

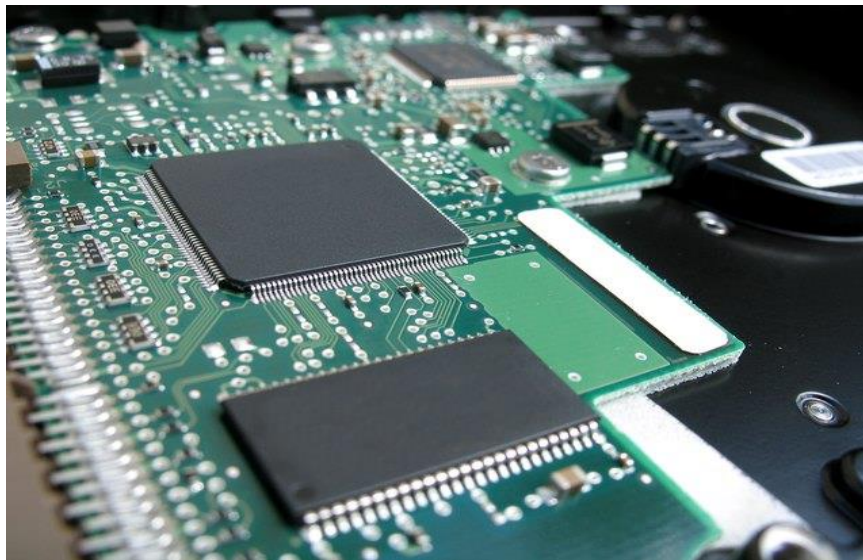


جامعة القادسية  
كلية التربية



## Lecture 13

# Microprocessors



Prepared By:

Firas Abdulrahman Yosif

## ***Arithmetic Instructions***

The 8086 can perform arithmetic operations on binary numbers (signed or unsigned) and on decimal. The arithmetic instructions in 8086 include addition, subtraction, multiplication, division.

1. Addition
2. Subtraction
3. Multiplication
4. Division

- **Addition Instructions**

The addition instructions include: ADD, ADC and INC.

### **1. ADD (Addition)**

In this operation we can add 8- or 16-bit operands. ADD source with destination operand and puts the result in the destination.

destination = destination + source

**The syntax: ADD destination, Source**

- ❖ It adds a byte to byte or a word to word.
- ❖ It effects AF, CF, OF, PF, SF, ZF flags.

### Example:

ADD AL, 07AH ; AL=AL+ 07AH

ADD DX, AX; DX=AX+DX

ADD AX, [BX] ; AX=AX+ [BX]

الحالات المسموح بها:

ADD reg. , reg.

ADD reg. , value

ADD reg. , mem.

ADD mem. , reg.

ADD mem. , value

### Examples:

- ADD AX, BX

adds the 16-bit contents of AX and CX and returns the result in AX.

- ADD CX, 3476H
- ADD AL, [BX]
- ADD [SI], CL
- ADD [1000H], 40H

الحالات الغير مسموح بها:

ADD value, reg.

ADD mem. , mem.

ADD segment reg. , reg.

ADD reg. , segment reg.

### Examples:

- ADD [DI], [BX]      False (لايجوز جمع ذاكرة مع ذاكرة مباشرة)

MOV AL, [BX]      التصحيح:

ADD [DI], AL

- ADD 30H, CX      False (لايجوز جمع رقم في الهدف مع سجل)

ADD CX, 30H      التصحيح:

- ADD DX, BL      False (لايجوز جمع سجل 16 بت مع سجل 8 بت)

ADD DX, BX      التصحيح:

### ❖ Types of Addition operations:

#### a. Register with register Addition

- ❖ Add the content of several registers.
- ❖ When arithmetic instructions executed, contents of the flag register change.
- ❖ Interrupt, trap, and other flags do not change.
- ❖ Any ADD instruction modifies the contents of the sign, zero, carry, auxiliary carry, parity, and overflow flags.

### ❖ Examples:

ADD AL, BL ;      AL=AL+BL

ADD CX, DI ;      CX=CX+DI

## b. Immediate Addition

❖ Immediate addition is employed whenever constant or known data are added.

### Examples:

ADD CL,44H;      CL=CL+44H

ADD BX, 245FH;    BX=BX+245FH

**Q1) What's contain of DL after execute this program?**

MOV DL, 12H

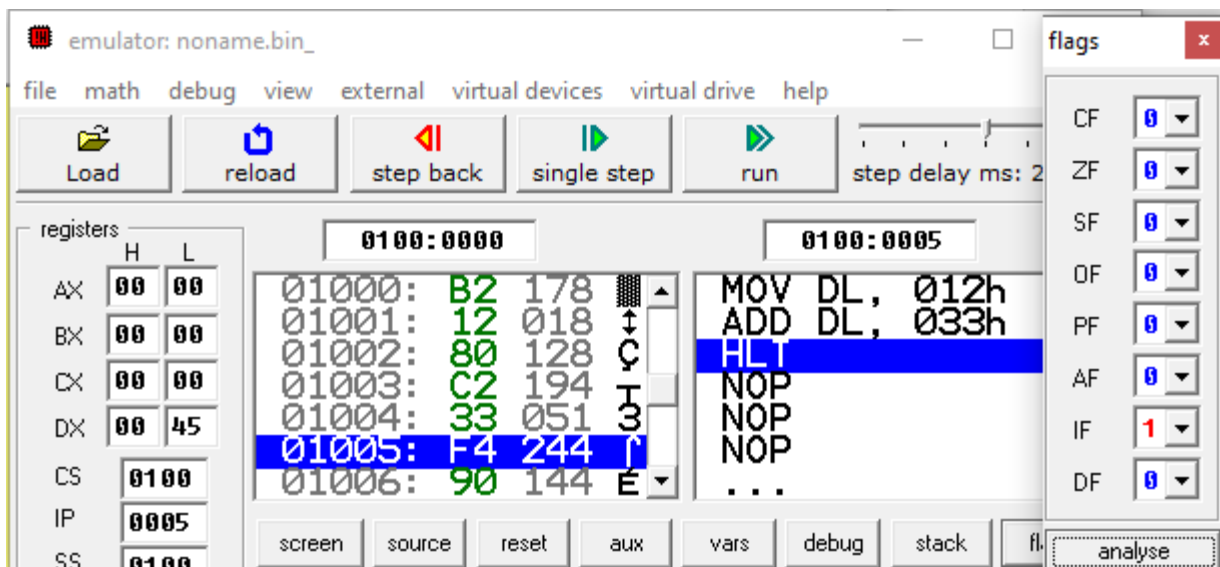
ADD DL, 33H

HLT

Answer)

DL= 45H

لاحظ التنفيذ باستخدام الـ 8086 emulator وان الـ flag register لم تتأثر بعملية الجمع.



## C. Memory with Register Addition

❖ Moves memory data to be added to a register or add register to memory.

### Examples:

- ADD CL, [BP] ;(The byte contents of the stack segment memory location addressed by BP add to CL with the sum stored in CL ).
- ADD [BX], AL ; (AL adds to the byte contents of the data segment memory location addressed by BX with the sum stored in the same memory location).

**Example) what's the result for these instructions after execution it?**

MOV AX, 00F5H

ADD AX, 000BH

HLT

**Solution)**

0000 0000 1111 0101

0000 0000 0000 1011 +

---

0000 0001 0000 0000

AX= 0100H

## 2. ADC(Addition-with-Carry)

ويقصد به جمع مع حمل ويشابه ايعاز ADD في جميع الخصائص باستثناء انه يجمع قيمة (CF) Flag مع الرقم Carry.

destination = destination + source + carry

**The syntax: ADC destination, Source**

- ❖ It adds the two operands with CF.
- ❖ It effects AF, CF, OF, PF, SF, ZF flags.

### Examples:

- ADC AL, AH;     AL=AL+AH+CF
- ADC CX, BX ;    CX=CX+BX+CF

**Example(1) What's contain of AH after execute this program?**

STC

MOV AH, 0FEH

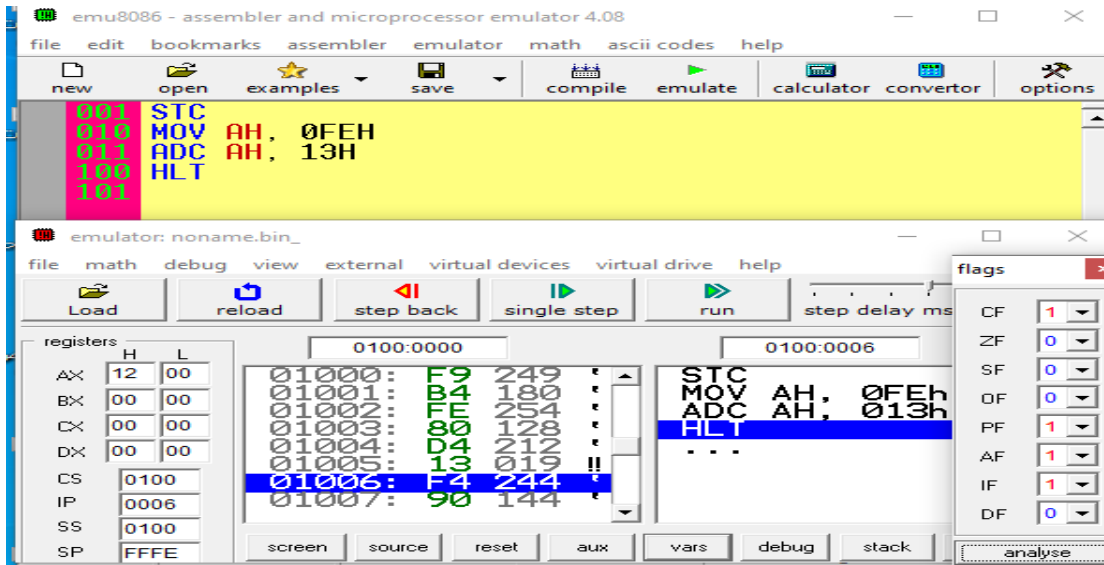
ADC AH, 13H

HLT

|                |             |
|----------------|-------------|
| <b>Answer)</b> | 1111 1110   |
|                | 0001 0011   |
|                | 0000 0001 + |
|                | <hr/>       |
| AH= 12H        | 0001 0010   |

# Microprocessors

لاحظ الناتج باستخدام برنامج 8086 emulator :



**Example(2) what's contain of [SI] and [SI+1] after execution this program?**

STC

MOV SI, 200H

MOV [SI], 034DH

ADC [SI], 23H

HLT

**SOL.)**

0000 0011 0100 1101

0000 0000 0010 0011

0000 0000 0000 0001 +

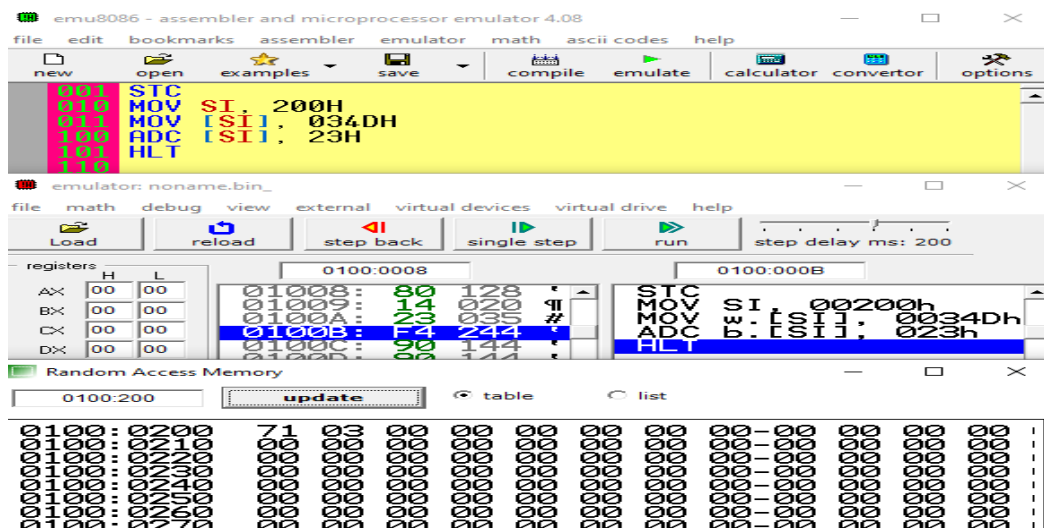
[SI]= 71H

0000 0011 0111 0001

[SI+1]= 03



# Microprocessors



**Example3) Write a program in assembly language to add contains in three memory locations started with address 200H, then put result in DX. Using indirect addressing mod.**

Solution)

CLC

MOV SI, 200H

MOV [SI], 88H

MOV [SI+1], 82H

MOV [SI+2], 6

MOV AL, [SI]

ADD AL, [SI+1]

ADC AL, [SI+2]

MOV DX, AL

HLT

200

|     |
|-----|
| 88h |
| 82h |
| 6   |
| 78H |
| 55H |
| 45H |

# Microprocessors

Answer) DX=0011H

لاحظ التنفيذ باستخدام 8086 emulator :

```
original source code
0001 MOV SI, 200H
0010 MOV [SI], 88H
0011 MOV [SI+1], 82H
0100 MOV [SI+2], 6
0101 MOV AL, [SI]
0110 ADD AL, [SI+1]
0111 ADC AL, [SI+2]
1000 MOV DX, AL
1001 HLT
1010
1011
1100
```

The screenshot shows an 8086 emulator window titled "emulator: noname.bin\_". The interface includes a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, run, and a step delay slider set to 200 ms. Below the toolbar is a "registers" window with columns for H and L bytes. The DX register is circled in red and contains the value 00 11. To the right is a "source" window showing the assembly code being executed, with the instruction "HLT" at address 01016 highlighted in blue. A red arrow points from the DX register value to a text box below.

DX= 0011H

### 3) INC (Increment)

The INC instruction adds 1 to a register or memory operand but, unlike ADD, does not affect the Carry Flag (CF).

ان ايعاز INC يضيف واحد الى محتوى مسجل او موقع الذاكرة ولكنها ليست مثل ايعاز ADD فانها لا يؤثر على CF .

**The syntax: INC Source**

Types of INC instruction:

#### a) INC with register (16 bit or 8 bit)

Examples:

- INC BL ; BL=BL+1
- INC SP ; SP=SP+1

#### b) INC with Memory

Examples:

- INC [SI] ; Increment a byte in memory (زيادة محتوى الذاكرة بمقدار واحد)
- INC [BP]

ملاحظة: لا يمكن استخدام البيانات بصورة مباشرة داخل المصدر (Source) في ايعاز الـ INC .

- INC 23H (false) الايعاز خطأ

التصحيح: MOV CL, 23H

INC CL

ملاحظة: لا يمكن استخدام مقاطع الذاكرة memory segmentations مع الايعاز INC .

# Microprocessors

**Q1) What's contain of DI & [DI] after execute this program?**

MOV DI, 600H

MOV [DI], 439H

INC [DI]

ADD DI, [DI]

HLT

**Answer) DI= A3AH, [DI]= 03AH**

The screenshot shows an emulator window titled "emulator: noname.bin\_". The registers window on the left shows the DI register with the value 0A3A, which is circled in red. The assembly code window on the right shows the following instructions: MOV DI, 00600h; MOV w:[DI], 00439h; INC b:[DI]; ADD DI, [DI]; HLT. The HLT instruction is highlighted in blue. The address window shows the current instruction address as 0100B: F4 244, also circled in red.

The screenshot shows the "Random Access Memory" window. The address window is set to 0100:600. The memory contents are displayed in a table format. The value 3A is circled in red at address 0100:0600. The memory contents are as follows:

|           |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|
| 0100:0600 | 3A | 04 | 00 | 00 | 00 | 00 | 00 |
| 0100:0610 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0100:0620 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0100:0630 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0100:0640 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0100:0650 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0100:0660 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0100:0670 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

DI= 0A3A

[DI]= 3AH