Lecture Six
Registers Addressing Mode

Addressing mode:-

The term addressing modes refers to the way in which the operand of an instruction is specified. Information contained in the instruction code is the value of the operand or the address of the result/operand. Following are the main addressing modes that are used on various platforms and architectures.

1) Register Addressing Modes:

   By specifying the name of the register as an operand to the instruction,You may access the contents of that register. Consider the 8086 mov instruction.

   ```
   mov     destination, source
   ```

   this instruction copies the data from the source operand to the destination operand.

   ```
   mov    Ax, bx  ; copies the value from bx into Ax
   mov    dl, al  ; copies the value from al into dl
   ```

2) Immediate Addressing Mode:

   This addressing mode transfers the source-immediate byte or word of data into the destination register or memory location.

   ```
   mov  Al, 22H
   ```

   This instruction copies a byte size 22H into register Al.

   ```
   mov  ESI, 12345678H
   ```

   this instruction copies a double-word sized 12345678H into register ESI.

3) Displacement Mode:

   This mode consists of a 16 bit constant that specifies the address of the target location. The instruction mov al, ds:[8088h] load the al register with a copy of the byte at memory location 8088h.

   Likewise, the instruction mov ds:[1234h],dl

   stores the value in the dl register to memory location 1234h:

Example: statement memory condition after implementation of this instruct when DS = 1512

MOV AL,99H

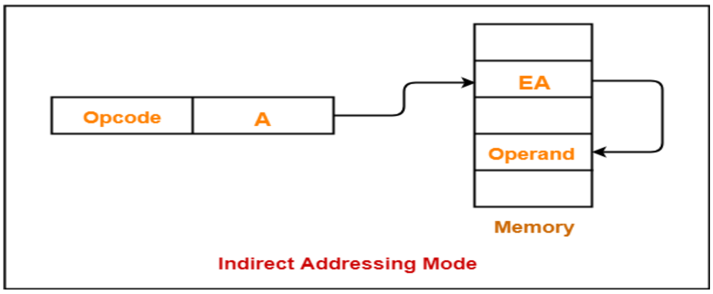
MOV [3518H],AL

Sol /

1- Ph =1512 * 10 =15120
2- 15120 +3518 =18638 99H put in this location

4) Indirect Mode:

The 80x86 CPUs let you access memory indirectly through a register using the register indirect addressing modes. There are four forms of this addressing mode on the 8086, best demonstrated by the following instructions:



mov al, [bx]
mov al, [bp]
mov al, [si]
mov al, [di]

As with the x86 [bx] addressing mode, these four addressing modes reference the byte at the offset found in the bx, bp, si, or di register,

Example: statement memory condition after implementation of this instruct when DS = 1120 , SI = 2498 , AX = 17FE ,

MOV [SI],AX

Sol /

1- Log DS = 1120
2- Ph .A = 1120 * 10 = 11200
3- 11200 + 2498 =13698

4- FE put in 13698 location but 17 put in 13698 +1=13699loc.

5) Indexed Addressing Mode:

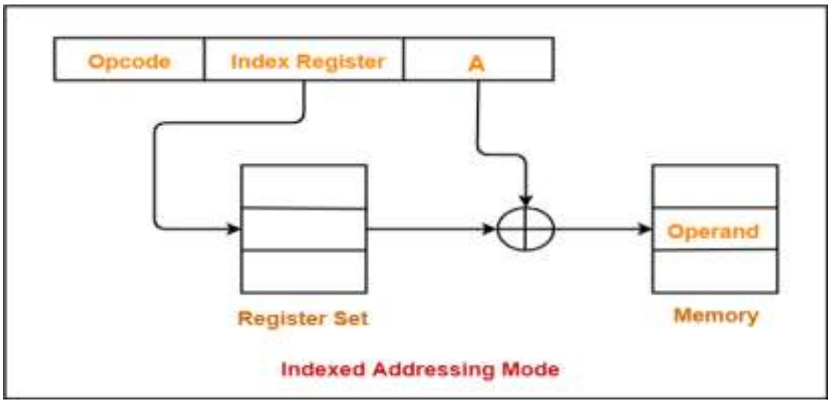
The indexed addressing modes use the following syntax:

mov al, disp[bx]
mov al, disp[bp]
mov al, disp[si]
mov al, disp[di]

If bx contains 1000h, then the instruction mov cl,20h[bx]will load cl from memory location ds:1020h. Likewise, if bp contains 2020h, mov dh,1000h[bp] will load dh from location ss:3020.

The offsets generated by these addressing modes are the sum of the constant and the specified register. The addressing modes involving bx, si, and di all use the data segment, the disp[bp] addressing mode uses the stack segment by default. As with the register indirect addressing modes, you can use the segment override prefixes to specify a different segment:

mov al, ss:disp[bx]
mov al, es:disp[bp]
mov al, cs:disp[si]
mov al, ss:disp[di]



6) Based Indexed Addressing Mode:

The based indexed addressing modes are simply combinations of the register indirect addressing modes. These addressing modes form the offset by adding together a base register (bx or bp) and an

index register (si or di). The allowable forms for these addressing modes are
mov al, [bx][si]
mov al, [bx][di]
mov al, [bp][si]
mov al, [bp][di]

Example: statement memory condition after implementation of this instruct when DS = 4500 , SS = 2000, BX =2100 , SI = 1486, DI = 8500, BD = 7814 , AX = 1512.

a) MOV [BX] +20,AX

sol /

1- Ph.A = 4500*10=45000
2- 45000 + 2100 =47100
3- 47100 + 20 = 47120

AX = $\underline{15}$ $\underline{12}$

Put 12 in location 47120

Put 15 in location 47121

b) MOV [SI]+10,AX

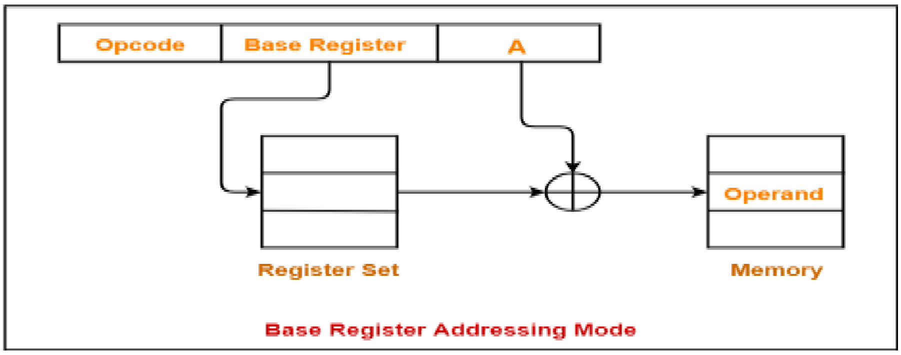
sol /

1- Ph.A= 4500*10=45000
2- 45000+1486 = 46486
3- 46486+10 = 46496 this location for 12
46496+1=46497 this location for 15

c) MOV CL,[BX][DI]+8

sol /

1- Ph.A=4500*10=45000
2- 45000+2100+8500+8= 4F608 this location for 12 and 4F609 for 15



8086 Address Modes

Type	Instruction	Source	Address Generation	Destination
1-Register	MOV AX,BX	register BX	→	register AX
2-Immediate	MOV CH3,3AH	Data 3AH	→	register CH
3-Direct	MOV [1234], AX	register AX	(DS*10H)+Displacement → 10000H + 1234	Memory 1234H
4-Indirect	MOV [BX],CL	register CL	(DS*10H)+BX → 10000+0300H	Memor 10300H
5-Index	MOV [BX+SI],BP	register BP	(DS*10H)+BX+SI → 10000H+0300H+0200H	Memory 10500H
6-Relative	MOV CL, [BX+4]	memory 10304H	(DS*10H)+BX+4 → 10000H+0300H+4	Register CL

ASSUME BX= 0300H, SI= 0200H, ARRAY= 1000H, DS= 1000H