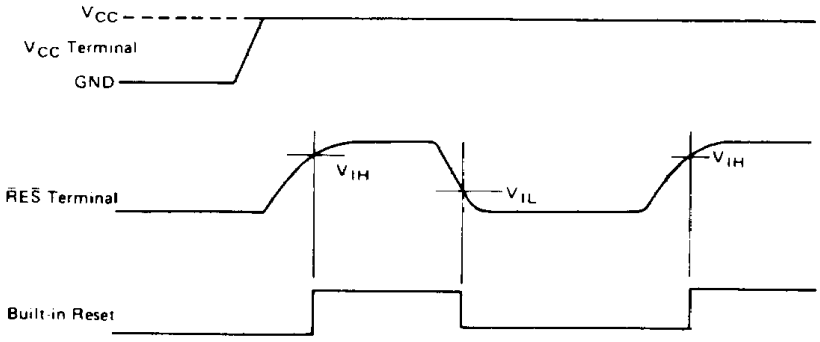


Figure 7 Application of Power and Reset Timing

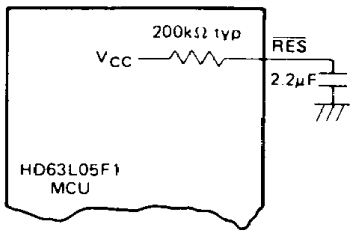
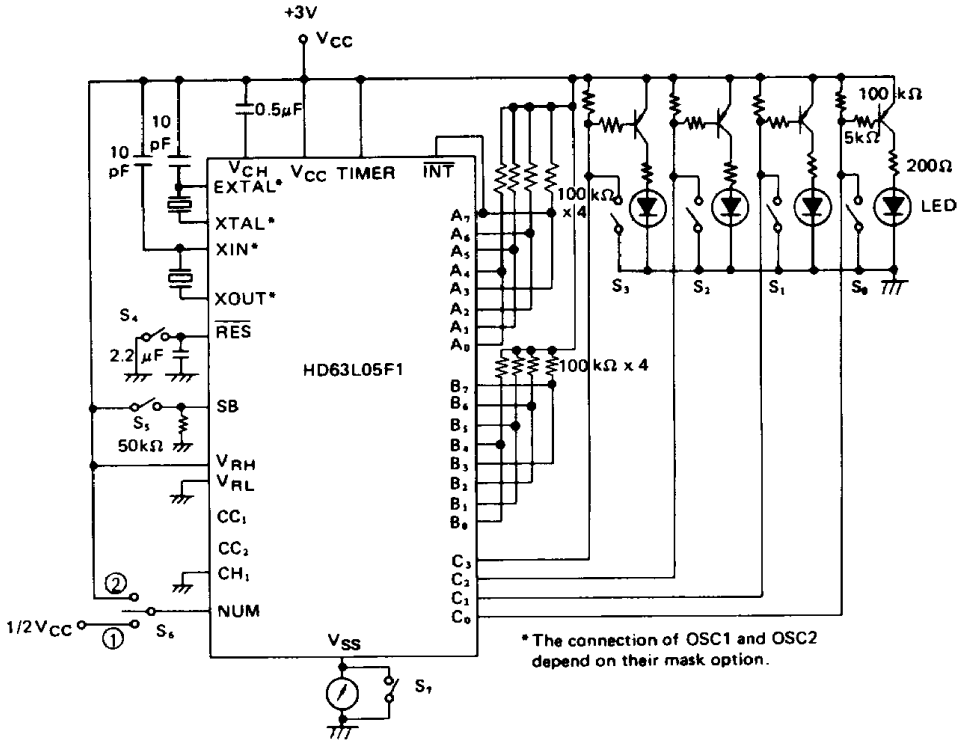


Figure 8 Input Reset Delay Circuit

■ SELF CHECK

The self check capability of the MCU provides an internal check to determine if the port is functional. Connect the MCU as shown in Figure 9 and monitor the output of port C bit 3 for an oscillation of approximately 0.5Hz. This self check capability also provides the internal state of the MCU to measure the LSI current. After a system reset, the MCU goes into each current measurement mode by the combination of the control switches. The LSI current can be measured when the NUM is returned to V_{CC} after setting of the current mode.



		Selection of Switch							
		S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
LSI Function		X	X	X	X	X	X	①	O
LSI Current	During operation	O	X	X	X	O→X	X	①→②	X
	Halt	O	O	O	X	O→X	X	①→②	X
	A/D	O	O	X	X	O→X	X	①→②	X
	Standby	O	O	O	X	O→X	X→O	①→②	X

X : OFF O : ON →: Change the state

Figure 9 Self Check Connections

■ INTERNAL OSCILLATOR OPTIONS

The MCU incorporates two oscillators: Oscillator 1 for system clock supply and Oscillator 2 for peripheral modules such as time base, A/D converter, LCD drivers, etc..

• Oscillator 1 (OSC1; XTAL, EXTAL)

The internal oscillator circuit can be driven by an external crystal or resistor depending on the stability. A manufacturing mask option is available to provide better matching between the external components and the internal oscillator. The oscillator 1 can stop when power is applied in Standby mode. Figure 10 shows the connection. A resistor selection graph is given in Figure 11.

- **Oscillator 2 (OSC2; XIN, XOUT)**

Clocks for time base, LCD drivers, an A/D converter and a timer are supplied by the OSC2 (32.768kHz crystal). In Halt mode, OSC2 and frequency divider operate and permit the operation of the peripheral modules with low power consumption. In Standby mode, the frequency divider is in reset state but only OSC2 keeps on running to control the delay time required when the MCU is released from Standby mode. Figure 12 shows the connection and the relation between oscillator 1 and oscillator 2 is shown Figure 13 and Table 1.

When OSC2 is not available in an user system, clocks for time base, LCD drivers, a A/D converter and a timer are supplied by the OSC1 through frequency divider. When OSC2 is not available or crystal option is selected for OSC1, OSC1 can not be stopped in Halt mode.

Only when CR option is selected for OSC1 and OSC2 is available in an user system, OSC1 can be stopped in Halt mode. (Note)

The accuracy of the time base (1 sec, 1/16 sec) is kept only when OSC2 is 32.768kHz crystal oscillator.

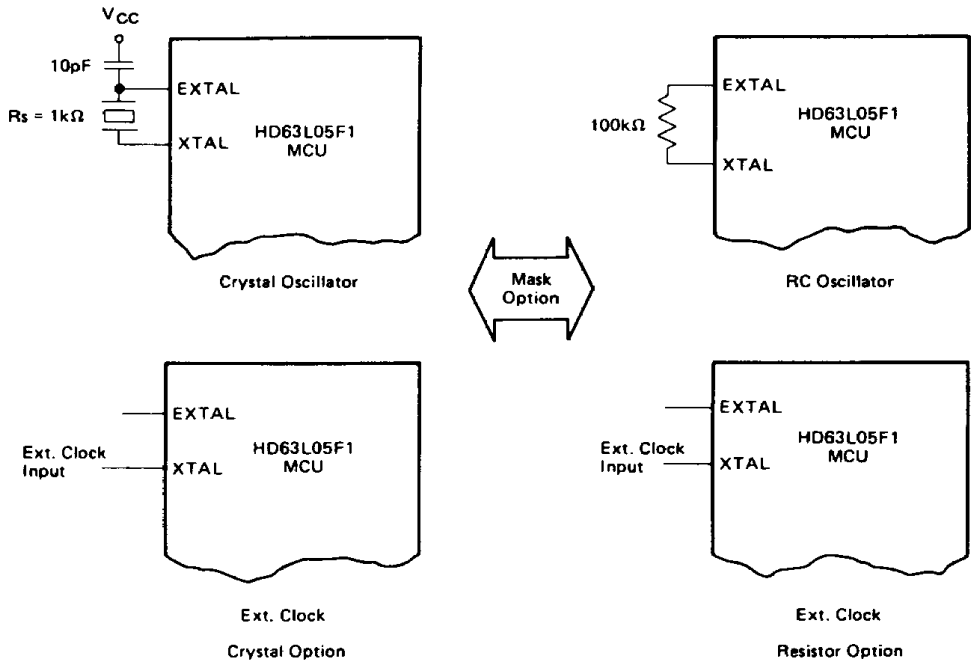


Figure 10 Mask Option for Oscillator 1

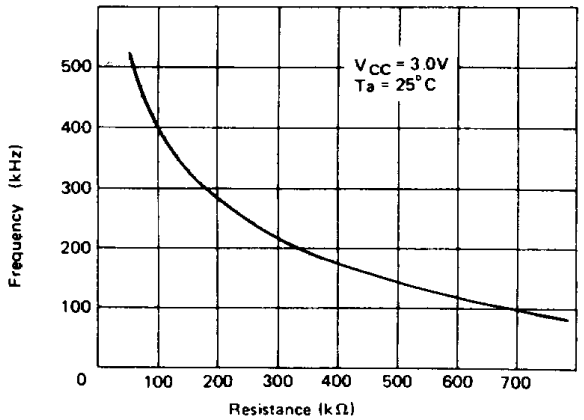


Figure 11 Typical Resistor Selection Graph

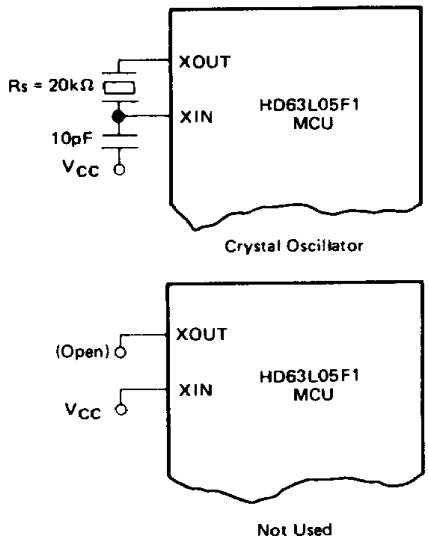


Figure 12 Connection of Oscillator 2

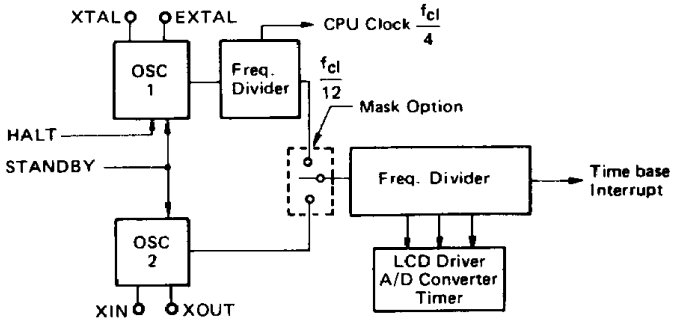


Figure 13 Relation between Oscillator 1 and Oscillator 2

Table 1 Oscillator 2 Mask-option and System Operation

Mask Option	When OSC1 is Crystal						When OSC1 is RC					
	OSC2 Not Available			OSC2 Available			OSC2 Not Available			OSC2 Available		
System / State	OSC1	CPU	Peripheral	OSC1	CPU	Peripheral	OSC1	CPU	Peripheral	OSC1	CPU	Peripheral
During System Operation	O	O	O	O	O	O	O	O	O	O	O	O
At Halt	O	X	O	O	X	O	O	X	O	O or X (mask option)	X	O
At Standby	X	X	X	X	X	X	X	X	X	X	X	X

(NOTE) O ..... run X ..... stop

Table 2 Mask-options of Oscillation Circuits and the Delay Time

Type of OSC1	Use of OSC2	Condition		Delay Time of Restart (second)			
				0	1/16	1/2	1
Crystal Option	Used	Standby mode	Used	X	X	O	O
			Not used	O	O	O	O
	Not used	Standby mode	Used	X	X	O	O
			Not used	O	O	O	O
CR Option	Used	Oscillation of OSC1 at HALT	Stop	O	X	X	X
			Continue	O	O	O	O
	Not used	Oscillation of OSC1 at HALT	Stop	X	X	X	X
			Continue	O	O	O	O

Note) Combinations of the mask-option indicated X is not available.

■ **STANDBY**

When the STANDBY (SB) terminal becomes "High" level, the MCU goes into standby mode at its instruction fetch cycle. On standby mode, only 32 kHz oscillator (OSC2) keeps on running while the others are stopped with holding the current data except A/D converter, timer, and time base. Restarting of the MCU from standby mode is controlled by the Delay Time which is available by counting the OSC2 oscillation or 1/12 frequency of the OSC1 in frequency divider after the STANDBY terminal turned to "Low" level. Therefore, the CPU restarts operation from the previous state after the Delay Time (0 sec, 1/16 sec, 1/2 sec, or 1 sec), and the accuracy of the Delay Time

is kept when OSC2 is 32.768 kHz crystal oscillator. When 1/12 frequency of OSC1 is provided to the frequency divider, the Delay Time depends on the stability of OSC1 after restarting from standby mode and is not acculate.

- **Delay Time**

Since OSC1 stops in standby mode, it is needed to inhibit restarting of CPU untill the OSC1 oscillation is stabilized after the STANDBY terminal has turned to "Low" level. To take this stabilizing time of OSC1, user can select the Delay Time out of 0 sec, 1/16 sec, 1/2 sec or 1 sec by mask-option depending on a combination in the Table 2. STANDBY terminal has to be kept at "Low" when resetting the MCU and has to be kept at "Low" during the Delay Time. Starting of the MCU by reset is also controlled by the Delay Time.

- **INTERRUPTS**

There are six different interrupts to the MCU: external interrupt via external interrupt terminal ($\overline{INT}$), internal timer interrupt, interrupt by termination of A/D conversion, time base interrupt, and software interrupt by an instruction (SWI).

When any interrupt occurs, processing is suspended, the present MCU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. The interrupt service routines normally end with a return from interrupt instruction (RTI) which allows the MCU to resume processing of the program prior to the interrupt. Table 3 provides a listing of the interrupts, their priority, and the vector address that contains the starting address of the appropriate interrupt routine.

Figure 14 shows the system operation flow, in which the portion surrounded with dot-dash lined contains interruption execution sequence.

(Note)

A clear interrupt bit instruction (CLI) allows to suspend the processing of the program by an interruption after execution of the next instruction while a set interrupt bit instruction (SEI) inhibits any interrupts before execution of the next instruction. When a mask bit of a control register is cleared by an instruction, interruption is allowed before execution of the next instruction.

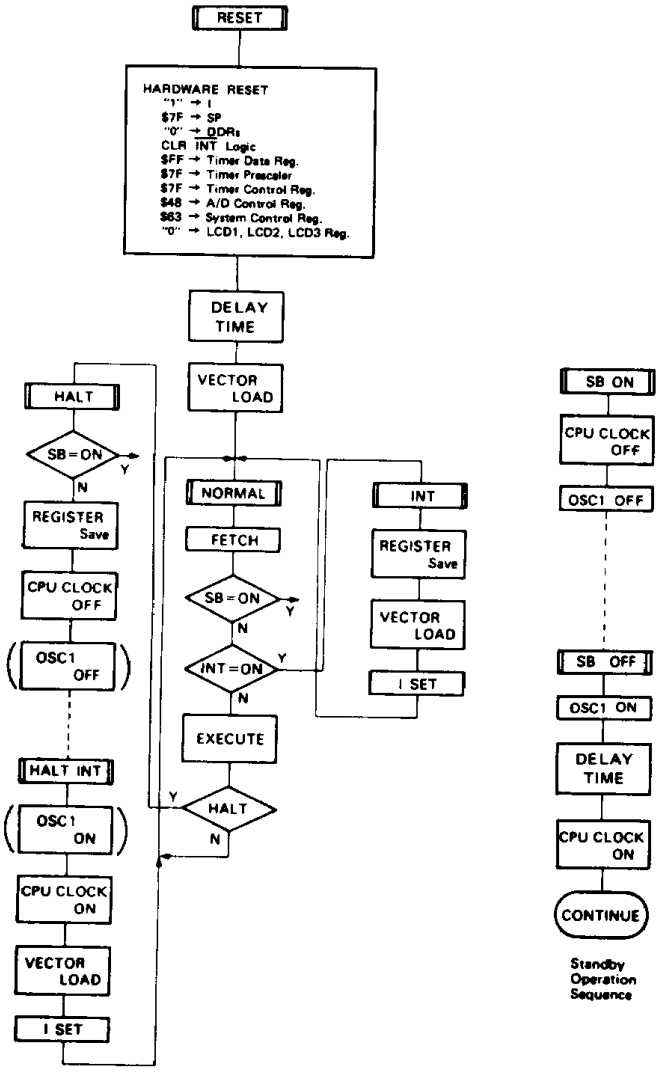


Figure 14 System Operation Flowchart

Table 3 Interruption Priority

Interruption	Priority	Vector Address
$\overline{RES}$	1	$FFE, $FFF
SWI	2	$FFC, $FFD
$\overline{INT}$	3	$FFA, $FFB
TIMER	4	$FF8, $FF9
A/D	5	$FF6, $FF7
TIME BASE	6	$FF4, $FF5

- **Acknowledging an INT in Halt mode**

In HALT mode, the CPU is not operating but the peripherals are operating. When an interruption is acknowledged, the CPU is activated and executes interruption service matching the interruption condition by means of vectoring.

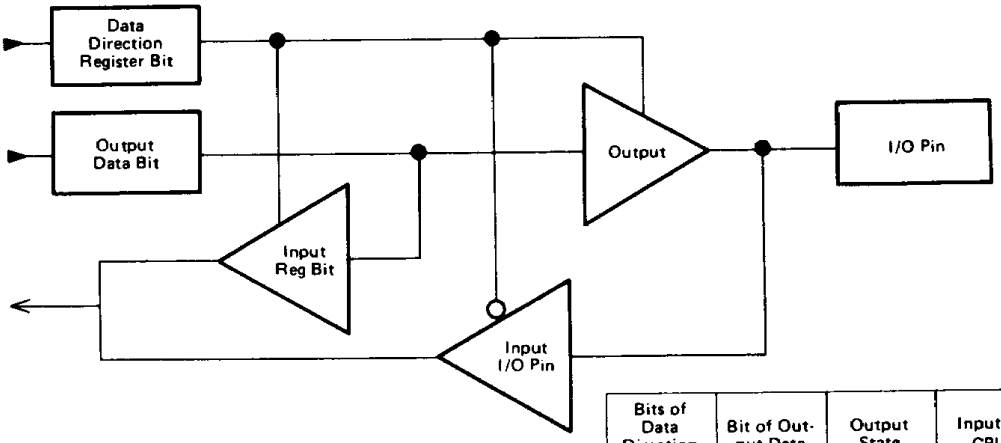
- **Acknowledging an INT in Standby mode**

In Standby mode, the system is not operating with power supplied to it, therefore, any interruption request (including $\overline{RES}$) is not acknowledged.

■ INPUT/OUTPUT

There are 20 input/output terminals, which are program controlled by data direction registers for use as either input or output. If an I/O port has been programmed as an output and is read, then the latched logical level data is read even though the output level changes due to the output load.

If a port is to be used as an input terminal, the user must specify whether or not it will be equipped with a pull-up PMOS. Figure 15 shows the port I/O circuit.



Bits of Data Direction Register	Bit of Output Data	Output State	Input to CPU
1	0	0	0
1	1	1	1

Figure 15 Port I/O Circuit

● Port Configuration

Figure 16 shows the port configuration.

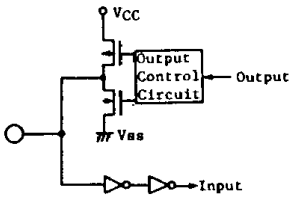
Each pin can individually select four types of configurations by mask option. The following explains these.

A: This port configuration is of general type.

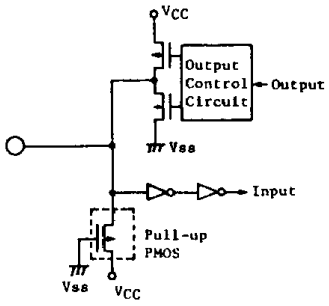
B: This configuration contains pull-up resistance to prevent input from floating when using the port as an input. Select this configuration in such cases as connecting the switch to port directly.

C: This port configuration is for driving key matrix. If key matrix is driven under this port configuration, the CPU can read the necessary data precisely, even if more than two keys are pushed simultaneously.

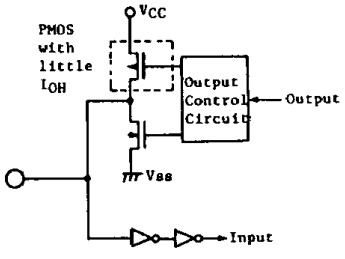
D: This port configuration is NMOS open drain output, which is convenient for removing wired OR or driving transistor circuit. Users should be careful of the maximum output voltage of open drain output. Output voltage of open drain output covers from "OV" to "V_{CC}".



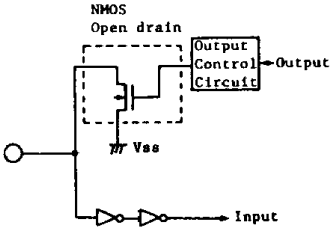
(A) Standard port



(B) Port with Input pull-up MOS



(C) Port for Key matrix



(D) Open drain output

■ A/D CONVERTER

The MCU incorporates an 8 bits A/D converter based on the resistor ladder system. Figure 17 shows its block diagram.

The "High" side of reference voltage is applied to V_{RH}, while the "Low" side of reference voltage is applied to V_{RL}. The reference voltage is divided by resistors into voltages matching each bit, which is compared with analog input voltage for A/D conversion. As the analog input voltage is applied to the MOS gate of the comparator through the analog multiplexer, this voltage comparison system achieves high input impedance.

The A/D Data Register stores the results of an A/D conversion or can be set 8 bit data for programmed comparator. These functions are controlled by software-controlled A/D CTRL Register. The result of A/D conversion is not assured if the conversion is interrupted by STANDBY. Figure 18 shows the configuration of the A/D control register.

• A/D Interrupt Request Flag (A/D INT)

The A/D INT bit is set to logical "1" after completion of A/D conversion and is cleared by program or by system reset. Only logical "0" can be written into this bit by software.

• A/D Interrupt Mask (A/D MASK)

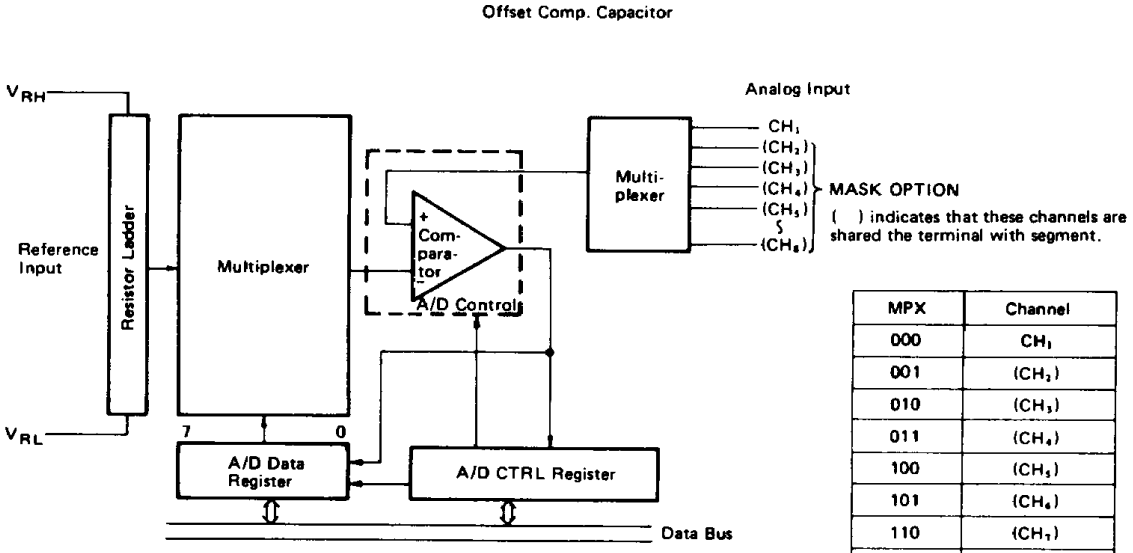
If this bit is set, interrupt from the A/D converter is not acknowledged. This bit can be written by program.

• A/D Conversion Flag (CNV)

To start auto A/D conversion, set this bit to logical "1". During conversion, data of this bit stays at "1". The bit is automatically reset to "0" when the auto A/D conversion ends. In auto A/D conversion, supply voltage is applied to the comparator only when CNV = "1". The digital data which is obtained by the A/D conversion is held in the A/D Data Register. This data is reset when the CNV is set to "1" again.

• A/D Operation Mode Select Bit (Auto/Program)

Used to select either auto-run 8 bits A/D conversion or 8 bit programmed comparator operation (Auto 8 bits A/D conversion at "0").



MPX	Channel
000	CH_1
001	(CH_2)
010	(CH_3)
011	(CH_4)
100	(CH_5)
101	(CH_6)
110	(CH_7)
111	(CH_8)

Figure 17 8 Bits A/D Converter Block Diagram

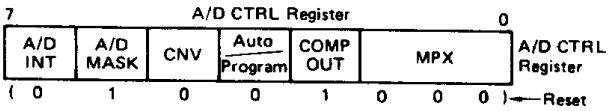


Figure 18 A/D Control Register Configuration

- **Comparator Output (COMP OUT)**

The result of comparator operation under program control can be read from this bit (Logical "1" means that input voltage is higher than programmed reference voltage).

- **Analog Input Channel Select Bits (MPX)**

Used to select 8-channel analog inputs. The multiplexer is an analog switch based on CMOS. Note that the analog inputs from CH_2 to CH_8 are mask option while CH_1 is exclusive.

When 1/3 bias – 1/3 duty or static LCD is used, CH_7 and CH_8 are not available because these two terminals are used for LCD power supply as V_1 and V_2.

■ LCD CIRCUIT

The system configuration of the LCD circuits is shown in Figure 19. Segment data for display are stored in data registers LCD1 to LCD8. Since the circuits are connected to the output terminals via pin location block, the user may specify a combination of data to be multiplexed to the segment output terminals.

The bit data of the LCD register is combined with the timing clock (ϕ_1, ϕ_2 or ϕ_3) and three combined bit data are gathered to make a segment output data for 1/3 bias – 1/3 duty driving in the pin location block. In case of static LCD drive or output port, timing is always fixed at ϕ_1 (always "High") and one bit data of the LCD register is transferred for an output terminal.

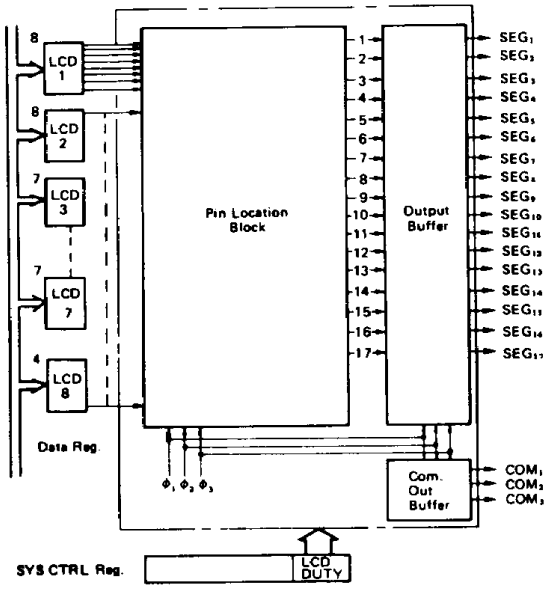
Note that the output terminals from SEG_{13} to SEG_{17} are mask option while the others (SEG_1 to SEG_{12}) are always available when the Duty bits are "01" or "11".

When the form of output port is selected by Duty bit ("00"), ϕWRITE can be got every time data is written into LCD1 register in the case that EXT bit is "1". As LCD1 register has 8 bits latches, it is easy to transfer the internal 8 bits data to external devices via output ports, with automatically generated write clock ϕWRITE. The cycle clock pulse can be also available as an internal data source for the output terminal when output port is selected as 1/4 OSC1.

Assignment of segment terminals to the bits of the LCD data register, including the case where they are used as output terminals, is to be specified by the user when he orders masks. In case of static LCD or output ports, only LCD1, LCD2, and LCD3 are allowed to be used. These registers are initialized at "0" by system resetting.

■ LIQUID CRYSTAL DRIVER WAVEFORMS

The LCD circuit is based on 1/3 bias – 1/3 duty driving. Figure 20 shows the common electrode output signal waveforms (COM_1, COM_2, COM_3), segment signal waveforms (SEG_1 to SEG_{17}) and LCD bias waveforms (between COM and SEGMENT).



System Cont. Duty	Contents of SEG_1 ~ SEG_{17}
00	OUTPUT PORT
01	STATIC LCD
10	———
11	1/3 Bias 1/3 Duty LCD

Note)
Both of mask-option and software control are needed to specify the contents of SEG_1 ~ SEG_{17}.

Figure 19 LCD Circuit System Configuration

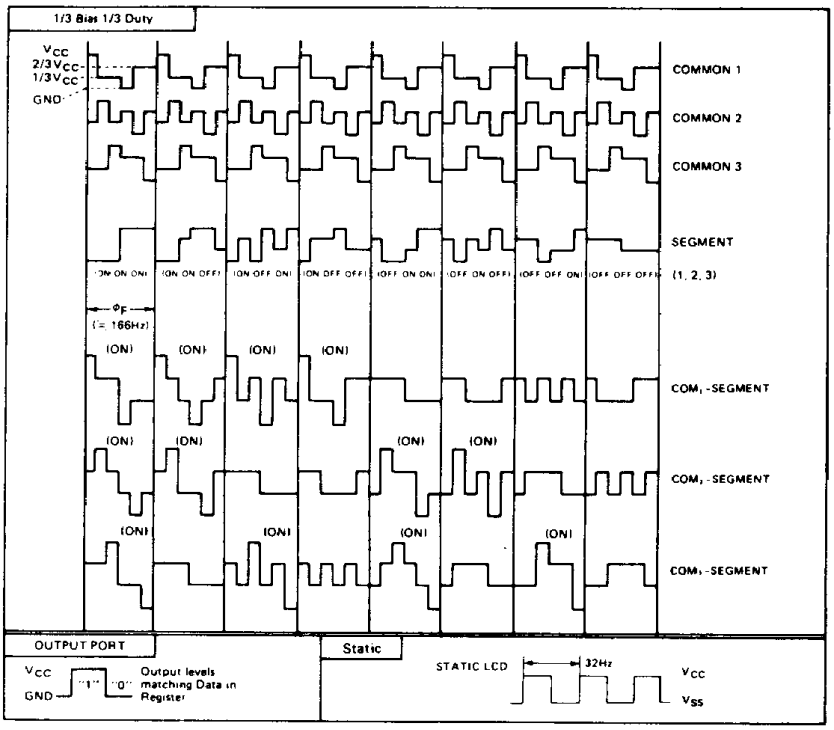


Figure 20 LCD Driving Waveforms

■ BIT MANIPULATION

The MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction registers) with a single instruction (BSET, BCLR). Any bit in the page zero read only memory can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

(Note)

It is needed to pay attention to the system control register, the timer control register, and A/D control register when BSET, BCLR, or Read/Modify/Write instructions are applied to them. If own interrupt request occured onto the interrupt request bit (bit 7) of the control register between read cycle and write cycle of these instructions, the bit 7 might be cleared in the write cycle and not acknowledged by CPU.

■ ADDRESSING MODE

There are 10 addressing modes available to the MCU for programming. Familiarize yourself with these modes by reading the information and referring to the diagrams that follow.

• Immediate

See Figure 21. In immediate addressing mode, constants that will not change during execution of a program are accessed. The instruction used for that purpose has a length of 2 bytes. The effective address (EA) is PC. The operand is fetched from the byte that follows the OP code.

• Direct

See Figure 22. In direct addressing mode, the address of the operand is contained in the second byte of the instruction. The user can gain direct access to the LSB 256 of memory. All RAM bytes, I/O registers, and 128 bytes of ROM are located on page 0 in order to utilize this useful addressing mode.

• Extended

See Figure 23. The extended addressing mode is used for referencing to all addresses of memory. The EA consists of the contents of the two bytes that follow the OP code. The instruction used for extended addressing has a length of 3 bytes.

• Relative

See Figure 24. Only Branch instructions are used in relative addressing mode. When a branching takes place, the contents of the byte next to the OP code are added to the program counter. EA = (PC) + 2 + Rel., where Rel. indicates signed 8 bits data at the address following the OP code. When no branching takes place, Rel. = 0. When a branching occurs, the program jumps to any byte of +129 to -127 of the current instruction. The length of the Branch instruction is 2 bytes.

- **Indexed (without Offset)**

See Figure 25. In this addressing mode, the lower 256 bytes of memory are accessed. The length of the instruction used for this mode is one byte. The EA consists of the contents of the index register.

- **Indexed (8 Bits Offset)**

See Figure 26. The EA consists of the contents of the byte following the OP code, and the contents of the index register. In this mode, the lower addresses of memory up to 511 can be accessed. Two bytes are required for the instruction.

- **Indexed (16 Bits Offset)**

See Figure 27. The EA consists of the contents of the two bytes following the OP code, and the contents of the index register. In this mode, the whole of the memory can be accessed. The instruction using this addressing mode has a length of 3 bytes.

- **Bit Set/Clear**

See Figure 28. This addressing mode can be applied to any instruction that permits any bit on page 0 to be set or cleared. The byte following the OP code indicates an address within page 0.

- **Bit Test, Branch**

See Figure 29. This addressing mode can be applied to instructions that test bits at the first 256 addresses ($00 to $FF) and are branched by relative qualification. The byte to be tested is addressed by the contents of the address next to the OP code. The individual bits of the byte to be tested are designated by the lower 3 bits of the OP code. The third byte indicates a relative value that is to be added to the program counter when a branch condition is satisfied. The instruction has a length of 3 bytes. The value of the bit that has been tested is written at the carry bit of the condition code register.

- **Implied**

See Figure 30. There is no EA for this mode. All information needed for execution of instructions is contained in the OP code. Operations that are carried out directly on the accumulator and index register are included in the implied addressing mode. In addition, the SWI and RTI instructions are also included in the group of this operation. The instruction using this addressing has a length of one byte.

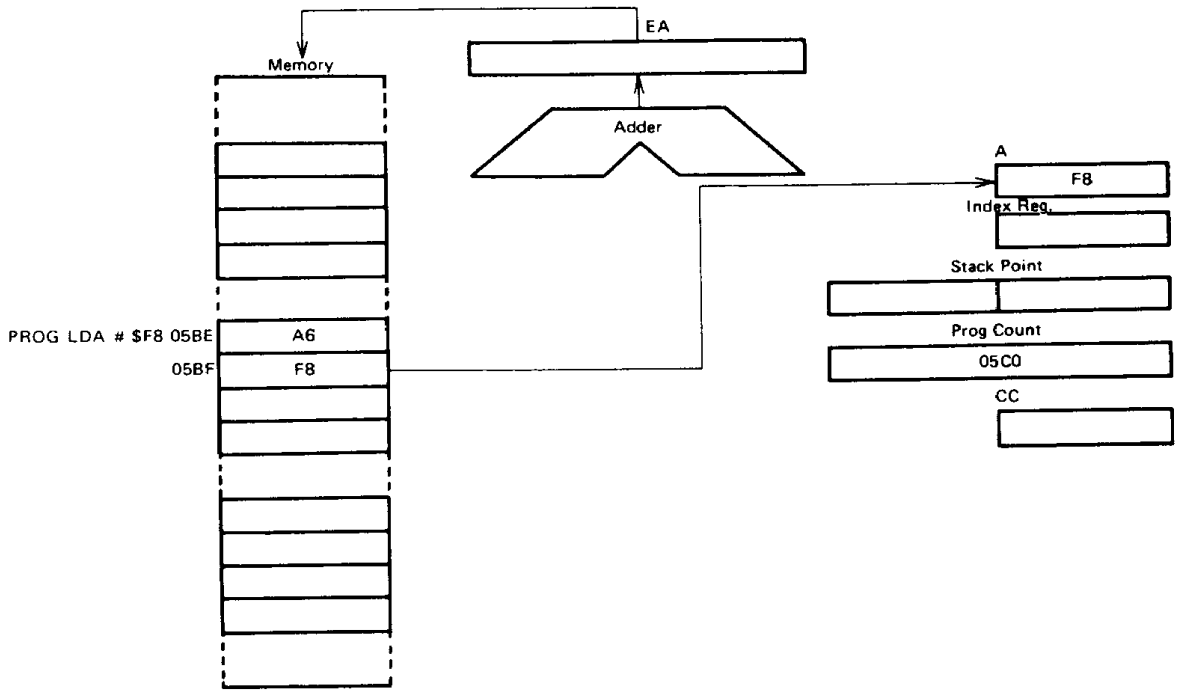


Figure 21 Example of Immediate Addressing

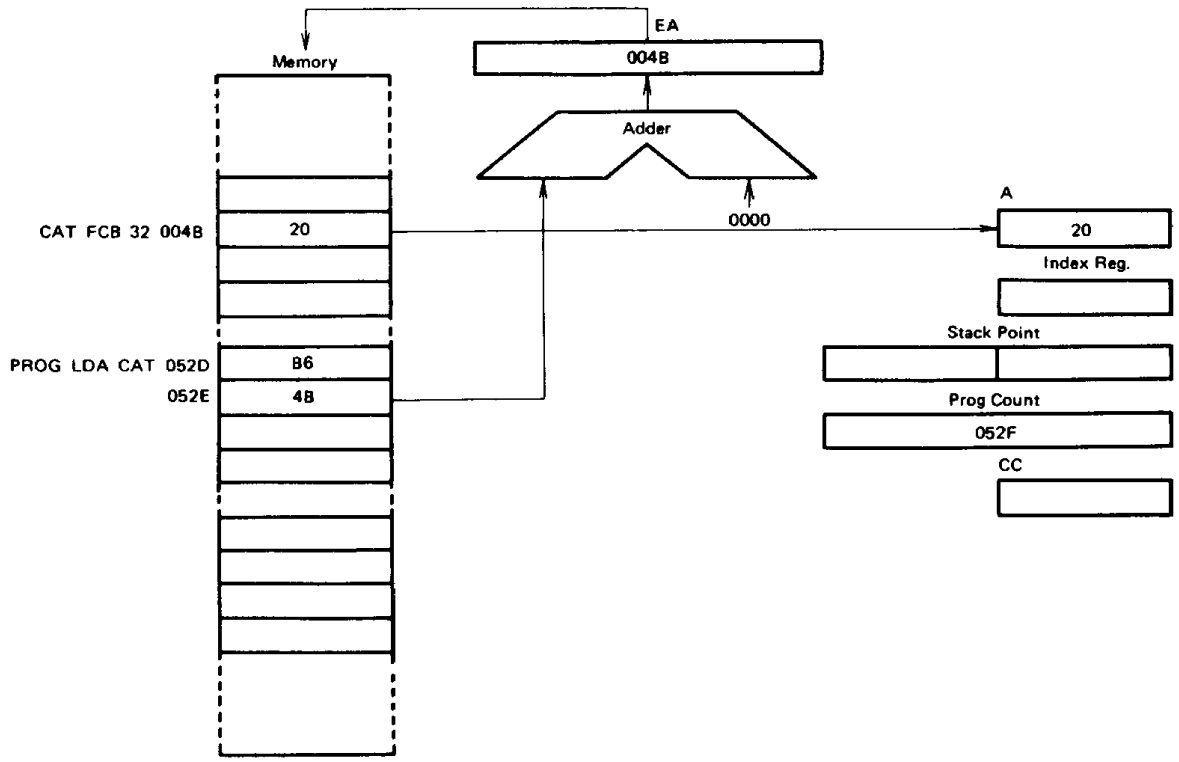


Figure 22 Example of Direct Addressing

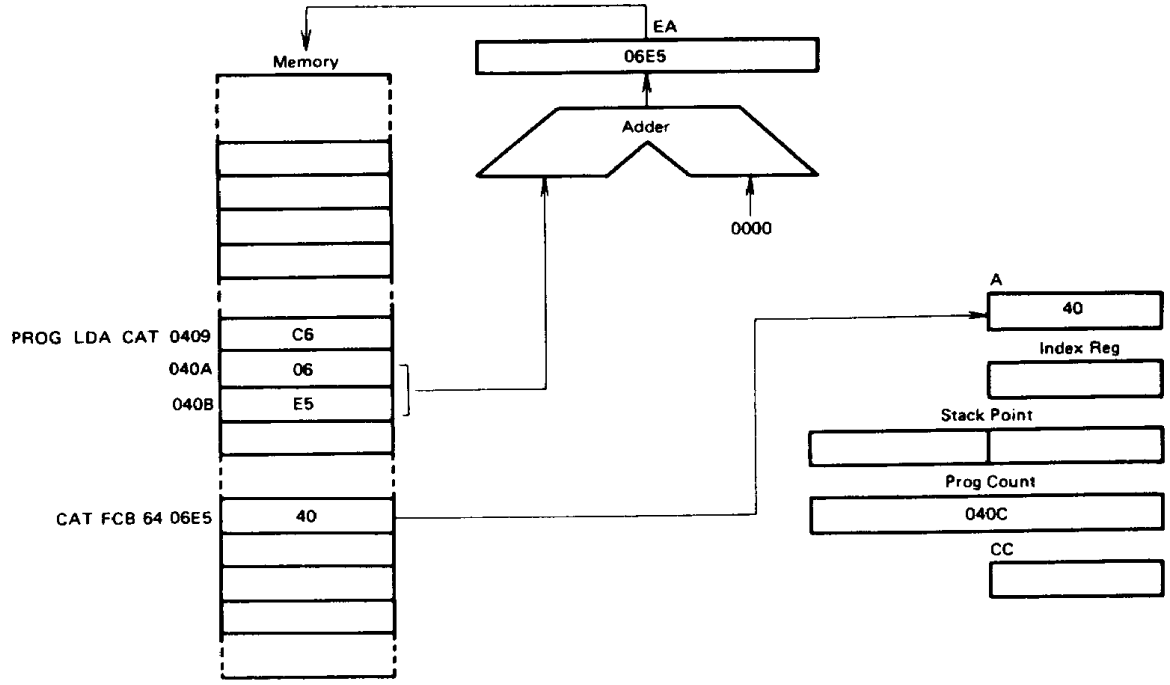


Figure 23 Example of Extended Addressing

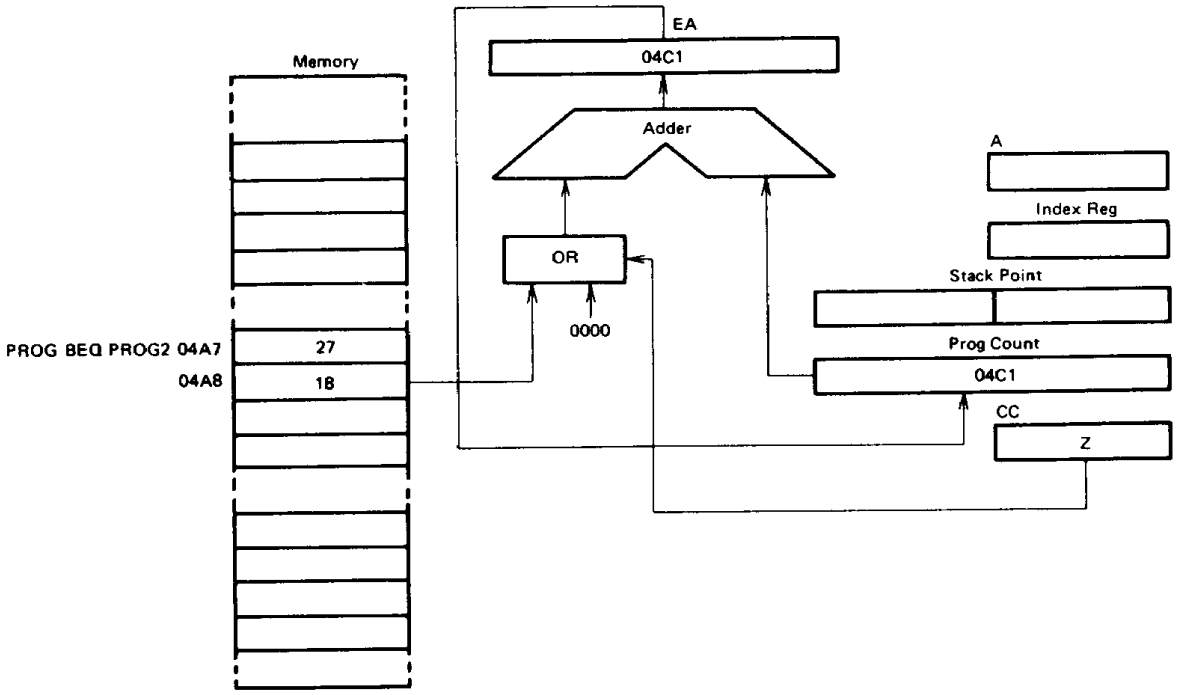


Figure 24 Example of Relative Addressing

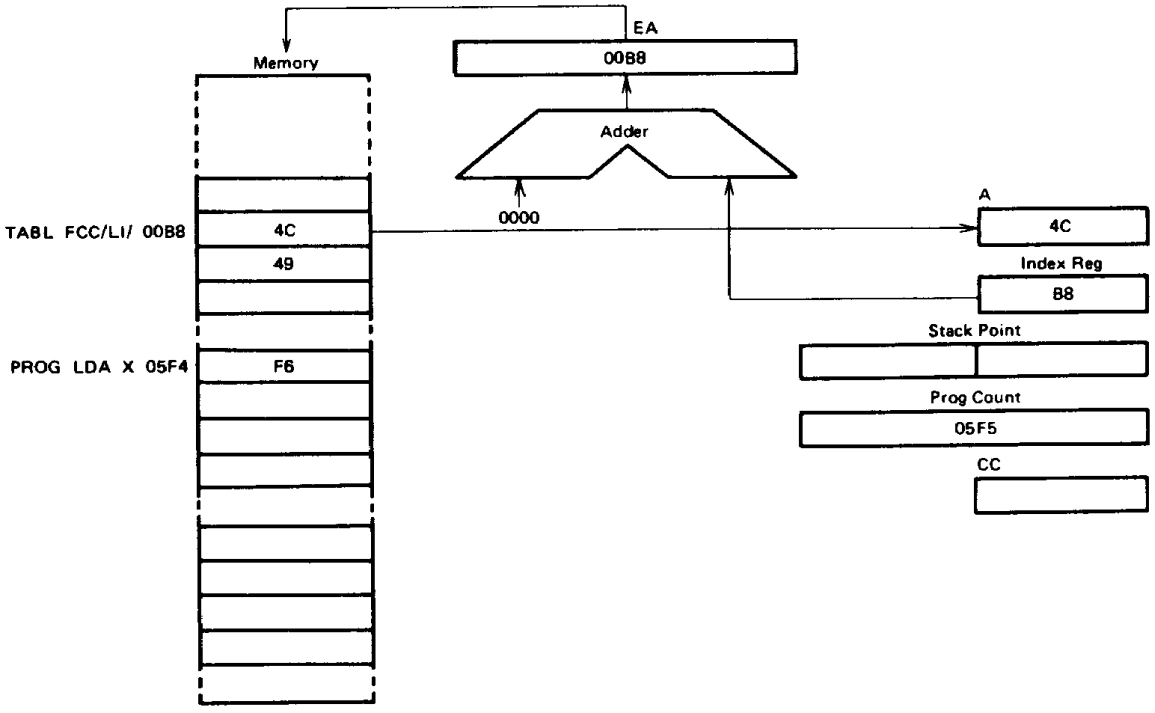


Figure 25 Example of Indexed (without Offset) Addressing

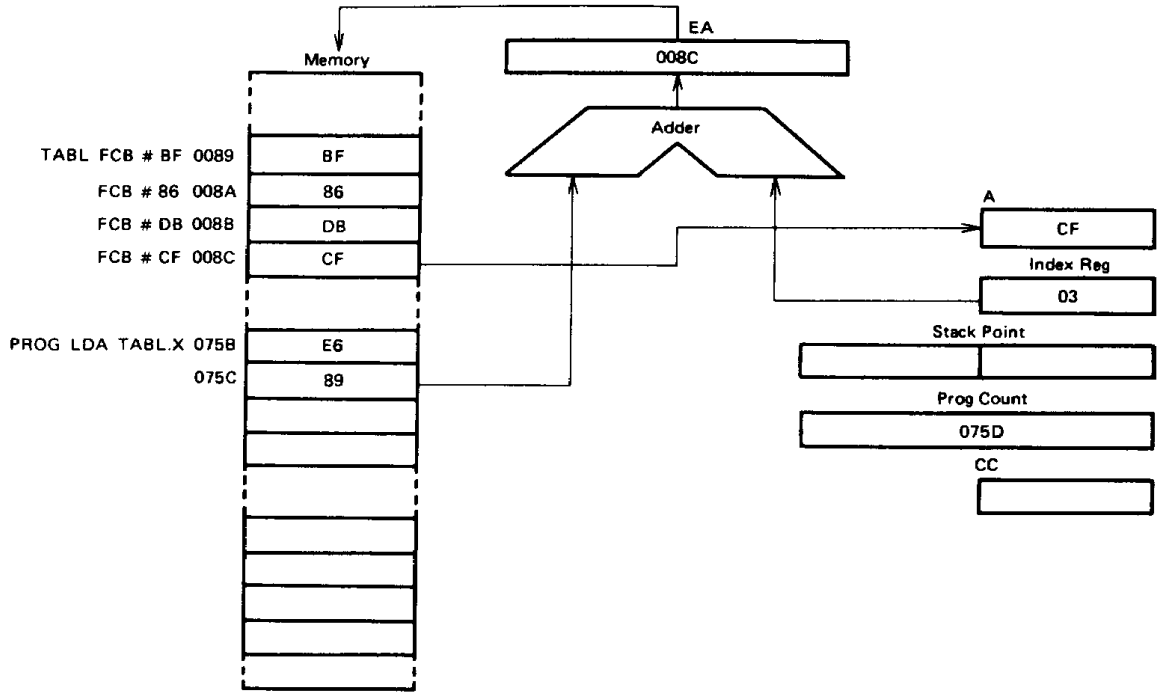


Figure 26 Example of Indexed (8 Bits Offset) Addressing

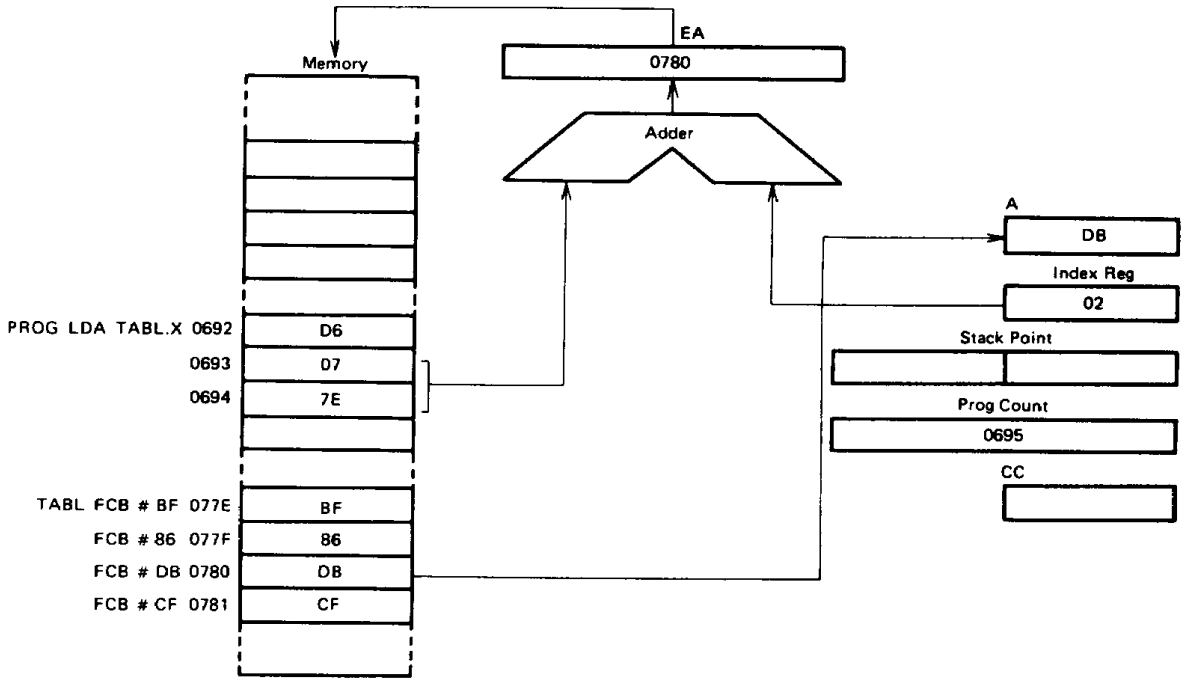


Figure 27 Example of Indexed (16 Bits Offset) Addressing

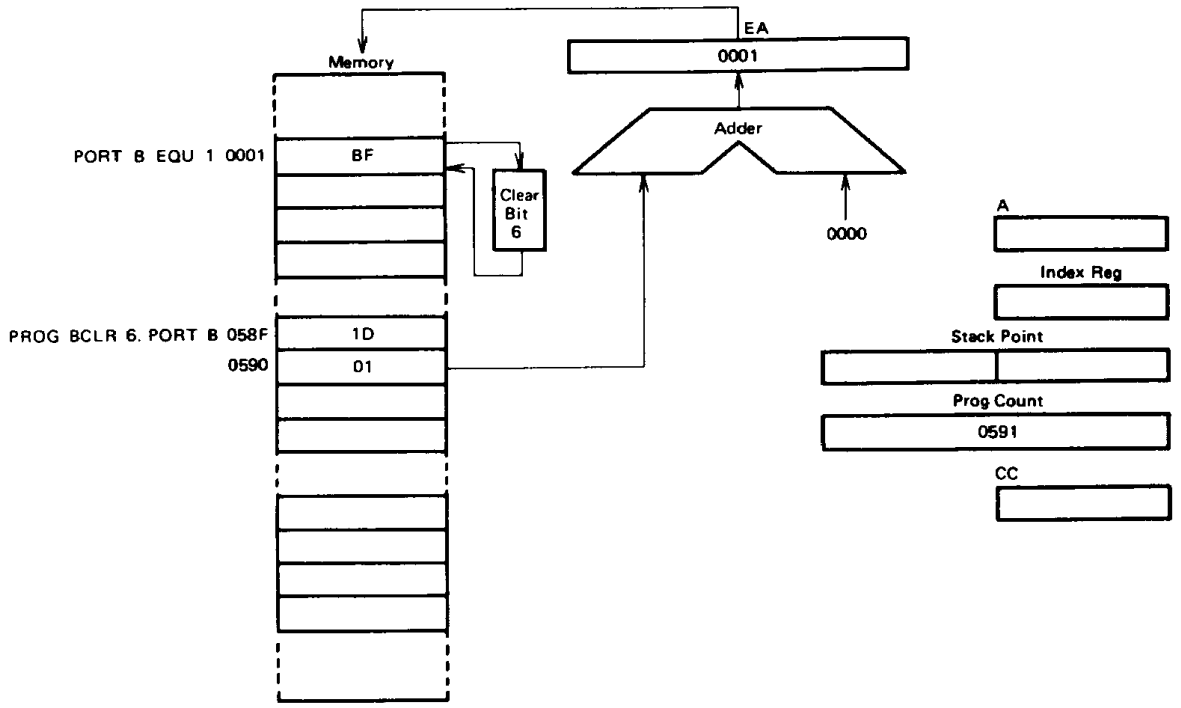


Figure 28 Example of Bit Set/Clear Addressing

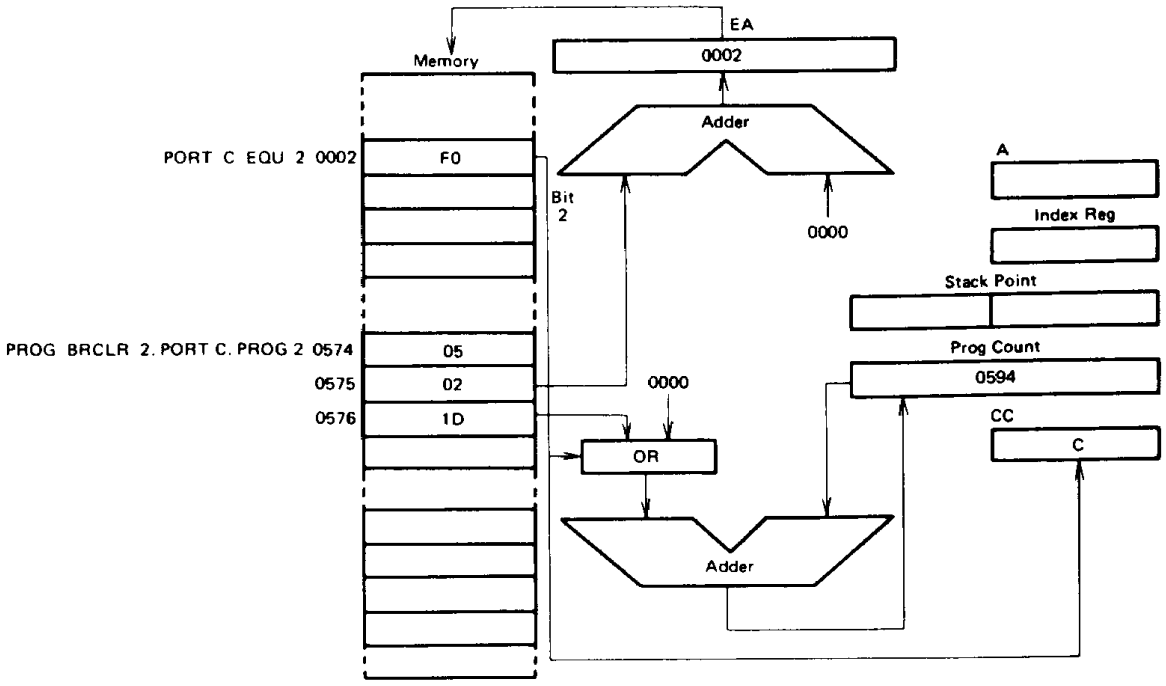


Figure 29 Example of Bit Test and Branch Addressing

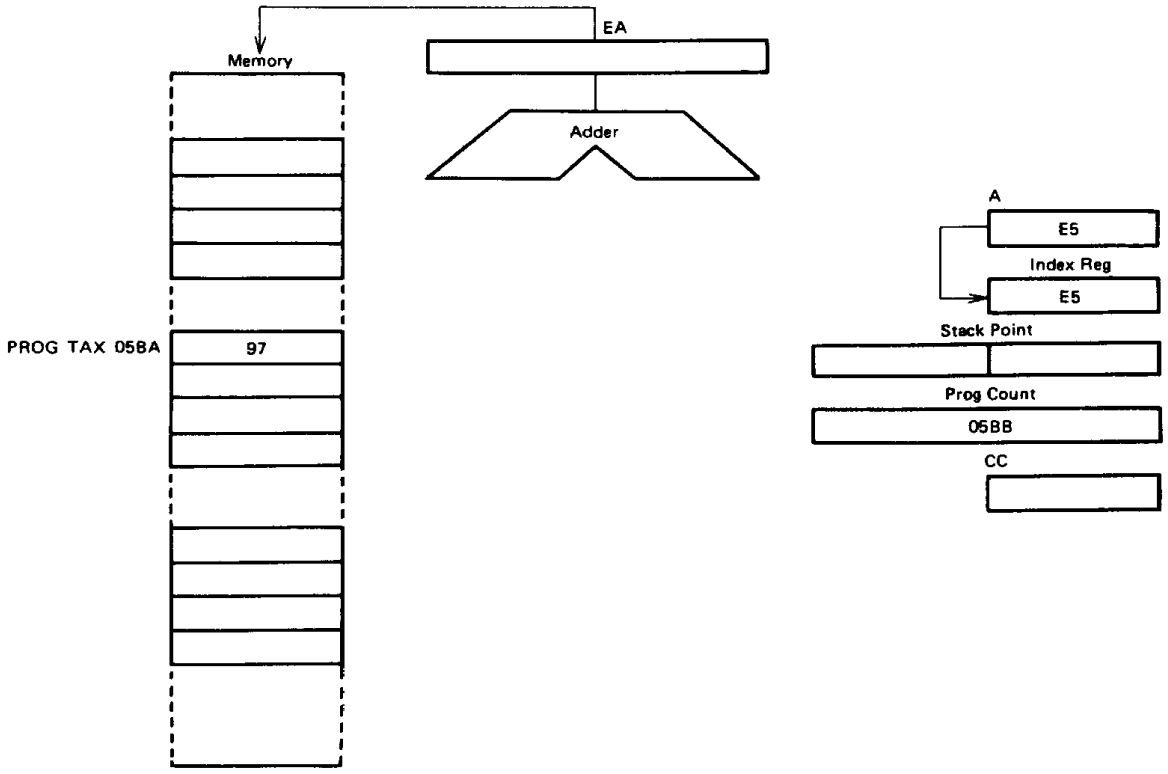


Figure 30 Example of Implied Addressing

■ INSTRUCTION SET

There are 59 instructions available to the MCU. They can be divided into five groups: Register/Memory, Read/Modify/Write, Branch, Bit Processing, and Control. All of these instructions are explained below according to the groups, and are summarized in individual tables.

• Register/Memory

Most of these instructions use two operands. One operand is either the accumulator or index register, while the other is acquired from memory using one of the addressing modes. No operand of register is available in the unconditional Jump (JMP) and Subroutine Jump (JSR) instructions. See Table 4.

• Read/Modify/Write

These instructions read a memory address or register, modify or test its contents, and writes a new value into the memory or register. Negative or Zero instructions (TST) do not provide writing, and are exceptions for the Read/Modify/Write. See Table 5.

• Branch

A Branch instruction will branch from the program sequence in progress if the specific branch condition is satisfied. See Table 6.

• Bit Processing

This instruction can be used for any bit of the first 256 bytes of memory. One group is used for setting or clearing, while the other is used for bit testing and branching. See Table 7.

• Control

The Control instruction controls the operation of the MCU for which a program is being executed. See Table 8.

• A List of Instructions Arranged in Alphabetical Order

All instructions are listed in Table 9 in the alphabetical order.

• OP Code Map

Table 10 shows an OP code map of the instructions used with the MCU.

Table 4 Register/Memory Instructions

		Addressing Mode																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Operation	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	2	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	2	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	3	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	3	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	2	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	2	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	2	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	2	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	2	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	2	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	2	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	2	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	2	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	2	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	1	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	4	CD	3	5	FD	1	3	ED	2	4	DD	3	5

Symbols: Op = Operation # = Instruction

Table 5 Read/Modify/Write Instructions

		Addressing Mode														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Operation	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	1	5C	1	1	3C	2	4	7C	1	3	6C	2	5
Decrement	DEC	4A	1	1	5A	1	1	3A	2	4	7A	1	3	6A	2	5
Clear	CLR	4F	1	1	5F	1	1	3F	2	4	7F	1	3	6F	2	5
Complement	COM	43	1	1	53	1	1	33	2	4	73	1	3	63	2	5
Negate (2's Complement)	NEG	40	1	1	50	1	1	30	2	4	70	1	3	60	2	5
Rotate Left Thru Carry	ROL	49	1	1	59	1	1	39	2	4	79	1	3	69	2	5
Rotate Right Thru Carry	ROR	46	1	1	56	1	1	36	2	4	76	1	3	66	2	5
Logical Shift Left	LSL	48	1	1	58	1	1	38	2	4	78	1	3	68	2	5
Logical Shift Right	LSR	44	1	1	54	1	1	34	2	4	74	1	3	64	2	5
Arithmetic Shift Right	ASR	47	1	1	57	1	1	37	2	4	77	1	3	67	2	5
Arithmetic Shift Left	ASL	48	1	1	58	1	1	38	2	4	78	1	3	68	2	5
Test for Negative or Zero	TST	4D	1	1	5D	1	1	3D	2	4	7D	1	3	6D	2	5

Symbols: Op = Operation # = Instruction

Table 6 Branch Instructions

Operation	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	2 or 3 *
Branch IF Higher	BHI	22	2	2 or 3 *
Branch IF Lower or Same	BLS	23	2	2 or 3 *
Branch IF Carry Clear	BCC	24	2	2 or 3 *
(Branch IF Higher or Same)	(BHS)	24	2	2 or 3 *
Branch IF Carry Set	BCS	25	2	2 or 3 *
(Branch IF Lower)	(BLO)	25	2	2 or 3 *
Branch IF Not Equal	BNE	26	2	2 or 3 *
Branch IF Equal	BEQ	27	2	2 or 3 *
Branch IF Half Carry Clear	BHCC	28	2	2 or 3 *
Branch IF Half Carry Set	BHCS	29	2	2 or 3 *
Branch IF Plus	BPL	2A	2	2 or 3 *
Branch IF Minus	BMI	2B	2	2 or 3 *
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	2 or 3 *
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	2 or 3 *
Branch IF Interrupt Line is Low	BIL	2E	2	2 or 3 *
Branch IF Interrupt Line is High	BIH	2F	2	2 or 3 *
Branch to Subroutine	BSR	AD	2	4

Symbol: Op = Operation # = Instruction
* If branched, each instruction will be a 3-cycle instruction.

Table 7 Bit Processing Instructions

Operations	Mnemonic	Addressing Mode					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is Set	BRSET n (n = 0.....7)	–	–	–	2 · n	3	4 or 5 *
Branch IF Bit n is Clear	BRCLR n (n = 0.....7)	–	–	–	01 + 2 · n	3	4 or 5 *
Set Bit n	BSET n (n = 0.....7)	10 + 2 · n	2	4	–	–	–
Clear Bit n	BCLR n (n = 0.....7)	11 + 2 · n	2	4	–	–	–

Symbol: Op = Operation # = Instruction
* If Branched, each instruction will be a 5-cycle instruction.

Table 8 Control Instructions

Operation	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	1
Transfer X to A	TXA	9F	1	1
Set Carry Bit	SEC	99	1	1
Clear Carry Bit	CLC	98	1	1
Set Interrupt Mask Bit	SEI	9B	1	1
Clear Interrupt Mask Bit	CLI	9A	1	1
Software Interrupt	SWI	83	1	9
Return from Subroutine	RTS	81	1	4
Return from Interrupt	RTI	80	1	7
Reset Stack Pointer	RSP	9C	1	1
No-Operation	NOP	9D	1	1

Symbol: Op = Operation # = Instruction

Table 9 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			^	•	^	^	^
ADD		x	x	x		x	x	x			^	•	^	^	^
AND		x	x	x		x	x	x			•	•	^	^	•
ASL	x		x			x	x				•	•	^	^	^
ASR	x		x			x	x				•	•	^	^	^
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	^	^	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•

(Continued)

Symbols for condition code:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 9 Instruction Set (Continued)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Symbols for condition code:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
∧ Test and Set if True, Cleared Otherwise
• Not Affected
? Load CC Register From Stack

Table 10 OP Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	←HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA (+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+1)		JSR		JSR(+1)	D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/4 or 5	2/4	2/2 or 3	2/4	1/1	1/1	2/5	1/3	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/2	

(NOTES) 1. "–" is an undefined operation code.
2. The figure in the lowest row of each column gives the number of bytes and the cycles needed for the instruction.
The number of cycles for the asterisked (*) mnemonics is a follows:

RTI	7
RTS	4
SWI	9
BSR	4

3. The parenthesized figure must be added to the cycle count of the associated instruction.
4. If the instruction is branched, the cycle count is the larger figure.

• PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

The cross talk may disturb normal oscillation if signal lines are set near the oscillation circuit or other signal lines cross ones connected to oscillation circuit. When designing a board, be careful of this. Crystal and capacitor should be put near XTAL, EXTAL pins and osc_1, osc_2 pins.

• PRECAUTION TO USE STAND-BY MODE

(1) When "High" level is applied to SB pin, osc_1, cpu and peripheral units stop. (Only osc_2 goes on oscillating). When power-on or when reset, "Low" level should be applied to SB pin.

(2) CPU restarts from stand-by mode after delay time of restart passed.

1 When osc_2 is used, system clock is provided from osc_1 after selected delay time of restart passed.

2 When osc_2 is not used, system clock is provided from osc_1 after the time that consists of t'oscf and td passed. (toscf is osillation stabilization time, td is the time the divider divide the clock pulse inputted the time delay circuit).

Because tosc is unstable and harmonization may occur, be careful that restart time does not have accuracy.

3 Peripherals such as Time Base, Timer, A/D and LCD Driver stops in stand-by mode.

When CPU goes into stand-by mode during operating of Time Base, Timer, A/D, right data may not be obtained.

S.B. input signal should be gated according to the program which makes LSI idle. (Fig. 32).

• PRECAUTION TO USE SWI

Be careful of the following since this MCU has restriction on SWI instruction.

(1) Phenomina

There is malfunction of interrupt sequence as following when execution of SWI and interruput occur at the same time.

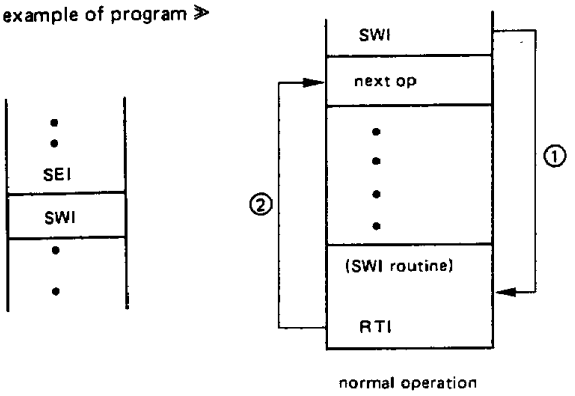
After SWI routine has been executed, return address indicates the address on which that SWI instruction is programmed again. So SWI instruction is executed repeatedly. (Fig. 31).

(2) Countermeasure

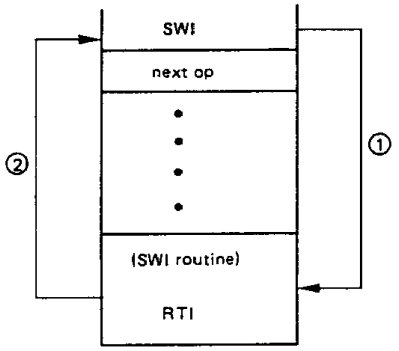
There is no problem in that case SWI instruction is not used or other interrupt request is masked when SWI instruction is executed.

SWI instruction is used like this so as to avoid occurrence of SWI and other interrupt at the same time.

≪example of program≫



normal operation



malfunction (when other interrupt and SWI) interruput occur at the same time.

Figure 31 Example of malfunction

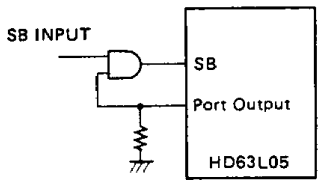


Figure 32 Example of inhibiting SB input signal

- **PRECAUTION TO PROGRAM WRITE ONLY REGISTER AND CONTROL REGISTER**

(1) Read/Modify/Write instructions should not be applied to Write Only Registers to modify its contents. Because Write Only Registers cannot be read out and Read/Modify/Write instruction sequence is as follows.
  (a) Read the data from indicated address.
  (b) Modify its contents.
  (c) Rostore the modified value into the register.

(2) Please do not use BSET and BCLR to DDR of I/O Ports and LCD Data Registers.

(3) Be careful of applying Read/Modity/Write instruction to SYS CTRL Reg., Timer CTRL Reg. and A/D CTRL Reg. Each control register has interrupt request bit (bit 7). If interruput request flag occurs between Read cycle and Write cycle of these instruction, that flag is cleared during write cycle and then CPU does not acknowledge that request.

(4) STA and STX should be applied to Write Only Registers and Control Registers instead of Read/Modity/Write instruction.

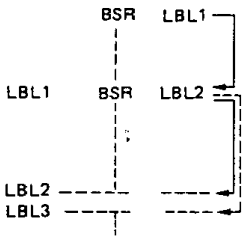
- **PRECAUTION TO USE BSR**

If there is 2nd BSR programmed on the address which is directed by first BSR, 2nd BSR may not be executed correctly. For this reason, BSR should not be programmed on the address which is directed by first BSR.

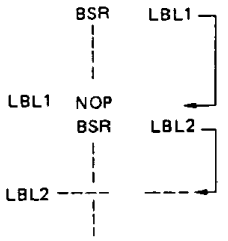
If necessary, please program as following.

(1) On the address which first BSR directed, NOP instruction should be inserted before second BSR.

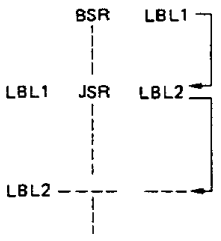
(2) On the address which first BSR direct, JSR instruction should be programmed instead of 2nd BSR.



example of malfunction of 2nd BSR execution



example of counter measure (NOP is inserted)



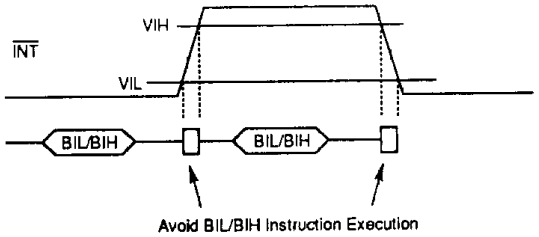
example of counter measure (JSR is used instead of BSR)

- **PRECAUTION WHEN USING BIL/BIH INSTRUCTION**

(1) Execute Instruction after the $\overline{INT}$ Voltage level has stabilized above V_{IH} or below V_{IL}.

(2) $\overline{INT}$ voltage level needs to be stabilized while BIL/BIH Instruction Execution.

There may be a malfunction by glitch on control signal if BIL/BIH Instruction Execution has exercized in unstablized $\overline{INT}$ signal level.



Avoid BIL/BIH Instruction Execution

LCD PIN LOCATION

LCD Register	Bit	Timing			Segment Output Terminal																
		COM_1	COM_2	COM_3	SEG_1	SEG_2	SEG_3	SEG_4	SEG_5	SEG_6	SEG_7	SEG_8	SEG_9	SEG_{10}	SEG_{11}	SEG_{12}	SEG_{13}	SEG_{14}	SEG_{15}	SEG_{16}	SEG_{17}
LCD1	0																				
	1																				
	2																				
	3																				
	4																				
	5																				
	6																				
	7																				
LCD2	0																				
	1																				
	2																				
	3																				
	4																				
	5																				
	6																				
	7																				
LCD3	0																				
	1																				
	2																				
	3																				
	4																				
	5																				
	6																				
LCD4	0																				
	1																				
	2																				
	3																				
	4																				
	5																				
	6																				
LCD5	0																				
	1																				
	2																				
	3																				
	4																				
	5																				
	6																				
LCD6	0																				
	1																				
	2																				
	3																				
	4																				
	5																				
	6																				
LCD7	0																				
	1																				
	2																				
	3																				
	4																				
	5																				
	6																				
LCD8	0																				
	1																				
	2																				
	3																				
ϕ WRITE		O																			
1/4 OSC1		O																			

- Specify the multiplex timing and segment terminal for each bit of LCD1 to LCD8.
- When static or output port is selected, the Multiplex timing is fixed at COM_1.
- If there are unspecified bits, Hitachi specifies them as dummy.
- ϕWRITE is generated when data is written into the LCD1.
- 1/4 OSC1 is a quarter of the OSC1 clock speed. When the MCU is in standby mode, it becomes "Low".
- Please do not specify timing and segment terminal when A/D input is selected.

HD63L05F1
MASK OPTION LIST

DATE OF ORDER	
CUSTOMER	
DEPT.	
ACCEPTED BY	
ROM CODE ID.	
LSI TYPE NO.	HD63L05F1

* Select one type for each item and check ●.

(1) OSC OPTION

Type of OSC1	Use of OSC2	Condition		Delay Time of Restart (sec.) 0	1/16	1/2	1
XTAL Option	Used	STANDBY mode	Used	***	***		
			Not Used				
	Not used	STANDBY mode	Used	***	***		
			Not Used				
CR Option	Used	Oscillation of OSC1 at HALT	Stop		***	***	***
			Continue				
	Not Used	Oscillation of OSC1 at HALT	Stop	***	***	***	***
			Continue				

* Specify a type of OSC option.
* Crystal option of OSC1 is not allowed to stop at HALT.
* If OSC2 is not used, the Delay Time is not accurate.
* Stand-By mode is used in any type.

(2) I/O OPTION

Port	Mask Option A	B	C	D
A_0				
A_1				
A_2				
A_3				
A_4				
A_5				
A_6				
A_7				
B_0				
B_1				
B_2				
B_3				
B_4				
B_5				
B_6				
B_7				
C_0				
C_1				
C_2				
C_3				

Pin	Mask Option E	F
$\overline{INT}$		

Pin	Mask option G	H	K
SEG_{13}/CH_6			***
SEG_{14}/CH_5			***
SEG_{15}/CH_4			***
SEG_{16}/CH_3			***
SEG_{17}/CH_2			***
CH_8/V_2		***	
CH_7/V_1		***	***

A : CMOS output without input pull-up PMOS
B : CMOS output with input pull-up PMOS
C : CMOS output for key scanning
D : NMOS open-drain output
E : Input without pull-up PMOS
F : Input with pull-up PMOS
G : A/D Input
H : Segment output
K : Terminals for LCD display

* Specify an I/O option for each terminal.

(3) LCD DRIVER

	Mask Option L	S	P
Segment			

Mask options indicated as *** are not available.

L : 1/3 bias-1/3 duty LCD
S : Static LCD
P : Output port

* Specify a type of LCD driver.

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