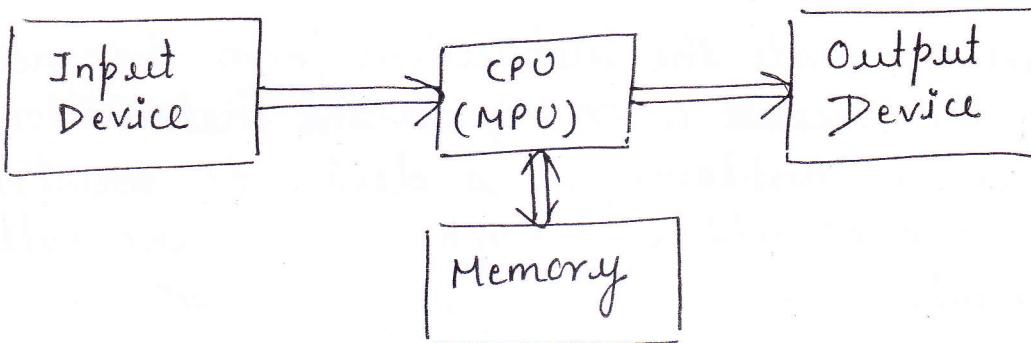


## Introduction to Microprocessor :-

- A microprocessor is a multipurpose, programmable logic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provide results as output.
- Fig shows the block dia. of a digital computer. It consists of four components microprocessor, memory, input & output.
- these four components interact with each other to complete a given task. the physical components of this system are called hardware.
- the set of instructions written for the microprocessor to complete a task is called as a program. A group of programs is called software.



Block dia. of Digital Computer

- Microprocessor works in binary digits 0 and 1. Each bit is a binary digit.
- the digits are represented in terms of electrical voltages in the machine "0" represents one voltage level and bit "1" represents other voltage level.
- A processor with an 8-bit word is called as a 8-bit microprocessor and a processor with 16/32 bit word is called as 16/32 bit microprocessor.

## Memory :-

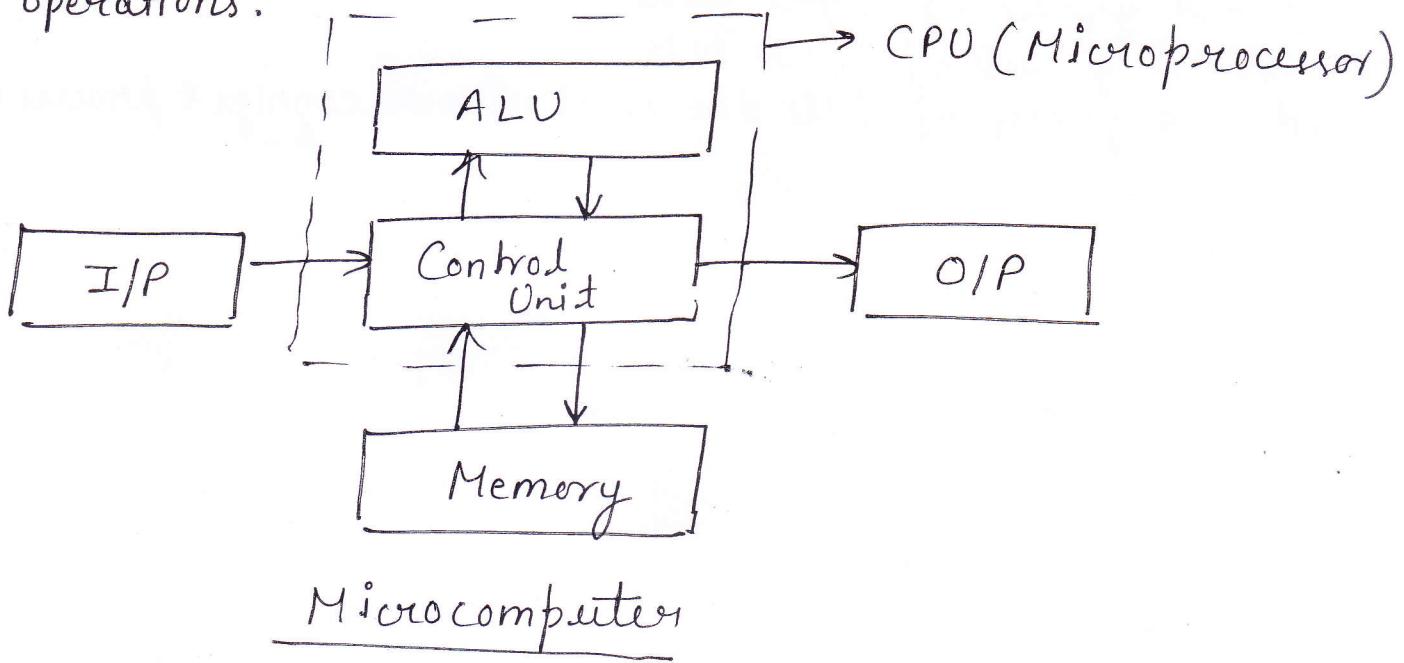
- It stores the binary instructions and data for the microprocessor.
- the memory is similar to the pages of notebook. Each line on the page has a fixed number of binary numbers.
- Each line is an 8-bit register that can store eight binary numbers. Each of these registers are arranged in a sequence referred to as memory.
- the registers are grouped in powers of two e.g. a group of 4096 ( $2^{12}$ ) 8 bit registers on a semiconductor chip is called as 4K bytes of memory.

## Input / Output Device :-

- the user can enter the instructions and data into memory through devices like keyboard or switches. Such devices are called as input devices.
- the microprocessor reads the instructions from the memory and processes the result according to the instructions. the result can be displayed by a device e.g. printer, LCDs, seven segment display. Such devices are called as output devices.

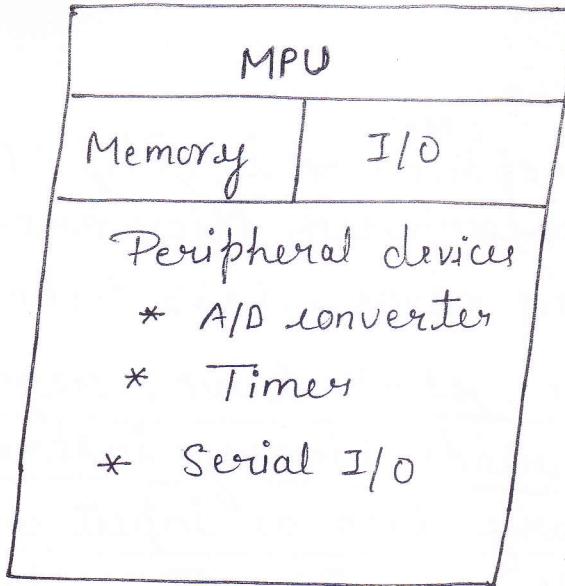
## Difference b/w Microprocessor, Microcontroller & Microcomputer :-

- a) Microprocessor :- Microprocessor is only CPU of ex:- 8085, 8086 etc.. <sup>(MPU)</sup> microcomputer. Microprocessor can't perform any operation alone. (LSI technique)
- It is a programmable, clock drive, register based electronic device that reads binary instruction from memory, accepts binary data as input and process according to instruction & given to output device.
- b) Microcomputer :- MPU along with I/P device, O/P device and memory is called as Microcomputer. Microcomputer perform various i/o, arithmetic & logical operations.



- c) Microcontroller :- "A device that includes microprocessor memory, I/O signal lines, clock circuits, timer, counter & some interfacing devices in a single chip is called microcontroller."

Ex: - 8051 etc.



## Microcontroller

### Important Terms :-

Bit :- A binary digit 0 or 1

Byte :- A group of eight bits

Nibble :- A group of four bits

Word :- A group of bits the computer recognize & process at a time

## Machine Language :-

"the binary medium of communication with a computer through a designed set of instructions specific to each computer is known as machine language."

- A program which has simply a sequence of the binary codes for the instructions is called machine level language program.
- To write a program in machine language, programmer has to memorize thousands of binary instruction codes for a processor. This task is difficult and error prone.
- These instructions are written in hexadecimal code.

Assembly language :-      ex: INR A  $\Rightarrow$  3CH  $\Rightarrow$  0011 1100

"A medium of communication with a computer in which programs are written in mnemonics is known as assembly language"

- It is specific to a given computer.
- Microprocessor has a symbolic code for each instruction called mnemonic.
- The mnemonics for a particular instruction consist of letters that suggest the operation to be performed by that instruction.

ex: Label : Mnemonic Operand1, Operand2 ; Comment

Again : MOV A, B ; Copy contents of B to A

Assembler :- Translates an assembly language program from mnemonics to the binary machine code of a computer.

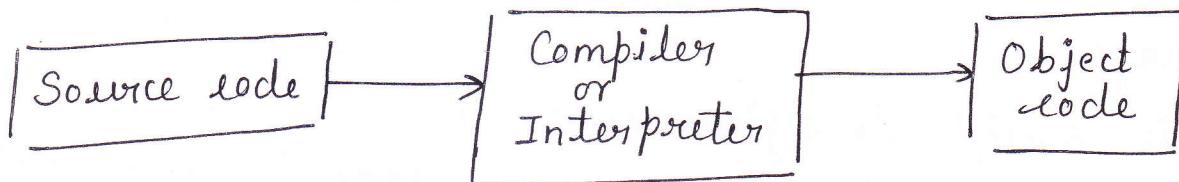
## High-level-language :- (Machine independent)

- " A medium of communication that is machine-independent or specific to a given computer
- " A medium of communication that is independent of a given computer. Programs are written in English-like words, and they can be executed on a machine using a translator (a compiler or an interpreter)."

Compiler :- the compiler reads the entire program first and translates it into the object code that is executed by the microprocessor.

Interpreter :- Interpreter reads one instruction at a time, produces its object code and executes the instruction before reading the next instruction

Ex:- BASIC, PASCAL, C, C++



Note :- Compiler & interpreters require large memory space.

→ Assembly language programs require less memory space.

## EVOLUTION OF MICROPROCESSORS:-

(4)

### First Generation MP:-

4004 :- Add/Data 10/16 bit

- The 1st MP was introduced in 1971 by Intel, U.S.A.
- 16 pin IC.
- It was a 4-bit microprocessor (MP).
- It addresses 640 bytes memory locations
- It has 45 basic instructions
- It contained 2300 PMOS transistors i.e. PMOS technology.

8008 :- 14/8 bit

- It is 8-bit MP design by Intel in 1972.
- Its memory size is 4 times more than 4004 i.e 16KB
- Its speed is 20 μs.
- It has 48 basic instructions.
- It required 20 or more additional device to form a functional CPU.

Not compatible with TTL ekts.

### 2<sup>nd</sup> Generation MP:-

8080 :- 16/8

- It is 8-bit MP design by Intel in 1974.
- Its memory size is 64KB.
- Speed is 10 times more than 8008 i.e 2 μs or 5 lakh instructions / sec.
- It required only two additional devices to form a CPU.
- Compatible with TTL ekts.
- Fabrication based on N-MOS technology.

After Intel 8080, Motorola came out with the MC 6800 another 8-bit up. the advantage of 6800 is that it required only +5v supply rather than the -5v, +5v & +12v supplies required by 8080.

Some of their competitors were the MOS technology 6502 & the Zilog's Z80.

### 8085:- 16/8

- 8-bit MP design by Intel in 1976 which required
- only a +5v supply.
- Memory size is 64KB.
- 40-pin IC.
- Clock speed about 3MHz
- It has 80 basic instructions & 246 op-codes.
- It has internal CLK generator, system controller &
- high CLK frequency.

### 3<sup>rd</sup> Generation MP :-

#### 8086 :- 20/16

- 16-bit MP designed by Intel in 1978.
- Memory size is  $2^{20} = 1\text{MB}$  or 1,048,576
- 40 pin IC.
- CLK frequency upto 8MHz. (execute 2.5 mill inst per sec.)
- It contains 29,000 transistors and is fabricated
- using the H-MOS technology.
- Its instruction set is larger than 8085.

## 8088 :- 20 / 8 bit

- Intel launched 8088, 8/16 bit MP in 1980.
- Memory capacity is 1MB same as 8086.
- 40 pin IC.
- Cdk freq<sup>n</sup> 5-8 MHr
- Widely used in PC / XT.

## 4<sup>th</sup> Generation MP :-

- ⇒ Intel 80186 & 80188 is an improved version of 8086 & 8088 respectively, each have programmable peripheral devices integrated in same package in 1982.
- ⇒ 80286 was designed for use as the CPU in a multiuse or multitasking microcomputer. 80386, 80486---
- ⇒ the evolution along this last path has continued on to 32-bit processors that work with Giga bytes i.e.  $10^9$  bytes of memory.

Pentium, Pentium Pro, Pentium II, Pentium III,  
Pentium IV

## 5<sup>th</sup> Generation MP :-

Pentium in 1993

<u>Processor</u>	<u>Year of Introduction</u>	<u>Data Bus</u>	<u>Address Bus</u>	<u>Addressable Memory</u>
1 <sup>st</sup> { 4004 8008	1971	4	10	640 bytes
	1972	8	14	16 KB
2 <sup>nd</sup> { 8080 8085	1974	8	16	64 KB
	1976	8	16	64 KB
3 <sup>rd</sup> { 8086 8088	1978	16	20	1 MB
	1980	8	20	1 MB
4 <sup>th</sup> { 80186 80188	1981	16	20	1 MB
	1982	8	20	1 MB
5 <sup>th</sup> { 80286 80386	1983	16	24	16 MB → ad <sup>n</sup> that manage (some its 250 ns)
	1985	16/32	26/32	64 MB   4 GB
6 <sup>th</sup> { 80486 Pentium Pro (150 MHz)	1989	32	32	4 GB   fast speed (25 ns) 60 MHz
	1995	64	36	64 GB (3 execute engine) at a time
7 <sup>th</sup> { P-II P-III (266 MHz) P-4. 1.5 GHz	1997	64	36	64 GB
	1999	64	36	64 GB
	2000	64	36	64 GB

## MICROPROCESSOR ARCHITECTURE & ITS OPERATION :-

(6)

All the various functions performed by the microprocessor can be classified in three general categories:-

- [I] → Microprocessor-initiated Operations
- [II] → Internal Operations
- [III] → Peripheral (or external initiated) operations

# [I] Microprocessor-Initiated Operations and 8085 Bus Organization

## Organization. (Bus structure)

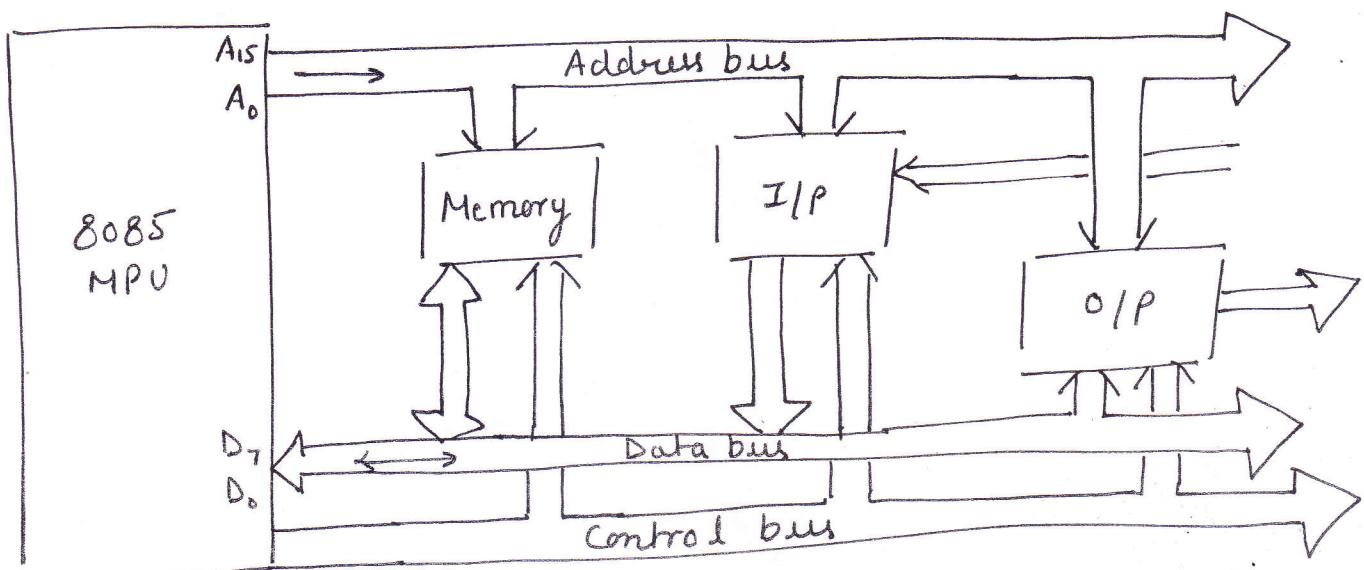
- 1.) Memory Read : Read data (or inst<sup>t</sup>) from memory
- 2.) Memory Write : Writes data (or inst<sup>t</sup>) into memory
- 3.) I/O Read : Accepts data from i/p devices.
- 4.) I/O Write : Sends data to o/p devices

To communicate with peripherals (or memory location), the MPU needs to perform following steps →

- (address bus)
- Step 1: Identify the peripheral or memory location (with address)
  - Step 2: Transfer binary information (data & instructions) (data bus)
  - Step 3: Provide timing & synchronization signals. (control bus)

8085 MPU performs these functions using three set of communication lines called buses :

- Address bus
- Data bus
- Control bus



8085 Bus Architecture

### Step 1: Address bus :-

- It is a group of 16 lines ( $A_0 - A_{15}$ )
- Unidirectional
- It is used to carry 16-bit address.
- Bits flow in one direction - from MPU to peripheral device
- Each peripheral identify by a binary no. called address
- In 8085, 16 address lines, capable of addressing  $2^{16} = 65,536 = 64\text{KB}$  memory locations.  
i.e.  $0000\text{H}$  to  $FFFF\text{H}$  in Hexadecimal.

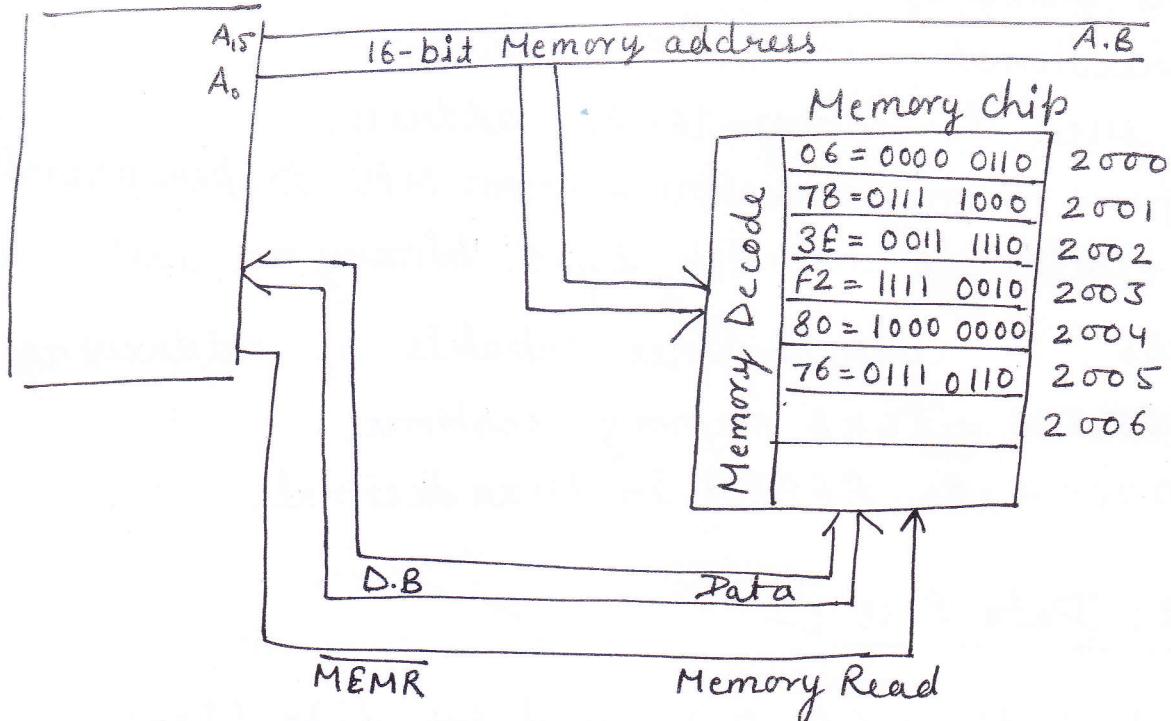
### Step 2: Data Bus :-

- Group of 8 lines ( $D_0 - D_7$ ) used for data flow.
- Bidirectional.
- Data flow in both directions b/w MPU & memory and MPU & peripheral devices.
- Data that appear on data bus -  
 $2^8 = 256$  numbers i.e. {  $00\text{H}$  to  $FF\text{H}$  in Hex  
 $00000000$  to  $11111111$  in decimal }

### Step 3: Control Bus :-

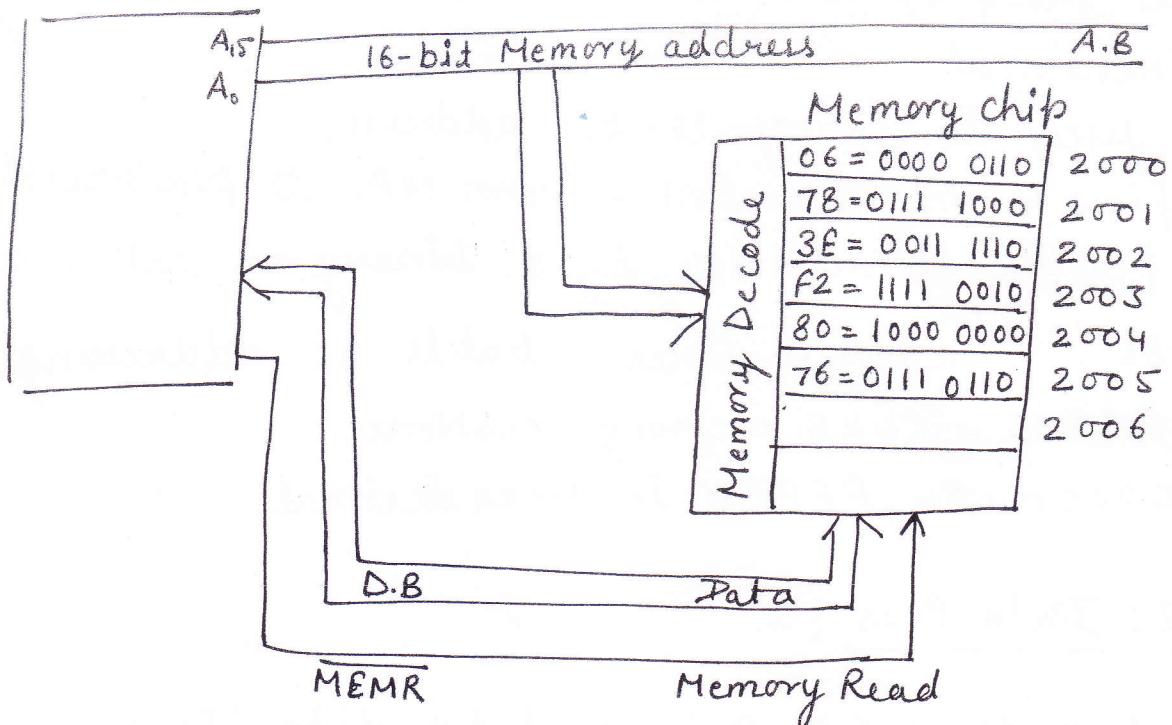
- It carry synchronization sig.
- MPU generates specific control signals for every operation (Memory Read & I/O Write) it performs.

## To Communicate with Memory :-



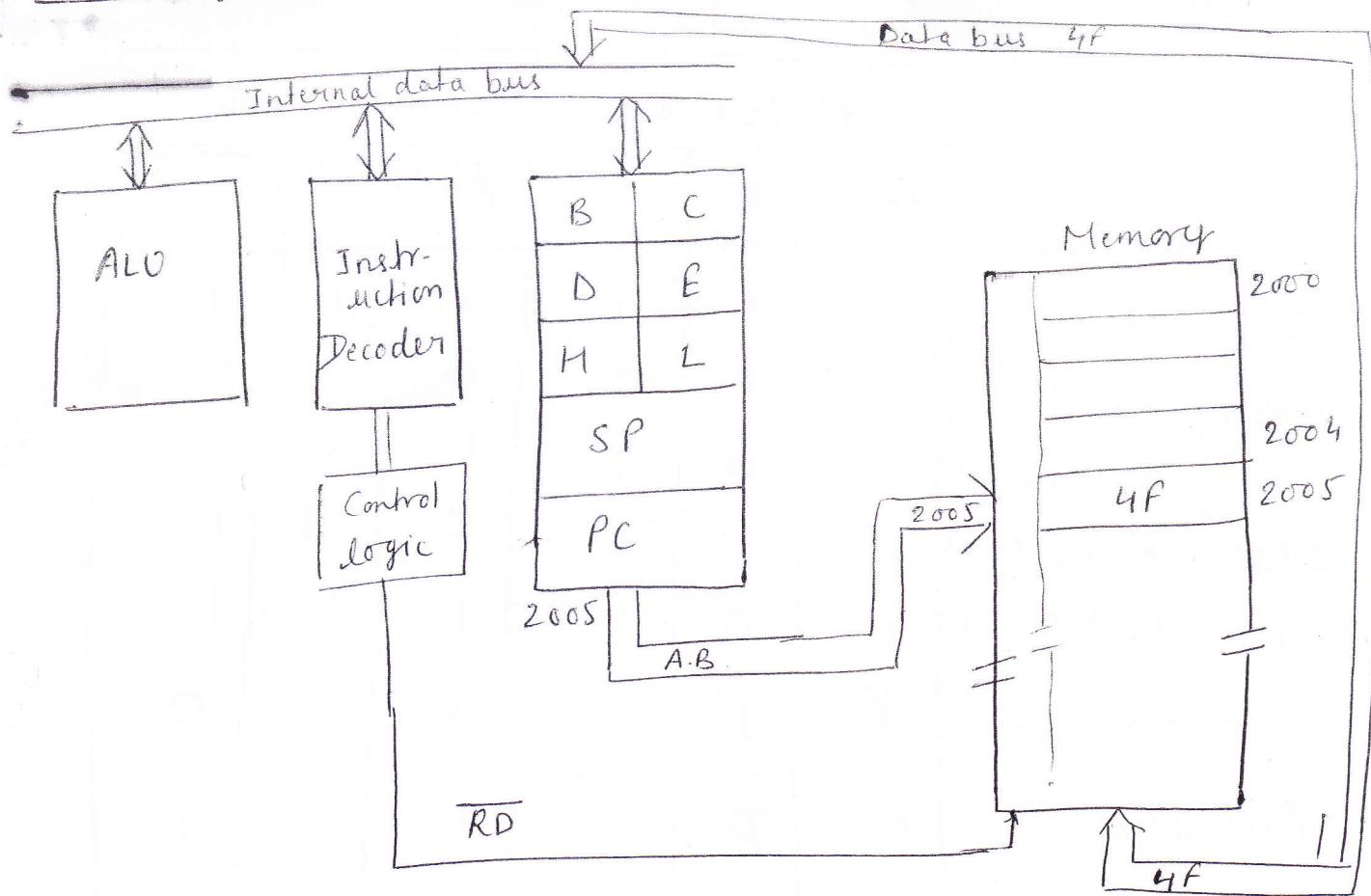
- MPV places the 16-bit address on the address bus.
- The address on the bus is decoded by an external unit & memory location identified.
- The MPV sends a control sig called Memory Read
- Signal MEMR activates the memory chip, and the contents of the memory location (8-bit data) are placed on the data bus & brought inside the MP.

## To Communicate with Memory :-



- MPU places the 16-bit address on the address bus.
- the address on the bus is decoded by an external decoder & memory location identified.
- the MPU sends a control sig called Memory Read
- Signal MEMR activates the memory chip, and the contents of the memory location (8-bit data) are placed on the data bus & brought inside the MP.

Data byte is transferred from memory to the MPU :- (9)



Step 1:- the MP places the 16-bit memory address from the PC on the address bus.

Step 2:- the control unit sends the control sig RD to enable the memory chip.

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

Step 4:- the byte is placed in the instruction decoder of the MP & task is carried out according to the instruction.

8085 Machine cycles :- (M.C)7 Machine cycles

M.C	status sig			Control sig		
	IO/M	S <sub>1</sub>	S <sub>0</sub>	$\overline{RD}$	$\overline{WR}$	<u>INTA</u>
1.) Opcode fetch (4T)	0	1	1	0	1	1
2.) Memory Read (3T)	0	1	0	0	1	1
3.) " Write (3T)	0	0	1	1	0	1
4.) I/O Read (3T)	1	1	0	0	1	1
5.) " Write (3T)	1	0	1	1	0	1
6.) INTR Acknowledge	1	1	1	1	1	0
7.) Bus idle	0	0	0	1	1	1

## 8085 FEATURES

## CHAPTER

(16)

- 1.) 8085 has 40 pin IC.
- 2.) 8085 is a 8-bit microprocessor. i.e it can read, compute or output 8-bit data at a time.
- 3.) It can operate with 3-MHz clock. (maximum = 5 MHz)
- 4.) It has 16-bit address bus to access memory locations  $2^{16} = 2^6 \times 2^{10} = 64 \times 1K = 64 KB$
- 5.) It has 8-bit internal & external data bus.
- 6.) It has 8-bit ALU & registers.
- 7.) It required only single +5v power supply.
- 8.) It has on-chip clock generator, only a crystal is to be connected to 8085.
- 9.) 8085 has 5 Hardware interrupts →

Higher Priority order	TRAP	— Vectored	— Non-maskable
	RST 7.5	— Vectored	— Maskable
	RST 6.5	— Vectored	— Maskable
	RST 5.5	— Vectored	— Maskable
	INTR	— Non-Vectored	— Maskable

- 10.) 8085 has 8 <sup>soft.</sup> hardware interrupts →  
→ Software interrupts are vectored interrupts.

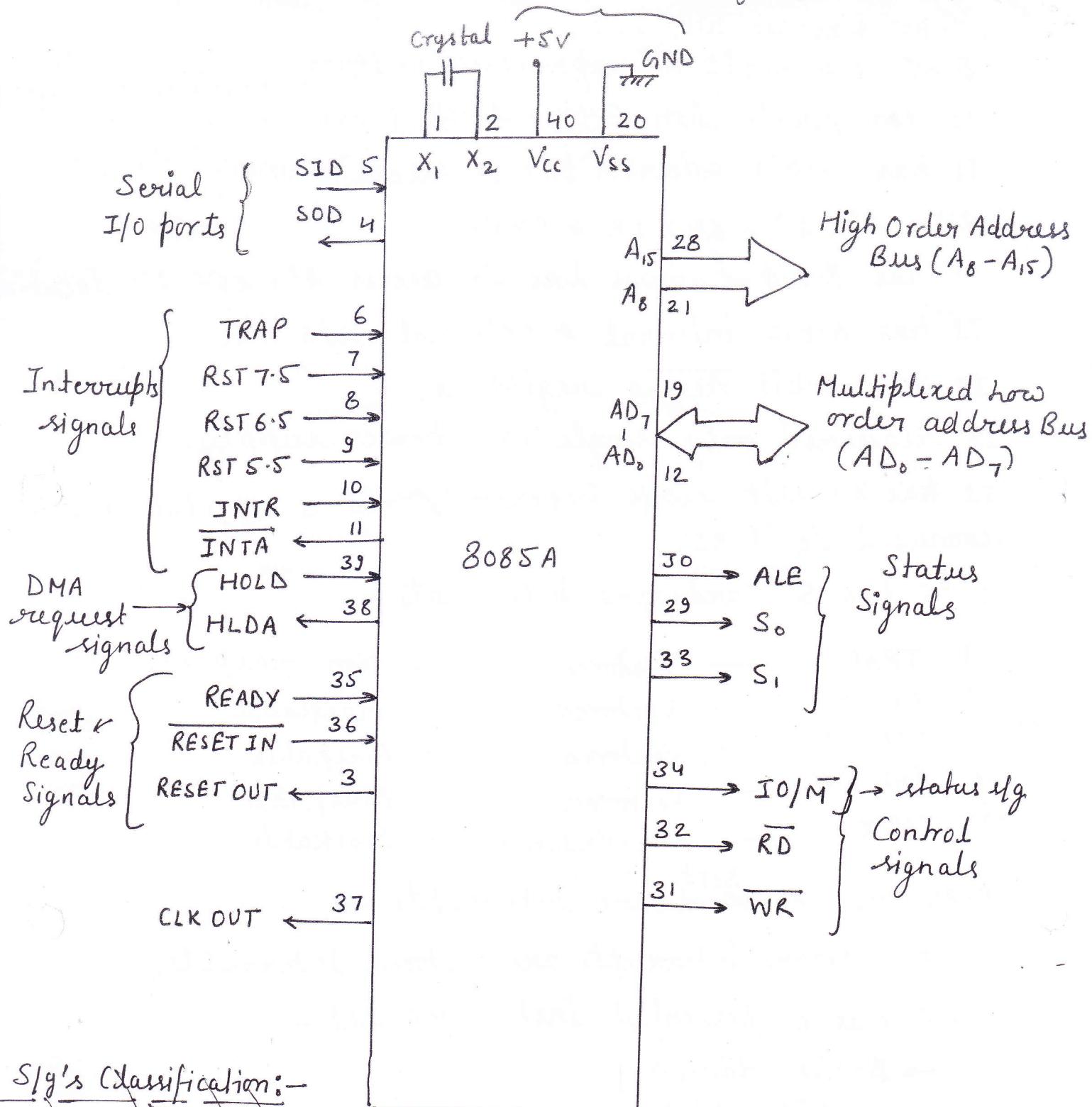
- 11.) 8085 has a powerful instruction set –
  - 8-bit addition
  - 16-bit addition
  - 8-bit subtraction, increment, decrement
  - AND, OR, EX-OR & NOT

- 12.) 8085 can perform operations on bit, byte and some word (16-bit) data.

- 13.) <sup>8085</sup> supports 5 addressing modes : –
  - Register
  - Direct
  - Indirect
  - Implied
  - Immediate

# 8085 Pin Diagram :-

Power supply sig's



## Sig's Classification:-

- Power supply sig's
- Address bus & Data bus
- Clock Signals
- Reset & Ready Signals
- Interrupt Signals
- Status & Control sig's
- DMA sig's
- Serial I/O sig's

b/s

## Signal's Classification :-

- a) Power supply & frequency sig's
- b) Data bus & address bus
- c) Control bus & status sig's
- d) Externally initiated sig's
- e) Serial I/O ports (f) Reset sig's

### a) Power Supply & Frequency :- <sup>CLK</sup>

- $V_{CC}$  : +5 V
- $V_{SS}$  : Ground
- $X_1, X_2$  : A crystal (or RC, LC) is connected at these two pins.  
The freq<sup>n</sup> is internally divided by two:
  - \* to operate a clk at 3MHz
  - \* the crystal should have a freq<sup>n</sup> of 6MHz
- CLK(OUT) : used as the clk for other devices

### b) Data bus & Address Bus :-

Address bus → 16 sig lines are apdlt into two segment  
 $A_{15} - A_8$  &  $A_7 - A_0$ .

→  $A_{15} - A_8$  are unidirectional used for higher order (MSB) addresses of 16-bit address.

### Multiplexed Address / Data bus :-

- $A_7 - A_0$  are used for dual purpose
- Used for lower order address bus (LSB) as well as the data bus.
- During the earlier part of the cycle, these lines are used as the low-order address bus
- During the later part of the cycle, these lines are used as the data bus

### c) Control & status Signals :-

ALE :- (Address latch enable) :-

- This is a +ve going pulse generated every time the processor begins an operation (machine cycle)
- It indicates that the bits on AD<sub>7</sub>-AD<sub>0</sub> are address bits.
- This is used to latch the low-order address (control) from the multiplexed bus & generate A<sub>7</sub>-A<sub>0</sub> lines

RD → (Read) :- active low

- Indicates that the selected I/O or memory device (control) is to be read and data are available on data bus

WR (Write) :- active low

- The data on the data bus are to be written into a selected memory or I/O location.

I/O/M :- (status sig)

- Used to differentiate b/w I/O & memory operations

High → it indicates an I/O operation

Low → " " memory operation

- This sig combined with RD & WR to generate I/O and memory control sig's.

S<sub>0</sub> & S<sub>1</sub> :- (status sig)

- They are used in small sig's

→ Indicate the type of machine cycle in progress

(d) Externally initiated sigs

INTR :- (I/P) Interrupt Request

→ used to interrupt a program execution

INTA :- (O/P) Interrupt Acknowledge

→ used to acknowledge an interrupt.

RST 7.5, RST 6.5, RST 5.5 :- <sup>(I/P)</sup> Restart Interrupt

→ Vectored interrupts that transfer the program control to specific memory locations.

→ they have higher priorities than INTR

TRAP :- (I/P)

→ non-maskable interrupt and has the highest priority

HOLD :- (I/P)

→ Peripheral such as DMA controller is requesting the use of the address & data buses.

HLDA :- (O/P)

→ CPU acknowledges the HOLD request

READY :- (I/P)

→ It is used to delay the CPU Read or Write cycles until a slow responding peripheral is ready to send or accept data

→ this sig goes low, the CPU waits for an integral no. of CLK cycles until it goes high.

(e) Serial I/O ports :-

→ In serial trans., data bits are sent over a single line, one bit at a time.

SID - serial I/P data

SOD - serial O/P "

#### (f) Reset Signals :-

RESET IN :- active low

→ the PC is set to zero, the buses are bi-stated and MPO is reset.

RESET OUT :- active high

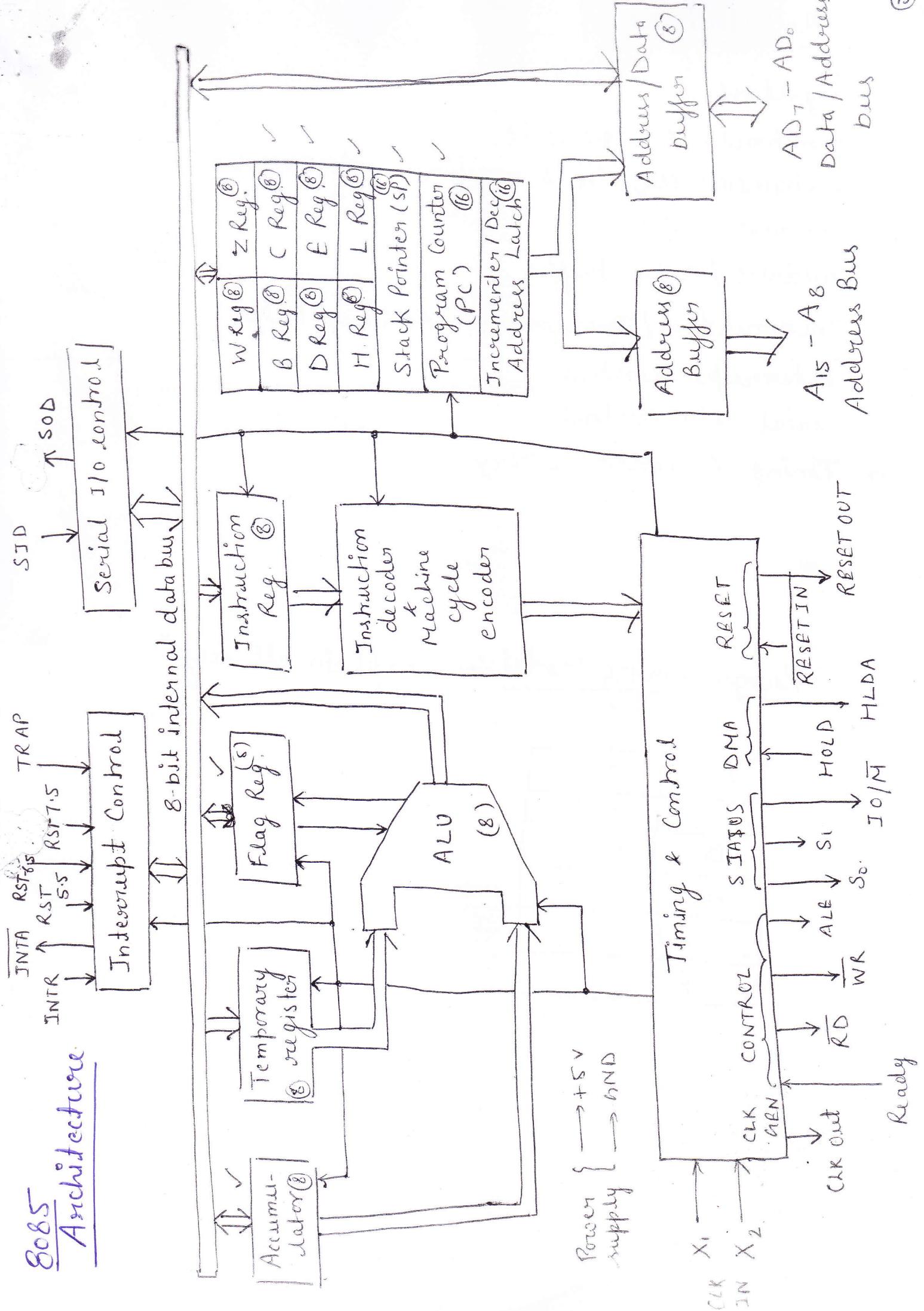
→ this sig indicates that MPO is being reset.

→ the sig can be used to reset

8085 Machine Cycle :- (7 Machine Cycles)

Machine Cycle	Status sig's			Control sig's
	I0/M	S <sub>1</sub>	S <sub>0</sub>	
1.) Opcode Fetch (4T)	0	1	1	$\overline{RD} = 0, \overline{WR} = 1$
2.) Memory Read (3T)	0	1	0	$\overline{RD} = 0, \overline{WR} = 1$
3.) Memory Write (3T)	0	0	1	$\overline{WR} = 0, \overline{RD} = 1$
4.) I/O Read (3T)	1	1	0	$\overline{RD} = 0, \overline{WR} = 1$
5.) I/O Write (3T)	1	0	1	$\overline{WR} = 0, \overline{RD} = 1$
6.) Interrupt Acknowledge	1	1	1	$\overline{INTA} = 0, \overline{RD} = 1$ $\overline{WR} = 1$
7.) Bus idle	0	0	0	$\overline{RD} = 1 \quad \overline{WR} = 1$ $\overline{INTA} = 1$

8085 Architecture



## Architecture of 8085 :-

- Registers
- Arithmetic & Logic Unit
- Instruction decoder & machine cycle encoder
- Address buffer
- Address / Data buffer
- Incrementer / Decrementer Address latch
- Interrupt control
- Serial I/O control
- Timing & control logic.

Programming Model :- explain all reg.

A	F
B	C
D	E
H	L
SP	
P C	

## REGISTER STRUCTURE

Temp Reg.	
W Reg.	Z Reg.
A Reg.	Flag Reg.
B Reg.	C Reg.
D Reg.	E Reg.
H Reg.	L Reg.
Stack Pointer (SP)	
Program Counter (PC)	

## Programming Model

(2)

- the shaded portion of this reg. model is called programmer's model of 8085.
- All these reg. are accessible by the programmer.
- Remaining reg's - temporary, W & Z are not accessible to the programme they are used by u.P. for internal operations.

(1.) General Purpose Registers

(2.) Temporary Registers

(3.) Special Purpose Registers

(4.) 16 bit Registers.

1. General Purpose Registers :-

→ B, C, D, E, H & L are 8-bit reg. can be used as a separate 8-bit reg. or as 16-bit reg. pairs BC, DE & HL.

2. Temporary Registers :-

→ these reg's are <sup>internally</sup> used for execution of some instructions.

→ Programmer can not access these registers.

a) temp Reg. :- ALU has 2 i/P's,

→ it is used for execution of most of the arithmetic & logic instructions.

b) W & Z Reg. :- To hold 8-bit data.

→ At the time of exchange W & Z reg's are used for temp' storage of data.

## Special Purpose Registers

- a.) Register A (Accumulator) :- It is used 8-bit reg.  
 → Used in arithmetic & logic, load & store operations as well as I/O operations.  
 → Result of arithmetic & logical operations is stored in reg. A.
- b.) Flag Reg :- 8-bit reg. → Five bit array in " in form of flag

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
S	Z		AC		P		CY

S (Sign flag) → \* if bit D<sub>7</sub> is 1, sign flag is set  
 the no. will be -ve no.

\* if D<sub>7</sub> is 0, the no. will be +ve no.

Z (Zero flag) → Zero flag is set to 1 when the result is zero, otherwise it is reset.

CY (Carry) → If an arithmetic operation results in a carry, the CY flag is set, otherwise it is reset.

P (Parity) → If the result has an even no. of 1s, the flag is set, for odd no. of ones the flag is reset.

AC (Auxiliary Carry) → In an arithmetic operation, when a carry is generated by digit D<sub>3</sub> and passed to digit D<sub>4</sub>, the AC flag is set. This flag is ~~not~~ used internally for BCD.

16-bit Registers:- These are 16-bit reg used to hold memory addresses.

a) Program Counter (PC):-

- The MP uses the PC register to sequence the execution of the instructions.
- The fun<sup>t</sup> of the program counter is to point to the memory address from which the next byte is to be fetched.
- When a byte (machine code) is being fetched, the PC is incremented by one to point to the next memory location.

b) Stack Pointer (SP):-

- Used as a memory pointer, it points to a memory location in R/W memory called stack.
- A reserved area of the memory in the RAM where temporary info<sup>m</sup> may be stored.
- SP is used to hold the address of the most recent stack entry.

## ALU :-

### Instruction Decoder & Instruction Reg :-

- the processor fetches the opode of instruction from memory and stores this opode in the instruction reg.
- It is then sent to the instruction decoder, Decoder decodes it and gives the timing & control signals.

### Machine encoder →

- 8085 executes 7 types of machine cycles.  
It gives the info<sup>m</sup> about which machine cycle is executing in the encoded form on the S<sub>0</sub>, S<sub>1</sub> & I/O/M line.

### Address Buffer :-

- 8-bit unidirectional buffer
- Used to drive external high order address bus (A<sub>15</sub>-1)

### Address / Data Buffer

- 8-bit bi-directional buffer
- Used to drive multiplexed address / data bus i.e low order address (A<sub>7</sub>-A<sub>0</sub>) & data bus (D<sub>7</sub>-D<sub>0</sub>)

### Inc./Dec. address Latch

- 16-bit reg. is used to increment or decrement the contents of PC or SP as a part of execution of instructions related to them.

## Serial I/O Control :-

- (22)
- Data trans<sup>m</sup> over long distance and comm<sup>n</sup> with CRT terminal, to transmit data bit by bit to reduce the cost of cabling.
  - Serial com<sup>n</sup> one bit is transferred at a time over a single line.
  - SOD :- Serial Output Data line is used to send data serially.
  - SID :- Serial Input data line is used to receive data serially.

## Timing & Control Cktry :-

- the control entry & operations in 8085 are synchronized with the help of 4R clg.

## Interrupt Control :-

- the processor fetches, decodes & executes instruction in a sequence.

→



S|W

RST n ( $8 \times n$ )H

$$\text{RST } 0 = (8 \times 0 = 0) = 0000H$$

$$\text{RST } 1 = (8 \times 1 = 8) = 0008H$$

$$\text{RST } 2 = (8 \times 2 = 16) = 0010H$$

$$\text{RST } 3 = (8 \times 3 = 24) = 0018H$$

$$\text{RST } 4 = (8 \times 4 = 32) = 0020H$$

$$\text{RST } 5 = (8 \times 5 = 40) = 0028H$$

$$\text{RST } 6 = (8 \times 6 = 48) = 0030H$$

$$\text{RST } 7 = (8 \times 7 = 56) = 0038H$$

H|W

$$\text{TRAP}(\text{RST } 4.5) = 8 \times 4.5 = 0024H$$

$$\text{RST } 7.5 = 8 \times 7.5 = 003CH$$

$$\text{RST } -6.5 = 8 \times 6.5 = 0034H$$

$$\text{RST } -5.5 = 8 \times 5.5 = 002CH$$

INTR

## Pin diagram

$X_1$	1	40	$V_{CC}$
$X_2$	2	39	HOLD
RESET OUT	3	38	HLDA
SOD	4	37	CLK OUT
SID	5	36	Reset/N
TRAP	6	35	Ready
RST 7.5	7	34	$I/O/M$
RST 6.5	8	33	$S_1$
RST 5.5	9	32	$\overline{RD}$
INTR	10	31	$\overline{WR}$
$\overline{INTA}$	11	30	ALE
$AD_0$	12	29	$S_0$
$AD_1$	13	28	$A_{15}$
$AD_2$	14	27	$A_{14}$
$AD_3$	15	26	$A_{13}$
$AD_4$	16	25	$A_{12}$
$AD_5$	17	24	$A_{11}$
$AD_6$	18	23	$A_{10}$
$AD_7$	19	22	$A_9$
$V_{SS}$	20	21	$A_8$

8085

## Demultiplexing Address and Data Bus ( $AD_0 - AD_7$ )

- Lower order address and data bus  $AD_0 - AD_7$  are multiplexed.
- 74LS373 latch IC is used to latch the address issued by 8085.
- It has 8 latches
- the clock pin of these latches are connected together and available at the output pin enable (G).

### Operation :-

- Address will appear on  $AD_0 - AD_7$  lines.
- → ALE will go high, forcing enable (G) = 1. The latch is transparent, means that the output changes according to input data.
- Output of the latch represents the low-order address bus  $A_0 - A_7$  after latching operation.
- When the ALE goes low, the data byte is latched until the next ALE appears.
- $AD_0 - AD_7$  will now be used as data bus, labeled as  $D_0$  to  $D_7$  only.
- Thus we have  $A_0 - A_{15}$  address bus and  $D_0 - D_7$  data bus.

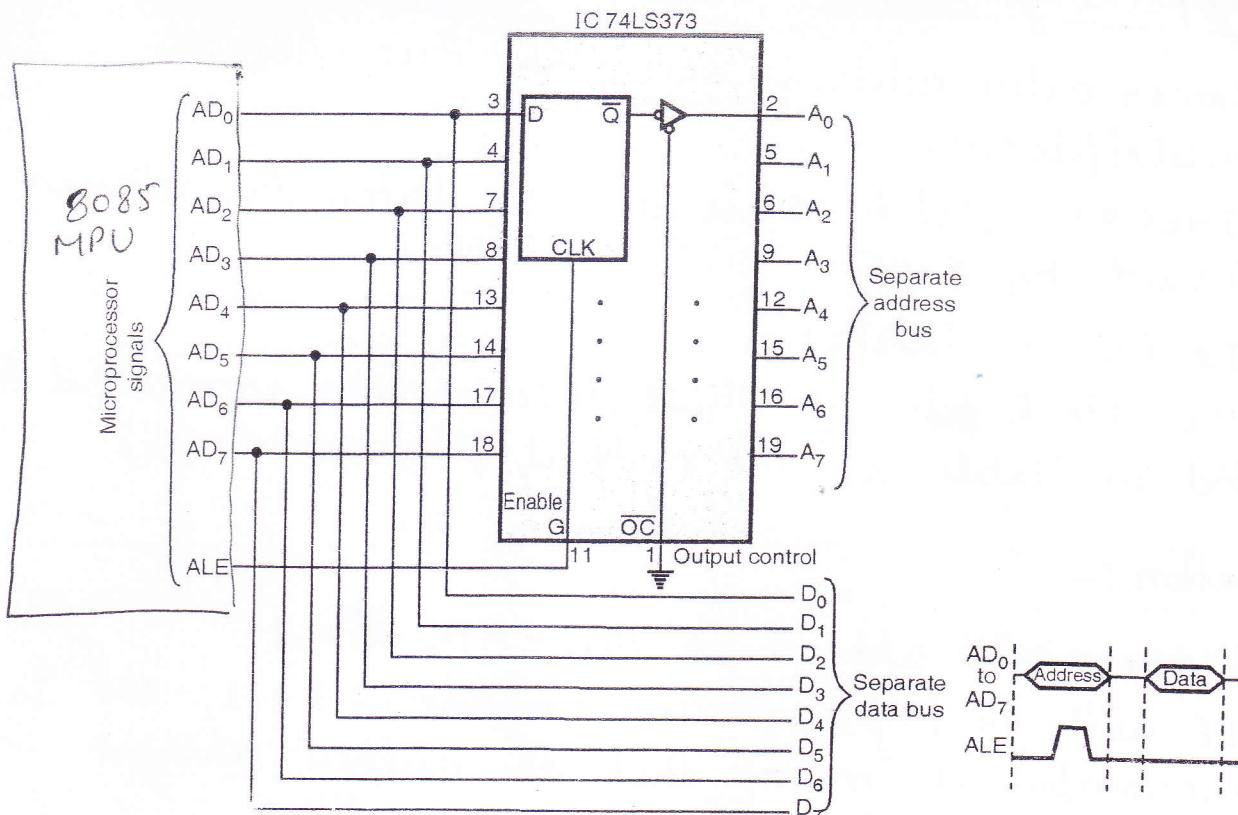


Fig 2.4 (a) Demultiplexing address/data

(b) Waveforms

## Generating control signals :-

- The 8085 microprocessor contains  $\text{IO/M}$  to differentiate between memory and I/O device,  $\overline{\text{RD}}$  to read data and  $\overline{\text{WR}}$  to write data.
- By using these lines we can generate four separate signals read/write for memory and read/write for I/O device.
- The signal  $\text{IO/M}$  goes low for the memory operation. This signal is logically ORed with  $\overline{\text{RD}}$  and  $\overline{\text{WR}}$  to get  $\overline{\text{MEMR}}$  and  $\overline{\text{MEMW}}$  signals.
- The signal  $\text{IO/M}$  goes high for the I/O operation. This signal is logically ORed with  $\overline{\text{RD}}$  and  $\overline{\text{WR}}$  to get  $\overline{\text{IOR}}$  and  $\overline{\text{IOW}}$  signals.

$\text{IO/M}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	Control Signal
0	0	1	$\overline{\text{MEMR}}$ (Read data from memory)
0	1	0	$\overline{\text{MEMW}}$ (Write data into memory)
1	0	1	$\overline{\text{IOR}}$ (Read data from I/O device)
1	1	0	$\overline{\text{IOW}}$ (Write data into I/O device)

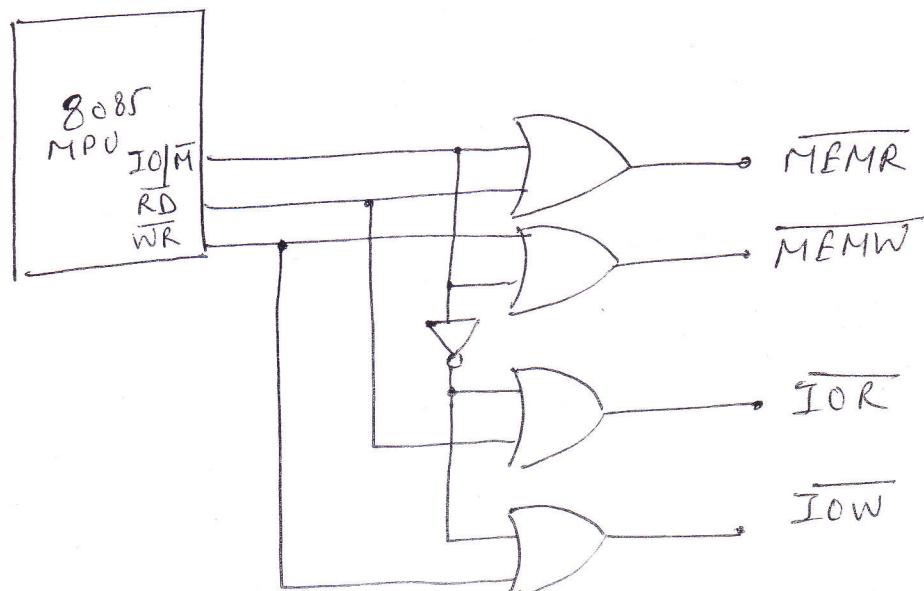


fig-2.5 Generation of control signals

## MEMORY :- Interfacing

→ Memory is a essential component of a microprocessor. It is used to store binary instructions and data for microprocessor.

- 1.) Primary (main) Memory
- 2.) Secondary (Storage) Memory

### 1.) Primary Memory :-

1.) RAM :- It is made of registers, and each register has a group of flip-flops or FET that store bits of information. (memory cell)  
this memory used to hold programs and store data.

2.) ROM :- It stores information permanently in the form of diodes; the group of diodes can be viewed as reg.

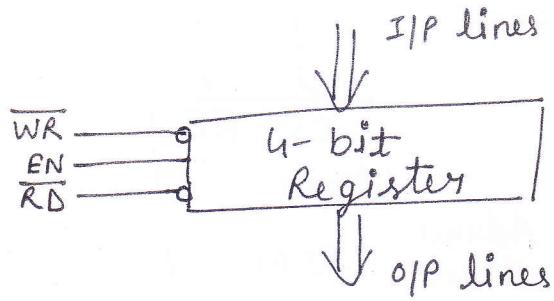
### 2.) Secondary Memory :-

→ Magnetic tapes or disks

## Flip-flop or latch :-

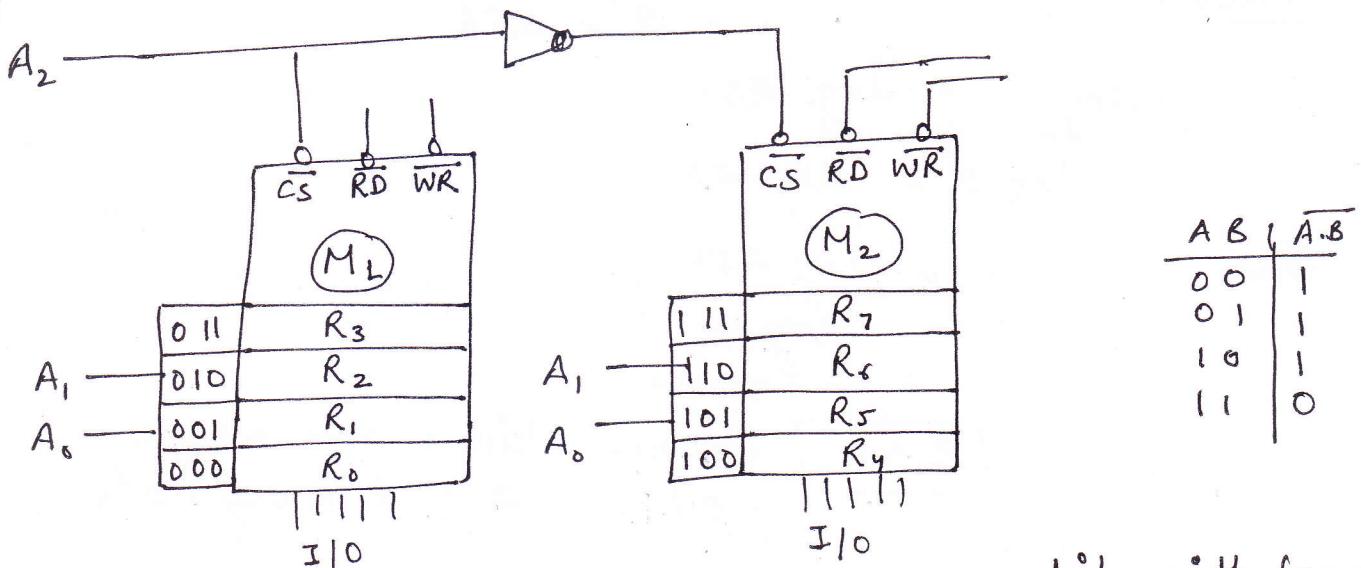
- A flip flop or a latch is a basic element of memory.
- To store or write a bit in the latch, we need an input data bit ( $D_{IN}$ ) and an enable signal (EN).
- The stored bit is always available on the output line  $D_{OUT}$ .

# Memory



size :- 1x4 bit (One Registers with four memory cell.)  
4x8 bit (Four " with Eight " " )

## Two MEMORY CHIPS WITH FOUR REGISTERS :-

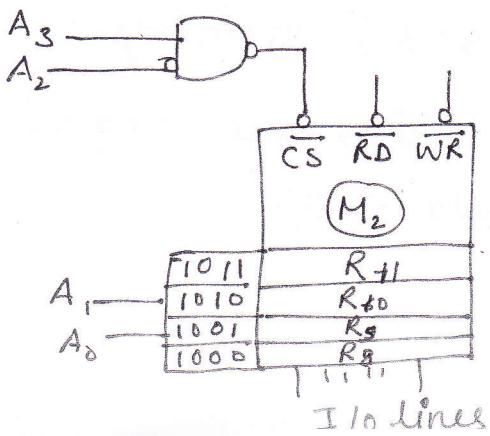
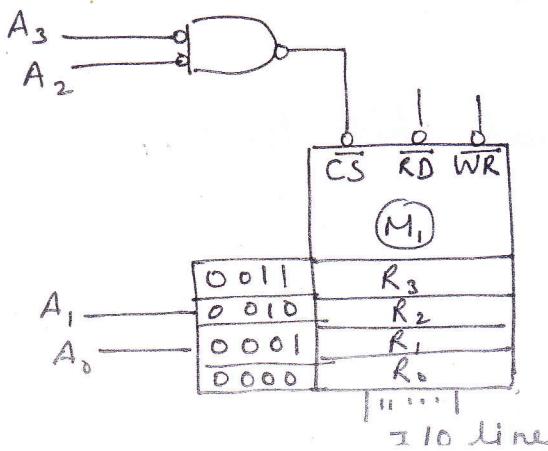


# We have 4-address line & Two memory chip with four reg each

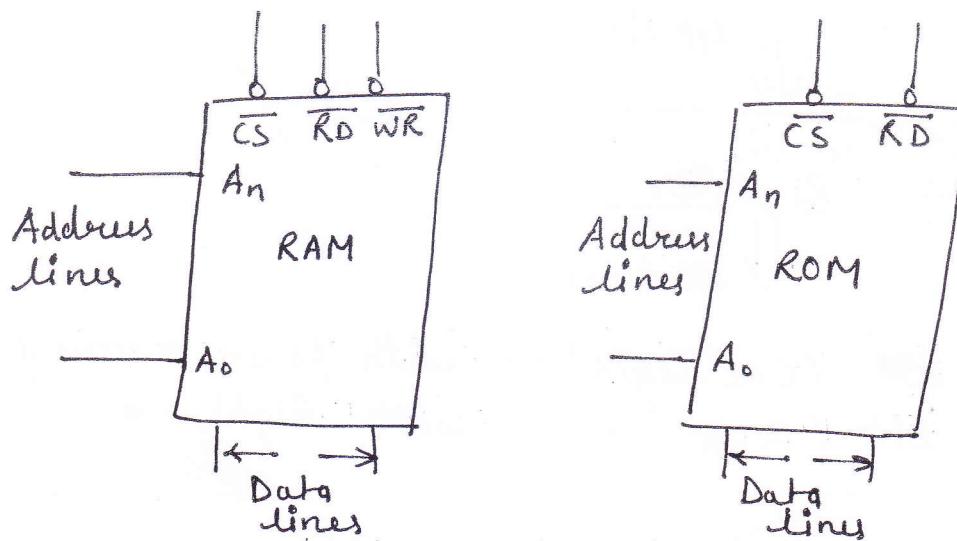
$$4\text{-addr} = 2^4 = 16 \text{ reg}^+$$

$\Rightarrow$  we need only three address lines to select eight reg

(M<sub>1</sub>)  $\Rightarrow A_3 A_2 = 00$  (0000 to 0011) (0 to 3) (M<sub>2</sub>)  $\Rightarrow A_3 A_2 = 10$  (1000 to 1011) (8 to 8)



## Memory Model :-



Memory Address lines :-  $2^x = 256$

$$\log 2^x = \log 256$$

$$x \log 2 = \log 256$$

$$x = \frac{\log 256}{\log 2}$$

$x = 8$  address lines are required to obtain 256 binary no.'s.

Memory Word size :- (1, 4 & 8) Memory chips

→ Total no. of bits it can store

ex:  $1024 \times 4$  (1024 reg. & each reg. store four bits)

so ( $1024 \times 4 = 4096$ ) bit store.

For  $(1024 \times 8)$  memory :-

Two chips are required each chip have four data lines.

Questions

Q.1 Calculate the address lines required for an 8K byte  
( $1024 \times 8 = 8192$  reg.) memory chip.

Sol.

No. of address lines  $\rightarrow$

$$x = \frac{\log 8192}{\log 2} = 13 \text{ address lines}$$

Q.2 Cal. the no. of chips needed to design 8K-byte memory if the memory chip size is  $1024 \times 1$ .

Sol.  $1024 \times 1$  ( $1024$  reg & each reg. can store one bit with one data line)

- $\Rightarrow$  We need eight data lines for byte-size memory.
- $\Rightarrow$  8-chips required for 1 K-byte memory.
- $\Rightarrow$  For 8K byte memory  $\Rightarrow$  we need 64 chips.

$$\frac{8 \text{ K byte}}{1 \text{ K byte}} = \frac{1024 \times 8}{1024 \times 1} = \frac{8192 \times 8}{1024 \times 1} = 64 \text{ chips}$$

## Difference b/w Memory mapped I/O & I/O mapped I/O port.

### I/O Mapped I/O

1. Port address is 8-bit (From 00H to FFH) so 256 I/O ports can be connected.
2. Control signals are used  $\overline{IORD}$ ,  $\overline{IOWR}$  and  $I0/\overline{M}=1$ .
3. The instructions which can be used for data transfer are 'IN' & 'OUT'.
4. Memory space is not lost.
5. 10T states are required for IN & OUT instructions.
6. Less hardware is needed to decode 8-bit of address.

### Memory Mapped I/O

1. Port address is 16-bit (from 0000H - FFFFH) so maximum a 64K I/O ports can be connected.
2. Control signals are used  $\overline{MRD}$ ,  $\overline{MWR}$  and  $I0/\overline{M}=0$ .
3. Any memory related instruction can be used! So arithmetic & logical operations are possible.  
ex. LDA, STA
4. The address assigned to I/O ports can't be assigned to any other memory location, hence space is lost.
5. 13T states are used for STA and LDA instructions.
6. More hardware is needed to decode 16 bit of address.