ADDRESSING MODES OF 8086

8086 provides different addressing modes for Data, Program and Stack Memory.

ADDRESSING MODES FOR DATA MEMORY {IMP}

I IMMEDIATE ADDRESSING MODE

In this mode the **operand** is specified in the **instruction** itself.
Instructions are **longer** but the **operands** are **easily identified**.

```
Eg:   MOV CL, 12H        ; Moves 12 immediately into CL register
      MOV BX, 1234H      ; Moves 1234 immediately into BX register
```

II REGISTER ADDRESSING MODE

In this mode **operands** are specified using **registers**.
Instructions are **shorter** but **operands cant** be **identified** by looking at the instruction.

```
Eg:   MOV CL, DL         ; Moves data of DL register into CL register
      MOV AX, BX         ; Moves data of BX register into AX register
```

III DIRECT ADDRESSING MODE

In this mode **address** of the operand is directly specified **in the instruction**.
Here **only** the **offset address is specified**, the segment being indicated by the instruction.

```
Eg:   MOV CL, [4321H]    ; Moves data from location 4321H in the data
                         ; segment into CL
                         ; The physical address is calculated as
                         ; DS * 10H + 4321
                         ; Assume DS = 5000H
                         ; ∴ P A= 50000 + 4321 = 54321H
                         ; ∴ CL ← [54321H]

Eg:   MOV CX, [4320H]    ; Moves data from location 4320H and 4321H
                         ; in the data segment into CL and CH resp.
```

IV INDIRECT ADDRESSING MODES

REGISTER INDIRECT ADDRESSING MODE

In this mode the µP uses any of the 2 **base registers** BP, BX or any of the two index registers SI, DI to provide the offset **address** for the data byte.

The segment is indicated by the Base Registers:
BX -- Data Segment, BP --- Stack Segment

```
Eg:   MOV CL, [BX]      ; Moves a byte from the address pointed by BX in Data
                        ; Segment into CL.
                        ; Physical Address calculated as DS * 10H + BX

Eg:   MOV [BP], CL      ; Moves a byte from CL into the location pointed by BP in
                        ; Stack Segment.
                        ; Physical Address calculated as SS * 10H + BP
```

REGISTER RELATIVE ADDRESSING MODE

In this mode the operand address is calculated using one of the **base registers** and a **8-bit** or a **16-bit displacement.**

```
Eg:MOV CL, [BX+4]       ; Moves a byte from the address pointed by BX+4 in
                        ; Data Seg to CL.
                        ; Physical Address: DS * 10H + BX + 4H

Eg:   MOV 12H [BP], CL  ; Moves a byte from CL to location pointed by BP+12H in
                        ; the Stack Seg.
                        ; Physical Address: SS * 10H + BP + 12H
```

BASE INDEXED ADDRESSING MODE

Here, operand address is calculated as **Base register plus** an **Index** register.

```
Eg:   MOV CL, [BX+SI]   ; Moves a byte from the address pointed by BX+SI
                        ; in Data Segment to CL.
                        ; Physical Address: DS * 10H + BX + SI

Eg:   MOV [BP+DI], CL   ; Moves a byte from CL into the address pointed by
                        ; BP+DI in Stack Segment.
                        ; Physical Address: SS * 10H + BP + DI
```

BASE RELATIVE PLUS INDEX ADDRESSING MODE

In this mode the address of the operand is calculated as **Base** register **plus Index** register **plus 8-bit** or **16-bit displacement**.

Eg: **MOV CL, [BX+DI+20]** *; Moves a byte from the address pointed by*
; BX+SI+20H in Data Segment to CL.
*; Physical Address: DS * 10_H + BX + SI+ 20H*

Eg: MOV [BP+SI+2000], CL *; Moves a byte from CL into the location pointed by*
; BP+SI+2000H in Stack Segment.
*; Physical Address: SS * 10_H + BP+SI+2000H*

V IMPLIED ADDRESSING MODE

In this addressing mode the operands are implied and are hence not specified in the instruction.
#Please refer Bharat Sir's Lecture Notes for this ...

Eg: **STC** *; Sets the Carry Flag.*

Eg: CLD *; Clears the Direction Flag.*

Important points for understanding addressing modes…

1) *Anything given in square brackets will be an Offset Address also called Effective Address.*

2) *MOV instruction by default operates on the Data Segment; unless specified otherwise.*

3) *BX and BP are called Base Registers.*
BX holds Offset Address for Data Segment.
BP holds Offset Address for Stack Segment.

4) *SI and DI are called Index Registers*

5) *The Segment to be operated is decided by the Base Register and NOT by the Index Register.*