National University of Computer and Emerging Sciences

Laboratory Manual

for

Computer Organization and Assembly Language Programming

(EL 213)

Course Instructor	Ms. Sana Fatima
Lab Instructor(s)	Mr. Hazoor Ahmad Mr. Muhammad Salman Mubarik
Section	BSR-A
Semester	Fall 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Learn different data types.
- ✓ Differentiate between code data segment.
- ✓ Declare and use variables in assembly language.

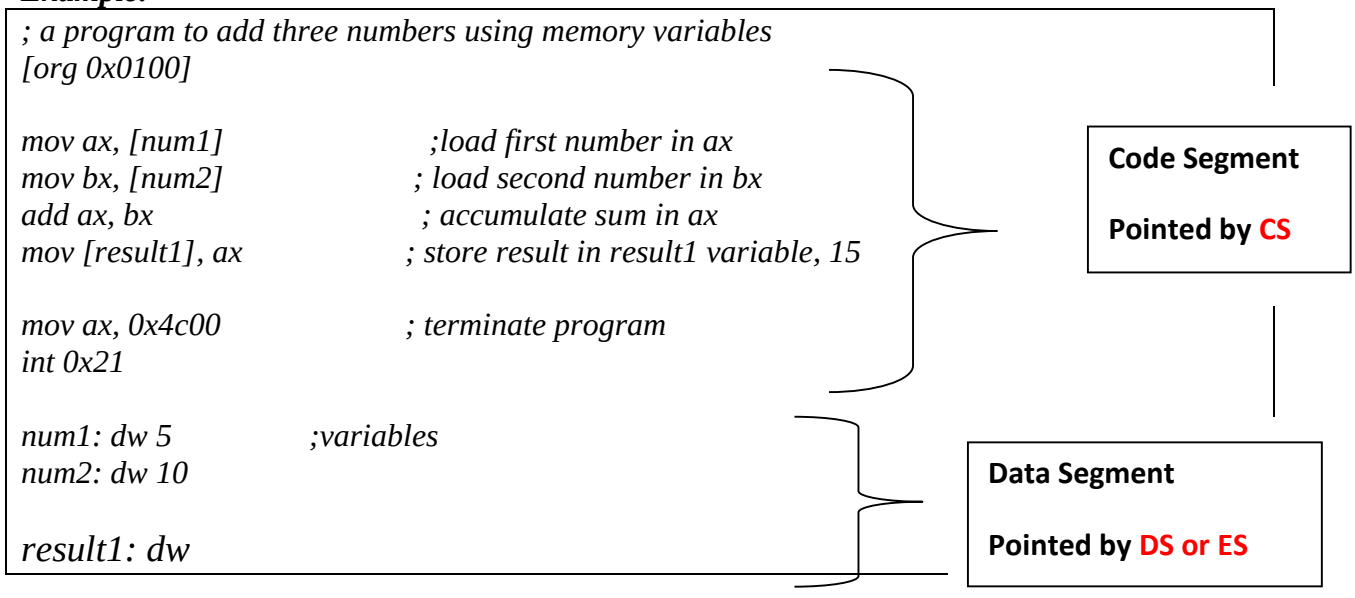
Data Types

Variables are declared in memory.

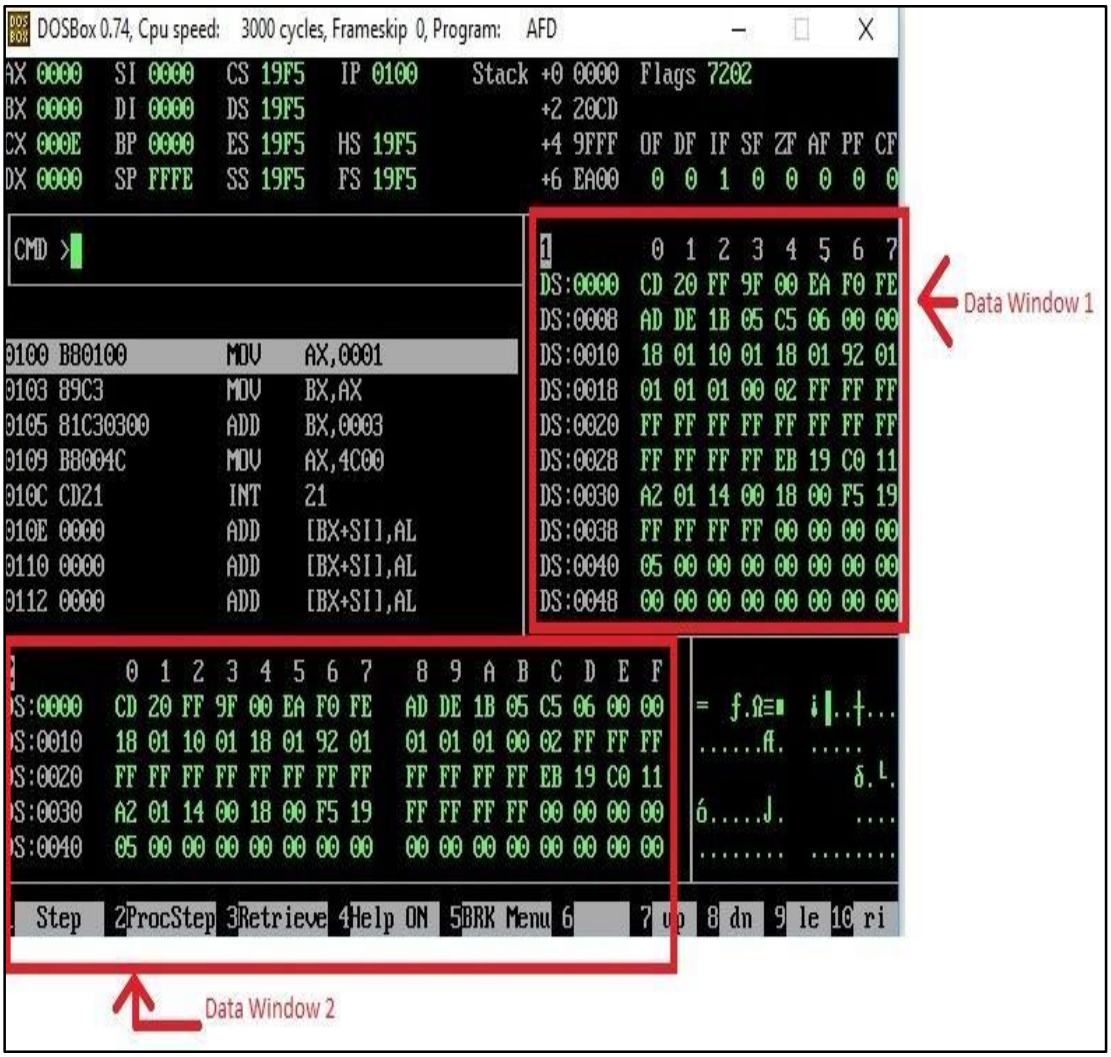
DB	Define Byte	allocates 1 byte $(0 – (2^8 – 1))$
DW	Define Word	allocates 2 bytes $(0 – (2^{16} – 1))$
DD	Define Doubleword	allocates 4 bytes $(0 – (2^{32} – 1))$
DQ	Define Quadword	allocates 8 bytes $(0 – (2^{64} – 1))$
DT	Define Ten Bytes	allocates 10 bytes $(0 – (2^{80} – 1))$

Example:

```
; a program to add three numbers using memory variables
[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 [result1], ax       ; store result in result1 variable, 15

mov ax, 0x4c00          ; terminate program
int 0x21

num1: dw 5              ;variables
num2: dw 10

result1: dw
```



How to View Memory in AFD



In above screenshot, 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 create 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. Offset value will be calculated after the addition of 0x0100 in the instruction address displayed by listing window since we ask assembler to start writing machine code from the address 0x0100 using the first line of code: [org 0x0100]

Exercise 1: Write instructions to do the following. Visualize the memory contents using memory windows to see if instruction is executed correctly. (Use m2 DS:offset to visualize the memory contents at the specified offset)
a. Copy contents of memory location with offset 0025 into AX.
b. Copy AX into memory location with offset 0FFF.
c. Move contents of memory location with offset 0010 to memory location with offset 002F.

Exercise 2: Write a program that swaps the value of two registers using a data label i.e. given these initial values: ax=100, bx=200. After rotation, ax=200, bx=100.

Exercise 3: Develop an assembly program that reads 1 number each from 5 different data labels to a register and stores their sum in some other memory location labeled as **result**.

Exercise 4: Move a number (6 for this question) from a memory location in AX, move 4 into BX then find num * 4 using ADD instruction and then divide that answer by 3 using SUB instruction. Store the results of multiplication and division (quotient) at different memory locations labeled as "**mresult**" and "**dresult**".

Exercise 5: Write a program to add three numbers using byte variables. What will happen if we add these numbers in ax?

Exercise 6:

a) Complete the following table:

Register	Default associated segment	Flexible (yes/no)
BX, SI, DI		
IP		
BP		
SP		

b) If BX= FFFE then what will be the effective logical address for [bx+0003h]?
c) If BX=0100h (some logical address in Data Segment), DS=FFF0h (base address of data segment). Then what will be the physical address generate by [bx+0x0100]?
d) How the following data will look like in the memory:
1. dw 5
2. dw 0A0Bh
3. dd: 0A0B0C0D

Post Lab

Exercise 5: Write a program to generate first 10 terms of the Fibonacci Series. The generated terms are to be placed at memory location named "**Fib**" using indirect addressing