MICROPROCESSOR ARCHITECTURE

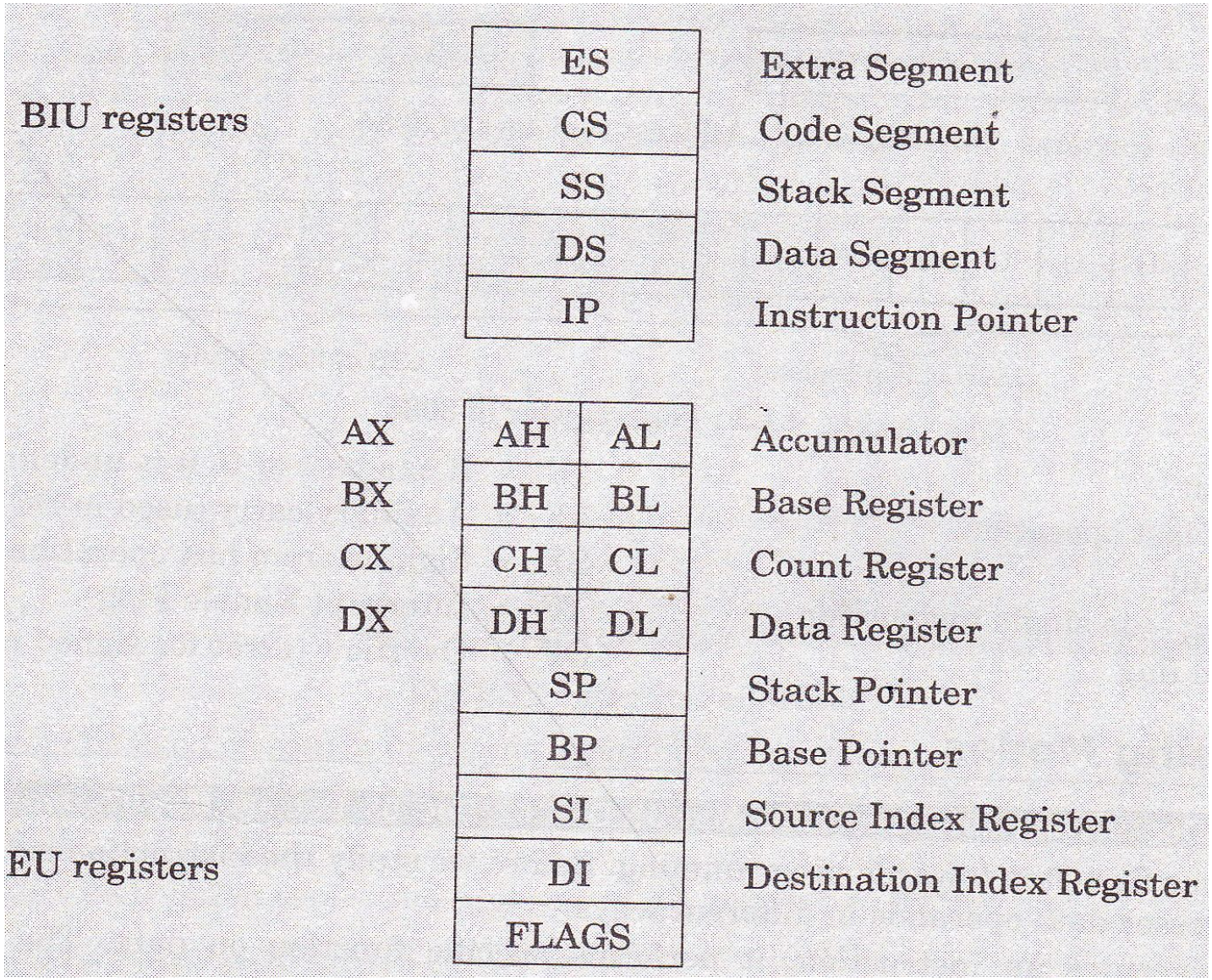
UOP S.E.COMP (SEM-I)

8086 MICROPROCESSOR ARCHITECTURE

Prof.P.C.Patil
Department of Computer Engg
Matoshri College of Engg.Nasik
pcpatil18@gmail.com.

8086 Programmers Model

8086 Programmers Model

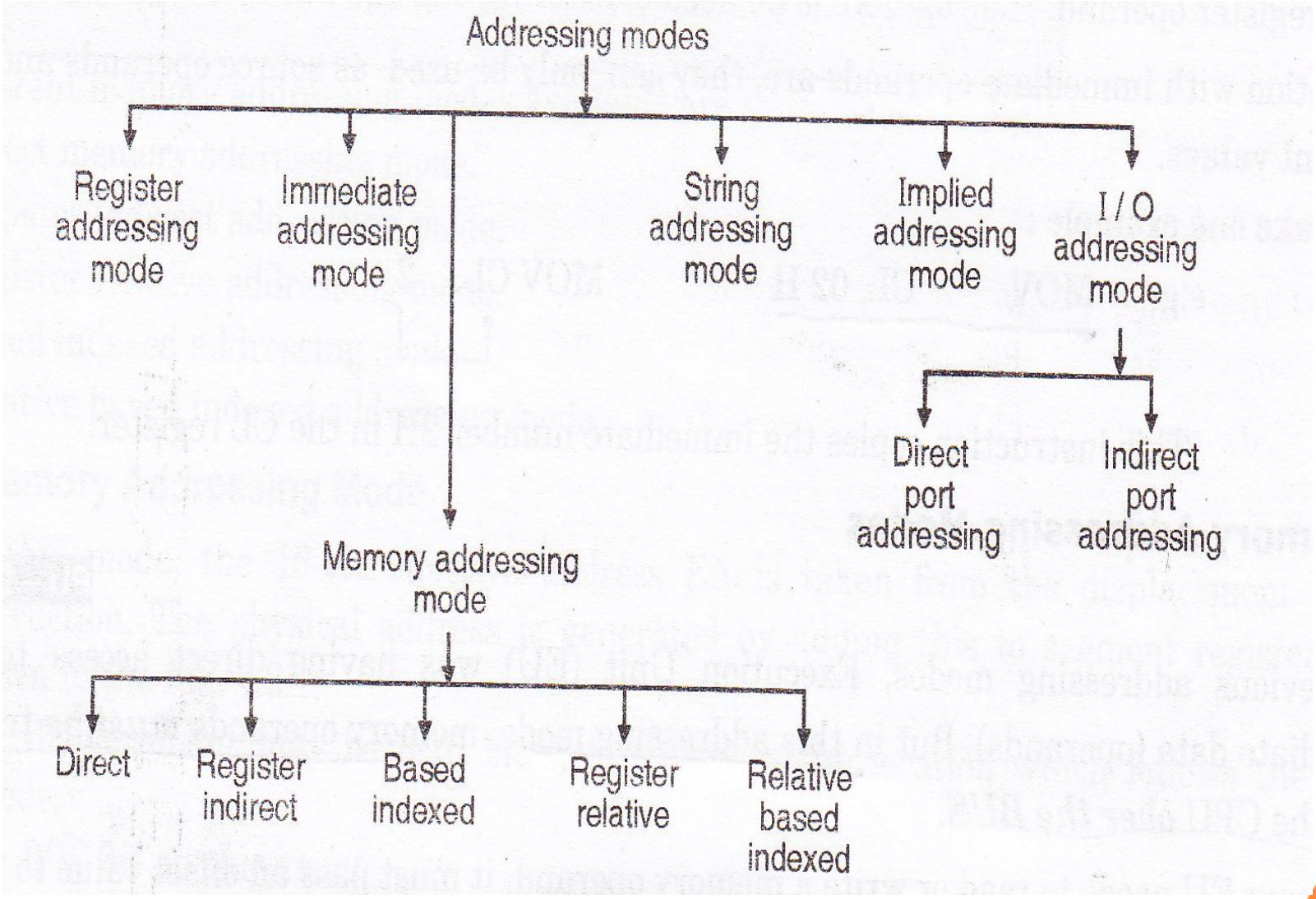


8086 Addressing Modes

8086 Addressing Modes

1. Register Addressing Mode
2. Immediate Addressing Mode
3. Memory Addressing Mode
4. String Addressing Mode
5. I/O Addressing Mode
6. Implied Addressing Mode

8086 Addressing Modes

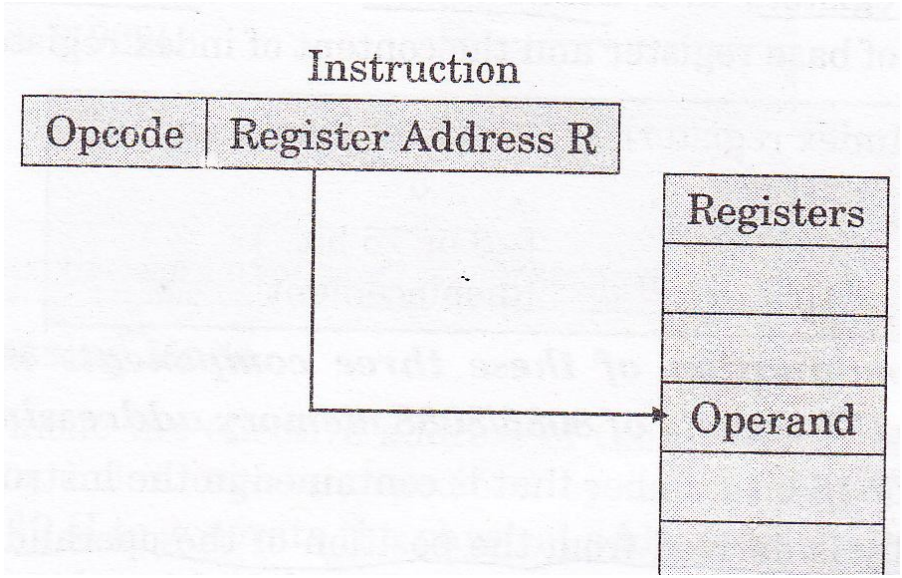


1.Register Addressing Mode

Register Addressing Mode

- Data is in register and Instruction Specifies the perticular register
- E.g

```
MOV AX,BX
```



2.Immediate Addressing Mode

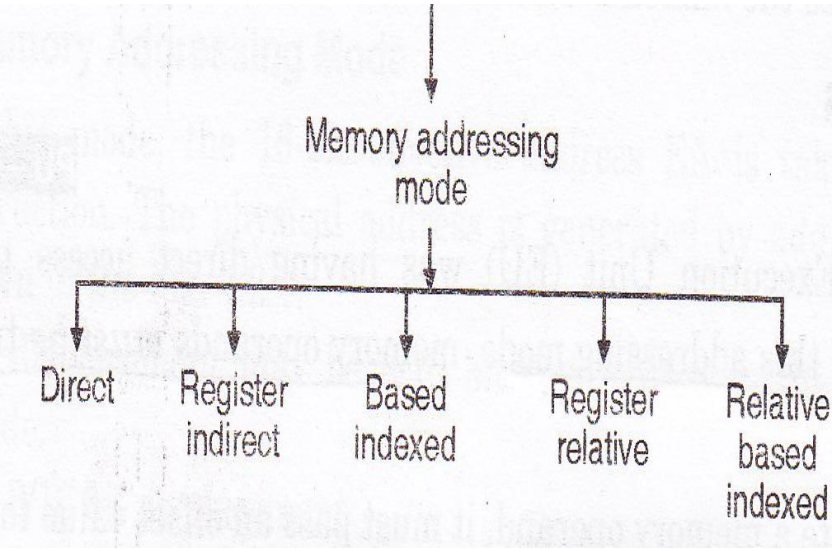
2.IMMEDIATE ADDRESSING MODE

- Immediate operand is *Constant* data contained in an *Instruction*
- *i.e.* The source operand is a part of instruction instade of register memory
- E.g

```
MOV CL,02H
```

3.Memory Addressing Mode

Memory Addressing Mode



3.1 Direct

3.2 Register Indirect

3.3 Based Indexed

3.4 Register Relative

3.5 Relative Based Indexed

EFFECTIVE ADDRESS

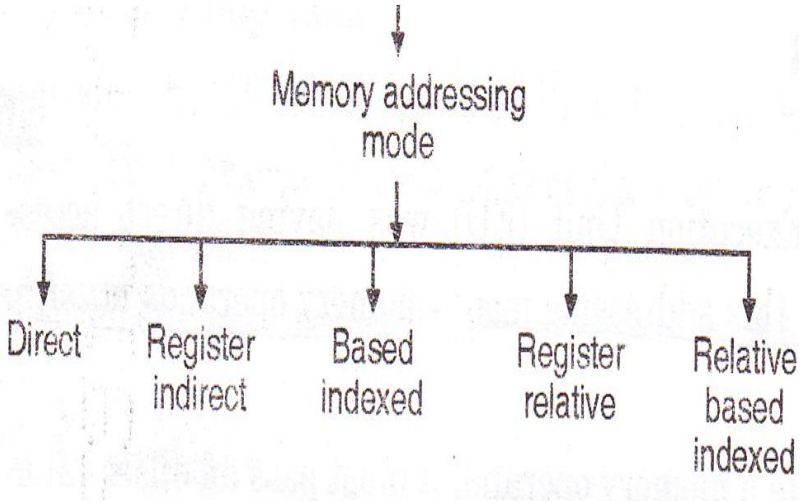
- The *offset of a memory operand* is called the operand's effective address (EA).
- Is an *unsiged 16 bit no*. That expresses the *operands distance* in byte from the *begining of the segment*
- 8086 has Base register and Index register
- So EU calculates EA by summing a *Displacement, Content of Base register and Content of Index register*.

$$\text{EA} = \{\text{Base register}\} + \{\text{Index register}\} + \{\text{8 or 16 bit displacement}\}$$

$$= \begin{Bmatrix}\text{BX}\\ \text{BP}\end{Bmatrix} + \begin{Bmatrix}\text{SI}\\ \text{DI}\end{Bmatrix} + \begin{Bmatrix}\text{8 or 16 bit}\\ \text{displacement}\end{Bmatrix}$$

Memory Addressing Mode

- Displacement is an 8 or 16 bit no
- It is generally derived from the position of operand name.
- It's value is constant.
- *Pogrammer may specify either BX or BP is to be used as Base Register*
- *Similarly either SI od DI may be specified as Index Register*



3.1 Direct Memory Addressing Mode

Memory Addressing Mode

- EA is taken from the *displacement field* of instruction.
- PA=This addr. Is added with Seg.Regi*10 H

`MOV[1023],AL`

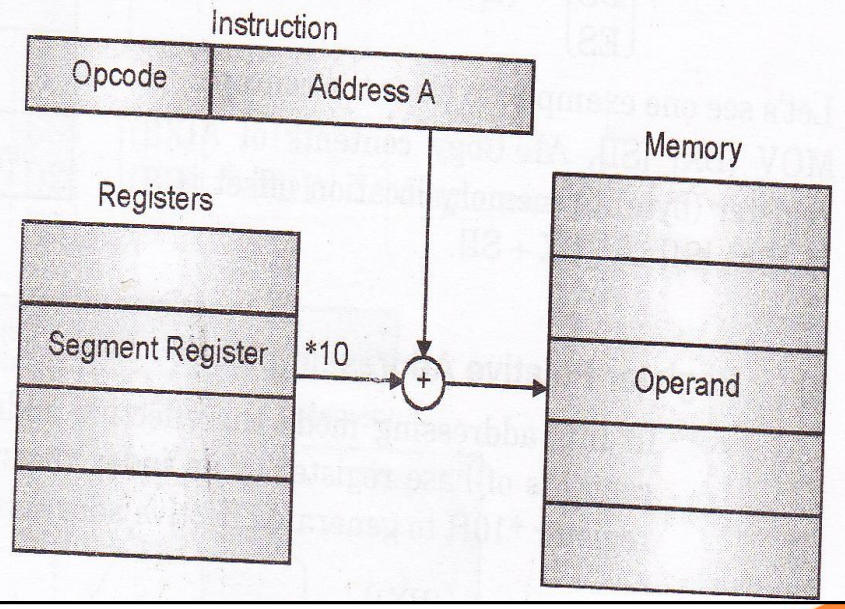
EA = 8/16 bit displacement

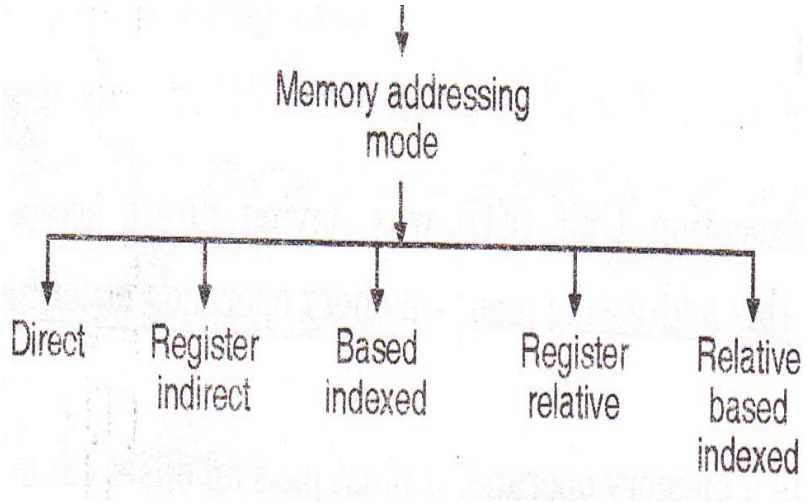
PA = segment : EA

$$PA = \begin{Bmatrix} CS \\ DS \\ ES \\ SS \end{Bmatrix} : \{ \text{Direct Address} \}$$

e.g. MOV [1023], AL

The contents of AL are copied to memory location whose offset is [1023].





3.2 Register Indirect Addressing Mode

Memory Addressing Mode

- EA of may be taken directly from *one of the base register or index register*.
- PA=This addr. Is added with Seg.Regi*10 H

```
MOV[SI],AL
```

$$EA = \begin{Bmatrix} (BX) \\ (BP) \\ (SI) \\ (DI) \end{Bmatrix} \quad PA = Segment : EA$$

$$PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \end{Bmatrix} : \begin{Bmatrix} BX \\ BP \\ SI \\ DI \end{Bmatrix}$$

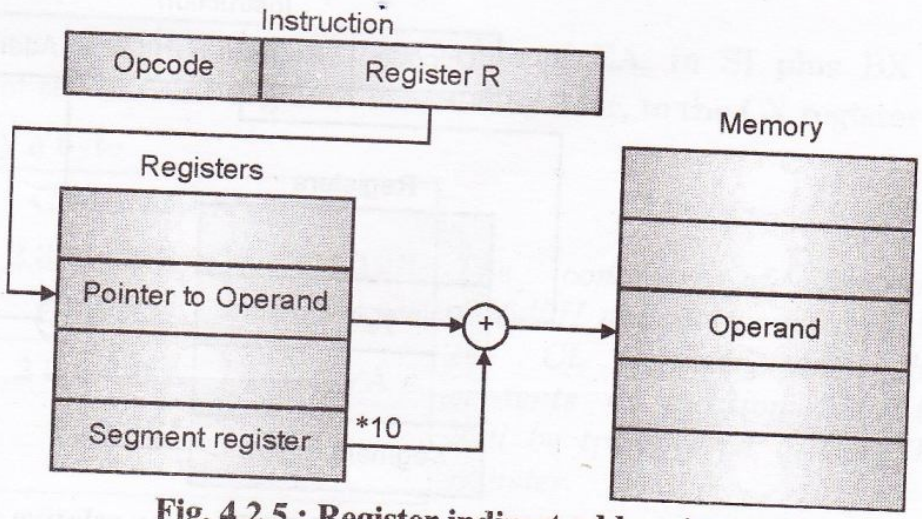
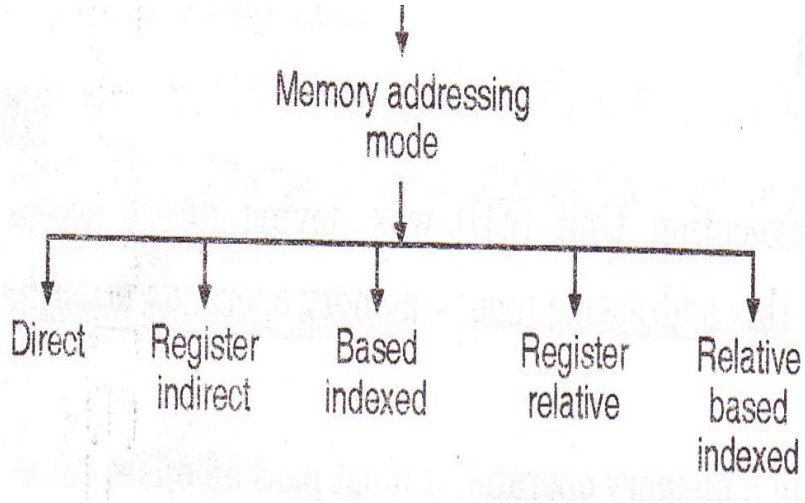


Fig. 4.2.5 : Register indirect addressing modes



3.3 BASED INDEXED ADDRESSING MODE

Memory Addressing Mode

- EA is sum of *Base register and Index register* .
- Both of which are specified by the instruction
- PA=This addr. Is added with Seg.Regi*10 H

```
MOV[BX+SI],AL
```

3) Based Indexed Addressing Mode

- In this addressing mode, the EA is sum of a base register and an index register, both of which are specified by the instruction. The sum is added to the segment register * 10 H to give effective address as shown in Fig. 4.2.6.

$$\therefore EA = \{\text{Base register}\} + \{\text{Index register}\}$$

$$= \begin{Bmatrix}(BX)\\(BP)\end{Bmatrix} + \begin{Bmatrix}(SI)\\(DI)\end{Bmatrix}$$

$$PA = \text{Segment register} : EA$$

$$= \begin{Bmatrix}CS\\SS\\DS\\ES\end{Bmatrix} : \begin{Bmatrix}(BX)\\(BP)\end{Bmatrix} + \begin{Bmatrix}(SI)\\(DI)\end{Bmatrix}$$

Let's see one example, to clear the concept.
MOV [BX] [SI], AL Copy contents of AL register (byte) to memory location offset is in [BX] [SI] i.e. [BX + SI].

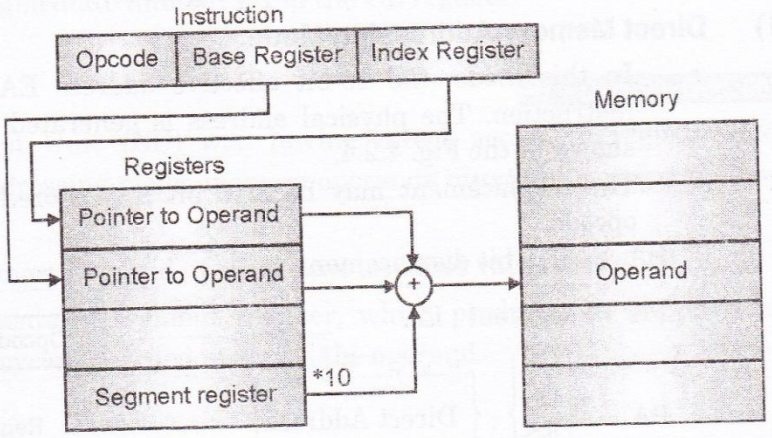
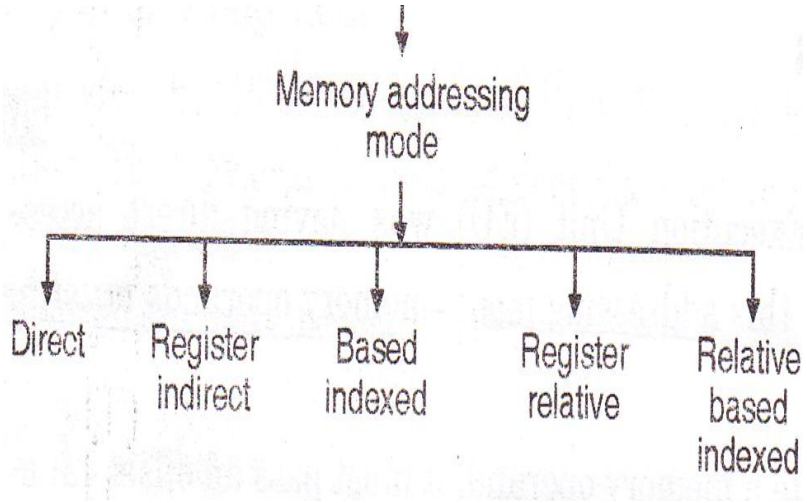


Fig. 4.2.6 : Based indexed addressing mode



3.4 Register Relative Addressing Mode

Memory Addressing Mode

- EA is Sum of 8 or 16 bit *displacement and contents of base register or an index register*
- PA=This addr. Is added with Seg.Regi*10 H

```
MOV[BX+1100],AL
```

$$EA = \begin{Bmatrix}(BX)\\(BP)\\(SI)\\(DI)\end{Bmatrix} + \begin{Bmatrix}\text{8 bitdisplacement}\\\text{(sign extended)}\\\text{16 bit displacement}\end{Bmatrix}$$

$$PA = \text{Segment} : EA = \begin{Bmatrix}CS\\ES\\DS\\SS\end{Bmatrix} : \begin{Bmatrix}(BX)\\(BP)\\(SI)\\(DI)\end{Bmatrix} + \begin{Bmatrix}\text{8/16 bit}\\\text{offset}\end{Bmatrix}$$

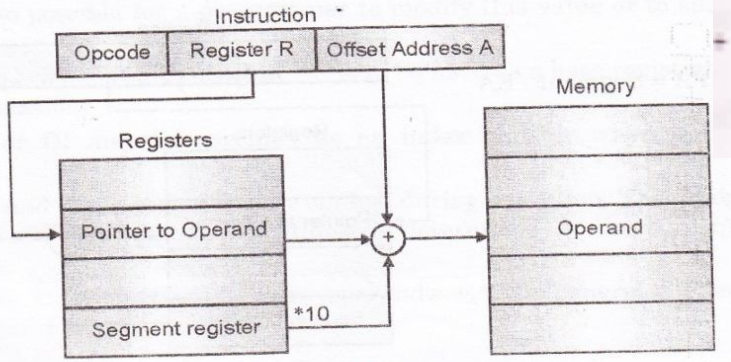
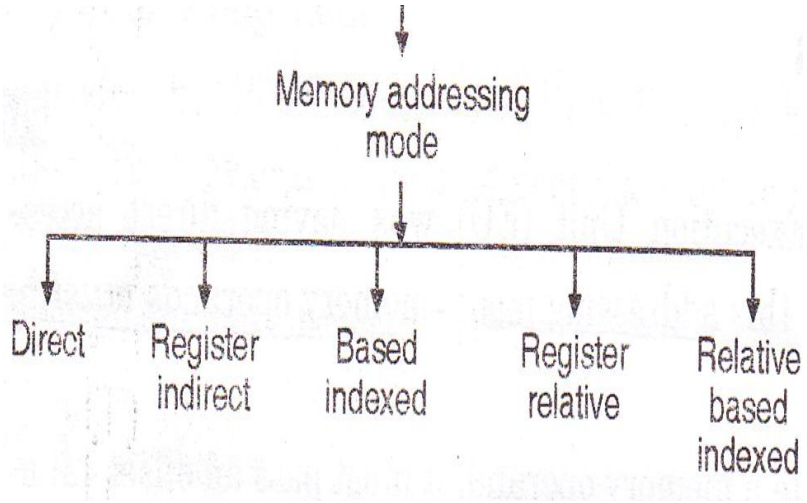


Fig. 4.2.7 : Register relative addressing mode

Generation of EA			Generation of PA		
1000H	→	[BX]	23140H	→	[DS]
- 1100H	→	[SI]	+ 2500H	→	EA
- 0400H	→	Displacement	25640H	→	PA
2500H	→	EA			



3.5 Relative Based Indexed Mode

Memory Addressing Mode

- EA is Sum of a *Base register* ,an Index Register and *Displacement* .
- PA=This addr. Is added with Seg.Regi*10 H

```
MOV CX,[BX+SI+0400]
```

$$EA = \{\text{Base register}\} + \{\text{Index register}\} + \left\{\begin{array}{c}\text{8 bit displacement}\\ \text{(sign extended)}\\ \text{16 bit displacement}\end{array}\right\}$$

$$= \left\{\begin{array}{c}(BX)\\(BP)\end{array}\right\} + \left\{\begin{array}{c}(SI)\\(DI)\end{array}\right\} + \left\{\begin{array}{c}\text{8/16 bit}\\ \text{displacement}\end{array}\right\}$$

$$PA = \text{Segment register} : EA = \left\{\begin{array}{c}CS\\SS\\DS\\ES\end{array}\right\} : \left\{\begin{array}{c}(BX)\\(BP)\end{array}\right\} + \left\{\begin{array}{c}(SI)\\(DI)\end{array}\right\} + \left\{\begin{array}{c}\text{8/16 bit}\\ \text{displacement}\end{array}\right\}$$

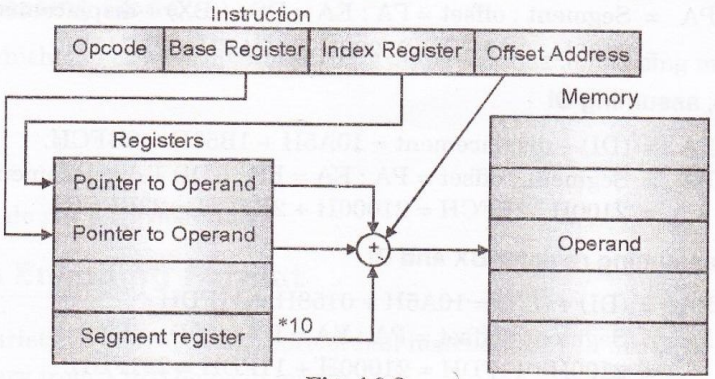


Fig. 4.2.8

Generation of EA			Generation of PA		
1000H	→	[BX]	23140H	→	[DS]
+ 1100H	→	[SI]	+ 2500H	→	EA
+ 0400H	→	Displacement	25640H	→	PA
2500H	→	EA			

The contents of location 25640H will be transferred to the CL register and the contents of location 25641H will be transferred to the CH register.