PERIPHERAL DEVICES AND INTERFACING

A microprocessor combined with memory and I/O devices, form a microcomputer. Memories and I/O devices are interfaced to microprocessor to form a microcomputer. In the case of large and minicomputers the memories and input/output devices are interfaced to CPU by the manufacturer. In a microprocessor based system the designer has to select suitable memories and input/output devices for his task and interface them to the microprocessor. The selected memories and I/O devices should be compatible with microprocessor.

ADDRESS SPACE PARTITIONING

The Intel 8085 uses 16-bit wide address bus for addressing memories and I/O devices. Using 16 bit wide address bus it can access 2^{16} = 64K bytes of memory and I/O devices· The 64K addresses are to be assigned to memories and I/O devices for their addressing.
There are two schemes for allocation of addresses to memories and I/O devices:

1. Memory mapped I/O scheme
2. I/O mapped I/O scheme
3.

1) Memory mapped I/O scheme

There is only one address space. Address space is defined as set of all possible addresses that a microprocessor can generate. Some addresses are assigned to memories and some addresses to I/O devices. An I/O device is also treated as a memory location and one address is assigned to it. One address is assigned to each memory location. The addresses for I/O devices are different from the addresses which have been assigned to memories and vice-versa. In this scheme all the data transfer instructions of microprocessor can be used for both memory as well as I/O devices· This scheme is suitable for a small system.

2) I/O mapped I/O scheme

In this scheme the addresses assigned to memory locations can also be assigned to I/O devices. Since the same address may be assigned to memory location or an I/O devices, the microprocessor must issue a signal to distinguish whether the address on address bus is for a memory location or an I/O devices· The Intel 8085 issues an IO/M' signal for this purpose. When it is high, address on address bus is for an I/O devices. If it is low, the address on address bus is for a memory location. Two extra instructions IN and OUT are used to address I/O devices. The IN instruction is used to read data of an input device. The OUT instruction is used to send data to an output device. This scheme is suitable for a large system.

MEMORY AND I/O INTERFACING

Several memory chips and I/O devices are connected to a microprocessor. The following diagram (i) shows a schematic diagram to interface memory chips or I/O devices to the microprocessor. An address decoding circuit is employed to select the required I/O device or memory chip. The diagram (ii) shows schematic diagram of the decoding circuit. If IO/M’ is high the decoder 2 is activated and required I/O device is selected. If it is low, decoder 1 is activated and required memory chip is selected. A few MSBs of address lines are applied to decoder to select a memory chip or an I/O devices.

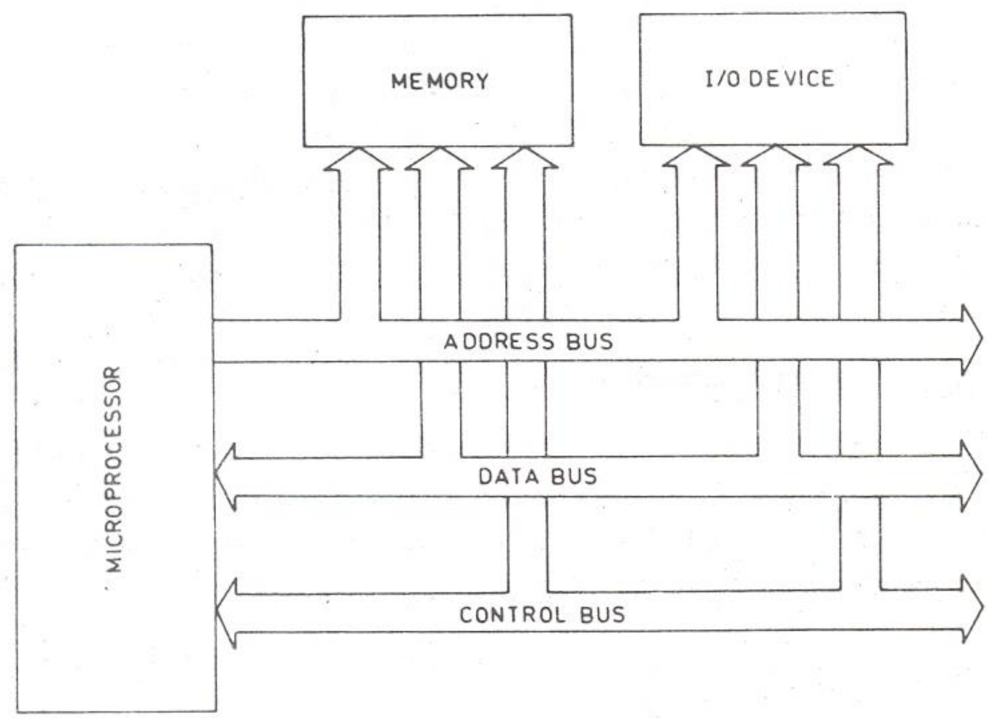
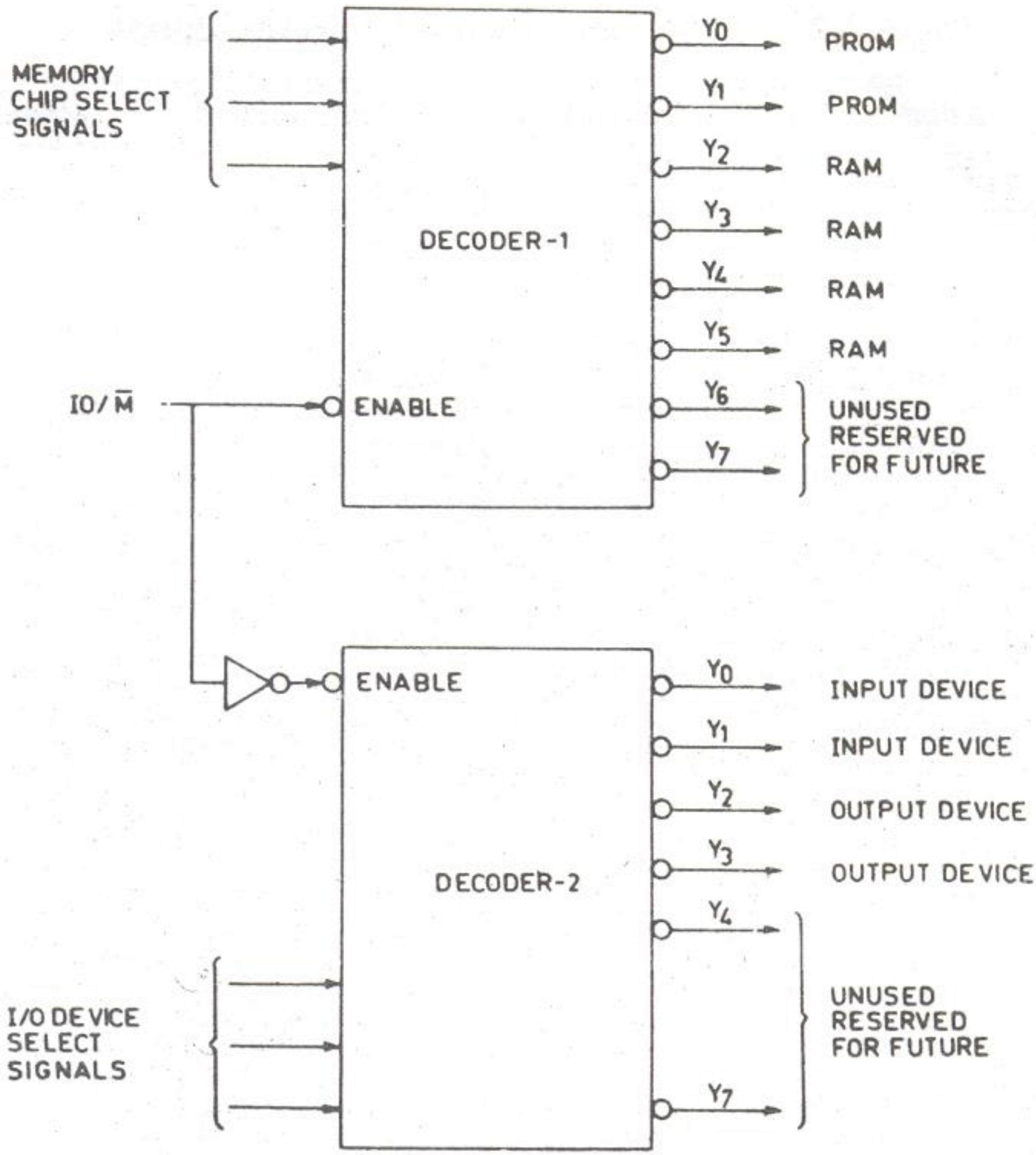


Diagram (i) - Schematic Diagram of Memory and I/O Interfacing



: Interfacing of Memory and I/O Devices.

Diagram (ii) – Schematic Diagram of decoding circuit.

Table 7.1 Truth table for 74LS138

INPUTS						OUTPUTS							
ENABLE			SELECT										
G1	G2A	G2B	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	H	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

X Denotes irrelevant

MEMORY INTERFACING

The address of a memory location or an I/O devices is sent out by microprocessor. The corresponding memory chip or I/O devices is selected by a decoding circuit. The decoding task can be performed by a decoder, a comparator, a bipolar PROM or PLA.
Here we use 74LS138 which is a 1 to 8 lines decoder. The following diagram (iii) shows the interface of memory chips with 74LS138. Here G1,G2A and G2B are enable signals. To enable 74LS138 , G1 should be high, G2A and G2B should be low. A,B and C are select lines . By applying proper logic to select lines any one of the outputs can be selected. Here Y0, Y1,…..Y7 are 8 output lines. An output lines goes low when it is selected. Other output lines remain high. The following table shows the truth table of 74LS138 . When G1 is low or G2A is high or G2B is high, all the output lines become high. Hence 74LS138 acts as decoder only when G1 is high, and G2A and G2B are low.

Decoder Output	Memory Device	Zones of the Address Space	Memory Locations Address
Y_0	EPROM 1	ZONE 0	0000 to 1FFF
Y_1	EPROM 2	ZONE 1	2000 to 3FFF
Y_2	RAM 1	ZONE 2	4000 to 5FFF
Y_3	RAM 2	ZONE 3	6000 to 7FFF
Y_4	RAM 3	ZONE 4	8000 to 9FFF
Y_5	RAM 4	ZONE 5	A000 to BFFF
Y_6	RAM 5	ZONE 6	C000 to DFFF
Y_7	RAM 6	ZONE 7	E000 to FFFF

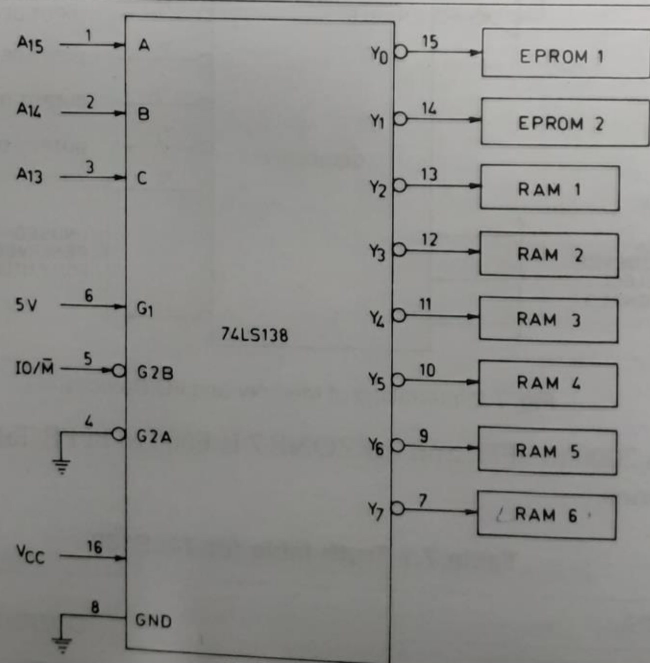


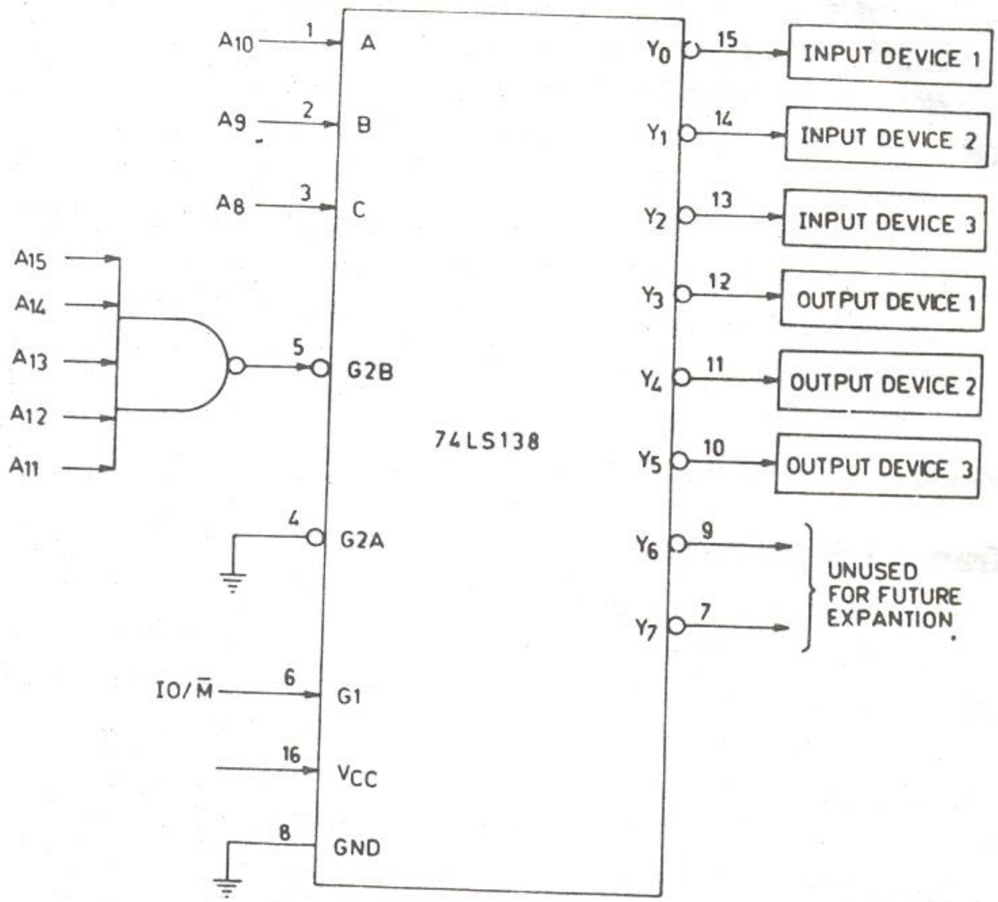
Diagram (iii) – Interfacing of memory chips using 74LS138

The memory locations for EPROM 1 will lie in the range of 0000 to 1FFF. These are the memory locations for ZONE 0 form the memory chip which is connected to the output line Y_0 of the decoder. Similarly other zones are give the range of different addresses.

The entire memory address (64K for 8085) has been dived into 8 zones. Address lines A_{15}, A_{14} and A_{13} have been applied to the select lines of A , B and C of the 74LS138. The logic applied to these lines selects a particular memory device, an EPROM or a RAM. Other address lines A_0, A_1 …….A_{12} (not in the diagram) go directly to the memory chip. They decide the address of the memory location within a selected memory chip. The IO/M’ signal connect to the G2B . When this IO/M’ signal is low for memory read/write operation, G2B goes low, G1 is connected to +5 V. and G2A is grounded.

I/O Interfacing

The following diagram shows the interface of I/O devices through decoder 74LS138. As the address of an I/O device is 8-bits, only A8-A15 lines of address bus are used for I/O addressing. The address lines A8,A9 and A10 have been applied to select lines A, B and C of the 74LS138. The address lines A11-A15 are applied to G2B through a NAND gate. G2B becomes low only when all address lines A11-A15 are high. G2A is grounded. Here IO/M' s connected to G1. When IO/M' goes high for I/O read/Write operation. The following table shows the addresses of I/O devices connected to 74LS138.



Interfacing of I/O Devices Using 74LS138.

Address of I/O Devices connected to 74LS138

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	*Selected Output Lines*	*Corres-ponding Address*	*I/O Device*
1	1	1	1	1	0	0	0	Y_0	F8	Input Device 1
1	1	1	1	1	0	0	1	Y_1	F9	Input Device 2
1	1	1	1	1	0	1	0	Y_2	FA	Input Device 3
1	1	1	1	1	0	1	1	Y_3	FB	Output Device 1
1	1	1	1	1	1	0	0	Y_4	FC	Output Device 2
1	1	1	1	1	1	0	1	Y_5	FD	Output Device 3
1	1	1	1	1	1	1	0	Y_6	FE	Unused
1	1	1	1	1	1	1	1	Y_7	FF	Unused

DATA TRANSFER SCHEMES

In a microprocessor-based system or in a computer data transfer takes place between two devices such as microprocessor and memory, microprocessor and I/O devices, and memory and I/O devices.

Semiconductor memories are compatible with microprocessor because the same technology is employed in manufacturing of both semiconductor memories and microprocessors. Hence, there is less problem associated with interfacing of memory.

A wide variety of I/O devices having wide range of speed and other different characteristics are available. They use different manufacturing technology such as electronic, electrical , mechanical, electro-mechanical, optical etcc.. Due these reasons designers face difficulties in interfacing I/O devices with microprocessor. A microprocessor based system or computer have several I/O devices of different speed. A Slow I/O device cannot transfer data when microprocessor issues instruction for the same because it takes some time to get ready. To solve the problem of speed mismatch between a microprocessor and I/O dev ices a number of data transfer techniques have been developed. The data transfer techniques are classified into two broad categories:

1. Programmed data transfer schemes
2. DMA (Direct Memory Access) data transfer scheme.

1) Programmed data transfer schemes

They are controlled by CPU. Data are transferred from an I/O device to CPU which reside in memory. These programs are executed by CPU when an I/O device is ready to transfer data. The programmed data transfer schemes are employed when small amount of data are to be transferred. They are classified into following categories.

i. **Synchronous data transfer scheme**
ii. **Asynchronous data transfer scheme**
iii. **Interrupt driven data transfer scheme**

DMA data transfer scheme

In DMA data transfer CPU does not participate. Data are directly transferred from an I/O device to memory or vice-versa. The data transfer is controlled by the I/O device or a DMA controller. It is employed when large amount of data are to be transferred. If bulk data are transferred through the CPU , it takes appreciable time and the process becomes slow. An I/O device which wants to send data using DMA technique, send the HOLD signal to the CPU. On receiving a HOLD signal from an I/O device the CPU gives the control of buses as soon as the current bus cycle is completed. The CPU sends a hold acknowledge signal to the I/O device to indicate that it has receive the HOLD request and it has released the buses. The I/O device takes over the control of buses and directly transfers data to the memory or reads data from memory.

This scheme is faster scheme as compared to programmed data transfer scheme. It is used to transfer data from mass storage devices such as hard disks, floppy disks etc. It is used for high speed printers. When data transfer is over, CPU regains the control over the buses.

They are of following types:

i. **Burst mode of DMA data transfer :**

A scheme of DMA data transfer, in which I/O device withdraws the DMA request only after all data bytes have been transferred, is called burst mode of data transfer. Block of data is transferred. It is employed by magnetic disk drives. In case of magnetic disks data transfer cannot be stopped or slowed without loss of data.

ii. Cycle Stealing Technique :

Long lock of data is transferred by a sequence of DMA cycles. In this method after transferring one byte or several bytes the I/O device withdraws DMA request. It reduces interference in CPUs activities. The interference can be eliminated completely by designing an interfacing circuitry which can steal bus cycle for DMA data transfer only when CPU is not using the system bus.

In DMA data transfer schemes I/O devices control data transfer and hence the I/O devices must have registers to store memory addresses and byte count. It must also have electronic circuitry to generate necessary control signals required for DMA operations. Examples of DMA are Intel 8237A , 8257 etc…

.

Synchronous Data Transfer

Synchronous means at the same time. The device which sends data and the device which receives data are synchronized with the same clock. When the CPU and I/O devices match in speed, this technique of the data transfer is employed. The data transfer with I/O device is performed executing IN or OUT instructions for I/O mapped I/O devices or using memory read/write instructions for memory mapped I/O devices. The IN instruction is used to read data from an input device or input port. The OUT instruction is used to send data from the CPU to an output device or output port. As the CPU and the I/O device match in speed, the I/O device is ready to transfer data when IN or OUT instruction is issued by the CPU.

But I/O devices compatible with microprocessors in speed are usually are not available. Hence this technique of data transfer is rarely used for I/O devices.

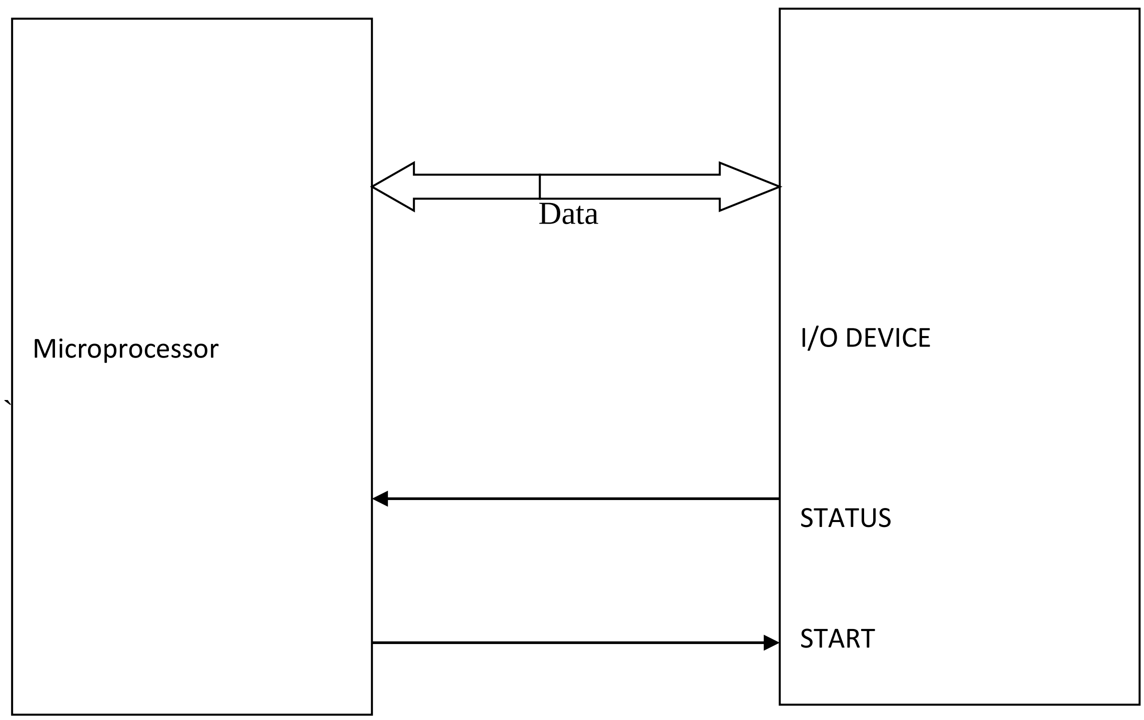
Asynchronous Data Transfer

Asynchronous means at irregular intervals. In this method data transfer is not based on per determined timing pattern. This technique of data transfer is used when the speed of an I/O device does not match the speed of the microprocessor, and the timing characteristic of I/O device is not predictable. In this technique the status of the I/O device is checked by the microprocessor before the data are transferred. The microprocessor initiates the I/O device to get ready and then continuously checks the status of the I/O device till the I/O device becomes ready to transfer data. This mode of data transfer is called handshaking mode of transfer .

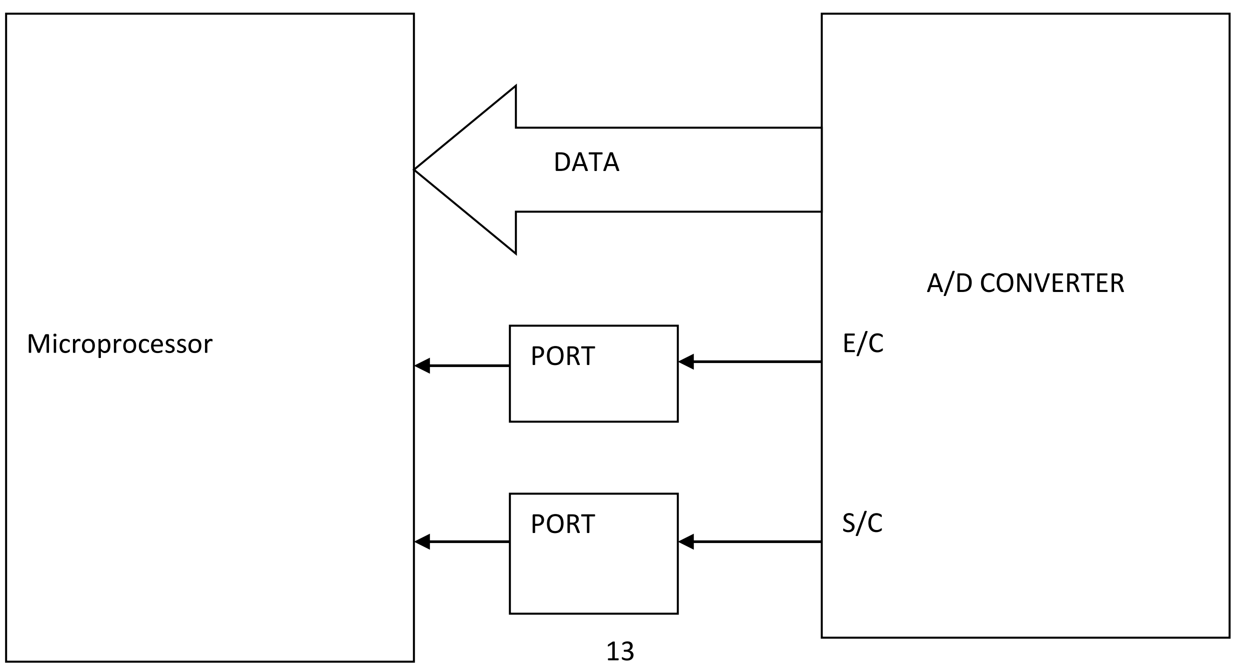
In this handshaking mode of data transfer some signals are exchanged between the I/O device and the microprocessor before the actual data transfer takes place. The microprocessor issues an initiating signal to the I/O device to get ready . When an I/O device becomes ready it sends signals to the processor to indicate that it is ready. Such signals are called handshake signals.

The following diagram shows the schematic diagram for asynchronous data transfer. Asynchronous data transfer is used for slow I/O devices. This technique is inefficient technique because time is wasted.

Also the other diagram is an simple example of asynchronous data transfer. Here an A/D converter has been interfaced to the microprocessor to transfer data in asynchronous mode. The microprocessor sends a start of conversion signal , S/C to the A/D converter. The A/D converter being a slow device as compared to a microprocessor, takes some time to convert analog signal to digital signal. When conversion is over the A/D converter makes the end of conversion signal, E/C high. The microprocessor goes on checking E/D till it becomes high. When E/C becomes high the microprocessor issues instructions for data transfer.



(Asynchronous data transfer)



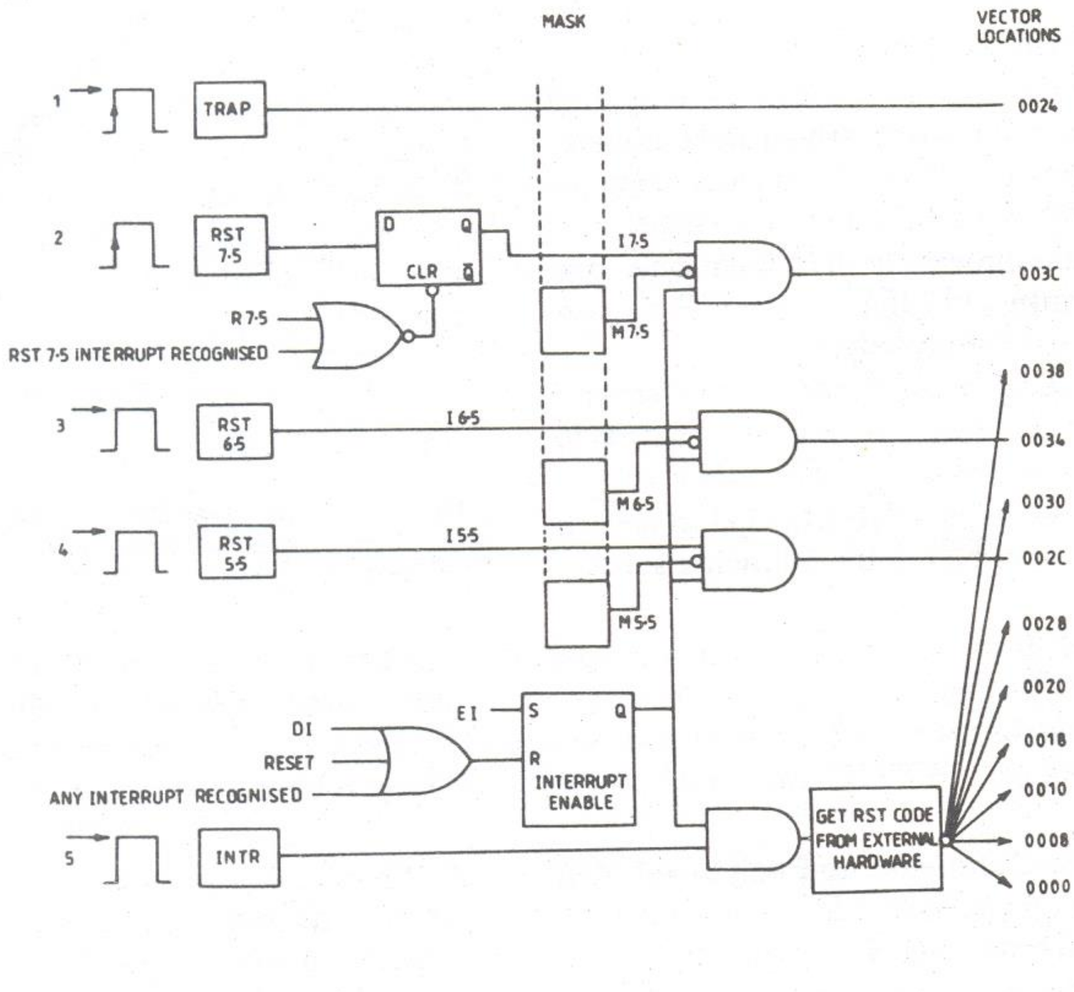
(Asynchronous Data Transfer Scheme for an A/D coverter)

INTERRUPTS OF INTEL 8085

It has five interrupts namely TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR.
TRAP has highest priority followed by RST 7.5, RST 6.5 and RST 5.5. The INTR has the lowest priority. When interrupts are to be used they are enabled by software using the instruction EI (Enable Interrupt) in main program. EI sets the interrupt enable flip-flop to enable the interrupts.
The instruction DI (Disable Interrupt) is used to disable interrupts. DI resets the interrupt enable flip-flop and disables all the interrupts except non-maskable interrupt TRAP. The system RESET also resets the Interrupt Enable flip-flop. When I/O device becomes ready to transfer data, it sends a high signal to microprocessor through a special input line called an *interrupt line.* It interrupts the normal processing sequence of microprocessor. On receiving an interrupt microprocessor completes current instruction at hand, and then attends I/O device. It saves contents of PC on stack first, and then takes up a subroutine called ISS (Interrupt Service Routine). It executes ISS to transfer data from or to the I/O device.

All the interrupts except TRAP are disabled by resetting the Interrupt Enable flip-flop.The resetting of this flip-flop can be done in one of the three ways: by software using DI, system reset or by recognition of an interrupt request.

The following is the schematic diagram of interrupts.



Schematic Diagram of 8085 Interrupts.

Intel 8085 has two categories of interrupts: **maskable and non-maskable.**

The interrupts which can be masked off are called **maskable interrupts**. Masking is done by software. RST 7.5, RST 6.5 and RST 5.5 are maskable. TRAP is non-maskable. It need not be enabled. It cannot be disabled. It is not accessible to user.

Hardware and Software Interrupts

Interrupts caused by I/O devices are called *hardware interrupt.* The normal operation of microprocessor can also be interrupted by abnormal internal conditions or special instructions. Such an interrupt is called *software interrupt.* RST n instructions of 8085 are used for software interrupt.

When RST n instruction is inserted in a program, the program is executed upto the point where RST n is inserted. The internal abnormal or unusual conditions which prevent the normal processing sequence of a microprocessor are also called as exceptions. Example is divide by zero will cause the exception.

When several I/O devices are connected to INTR interrupt line, an external hardware is used to interface I/O devices. The external hardware circuit generates RST n codes to implement the multiple interrupt scheme.

Interrupts Call-Locations

When an interrupt occurs program is transferred to specific memory location. Then monitor transfers program from specific memory location to memory location in RAM, from where user can write program for interrupt service sub-routine (ISS). For TRAP, RST 7.5, 6.5, 5.5 the program is automatically transferred to specific memory locations without any external hardware.

Interrupt	Call-Location in Hex
TRAP	0024
RST 7.5	003C
RST 6.5	0034
RST 5.5	002C

Any interrupt for which hardware automatically transfers program to a specific memory location is known as *vectored interrupt.*

INTR CALL Locations

There are 8 numbers of CALL – locations for INTR interrupt. The following table contains CALL locations with hex code. For INTR external hardware is used to transfer program to specific CALL location. The hardware circuit generates RST codes for this purpose. The INTR line is sampled by the microprocessor in the last state of the last machine cycle of each instruction.

RST n	Hex-code	CALL-locations
RST 0	C7	0000
RST 1	CF	0008
RST 2	D7	0010
RST 3	DF	0018
RST 4	E7	0020
RST 5	EF	0028
RST 6	F7	0030
RST 7	FF	0038

\

RST 7.5, 6.5, 5.5

They are maskable interrupts. These interrupts are enabled by software using instructions using EI and SIM (Set Interrupt Mask). The execution of instruction SIM enables/disables interrupts according to bit pattern of the accumulator. The following diagram shows the accumulator contents for SIM instructions. Here bits 0-2 set/rest the mask bits of interrupt mask register for RST 5,.5,6.5,7.5. Here the bit 0 is for RST 5.5, bit 1 is for RST 6.5 and bit 2 is for RST 7.5. If a bit is set to 1 the corresponding interrupt is masked off i.e. disabled. If it is set to 0 , the corresponding interrupt is enabled. Bit 3 is set to 1 to make bits 0 to 2 effective. Bit 4 is an additional control for RST 7.5 . If it is set to 1, the flip flop for RST 7.5 is reset.

Triggering Levels.

TRAP is edge as well as level triggered. This means that TRAP should go high and stay high until it is acknowledged. In this way false triggering caused by noise and transients is avoided. The flip-flop is cleared when interrupt is acknowledged so that future interrupt may be entertained.

RST 7.5 is positive edge triggered interrupt. It can be triggered with a short duration pulse. RST 6.5 and 5.5 are level triggered interrupts. Triggering level for RST 6.5 and 5.5 has to be kept high until microprocessor recognizes these interrupts. If processor does not recognize these interrupts immediately, their triggering level should be held by external hardware.

Pending Interrupts.

When one interrupt request is being served, other interrupt may occur resulting in a pending request.

When more than one interrupts occur simultaneously, interrupt having higher priority is served and interrupts with lower priority remain pending.

After executing interrupt service routine (ISR) processor checks whether any other interrupt is pending using RIM instruction. If an interrupt is pending the processor executes its interrupt service subroutine before it returns to main program.

INTERFACING DEVICES AND I/O DEVICES

To communicate with outside world microcomputers use peripherals (I/O devices).

Commonly used peripherals are: A/D converter, D/A converter, CRT, printers, hard disks, floppy disks, magnetic tapes, etc.

Peripherals are connected to microcomputer through electronic circuits known as *interfacing circuits.*

The interface associated with output device converts output of microcomputers into the desired peripheral format.

Some of general-purpose interfacing devices:

i. I/O port
ii. Programmable Peripheral Interface (PPI)
iii. DMA controller
iv. Interrupt controller
v. Communication Interface

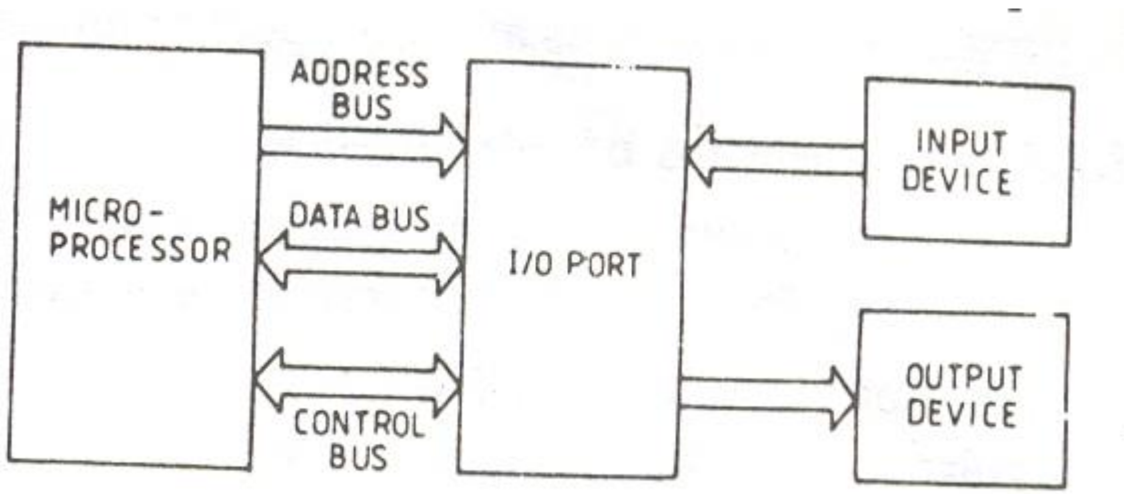
Special purpose interfacing devices are designed to interface a particular type of I/O device to microprocessor. Examples of such devices are:

i. CRT controller
ii. Floppy Disk Controller
iii. Key Board and Display Interface.

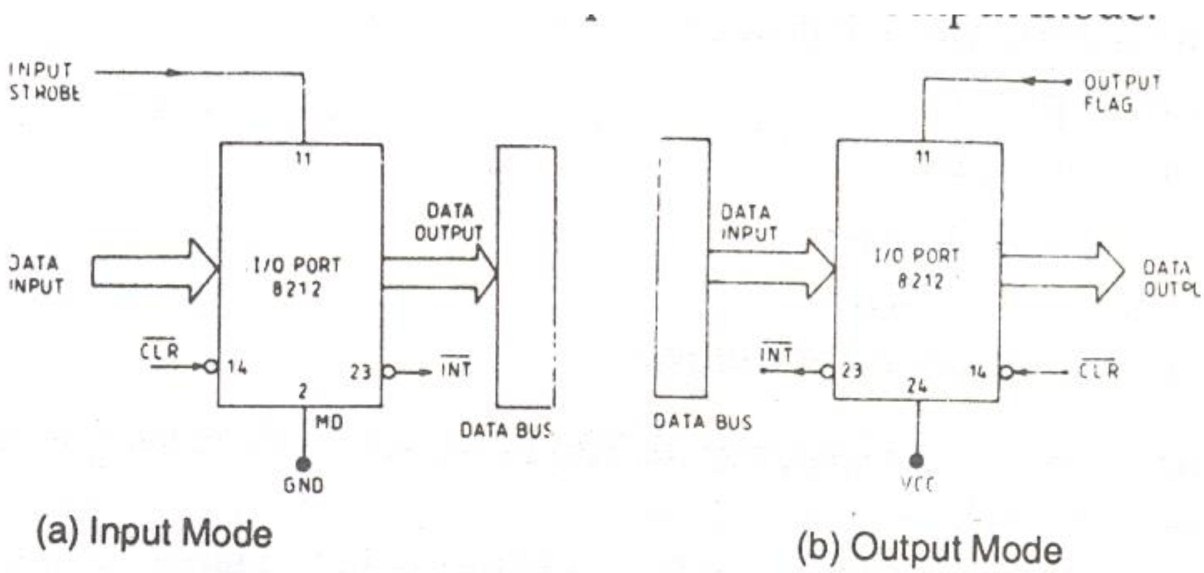
I/O PORTS

An input device is connected to microprocessor through an input port. An input port is a place for unloading data. An input device unloads data into port. The microprocessor reads data from input port.

Similarly, an output device is connected to microprocessor through an output port. As the output port is connected to output device, data are transferred to output device. An I/O port may be programmable or non-programmable. A non-programmable port behaves as an input port if it has been designed and connected in input mode. A port connected in output mode acts as an output port. But, a programmable I/O port can be programmed to act either as an input port or output port; electrical connections remain same. The diagram shows the interfacing of I/O device through the I/O port.



Intel 8212 is an 8-bit non-programmable I/O port. It can be connected to microprocessor either as an input port or an output port. If we require one input port and one output port two units of 8212 are required. One of them will be connected in input mode and other in the output mode. The following diagram shows the connections of 8212.



Intel 8155 is RAM with I/O ports. It contains 256 byte RAM, 3 I/O ports and a 14-bit timer/counter. The port A and Port B are 8-bits and Port C are 6 bits. Each port can be programmed either as input port or output port. The port C may be programmed as a control port of Port A and Port B. The 8155 is not widely used in Indian microprocessor kits. Generally in India We use 8255. It is called as Programmable peripheral Interface.

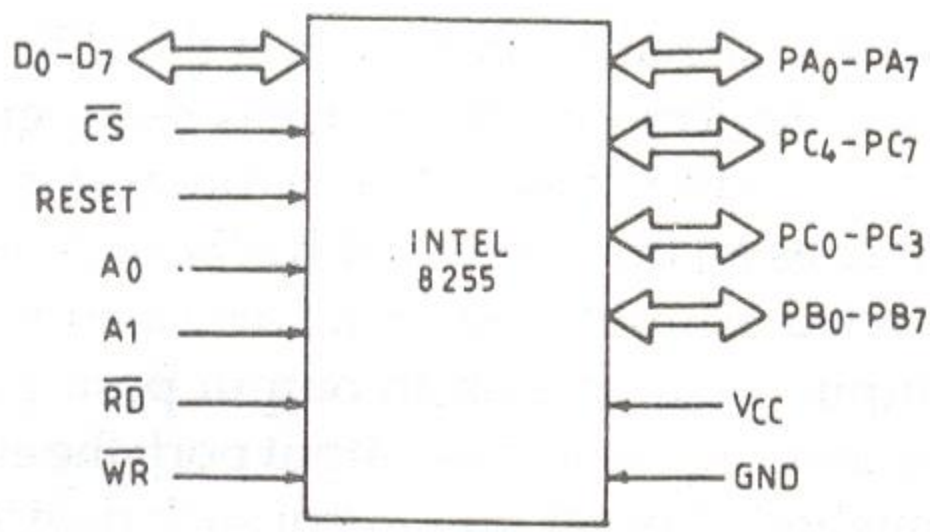
PROGRAMMABLE PERIPHERAL INTERFACE (PPI)

A programmable peripheral interface is a multiport device. The ports may be programmed in a variety of ways as required by the programmer . The device is very useful for interfacing peripheral devices. The Term PIA called as peripheral Interface Adapter is also used by some manufacturer.

INTEL 8255.

The Intel 8255 is a PPI. It has two versions, Intel 8255A and Intel 8255A-5. General descriptions for both are same. There are some differences in their electrical characteristics. Its main functions are to interface peripheral devices to microcomputer. It has three 8-bit ports, Port A, B and C. The Port C has been further divided into two of 4-bit ports, port C upper and port C lower. Thus a total of 4 ports are available, two 8-bit ports and two 4-bit ports. Each port can be programmed either as an input port or an output port.

ARCHITECTURE OF INTEL 8255A



Schematic Diagram of Intel 8255 A.

The diagram shows the ping configuration of 8255 A
It is a 40-pin IC package. It operates on a single $5V_{dc}$ supply.
Its important characteristics are as follows:

(i) Ambient temperature 0 to 70^{o}C.
(ii) Voltage on any pin: 0.5 V to 7 V.
(iii) Power dissipation 1 watt.
(iv) V_{IL} = Input low voltage = Minimum 0.5 V, Maximum 0.8V
(v) V_{IH} = Input high voltage = Minimum 2 V, Maximum V_{CC}
(vi) V_{OL} = Output low voltage = 0.45 V
(vii) V_{OH} = Output high voltage = 2.4 V
(viii) I_{DR} = Darlington drive current = Minimum 1mA, Maximum 4 mA of any 8 pins of the port.

The pins for various ports are as follows:
$PA_0 – PA_7$ 8 pins of port A
$PB_0 – PB_7$ 8 pins of port B
$PC_0 – PC_3$ 4 pins of port C_{lower}
$PC_4 – PC_7$ 4 pins of port C_{upper}

The important control signals are as follows:

1. $\overline{CS}$ (chip select). It is a chip select signal. The LOW status of this signal enables communication between the CPU and 8255.

2. $\overline{RD}$ (Read). When it goes LOW the 8255 sends out data or status information to CPU on data bus. In other words it allows CPU to read data from input port of 8255.

3.$\overline{WR}$ (Write). When it goes LOW the CPU writes data or control word into 8255. The CPU writes data into output port of 8255 and control word into control word register.

4. A_0 and A_1. The selection of input port and control word register is done using this in conjunction with $\overline{RD}$ and $\overline{WR}$. They are normally connected to LSBs of address bus.

For the 1st unit of 8255, i.e. 8255.1:

Port/Control word register	*Port/Control word register Address*
Port A	00
Port B	01
Port C	02
Control word register	03

Operating modes of 8255:

The 8255 has the following three modes of operation with are selected by the software

The 8255 has two 8-bit ports called Port A and Port B and two 4-bit ports called Port C upper and Port C lower . Each four ports of 8255 can be programmed to be either an input or output port.

1. **Mode 0 – Simple I/O**
   In this mode, a port can be operated as simple I/O port. Each of four ports of 8255 can be programmed to be either an input or output port.
2. **Mode 1 – Strobed I/O**
   Mode 1 is a strobed input/output mode of operation. The port A and Port B both are designed to operate in this mode of operation. When Port A and Port B are programmed in Mode1, six pins of Port C are used for their control. PC0 ,PC1 and PC2 are used for the control of the Port B , which can be used either as input or output port. If the Port A is operated as in input port, PC3, PC4 and PC5 are output. When Port A is operated as an output port , pins PC3,PC6 and PC7 are used for its control. The pins PC4 and PC5 can be used as either input or output.

3. **Mode 2 – Bidirectional port**
   Mode 2 is strobed bidirectional mode of operation. In this mode port A can be programmed to operate as a bidirectional port. The Mode 2 operation is only for port

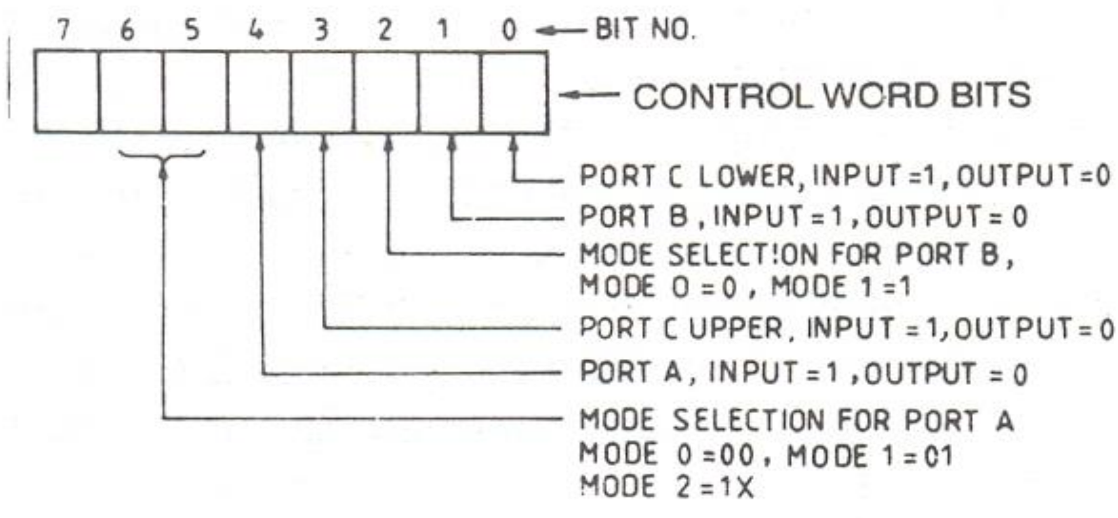
A. When Port A is pro0grammed in Mode 2, the Port B can be used in either Mode 1 and Mode 0.

The combination of Mode 1 and Mode 0 operation is also possible. For example, Port A is programmed to operate in Mode 1` and the Port B can be operated in Mode 0.

CONTROL GROUPS

For the control purpose 24 lines of I/O ports are divided into two groups, namely Group A and Group B. the group A contains the Port A and Port C_{upper}. The Group B contains the Port B and the Port C_{lower}. The function of each port is programmed as desired. A control word is formed which contain the information regarding the function and modes of the ports. The CPU outputs the control word to 8255. The control word initializes the ports. Each of control blocks accept commands from Read/Write control logic which is within 8255. The control blocks receive control words from internal data bus of 8255 and issue the proper commands to their associated ports.

CONTROL WORD OF 8255



Control Word Bits for Intel 8255

According to the requirement a port an programmed to act either as n input port or an output port. For programming the ports of 8255 a control word is formed. The bits of the control word are shown in the following diagram.

The control word is written onto the control word register which is within the 8255. No read operation of the control word register is allowed. The control word bit corresponding to a particular port is set to either 1 or 0 depending upon the definition of the port. If a particular port to be made an input port, the bit corresponding to that port is set to 1. For making a port an output port, the corresponding bit for the port is set to 0. Here we see the description of the control word as follows

Bit No. 0	-	It is for the Port Clower To make Port Clower an input port, the bit is set to 1 To make Port Clower an output port, the bit is set to 0.
Bit No 1	**-**	It is for Port B To make Port B an input port, the bit is set to 1. To make Port B an output port , the bit is set to 0.
Bit No.2	-	It is for the selection of the mode for the Port B. If the Port B has to operate in Mode 0, the bit is set to 0. For Mode 1 operation of the port B, it is set to 1.
Bit No.3	-	It is for the Port Cupper. TO make Port Cupper an input port, the bit is set to 1. To make Port Cupper an output port, the bit is set to 0.
Bit No.4	-	It is for Port A To make Port A an input port, the bit is set to 1. To make Port A an output port , the bit is set to 0.
Bit No. 5 and 6	-	These bits are to define the operating mode of the Port A. For the various modes of Port A these bits are defined as follows :

Mode of Port A	Bit No. 6	Bit No.5
Mode 0	0	0
Mode 1	0	1
Mode 2	1	0 or 1

For mode 2 , bit no. 5 is set to 0 or 1.

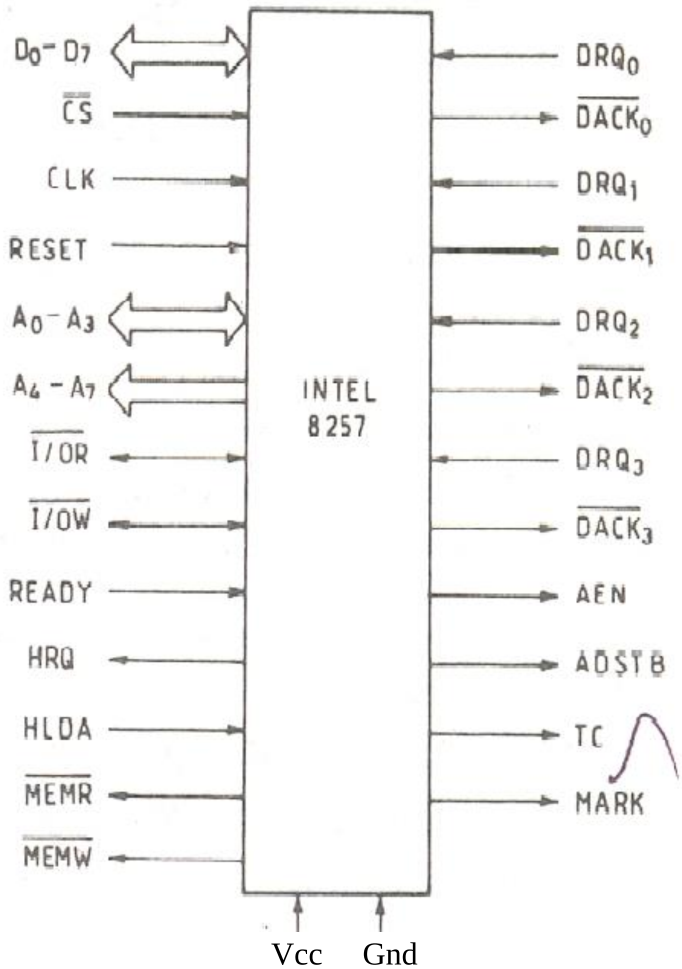
Bit No. 7	It is set to 1 if Port A, B and C are defined as I/O port. It is set to 0 if the individual pins of the Port C are to be Set or reset.

PROGAMMABLE DMA CONTROLLER

The bulk data transfer from fast I/O devices to the memory to or from the memory to I/O devices through the accumulator is a time consuming process. For this situation the direct memory access(DMA) technique is used. In DMA data transfer scheme, data are directly transferred from an I/O device to RAM or from RAM to an I/O device . For DMA data transfer, the data and address buses comer under the control of the peripheral device which wants DMA data transfer. The microprocessor has to relinquish the control of the address and data buses for DMA operation on the request of the I/O device. For DMA data transfer the I/O device must have its own registers to store byte count and memory address. It must also be able to generate control signals required for DMA data transfer. Examples of DMA controllers are 8257 and 8237

The Intel 8257

The Intel 8257 is a programmable DMA controller. It is a 4-channel programmable direct memory access controller. It is a 40 pin IC package and requires a single +5V supply for its operation. Four I/O devices can be interfaced to the microprocessor through this device. It is capable of performing three operations namely, read, write and verify.



During the read operation data are directly transferred from the memory to the I/O device. During the write operation data are transferred from the I/O device to the memory. On receiving a request from an I/O device, the 8257 generates a sequential memory address which allows the I/O device to read or write directly to or from the memory. Each channel incorporates two 16-bit registers called as DMA address register and byte count register. These registers are initialized before a channel is enabled. Initially, the DMA address register is loaded with the address of the first memory location to be accessed.

During DMA operation it stores the next memory location to be accessed in the next DMA cycles. 14-LSBs of the byte count register store the number of bytes to be transferred. 16384 bytes of data can directly be transferred to the memory from the I/O dev ice or from the memory to the I/O device.

The important pins of 8257 are as follows :

1. **DRQ0-DRQ3 :** These are DMA request lines . An I/O device sends its DMA request on one of these lines. A HIGH status of the line generates the DMA request.
2. **A0-A7 :** These are address lines. A0-A3 are bidirectional lines. In the master mode these lines carry 4 LSBs of 16-bit memory address generated by the 8257. In the slave mode these lines are input lines.
3. **D0-D7 :** These are data lines. These are bidirectional three state lines. While programming the controller the CPU sends data for the DMA address register and byte count register and mode set register through these data lines. During DMA cycle, the 8257 sends the 8 MSBs of the memory address through these lines at the beginning of the cycle.
4. **AEN : Address Latch Enable.**
5. **ADSTB :** A HIGH on this line latches the 8 MSBs of the address, which are sent on D-bus, into Intel 8212 connected for this purpose.
6. **$\overline{\text{CS}}$:** It is chip select.
7. **$\overline{\text{I/OR}}$:** I/O read . It is a bidirectional line. In output mode it is used to access data from the I/O device during the DMA write cycle.
8. **I/OW :** I/O write. It is a bidirectional line. In output mode it allows the transfer of data to the I/O device during the DMA read cycle. Data is transferred from the memory.
9. **$\overline{\text{MEMR}}$:** Memory Read
10. **$\overline{\text{MEMW}}$:** Memory Write
11. TC : Byte count
12. CLK : Clock

Etc…

An I/O device sends its request for DMA transfer through one of the DRQ lines. On receiving the DMA request for DMA data transfer from an I/O device, the Intel 8257 sends the hold request to the CPU through the HRQ lines. HRQ stands for Hold request. The 8257 receives the hold acknowledge signal from the CPU through the HLDA line. After receiving the hold acknowledge from the CPU, it sends DMA acknowledge to the I/O device through the $\overline{\text{DACK}}$ line. The memory address is sent out on address and data lines. The 8257 sends the 8 MSBs of the memory address over D-bus. These 8 MSBs of the memory address are latched into 8212 using ADSTB signal.

ADSTB is similar to ALE of Intel 8085. For DMA read cycle,

in which data are transferred from memory to I/O devices, they use two control signals $\overline{MEMR}$ and $\overline{I/OW}$ are issued by 8257. The $\overline{MEMR}$ enables the addressed memory for reading data from it.

The byte count is decremented by one after the transfer of one byte of data. When byte count becomes zero, TC goes high indicating that the data transfer using DMA is complete. The four DMA channels are programmed either in a fixed priority mode or rotating mode of operation. READY line is used by slow memory or I/O devices.

DELAY SUBROUTINE

A delay program is used to provide the desired delay in industrial control before issuing control signal by microcomputer. To generate delay a few registers of microprocessor are loaded with desired numbers and then decremented to zero. The delay time depends on numbers loaded in registers. Delay can also be generated using 8253.

DELAY SUBROUTINE USING ONE REGISTER

Labels	Mnemonics	Operands	Comments
	MVI	B,10H	Get 10 in register B
LOOP	DCR	B	Decrement register G
	JNZ	LOOP	Has the count of the register B become zero ? No jump to LOOP. Yes , proceed ahead.
	RET		

The above program explain the delay subroutine using one register. To generate very small delay only one register can be used. In the above program register B is loaded by 10H(16 decimal). Then the register B is decremented and program moves in a loop till the content of register B becomes zero. After this the program returns to the main program. The delay time can be calculated as follows. To calculate the delay time it is examined that how many times each instruction of the above program is executed. The number states required for the execution of each instruction is as follows :

Instructions		States
MVI	B,10H	7
DCR	B	4
JNZ		7/10
RET		10

The instruction JNZ takes 10 states when the content of register B is not zero and the program jumps to the label LOOP. JNZ takes only 7 states when the content of register B has become zero and the program proceeds further to execute RET instruction. The instruction MVI B, 10H and RET are executed only one. The instruction DCR B is executed 16 times. The instruction JNZ is executed 16 times, out of which 15 times the program jumps to the label LOOP as the content of register B has not become zero, and takes 10 states each time. At last when the content of register B becomes zero, JNZ is executed and the program proceeds further. The last execution of instruction JNZ takes only 7 states. The number of states required for the execution of each instruction and how many times each instruction has been executed is as follows :

Instruction	How many times The instruction is Executed	States
MVI B,10H	1	7 x 1
DCR B	16	4 x 16
JNZ	16	10 x 15 + 7 x 1
RET	1	10 x 1

Hence total states = 7 x 1 + 4 x 16 + (10 x 15 + 7 x 1) + 10 x 1
= 7+64+150+7+10
= 238

Time for one state for Intel 8085 is 320 ns

Delay time = 238 x 320 x 10^{-9} second
= 0.238 x 0.320 millisecond
= 0.07616 millisecond.

To generate maximum delay register B is loaded by FF (255 decimal) . The maximum delay using one register is as follows :

= 7 x 1 + 4 x 255 + (10 x 254+ 7 x 1) + 10 x 1
= 7 + 1020 + 2540 + 7 + 10
= 3584 states
= 3584 x 320 x 10^{-9} second
= 1.11688 milliseconds.

Delay subroutine using Register pair

Label	Mnemonics	Operands	Comments
	LXI	D, FFFF	Get FFFF in rp D-E
LOOP	DCX	D	Decrement Count
	*MOV	A, D	Move content of D to A
	ORA	E	Check if D and E are zero
	JNZ	LOOP	If D-E is not zero, jump to LOOP
	RET		Return to main program.

States required for each instruction of above program are:

Instruction		States
LXI	D	10
DCX	D	6
MOV	A, D	4
ORA	E	4
JNZ		7/10
RET		10

If the count in register pair D-E is N total number of states are:

States = 10 + N (6 + 4 + 4) + (N -1) x 10 + 1 x 7 + 10
= 24 N + 17

Delay = (24 N + 17) x time for one state.

Maximum delay will occur when count N = FFFF hex
= 65,535 decimal.

Maximum delay = (24 x 65,535 + 17) x 320 x 10^{-9} second
= 20.97664 milliseconds.

This delay subroutine is given in the monitor program of 8085 microprocessor-kits. The count in register pair D-E is to be stored by programmer. From DCX D to RET of program are stored in monitor program, memory location being 03BC to 03C2. One can call delay subroutine as shown below:

LXI	D, 5000H	Get count = 5000H
CALL	03BC	Call DELAY

5000H is a typical count value desired by programmer. One can take any count value in between 0000 – FFFF as required. If required delay is of few milliseconds the delay subroutine given in monitor's program may be called.

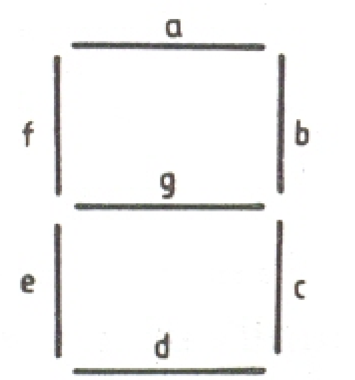
DELAY SUBROUTINE USING THREE REGISTERS

PROGRAM

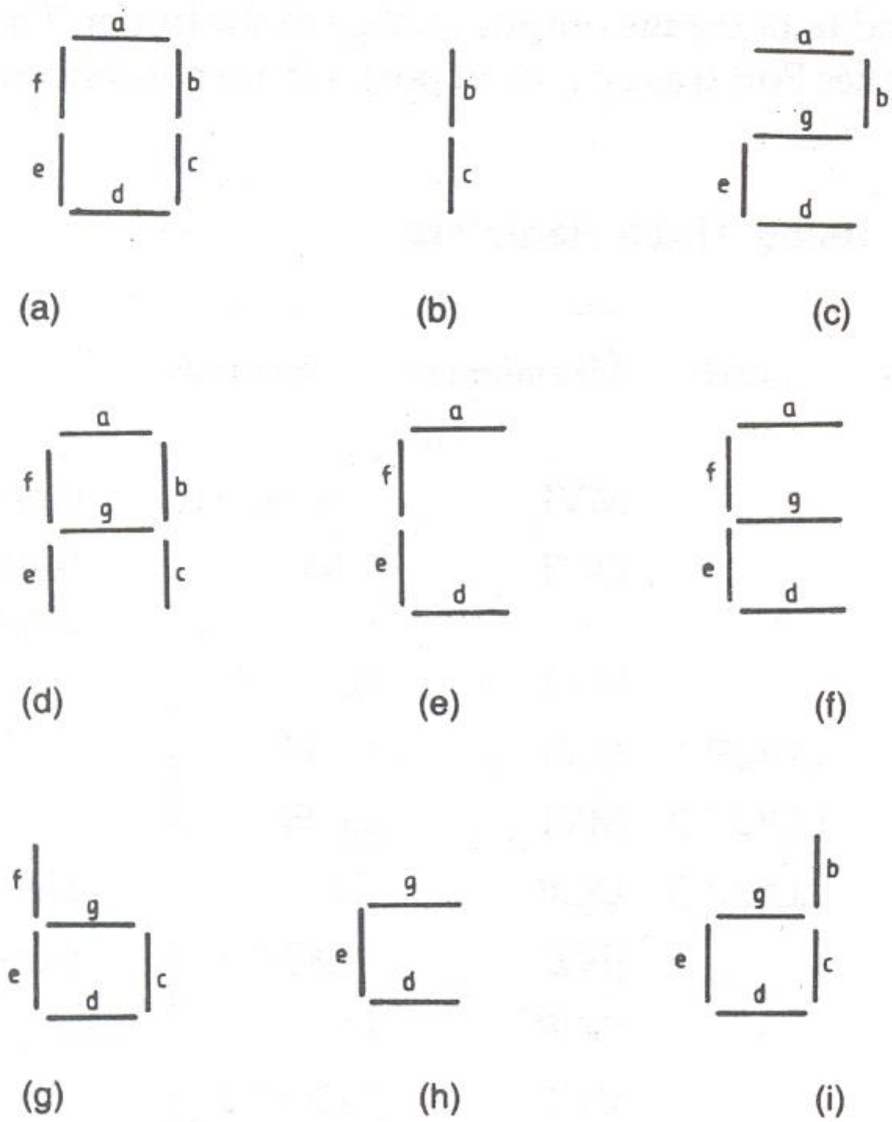
Memory address	*Machine Codes*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2400	3E, 98		MVI	A, 98 H	Get control word.
2402	D3, 03		OUT	03	Initialize port for LED display.
2404	06, 50		MVI	B, 50 H	
2406	0E, FF	LOOP 1	MVI	C, FF	
2408	16, FF	LOOP 2	MVI	D, FF	
240A	15	LOOP 3	DCR	D	Delay subroutine with
240B	C2, 0A, 24		JNZ	LOOP 3	three registers.
240E	0D		DCR	C	
240F	C2, 08, 24		JNZ	LOOP 2	
2412	05		DCR	B	
2413	C2, 06, 24		JNZ	LOOP 1	
2416	3E, 01		MVI	A, 01	
2418	D3, 01		OUT		Output for LED
241A	76		HLT		Stop.

SEVEN SEGMENT LED DISPLAY

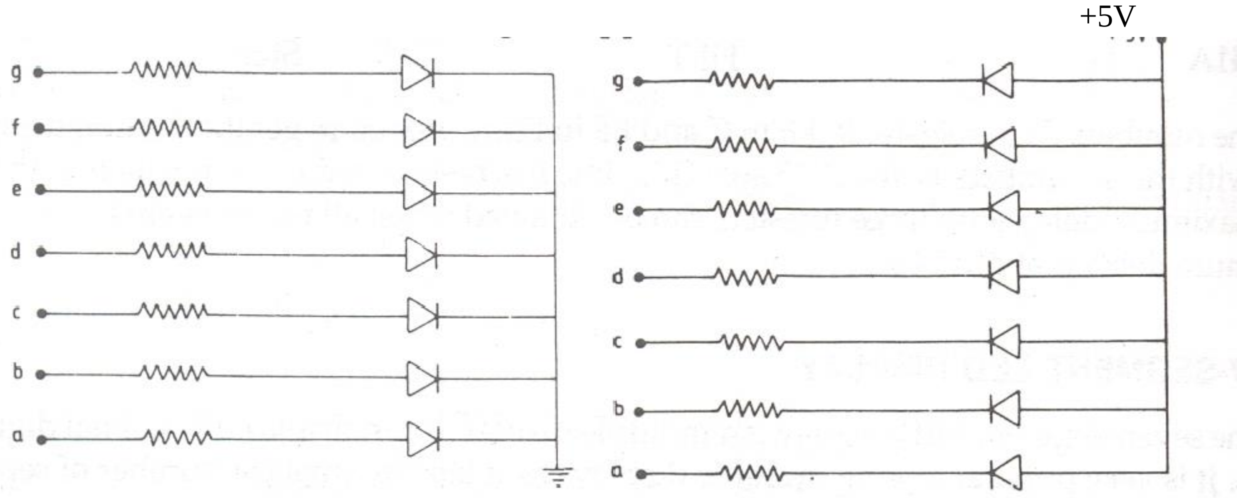
The seven-segment LED display is a multiple display. It can display all decimal digits and some letters. It is very popular among multiple displays as it has smallest number of separately controlled light emitting diode (LED). Multiple displays of 9-segment LED, 14-segment LED and dot matrix type are available. These displays give better representation of alphanumeric characters but require complex circuitry.



In 7-segment displays there are 7 LED. Each LED can be controlled separately to display a digit or letter the desired segments are made ON. The following diagram shows a digit or letter the desired segments are made ON. There are two types of seven segment displays , they are common-cathode type and common-anode type.

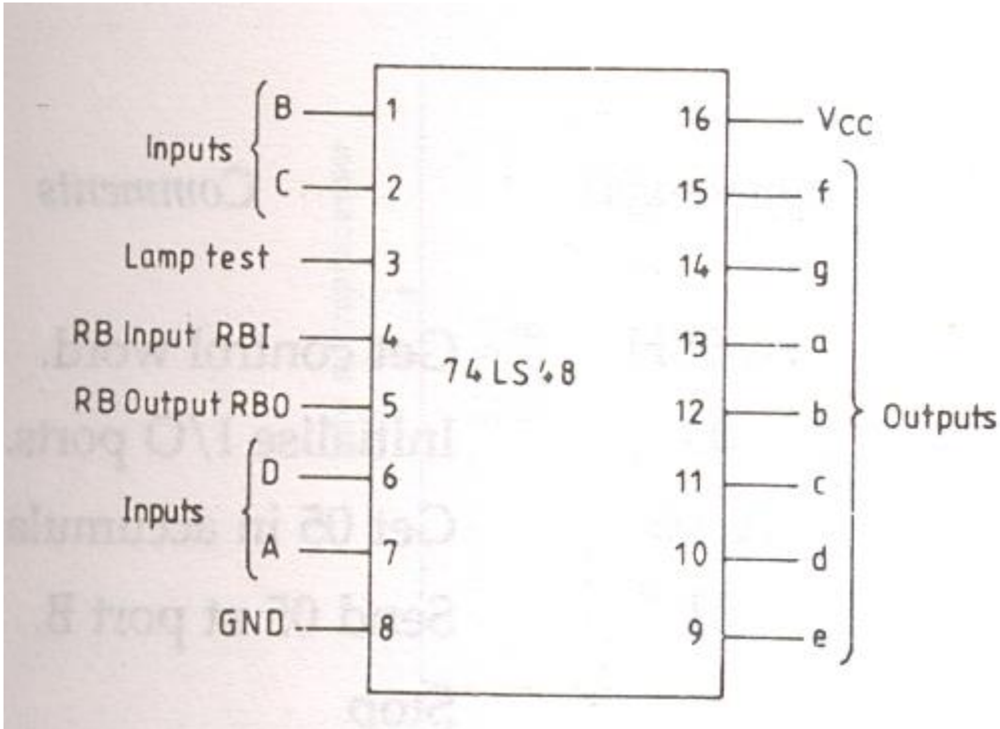


In a common cathode type display all the 7 cathodes of LEDS are tied together to the ground as in the diagram. When a + 5 V d.c. is applied to any segment, the corresponding diode emits light. Thus applying logic 1, that is positive logic to the desired segments. The desired letter or decimal number can be displayed. In a common anode type display all the 7 anodes are tied together and connected to + 5 V supply as in the diagram.



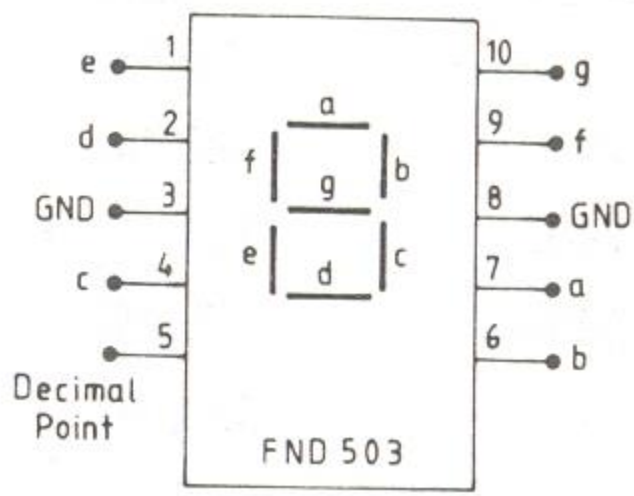
(a) Common-Cathode 7-Segment Display (b) Common-Anode 7-Segment Display

The seven segment displays are not connected to I/O ports directly. They are connected through buffers or drivers or decoders. 7446A, 74L46, 7447, 74L47 and 74LS47 are decoders/drivers for common anode type seven segment displays. The following diagram shows the pin configuration of 74LS48



FND 500 and FND 503

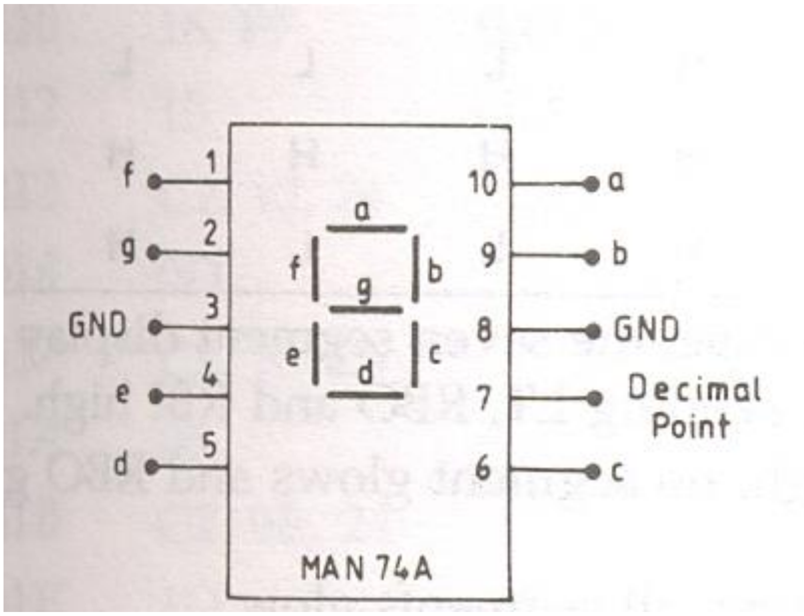
They are common-cathode 7-segment displays. The diagram shows the pin configuration of FND 503.



FND 507/510. FND 507 and FND 510 are common-anode 7-segment displays.

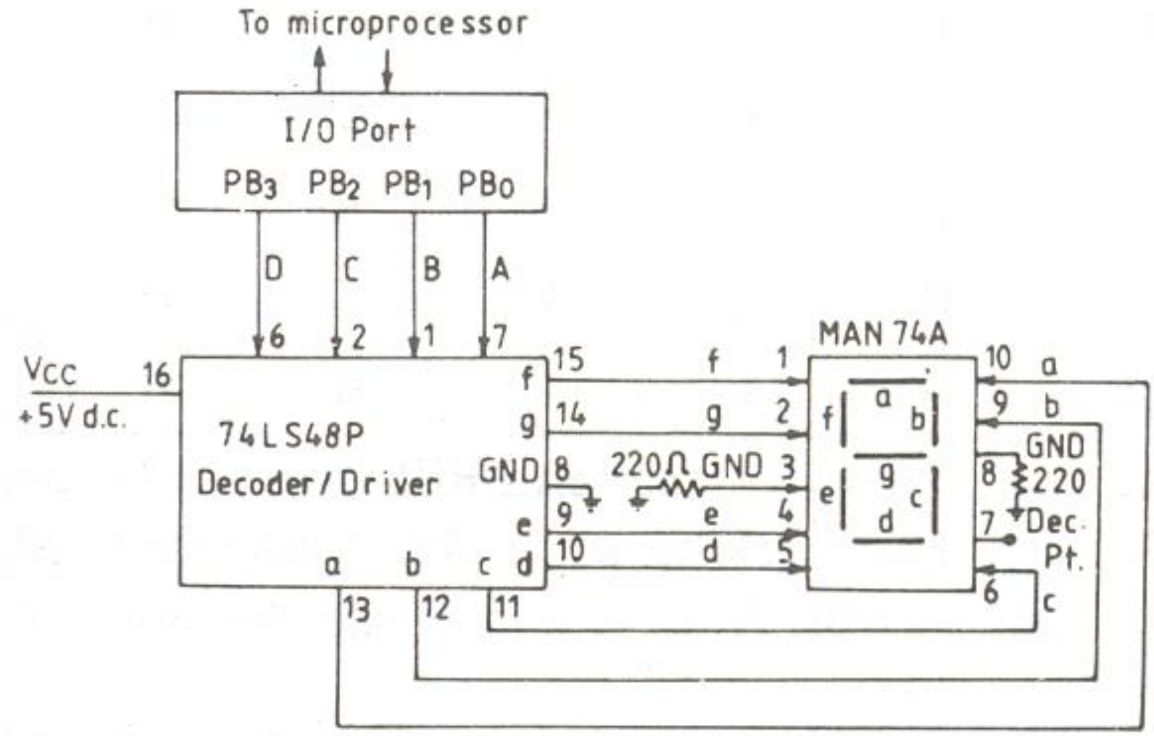
MAN 74 A

MAN 74A is a common-cathode 7-segment display. The pin configuration is as shown below.



MAN 72 is a common anode 7 segment display. If pin diagram of a 7-segment display is not known, it can be checked using multimeter. Set multimeter for ohm measurement. Connect the terminals to any two pins of the 7-segment display. Find the correct pins by trial and error.

The following diagram shows the interfacing of decoder/driver 74LS48 and 7-segment display MAN 74A with microprocessor as shown



The program to display number is as follows, this program display a number 5.

Mnemonics	**Operands**	**Comments**
MVI	A,98H	Get control word.
OUT	03	Initialise I/O ports
MVI	A,05	Get 05 in accumulator
OUT	01	Send 05 at port B.
HLT		Stop

The control word 98H makes the Port B an output port. The microprocessor outputs 05 at the Port B. The pins PB0-PB3 of the port B are connected to the decoder / driver 74LS48. Thus the binary bits corresponding to the decimal number 5 are applied to 74LS48 and it is displayed by the 7-segment display. 0 is the MSB of the decimal number 05. The logic for 0 is output on the pins PB4-PB7. These pins are not connected anywhere and consequently 0 is not displayed. If we use decoder/drivers and 7-segment displays , No.1 display will display 5, No.2 display will show 0. Thus two units of display will display 2 desired decimal numbers.

Display of Decimal Numbers 0 to 9

A program has been developed to display the decimal numbers 0 to 9, one by one. First of all decimal number 0 is displayed for some time, then 1 and so on. After displaying all numbers, program will repeat the process of displaying the numbers again and again. A delay subroutine has been included in program. The time for displaying the numbers can be adjusted by adjusting data for delay subroutine. The program is as follows :

Labels	Mnemonics	Operands
	MVI	A,98H
	OUT	03
ABOVE	LXI	H,2500H
	MOV	E,M
LOOP	INX	H
	MOV	A,M
	OUT	01
	MVI	B,0F
BEHIND	MVI	C,FF
BACK	MVI	D,FF
GO	DCR	D
	JNZ	GO
	DCR	C
	JNZ	BACK
	DCR	B
	JNZ	BEHIND
	DCR	E
	JNZ	LOOP
	JMP	ABOVE

If this program is executed using the one displaying unit of MAN 74A , the 7-segment display will display the decimal numbers 0 to 9 one by one. If this program is executed using two display units (not in diagram) the first unit displays 0 to 9 and the second unit display always 0.

Display of Alphanumeric Characters

To display of alphanumeric characters the segments of a 7-segment LED are controlled independently. Buffers are used to interface 7-segment displays to microprocessor. Hex buffer 7407 can be used for the purpose. As 7407 is an open collector buffer, it uses pull up resistor at its output terminals. It has several codes for capital/small letters and some special characters. To display a decimal number/letter the corresponding code is to be output to port B by microprocessor.