Computer Organization & Assembly Language
(Lecture 04)

Omar Bin Samin
Lecturer
Institute of Management Sciences, Peshawar.

Label

- A label is the address of specific line of code
- Syntax
  - Lable1:
  - Lable_1:

- Consider the following piece of code:

```
[org 0x0100]
mov ax,2
mov bx,3
add ax, bx
mov ax,0x4c00   ; exit
int 0x21
```

- It the above given piece of code, values of “ax” and “bx” are hard coded and the data is available inside the instruction. Such operands are referred to as “Immediate Operands”

- If we want to refer a particular instruction to perform any required operation, then we need to recall the address of the referred instruction, which is typically not feasible

Program with Multiple Labels

- To resolve this issue, replace data with label/ tag

```
[org 0x0100]
mov ax,[Tag1]
mov bx,[Tag2]
add ax, bx
mov ax,0x4c00      ; exit
int 0x21

Tag1: dw 2
Tag2: dw 3
```

Global Variables

- “dw” stands for “Define Word”
- “dw” allocates 16 bits

Listing File

```
1                       [org 0x0100]
2   00000000  A1[0E00]   mov ax,[Tag1]
3   00000003  8B1E[1000]  mov bx,[Tag2]
4   00000007  01D8        add ax, bx
5   00000009  B8004C      mov ax,0x4c00     ; exit
6   0000000C  CD21        int 0x21
7
8   0000000E  0200        Tag1: dw 2
9   00000010  0300        Tag2: dw 3
```

- Keep the following in mind:

```
mov ax, 2              (Legal)
mov ax, bx             (Legal)
mov ax,[Tag1]          (Legal)
mov [Tag1],bx          (Legal)
mov [Tag1],[Tag2]      (Illegal)
```

- Due to the use of memory addresses (labels/ tags), the machine code for “mov ax” and “mov bx” are also changed

Program with Single Label (Method 1)

- Consider the following piece of code:

```
[org 0x0100]
mov ax,[Tag1]
mov bx,[Tag1+2]
add ax, bx
mov ax,0x4c00      ; exit
int 0x21

Tag1: dw 2
  dw 3
```

Global Variables

Listing File

```
1                    [org 0x0100]
2  00000000  A1[0E00]    mov ax,[Tag1]
3  00000003  8B1E[1000] mov bx,[Tag1+2]
4  00000007  01D8        add ax, bx
5  00000009  B8004C  mov ax,0x4c00 ; exit
6  0000000C  CD2         int 0x21
7
8  0000000E  0200        Tag1: dw 2
9  00000010  0300             dw 3
```

Program with Single Label (Method 2)

- Consider the following piece of code:

```
[org 0x0100]
mov ax,[Tag1]
mov bx,[Tag1+2]
add ax, bx
mov ax,0x4c00    ; exit
int 0x21

Tag1: dw 2,  3
```

Global Variables

Listing File

```
1                    [org 0x0100]
2  00000000  A1[0E00]    mov ax,[Tag1]
3  00000003  8B1E[1000] mov bx,[Tag1+2]
4  00000007  01D8        add ax, bx
5  00000009  B8004C  mov ax,0x4c00      ; exit
6  0000000C  CD21        int 0x21
7
8  0000000E  02000300   Tag1: dw 2,3
```

Task 01

- Show the values (data) of ax, bx and Tag1 for all executable instructions:

```
[org 0x0100]
mov ax,[Tag1]
mov [Tag1+6],ax

mov ax,[Tag1+2]
add [Tag1+6],ax

mov ax,[Tag1+4]
add [Tag1+6],ax

mov ax,0x4c00        ; exit
int 0x21

Tag1: dw 2, 5, 7, 0
```

Task 02

- Show the values (data) of ax, bx and Tag1 for all executable instructions:

```
[org 0x0100]
mov ax,[Tag1]
mov [Tag2],ax

mov ax,[Tag1+2]
add [Tag2],ax

mov ax,[Tag1+4]
add [Tag2],ax

mov ax,0x4c00        ; exit
int 0x21

Tag1: dw 2, 5, 7
Tag2: dw 0           ; Result
```

Task 03

- Show the values (data) of ax, bx and Tag1 for all executable instructions:

```
[org 0x0100]
mov ax,[Tag1]
mov bx,4
add ax,bx

mov ax,0x4c00      ; exit
int 0x21

Tag1: db 2, 1
```