Microprocessor Q.B. 3

H. * 2 marks

1 What is role of XCHG instruction in assembly language program? Give example

→ The XCHG instruction exchanges contents of the destination and source.
Here, destination & source can be register and register or register and memory location, but XCHG cannot interchange value of 2 memory locations
For example,

XCHG BX, CX (Exchang data in BX with data in CX)
~~XCHG AX,~~

2 State the use of STC & CMC instruction of 8086

→ STC & CMC are both flag manipulation instructions
- STC → Sets carry flag & doesn't affect any other flag.
- CMC → Complements the carry flag & doesn't affect any other flag

B. * ~~10~~ 4 marks

1. Difference between intersegment & intrasegment CALL

→

Intrasegment call	Intersegment call
· The intrasegment call is a call for which the addresses must lie within current code segment. ~~It is achieved only~~	· The intersegment call is a call from one code segment to another
· It is achieved by only modifying value of IP	· ~~It~~ For this to be effective, both CS & IP values are to be modified

2. Explain logical instructions of 8086

→ Logical instructions of 8086 are as follows:

- ADD instruction - It is used to add current contents of destination with that of source & store result in destination
- ADC - It adds contents like ADD instruction but also adds the carry flag
- SUB - It is used to subtract the current contents of destination with that of source & store result in destination
- SBB - Performs instructions like SUB, but uses the carry flag & may affect all flags
- CMP - Compares source operand with destination which internally subscribes source from ~~dest~~ destination without storing result. If they're equal, zero flag is set
- ~~AND~~ AND - Logically ANDs each bit of source with destination & stores result in destination
- OR - Logically ORs each bit of source with corresponding bit in destination & stores result in destination
- NOT - Inverts content of operand bit by bit

3. Describe any 4 ~~ass~~ String instructions of 8086 assembly language

→ - MOVS - Used to copy a word or byte from location in data segment to a location in extra segment.

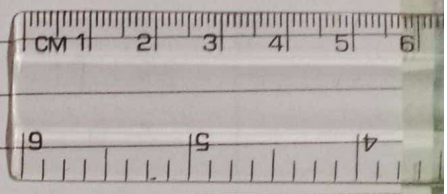
- REP - It is used with string instruction; it repeats an instruction until the specified condition becomes false.
- LODS - It copies a byte from string location pointed to by SI to AL or word from string location pointed to by SI to AX

· CMPS - It is used to compare strings, byte wise or word wise

4. List any 4 instructions from bit manipulation instructions of 8086

→ Instructions from bit manipulation instructions of 8086 are as follows:

- SHL (Shift left)
- SHR (Shift right)
- ROL (Rotate left)
- ROR (Rotate right)



5. Write classification of instruction set of 8086. Explain any one type out of them.

→ Instruction set of 8086 MP is classified as:

1. Data transfer instructions
2. Arithmetic instructions
3. Bit Manipulation instructions
4. String instructions
5. Program execution transfer instructions
6. Process control instructions
7. Iteration control instructions

· Data copy / Transfer instructions
These type of instructions are used to transfer data from source operand to destination
eg. MOV, PUSH, POP, etc

6. Describe any 4 arithmetic instructions with example

→ ① ADD - Adds current content of source & destination & stores result in destination. flags affected.
eg ADD AX, BX, AX ← AX + BX

② SUB - Subtracts current contents of distination from source & store result in destination
eg SUB AX, BX

(3) ADC - ADDs source & destination including carry & stores result in destination.
eg ADC CX, BX

(4) SBB - Subtracts source operand with carry like Normal SUB. Uses carry flag
eg SBB AX, BX

7. Explain any 2 assembler directives & 8086
→ Assembler directives:
- ASSUME - It is used to tell name of the logical segment the assembler use for a specified segment
- ENDP - Used along with name of procedure to indicate end of procedure.

8. Difference between following instructions
→

(i)

ROL	RCL
• Used to rotate bits in a specified byte or word to the left by some no. of positions	• Used to rotate bits in specified byte or word to left by some no. of positions through the carry
• Direction is, MSB to LSB and to carry flag	• Direction is, MSB to CF & CF to LSB

(ii)

ADD	ADC
• Adds contents of source, and destination and stores result in destination	• Adds contents of source & destination as well as carry flag bit and stores result in destination
• Operation is, D = D+S	• Operation is, D = D+S+ Carry bit

(iii)

MOV	LXI
• This instruction is used to copy the data from ~~one~~ Source to destination	• This instruction stores 16-bit data into register pair designated in the operand
• Example, MOV CL, [2000H]	• Example, LXI B, FE50

(iv)

JMP	JNC
• Used to jump to the provided address to proceed to the next instruction Unconditionally	• Used to jump if no carry flag set. i.e CF=0
• Example, ~~JMP~~ JMP 008H	• Example, JNC 008H

9. Write any 2 conditional & 2 unconditional branching instruction with their function. Give Syntax with one example

→ Conditional branching instructions

(1) JNC - Jumps to given instruction if ~~no~~ carry flag is not set
Syntax - JNC <target address>
Example - JNC 4000H

(2) JC - Jump to given instruction if carry flag is set
Syntax - JC <target address>
Example - JC 4000H

Unconditional branching:

(1) CALL - Transfers execution to a subprogram or procedure
Syntax: CALL ~~PROGNAME~~ <Subprogram name>
Example - CALL FACT

(2) RET : It will return execution from procedure to next instruction after CALL instruction in calling program
Syntax: RET
Example
```
p1 PROC;
   MOV AX;
   RET;
```

C * 6 marks questions

① Explain the use of assembler directives

→ ① ASSume - Tells name of logical segment the assembler to use for a specified segment

② DB - It is used to reserve byte or bytes of memory locations in available memory. While preparing EXE file, it directs assembler to allocate specified number of bytes for data type

③ DD - It is used to declare a double word type variable or to reserve memory locations accessible as double word

④ DQ - It is used to direct assembler to reserve 4 words (8 bytes) of memory for specified variable

⑤ ENDP - It is used along with name of procedure to indicate the end of a procedure

⑥ LABEL - Used to give a name to current value in location counter

② Describe any 6 ~~ass~~ addressing modes ~~of~~ of 8086 with one example

→ ① Immediate addressing mode - Here, immediate data is a part of instruction & appears in form of successive byte/bytes

eg. MOV AX, 0050H

② Register addressing mode - Here, data is stored in register, & register is used to refer to that data

eg. MOV AX, BX

③ Based indexed addressing mode - The effective address of data is formed in this addressing mode, by adding content of base register (BX/BP) to content of an index register (SI/DI). Default segment register may be ES or DS

eg. MOV AX, [BX][SI]

④ Indexed addressing mode - Here, offset of operand is stored in one of index registers

eg MOV AX, [SI]

⑤ Register relative addressing mode - In this addressing mode data is available at an effective address formed by adding an 8 bit or 16 bit displacement with content of any one of register.
eg MOV AX, 50H [BX]

⑥ Relative based ~~addressing~~ indexed - The effective address of of data is formed in this addressing mode, by adding content of base register (BX/BP) to content of an index register (SI/DI).
~~The default~~ eg. MOV AX, 50H [BX][SI]

3. Explain 4 rotate instructions with their syntax, operation & example

→ ① SHR - Shift Right instruction shifts mentioned bits in register to right side one by one inserting the same number, Rightmost bit is stored in carry flag
~~eg. Syntax: MOV BX~~ Syntax: SHR destination, count
Example: SHR BL, 1 (Moves by 1 place)

② ROL - This rotates all bits in a specified byte or word to the left, some no. of positions. MSB is placed as a new LSB & a new CF
eg. Syntax - ROL destination, count
eg - ROL BL, 1

③ ROR - Rotates all bits to right, by no. of positions. LSB is placed as a new MSB & a new CF
Syntax - ROR destination, count
eg - ROR BL, 1

④ RCR - Rotates all bits ~~in~~ to right by certain no. of bits along with carry flag placed in new MSB & LSB in CF
eg ~~P~~ Status: ROR destination, count
eg. ROR BL, 1

4. Select assembly language for each:

i) rotate register BL right 4 times: ROR BL, 4;

ii) Multiply AL by 04H MOV AX, 04H, MUL BX

iii) Signed division of AX by BL:

iv) Move 2000h in BX register MOV BX, 2000H

v) Increment Counter of AX by 1: INC AX

vi) Rotate the contents of BX register by 4: ROR BX, 4

vii) Compare AX with BX: CMP AX, BX

viii) Add 100H to contents of AX register ADD AX, 100H

ix) Rotate the contents of AX towards left by 2 bits: ROL AX, 2

x) Transfer 1234H to DS register: MOV DS, 1234H

5. Illustrate use of any 3 branching instructions

→ i) CALL instruction is used to transfer execution to a Sub program or procedure.

It may be used as:

- NEAR CALL → Call to a procedure in same segment
- FAR CALL → Call to a procedure in different segment

2) RET - It returns execution from a procedure to next instruction after the CALL instruction in the calling program

```
e.g  p1 PROC;
       MOV AX, 8H;
       RET;          ← Return to caller, p1 ENDP
```

3) JNC - Jumps to specified address if carry flag is not set

i.e if CF = 1, JNC doesn't occur

if CF = 0, JNC occurs

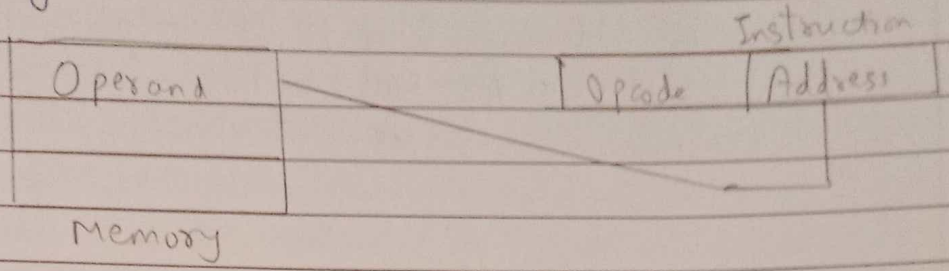
6. Describe any six addressing modes of 8086 with suitable diagram

→ (i) Direct addressing mode

In this type of addressing mode, a 16 bit memory address (offset) directly specified in the instruction as a part of it.

Example : MOV AX, [5000H]

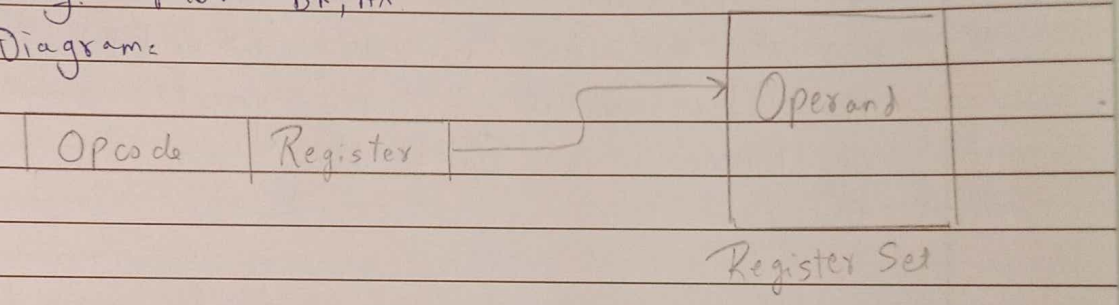
Diagram :



(2) Register addressing mode : In this type of addressing mode, the data is stored in a register & it is referred using the particular register name.

eg. MOV BX, AX

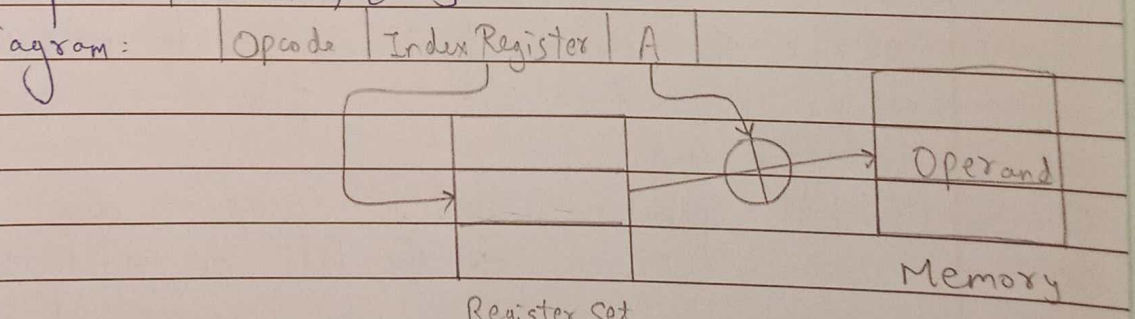
Diagram :



(3) Indexed addressing mode : In this type, offset of operand is stored in one of index registers.

Example : MOV AX, [SI]

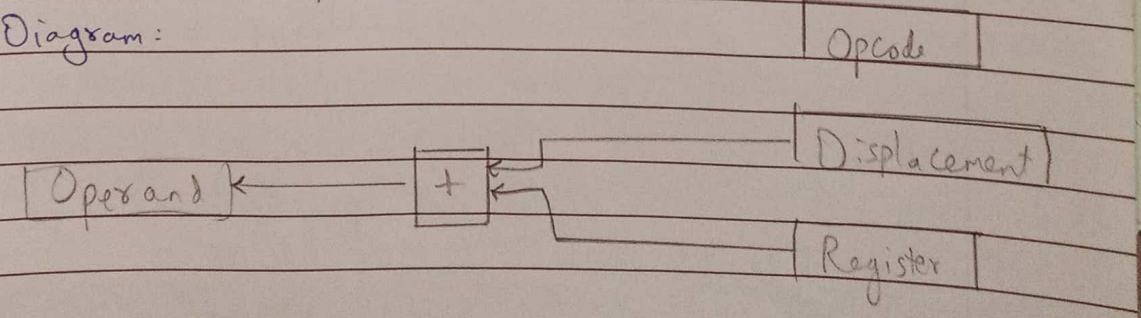
Diagram :



(4) Register relative addressing mode : In this type, data is available at an effective address formed by adding an 8 or 16 bit displacement.

eg. MOV AX, 50H [BX]

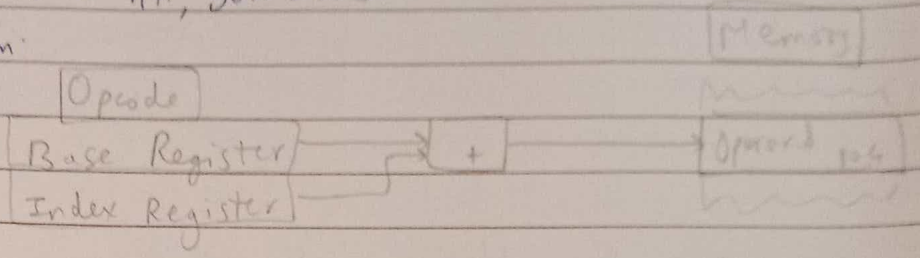
Diagram :



(5) Based indexed addressing mode:- In this type, data is available at an effective address formed by adding an 8 or 16 bit displacement

eg. MOV AX, 50H [BX]

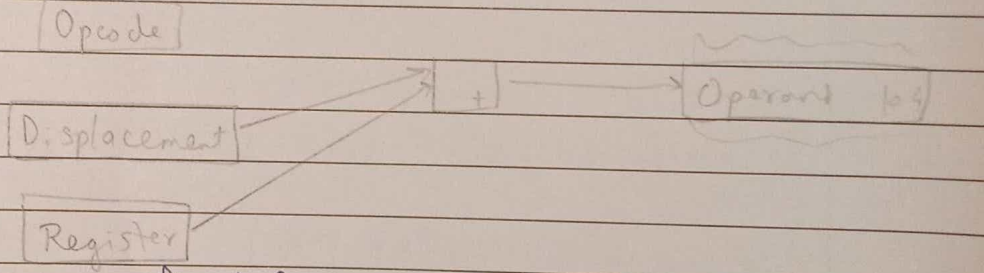
Diagram:



(6) Relative based indexed: The effective address is formed by adding an 8 or 16 bit displacement with the sum of contents of any one of the base registers (BX or BP) & any one of the index registers in a segment

eg. MOV AX 50H [BX] [SI]

Diagram:



(7) Describe the use of shift & rotate instruction as well as string instruction with the help of one relevant examples of each.

→ (1) Shift operations: These are used to move the binary data to the left or right by shifting them within register or memory location.

eg MOV BL, B7H & SAL BL, 1

(2) Rotate instruction: These instructions rotate info in register or memory either from one end to another or through carry flag.

eg. MOV BL B7H
ROL BL, 1

(3) String instructions: String instructions are used to perform different operations on strings

eg. MOV AX, 0000H / MOV DS, AX
MOV DS, AX / MOV ES, AX