



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 3	
Topic	1. Mov instruction 2. Add, sub with variations 3. Addressing Modes with variations.

PART 1

Types of Registers:-

The registers are grouped into three categories:-

1. General Purpose 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

Note: size of both operands must be same for any type of instruction.

For example:

Mov ax,dh is wrong because destination is 2 bytes and source is 1 byte.

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 windows.

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



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 type the command “m2 ds:Addressofvariable”
example: m2 ds:011F and hit enter
- A variable with name “num1” is initialized at memory location 11F with value 65 decimal.

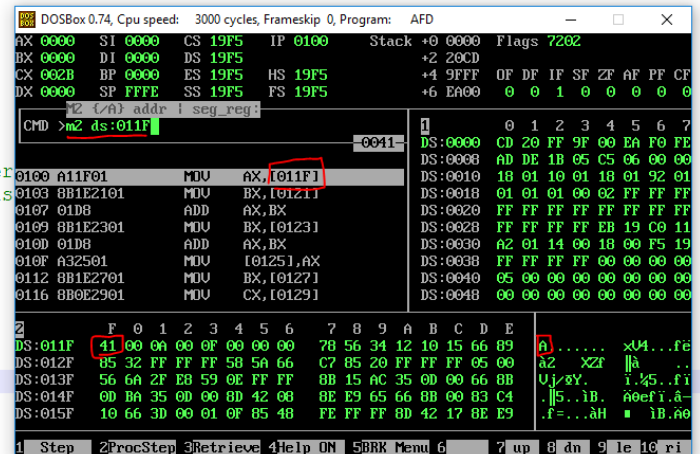
41 hex = 65 decimal is the ascii of “A”.

```
[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
```





University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

Types of Addressing Modes

Direct A fixed offset is given in brackets and the memory at that offset is accessed. For example “mov [1234], ax” stores the contents of the AX registers in two bytes starting at address 1234 in the current data segment. The instruction “mov [1234], al” stores the contents of the AL register in the byte at offset 1234.	<ul style="list-style-type: none">• Mov ax,[num1] ;reading• Mov [num2],ax ;writing
Based Register Indirect A base register is used in brackets and the actual address accessed depends on the value contained in that register. For example “mov [bx], ax” moves the two byte contents of the AX register to the address contained in the BX register in the current data segment. The instruction “mov [bp], al” moves the one byte content of the AL register to the address contained in the BP register in the current stack segment.	<ul style="list-style-type: none">• Mov bx,var• Mov cx,[bx]• Mov [bx],ax
Based Register Indirect + Offset A base register is used with a constant offset in this addressing mode. The value contained in the base register is added with the constant offset to get the effective address. For example “mov [bx+300], ax” stores the word contained in AX at the offset attained by adding 300 to BX in the current data segment. The instruction “mov [bp+300], ax” stores the word in AX to the offset attained by adding 300 to BP in the current stack segment.	<ul style="list-style-type: none">• mov [bx+3], ax• mov cl,[bp+5]



University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

Example 1 (Direct Addressing)

[org 0x100]

Mov al,[var1]

Mov bx,[var2]

Mov cx,[var3+2]

Mov dx,[var3]

mov ax,0x4c00

int 21h

var1: db 10 ;

var2: dw 20

var3: dd 0xABCDEF56

Example 2 (Based Register Indirect Addressing)

[org 0x100]

Mov bx,var1

Mov al,[bx]

Mov bx,var2

Mov ax,[bx]

mov ax,0x4c00

int 21h

var1: db 10

var2: dw 10101010b



University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

Example 3 (Based Register Indirect + Offset)

```
[org 0x100]
Mov bx,var1
Mov al,[bx]
Mov cx,[bx+1]
Mov ax,[bx+3]
Mov dx,[bx+5]
mov ax,0x4c00
int 21h
var1: db 10
var2: dw 10101010b
var3: dd 0xFFAABBCC
```

Execute every part of Question 1 in *Nasm with Dosbox* and observe the memory variables and register values.

Q1.

- a) Create simple variables of type byte, word and double word.

```
[org 0x100]
MOV ah,[var1]
MOV bx,[var2]
add ax,bx
MOV [var3],ax
MOV cx,[var4]
Mov ax,[var5]
Mov bx,[var5+2]
```



University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

```
mov ax,0x4c00
```

```
int 21h
```

```
var1: db 10 ;
```

```
var2: dw 20
```

```
var3: dw 20h
```

```
var4: dw 40h
```

```
var5: dd 0xABCDEF56
```

b) Direct addressing of variables by using the address of only one variable

```
[org 0x100]
```

```
mov bl,[v]
```

```
add bl,[v-1]
```

```
add bl,[v+1]
```

```
mov ax,0x4c00
```

```
int 21h
```

```
u: db 34h
```

```
v: db 35h
```

```
w: db 36h
```

c) Indirect accessing of memory of byte size

```
[org 0x100]
```

```
mov ax,0
```

```
mov bx,0
```

```
mov cx,0
```

```
mov dx,0
```

```
mov bx, var1
```

```
mov al,[bx]
```

```
mov cl,[bx+1]
```

```
mov ch,[bx+2]
```

```
sub ch,cl
```



University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

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

```
var1: db 10  
var2: db 55h  
var3: db 13
```

- d) Copy this code and observe what's wrong with that and correct it.

```
[org 0x100]  
MOV ax,[var1]  
MOV bx,[var2]  
add ax,bx  
MOV [var3],ax  
MOV cx,[var4]
```

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

```
var1: db 5  
var2: dw 6  
var3: dw 7  
var4: dw 8
```

- e) Indirect accessing of Word size

```
[org 0x100]
```

```
mov ax,0  
mov bx,0  
mov cx,0  
mov dx,0
```

```
mov bx, var1  
mov ax,[bx]  
mov cx,[bx+2]
```




University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

```
mov dx,[bx+4]
```

```
add ax,dx
```

```
mov [bx+2],ax
```

```
mov ax,0x4c00
```

```
int 21h
```

```
var1: dw 10
```

```
var2: dw 20h
```

```
var3: dw 13
```

f) Reading and writing in memory through indirect memory address.

```
[org 0x100]
```

```
mov ax,0
```

```
mov bx,0
```

```
mov cx,0
```

```
mov dx,0
```

```
mov bx, var1
```

```
mov ax,[bx]
```

```
mov cx,[bx+2]
```

```
add ax,cx
```



University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

```
mov [bx],ax
```

```
mov [bx+2], word 0
```

```
mov dx,[bx+3]
```

```
add dx,[bx+3]
```

```
mov [bx+3],dx
```

```
mov ax,0x4c00
```

```
int 21h
```

```
var1: dw 60
```

```
var2: db 5
```

```
var3: dw 100
```

g) When using variables of different sizes

```
[org 0x100]
```

```
mov ax,0
```

```
mov bx,0
```

```
mov cx,0
```

```
mov dx,0
```

```
; add two variables their sum should be 5163 or 142B
```

```
;option1
```

```
Mov al,[var1]
```

```
Mov bl,[var2]
```

```
Add al,bl
```

```
;option2
```

```
Mov al,[var1]
```

```
Mov bx,[var2]
```

```
Add al,bx ;this will show error because of size mismatch change al to ax then run again
```

```
;option3
```



University of Central Punjab

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

FACULTY OF INFORMATION TECHNOLOGY

Mov ax,[var1] ;why ax is not showing the correct value of var1

Mov bx,[var2]

Add ax,bx

;option4

Mov al,[var1]

Mov ah,0 ;already 0 in ah

Mov bx,[var2]

Add ax,bx

mov ax,0x4c00

int 21h

var1: db 60 ;0x3C

var2: dw 5103 ;0x13EF