Lab (i): Addressing data in memory and segments

Part I

1. How to open the debug program?
   - mount the location/directory of the 8086 file mount c c:\8086
   - enter the directory c:
   - type debug -?

2. What is the DEBUG command to perform the following operations?

   a) Begin assembling statements in assembly language, and that will be converted and saved in machine language.

   - a [address]

   b) Display the contents of registers.

   - r

   c) Execute, then display contents of registers.

   - t

   d) Enter machine instructions into memory.

   - e [address]

   e) Display the contents of memory segments.

   - d [address]

3. Use DEBUG to enter the following commands:

```
a 100
mov BL, 42
mov DL, 2A
add BL, DL
jmp 100
```

```
-a 100
073F:0100 mov bl, 42
073F:0102 mov dl, 2a
073F:0104 add bl, dl
073F:0106 jmp 100
073F:0108
```

What you can see when the following command is typed?

a) U 100,107

```
-u 100,107
073F:0100 B342          MOV     BL,42
073F:0102 B22A          MOV     DL,2A
073F:0104 00D3          ADD     BL,DL
073F:0106 EBF8          JMP     0100
```

b) D CS:100

```
-d cs:100
073F:0100  B3 42 B2 2A 00 D3 EB F8-00 00 00 00 00 00 AE FE   .B.*............
```

c) E CS:100 A2 00 02 03 06 02 02

```
-e cs:100 a2 00 02 03 06 02 02
-d
073F:0180  04 E8 9E EF EB 02 33 C0-5E 5F C9 C2 06 00 C8 26   ......3.^_.....&
073F:0190  04 00 57 56 8B 76 06 8B-1C 2A E4 8A 47 04 89 46   ..WV.v...*..G..F
073F:01A0  F8 8D 46 D8 50 68 B4 0F-E8 F5 21 C6 86 DA FB 20   ..F.Ph....!.... 
073F:01B0  8A 44 10 2A E4 03 44 02-89 46 FA 8B F8 EB 7B 39   .D.*..D..F....{9
073F:01C0  7E FA 75 0D 8B 5E 06 2A-ED 8A 4F 11 03 4F 04 EB   ~.u..^.*..O..O..
073F:01D0  02 8B C8 8B D1 8D 86 D9-FB 03 D0 89 56 FE 89 7E   ............V..~
073F:01E0  FC 8B F2 EB 48 8A 1C C0-EB 03 2A FF 8D 46 D8 03   ....H.....*..F..
073F:01F0  D8 2A E4 8A 07 8A 0C 80-E1 07 BA 01 00 D3 E2 85   .*..............
```

d) U 100,106

```
-u 100, 106
073F:0100 A20002        MOV     [0200],AL
073F:0103 03060202      ADD     AX,[0202]
-
```

4. Use DEBUG to enter the following commands:

A 100
MOV AX, 0123
ADD AX, 0025
MOV BX, AX
ADD BX, AX
MOV DX, BX
SUB DX, AX
SUB AX, AX
JMP 100

```
AX=0123  BX=006C  CX=0000  DX=002A  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PE NC
073F:0103 052500        ADD     AX,0025
```

What you can see when the following command is typed?

a) U 100, 111

```
-u 100,111
073F:0100 B82301        MOV     AX,0123
073F:0103 052500        ADD     AX,0025
073F:0106 89C3          MOV     BX,AX
073F:0108 01C3          ADD     BX,AX
073F:010A 89DA          MOV     DX,BX
073F:010C 29C2          SUB     DX,AX
073F:010E 29C0          SUB     AX,AX
073F:0110 EBEE          JMP     0100
```

b) R

```
-r
AX=0123  BX=006C  CX=0000  DX=002A  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PE NC
073F:0103 052500        ADD     AX,0025
```

c) T (repeat 8 times)

```
-t

AX=0123  BX=0290  CX=0000  DX=0148  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL ZR NA PE NC
073F:0103 052500        ADD     AX,0025
```

What is the value of AX,BX,CX,DX after executing 8 times from the address 100?

AX = 0000/0123

BX = 0290

CX = 0000

DX = 0148

What is the alternative to execute 8 times in a single command?

5. Use DEBUG to enter the following command:

E CS:100 12 43 56 78 9A BC

Now, what you see after D CS:100 ?

The hexadecimal value 43 at address CS:101 was supposed to be 34. Code another E command to correct only the one byte that is incorrect, that is change 43 to 34 directly.

E cs: 101 34
D cs: 100

6. Assume that you have used DEBUG to enter the following E command:

E CS:100 B8 05 1B 00 2C EB F8

Use U command to find what are the 3 assembly/symbolic instruction represented here.

ADD
SUB
JMP

Part II

1. Enter these machine language instructions into the code segment address

100. B0 1C D0 E0 B3 12 F6 E3 EB F6

```
-e cs:100
073F:0100  00.b0   00.1c   00.d0   00.e0   00.b3   00.12   00.f6   00.e3
073F:0108  00.eb   00.f6   00.     00.     00.

-d cs:100
073F:0100  B0 1C D0 E0 B3 12 F6 E3-EB F6 00 00 00 00 00 00   ................
073F:0110  00 00 00 00 00 00 00 00-00 00 00 00 34 00 2E 07   ............4...
073F:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

Determine which DEBUG command to use. Which memory byte above performs the following operations?

(Hint:you need to convert the above machine language to assembly language)

```
-u cs:100
073F:0100 B01C          MOV     AL,1C
073F:0102 D0E0          SHL     AL,1
073F:0104 B312          MOV     BL,12
073F:0106 F6E3          MUL     BL
073F:0108 EBF6          JMP     0100
073F:010A 0000          ADD     [BX+SI],AL
073F:010C 0000          ADD     [BX+SI],AL
073F:010E 0000          ADD     [BX+SI],AL
073F:0110 0000          ADD     [BX+SI],AL
073F:0112 0000          ADD     [BX+SI],AL
073F:0114 0000          ADD     [BX+SI],AL
073F:0116 0000          ADD     [BX+SI],AL
073F:0118 0000          ADD     [BX+SI],AL
073F:011A 0000          ADD     [BX+SI],AL
073F:011C 3400          XOR     AL,00
073F:011E 2E            CS:
073F:011F 07            POP     ES
-_
```

a) Move hex value 1C to the AL register.

Machine code : B0 1C , Memory address : 0100

This instruction (MOV AL, 1C) moves the hexadecimal value 1C into the AL register.

b) Shift the contents of AL one bit to the left.

Machine code : D0 E0, Memory address : 0102

This instruction (SHL AL, 1) shifts the contents of the AL register left by one bit

c) Move the hex value 12 to BL.

Machine code : B3 12, Memory address : 0106

This instruction (MOV BL, 12) moves the hexadecimal value 12 into the BL register.

d) Multiply AL by BL.

Machine code : F6 E3, Memory address : 0106

This instruction (MUL BL) multiplies the contents of AL by BL.

e) Jump back to 100

Machine code : EB F6, Memory address : 0108

This instruction (JMP 100) causes the program to jump back to address 100, creating a loop.

Execute the program and identify what is the final product in AX?

2. What is the output in AH and AL? Why?

```
A 100
MOV AH, 0
MOV AL, 7
MOV BL, 10
MUL BL ;BL is 8-bit register
JMP 100
```

Which arithmetic operation is performed when you execute MUL BL?

- AL x BL
- AX x BX
- AX x BL
- AL x BX

Why? Would you be able to explain?

3. What is the output in AH and AL? Why?

```
A 100
MOV AH, 0
MOV AL, 83
MOV BL, 2
DIV BL ;BL is 8-bit register
JMP 100
```

Which arithmetic operation is performed when you execute DIV BL?

- AL / BL
- AX / BX
- AX / BL
- AL / BX

Why? Would you be able to explain?

4. What is the output in AX and DX? Why?

```
A 100
MOV DX, 0
MOV AX, 8003
MOV CX, 100
DIV CX ;CX is 16-bit register
JMP 10
```

Which arithmetic operation is performed when you execute DIV CX?

- AL / CL
- AX / CX
- AX / CL
- AL / CX

Why? Would you be able to explain?

5. Enter into AL and BL register so that AL contains AA and that an item named BL contains F0. Determine the result on AL for the following unrelated operations by using debug program:

a) AND AL, BL

b) OR AL, BL

c) XOR AL, BL

d) NOT AL

6. What is the output in AX?

A 100
MOV AL, 8
SHR AL, 1
MOV BL, 8
SHL BL, 1
JMP 100

	AX
MOV AL, 8	
SHR AL, 1	
MOV BL, 8	
SHL BL, 1	

Part III: Independent Practical

1. Update CS register to 116E (Using R command), and enter the following instructions into DEBUG program.

-A 100
116E:0100 MOV AX, 0010
116E:0103 MOV BX, 0020
116E:0106 MOV CX, 0030
116E:0109 ADD AX, BX
116E:010B INC BX
116E:010C SUB CX, AX
116E:010E DEC CX
116E:010F JMP 0100
116E:0111 <enter>

What is the content of register AX, BX, CX and IP?

AX	**BX**	**CX**	**IP**

2. Trace the content of the registers used in the following program segment:

```
    MOV AX, 1
MOV BX, 1
MOV CX, 5
MOV DX, 0
A10:
ADD AX, BX
MOV DX, AX
MOV AX, BX
MOV BX, DX
LOOP A10
```

	AX	**BX**	**CX**	**DX**
Before Loops				
After 1st loop				
After 2nd loop				
After 3rd loop				
After 4th loop				
After 5th loop				

3. Trace the execution of the following instructions and record the values of the register.

```
MOV AX,010
MOV BX,020
MOV CX,030
ADD AX,BX
INC BX
SUB CX,AX
DEC CX
JMP 100
```

AX	BX	CX

4. What is the value of AX and BX?

```
        MOV AX,00
        MOV BX,00
        MOV CX,3 ;Initialize for 3 loops
A20:
        INC AX
        ADD BX,AX
        LOOP A20 ;Decrement CX ;Repeat if nonzero
```

AX	BX	CX

5. What is the final value of AX and BX?

```
        MOV AX,0 ;Initialize AX and
        MOV BX,0 ;BX to zero,
        MOV CX,4 ;CX for 4 loops
A20:
        INC AX ;Add 01 to AX
        ADD BX,AX ;Add AX to BX
        LOOP A20 ;Decrement CX, loop if nonzero
```

AX	**BX**