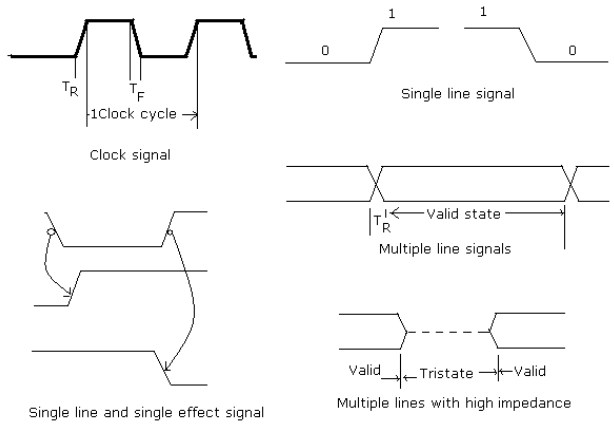
B.Sc. Part II
(ELECTRONICS)
Semister IV
Paper VIII Interfacing with Microprocessor and VHDL Programming

UNIT 1 (10)

Interfacing Techniques: Data transfer methods: Unconditional, Polling, Interrupts, Wait state generation. Study of 8255PPI, its operating modes, control word,

Timing diagram: µp performs two operations instruction fetch & instruction execution. During operation µp fetches instruction into instruction register & then performs instruction execution operation. There are two types of execution operations, internal if the operands are in internal registers & external operation if the operands are available in memory or in I/O. Internal operations are arithmetic, logical, decision making & internal data transfer operations. External operations are data read, data write, stack read, stack write, I/O read & I/O write operations.



To study how µp executes instructions, which signals are activated, which signal act as timing reference, what is the duration of each signal, we have to study the graphical representation of steps with respect to time i.e. clock called Timing diagram.

Practical clock contains some rise time T_R & fall time T_F. The status of single line will be LOW or HIGH. It requires some time to go from one stare to another. Multiple lines are grouped & shown in the form of block. At the time of signal change crossing are shown & for rest of the time the content of those lines are valid. A change in signal line causes change in other signal which is shown in figure.

Introduction to machine cycles: To execute an instruction the µp first takes the opcode from memory, this is called as opcode fetch. Then this instruction is decoded by instruction decoder & µp performs the operation specified i.e. execution. µp performs an operation in a specific time period (clock cycle). Each clock cycle is called as T state. The number of T states required to perform an operation is called as machine cycle. The number of machine cycles required to fetch & execute an instruction is called as instruction cycle. The machine cycles of 8085 are: opcode fetch, memory read, memory write, I/O read, I/O write, interrupt acknowledge and bus idle.

Opcode fetch machine cycle: Every instruction is a binary word containing 1s & 0s. Hexadecimal representation of the binary word is called as opcode of that instruction. Instructions are written by entering their opcodes in the memory of the µp.

The very first step in an instruction cycle is to fetch the opcode of the instruction from memory. Therefore, opcode fetch is the first machine cycle in any instruction cycle.

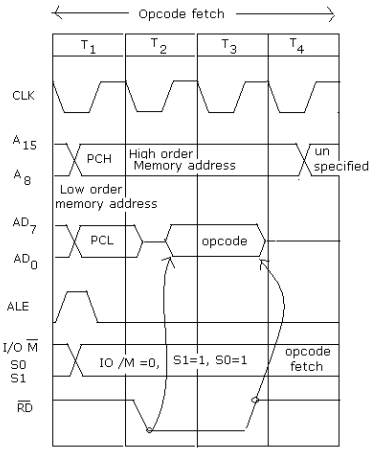


Fig: Timing diagram of opcede fetch cycle

T1: 1) Address of instruction to be fetched exists in program counter (PC). During T1, contents of PC are placed on high(A8-A15) & low (AD0-AD7) order addresses buses.

2) ALE signal goes high.

3) Status signal & IO/M reflect about the undergoing machine cycle (opcode fetch- S0=S1=1 & IO/M =0).

T2: 1) Rd signal is activated to initiate memory read operation.

2) µp waits for the opcode to appear on AD0-AD7 bus.

3) After access time of memory, contents of the addressed memory location, i.e. the opcode, appear on the AD0-AD7 bus.

4) The PC is incremented by 1.

T3: 1) The µp reads the opcode from the bus & places it in IR.

2) RD signal is deactivated.

T4: Instruction is decoded & its execution begins. In some instructions, two or more T states are present in the opcode fetch cycle.

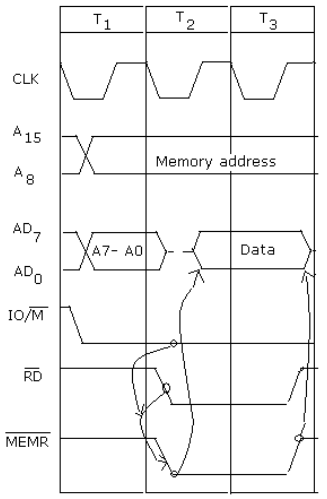


Fig: Timing diagram of Memory Read Cycle

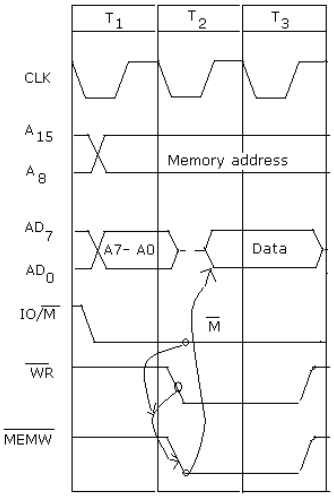


Fig: Timing diagram of Memory write cycle

Memory read Machine cycle: During memory read machine cycle, a data byte is read from memory by selecting a memory location. It has 3 T-states.

T1: 1) Address of memory location is placed on address bus (A0-A15) to select the memory location.

2) ALE goes high.

3) IO/M, S1 & S0 (0, 1, 0) indicate that memory read machine cycle is undergoing.

T2 & T3: 1) RD signal goes low to initiate read operation.

2) The data byte from selected memory location becomes available on data bus after the response time of the memory.

Memory Write Machine cycle: In this machine cycle, a data byte is written to memory after selecting a memory location by selecting it. It has 3 T states.

T1: 1) It is similar to T1 of the memory read cycle, except the indication on IO/M, S1 & S0 lines, which is (0, 0, and 1).

T2 & T3: 1) The µp places data on AD0-AD7 bus & activates WR signal. This makes the data to be written at the selected location.

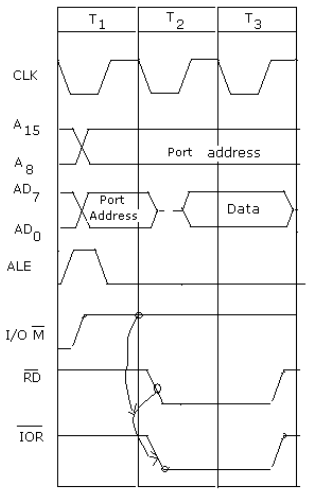


Fig: Timing diagram of I/O Read Cycle

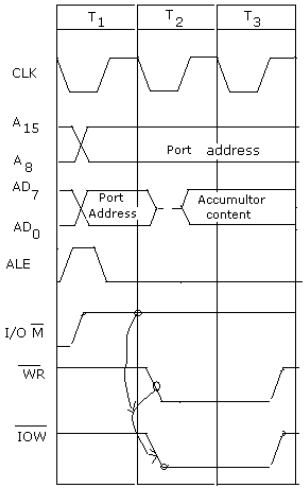


Fig: Timing diagram of I/O write Cycle

I/O Read: A data byte is received from an I/O device by selecting it.

Address of an I/O device is called as the port address. Port address is always of 8 bits. The µp sends same address on higher (A8-A15) & lower (AD0-AD7) order address buses.

T1: Operations performed are similar to a memory read cycle.

1) The port address is sent on both the buses for selecting I/O device.
2) ALE goes high.
3) IO/M , S1, S0 become 1, 1, 0 to indicate I/O read cycle

T2 & T3: 1) Rd signal goes low.

2) Data is placed on AD0-AD7 bus by the I/O device. It is read by the µp.

I/O Write: It has three states.

T1: It is similar to T1 of I/O read cycle. I/O M , S1, S0 become 1, 0, 1.

T2 & T3: WR signal is activated after placing the data on theAD0-AD7 bus.

Interfacing Techniques: The µp receives the data from the input unit & sends the results to the output unit. In all the µp based systems, these two units are must. The process of data transfer between the µp & the I/O units is referred to as input/output or I/O. The input or output devices or circuits are called as peripherals. For proper data transfer between µp & the peripherals a proper flow of address, control & data signals is necessary. It necessitates a well designed logical circuit. A process of designing the logic circuit & writing instructions to transfer the data between the µp & peripherals is called as interfacing.

Modes of data transfer: the data between the µp & I/O device is transferred in either of the two modes, Parallel data transfer & serial data transfer.

Parallel data transfer: The µp 8085 handles the binary data in form of a word of 8 bits. In the parallel data transfer, all 8-bits of the word are transmitted & received simultaneously. For an 8-bit data transfer 8 lines are

required. As all the bits of data are transferred simultaneously, the parallel data transfer is fast but the hardware required to implement it is relatively large. The parallel mode of data transfer is used for the peripherals like LEDs, seven segment displays, thumb wheel switches, keyboards, data converter, memory etc.

Serial data transfer: In serial mode of data transfer, the data is transferred bit wise. As a single bit is transferred at a time, a single line is enough for the data transfer. The word is first converted into a stream of 8-bits & then transferred over the single line. On the receiving end the bits are collected & converted into 8-bit words.

The serial data transfer is slow because each bit is transmitted separately but there is tremendous saving in the hardware required to implement the data transfer. The serial mode of data transfer is used for the peripherals like printers, CRT terminals, teletypes, cassette tapes etc.

Parallel I/O: In parallel I/O the µp sends or receives all the bits of the data word simultaneously on the data bus. As the data bus is shared by all the peripherals, simultaneous communication with all the peripherals is not possible. Therefore, the µp selects a device for communication by sending its address on the address bus. Each device, thus selected is called as a port & the address is called as the port address. The ports from which the µp receives the data are called as input ports, & where the data is transmitted is called as output ports.

Data transfer techniques: 1) µp controlled data transfer: It is used when peripheral device is slower than µp. Ex key boards, displays, D to A & A to D converters, printers, CRT terminals, memories 2) Peripheral controlled data transfer: It is used when peripheral device is faster than µp. In such a case, the peripheral device sends a HOLD signal to µp & requests it to release the bus control. The µp release the control on the buses & remains idle. The data is then transferred under control of the peripheral devices. This process of data transfer is called Direct Memory Acess(DMA).

1) µp controlled data transfer: The µp controlled data transfer is further classified in two types

i) Unconditional or simple data transfer: in this type it is assumed that the peripheral device is always ready for the data transfer. µp simply enables the peripheral device, transfers the data & goes to the execution of next instruction. e.g. to display data on LED or Seven segment display.

ii) Conditional data transfer: In this case the readiness of peripheral device is checked during a data transfer. There are several types

a) **Data transfer with polling or status check I/O:** In many cases, the data is not always available at a port. µp is kept in a loop to check whether data is available. As the data becomes available, it is read & the µp goes to execution of next instruction. This mode of data transfer is called as polled or status check I/O. e.g. Switches or key board is connected to one of the port .then µP has to wait until key to be pressed following program shows how it is.

```
Start :   IN Port—Take the data in
          CPI FF  compare data with FF if key is not pressed answer of comparison is zero
          JZ start  if answer is zero take again data
```

In this way it is in loop until key is not pressed

b) **Data transfer with interrupt:** In polling the µp is kept in the loop until the data is available, which wastes the µp time. In interrupt driven I/O, µp keeps on doing its operation. When a peripheral is ready it interrupts the µp, then µp stops current execution & transfers the data with the peripheral & returns the main program. This is very efficient way of data transfer.

c) **Data transfer with READY signal:** When the peripheral cannot complete the data transfer within the allowed read or write time, it can make the microprocessor wait for some time by requesting it on the READY line. As long as the ready signal is low, the µP waits in the same state. Thus the slow peripherals (ex. Printer) get an additional time to complete the data transfer.

d) **Data transfer with Handshake signal:** In this type of data transfer method a few signals indicating the readiness are passed before the data transfer. These signals are called as hand shake signals. The handshake signals are used to indicate readiness & to synchronize the timing of the data transfer. This type of data transfer provides a very safe way of data transfer. The chances of losing a data are almost nil. But an additional hardware is required for implementation.

Wait state generation: The wait state is added in a machine cycle if the READY is found low. The READY signal is used to check readiness of the memory or peripheral device. If READY signal is high during a read or write cycle, it indicates that the selected device is ready to send or receive the data. If READY is low, the device is not in position to do so & µp waits for an integer number of clock cycles until READY goes high. During wait state, the address bus, data bus & control bus are maintained as before.

Addition of 1 wait state: 1) The ALE is generated in first clock cycle of every machine cycle, it is connected as clock of FF1 & its D1 input is HIGH. So Q1 will be 1.

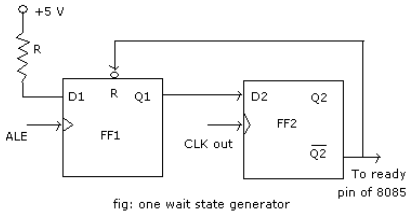
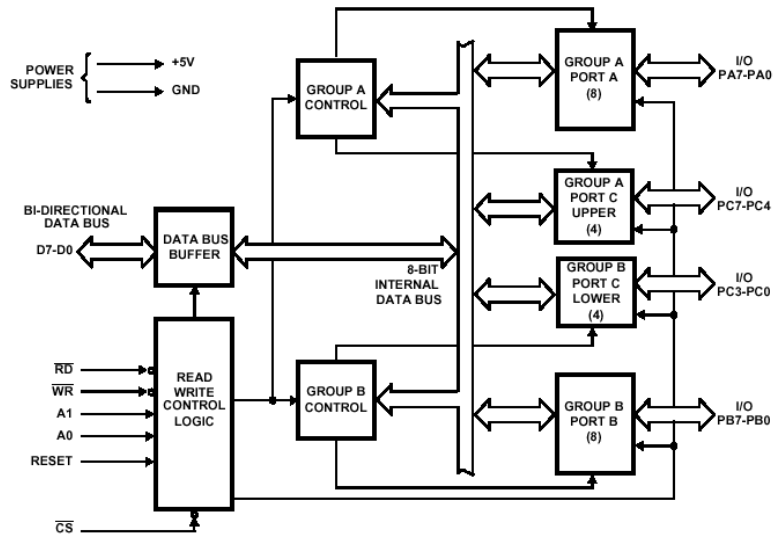


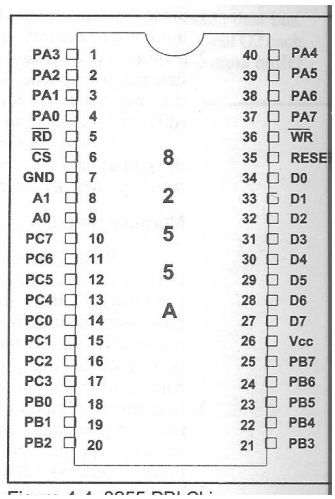
fig: one wait state generator

2) The clock out is connected as CLK in of FF2 & during T1 its D input is HIGH because of FF1. On next rising pulse Q2 becomes 1, which resets FF1& ready input of 8085, so it will reset the FF & introduce wait state after T2.
3) As ALE is used as clock input to FF1, it will be available in next machine cycle, so output of FF1 will remain low for that machine cycle.
4) As input to FF2 is low on next pulse, it will make Q2 low,& Q2 high. This removes ready input low to high.
5) 8085 will check its ready input in wait state. As it is high, it will go for next states.
6) The output of FF2 will make reset input of FF1 high. But the output of FF1 will not change as ALE is not present.

Study of 8255PPI: External devices like LED, Switches, Printers etc.(Known as Peripherals) are not connected or interfaced directly with microprocessor for this purpose Programmable Peripheral Interface IC 8255 is used. IC is programmable means we can use it as Input or Output device by giving control word to it.

Block diagram of 8255 is as shown in fig. below.





PA0-PA7	I/O	Port A Pins	RESET	I	Reset pin
PB0-PB7	I/O	Port B Pins	RD¯	I	Read input
PC0-PC7	I/O	Port C Pins	WR ¯	I	Write input
D0-D7	I/O	Data Pins	A0-A1	I	Address pins
			CS ¯	I	Chip select
			Vcc , Gnd	I	+5volt supply

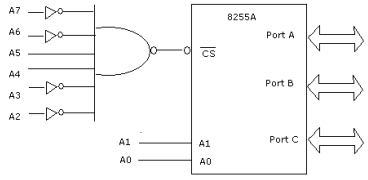
Ports:

The 8255 IC has three I/O ports through which we communicate between microprocessor and external devices. These ports are named as port A, port B, port C. Each port have eight pins these pins are named as PA0, PA1, PA2, ..PA7. Similarly for Port B pins are PB0-------PB7. Port C has also 8 pins PC0---to PC7. But Port C is divided in to two groups Port C lower (Pins PC0,PC1,PC2 and PC3) and Port C Upper (PC4,PC5,PC6 and PC7) Port C lower

and upper can be used separately as input or output port or it can be used single as input or output port. Similarly PORT A & PORT B can be used as input port (to transfer data from peripherals to microprocessor) or can be used as output port (to transfer data from microprocessor to peripherals)

Read /Write logic Control logic: The function of this block is to control all internal & external data transfer & to coordinate working of different blocks. The read/write & control logic receives following inputs from the µp system

$\overline{CS}$ chip select: When a low level appears on this pin, the chip is selected & communication between 8255A & the µp is enabled. Normally, the $\overline{CS}$ signal is generated by decoding A2-A7 address lines. A typical address decoding circuit is as shown in fig. this circuit produces a $\overline{CS}$ signal whenever the address bits are 001100X.



A_0 & A_1: These pins are used to select PORT A,PORT B,PORT C or control register.

Ao,A1 pins are connected to address bus pins Ao,A1 of µp and remaining pins A2----A7 are connected to CS^- with logic combination hence if $CS^- = 0$

A_O	A_1	Action
0	0	Port A is selected
0	1	Port B is selected
1	0	Port C is selected
1	1	Control register

Control Register: Control register is as shown in following table:

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Hex Address	Port
0	0	1	1	0	0	0	0	30 H	A
0	0	1	1	0	0	0	1	31 H	B
0	0	1	1	0	0	1	0	32 H	C
0	0	1	1	0	0	1	1	33 H	Control register

$\overline{RD}$: This signal when active, enables the read operation. When $\overline{RD}$ goes low, data from a selected port of 8255A is send to the µp. The $\overline{RD}$ input is connected to IOR line of the µp system.

$\overline{WR}$: This signal enables the write operations when $\overline{WR}$ goes low, a data byte is written from the data bus to a selected port or to the control register.

Following table lists the operations selected by the five input signals.

$\overline{CS}$	A_1	A_0	$\overline{WR}$	$\overline{RD}$	operation
0	0	0	0	1	Write to port A
0	0	1	0	1	Write to port B
0	1	0	0	1	Write to port C
0	1	1	0	1	Write to control register
0	0	0	1	0	Read port A
0	0	1	1	0	Read port B
0	1	0	1	0	Read port C
0	1	1	1	0	Illegal condition
0	X	X	1	1	No operation, data bus tristated
1	X	X	X	X	Device not selected

RESET: A high on this input clears the control register of 8255A & all ports are set to the input mode. The RESET OUT signal of 8085A is connected to this pin.

Data Bus Buffer: It is a tristate buffer. It is in the high impedance state either when the device is not selected or when the read or write operations are not being performed. All the data transfer between the µp & the 8255A are performed through the data bus buffer.

Group A & group B controls: these control the operations of the ports. The Group A control block deals with the port A & port C_{upper}. The Group B controls block deals with the port B & port C_{lower}.

Port A: It has one input latch & one output latch/buffer. Due to separate latches for input & output, port A can be used for bidirectional data transfer.

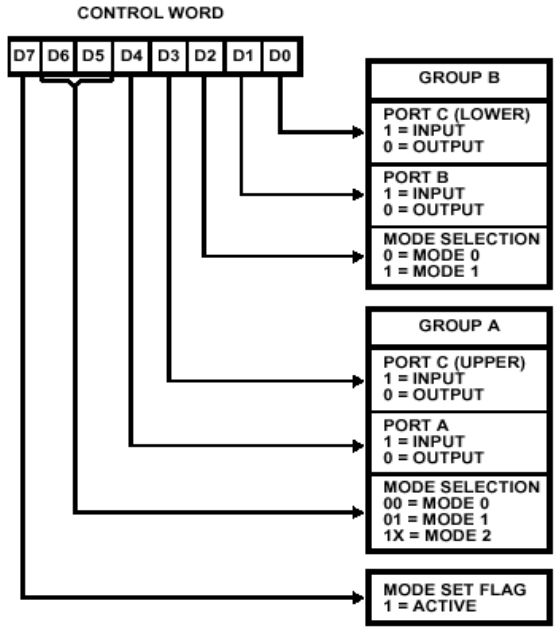
Port B: It has one 8-bit input/output latch/buffer & one 8-bit input buffer.

Port C: It has one 8-bit output latch/buffer & one 8-bit input buffer. The inputs are not latched.

OPRATING MODES OF 8255:

The 8255 has two basic modes.

1. **Bit set/reset (BSR) mode:** In this mode, the bits of port C are selectively set or reset. This mode does not have any effect on A & B ports.
2. **I/O Mode:** The I/O mode is used to perform data transfer on the ports of 8255. The I/O mode is further divided into three modes.



Mode 0: Simple I/O for all the three ports.

Mode 1: Strobbed (handshake) I/O for A& B ports. In this mode, the port C carries handshake signals for the data transfers taking place on A & B ports.

Mode 2: In this mode, port A can be used as bidirectional port. The port B can be configured either in mode 0 or 1. The port C bits are used for handshake signals.

The modes of 8255 are selected by writing appropriate control word into its control register. The format of the I/O mode control word is explained in figure.

The lower three bits D_0, D_1, D_2 define the modes of group B ports. The bits D_0 & D_1 configure port C_L & port B as input or output port depending on the values written into them. The ports operate in mode 0 when D_2 is 0 & in mode 1 when D_2 is 1. Note that mode is not available for group B ports.

Similarly, the bits D_3 through D_6 configure the mode of port A & port C_U. The D_3 & D_4 bits define the function of the ports & the D_5 & D_6 bits define the function of the ports & the D_5 & D_6 bits define the mode.

The D_7 bit should be strictly a 1 for the I/O mode. If D_7 is zero, the 8255 interprets the word as the BSR mode control word.

BSR mode: The BSR (Bit Set Reset mode) is only related to Port C. The bits of port C are set or reset according to the BSR control word written into the control register. The format of the BSR control word is as shown in fig below.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	B	B	B	S/R
For BSR mode	These bits are not used=000			These bits selects bits of port c			1=Set bit 0=reset bit

D3	D2	D1	Selected Bit
0	0	0	PC0
0	0	1	PC1
0	1	0	PC2
0	1	1	PC3
1	0	0	PC4
1	0	1	PC5
1	1	0	PC6
1	1	1	PC7

For BSR control word bit D_7 is zero. Number of the port C bit, which is to be set or reset is given by D_1, D_2, and D_3 bits. The selected bit is set if D_0 is 1 & reset if D_0 is 0.

Ex: Write a control word and program to set PC3 bit and after some time reset it.

Control Word;

D7	D6	D5	D4	D3	D2	D1	D0
0 For BSR Mode	0 not used	0 not used	0 not used	0	1	1	1 set bit

----------Bit PC3----------

---------- 0----------/----------7----------H.

Hence control world =07 H to Set PC3 bit And To Reset PC3 bit control world(CW) =06H

Prog.

Start: MVI A 07 Load acc. With 07 to set PC3 in BSR mode

OUT FF Send CW to control register.

CALL DELAY Wait for some time

MVI A 06 Load acc. With 06 to reset PC3 in BSR mode

OUT FF Send CW to control register.

End

Note that in BSR mode control word is given to control register no any instruction is given to port C.

*Instead of End instruction if instructions

CALL DELAY

JMP START

Are given a square wave is generated at PC3.

Ex:- Find contol word for 8255 to confugrate its all ports are as Out Put Ports in Mode0

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0
For I/O mode	Mode 0 for A	And C u	Port A as out put	Port c u as O/P port	Mode O for Port B and Cl	Port B as O/P	Port CLower as O/P

= 8 Hex of 1 0 0 0 of D7,D6,D5,D4	=0 Hex Of 0 0 0 0 Of D3,D2,D1,D0

Hence control world to confugrate all ports as output ports is 80H When it is transferred to control register by following program it will confugrates ports as output ports

MVI A 80H(Control Word)

OUT FF (FF is address of Control Register)

Ex :Switches are connected to PORT B and LEDS Are connected to PORT A find proper control word and write a program to show position of switches on LED.

In this example port B is used as input port to take information of switches and Port A is used as out put port to send information to LEDs for this mode 0 is used. hence control word is

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0

Hex =82h

Assume address for PORT A=FC,PORT B=FD,PORT C=FE,CONTROL REGISTER=FF.

Prog.

MVI A 82H Control word on accumulator

OUT FF Send control Word to Control Register

IN FD Take information from Port B to Accumulator

OUT FC Send information from accumulator to Port A.

End