**Lab 03 Sec C 5/09/2017**

**Computer Organization and Assembly Language Fall 2017**

**Data Types**

|  |  |  |
| --- | --- | --- |
| DB | Define Byte | allocates 1 byte (0 - 28 – 1) |
| DW | Define Word | allocates 2 bytes (0 – 216 – 1) |
| DD | Define Doubleword | allocates 4 bytes (0 – 232 – 1) |
| DQ | Define Quadword | allocates 8 bytes (0 – 264 – 1) |
| DT | Define Ten Bytes | allocates 10 bytes (0 - 280 – 1) |

***Example:***

*; a program to add three numbers using memory variables*

*[org 0x0100]*

*mov ax, [num1] ;load first number in ax*

Code Segment

Pointed by **CS**

*mov bx, [num2] ; load second number in bx*

*add ax, bx ; accumulate sum in ax*

*mov [result1], ax ; store result in result1 variable, 15*

*mov bx, [num3] ; load third number in bx*

*add ax, bx ; accumulate sum in ax*

*mov [num4], ax ; store result in result2 variable, 30*

*mov ax, 0x4c00 ; terminate program*

*int 0x21*

*num1: db 5 ;variables*

Data Segment

Pointed by **DS or ES**

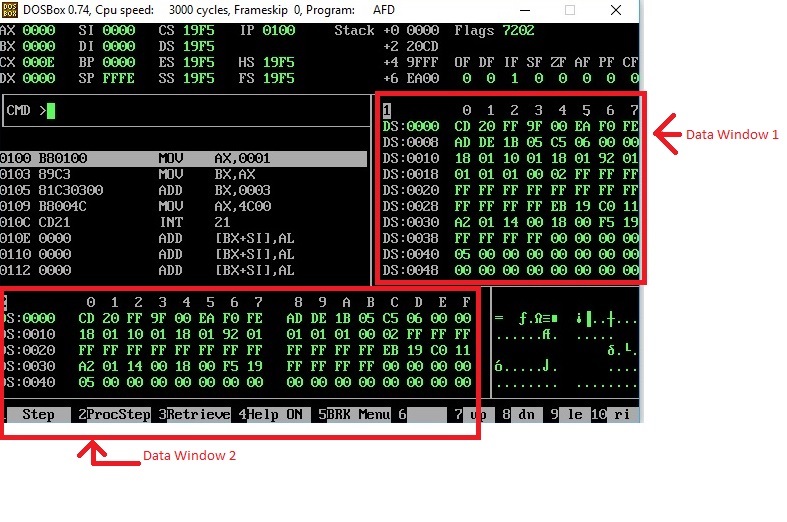
*num2: db 10*

*num3: db 15*

*result1: db*

*result2: db*

**How to View Memory In AFD:**

**

In above screen shot, there are two data windows, each window is showing the contents of Memory. Such as at Offset 0000, we can see that the data is CD and at Offset 0001, the data is 20.

If you want to see the data at offset 0040, simply write m1 DS:0040 or m2 DS:0040 on AFD console.

(m1 is for window 1, and m2 is for window 2).

If you want to check your declared memory content, you have to crate listing file of your program, then note offset of your data label from listing file, and then simply write m1 DS:offset, then you will see your data label content in m1 window

**In Lab Problems**

**Problem 1:**

**Task i: Which of the following mov instructions are valid or invalid?** *Do this task on a paper*

**For example**:

MOV Register1, Register2 (Valid: Mov Ax, Bx)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Immediate 2** | **Registers 2**  **(16 bit)** | **Registers 2**  **(8 bit)** | **Memory 2** |
| **Immediate 1** |  |  |  |  |
| **Register 1 (16 bit)** |  |  |  |  |
| **Register 1 (8 bit)** |  |  |  |  |
| **Memory 1** |  |  |  |  |

**Task ii: Following code contain lot of bugs. You need to identify bugs and re-write correct program without adding, removing and rearranging any instruction. Logic of the program isn’t important.**

*[org 0x100]*

*Mov ax, n1*

*mov bx, [n2]*

*add ax, bx*

*mov 5, ax*

*mov ax, bl*

*add [n1], [n2]*

*mov [n1],20*

*movch, [n3]*

*mov ax, 0x4c00 ; terminate program*

*int 0x21*

*n1 :db 5*

*n2: db 10*

*n3: dw 20*

**Problem 2:**

Write instructions to do the following.

**a.** Copy contents of memory location with offset 0025 in the current data segment into AX.

**b.** Copy AX into memory location with offset 0FFF in the current data segment.

**c.** Move contents of memory location with offset 0010 to memory location with offset 002F in the current data segment.

**Problem 3:**

**Make 2 word Type Arrays with 10 numbers each, add the corresponding elements of the 2 arrays and store them in a third array of type Word. Don’t use loops**

*Example:*

*Array 1 = 101, 200, 500,320,550, 632, 400, 100, 200 , 900 (DW type)*

*Array 2 = 50, 99, 256, 230, 550, 600, 220, 100 , 200 , 300 (DW type)*

*Array 3 = 151, 299, 756, 550, 1100, 1232, 620, 200, 400, 1200 (DW type)*

