CIE-1 **M.P.D** × Radhe × Radhe

- Features of 8085 μP
- " " 8086 "
- Architecture of 8085 μP
- " " 8086 μP
- Assembler Directive of 8085 μp
- Addressing modes of 8085 μP
- " " " 8086 μP
- Instruction set of 8085 μP

Ques 1. Features of 8085 μP. :->

Ans 1 Here as some key features of 8085

- Architecture : It follows the von neumann architecture, where program and data are stored in the same memory.
- Word length : It is an 8-bit ~~m~~ processor. It can ~~store~~ process data in 8-bit chunks at a time.
- Registers: There are several register in 8085. including Accumulator (A), General purpose registers (B, C, D, E, H, L), stack pointer (SP). and PC (program counter).
- Instruction set : It consist of around 74 instruction. Instruction like Arithmatic logical, data transfer and control.

- Clock speed : 8085 μp typically operates at a clock speed of 3MHz, although different versions and implementations may have different clock speed or frequencies.

- Storage / memory : It can adress upto 64KB of memory through its 16 bit address bus. The memory can be both read from and written to.

- ~~#~~ Interrupts :-> It supports five hardware interrupts, which can be enabled & disabled as needed.

* FEATURES OF 8086 μP :->

- It is a 16-bit μP, which means it can processes data and address 16-bit chunks.

- It has 16 bit data bus that allows it to transfer 16 bits of data at a time.

- The 8086 μp has a 20 bit address bus allowing it to address up to 2^20 unique memory location.

- Due to its 20 bit address bus, the 8086 can theoretically address up to 1 MB. of memory

- It has set of (4) register, including general purpose register (AX, BX, CX, DX), segment register (BS, DS, SS, ES) and pointer index segment (SI, DI, BP, SP).

- The original speed ranging from 5MHz to 10MHz
- It has PIC (priority interrupt Controller) which can handle multiple interrupts simultaneously.

- Addressing Modes of 8086 μP.

1. 8085 Register address : The operand is placed in one of the 16 bit or 8-bit general purpose register

ex :-> MOV AX, BX
ADD AL, BL
ADD CX, DX.

2. 8085 Immediate Addressing : The operand is specified in instruction itself.

ex :-> MOV AL, 35H
MOV BX, 1050H
MOV [0500], 3598H.

3. 8085 Direct Addressing : The address of the operand explicitly in the instruction.

Ex :-> MOV AX, [SI] loads the content of the memory location pointed to by the 'SI' register into the 'AL' register.

4. 8085 Register Indirect Addressing :->
The instruction specifies a register which contain the address of the operand.

8085 5. Implicit / Implied :-> CMA, RAR, RAL. etc.

Ex:-> MOV AX,[BX] loads the content of the memory location pointed to by the 'BX' register into the 'AX' register.

5. Base register Addressing: The address is calculated by adding an offset to the content of an index register.

Ex:-> MOV AX, [BX + SI] , access the memory location at the address 'BX + SI'.

Effective Address (offset) => [BX + 8-bit or 16-bit displacement]

• MOV AL, [BX + 05] → 8 bit
MOV AL, [BX + 0105H] → 16-bit.

7. ~~Based~~ Indexed Addressing :-

The offset of an operand is the sum of the content of an Index register SI or DI and an 8-bit or 16-bit displacement,

effective address(offset) = [SI or DI + 8-bit or 16 bit displacement].

8. Based Indexed Addressing:
The offset of operand is the sum of the content of base register BX or BP and an index register SI or DI.

Effective Address (offset) : [BX or BP] + [SI or DI].

9. Base Indexed with Displacement :-

In this mode of addressing, the operand's offset is given by.

effective address :- BX + SI or DI + 8-bit or 16-bit displace

Ex :- MOV AX, [BX + SI + 05] . 8 bit displ.

MOV AX, [BX + SI + 0105H] . 16-bit disp.

• ~~Instruction~~ Assembler Directives of 8085 μP

1. DB : Define Byte

It is used for allocating and initializing single or multiple data ~~types~~. bytes.

AREA DB 30H, 52H, 35H.

2. DW : Define Word.

It is used for initializing single or multiple data words. (16-bits).

MARK DW 1020H, 4216H.

3. END : End of program

It is used at the time of program termination

4. EQU : Equate

It is used to Assign any numerical value or constant to the variable

DONE EQU 10H.

Variable name 'DONE' has value 10H.

- MACRO : represents beginning.
  Shows the beginning of macro along with defining and parameters.
- ENDM : End of macro.
  Indicates the termination of macro.
- [illegible] ORG :
  The directive is used at the time of assigning starting address for module or segment.

  ORG. 1050H.

  By this instruction, the assembler gets to know that the statements following this instruction must be stored in the memory location beginning with address 1050H.

• **8085 Instruction Set :-**

• The various techniques to specify data for instruct

1. 8-bit or 16-bit data may be directly given in the instruction itself.
2. The address of the memory location, I/O port or I/O device, where data resides, may be given in the instruction itself.
3. In some instructions, only one register is specified. The content of the specified register is one of the operands.
4. Some instructions specify two registers. The contents of the registers are the required data.
5. In some instructions, data is implied. The most instructions of this type operate on the content of the accumulator.

Due to different ways of specifying data for instruction the machine codes of all instructions are not of the same length. It may 1-byte, 2 byte and 3-byte instruction.

• Data transfer Group :-

```
    r, M
Mov r1, r2 ;  MVI  r, data  ;  LDA addr ,  LHLD addr
    M, r           M, data     STA addr ,  SHLD addr
```

rithmetic Group :->

ADD r		ADC r		SUB r		SBB r		INR r
ADD M	;	ADC M	;	SUB M	;	SBB M	;	INR M ;
ADD data		ADC data		SUI data		SBI data		DCR r
								DCR M

• Logic Group :->

ANA r		XRA r		ORA r		CMP r
ANA M	;	XRA M	;	ORA M	;	CMP M ;
ANI data		XRI data		ORI data		CPI data

RLC		CMA	
RRC	;	CMC	;
RAL		STC	
RAR			

• Branch Group :->

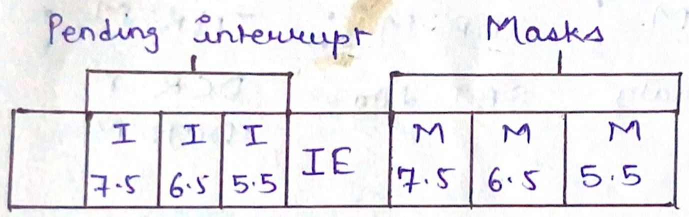
JMP addr		CALL addr		RET		RST ;
Jcondition addr	;	Ccondition addr	;	Rcondition	;	
y (ccc)		iy (ccc)		iy (ccc)		

• Stack I/O A Machine control group :->

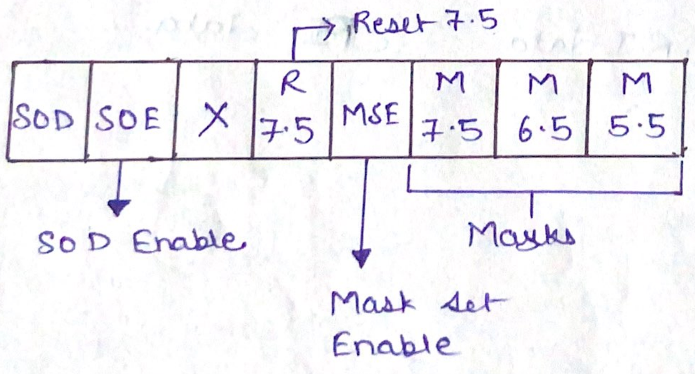
PUSH rp	;	PUSH PSW	;	SPHL	;	IN Port	;	EI	;	HLT ;
POP rp		POP PSW		XTHL		OUT Port		DI		NOP

RIM	;
SIM	

• RIM :-



• SIM :-



Conditions	C	C	C
NZ	0	0	0
Z	0	0	1
NC	0	1	0
C	0	1	1
PO	1	0	0
PE	1	0	1
P	1	1	0
M	1	1	1