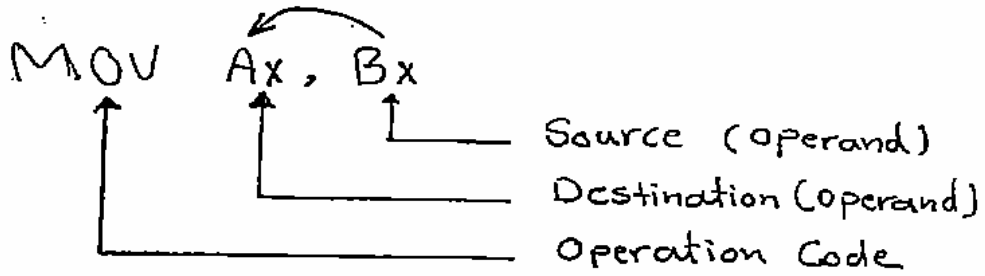
كلية العلوم الجامعة
هندسة الاتصالات وهندسة الروبوتات ونظم المعلومات والحاسبات
الحاسبات / المرحلة الثانية

# Chapter four
# Addressing modes

مادة المادة : المهندس مصطفى

Efficient software development for the Up requires a complete familiarity with the addressing mode employed by each instruction. To provide a basis for the explanation of the data addressing modes, we use the MOV instruction because it is common and flexible instruction.




This instruction transfers the word contents of the source register (Bx) into the destination register (Ax). It is essential to remember that a MOV instructions always copies the source data into the destination (The source NEVER changes). Also, note that the Flag register remains unaffected by data transfer instructions.

* Addressing Modes :- It is the way that the operand of an instruction is reffered to during a program execution. So, it specifies a rule for interpreting or modifying the address field of the instruction befor the operand is actually referenced.

Computers use addressing mode techniques for purpose of accommodating the following provisions :-

1- To give programing versatility to the user by providing such facilities.
2- To reduce the number of bits in the addressing field of the instruction.

* The actual location of the operand value is refferred to as the (Effective Address) (EA) of the operand.

## 1- Data-Addressing Modes :-

The data addressing Modes :-

1 Register Addressing :- register addressing is the most common form of data addressing and, once the register names are learned, it become the easiest to apply.

- 8 bits registers (AH, AL, BH, BL, CH, CL, DH and DL)
- 16 bits registers (AX, BX, CX, DX, SP, BP, SI and DI)

Notes :-

1- Never mix an 8 bit register with a 16-bit register.

2- A Segment-to-Segment register MOV instruction is about the only type that is <u>NOT</u> allowed.

3- Changing the CS register with a MOV instruction is <u>NOT</u> allowed.

The following table shows many variations of register Move instructions.

| Assembly language | size of data moved | operation |
|---|---|---|
| MOV AL, BL | 8-bit moved | AL ← BL |
| MOV CH, CL | 8-bit // | CH ← CL |
| MOV AX, CX | 16-bit // | AX ← CX |
| MOV SP, BP | 16-bit // | SP ← BP |
| MOV DS, AX | 16-bit // | DS ← AX |
| MOV SI, DI | 16-bit // | SI ← DI |

| | | |
|---|---|---|
| MOV ES, DS | x | Not allowed (segment-to-segment) |
| MOV BL, DX | x | Not allowed (mixed size) |
| MOV CS, AX | x | Not allowed (the ES register can not be the destination register.) |

## 2. Immediate Addressing

The term "immediate" implies that the data is immediately following the hexadicimal opcode in the memory. Also note that immediate data are constant data, while the data transfered from a register are variable data. Immediate addressing operates upon a byte or a word. Table 2-2 shows many different variations of MOV instruction that apply immediate data.

| Assembly Language | Size | Operation |
|---|---|---|
| MOV BL, 44 | 8-bit | BL ← 44 (2CH) |
| MOV AX, 0F4H | 16-bit | AX ← 00F4 H |
| MOV SI, 0 | 16-bit | SI ← 0000H |
| MOV CH, 100 | 8-bit | CH ← 100 (64H) |
| MOV AL, 'A' | 8-bit | AL ← ASCII (A) |
| MOV AX, 'AB' | 16-bit | AX ← ASCII (AB) |
| MOV CL, 11010011B | 8-bit | CL ← 11010011 B (D3H) |

## 3. Direct Data Addressing.

There are two basic forms of direct data addressing

1. direct addressing.
2. displacement addressing.

Direct Addressing: Transfers data between a memory location, located within the data segment, and the AL (8-bit), or AX (16-bit). A MOV instruction using this type of addressing is usually a 3-byte long instruction. Table 2.3 lists some direct addressing instructions.

| Assembly Language | Size | Operation |
|---|---|---|
| MOV AL, [1000H] | 8-bit | AL ← DS:[1000H] |
| MOV AX, [2000H] | 16-bit | AL ← DS:[2000H]<br>AH ← DS:[2001H] |
| MOV [4000H], AX | 16-bit | DS:[4000H] ← AL<br>DS:[4001H] ← AH |
| MOV ES:[5000H], AL | | ES:[5000H] ← AL |

The [5000H], [4000H], [2000H], & [1000H] is an offset address of a segment. Sometimes we direct the μP to access a required segment as in the instruction

MOV ES:[5000H], AL

this instruction directs the μP that the offset 5000H is in the Extra Segment (ES). If we don't write this form (as in the other instructions), the μP access the data Segment DS automatically.

Example: Assume that the Stack Segment (SS = 0100H) calculate the physical memory addresses accessed by the following instruction:

MOV SS:[4010H], AX

and that AX = 3E2AH.

Solution:

$\therefore SS = 0100\,H$

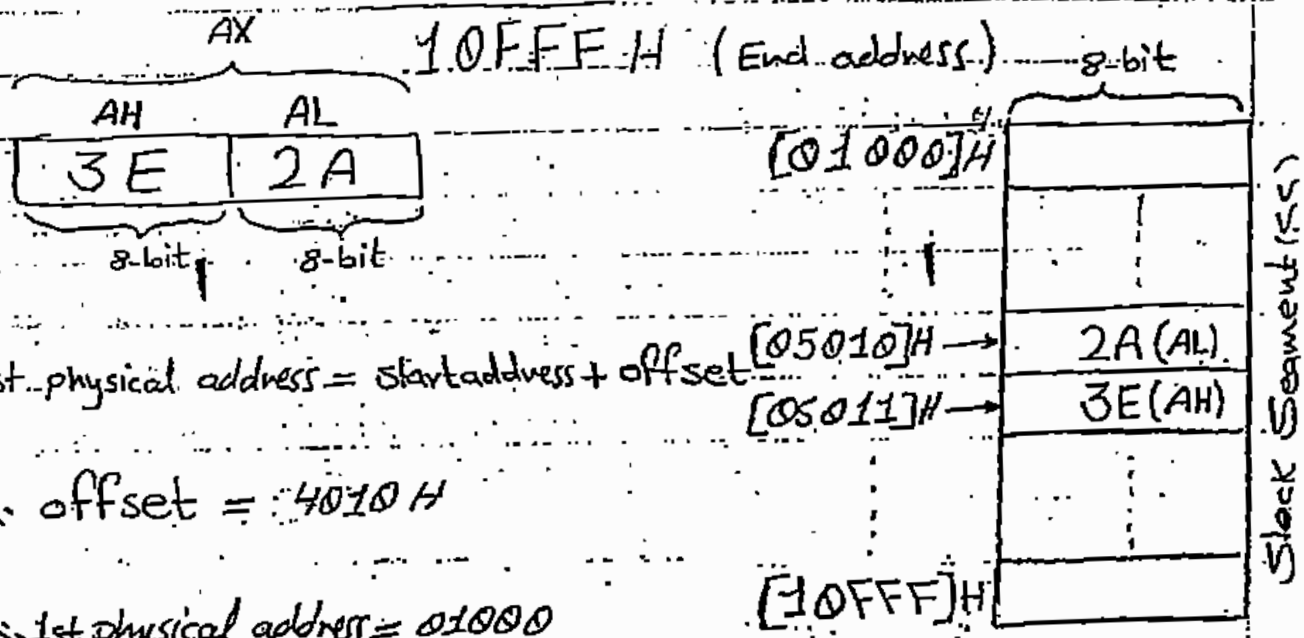
$\therefore$ Start address of the stack segment $= SS * (10)_H$

$= (0100)H * (10)H = 01000\,H$

End address of the stack segment =

Start address + 0FFFF H

$$\begin{array}{r} = 01000 \\ 0FFFF + \\ \hline 10FFF\,H \end{array}$$ (End address)




1st physical address = Startaddress + offset

$\therefore$ offset = 4010 H

$\therefore$ 1st physical address $= 01000$
$\quad\quad\quad