



University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

Computer Organization and Assembly Language

Lab 2	
Topic	<ol style="list-style-type: none">1. Mov instruction2. Viewing memory of dosbox3. Declare variables

PART 1 (B)

Types of Registers:-

The registers are grouped into three categories:-

1. General registers

1.1. *Data registers*

- 1.1.1. **AX** is the primary accumulator.
- 1.1.2. **BX** is known as the base register.
- 1.1.3. **CX** is known as the count register.
- 1.1.4. **DX** is known as the data register.

1.2. *Pointer registers*

- 1.2.1. Instruction Pointer **IP**
- 1.2.2. Stack Pointer **SP**
- 1.2.3. Base Pointer **BP**

1.3. *Index registers*

- 1.3.1. Source Index **SI**
- 1.3.2. Destination Index **DI**

2. Control registers

- 2.1. Instruction Pointer and Flag register

3. Segment registers

- 3.1. Code Segment **CS**
- 3.2. Data Segment **DS**
- 3.3. Stack Segment **SS**
- 3.4. Extra Segment **ES**



University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

Types of variables

Type	No. of bits	Example declaration:
Byte	8	Num1: db 43
Word=> 2 bytes	16	Num2: dw 0xABFF
double word=> 2 words	32	Num3: dd 0xABCDEF56

Viewing memory in DOSBOX

Areas highlighted in red(memory 1) “m1” and blue (memory 2) “m2” are showing the memory contents. *Note:* Two copies of the same memory is displayed in the given window.

Area highlighted with yellow is showing the ascii values of the contents displayed in the memory m2.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0028 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

0100 8A261D01 MOV AH,[011D]
0104 8B1E1E01 MOV BX,[011E]
0108 01D8 ADD AX,BX
010A A32001 MOV [0120],AX
010D 8B0E2201 MOV CX,[0122]
0111 A12401 MOV AX,[0124]
0114 8B1E2601 MOV BX,[0126]
0118 B8004C MOV AX,4C00

DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

= f.Ω≡ i |..†...
.....ff. ....
ó.....J. ....
.....
```



University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)
FACULTY OF INFORMATION TECHNOLOGY

Viewing sample variable in memory.

- To view memory from window m2 run the command “m2 ds:Addressofvariable” example: m2 ds:011F
- A variable with name “num1” is initialized with value 65 decimal.
41 hex = 65 decimal is the ascii of “A”.

```
[org 0x0100]
```

```
mov ax, [num1] ; load first number in ax
mov bx, [num2] ; load second number in bx
add ax, bx ; accumulate sum in ax
mov bx, [num3] ; load third number in bx
add ax, bx ; accumulate sum in ax
mov [num4], ax ; store sum in num4
mov bx, [num5]; load lower 2 bytes of num5 in bx register
mov cx, [num5+2]; load higher 2 bytes of num5 in cx register

mov ax, 0x4c00 ; terminate program
int 0x21
```

```
num1: dw 65
num2: dw 10
num3: dw 15
num4: dw 0
num5: dd 0x12345678
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 002B BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

M2 {/A} addr i seg_reg:
CMD > m2 ds:011F

0100 A11F01 MOV AX, [011F]
0103 8B1E2101 MOV BX, [0121]
0107 01D8 ADD AX, BX
0109 8B1E2301 MOV BX, [0123]
010D 01D8 ADD AX, BX
010F A32501 MOV [0125], AX
0112 8B1E2701 MOV BX, [0127]
0116 8B0E2901 MOV CX, [0129]

DS:011F 41 00 0A 00 0F 00 00 00 78 56 34 12 10 15 66 89 A xU4...fë
DS:012F 85 32 FF FF FF 58 5A 66 C7 85 20 FF FF FF 05 00 à2 xZf ||à ..
DS:013F 56 6A 2F E8 59 0E FF FF 8B 15 AC 35 0D 00 66 8B Uj/sY. i.%5..fi
DS:014F 0D BA 35 0D 00 8D 42 08 8E E9 65 66 8B 00 83 C4 .||5..iB. Åøefi.â-
DS:015F 10 66 3D 00 01 0F 85 48 FE FF FF 8D 42 17 8E E9 .f=...àH ■ iB.Åø

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri



University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

Add Comment in code:

Use ; to start comment

```
[org 0x100]
mov al,[num1]
mov bh,[Num1]
add ax,bx

mov cl,[num2]
mov dx,[mynum] ;when using dw variables use a 16-bit register.

add cx,dx

mov ax,0x4c00
int 21h
```

```
num1: db 01100001b ;b is for binary
Num1: db 97 ;decimal by default, case sensitive names of variables
num2: db 0x61 ; 0x treats it as a hexadecimal number
mynum: dw 6100h; h at the end treats it as a hexadecimal number
temp: dw 0xABCD ;when using characters as a hex values, use 0x
```

or open your code in notepad++ select the code you want to comment/uncomment press ctrl+Q

Example 1:

```
[org 0x100]
mov al,9
mov bl,5
add al,bl
mov cl,-10
mov dl,11
add cl,dl
add al,cl

mov ax,0x4c00
int 21h
```



University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

Example 2:

```
[org 0x100]
mov ax,75
mov bx,517
add al,bh
mov cl,200
mov dl,56
add cl,dl ;why we are getting 0 after addition ?
```

```
mov ax,0x4c00
int 21h
```

Example 3:

```
[org 0x100]
mov ax,75
mov bx,517
add al,bh
mov cl,200
mov dl,56
add cl,dl ;why we are getting 0 after addition ?
```

```
mov ax,0x4c00
int 21h
```



University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

Example 4:

```
[org 0x100]
```

```
mov al,[num1]
```

```
mov bh,[Num1]
```

```
add ax,bx
```

```
mov cl,[num2]
```

```
mov dx,[mynum] ;when using dw variables use a 16-bit register.
```

```
add cx,dx
```

```
mov ax,0x4c00
```

```
int 21h
```

```
num1: db 01100001b ;b is for binary
```

```
Num1: db 97 ;decimal by default, case sensitive names of variables
```

```
num2: db 0x61 ; 0x treats it as a hexadecimal number
```

```
mynum: dw 6100h; h at the end treats it as a hexadecimal number
```

```
temp: dw 0xABCD ;when using characters as a hex values, use 0x
```

```
;uncomment the following line and then assemble again
```

```
; temp2: dw EFh ;why dosbox is showing error on this line???
```