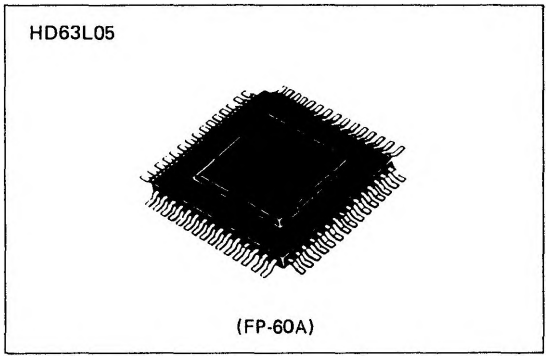
HD63L05
CMOS MCU (Microcomputer Unit)

—PRELIMINARY—

The HD63L05 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 HD63L05 is equipped with a 4k bytes ROM, 96 bytes RAM, I/O, timer, 8 bits A/D, and LCD (6 x 7 segments) drivers, all on one chip.



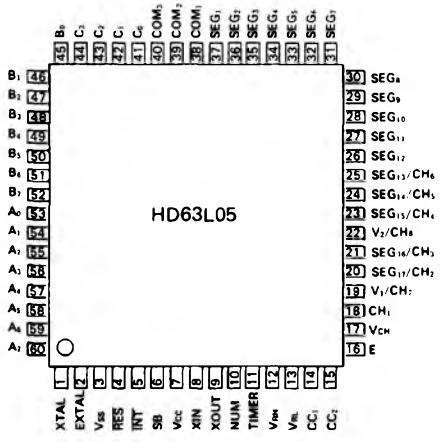
■ HARDWARE FEATURES

- 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 6 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

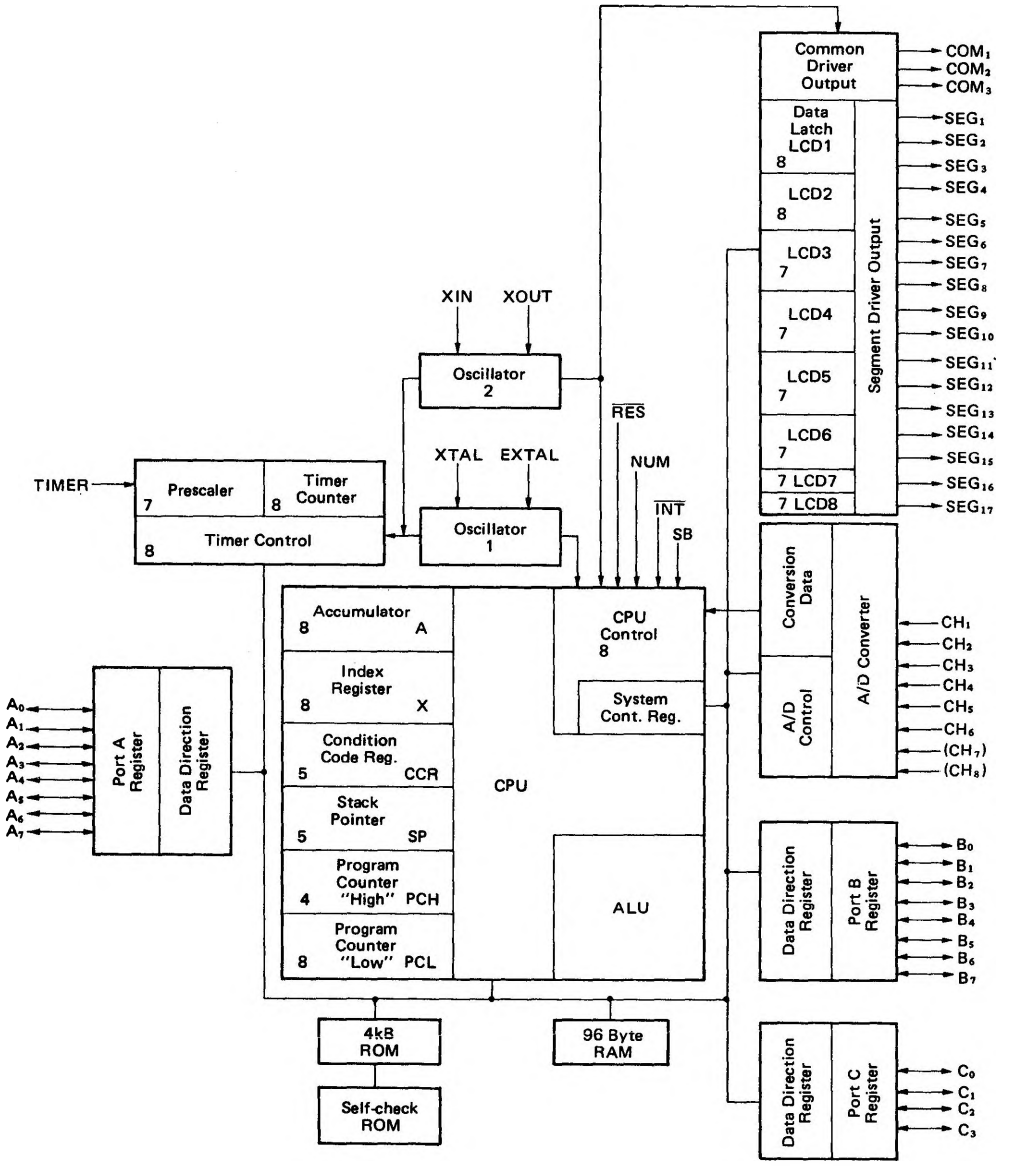
- An Instruction Set Similar to That of The HMCS6800 Family (Compatible with The HD6805S)
- HMCS6800 Family Software Development System Is Applicable

■ PIN ARRANGEMENT



(Top View)

■ 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 Temperature	T_{opr}	-20 ~ +75	°C
Storage Temperature	T_{stg}	-55 ~ +125	°C

(NOTE) If LSI's are used at rating exceeding the absolute maximum rating, they can be permanently destroyed.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 3.0V ±0.8V, V_{SS} = 0V, Ta = -20 ~ +75°C, typ means typical value at 3V unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		–	2.4	–	V
	$\overline{INT}$			–	2.4	–	V
	Others			–	2.2	–	V
	Timer Mode			–	2.2	–	V
	Self-check Mode			–	1.5	–	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		–	0.6	–	V
	$\overline{INT}$			–	0.6	–	V
	Others			–	0.8	–	V
Current Dissipation	During System Operation	I_{CC}		–	100	–	μA
	At Halt			–	40	–	μA
	At Stand-By			–	2	–	μA
	During A/D Operation			–	300	–	μA
Input Leakage Current	TIMER	I_{IL}	V_{in} = 0V ~ V_{CC}	–	0.1	–	μA
	$\overline{INT}$			–	0.1	–	μA

• AC CHARACTERISTICS (V_{CC} = 3.0V ±0.8V, V_{SS} = 0V, Ta = -20 ~ +75°C, typ means typical value at 3V unless otherwise noted.)

Item		Symbol	Test Conditions	min	typ	max	Unit
Operating Clock Frequency		f_{cl}		–	400	–	kHz
Cycle Time		t_{cyc}		–	10	–	μs
$\overline{INT}$ Pulse Width		t_{IWL}		–	t_{cyc} +1	–	μs
$\overline{RES}$ Pulse Width		t_{RWL}		–	t_{cyc} +1	–	μs
TIMER Pulse Width		t_{TWL}		–	t_{cyc} +1	–	μs
Oscillation Start Time (Crystal Option)		t_{oscf}	C_L = 10 pF ±20%	–	100	–	ms
Oscillation Start Time (32kHz)		t_{oscl}	C_G = 10 pF ±20%	–	1.0	–	s
Reset Delay Time		t_{RHL}	Ext. Capacitance = 2.2 μF	–	400	–	ms
Oscillation Frequency (Resistor Option)		f_{EXT}	R = 90 kΩ ±1%	–	400	–	kHz
Input Capacitance	EXTAL	C_{in}		–	10	–	pF
	XOUT			–	10	–	pF
	Others			–	5	–	pF

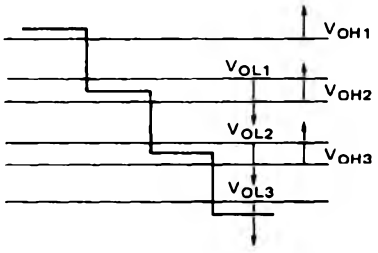
- **PORT CHARACTERISTICS (V_{CC} = 3.0V ±0.8V, V_{SS} = 0V, Ta = -20 ~ +75°C, typ means typical value at 3V unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A, B, C	V_{OH}	I_{OH} = -100 μA	–	2.7	–	V
Output "Low" Voltage	Port A, B, C	V_{OL}	I_{OL} = 100 μA	–	0.3	–	V
Input "High" Voltage	Port A, B, C	V_{IH}		–	2.2	–	V
Input "Low" Voltage	Port A, B, C	V_{IL}		–	0.8	–	V
Input Leakage Current	Port A, B, C	I_{IH}	V_{in} = 3.0V	–	0.1	–	μA
Input Leakage Current	Port A, B, C	I_{IL}	V_{in} = 0V	–	0.1	–	μA
Input Leakage Current (Resistor Option)	Port A, B, C	I_{ILR}	V_{in} = 0V	–	20	–	μA

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

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Segment	V_{OH1}	I_{OH} = -1 μA	2.8	–	–	V
		V_{OH2}	I_{OH} = -1 μA	1.8	–	–	V
		V_{OH3}	I_{OH} = -1 μA	0.8	–	–	V
Output "Low" Voltage	Segment	V_{OL1}	I_{OL} = 1 μA	–	–	2.2	V
		V_{OL2}	I_{OL} = 1 μA	–	–	1.2	V
		V_{OL3}	I_{OL} = 1 μA	–	–	0.2	V
Output "High" Voltage	Common	V_{OH1}	I_{OH} = -5 μA	2.8	–	–	V
		V_{OH2}	I_{OH} = -5 μA	1.8	–	–	V
		V_{OH3}	I_{OH} = -5 μA	0.8	–	–	V
Output "Low" Voltage	Common	V_{OL1}	I_{OL} = 5 μA	–	–	2.2	V
		V_{OL2}	I_{OL} = 5 μA	–	–	1.2	V
		V_{OL3}	I_{OL} = 5 μA	–	–	0.2	V

(NOTE) V_{OH1} and V_{OL3} characteristics apply to the output obtained when segment terminals are used as output ports (I_{OH} = -30μA, I_{OL} = 30μA).



- **A/D CONVERTER CHARACTERISTICS (V_{CC} = 3.0V, V_{SS} = 0V, Ta = -20°C ~ +75°C, C = 300 pF, typ means typical value at 3V unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Accuracy	Resolution			–	8	–	Bit
	Non-Linear Error			–	±1.5	–	LSB
Reference Voltage	"High" Side	V_{RH}		–	2.2	–	V
	"Low" Side	V_{RL}		–	0.2	–	V
	$V_{RH} - V_{RL}$	ΔV_{REF}		–	2.0	–	V
Input Voltage Range		V_{IN}		V_{RL}	–	V_{RH}	V
Conversion Time		t_{CNV}		2	–	4	ms

■ 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".

- **XTAL, EXTAL**
  These are control input ports to the built-in clock circuit. A crystal or 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**
  Connected to these terminals are crystals 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**
  An external input terminal used halt all MCU operations 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**
  Connected to CC_1 and CC_2 are A/D converter offset compensating capacitors. For details, see "A/D Converter".

- **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 ($COM_1 \sim COM_3$, $SEG_1 \sim SEG_{17}$)**
  These are 6 x 7 segments LCD terminals. $COM_1 \sim COM_3$ are for driving common electrodes, while $SEG_1 \sim SEG_{17}$ are for driving segments. They can be used as outputs only under program control. For details, see "LCD Circuit".

- **V_{CH}**
  Output Terminal from Internal Voltage Regulator (A capacitor is connected between V_{CH} and V_{CC}).

- **E**
  System Clock Output (Cycle clock).

- **V_1, V_2**
  These are terminals for LCD driver. Capacitors are connected between V_1, V_2 and V_{CC}.

■ MEMORY

The memory map of the MCU is shown in Figure 1. During processing of an interruption, the contents of the MCU registers are saved into the stack in the order shown in Figure 2. During saving, the stack pointer is decremented and the lower byte (PCL) of the program counter is the first to be stacked. Then the upper 4 bits (PCH) are stacked. For pulling, the saved contents are pulled in order while the stack pointer is being incremented. In the case of a subroutine call, the contents of only the program counters (PCH, PCL) are saved into the stack.

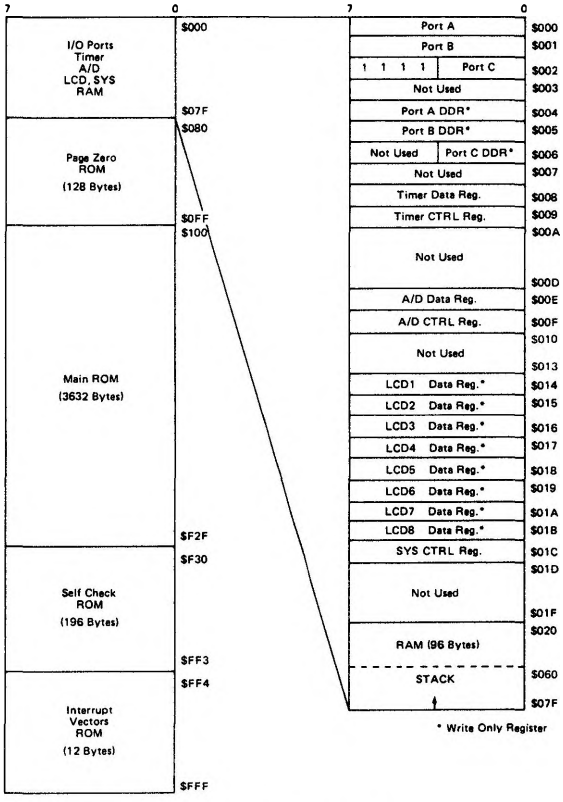


Figure 1 MCU Memory Map

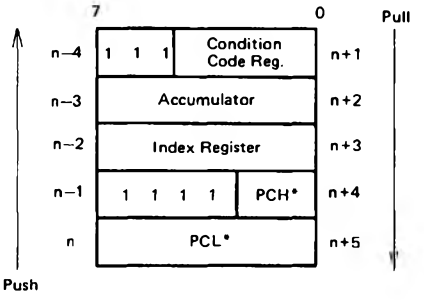


Figure 2 Interruption Stack Sequence

■ REGISTER

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

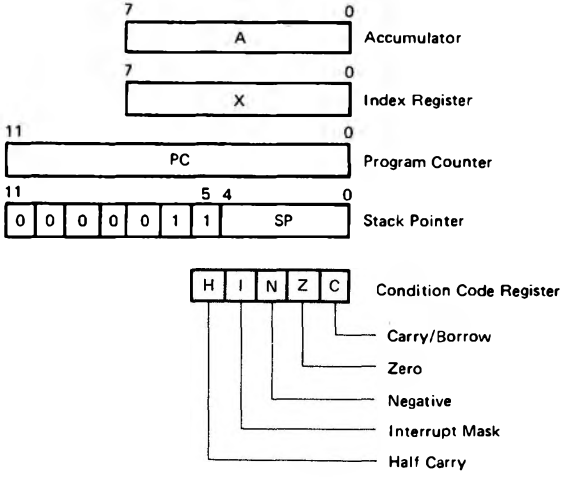


Figure 3 Programming Model

• Accumulator (A)

This accumulator is an ordinary 8 bits register. It is used to accumulate operands and the results of arithmetic operations or data processing.

• Index Register (X)

Being an 8 bits register, this index register is used for index addressing mode. The address contained in the register is composed of 8 bits. An execution address can be obtained by adding the 8 bits to an offset value.

The index register X may also be used for processing a limited range of data at a Read/Modify/Write instruction. When the register is not referenced to by the instruction being executed, it can be used as a temporary storage area.

• Program Counter (PC)

The program counter, a 12 bits register, contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 12 bits register that indicates the address the next save space on the stack. At the beginning, the stack pointer is set at address $07F. It is decremented each time data is saved, and incremented each time data is reset. The upper 7 bits of the stack pointer are fixed to 0000011.

During MCU resetting or a reset stack pointer (RSP) instruction, the stack pointer is set to address $07F. Since a subroutine or interruption can use addresses up to $061 for saving, it is possible to call subroutines up to level 15.

• Condition Code Register (CC)

The condition code register is a 5 bits register, each bit showing the result of an instruction having just been executed. All bits can be tested by conditional Branch instructions. The five bits of the condition code register are used as follows:

Half Carry (H)

Used to indicate that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I)

When this bit is set, all of the timer, external ($\overline{INT}$), A/D and time base interruptions are masked. If an interruption occurs with this bit (I) set, the interruption information is held. It is processed immediately after the interruption mask bit (I) is reset.

Negative (N)

Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is negative (i.e., bit 7 being at logical "1").

Zero (Z)

Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is a zero.

Carry/Borrow (C)

Indicates the carry/borrow that occurred in the most recent arithmetic operation. This bit is affected by the Bit Test and Branch instruction, Shift instruction and Rotate instructions.

■ 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.

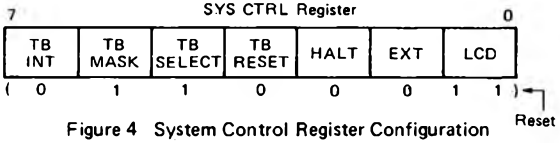


Figure 4 System Control Register Configuration

• Time Base Interruption Request Flag (TB INT)

Stores an interruption request flag from the time base which is selected by the TB select bit. If the TB mask or I (Interrupt Mask Bit in the CCR) is set, the Interruption Request Flag is not acknowledged.

• Time Base Interruption Mask (TB MASK)

If this bit is set, any 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 interruption 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. As this bit has not a register, it indicates logical "0" to the CPU.

The frequency divider provides the system clocks to the A/D converter and LCD driver. So, it is need to pay an attention when "TB RESET" is used.

- **Halt (HALT)**

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

If the bit is reset by an external interruption or time base interruption, the CPU restarts operating. A combined use of the Halt and Time Base Interruption functions permits the CPU to operate intermittently.

- **EXT**

Used to switch the hardware configuration for expanded LCD capabilities. Normally, it is reset.

- **Duty Select Bit (DUTY)**

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

- **TIMER**

Figure 5 shows a block diagram of the MCU timer. This 8 bits counters is loaded under program control. It starts counting-down immediately after clock inputs are applied. When the timer count comes to zero, the timer interruption request bit (bit 7) in the timer control register is set. In response to the interruption request, the MCU saves its contents into the stack.

Then it fetches a timer interruption vector and executes an interruption routine. Any timer interruption can be masked by setting the timer mask bit (bit 6) within the timer control register. The interruption mask bit (I) within the condition code register also inhibits a timer interruption.

Clock inputs to the timer may be the input signal that is applied from an external source to the timer input terminal, or the clock signal within the MCU. If 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. Also, there are two types of internal clock signals within the MCU to allow timer operation when the CPU is halted. These clock signals are under program control.

A 7 bits prescaler is provided to increase the timer's timing interval. The number of bits of the prescaler can be program controlled by the lower 3 bits within the timer control register. If the count comes to below zero, the timer continues counting; the count below zero can be monitored anytime by reading the timer data register, without disturbing the contents of the counter.

At the time of resetting, the prescaler and the counter are all initialized to logical "1". Then the timer interruption request bit is cleared and the timer interruption request mask bit is set.

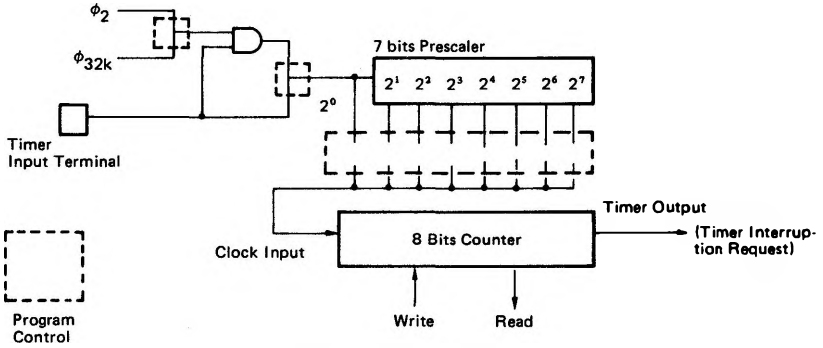


Figure 5 Timer Block Diagram

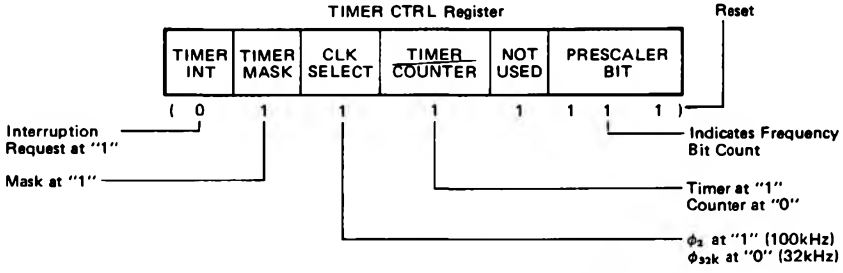


Figure 6 Timer Control Register Configuration

■ RESET

The MCU can be reset either by an external reset input ($\overline{RES}$) or by applying power. In the latter case, the reset input must be "Low" for a sufficient length of time to have the internal oscillator stabilized. A sufficient time of delay is generated by connecting a capacitor to the $\overline{RES}$ input as shown.

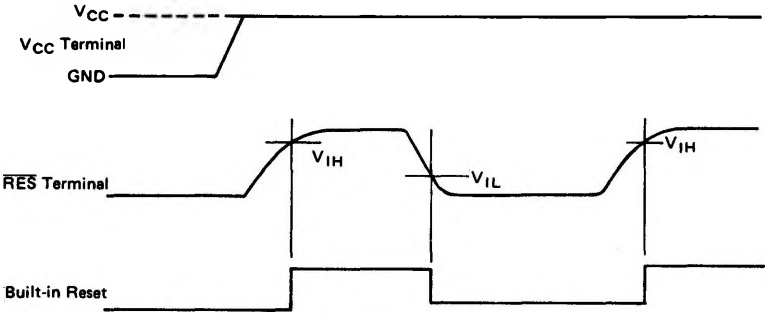


Figure 7 Application of Power and Reset Timing

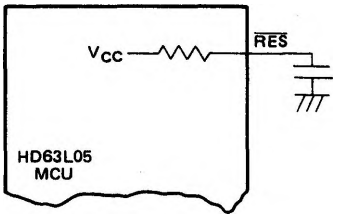


Figure 8 Input Reset Delay Circuit

■ INTERNAL OSCILLATOR OPTION

The MCU incorporates two oscillators: oscillator 1 for system clock supply and oscillator 2 for time base interruption, LCD driving and clock supply.

• Oscillator 1 (XTAL, EXTAL)

The internal oscillator circuit can be driven by an external crystal or resistor depending on the stability. Which to select, crystals or resistors, is determined by the mask option at the time of LSI production. The oscillator 1 can stop when power is applied in either Halt or Standby status. Figure 9 shows the connection.

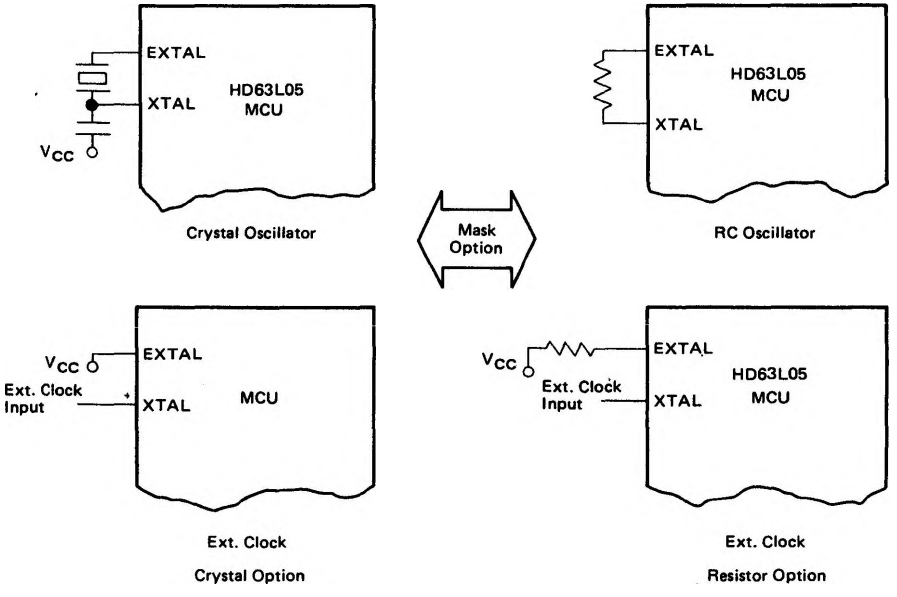


Figure 9 Mask Option for Oscillator 1

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

Clocks for time base interruption and LCD driving can be supplied by connecting a 32.768kHz crystal. In Halt status, oscillator 2 operates and this permits low power dissipation, as well as steady LCD driving and clock operation. In Standby status, this oscillator stops when power is applied. Figure 10 shows the connection; the relation between oscillators 1 and 2 is shown in Figures 11 and 12.

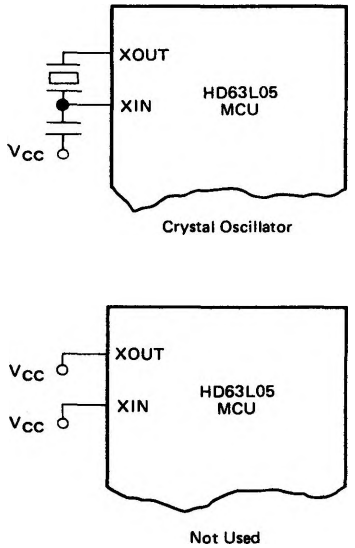


Figure 10 Connection of Oscillator 2

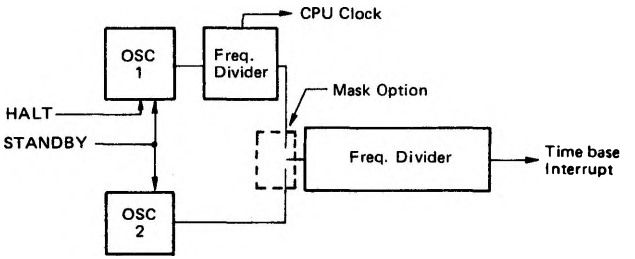


Figure 11 Relation between Oscillators 1 and 2

Mask Option	When OSC1 is X-TAL						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	X	X	O
At Standby	X	X	X	X	X	X	X	X	X	X	X	X

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

Figure 12 Oscillator 2 Mask Option and System Operation

NOTE IN OSCILLATOR SELECTION

When OSC2 is not available, the clocks for the A/D converter and LCD drivers are provided by the OSC1 through the frequency divider. When OSC1 is crystal, OSC1 is not allowend to stop at HALT. Because the response of the oscillator is not so fast. The accuracy of the time base is kept when OSC2 is 32.768 kHz crystal oscillator.

■ INTERRUPTION

There are six different interruptions to the MCU: external interruption ($\overline{INT}$), interruption envoked via an input terminal, internal timer interruption, interruption by termination of A/D conversion, time base interruption (2 types), and interruption by an instruction (SWI).

When an interruption is envoked, the job in progress is suspended and the state of the MCU is saved into the stack. Also, the interruption mask bit (I) of the condition code register is set and the start address of the interruption routine is obtained from the specified vector address. Then, the routine is executed. The RTI instruction is used where control is returned to the program to which the interruption was envoked, after the interruption service routine has been completed.

Table 1 shows the relation between interruptions, priority and vector addresses. Figure 13 shows the system operation flow, in which the portion surrounded with dot-dash lines contains the interruption execution sequence.

Table 1 Interruption Priority

Interruption	Priority	Vector Address
$\overline{RES}$	1	$FFFE, $FFFF
SWI	2	$FFFC, $FFFD
$\overline{INT}$	3	$FFFA, $FFFB
TIMER	4	$FFF8, $FFF9
A/D	5	$FFF6, $FFF7
TIME BASE	6	$FFF4, $FFF5

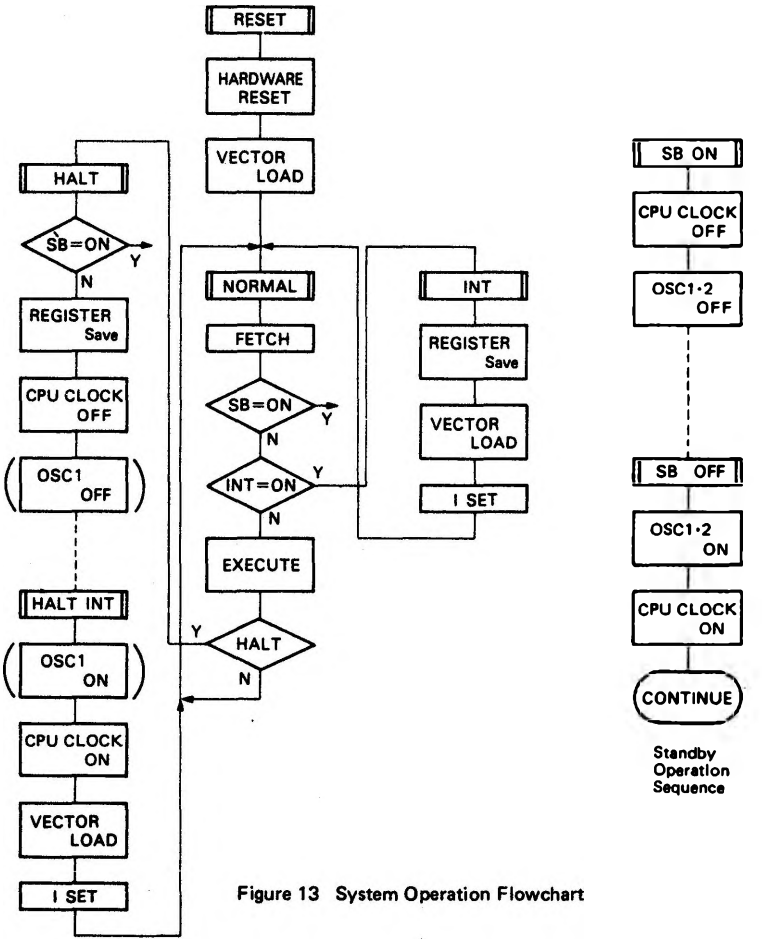


Figure 13 System Operation Flowchart

- **Acknowledging an INT in HALT Status**

In HALT status, 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.

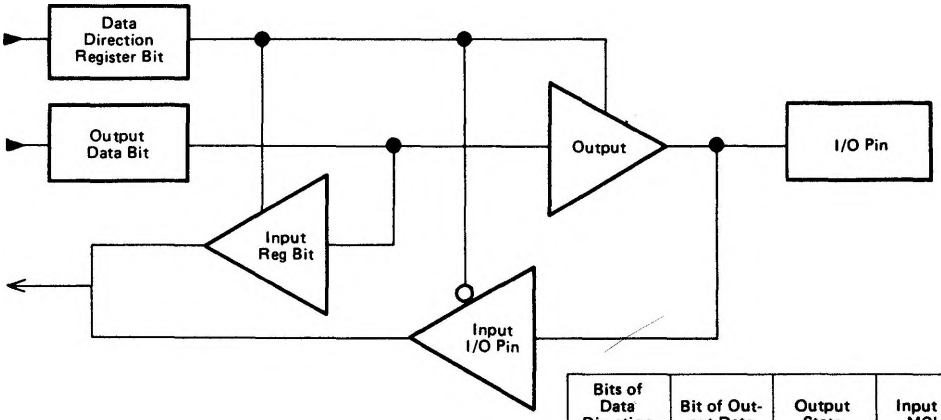
- **Acknowledging an INT in Standby Status**

In Standby status, 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 14 shows the port I/O circuit.



Bits of Data Direction Register	Bit of Output Data	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

Figure 14 Port I/O Circuit

• Configuration of Port

Figure 15 shows the configuration of I/O ports. As the output is on/off controlled by a data direction register, an I/O port may directly be applied as an input terminal. No problem is involved with the input if both "High" and "Low" levels are applied. For only one level, the user must specify the use of a pull-up PMOS for "Open/Low" input application.

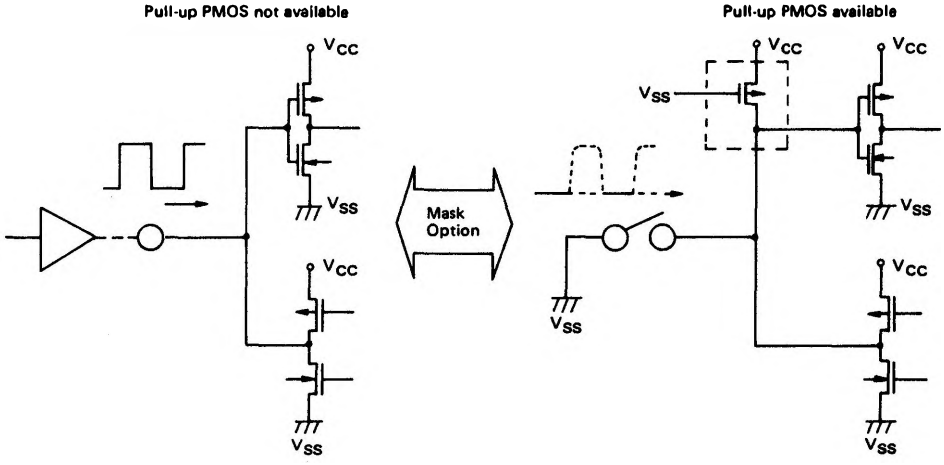
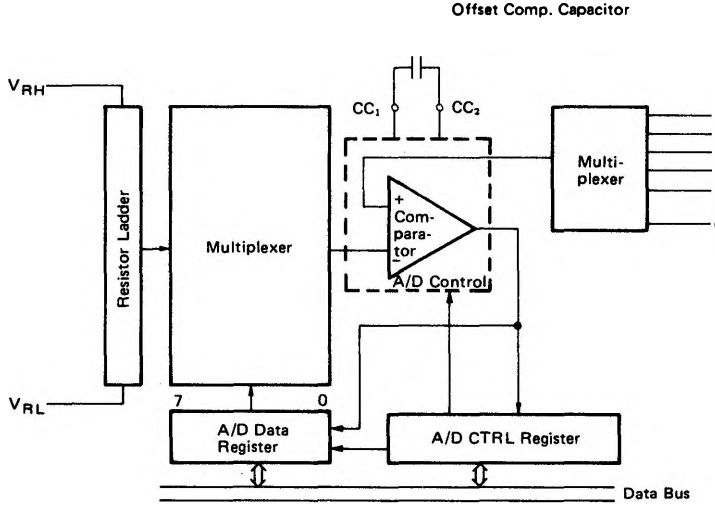


Figure 15 Selection of Input Configuration for I/O Port

■ A/D CONVERTER

The MCU incorporates an 8 bits A/D converter based on the resistor ladder system. Figure 16 shows its block diagram.



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 16 8 Bits A/D Converter Block Diagram

The "High" signal of reference voltage is applied V_{RH}, while the "Low" signal is applied to V_{RL}. The reference voltage is divided by resistors into voltages matching each bit, which then is compared with analog input voltage for A/D conversion.

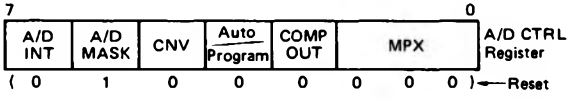


Figure 17 A/D Control Register Configuration

This voltage comparison system achieves high input impedance. Offset are compensated for by external capacitors. Figure 17 shows the configuration of the A/D control register.

• A/D INT

Used to request an interruption after completion of A/D conversion (Request at "1").

• A/D MASK

Used to mask interruptions after completion of A/D conversion (Masking at "1").

• CNV

To start A/D conversion, set this bit to "1". During conversion, "1" is held. The bit is automatically reset to "0" when the A/D conversion ends.

In A/D conversion, supply voltage is applied to the comparator only when CNV = "1". The digital data 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.

• Auto/Program

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

• COMP OUT

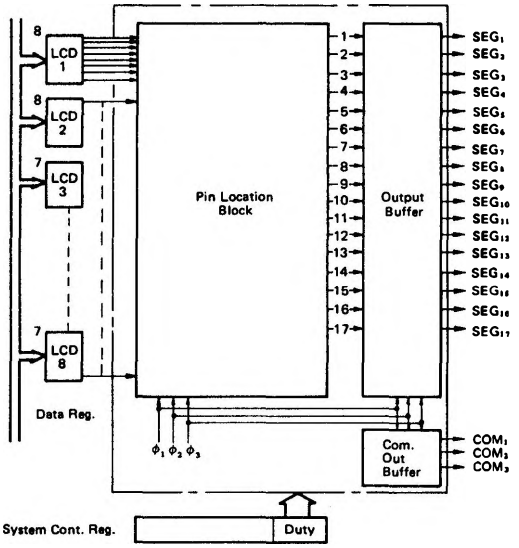
The result of comparator operation under program control can be read from this bit (At "1", input > reference voltage).

• MPX

Used to select 8-channel analog inputs. The multiplexer is an analog switch based on CMOS.

■ LCD CIRCUIT

The system configuration of the LCD circuits is shown in Figure 18. Segment data for display are stored in data registers LCD1 to LCD8. Since the circuits are connected to the output terminals via a pin location block, the user may specify a combination of data to be multiplexed to the segment output terminals.



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

Figure 18 LCD Circuit System Configuration

■ LIQUID CRYSTAL DRIVER WAVEFORMS

The LCD circuit is based on 1/3 bias – 1/3 duty driving. Figure 19 shows the common electrode output signal waveforms (COM1, COM2, COM3), segment signal waveforms (SEG_1 through SEG_{17}), and LCD bias waveforms (COM-SEGMENT).

The segment output terminal may be used as an output-only terminal if the duty of the system control register is so specified. Assignment of segment terminals to the bits of the LCD data register, including the case where they are used as output-only terminals, is to be specified by the user when he orders masks.

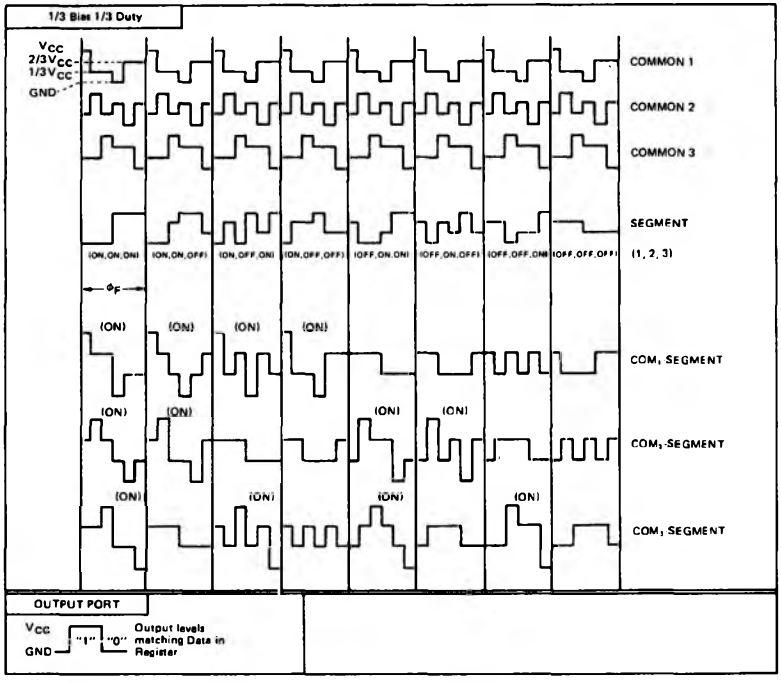


Figure 19 LCD Waveforms

■ BIT PROCESSING

This MCU can use one instruction (BSET, BCLR) to set or clear one bit of the RAM or I/O port (except for the data direction register). All bits of I/O or memory on page 0 are tested by the BRSET and BRCLR instructions. Depending on the result of the test, the program can be branched. Since the bits within the RAM, ROM or I/O can be processed by the MCU, the user can easily use a bit in the RAM as a flag or utilize a single I/O bit as an independent control terminal.

■ 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 20. 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 21. 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 22. 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 23. 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 24. 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 25. 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 26. 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 27. 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 28. 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 29. 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.

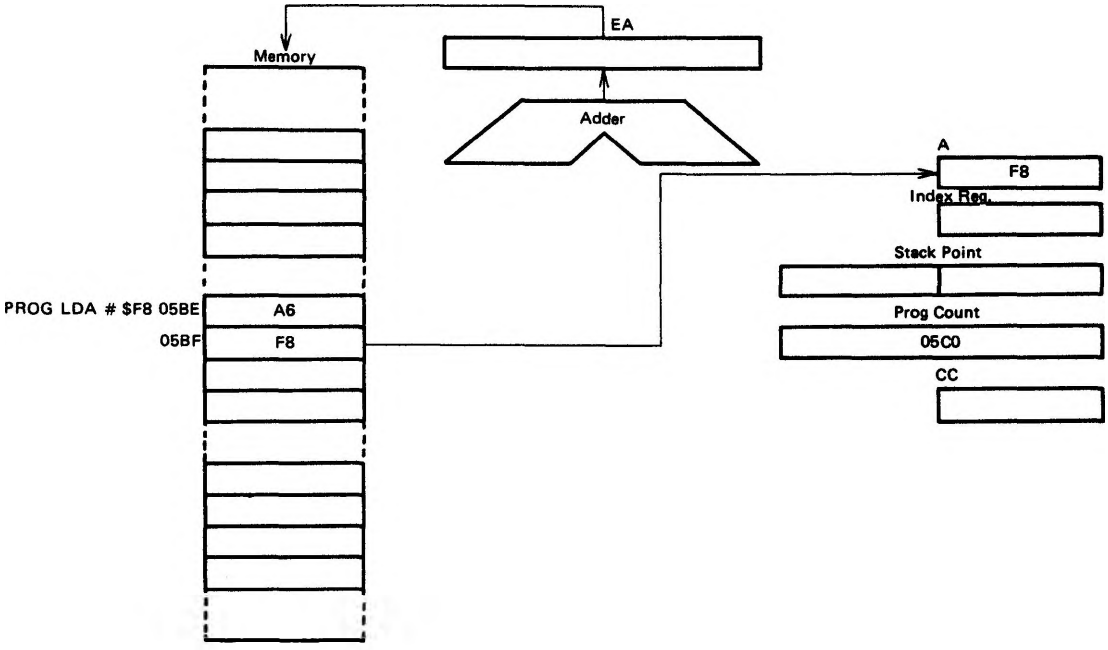


Figure 20 Example of Immediate Addressing

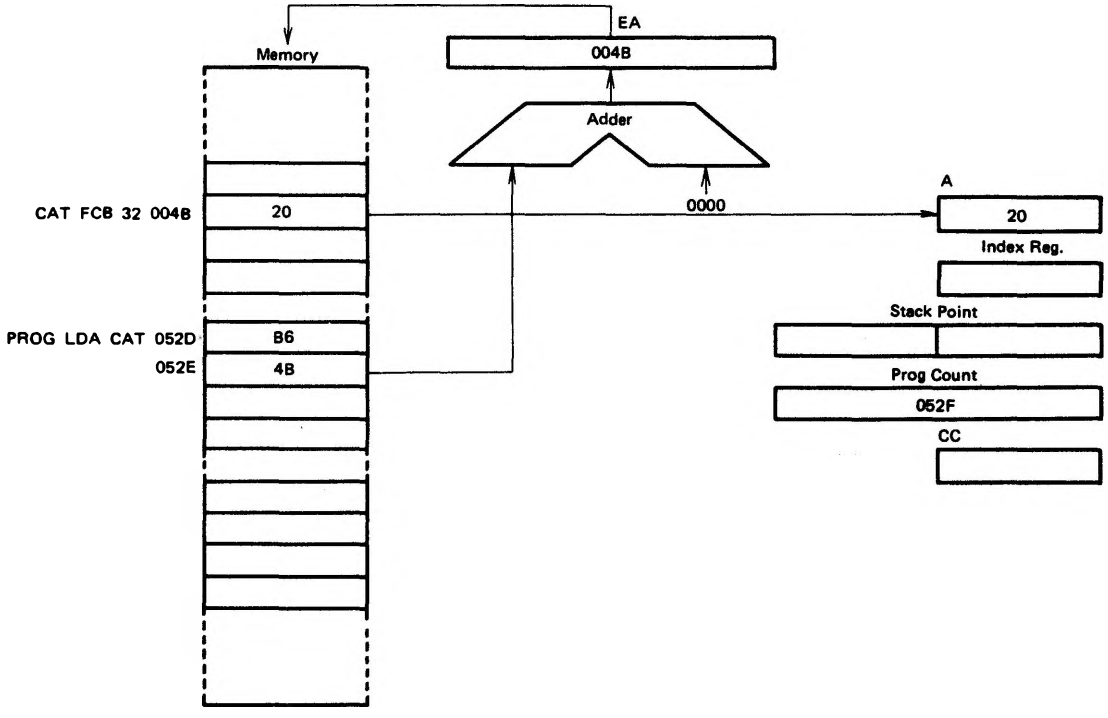


Figure 21 Example of Direct Addressing

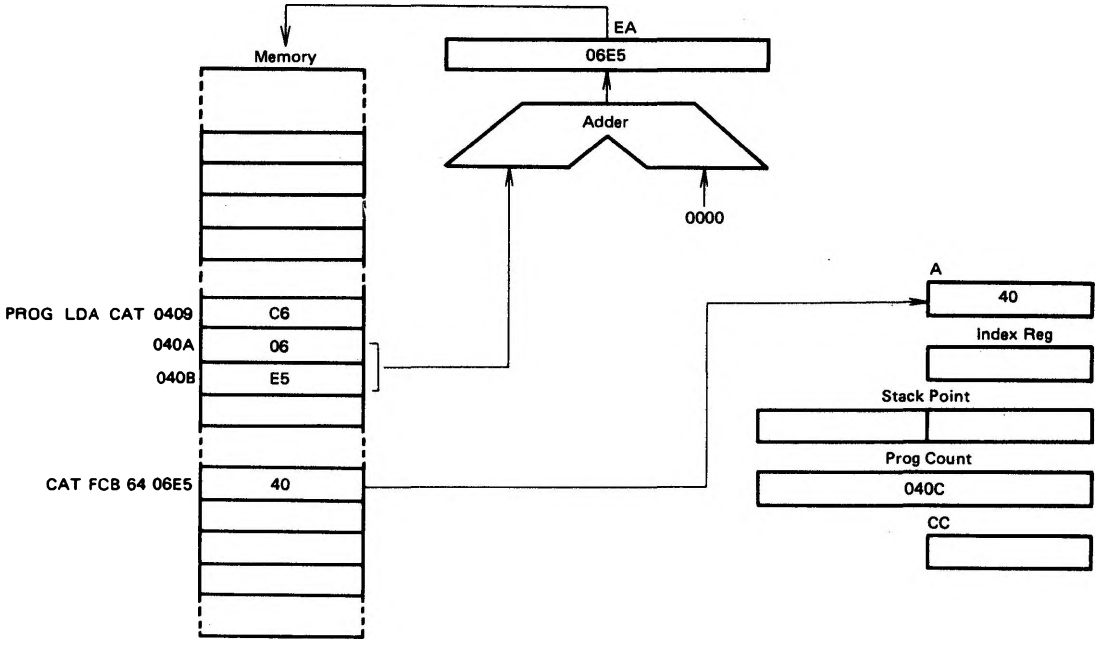


Figure 22 Example of Extended Addressing

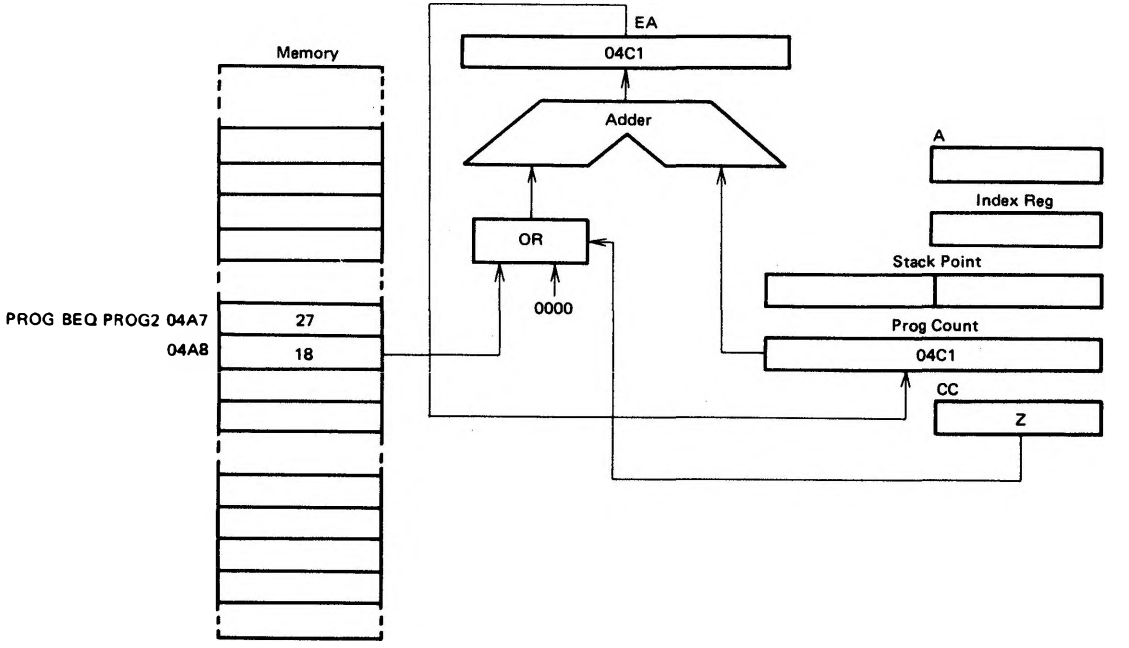


Figure 23 Example of Relative Addressing

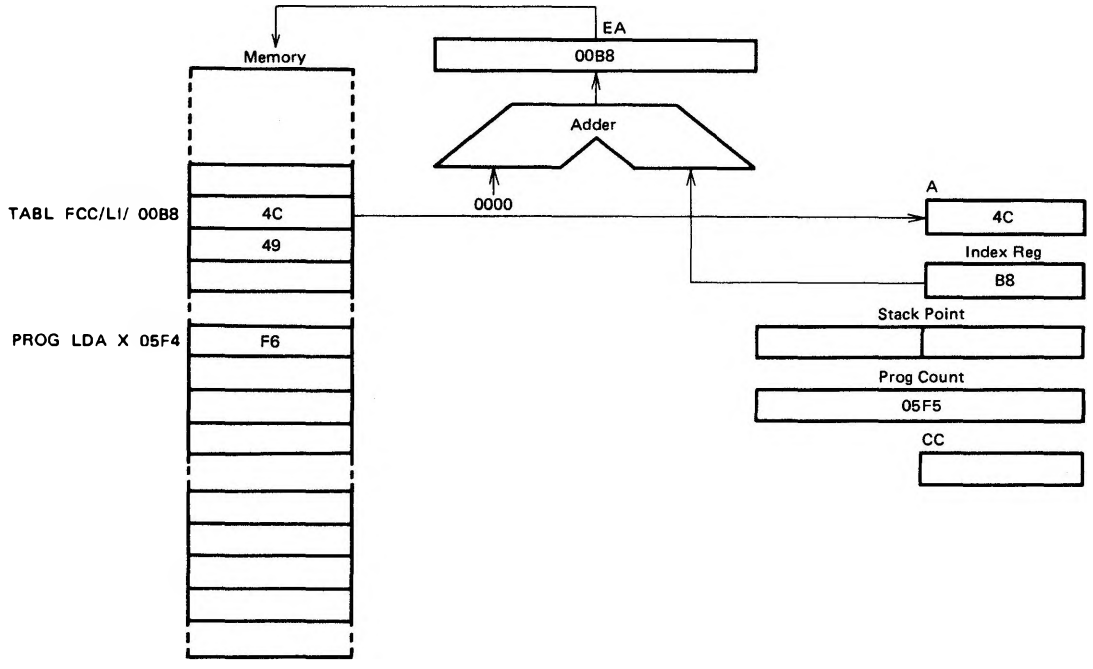


Figure 24 Example of Indexed (without Offset) Addressing

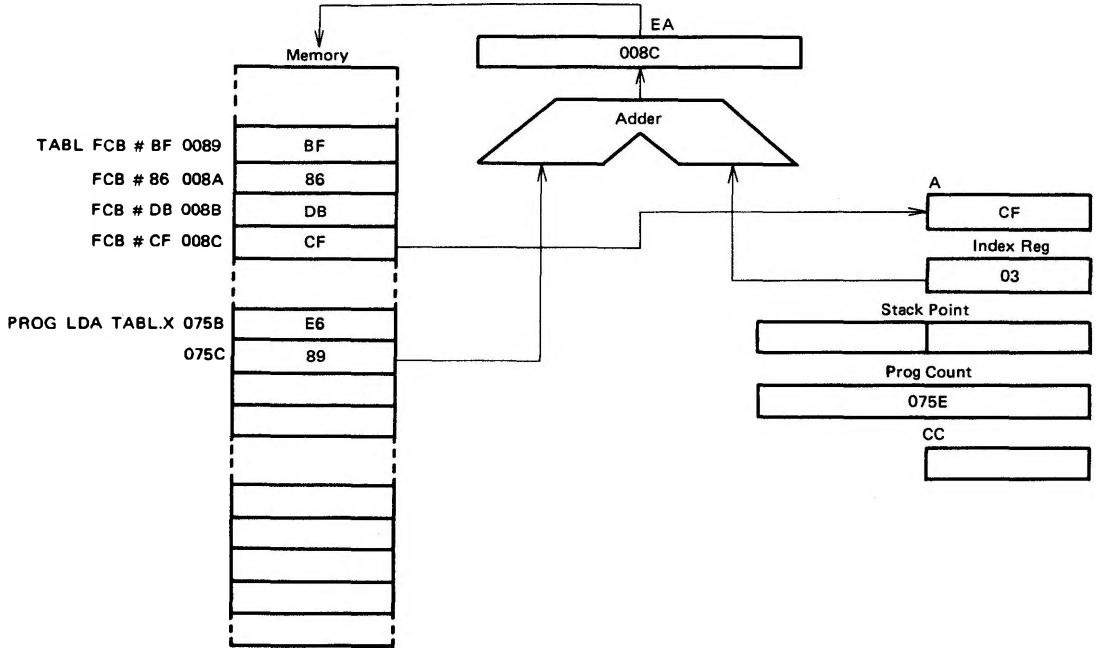


Figure 25 Example of Indexed (8 Bits Offset) Addressing

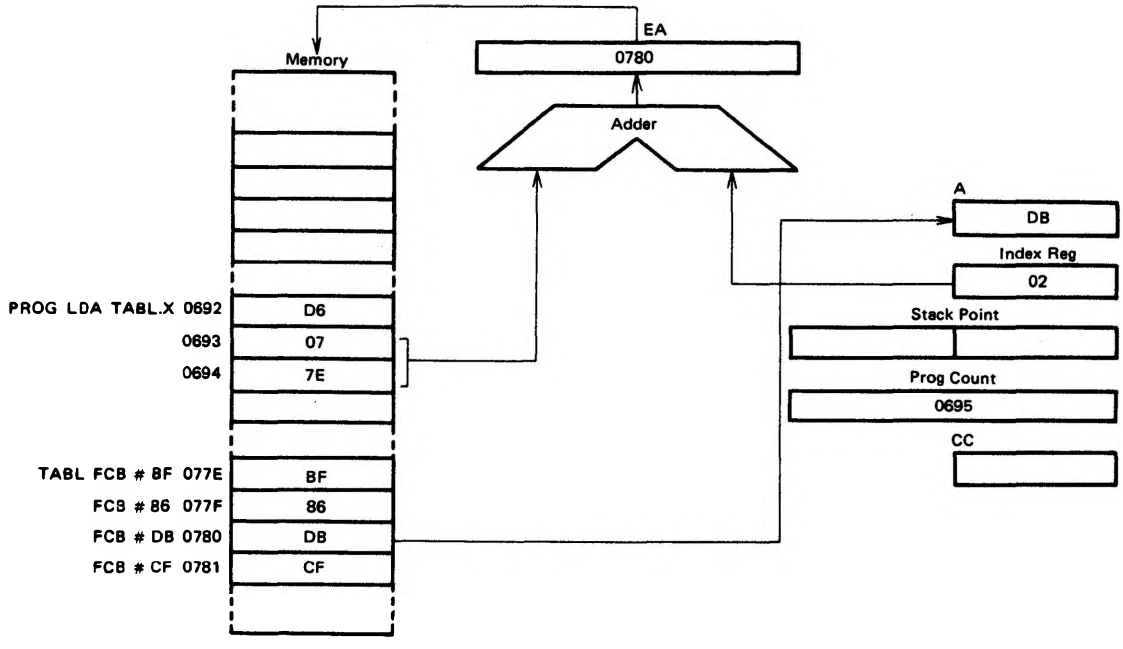


Figure 26 Example of Indexed (16 Bits Offset) Addressing

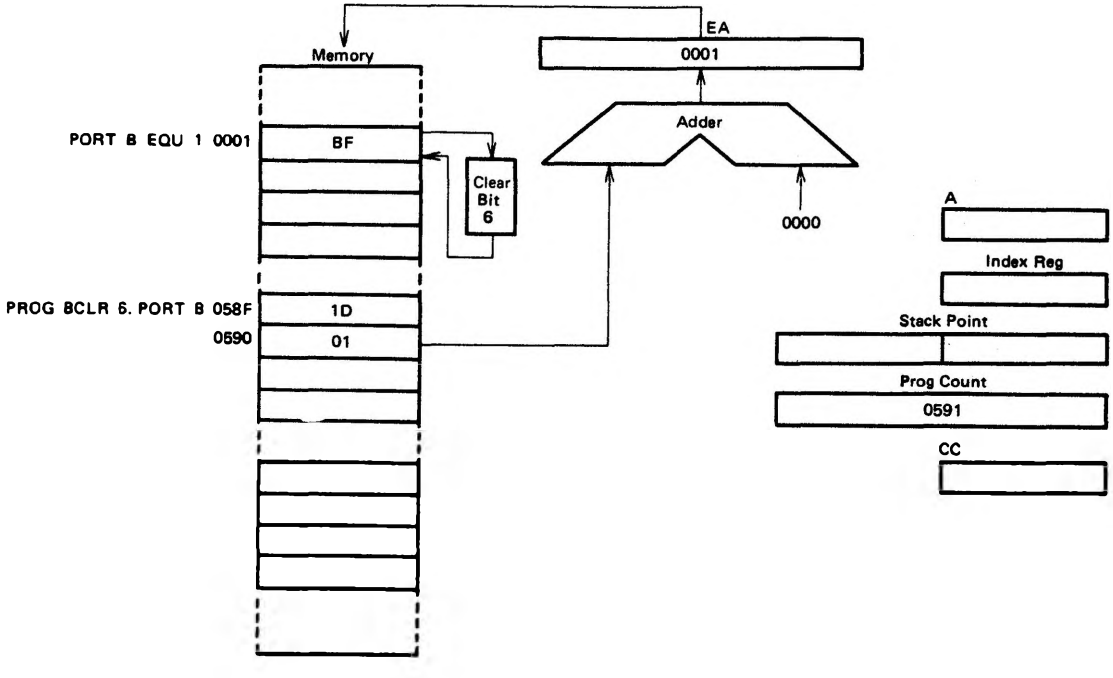


Figure 27 Example of Bit Set/Clear Addressing

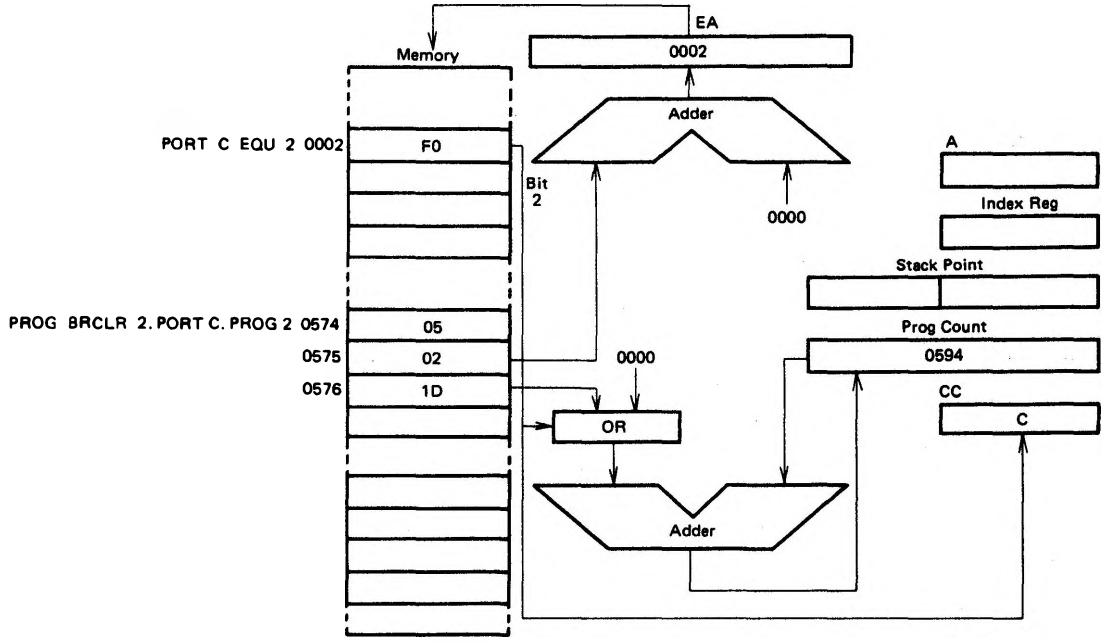


Figure 28 Example of Bit Test and Branch Addressing

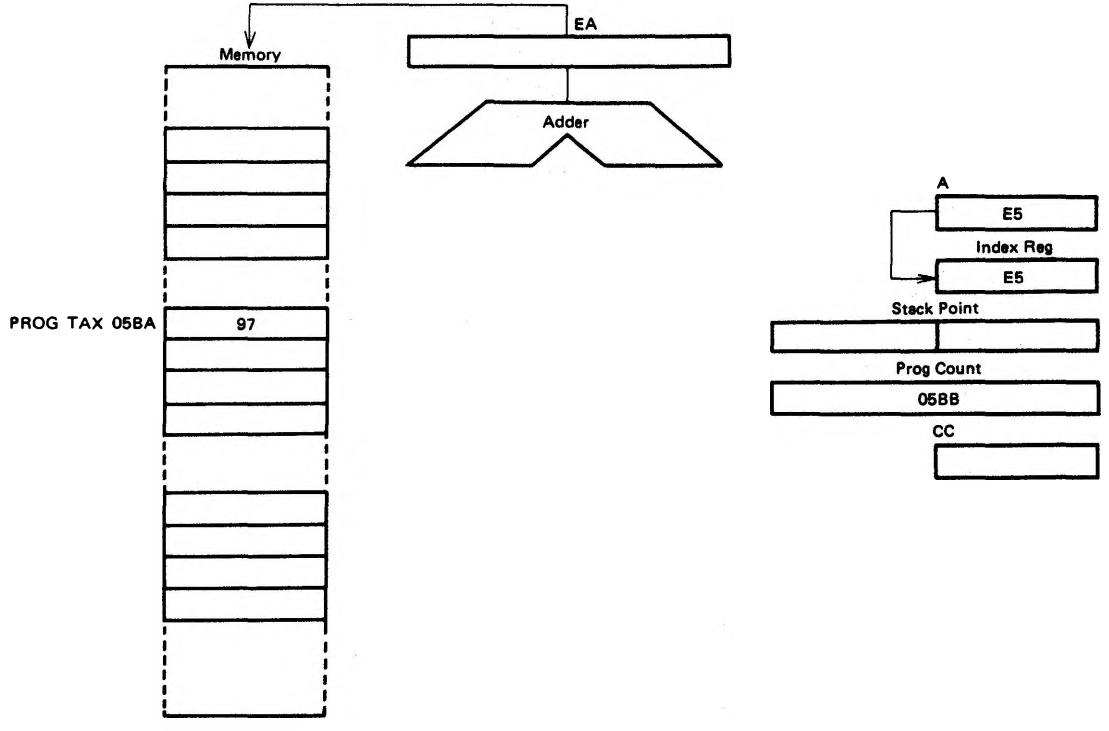


Figure 29 Example of Implied Addressing

■ 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 2.

• 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 3.

• Branch

A Branch instruction will branch from the program sequence in progress if the specific branch condition is satisfied. See Table 4.

• 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 5.

• Control

The Control instruction controls the operation of the MCU for which a program is being executed. See Table 6.

• A List of Instructions Arranged in Alphabetical Order

All instructions are listed in Table 7 in the alphabetical order.

• OP Code Map

Table 8 shows an OP code map of the instructions used with the MCU.

Table 2 Register/Memory Instructions

Operation	Mnemonic	Addressing Mode: Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		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	2	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 3 Read/Modify/Write Instructions

Operation	Mnemonic	Addressing Mode: Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		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 4 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 5 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 6 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 7 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 7 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 8 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.

■ HD63L05 LCD PIN LOCATION COMPOSITION TABLE

Register / Bit		Multiplexed Timing			Segment Output Terminal																
		COM1	COM2	COM3	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17
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																				
	4																				
	5																				
	6																				
ϕWRITE		○																			

(NOTE) Mark a selected Multiplexed Timing and Segment Output Terminal with a circle (0).
In the case of Output or Static LCD driver, Multiplexed Timing is fixed at COM_1.
ϕ_{WRITE} is a write clock for the external option (EXT = "1").
It is generated when LCD1 is rewritten by the CPU.

■ **HD63L05 I/O COMPOSITION TABLE**

Pin Name	I/O	I/O Option				Remarks
		A	B	C	D	
A_0	I/O					
A_1	I/O					
A_2	I/O					
A_3	I/O					
A_4	I/O					
A_5	I/O					
A_6	I/O					
A_7	I/O					
B_0	I/O					
B_1	I/O					
B_2	I/O					
B_3	I/O					
B_4	I/O					
B_5	I/O					
B_6	I/O					
B_7	I/O					
C_0	I/O					
C_1	I/O					
C_2	I/O					
C_3	I/O					
$\overline{INT}$	I					

Pin Name	I/O	I/O Option				Remarks
		E	F	G	H	
SEG_1	O					
SEG_2	O					
SEG_3	O					
SEG_4	O					
SEG_5	O					
SEG_6	O					
SEG_7	O					
SEG_8	O					
SEG_9	O					
SEG_{10}	O					
SEG_{11}	O					
SEG_{12}	O					
SEG_{13}/CH_6	O, I					
SEG_{14}/CH_5	O, I					
SEG_{15}/CH_4	O, I					
SEG_{16}/CH_3	O, I					
SEG_{17}/CH_2	O, I					
V_1/CH_7	O, I					
V_2/CH_8	O, I					

(NOTE) Mark a selected composition with a circle (0).

A. No pull up MOS
B. With pull up MOS
C. CMOS Output
D. Open Drain Output
E. A/D Input
F. Segment Output
G. Output Port
H. LCD Power Supply