# LABORATORY 2

# Introduction to Assembly Programming

## Objectives:

- To learn 8086 16-bit Intel Microprocessor, its register and assembly level programming.
- To learn the EMU8086 Emulator.
- To learn assembly programming by practicing simple programs including average calculation of 3/4/5/more numbers, calculate area of a rectangle and a triangle, temperature conversion from °C to °F, conversion from °F to °C, conversion from °C to °K, conversion from °K to °C and counting tiles problems.

## Experimental tools:

MDA-Win8086, Computer, Microprocessor emulator Software with Integrated Assembler.
The following rules are needed to perform the lab work.

## Instructions:

RES System Reset

AD Set Memory address

DA Update segment && offset.

STP Execute user's program, a single step.

GOGo to user's program or execute monitor functions.

MON Immediately break user's program and Non makeable interrupt.

REG Register display

+ Segment & offset +1 increment. Register display increment.

- Segment & offset -1 increment. Register display increment.

From the emulator you will get the **HEX** code for your assembly language program. For execute code and get the result first of all press **RES** button then press **DA** button, now type your **HEX** code here go to next address pressing **+** button. After completing the code typing you have to press **STP** button for executing the code. Now for watch result press **REG** button then you can see the result in the display of MDA-Win8086.

## Instruction set:

**MOV**: This instruction allows copying the value of one register into another register.

**ADD:** This instruction adds two numbers.

**SUB:** This instruction subtracts a number from another number.

**MUL:** This instruction multiplies two numbers.

**DIV:** The instruction divides a number by another number.

**AX**= It is called accumulator register.

**BX**= It is called base register.

**CX**= It is called count register.

**DX**=It is called data register

<table>
<tr><td colspan="2">Problem 1: Temperature conversion from °C to °K<br>let, temperature = 39°C<br>1°K=1°C + 273</td></tr>
<tr><td colspan="2">MOV AX, 39<br>MOV BX, 273<br>ADD AX, BX<br>INT 3</td></tr>
<tr><td><u>Output:</u></td><td>AX=0138 BX=0111<br>CX=0000 DX=0000</td></tr>
<tr><td><u>Result Verification:</u></td><td>K= 39+273= 312</td></tr>
<tr><td colspan="2"><u>Discussion</u>: We know, °K= °C + 273<br>At first, 27 loaded in AX register and the address is 0404 and 111 replaced in BX register and its address is 0407. Now, AX, BX are added in address 041A. After pressing STP and REG, it shows the result.<br><br>INT 3: INT 3 is a special one-byte instruction having op-code is CCH. that is inserted by debuggers at the instruction where the user has set a breakpoint to occur. When it's hit, the interrupt handler breaks into the debugger and then replaces the original instruction so that execution can proceed when the user is ready.</td></tr>
</table>

Merge Problem 1 and Problem 2 and show the students about the task done by the INT3

**Problem 2:** Temperature conversion from °K to °C
let, temperature = 270°K

```
MOV   AX, 270
MOV   BX, 273
SUB    AX, BX
INT     3
```

| **Output:** | AX=FFFD BX=0111<br>CX=0000 DX=0000 |
|---|---|
| **Result Verification:** | C= 270-273= -3 = FFFDH |

**Discussion:** We know, °C = °K – 273
At first, 10E replaced in Ax register and the address is 0404 and 111 replaced in Bx register and its address is 0407. Now, Ax, Bx are subtract in address040A.After pressing STP and REG, it shows the result.

**Problem 3:** Average of 3 numbers: (2+3+5)/3

```
MOV    AX, 2
MOV    BX, 3
ADD    AX, BX
MOV    BX, 5
ADD     AX, BX
MOV   BX, 3
DIV   BL
INT 3
```

| **Output:** | AX=0103 BX=0003<br>CX=0000 DX=0000 |
|---|---|
| **Result Verification:** | Avg= (2+3+5)/3= 3<br>AL = 3, AH = 1 |

**Discussion:**
At first, load 2 in AX register and the address is 0404 and load 3 in BX register and its address is 0407. Now, AX, BX are added in 040A then load 5 in BX in 040C, and AX, BX are added

again in 040F. Now, load 3 in BX and the address is 0411. Then BL is divided in 0413 address. After pressing STP and REG, it shows the result.

**Problem 4:** Average of 5 numbers: ((2+3+4+1+5)/5)

```
MOV  AX, 2
MOV  BX, 3
ADD  AX, BX
MOV  BX, 4
ADD  AX, BX
MOV  BX, 1
ADD  AX, BX
MOV  BX, 5
ADD  AX, BX
MOV BX, 5
DIV   BL
INT 3
```

| | |
|---|---|
| **Output:** | AX=0003 BX=0005<br>CX=0000 DX=0000 |
| **Result Verification:** | Avg= (2+3+4+1+5)/5= 15/5<br>AL = 3, AH = 0 |

**Discussion:**
At first, 2 replaced in Ax register and the address is 0404 and 3 replaced in Bx register and its address is 0407. Now, Ax, Bx are added in 040A then 4 replaced in Bx in 040C, and Ax, Bx are added again in 040F. Now, 1 replaced in Bx and the address is 0411. Then Bx is added in 0414 address and again 5 replaced in Bx and the address is 0416 then Bx is added in 0419. Now, 5 replaced in Bx and the address is 041B. Then Ax is divided by BL in 041D address. After pressing STP and REG, it shows the result.

**Problem 5:** Floor size 20*20, Tiles size 2*2. How many tiles are needed to cover up the floor?

```
MOV   AX, 20
MOV   BX, 20
MUL   BL
MOV   CX, AX
MOV   AX, 2
```

```
MOV   BX, 2
MUL   BL
MOV   BX, AX
MOV   AX, CX
DIV    BL
INT 3
```

| | |
|---|---|
| **Output:** | AX=0064 BX=0004<br>CX=0190 DX=0000 |
| **Result Verification:** | Tiles = (20*20)/ (2*2) = 400/4= 100 = 64H |

**Problem 6:** Factorial Operation: 5! – 3!

```
MOV   AX, 1
MOV   CL, 5
L1:      MUL CL
LOOP L1
MOV  DX, AX
MOV  AX, 1
MOV  CL, 3
L2:   MUL CL
LOOP   L2
MOV   BX, AX
MOV    AX, DX
SUB    AX, BX
INT     3
```

| | |
|---|---|
| **Output:** | AX=0072 BX=0006<br>CX=0000 DX=0078 |
| **Result Verification:** | 5! – 3! = 114 = 72H; AH = 00, AL = 72 |

**Discussion:**

At first, we load 1 in AX register and load 5 in CL register then do multiply by giving loop with CL address and move AX value in DX register. Now, again entered value 1 in AX register and 3 replaced in CL register then do multiply by giving loop with CL address and

move AX value in BX register. Then move the DX value in AX register and do subtraction of AX and BX. After pressing STP and REG, it produces the result.

**Problem 7:** (5! / 3!) + 4!

```
MOV   AX, 1
MOV  CL, 5
L1:MUL   CL
LOOP  L1
MOV   DX, AX
MOV   AX, 1
MOV   CL, 3
L2:MUL   CL
LOOP  L2
MOV   BX, AX
MOV   AX, DX
DIV    BL
MOV  DX, AX
MOV  AX, 1
MOV  CL, 4
L3:      MUL   CL
LOOP  L3
ADD   AX,DX
INT     3
```

| **Output:** | AX=002C BX=0006<br>CX=0000 DX=0014 |
|---|---|
| **Result Verification:** | (5! / 3!)+4!=(120/6)+24=20+24=44=2C H, AH=00, AL=2C |

**Discussion:**

At first, load 1 in AX register and load 5 in CL register then do multiply by giving loop with CL register and move AX value in DX register. Again, load value 1 in AX register and 3 in CL register then do multiply by giving loop with CL register and move AX value in BX register. Then move the DX value in AX register and do division by BL. Now, move AX value in DX and again entered value 1 in AX register and 4 replaced in CL register then do multiply by giving loop with CL address. At last, we do addition of DX and AX. After pressing STP and REG, we get the result.

**Problem 8: (2! *3! *4!) +4!**

```
MOV   AX, 1
MOV   CL, 2
L1:MUL   CL
LOOP  L1
MOV   DX, AX
MOV   AX, 1
MOV   CL, 3
L2:MUL   CL
LOOP  L2
MOV   BX, AX
MOV   AX, DX
MUL    BL
MOV   DX, AX
MOV   AX, 1
MOV   CL, 4
L3:MUL   CL
LOOP  L3
ADD   AX, DX
INT     3
```

| | |
|---|---|
| **Output:** | AX=0138 BX=0018<br>CX=0000 DX=0120 |
| **Result Verification:** | (2! * 3!) + 4! = (2 * 6) + 24 = 12 + 24 = 36 = 24 HDH=00, DL=24 |

**Discussion:**

At first, load 1 in AX register and load 2 in CL register then do multiply by giving loop with CL address and move AX value in DX register. Again, enter value 1 in AX register and 3 in CL register then do multiply by giving loop with CL address and move AX value in BX register. Then move the DX value in AX register and do division by BL. Now, move AX value in DX and again load value 1 in AX register and 4 in CL register then do multiply by giving loop with CL address. Now, we do addition of DX and AX in address. At last, we do addition of DX and AX. After pressing STP and REG, we get the result.

**Problem 9: (4!/2!)/3!**

```
MOV   AX, 1
MOV   CL, 4
L1:
   MUL   CL
LOOP   L1
MOV   DX, AX
MOV   AX, 1
MOV   CL, 2
L2:
   MUL   CL
LOOP   L2
MOV    BX, AX
MOV   AX, DX
DIV    BL
 MOV   DX, AX
MOV   AX, 1
MOV   CL, 3
L3:
   MUL   CL
LOOP   L3
MOV   BX, AX
MOV   AX, DX
DIV    BL
INT     3
```

| | |
|---|---|
| **Output:** | AX=0002 BX=0006<br>CX=0000 DX=000C |
| **Result Verification:** | (4! / 2!)/3! =(24/2)/6=12/6=2=2HAH=00, AL=02 |

**Discussion:**

At first, 1 replaced in AX register and 4 replaced in CL register then do multiply by giving loop with CL address and move AX value in DX register. Again, entered value 1 in AX register and 2 replaced in CL register then do multiply by giving loop with CL address and move AX value in BX register. Then move the DX value in AX register and do division by BL. Now, move AX value in DX and again entered value 1 in AX register and 3 replaced in CL register then do multiply by giving loop with CL address. Now, we move the AX value in BX register then move the DX value in register AX and divide the value by BL. After pressing STP and REG, we get the result.

**Problem 10: Byte with Byte Division**

```
ORG 100h
.MODEL SMALL
.DATA
NUM_1 DB 0F2H
NUM_2 DB 4H
.CODE
MOV BH, NUM_2     ;Load numerator in BH
MOV AL, NUM_1     ;Load denominator in AL
DIV BH            ;Divide BH by AL
RET
```

| **Output:** | AX=023C |
|---|---|
| The DIV instruction divides BH by AL. F2 divided by 04 gives quotient of 3C and give 02 as a remainder. AL stores the quotient and remainder is stored in AH register. | |

- ORG (abbr. for ORiGin) is an assembly directive (not an instruction). It defines where the machine code (translated assembly program) is to place in memory. As for ORG 100H this deals with 80x86 COM program format (COMMAND) which consist of only one segment of max. 64k bytes. 100H says that the machine code starts from address (offset) 100h in this segment, effective address is CS:100H.
- With .model small you get a program where CS points to a 64k bytes code segment and DS point to 64k bytes data segment. Thus, code and data both use 64k bytes maximum space.

```
.MODEL MEDIUM     ;the data must fit into 64K bytes
                  ;but the code can exceed 64K bytes of memory
.MODEL COMPACT    ;the data can exceed 64K bytes
                  ;but the code cannot exceed 64K bytes
.MODEL LARGE      ;both data and code can exceed 64K
                  ;but no single set of data should exceed 64K
.MODEL HUGE       ;both code and data can exceed 64K
                  ;data items (such as arrays) can exceed 64K
.MODEL TINY       ;used with COM files in which data and code
                  ;must fit into 64K bytes
```

- 

**Problem 11: Word with Word Division**

```
ORG 100h
.MODEL SMALL
.DATA
NUM_1 DW 0F213H
NUM_2 DW 41A8H
.CODE
```

| | |
|---|---|
| MOV AX, NUM_1 ;Load numerator in AX<br>DIV NUM_2 ;Divide AX by NUM_2<br>RET | |
| **Output:**<br><br>The output window shows that the division of F213H by 41A8 gives the remainder of 2D1B into DX register and 03 as a quotient into AX. | AX=0003 quotient<br><br>DX=2D1B remainder |
| | |

**Conclusion:**

In this experiment, we have learnt conversion from °C to °F, conversion from °F to °C, conversion from °C to °K, Conversion from °K to °C, Average of 3 numbers, average of 5 numbers, area of rectangle, area of triangle, find how many tiles. After performing those operation, we use assembly language in 8086 microprocessors which results in getting the correct output.

**Example: HomeWorks**

1. Temperature conversion from °C to °F (37°C)
2. Temperature conversion from °F to °C (110°F)
3. Temperature conversion from °F to °K (130°F) ; AX=0547H; K=(F-32)*5/9+273;
4. Temperature conversion from °K to °F (300°K); F=9(K-273)/5+32
5. 3! +4!
6. (4! + 3!) - 2!
7. (1! * 2!) * 6!
8. Find out the average of the ten numbers.
9. Factorial Operation: 7! – 4! + 2!
10. Floor size 80*80, Tiles size 4*4. How many tiles will be required to pave up the floor?