



# **Chapter 5**

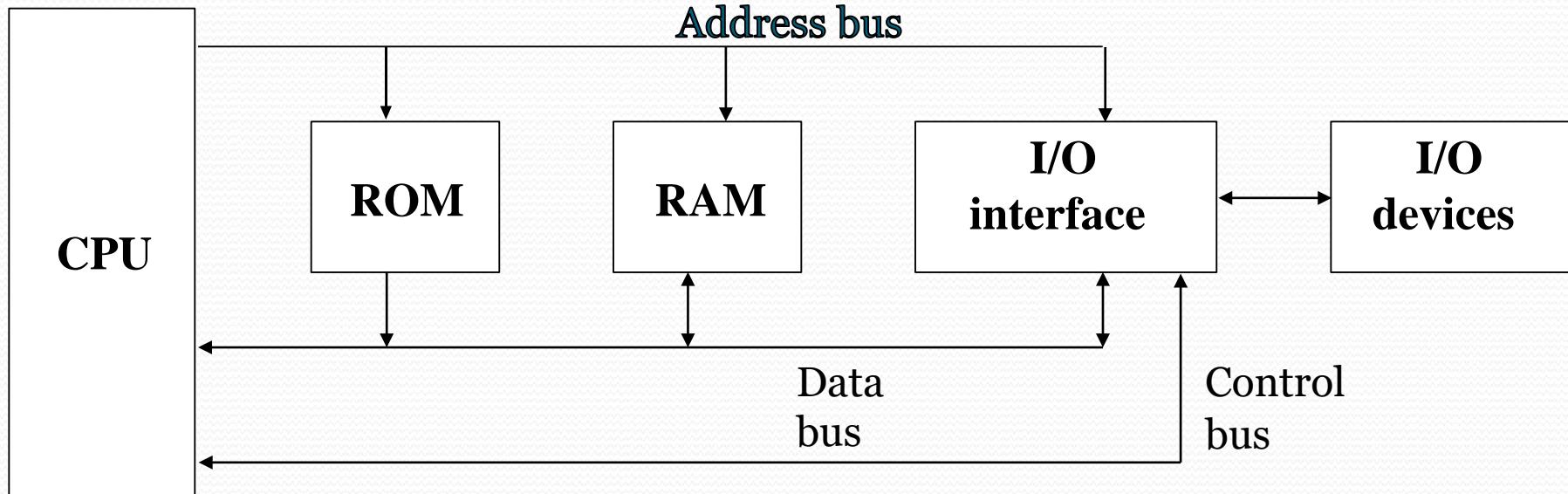
## **Introduction to Microprocessor**

**By**  
**Namrata Limbad**

# Introduction

**Computer:** A computer is a programmable machine that receives input, stores and manipulates data/information, and provides output in a useful format.

Basic computer system consist of a CPU, memory and I/O unit.



**Block diagram of a basic computer system**

# Basic Concepts of Microprocessors

- **Microcomputer:-** It is a programmable machine. The two principal

characteristics of a computer are:

- Responds to a specific set of instructions in a well-defined manner.
- It can execute a prerecorded list of instructions (a program)
- Its main components are
  - CPU
  - Input & Output devices
  - Memory
- **Microprocessor:-** It is a programmable VLSI chip which includes ALU, register circuits & control circuits. Its main units are-
  - ALU
  - Registers
  - Control Unit
- **Microcontroller:-** Silicon chip which includes microprocessor, memory & I/O in a single package

# Microprocessor vs. Microcontroller

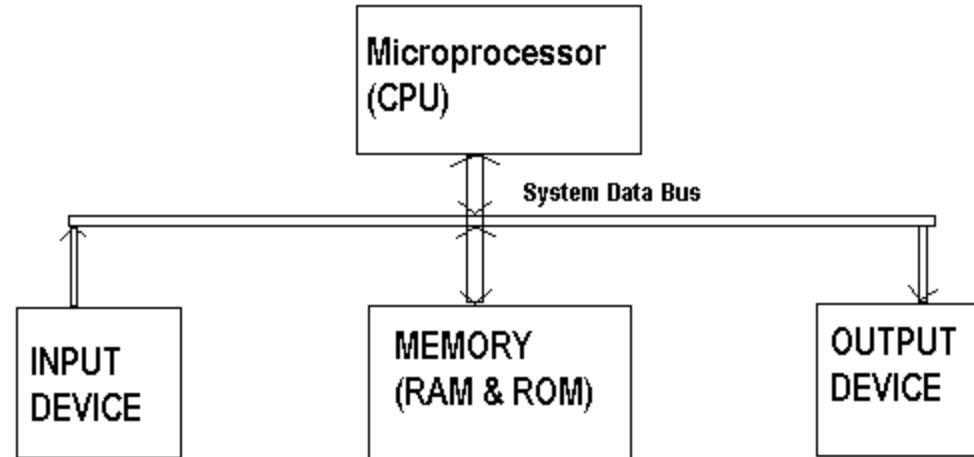
## Microprocessor

- CPU is stand-alone,  
*RAM, ROM, I/O, timer  
are separate*
- designer can decide on the  
*amount of ROM, RAM and  
I/O ports.*
- expansive versatility  
general-purpose*

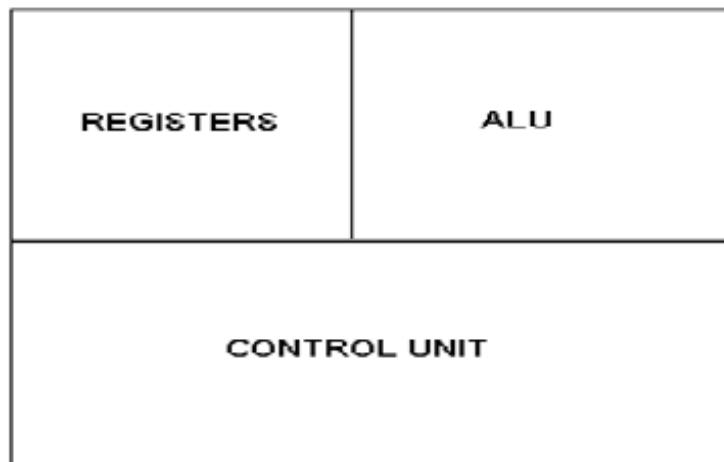
## Microcontroller

- CPU, RAM, ROM, I/O and timer are all on a single chip
  - fix amount of on-chip ROM, RAM, I/O ports
- for applications in which cost, power and space are critical
- single-purpose

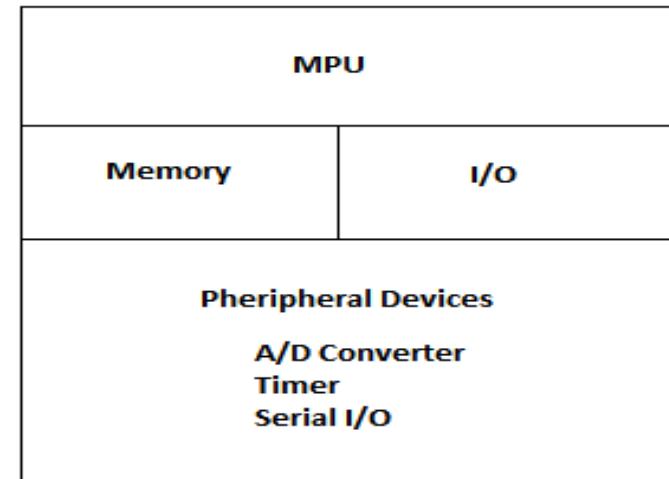
# Block Diagram



Microcomputer



Microprocessor

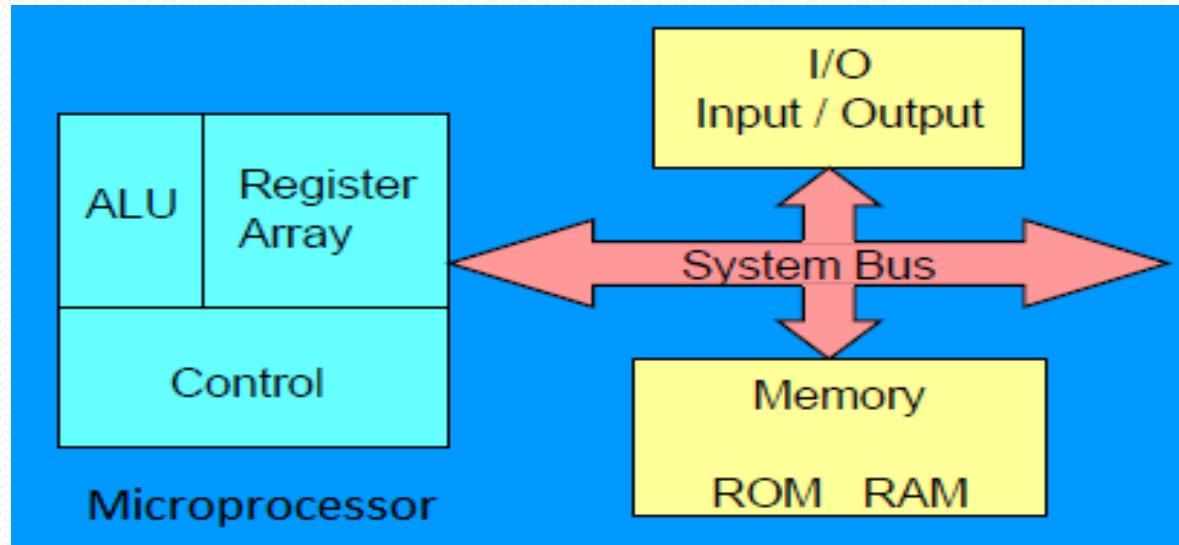


Microcontroller

# **Definition of the Microprocessor**

- ✓ Microprocessor is a **Programmable, Clock driven, Register based, Electronic device** that reads instruction from a storage device, takes the data from input unit and process the data according to the instructions and provides the result to the output unit.
  - **Programmable**- Perform Different set operation on the data depending on the sequence of instructions supplied by the programmer.
  - **Clock Driven** – Whole task is divided into basic operations, are divided into precise system clock periods.
  - **Register Based** – Storage element
  - **Electronic Device** – fabricated on a chip

# Microprocessor Based System with bus Architecture



- **ALU:**- Arithmetic and logical operations like add, subtraction, AND & OR.
- **Register Array:** - Store data during the execution of program.
- **Control Unit:** Provides necessary timing & control signal. It controls the flow of data between microprocessor and peripherals.

\*Microprocessor is one component of microcomputer.

- **Memory:**

- Stores information such as instructions and data in binary format (0 and 1).
- “Sub-system” of microprocessor-based system. sub-system includes the registers

inside the microprocessor .

- ✓ **Read Only Memory (ROM):** used to store programs that do not need alterations.
- ✓ **Random Access Memory (RAM) (R/WM):** used to store programs that can read and altered like programs and data.

- **Input/output:** Communicates with the outside world.

- **System Bus:** Communication path between the microprocessor and peripherals.

- group of wires to carry bits.

# How does a Microprocessor works

- ✓ To execute a program, the microprocessor “reads” each instruction from memory, “interprets” it, then “executes or performs” it.
- ✓ The right name for the cycle is
  - Fetch
  - Decode
  - Execute
- ✓ This sequence is continued until all instructions are performed.

# Machine Language

- ✓ To communicate with computer, instruction is given in binary language.
- ✓ MP has 8 bit data so  $2^8 = 256$  combinations. So difficult to write programs in set of 0's and 1's .

for eg:

- 0011 1100 – is an instruction that increments the number in accumulator by 1
- 1000 0000 – is an instruction that add the number in register B to the accumulator content , and keep the result in A.

So it is very tedious and error inductive. For convenience, written in Hexadecimal code. For example 0011 1100 is equivalent to 3C H

Decimal	b3	b2	B1	B0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

# 8085 Assembly Language

- ✓ Even program is written in Hexa decimal .. It is difficult to understand.
- ✓ Program is written in mnemonic.

For E.g.: binary code 0011 1100 (3C H in hexadecimal) is represented by

INR A

- INR A –INR stands for Increment, A is accumulator... this symbol suggest the operation of incrementing the accumulator by 1

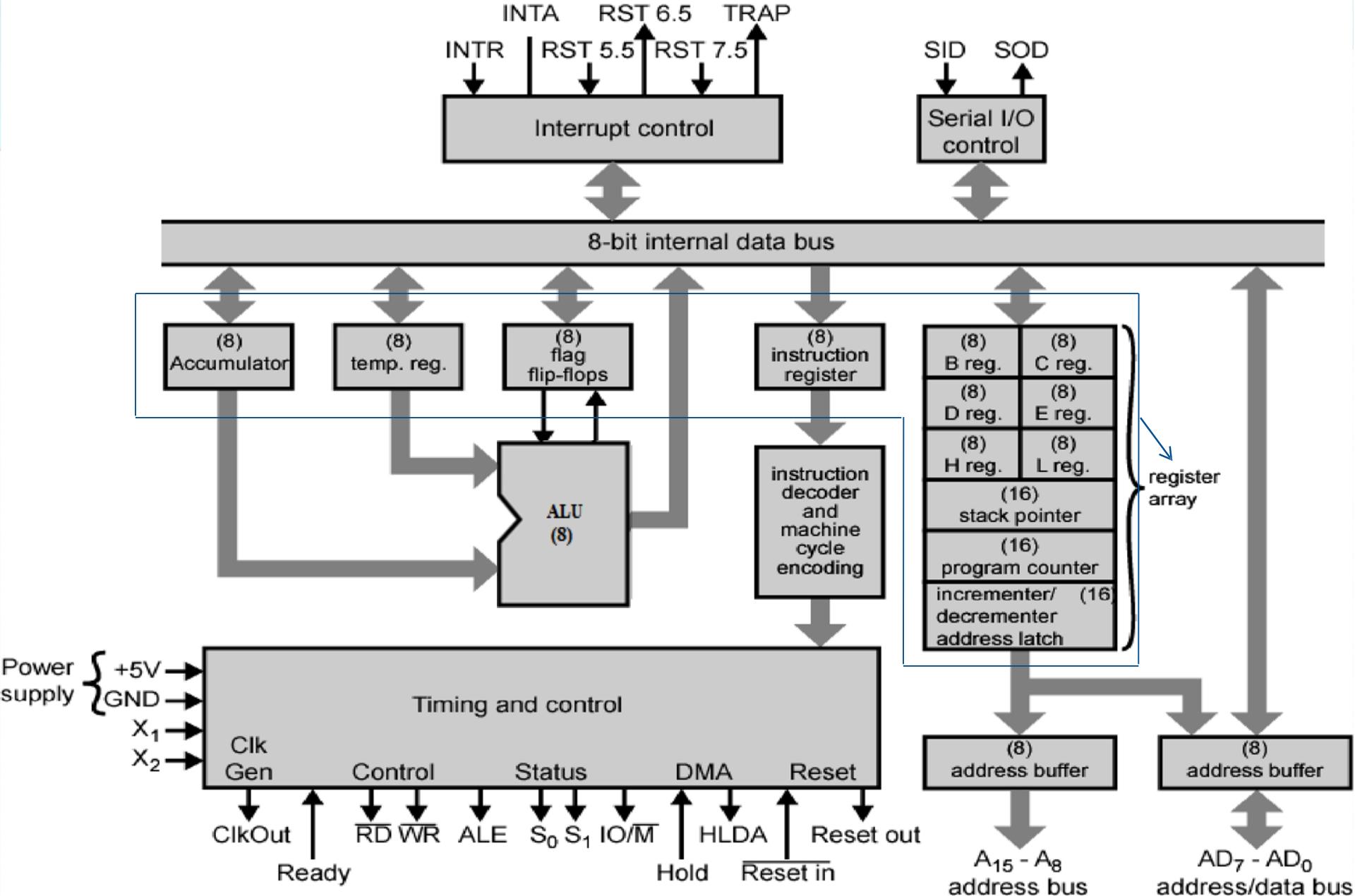
Similarly 1000 0000 is equivalent ( 80 H) is represented as

- ADD B– ADD stands for addition and B represents content in register B. this symbol suggest that addition of the number in register B to the accumulator content , and keep the result in A.

\*\*\*So MP has 246 such bit pattern amounting to 74 different instruction for performing various operations

# Features of Microprocessor- 8085

- ✓ 8085 is developed by INTEL
- ✓ 8 bit microprocessor: can accept 8 bit data simultaneously
- ✓ Operates on single +5V D.C. supply.
- ✓ Designed using NMOS technology
- ✓ 6200 transistor on single chip
- ✓ It provides on chip clock generator, hence it does not require external clock generator.
- ✓ Operates on 3MHz clock frequency.
- ✓ 8-bit multiplexed address/data bus, which reduce the number of pins.
- ✓ 16-bit address lines, hence it can address  $2^{16} = 64$  K bytes of memory
- ✓ It generates 8 bit I/O addresses, hence it can access  $2^8 = 256$  I/O ports.
- ✓ 5 hardware interrupts i.e. TRAP, RST6.5, RST5.5, RST4.5, and INTR
- ✓ It provides DMA.



Internal Architecture (functional block diagram) of 8085

# 8085 Architecture.....cont...

8085 architecture consists of following blocks:

- 1.Register Array
- 2.ALU & Logical Group
- 3.Instruction decoder and machine cycle encoder
4. Timing and control circuitry
- 5.Interrupt control Group
- 6.Serial I/O control Group

## 1. Registers Array : 14 register out of which 12 are 8 bit capacity and 2 of 16 bit. Classify into 4 types

### (a) General purpose register: (user accessible)

- B,C,D,E,H,L are 8 bit register.(can be used singly)

Can also be used for 16-bit register pairs- BC, DE & HL. Used to

- store the intermediate data and result
- H & L can be used as a data pointer(holds memory address)

### (b) Special Purpose Register[A, Instruction Register and Flag]

#### (b.1) Accumulator (A): (user accessible)

- 8 bit register

All the ALU operations are performed with reference to the contents of Accumulator.

- Result of an operation is stored in A.

➤ Store 8 bit data during I/O transfer

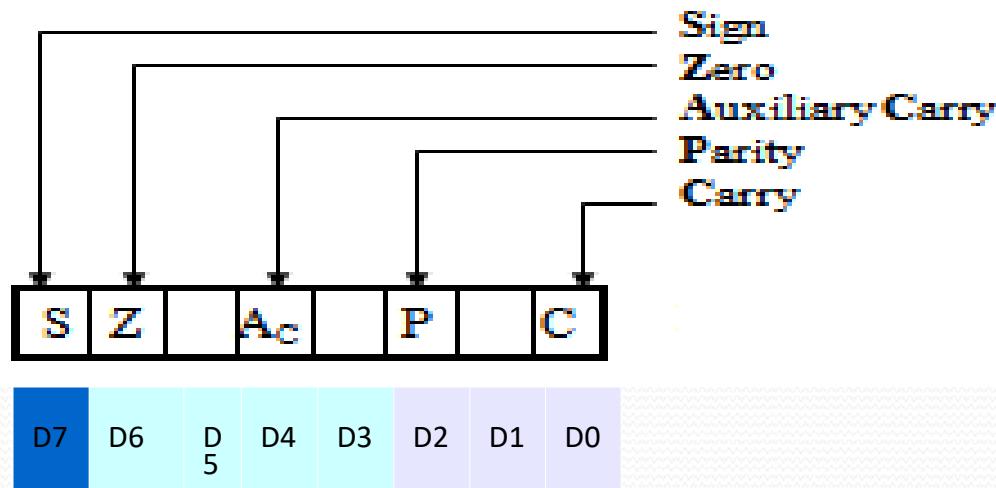
#### (b.2) Instruction Register: (user not accessible)

- When an instruction is fetched from memory, it is loaded in IR. Then transferred to the decoder for decoding.

It is not programmable and can not be accessed through any instruction.

### (b.3) Flag Register(F): (user accessible)

- 8 bit Register
- Indicates the status of the ALU operation.
- ALU includes 5 flip flop, which are set or reset after an operation according to data conditions of the result in the accumulator.



(Flag Register)

Flag Register/Status

1	0		0		0		1
S	Z	-	AC	-	P	-	CY

Content of B register is A5H and A = B7 H

Perform ADD B..What will be the contents of A after execution and Draw Flag register

ADD B means  $[A]+[B] = [A]$

$$B7 + A5 = \dots\dots\dots$$

1011 0111

+1010 0101

---

1 1101 1100 DC H

<b>Decimal (Base 10)</b>	<b>Binary (Base 2)</b>	<b>Hexadecimal (Base 16)</b>
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Flag	Significance
<b>C or CY (Carry)</b>	CY is set when an arithmetic operation generates a carry out, otherwise it is 0 (reset)
<b>P (Parity)</b>	$P = 1$ ; if the result of an ALU operation has an even number of 1's in A; $P = 0$ ; if number of 1 is odd.
<b>AC (Auxiliary carry)</b>	Similar to CY, $AC = 1$ if there is a carry from D <sub>3</sub> to D <sub>4</sub> Bit AC= 0 if there is a no carry from D <sub>3</sub> to D <sub>4</sub> Bit (not available for user) $  \begin{array}{r}  \textcolor{red}{1} \\  1111 \textcolor{red}{0}111 \\  +0110 \textcolor{red}{1}111 \\  \hline  \textcolor{red}{1}0001000  \end{array}  $
<b>Z(zero)</b>	Z = 1; if result in A is 00H 0 otherwise
<b>S(Sign)</b>	S=1 if D <sub>7</sub> bit of the A is 1(indicate the result is -ive) S= 0 if D <sub>7</sub> bit of the A is 0(indicate the result is +ive)

(c) **Temporary Register[ W, Z, Temporary data register]**

- Internally used by the MP(user not accessible)

**(c.1) W and Z register:**

- 8 bit capacity

Used to hold temporary addresses during the execution of some instructions

**(c.2) Temporary data register:**

- 8 bit capacity

Used to hold temporary data during ALU operations.

## (d) Pointer Register or special purpose [SP, PC]

### (d.1) Stack Pointer(SP)

- 16 bit address which holds the address of the data present at the top of the stack memory
- It is a reserved area of the memory in the RAM to store and retrieve the temporary information.
- Also hold the content of PC when subroutines are used.

When there is a subroutine call or on an interrupt. ie. pushing the return address on a jump, and retrieving it after the operation is complete to come back to its original location.

### (d.3) Program Counter(PC)

- 16 bit address used for the execution of program

Contain the address of the next instruction to be executed after fetching the instruction it is automatically incremented by 1.

Not much use in programming, but as an indicator to user only.

**In addition to register MP contains some latches and buffer**

✓ **Increment and decrement address latch**

- 16 bit register

Used to increment or decrement the content of PC and SP

✓ **Address buffer**

- 8 bit unidirectional buffer

Used to drive high order address bus(A8 to A15)

When it is not used under such as reset, hold and halt etc this buffer is used tristate high order address bus.

✓ **Data/Address buffer**

- 8 bit bi-Directional buffer

Used to drive the low order address (A0 to A7) and data (D0 to D7) bus.

Under certain conditions such as reset, hold and halt etc this buffer is used tristate low order address bus.

**(2) ALU & Logical Group:** it consists ALU, Accumulator, Temporary register and Flag Register.

**(a) ALU**

- Performs arithmetic and logical operations
- Stores result of arithmetic and logical operations in accumulator

**(b) Accumulator**

- General purpose register
- Stores one of the operand before any arithmetic and logical operations and result of operation is again stored back in Accumulator
- Store 8 bit data during I/O transfer

(2) **ALU & Logical Group.....cont.....**

(c) **Temporary Register**

- 8 bit register
- During the arithmetic and logical operations one operand is available in A and other operand is always transferred to temporary register

For Eg.: ADD B – content of B is transferred into temporary register before actual addition

(d) **Flag Register**

- Five flag is connected to ALU
- After the ALU operation is performed the status of result will be stored in five flags.

(3) **Instruction decoder and machine cycle encoder, Timing and control circuitry**

(a) **Instruction decoder and machine cycle encoder :**

- Decodes the op-code stored in the Instruction Register (IR) and establishes the sequence of events to follow.
- Encodes it and transfer to the timing & control unit to perform the execution of the instruction.

(b) **Timing and control circuitry**

- works as the brain of the CPU
- For proper sequence and synchronization of all the operations of MP,

this unit generates all the timing and control signals necessary for communication between microprocessor and peripherals.

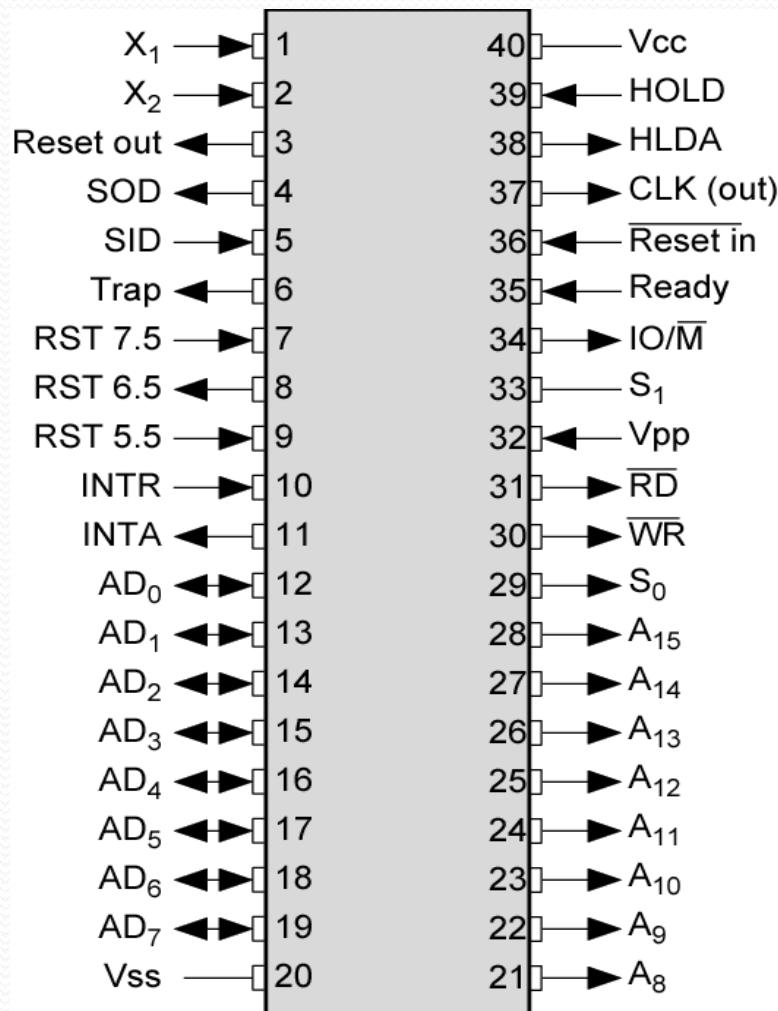
## (4) **Interrupt Control group**

- **Interrupt:-** Occurrence of an external disturbance
- After servicing the interrupt, 8085 resumes its normal working sequence
- Transfer the control to special routines
- Five interrupts: - TRAP, RST7.5, RST6.5, RST5.5, INTR
- In response to INTR, it generates INTA signal

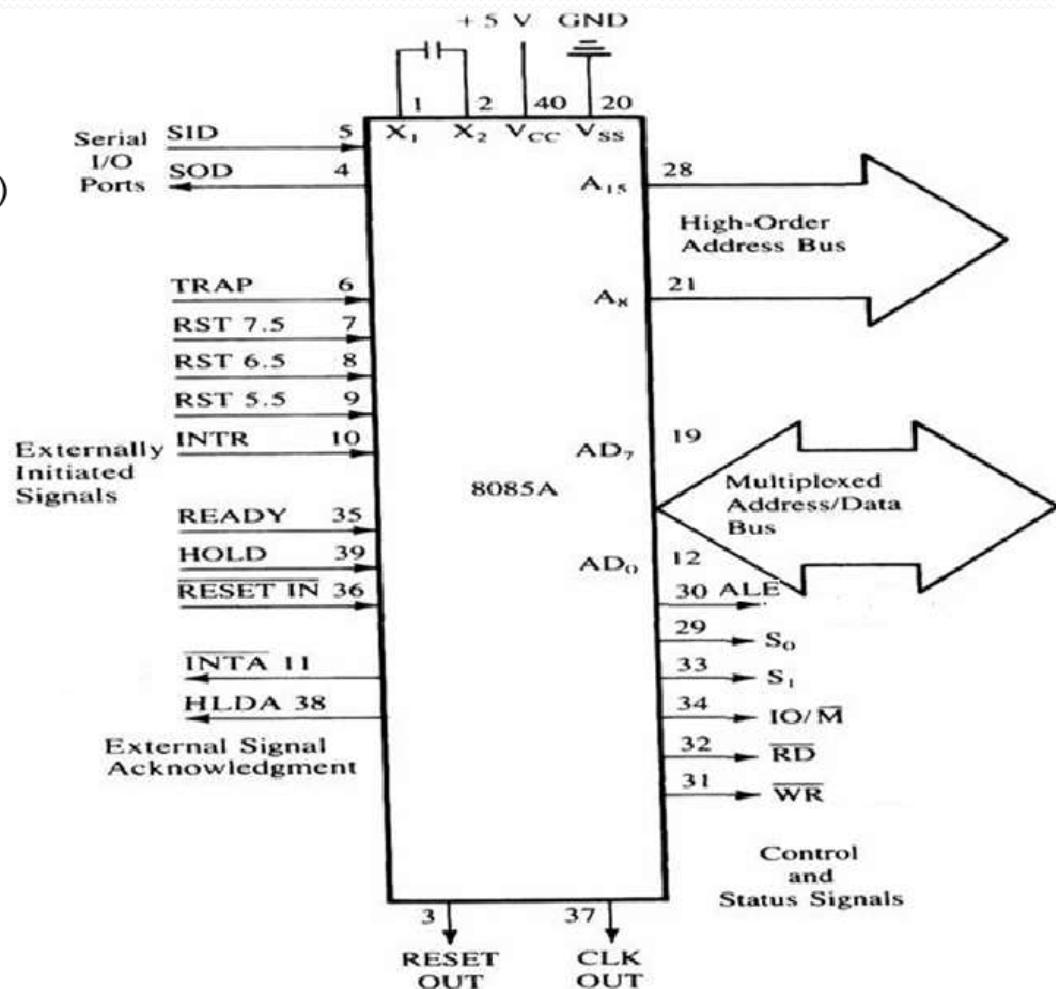
## (4) **Serial I/O control Group**

- Data transfer red on Do- D7 lines is parallel data
- But under some condition it is used serial data transfer
- Serial data is entered through SID(serial input data) input (received)
- Serial data is outputted on SOD(serial output data) input (send)

# 8085 Pin Diagram



Pin Configuration



Functional Pin diagram

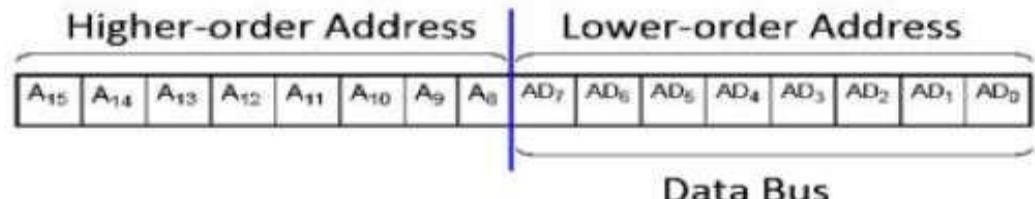
# 8085 Pin Description

- The 8085 is an 8-bit general purpose microprocessor that can address 64K Byte of memory.
- It has 40 pins and uses +5V for power. It can run at a maximum frequency of 3 MHz.
- The pins on the chip can be grouped into 6 groups:
  - Address Bus and Data Bus.
  - Status Signals.
  - Control signal
  - Interrupt signal
  - Power supply and Clock signal
  - Reset Signal
  - DMA request Signal
  - Serial I/O signal
  - Externally Initiated Signals.

# The Address and Data Busses

## □ Address Bus (Pin 21-28)

- 16 bit address lines A<sub>0</sub> to A<sub>15</sub>
- The address bus has 8 signal lines A<sub>8</sub> – A<sub>15</sub> which are unidirectional.
- The other 8 address lines A<sub>0</sub> to A<sub>7</sub> are multiplexed (time shared) with the 8 data bits.



## □ Data Bus (Pin 19-12)

- To save the number of pins lower order address pin are multiplexed with 8 bit data bus (bidirectional)

- So, the bits AD<sub>0</sub> – AD<sub>7</sub> are bi-directional and serve as A<sub>0</sub> – A<sub>7</sub> and D<sub>0</sub> – D<sub>7</sub> at the same time.
- During the execution of the instruction, these lines carry the address bits during the early part (T<sub>1</sub> state), then during the late parts(T<sub>2</sub> state) of the execution, they carry the 8 data bits.

# Status Signals

## Status Pins – ALE, S1, S0

### 1.ALE(Address Latch Enable): (Pin 30)

- Used to demultiplexed the address and data bus
- +ive going pulse generated when a new operation is started by uP.
  - ALE = 1 when the AD0 – AD7 lines have an address
  - ALE = 0 When it is low it indicates that the contents are data.
  - This signal can be used to enable a latch to save the address bits from the AD lines.

### 1.S1 and S0 (Status Signal): (Pin 33 and 29)

- Status signals to specify the kind of operation
- Usually un-used in small systems.

S1	S0	Operation
0	0	HALT
0	1	WRITE
1	0	READ

# Control Signals

## Control Pins – RD, WR, IO/M(active low)

### 1.RD: Read(Active low) (Pin 32)

- Read Memory or I/O device
- Indicated that data is to be read either from memory or I/P device and data bus is ready for accepting data from the memory or I/O device.

### 2.WR: Write(Active low) (Pin 31)

- Write Memory or I/O device
- Indicated that data on the data bus are to be written into selected memory or I/P device.

### 3.IO/M: (Input Output/Memory-Active low) (Pin 34)

- Signal specifies that the read/write operation relates to whether memory or I/O device.
- When (IO/M=1) the address on the address bus is for I/O device
- When (IO/M=0) the address on the address bus is for memory

IO/M(active low)	RD	WR	Control Signal	Operation
0	0	1	MEMR	M/M Read
0	1	0	MEMW	M/M write

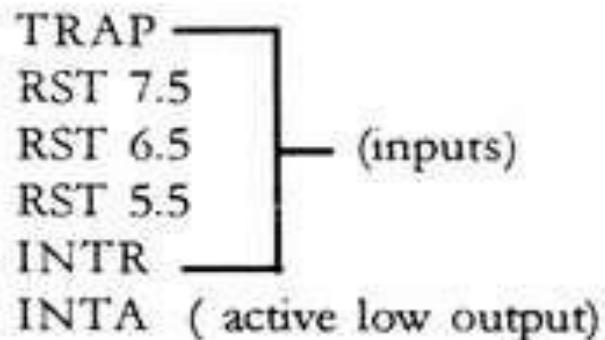
# Control and status Signals

When S0, S1 is combined with IO/M(active low), we get status of machine cycle

IO/M	S1	S0	OPERATION	Control Signal
0	1	1	Opcode fetch	$\overline{RD} = 0$
0	1	0	Memory read	$\overline{RD} = 0$
0	0	1	Memory write	$\overline{WR} = 0$
1	1	0	I/O read	$RD = 0$
1	0	1	I/O write	$\overline{WR} = 0$
1	1	0	Interrupt Acknowledge	$INTA = 0$
Z	0	0	Halt	$RD = WR = Z$

# Interrupts

- They are the signals initiated by an external device to request the microprocessor to do a particular task or work.
- There are five hardware interrupts called, (Pin 6-11)

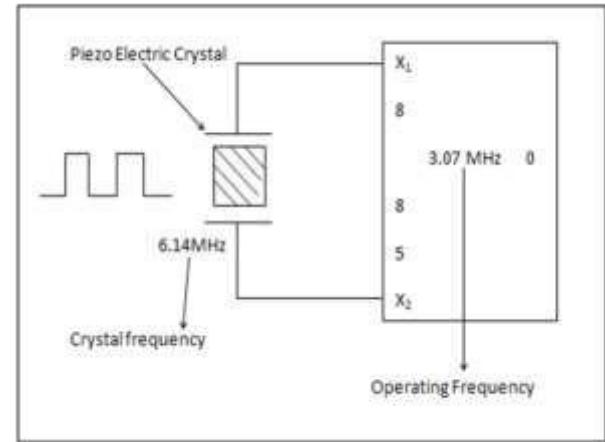


- On receipt of an interrupt, the microprocessor acknowledges the interrupt by the active low INTA (Interrupt Acknowledge) signal.

# Power supply and Clock Signal

Vcc (Pin 40) : single +5 volt power supply  
Ground

There are 3 important pins in this group.



## ✓ X0 and X1 :((Pin 1-2)

- Crystal or R/C network connections to set the frequency of internal clock generator.
- The frequency is internally divided by two.
- Since the basic operating timing frequency is 3 MHz, a 6 MHz crystal is connected to the X<sub>0</sub> and X<sub>1</sub> pins.

## ✓ CLK (output): (Pin 37)

- Clock Output is used as the system clock for peripheral and

# Reset Signals

## ✓ Reset In (input, active low) (Pin 36)

- This signal is used to reset the microprocessor.
- The program counter inside the microprocessor is set to zero(0000H)
- The buses are tri-stated.

## ✓ Reset Out (Output, Active High) (Pin 3)

- It indicates MP is being reset.
- Used to reset all the connected devices when the microprocessor is reset.

# DMA Request Signals

## ❑ DMA:

- When 2 or more devices are connected to a common bus, to prevent the devices from interfering with each other, the tristate gates are used to disconnect all devices except the one that is communicating at a given instant .
- The CPU controls the data transfer operation between memory and I/O device.
- DMA operation is used for large volume data transfer between memory and an I/O device directly.
- The CPU is disabled by tri-stating its buses and the transfer is effected directly by external control circuits.

### ✓ HOLD (Pin 38)

- This signal indicates that another device is requesting the use of address and data bus.
- So it relinquish the use of buses as soon as the current machine cycle is completed.
- MP regains the bus after the removal of a HOLD signal

### ✓ HLDA (Pin 39)

- On receipt of HOLD signal, the MP acknowledges the request by sending out HLDA signal and leaves out the control of the buses.
- After the HLDA signal the DMA controller starts the direct transfer of data.
- After the removal of HOLD request HLDA goes low.

# Serial I/O Signals

These pins are used for serial data communication

## ✓ **SID (input) Serial input data (Pin 4)**

- It is a data line for serial input
- Used to accept serial data bit by bit from external device
- The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.

## ✓ **SOD (output) Serial output data (Pin 5)**

- It is a data line for serial output
- Used to transmit serial data bit by bit to the external device
- The 7th bit of the accumulator is outputted on SOD line when SIM

instruction is executed.

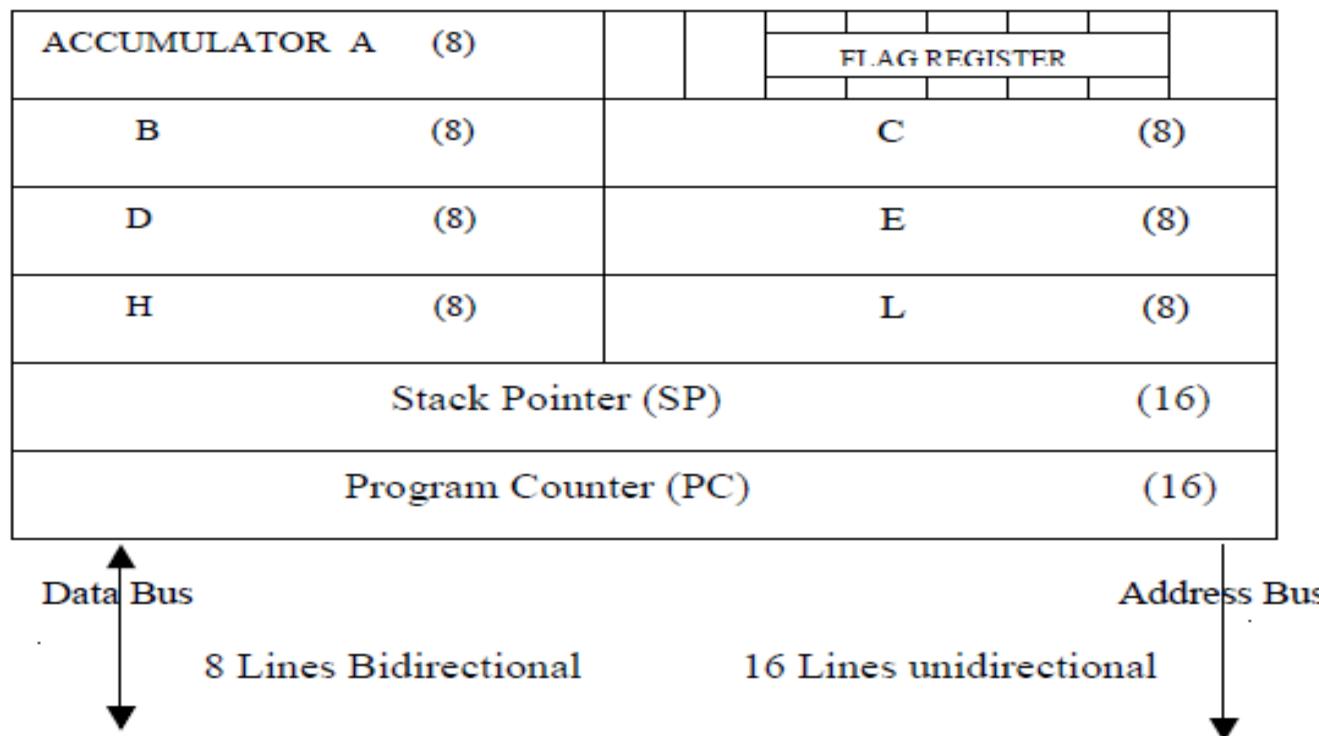
# Externally Initiated signal

## ✓ Ready (input) (Pin 35)

- Memory and I/O devices will have slower response compared to microprocessors.
- Before completing the present job such a slow peripheral may not be able to handle further data or control signal from CPU.
- The processor sets the READY signal after completing the present job to access the data.
- It synchronize slower peripheral to the processor.
- The microprocessor enters into WAIT state while the READY pin is disabled.

# 8085 Programming register and programming model

- The register which are programmable and available for the use are six general purpose register, A, F, PC, SP.



8085 programming  
model

# Fetching & Exécution Cycles

## □ Fetching Cycles

- The fetch cycle takes the instruction required from memory,

stores it in the instruction register, and

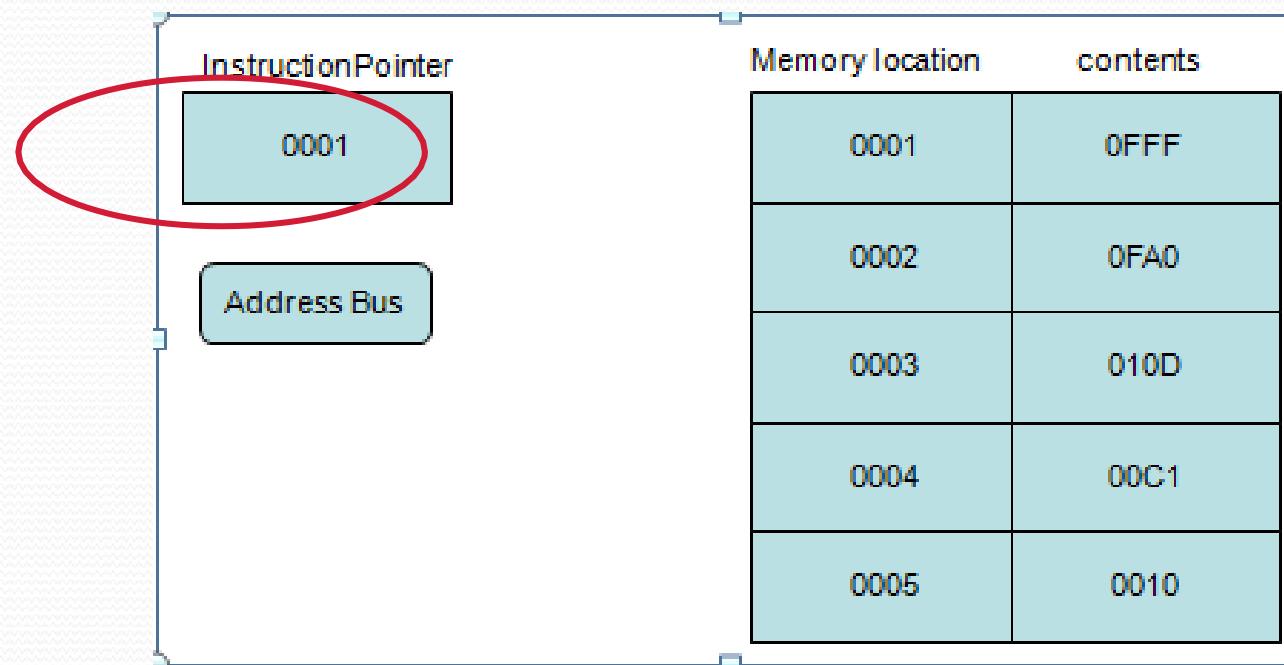
- Moves the program counter on one so that it points to the next instruction.

## □ Execute cycle

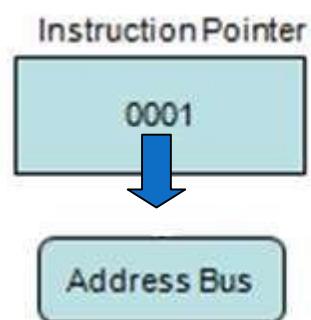
- The actual actions which occur during the execute cycle of an instruction.
- Depend on both the instruction itself and the addressing mode specified to be used to access the data that may be required.

## Fetching an instruction

**Step 1:** Instruction pointer (program counter) hold the address of the next instruction to be fetch.



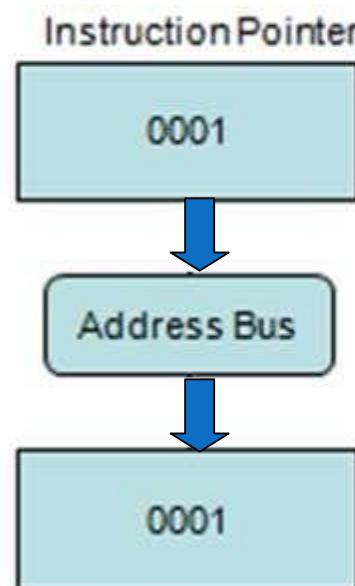
# Fetching an instruction....Cont....



Contents of the  
Program  
Counter are  
passed across  
the Address Bus

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

# Fetching an instruction....Cont....



The address moves over the address bus to the Memory Access Register

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

Memory Access Register

# Fetching an instruction....Cont....

Step 4

The memory location of the next instruction is located.

0001

Memory Access Register

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

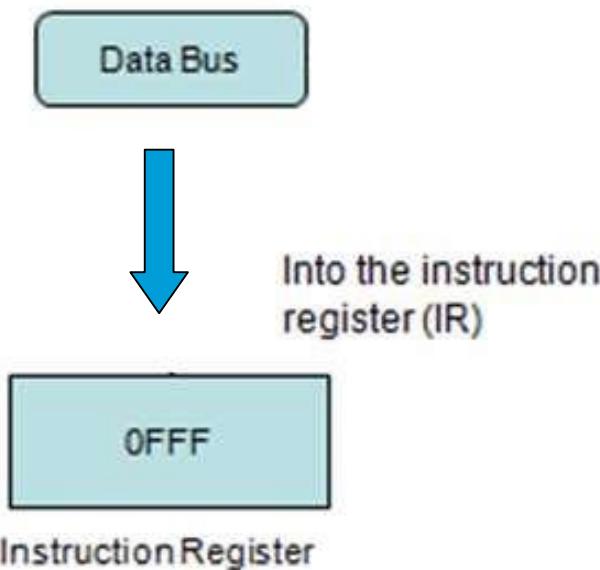
# Fetching an instruction...Cont....

## Data Bus

The contents of memory at the given location are moved across the data bus

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

# Fetching an instruction...Cont....

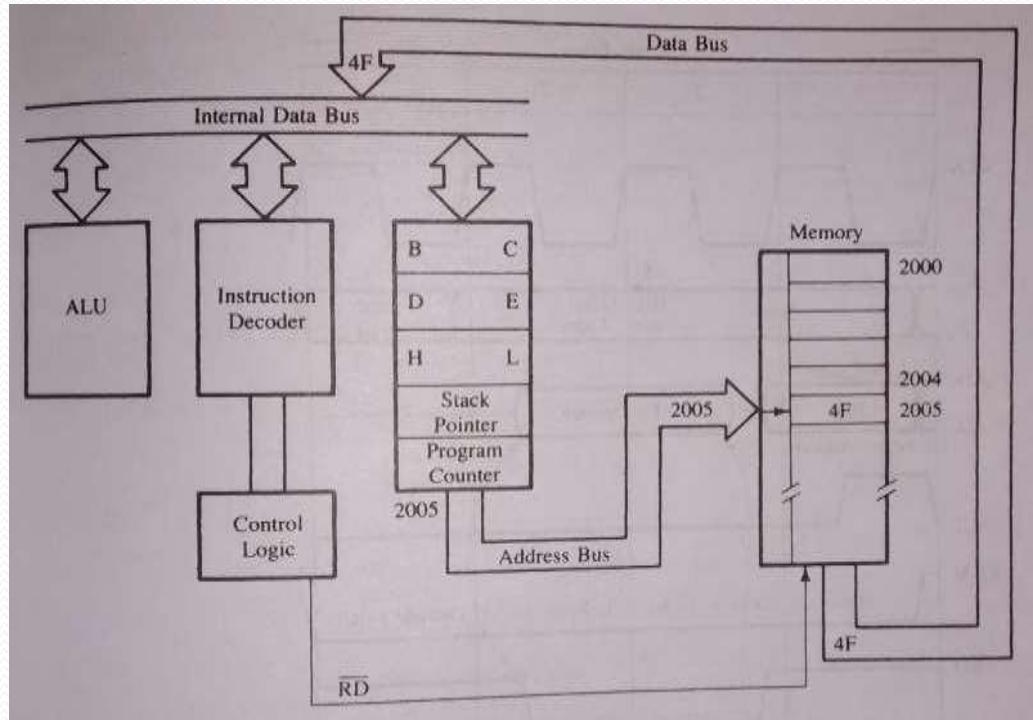


Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

# Data flow from memory to MPU

Steps and data flow, when the instruction code 01001111 (4FH)

—  
V  
  
d  
MPU needs data identify the location 2005H, is the fetch. and memory  
MO  
C,  
A)  
store  
in the  
flow from



Fetch Cycle: Step 1: MPU places the 16 bit memory address from PC on the address bus

Step 2: Control unit send the signal RD to enable memory chip

Step 3: The byte from the memory location is placed on the data bus.

Step 4: The byte is placed on the instruction decoder of the MPU and task is carried out according to the instruction.

## Timing: Transfer of byte from memory to MPU

□ How a data byte is transfer from memory to the MPU.

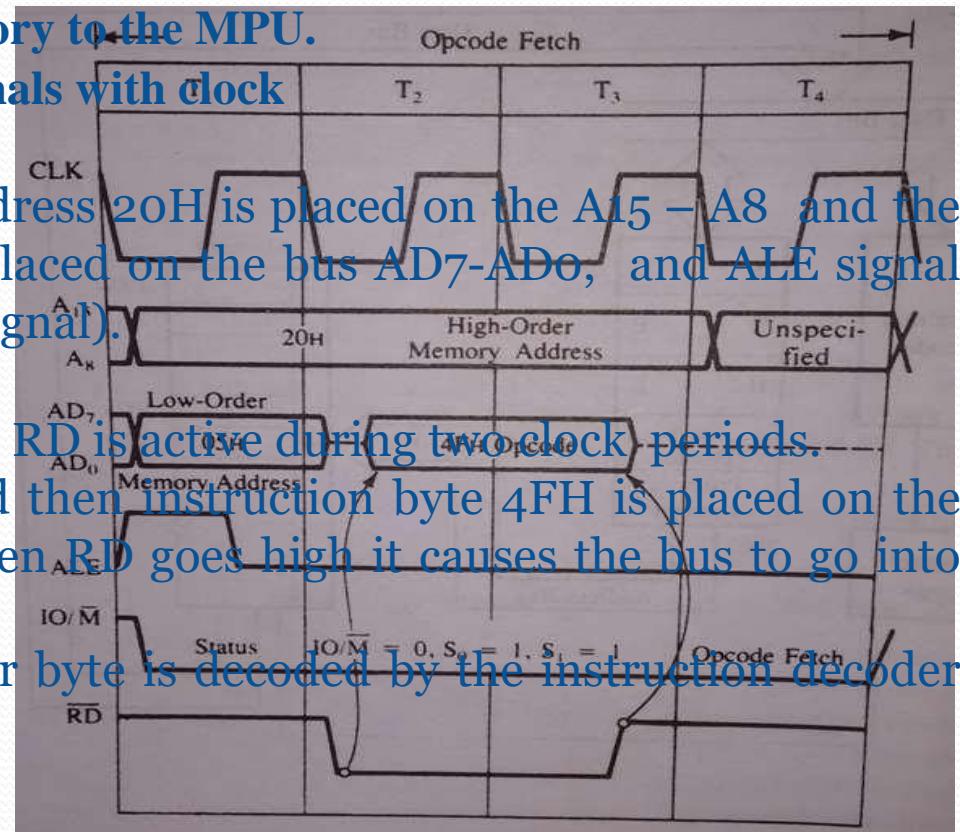
□ It shows the five different group of signals with clock

Step 1: At T1 higher order memory address 20H is placed on the A15 – A8 and the lower order memory address 05H is placed on the bus AD7-AD0, and ALE signal high. IO/M goes low(memory related signal).

Step 2: During T2 RD signal is sent out. RD is active during two clock periods.

Step 3 : During T3, Memory is enabled then instruction byte 4FH is placed on the data bus and transferred to MPU. When RD goes high it causes the bus to go into high impedance state.

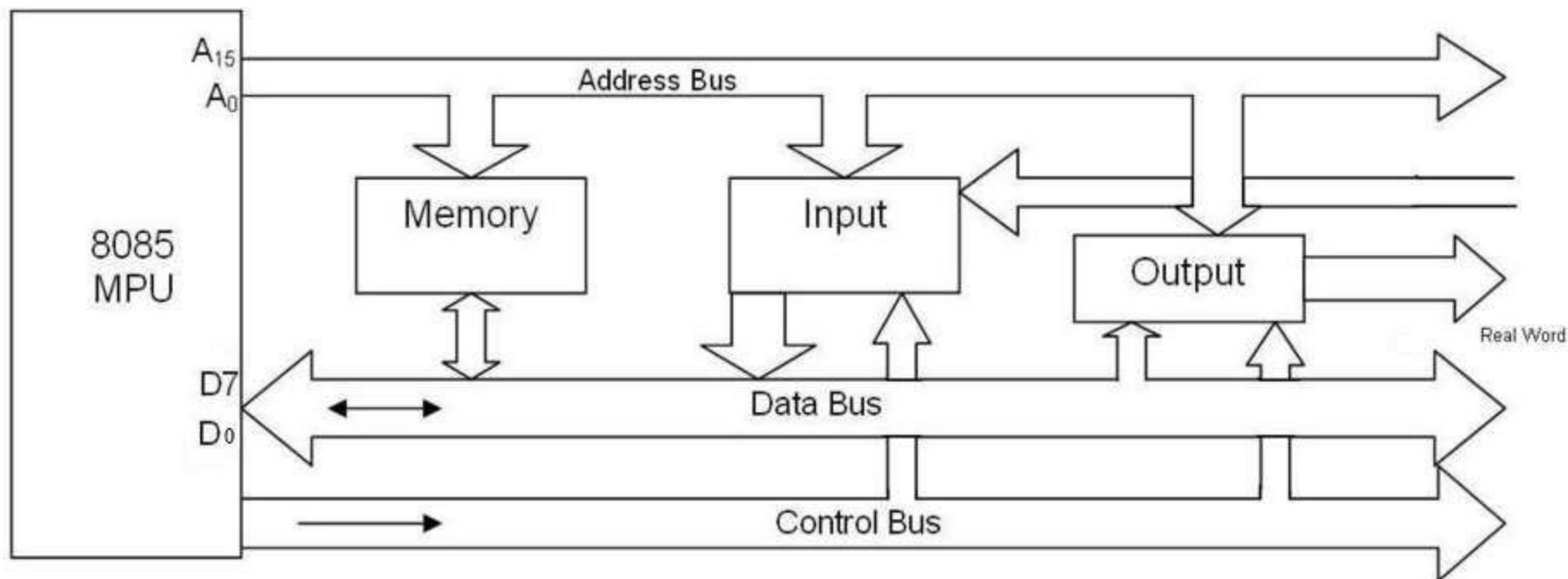
Step 4: During T4, the machine code or byte is decoded by the instruction decoder and content of A is copied into register.



# Buses Structure

- Various I/O devices and memories are connected to CPU by a group of lines

called as bus



8085 Bus  
structure

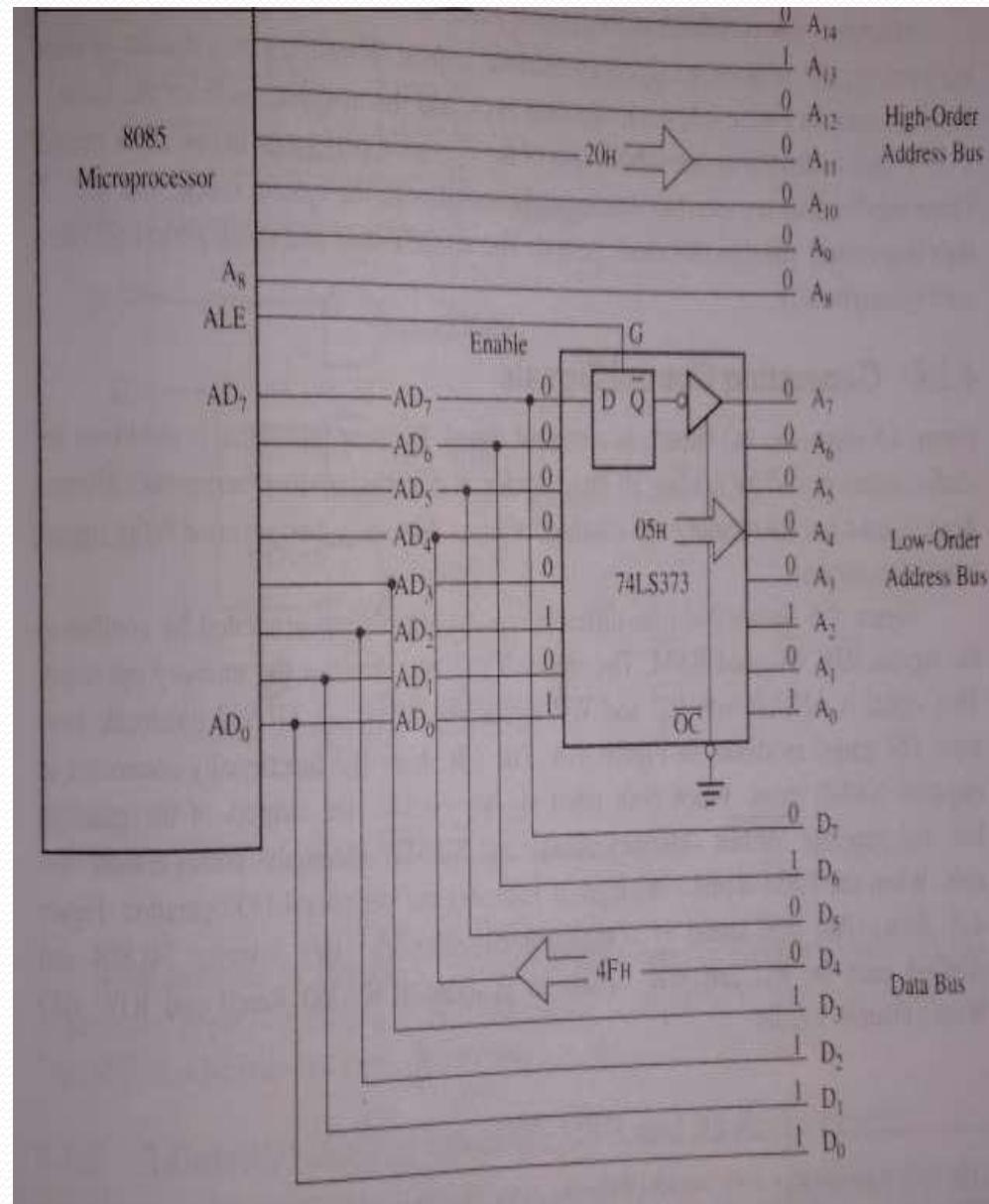
# De-multiplexing AD7-AD0

- AD7– ADo lines are serving a **dual purpose** and that they need to be demultiplexed to get all the information.
- The high order bits(20 H) of the address remain on the bus for **three clock periods**. However, the low order bits (05H) remain for only **one clock period** and they would be lost if they are not saved externally. The low order bits of the address disappear when they are needed most.
- To make sure we have the entire address for the full three clock cycles, we will use an external latch to save the value of AD7– ADo when it is carrying the address bits. We use the **ALE signal** to enable this latch. ALE signal is connected to the enable (G) pin of the latch.

Given that ALE operates as a pulse during T1, ALE high the latch is transparent; output changes according to input. So during T1 output of latch is 05H.

When ALE goes low, the data byte 05H is latched

until the next the ALE, output the latch represent the low address bus A7- r latching Ao after operation.



# Generating Control Signals

Signals are used both for memory and I/O related operations. So four different control signals are generated by combining

the signals RD, WR and IO/M. **MEMR** = Reading from memory **MEMW** = writing into memory **IOR** = Reading from input port **IOW** = writing to an output port

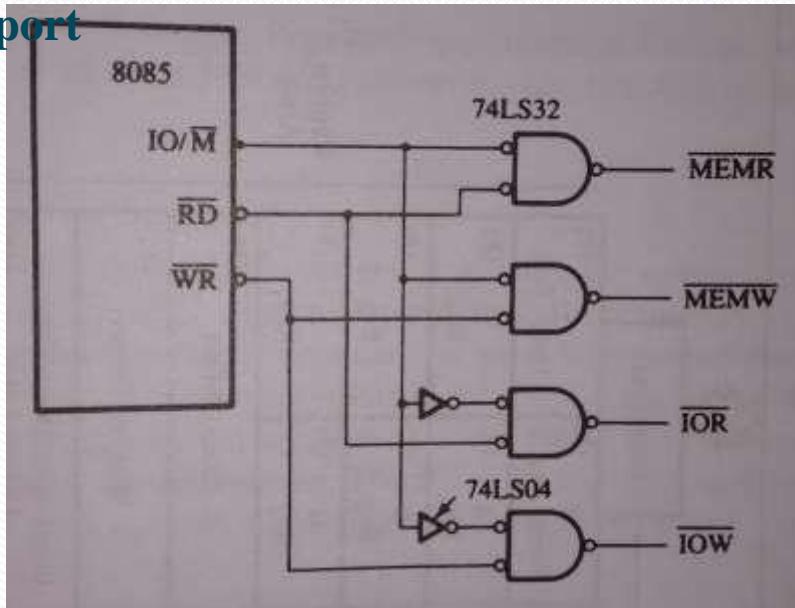


Fig: Generate Read/write control signal for memory and I/O

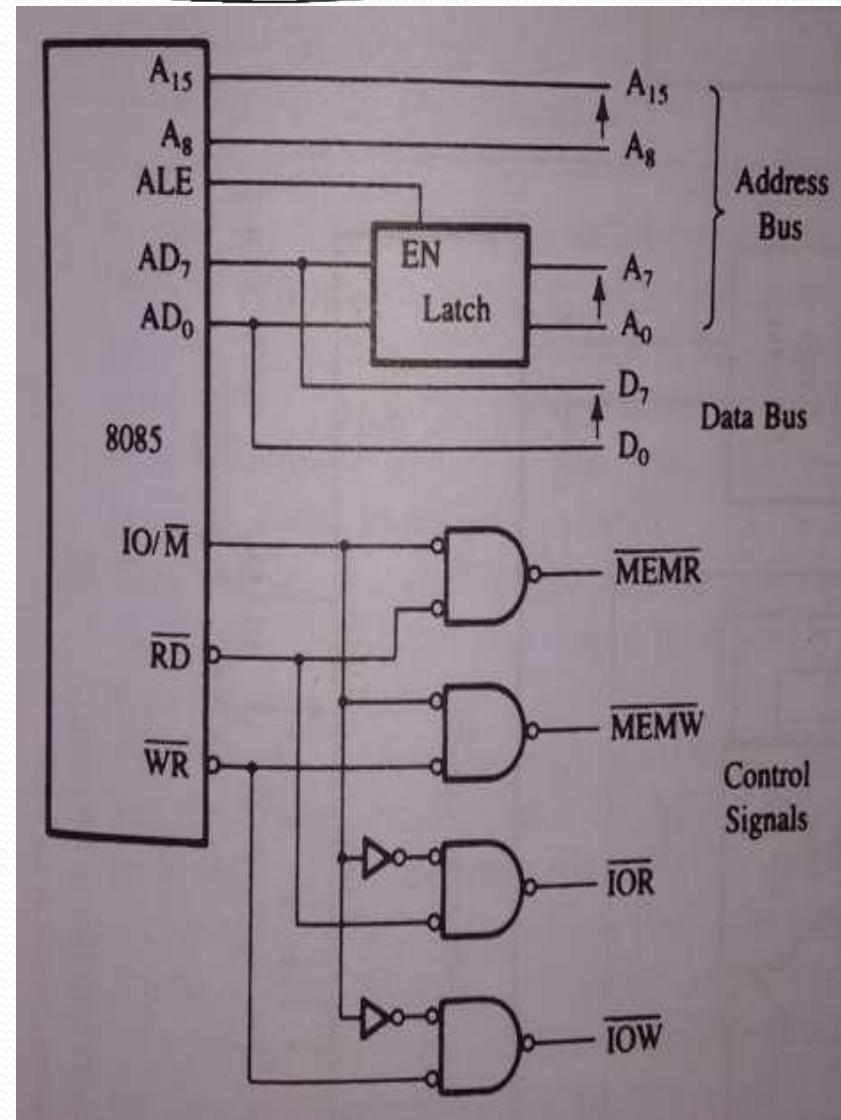
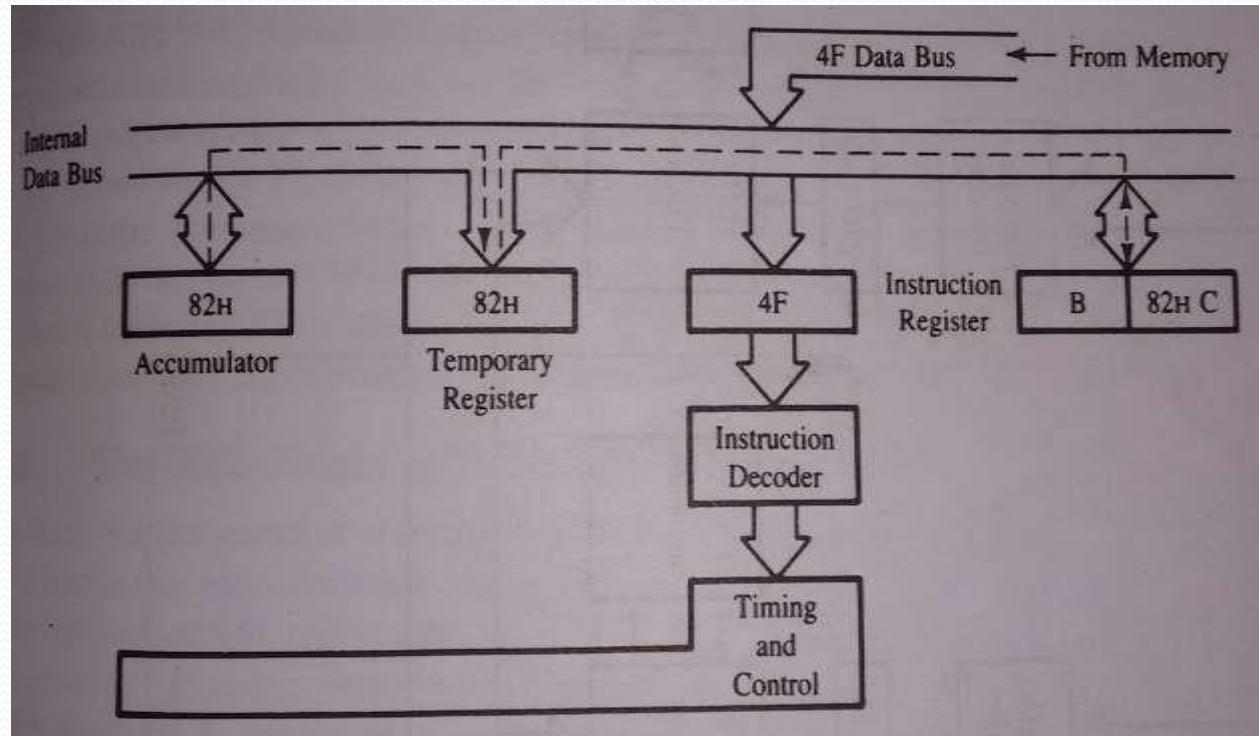


Fig: 8085 De-multiplexed address and data bus with control signal

# Instruction Decoding & Execution

Assume A= 82 H  
Instruction MOV  
C,A(4FH) is fetched.

List of steps in  
decoding & execution



- Step 1:** Places the content of data bus (4FH) in the IR and decode the instruction.
- Step 2:** Transfer the content of the accumulator(82H) to the temporary register in the ALU.
- Step 3:** Transfer the content of temporary register to C register.