# Addressing modes of 8086

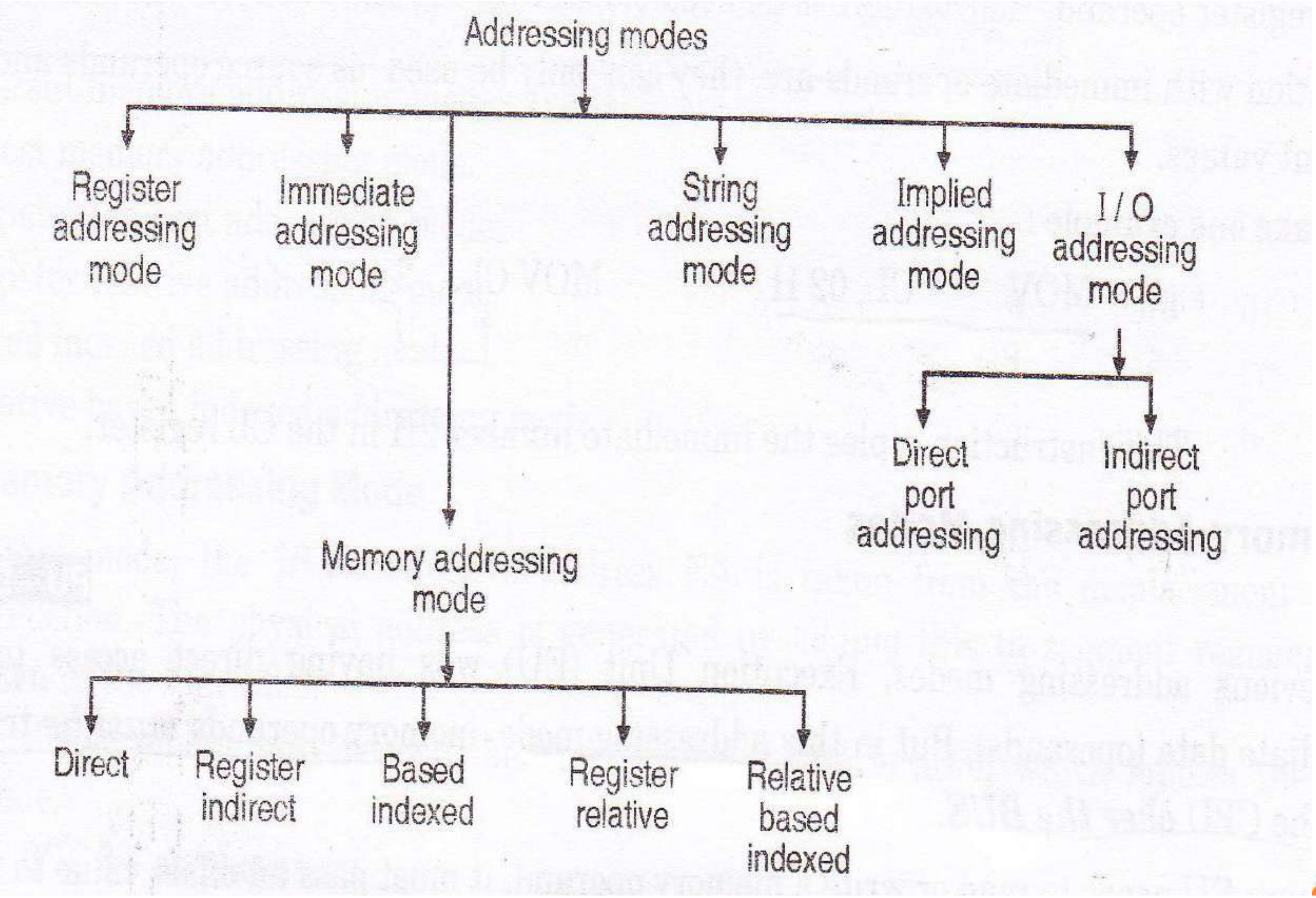
**Prof. Poonam Pangarkar**

Department of CSE – Data Science

# 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

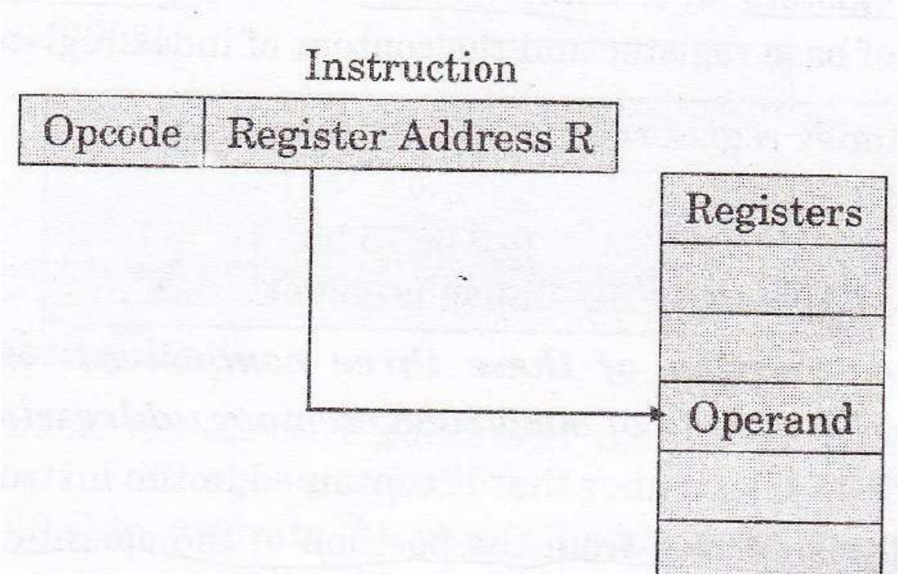


# 1.Register Addressing Mode

# Register Addressing Mode

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

```
MOV AX,BX
```



# 2.Immediate Addressing Mode

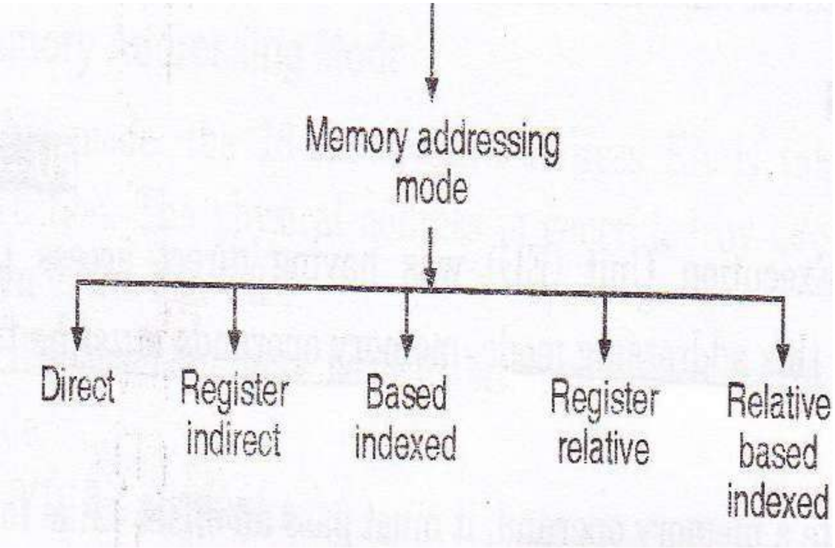
# 2.IMMEDIATE ADDRESSING MODE

- 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

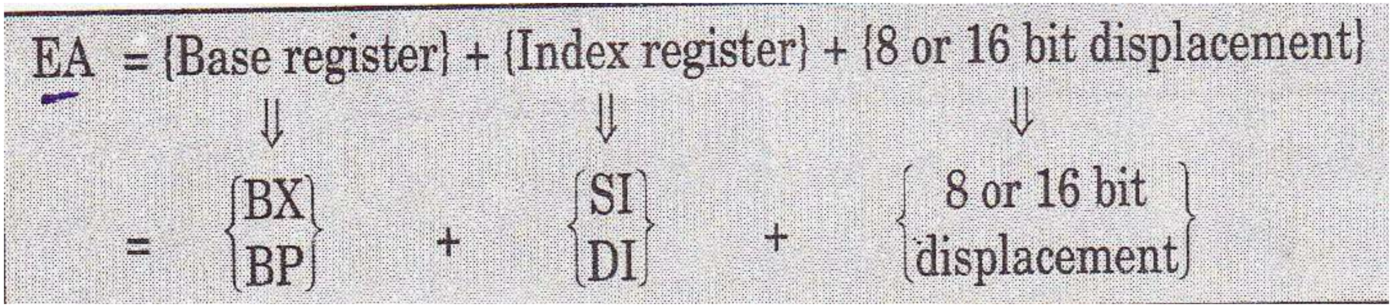


1. Direct
2. Register Indirect
3. Based Indexed
4. Register Relative
5. Relative Based Indexed

# Memory Addressing Mode

## 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*.



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

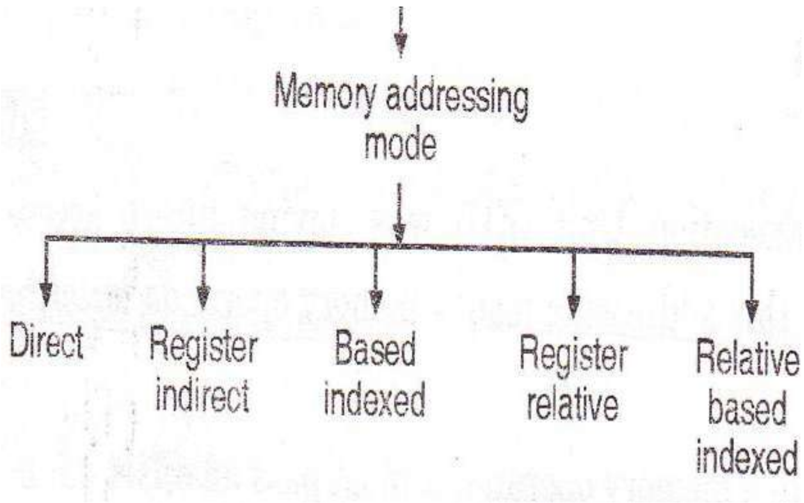
$$\Downarrow \qquad\qquad \Downarrow \qquad\qquad \Downarrow$$

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

# 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*

# Memory Addressing Mode

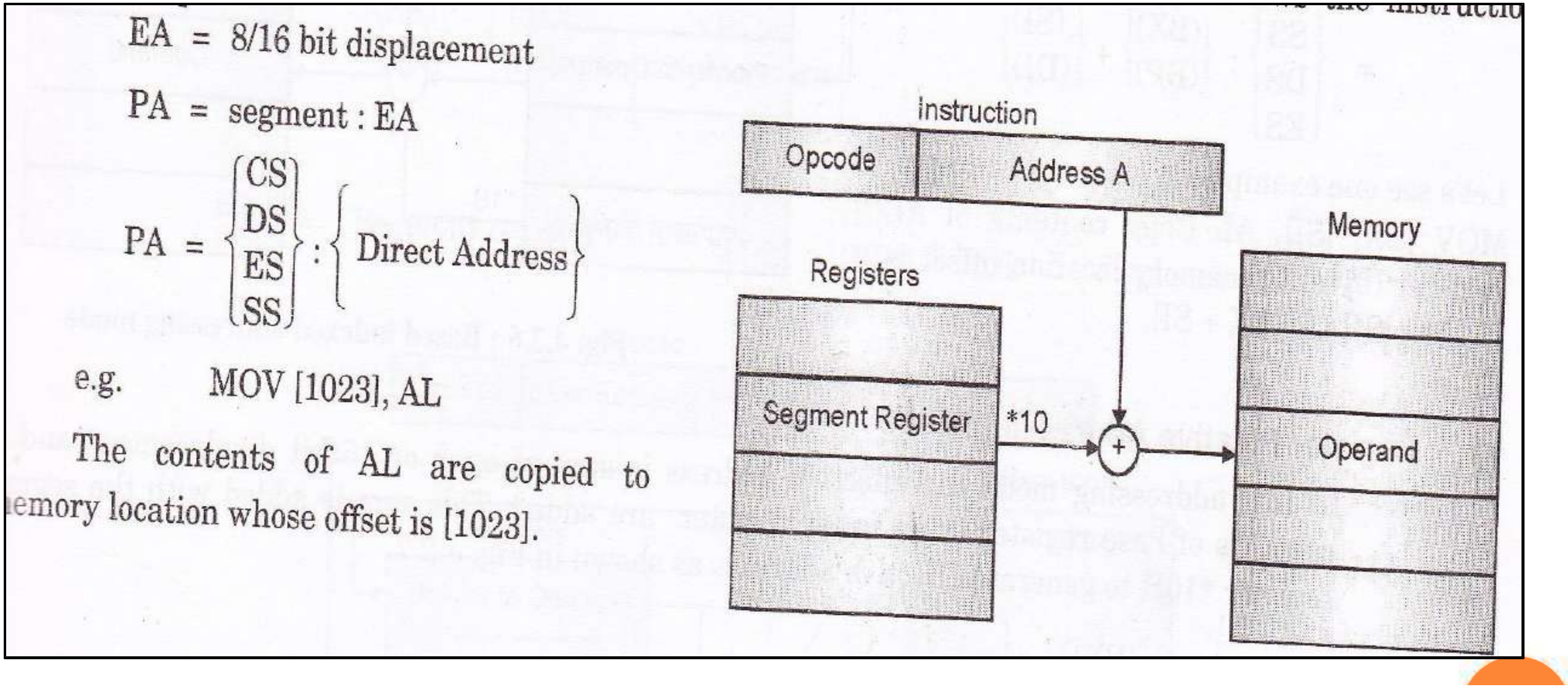


## 3.1 Direct Memory Addressing Mode

# 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

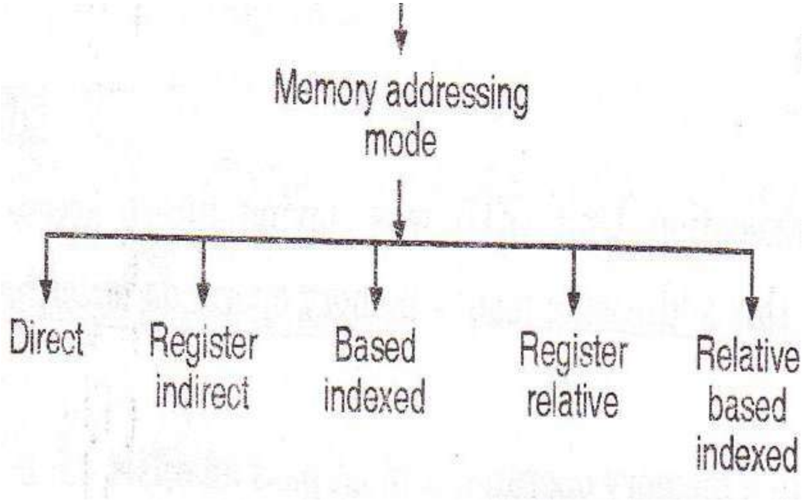
$$PA = \left\{\begin{array}{c} CS \\ DS \\ ES \\ SS \end{array}\right\} : \left\{ \text{Direct Address} \right\}$$

e.g. MOV [1023], AL

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

Instruction
Opcode
Address A
Registers
Segment Register
*10
+
Memory
Operand

# Memory Addressing Mode



## 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
```

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

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

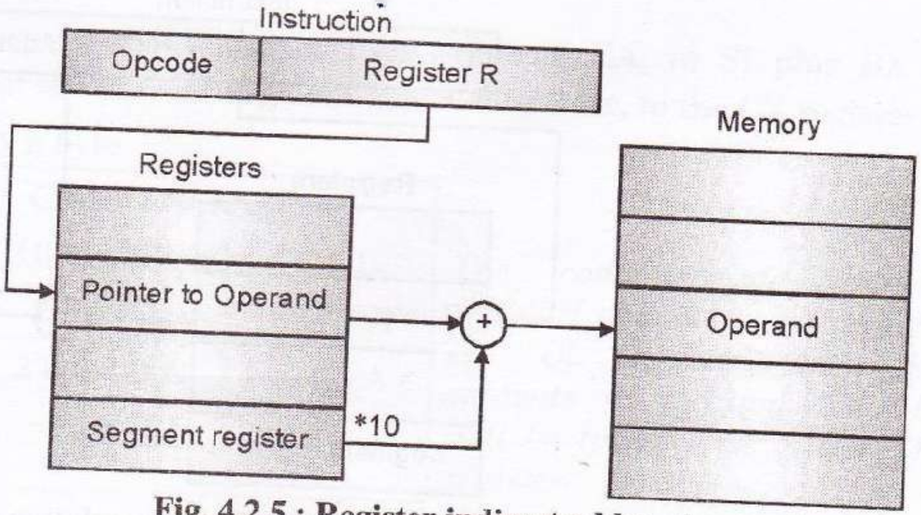
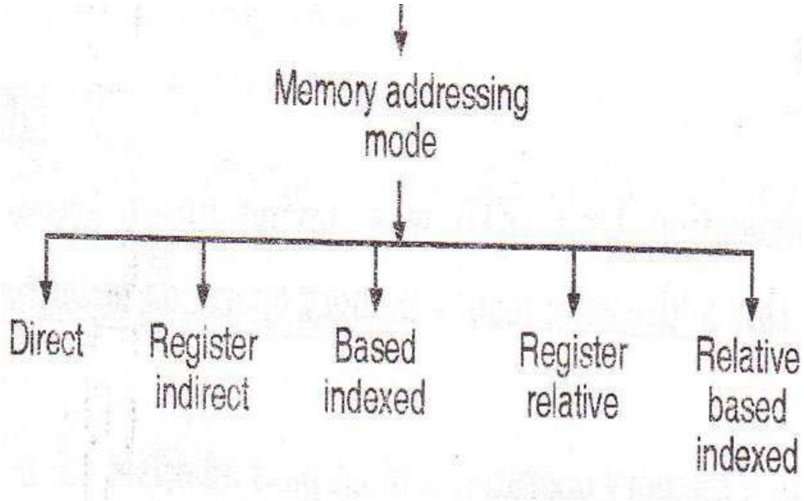


Fig. 4.2.5 : Register indirect addressing modes

# Memory Addressing Mode

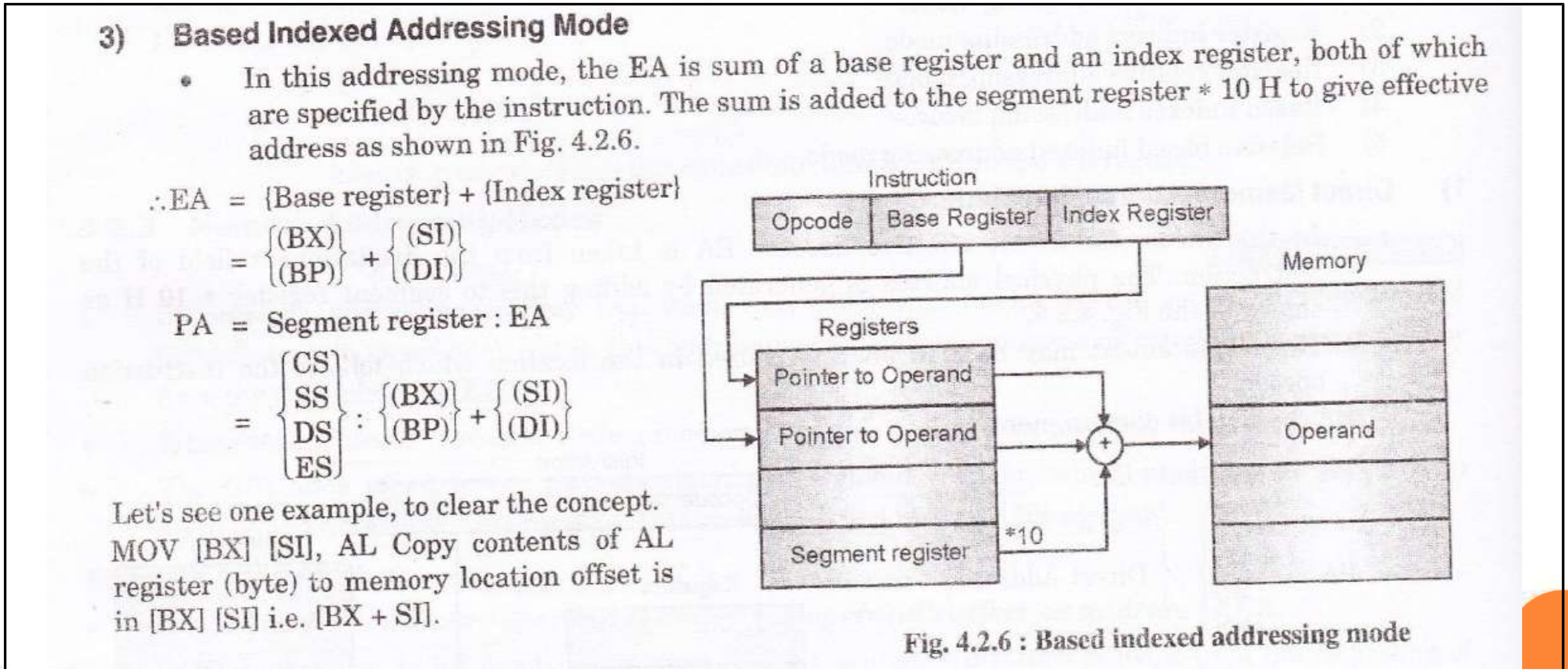


## 3.3 Based Indexed Addressing Mode

# 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.

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

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

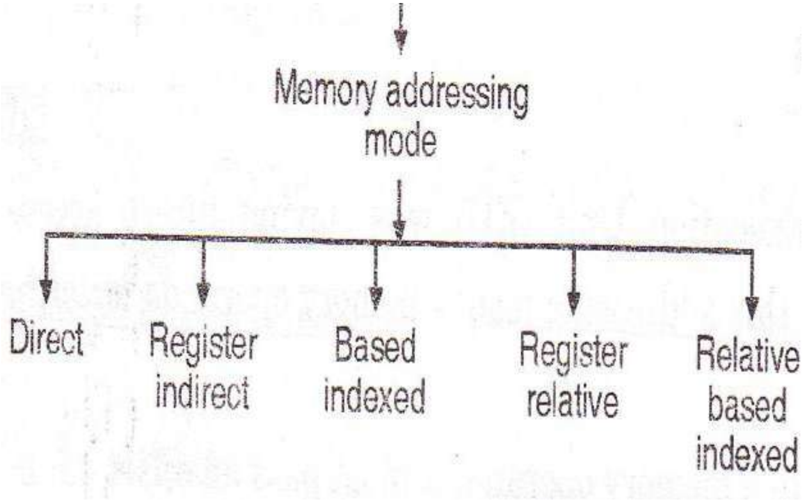
$$\text{PA} = \text{Segment register : EA}$$

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

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

# Memory Addressing Mode

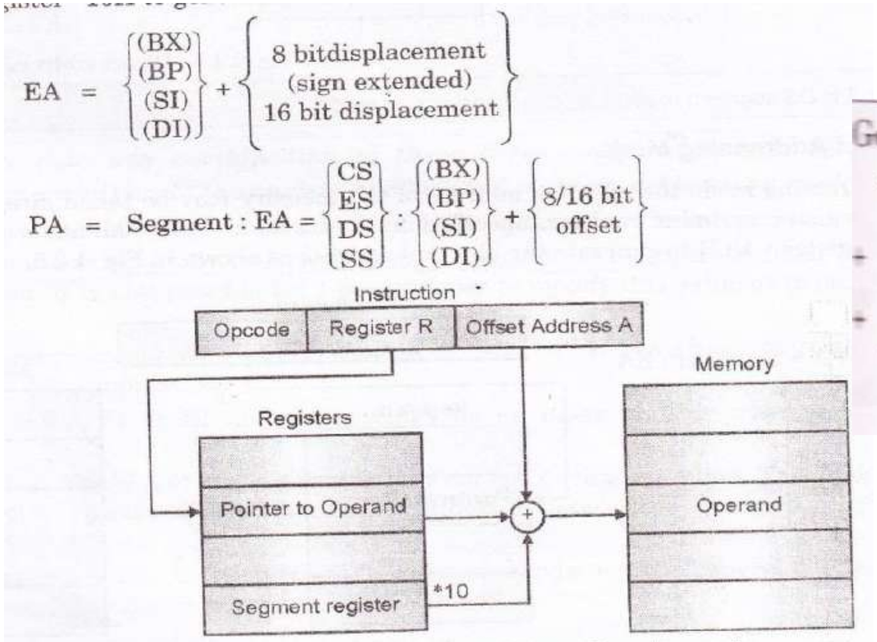


## 3.4 Register Relative Addressing Mode

# 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

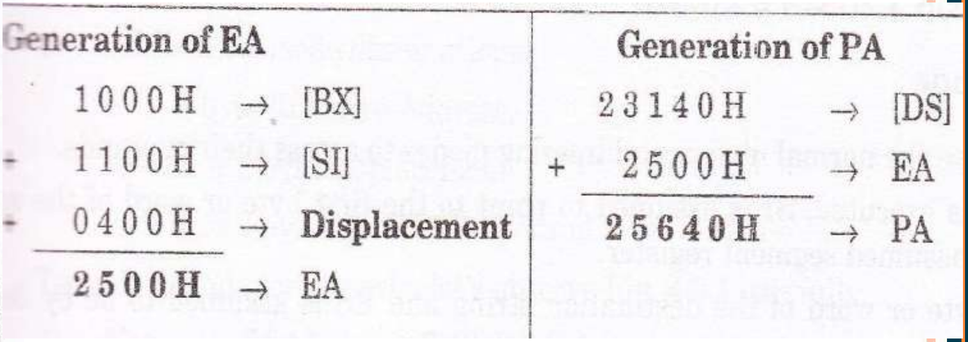
```
MOV[BX+1100],AL
```



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

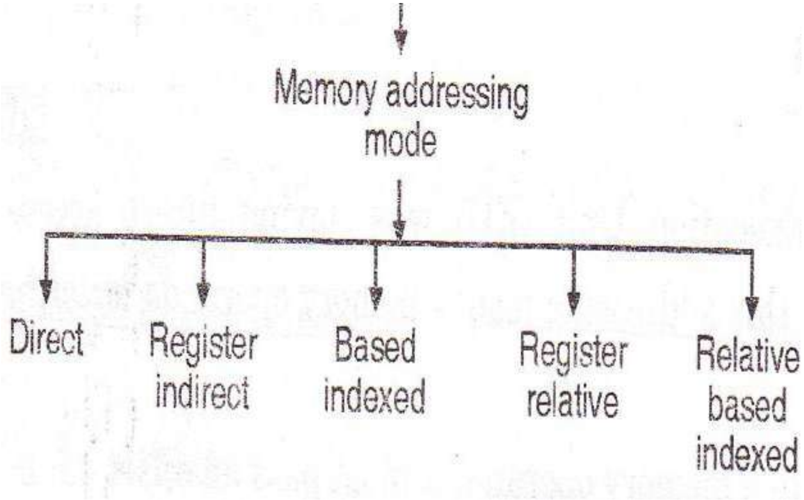
$$PA = \text{Segment} : EA = \begin{Bmatrix} CS \\ ES \\ DS \\ SS \end{Bmatrix} : \begin{Bmatrix} (BX) \\ (BP) \\ (SI) \\ (DI) \end{Bmatrix} + \begin{Bmatrix} \text{8/16 bit} \\ \text{offset} \end{Bmatrix}$$

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 | | | | |

# Memory Addressing Mode



## 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]
```

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

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

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

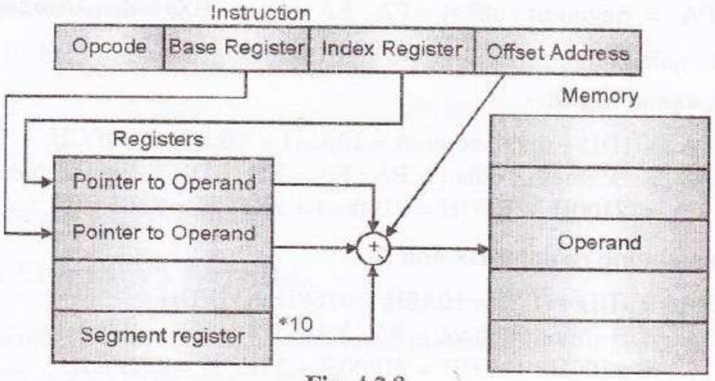
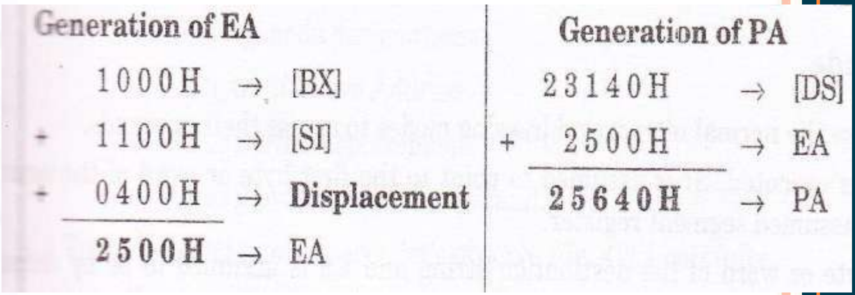


Fig. 4.2.8



| 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.*