Chapter - 3
Addressing Modes of 8086

CSE-231: MICROPROCESSOR & ASSEMBLY LANGUAGE

REFERENCES:

1. Assembly language programming and organization of the IBM PC – Charles Marut – chapter 1 (sec 1.1.2, 1.2), chapter 3.

2. Microprocessors and interfacing programming and hardware, second edition, D. V. Hall – chapter 2

Before Starting this lecture
You must know

- ***How to calculate physical address***
  - Covered in Week 2- Lecture 1
- ***How to do addition and subtraction in Hexadecimal***
  - Covered in Number System Basics Video
- ***How MOV instruction works***
  - Will be covered in this Lecture

How MOV Instructions Works

MOV operand1, operand2

MOV Destination, Source

Source: Register, Memory Location and immediate values (in any number system)

Destination: Register and Memory Location

MOV instruction copies the content of sources and pastes into the destination. Source content is not updated/lost, but destination content is overwritten, previous value will be lost.

ADDRESSING MODES OF 8086

- The different ways in which a source operand is denoted in an instruction is known as **addressing modes**.
- There are 8 different addressing modes in 8086 assembly programming –

1. Immediate
2. Direct
3. Register
4. Register Indirect
5. Indexed
6. Register Relative
7. Based Indexed
8. Relative Based Indexed

ADDRESSING MODES OF 8086 (1)

1. ***Immediate***: In this type of addressing, immediate data is a part of instruction, and appears in the form of successive byte or bytes.

   Example: MOV AX, 0005H In the above example, 0005H is the immediate data.

   The immediate data may be 8-bit or 16-bit in size.

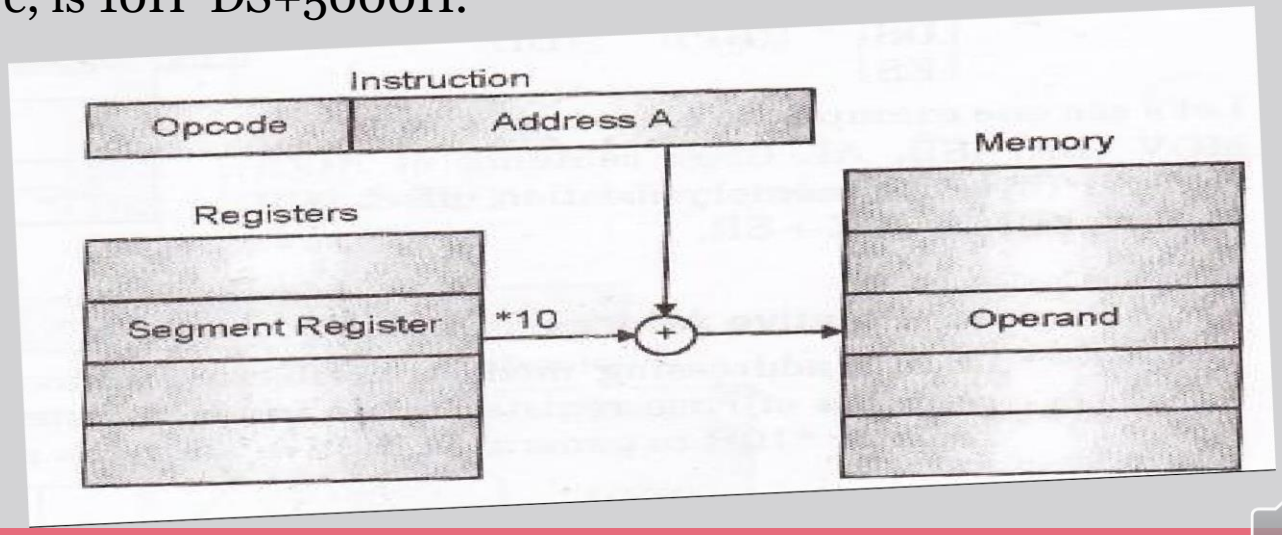
Instruction

Opcode	Operand

ADDRESSING MODES OF 8086 (2)

2. ***Direct***: In the direct addressing mode, a 16-bit memory address (offset) is directly specified in the instruction as a part of it.

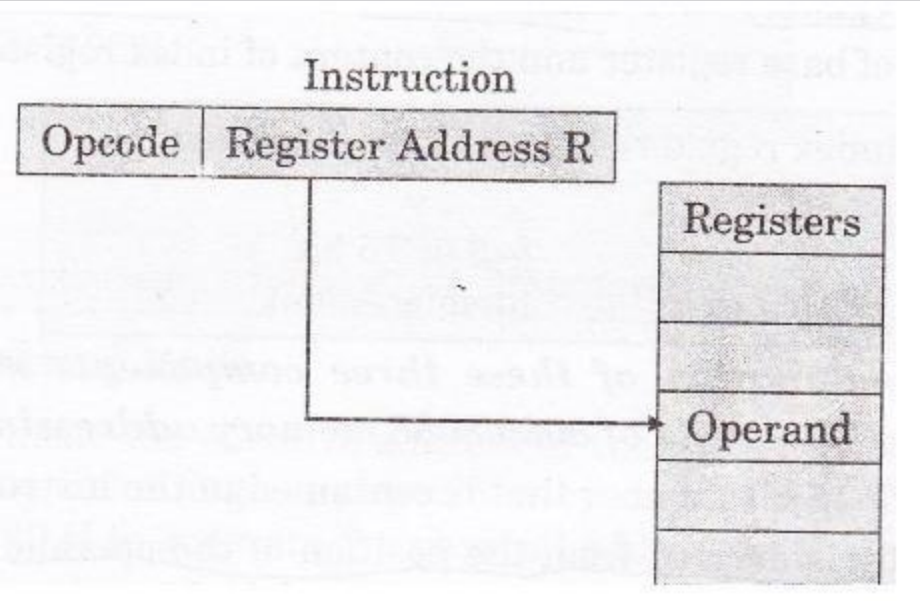
   Example: MOV AX, [5000H]. Here, data resides in a memory location in the data segment, whose effective address may be computed using 5000H as the offset address and content of DS as segment address. The effective address, here, is 10H*DS+5000H.



ADDRESSING MODES OF 8086 (3)

3. ***Register***: In register addressing mode, the data is stored in a register and it is referred using the particular register. All the registers, except IP, may be used in this mode.

Example: MOV AX, BX.

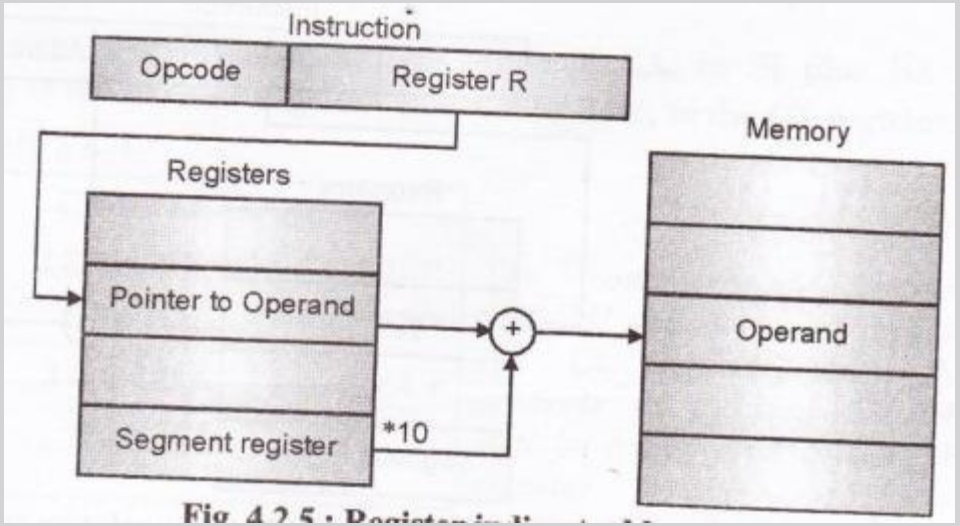


ADDRESSING MODES OF 8086 (4)

4. ***Register Indirect***: In this addressing mode, the offset address of data is in either BX or SI or DI registers.

   The default segment is either DS or ES.

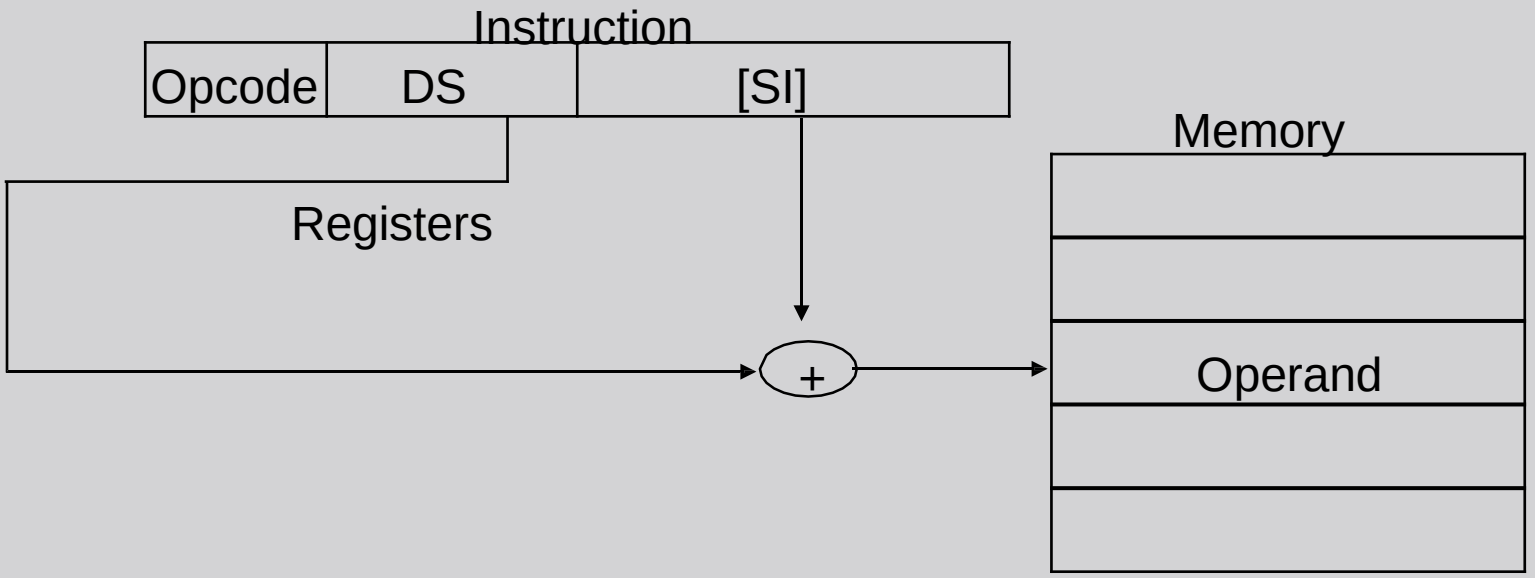
   Example: MOV AX, [BX]. Here, data is present in a memory location in DS whose offset address is in BX. The effective address of the data is given as 10H*DS+ [BX].



ADDRESSING MODES OF 8086 (5)

5. ***Indexed***: In this addressing mode, offset of the operand is stored in one of the index registers. DS and ES are the default segments for index registers SI and DI respectively. This mode is a special case of the above discussed register indirect addressing mode.

   Example: MOV AX, [SI]. Here, data is available at an offset address stored in SI in DS. The effective address, in this case, is computed as 10H*DS+ [SI].



ADDRESSING MODES OF 8086 (6)

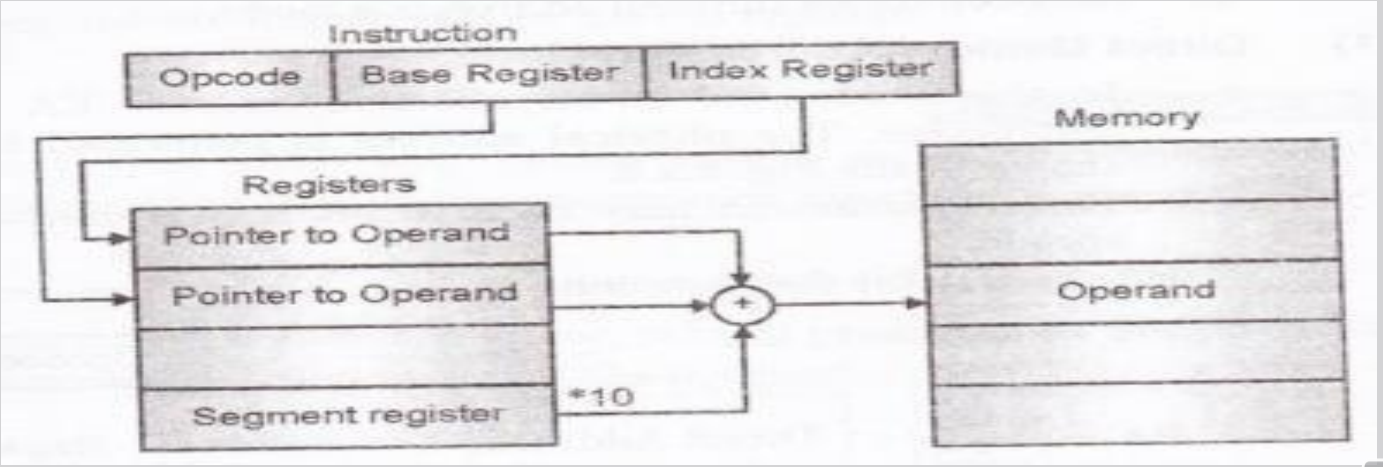
6. ***Register Relative***: In this addressing mode, the data is available at an effective address formed by adding an 8-bit or 16-bit displacement with the content of any one of the registers BX, BP, SI and DI in the default (either DS or ES) segment. The example given before explains this mode.

   Example: MOV Ax, 50H [BX]. Here, effective address is given as 10H*DS+50H+ [BX].

ADDRESSING MODES OF 8086 (7)

7. ***Based Indexed***: The effective address of data is formed, in this addressing mode, by adding content of a base register (any one of BX or BP) to the content of an index register (any one of SI or DI).

   Example: MOV AX, [BX] [SI]. Here, BX is the base register and SI is the index register. The effective address is computed as 10H*DS+ [BX] + [SI].

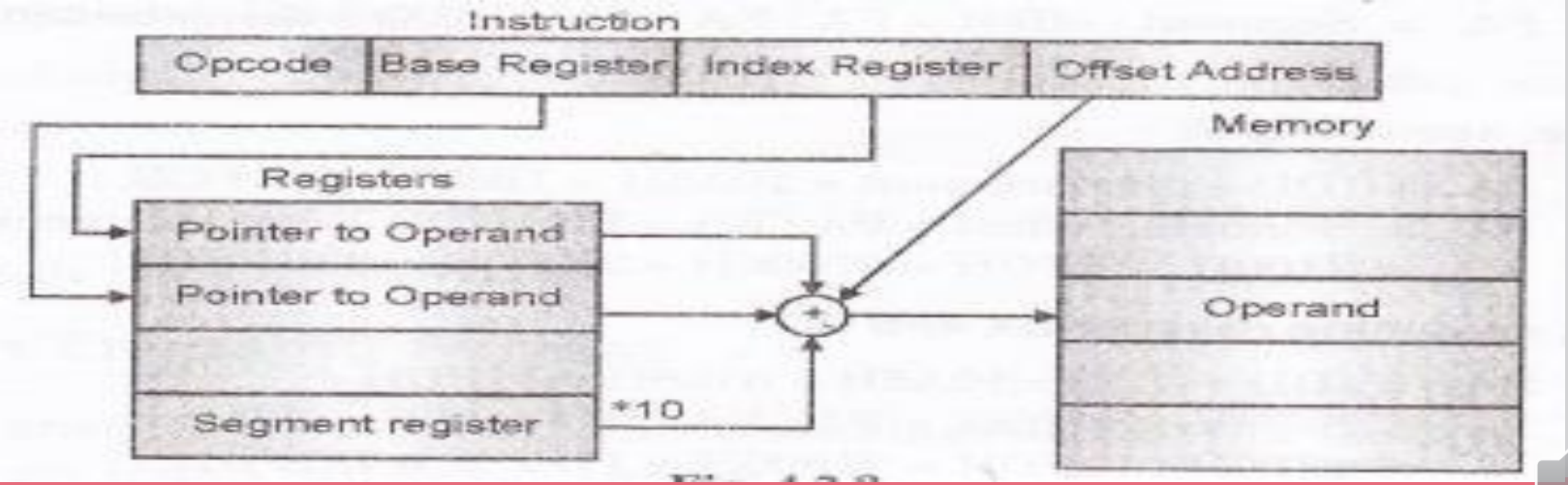


ADDRESSING MODES OF 8086 (8)

8. ***Relative Based Indexed***: The effective address is formed by adding an 8-bit or 16-bit displacement with the sum of contents of any one of the bases registers (BX or BP) and any one of the index registers, in a default segment.

   Example: MOV AX, 50H [BX] [SI]

   Here, 50H is an immediate displacement, BX is a base register and SI is an index register. The effective address of data is computed as 10H*DS+ [BX] + [SI] + 50H.



Examples and EA Calculations

SN	Addressing Mode	Example	EA Calculation
1	Immediate	MOV AX, 5h	EA= Value of Source Operand
2	Direct	MOV AX, [5000H]	**EA=DSX10h + 5000h**
3	Register	MOV AX, BX	EA= Value of Source Register
4	Register Indirect	MOV AX, [BX]	**EA=DSX10h + BX**
5	Indexed	MOV AX, [SI]	**EA=DSX10h + SI**
6	Register Relative	MOV AX, 50h[BX]	**EA=DSX10h + BX + 50h**
7	Based Indexed	MOV AX, [BX][SI]	**EA=DSX10h + BX + SI**
8	Relative Based Indexed	MOV AX, 10h[BX][DI]	**EA=DSX10h + BX + DI + 10h**

Exercise of Addressing Modes
Self Study

- **Consider AX = 1212h; BX= 1B1Bh; CX=2323h; DX=2C2Ch; SI= 0050h; DI = 0110h; SP= 23h;**
- **BP= 0100h; DS=1010h**
- **Calculate the effective address of the following instructions:**

  (i) MOV AX, [2000h]

  (ii) MOV AX, 20h [BP]

  (iii) MOV AX, 40h [BX][DI]

  (iv) MOV AX, 500h