



كلية المأمون الجامعة

قسم تقنيات هندسة الحاسوب

المرحلة الثانية

# [MICROPROCESSORS ARCITECTURE]

## [Chapter3: Addressing Modes]

Dr. DARIN Shafek Salim

# Chapter Content

## ▪ Addressing Modes

- Immediate addressing mode
- Register addressing mode
- Direct addressing mode
- Register indirect addressing mode
  - Based addressing mode
  - Indexed addressing mode
  - Based Indexed addressing mode
  - Based with displacement addressing mode
  - Indexed with displacement addressing mode
  - Based Indexed with displacement addressing mode

## Addressing Modes

Addressing Modes of 8086 When 8086 executes an instruction, it performs the specified function on data. These data are called its operands and may be part of the instruction, reside in one of the internal registers of the microprocessor, stored at an address in memory or held at an I/O port, to access these different types of operands, the 8086 is provided with various addressing modes (Data Addressing Modes). The 8086 has 12 addressing modes. The various 8086 addressing modes can be classified into five groups.

- Addressing modes for accessing immediate and register data (register and immediate modes).
- Addressing modes for accessing data in memory (memory modes)
- Addressing modes for accessing I/O ports (I/O modes)
- Relative addressing mode
- Implied addressing mode

## Immediate addressing mode

In this mode, 8- or 16-bit data can be specified as part of the instruction.

|         |                   |
|---------|-------------------|
| OP Code | Immediate Operand |
|---------|-------------------|

### Example 1

MOV CL, 03 H

Moves the 8-bit data 03 H into CL

### Example 2

MOV DX, 0525 H

Moves the 16-bit data 0525 H into DX

In the above two examples, the source operand is in immediate mode and the destination operand is in register mode.

## Register addressing mode

The operand to be accessed is specified as residing in an internal register of 8086. The table below shows internal registers, one can be used as a source or destination operand, however only the data registers can be accessed as either a byte or word.

| Register          | Operand sizes |               |
|-------------------|---------------|---------------|
|                   | Byte (Reg 8)  | Word (Reg 16) |
| Accumulator       | AL, AH        | Ax            |
| Base              | BL, BH        | Bx            |
| Count             | CL, CH        | Cx            |
| Data              | DL, DH        | Dx            |
| Stack pointer     | -             | SP            |
| Base pointer      | -             | BP            |
| Source index      | -             | SI            |
| Destination index | -             | DI            |
| Code Segment      | -             | CS            |
| Data Segment      | -             | DS            |
| Stack Segment     | -             | SS            |
| Extra Segment     | -             | ES            |

### Example 1

MOV DX (Destination Register), CX (Source Register)

Which moves 16-bit content of CS into DX.

### Example 2

MOV CL, DL

Moves 8-bit contents of DL into CL

MOV BX, CH is an illegal instruction.

\* The register sizes must be the same.

## Direct addressing mode

In this mode, one operand references a memory location and other operand references a register.

### Example:

MOV AX, [6107 h]

SUB [6107 h], CL

Here, data resides in a memory location in the data segment, whose

**The default segment is always DS.**

physical address may be computed using 6107 h as the offset address and content of DS as segment address. The physical address, here, is : physical address = DS \* 10 h + 6107 h.

But by using a segment override prefix (SOP) in the instruction, any of the four segment registers can be referenced,

## indirect addressing mode

The EA is specified in either pointer (BX) register or an index (SI or DI) register. The 20-bit physical address is computed using DS and EA.

## Example

MOV [DI], BX

indirect

If [DS] = 5004, [DI] = 0020, [BX] = 2456 ,PA(Physical address)=50060.

The content of BX (2456) is moved to memory locations 50060 H and 50061 H.

- Based addressing mode**

Here memory is [BX] or [BP].

Example: MOV AX, [BX] ; segment is DS

SBB [BP], CX ; segment is SS

INC wordptr[BP] ; segment is SS

## Example

Operation of  
MOV AX, [BX]  
instruction

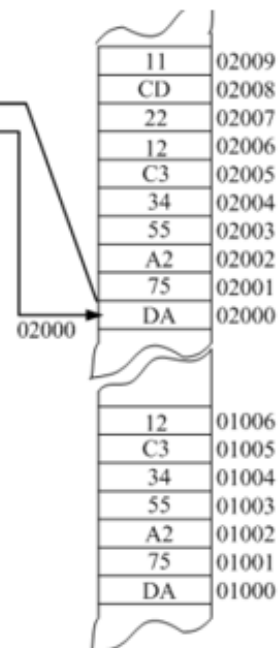


*Handwritten red scribble*

|    |    |    |
|----|----|----|
| AX | 75 | DA |
| BX | 10 | 00 |
| CX | 55 | C2 |
| DX | 32 | 67 |

|    |      |
|----|------|
| CS | 0000 |
| DS | 0100 |
| SS | 2000 |
| ES | 3000 |

|    |      |
|----|------|
| SP | 0000 |
| BP | 0100 |
| SI | 2000 |
| DI | 3000 |



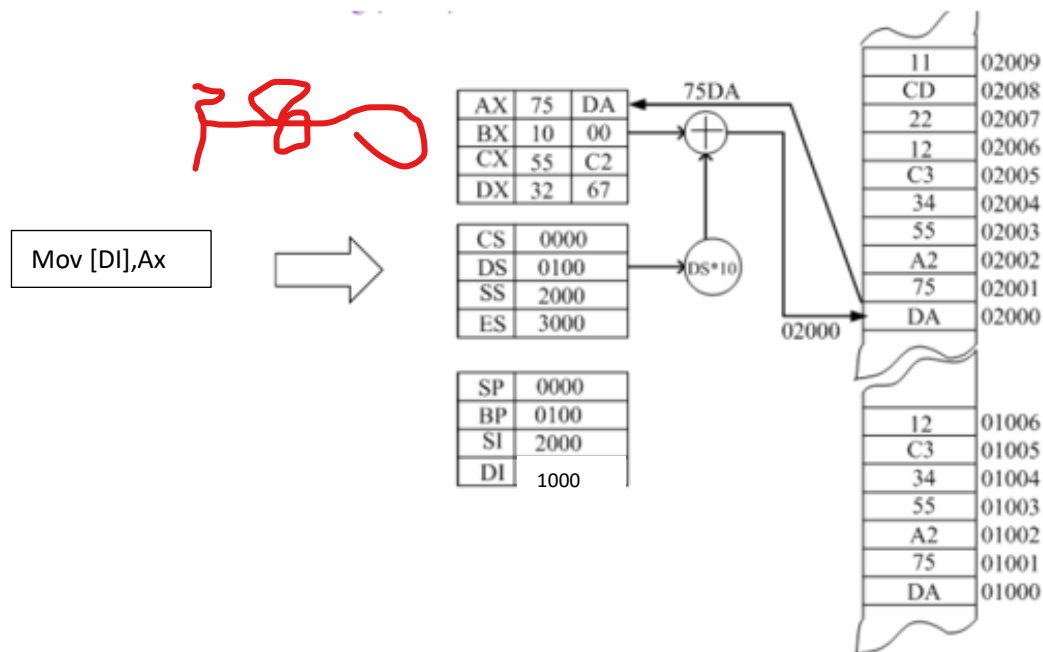
- Indexed addressing mode**

Here memory is [SI] or [DI]. Here segment is DS.

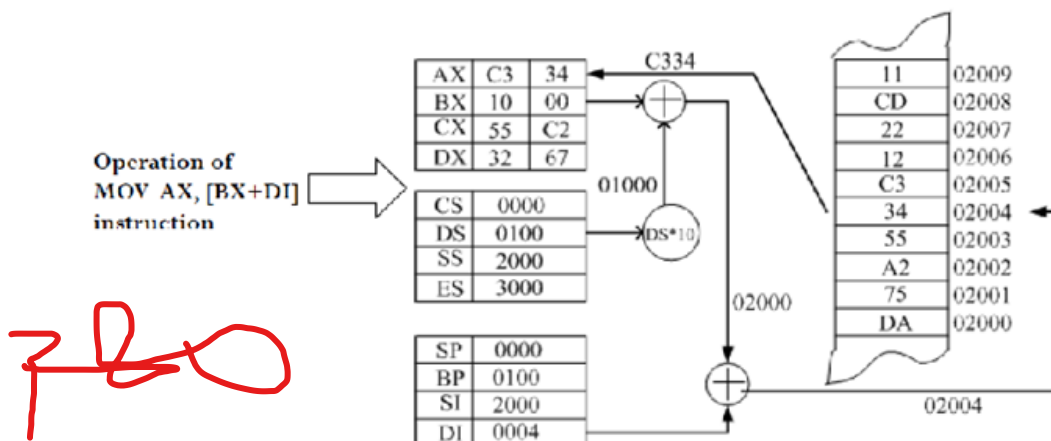
Example: AND [SI], DX

SUB BX, [DI]

### Example



- Based Indexed addressing mode**



## • Base with displacement addressing mode

Here memory is  $[BX + \text{displacement}]$  or

$[BP + \text{displacement}]$ .

Note: Displacement is 8 bit or 16 bit signed value, so it can be both positive or negative.

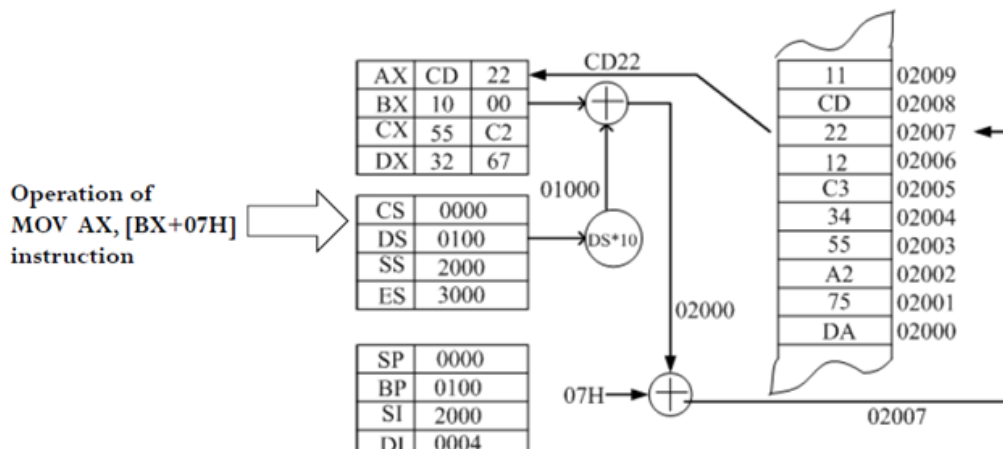
Example: `DEC byteptr [BX+0825 h] ; segment is DS`

`SBB AX , [BP+ 9040 h] ; segment is SS`

`XOR CX,[BX-9000 h] ; segment is DS`

`OR wordptr[BP-7000 h],DX ; segment is SS`

`MOV AX ,[BX+07 h] ; segment is DS`



## • Index with displacement addressing mode

Here memory is  $[SI + \text{displacement}]$  or  $[DI + \text{displacement}]$ . Here segment is DS.

Example: `SUB [SI+33 h] , CX`

`ADC BX , [DI - F122 h]`

## • Base index with displacement addressing mode

Here memory is

[BX+SI+displacement] or [BX+DI+displacement] or [BP+SI+displacement] or [BP+DI+displacement].

Operation of  
MOV AX, [BX+SI+02H]  
instruction

Handwritten signature in red ink.

