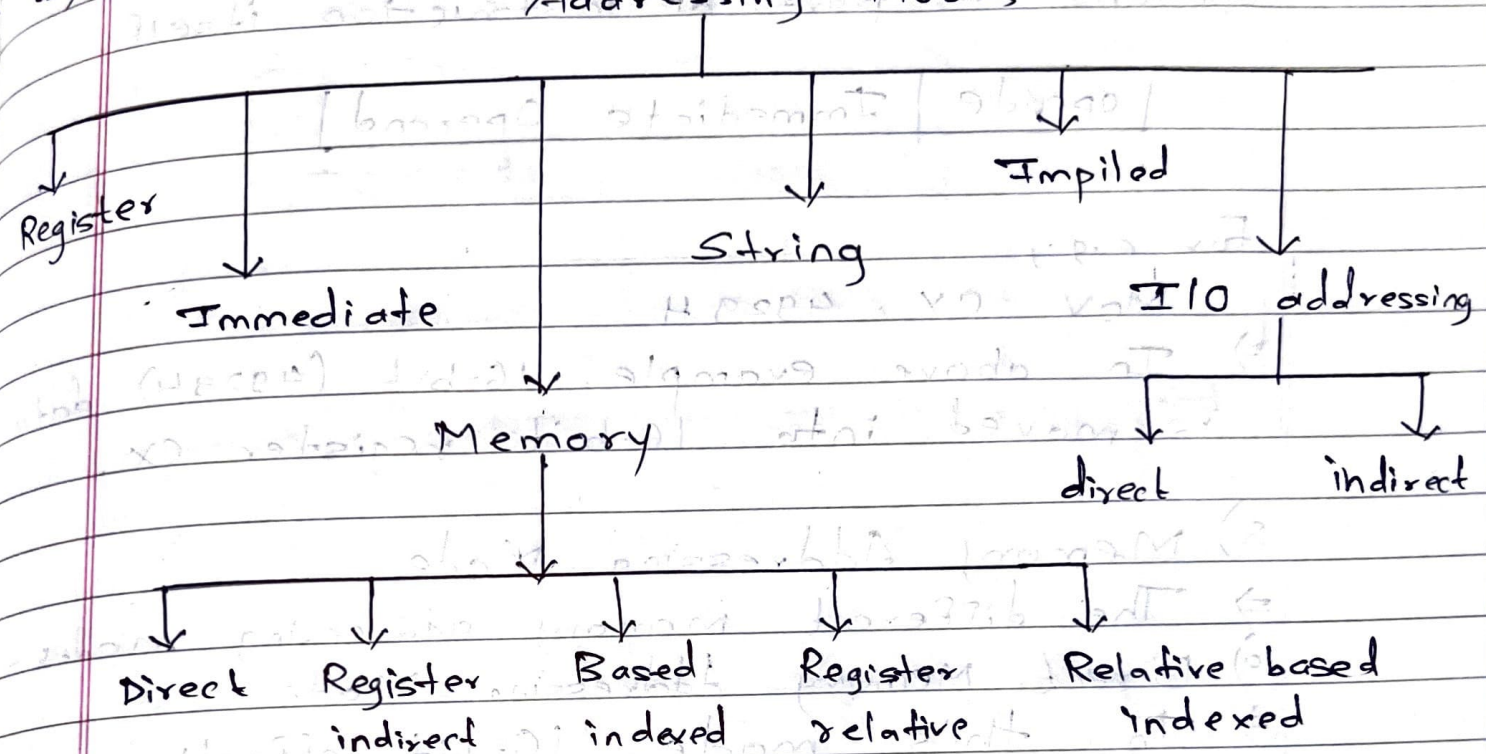
Instruction Set & Programming

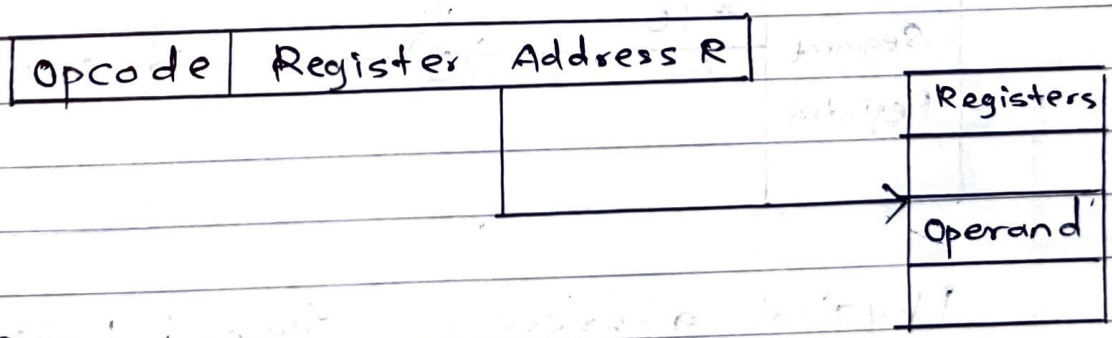
Q.1 Explain different addressing modes of 8086

⇒



1) Register Addressing Mode

⇒ In this mode, operand is stored in a register, and the name of register is specified in an instruction.



For e.g.,

MOV CX, AX

⇒ It copies the content of 16-bit register AX into 16-bit register CX

2) Immediate Addressing Mode

⇒ The addressing mode in which data operand is part of instruction itself.

opcode	Immediate Operand

For e.g.;

MOV CX, 4929H

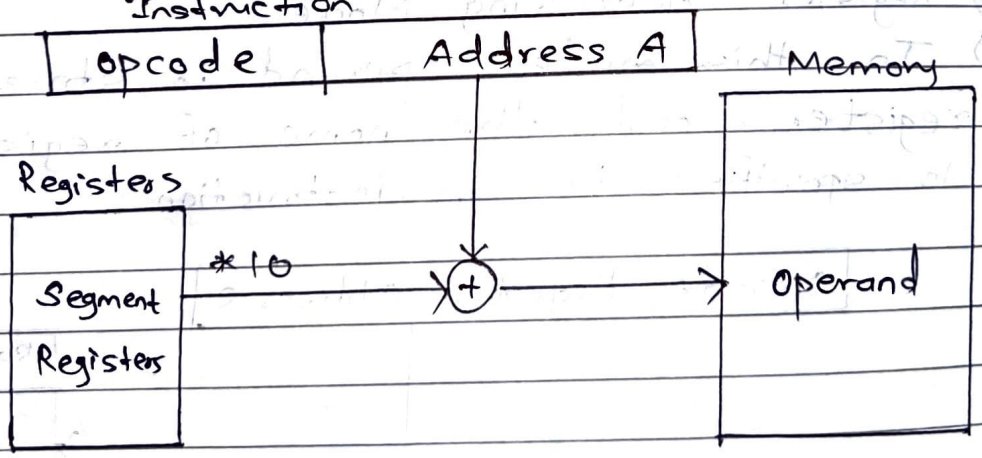
⇒ In above example, 16-bit (4929H) data is moved into 16-bit register CX.

3) Memory Addressing Mode

⇒ The different memory addressing modes-

a) Direct Memory Addressing Mode

⇒ In this mode, 16-bit effective address is directly given in the instruction.



Physical address = segment : EA

For e.g.;

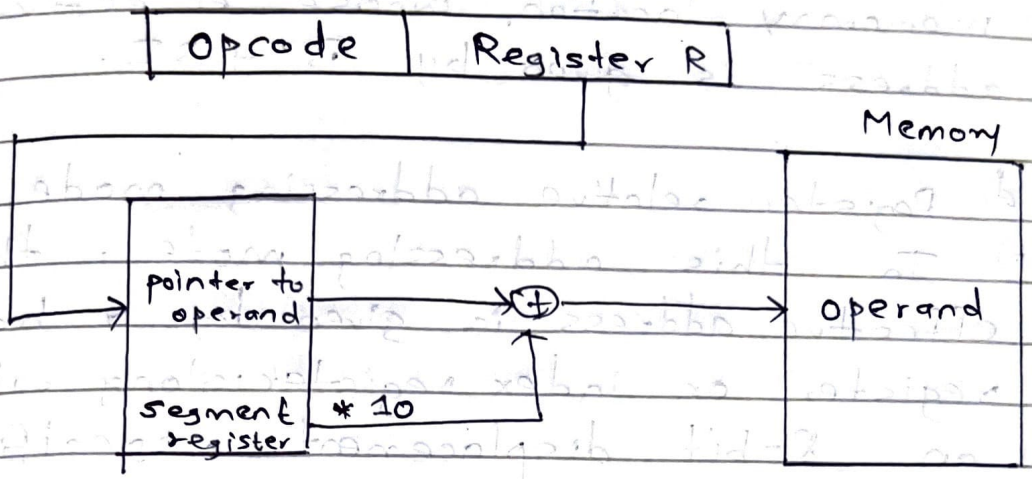
MOV [1023], AL

⇒ It copies the content of AL register to memory location whose effective address is [1023]

∴ PA = DS* 10H + 1023

b) Register indirect addressing mode

⇒ In this mode, effective address is given by a base register or index register, specified in the instruction.

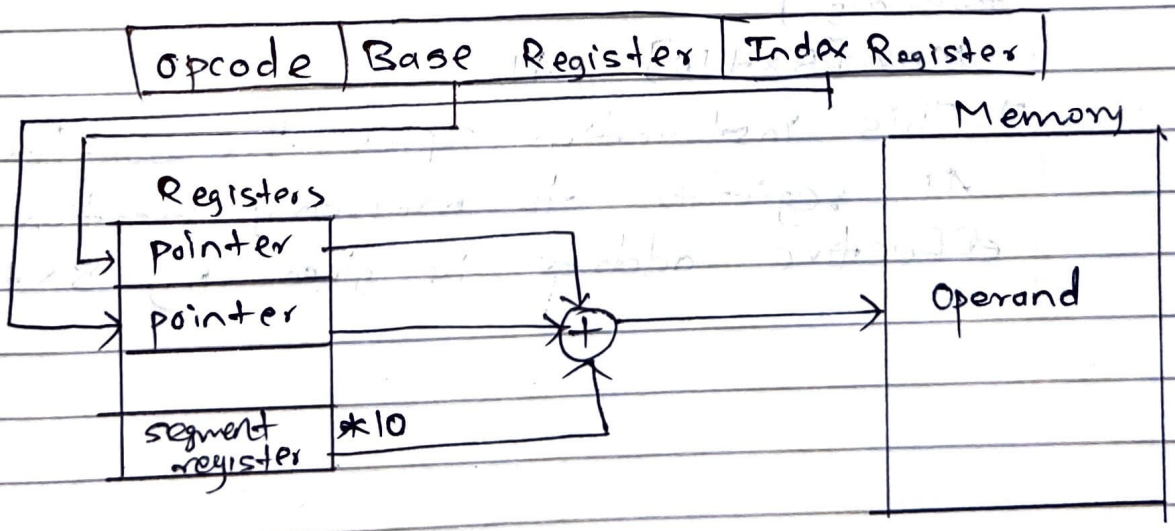


For e.g., MOV [SI], AL

⇒ The contents of AL registers are copied to Memory location whose effective address is given by SI i.e. the PA = DS * 10H + $I.

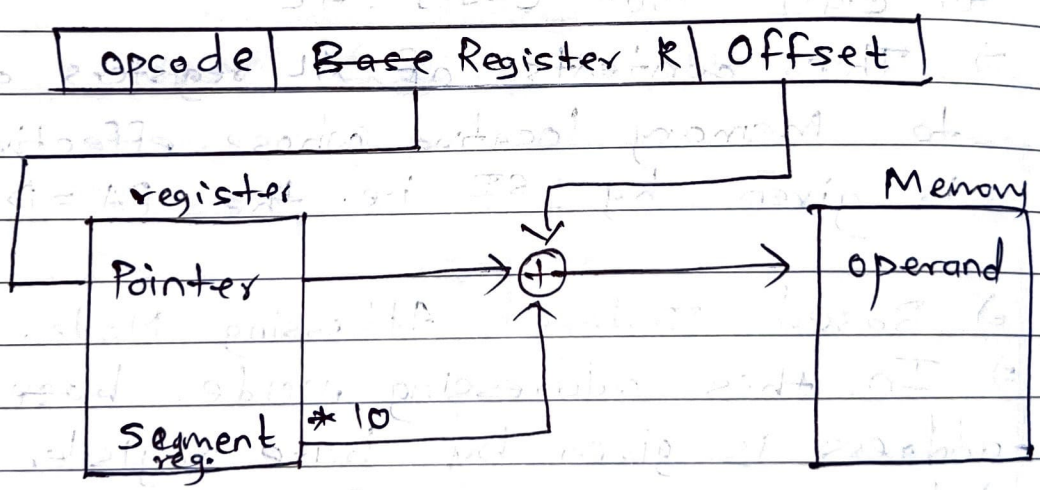
c) Based Indexed Addressing Mode.

⇒ In this addressing mode, ~~base~~ effective address is given by base register and an index register, specified in the instruction



For e.g.:
MOV [BX+SI], AL
In above example, this instruction copies the content of AL register to memory location whose effective address is given by BX+SI.

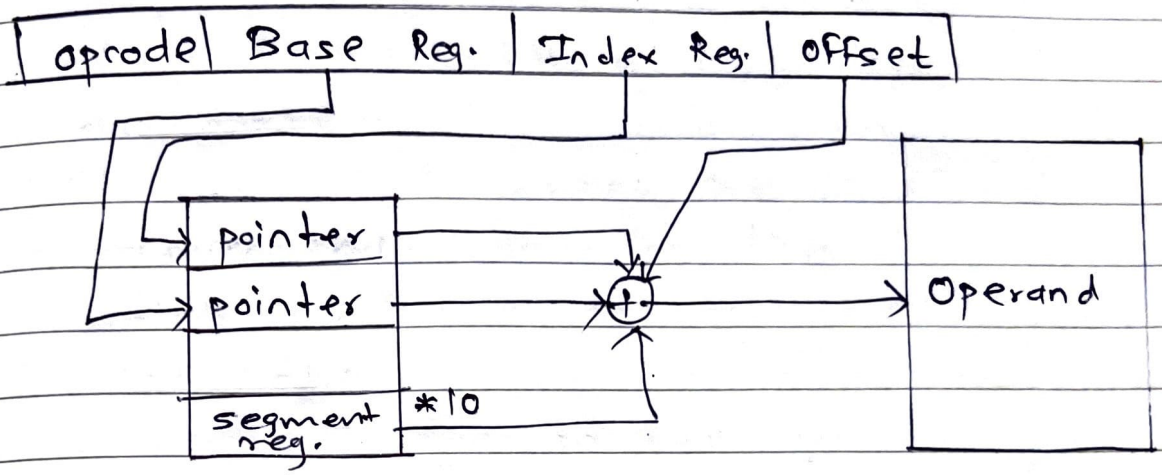
4) Register relative addressing mode
⇒ In this addressing mode, the effective address is given by a base register or index register along with an 8-bit displacement, specified in an instruction.



For e.g.:
MOV [BX+10], AL
⇒ This instruction copies the content of AL register to memory location whose effective address is given by BX+10H.

e) Relative Based Indexed Addressing.

⇒ In this addressing mode the effective address is given by base register and index register along with an 8-bit displacement, specified in an instruction.



For eg.,

MOV, CX, [BX + SI + 0400]

⇒ This instruction copies the content of AL register to memory location whose effective address is given by BX+SI+0400h

4) String Addressing Mode

⇒ String instructions use a different addressing mode wherein the pointers SI and DI along with segment registers DS and ES respectively are used to access the source and destination memory locations.

Q.2 Explain the following instructions in 8086:

1) XLAT.

⇒ i) Translate byte in AL register with byte from lookup table in memory.

ii) This instruction replaces a byte in AL register with a byte from lookup table in the memory.

iii) That is it copies the value of memory byte at location -
DS: [BX + unsigned AL]

iv) Here the content of AL register acts as index to the desired location in lookup table.

v) Segment register cannot be used as a register in this instruction.

vi) In this instruction, flags are not affected

vii) Addressing mode is implied addressing mode.

2) LAHF (Load AH register From Flags)

⇒ i) The lower byte of flag register is copied to the AH register.

ii) Before Execution -

		U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF
A0	03	0	0	0	0	1	1	0	0	1	1	0	0	0	1	1	0
AH	AL	MSB = 0C H								LSB = C6 H							

After Execution -

		U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF
C6	03	0	0	0	0	1	1	0	0	1	1	0	0	0	1	1	0
AH	AL																

Q. 3 Differentiate b/w procedure and macros.

⇒

MACRO	PROCEDURE
1) Macro definition contains the set of instruction to support modular programming	1) Procedure definition contains set of instruction which can be called repetitively which can perform a specific task.
2) It is used for small set of instruction mostly less than ten instruction.	2) It is used for large set of instructions mostly more than ten instructions.
3) In case of macro memory requirement is high.	3) In case of procedure memory requirement is less.
4) CALL and RET instructions are not required in macro.	4) CALL and RET instructions are required in procedure.
5) Execution time of macro is less as it executes faster than procedure.	5) Execution time of procedure is high as it executes slower than macro
6) In macro parameter is passed as part of statement that calls macro.	6) In procedure parameters are passed in registers and memory locations of stack.

Q.4 Write short note on mixed language programming

=> i) 'C' generates an object code that is extremely fast and compact, but it is not as fast as the object code generated by a good programmer using assembly language.

ii) It is true that time needed to write program in assembly language is much more than the time taken in Higher level language like C.

iii) However, there are special cases where a function coded in assembly language to reduce execution time.

iv) Example: Floating point math package must be coded in assembly language as it is frequently used and its execution speed will have great impact on the overall speed of the program that uses it.

v) There are also several occasions when some hardware devices need exact timing and then it is necessary to write assembly level programs to meet such strict timing restrictions.

vi) In addition, certain instructions cannot be executed in C.

vii) Example: C does not have an instruction for performing bitwise rotate operation.

viii) Here, built-In-Inline assembler is used to include assembly language routines in the C program without any need for a specific assembler.

CASE 2: Prefix keyword asm

For e.g.;

```
#include <bits/stdc++.h>

void main()
{ int a,b,c;
  cout << "Enter two no: " << endl;
  cin >> a >> b;

  asm mov ax, a;
  asm mov bx, b;
  asm add ax, bx;
  asm mov c, ax;

  cout << "The sum is : " << c;

}
```

CASE 2: Prefix the keyword asm for a function.

For eg:,

```
#include<bits/stdc++.h>

void main()
{
  int a,b,c;
  cout<<"Enter two no: ";
  cin>>a>>b;

  asm{
    mov ax,a;
    mov bx,b;
    add ax,bx;
    mov c,ax;
  }

  cout<<"The Sum: "<<c;
}
```