**Computer Organization and Assembly Language Programming Lab Manual 1 – Setup and Running your first program**

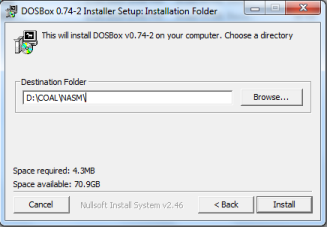
**Activity 1: Setup**

Follow this link to get started: https://github.com/ASD0x41/Assembly-Programming-Package If above link doesn’t work for you then follow this:

**Suggested version of NASM** (you will need to install this in coming instructions)**:** https://www.nasm.us/pub/nasm/releasebuilds/2.15.05/

**Option 3:**

Make a separate folder COAL and NASM in your machine for example “D:\COAL\NASM”. Visit the link given below. Download and install NASM, AFD and DOSBOX, according to the instructions, in your NASM folder.

http://wetolearn.blogspot.com/2013/09/setting-up-afd-nasm-and-dosbox-for-8086.html 

After installations double click “DOSBox 0.74-2 Options.bat” file and at the end of the file paste following lines:

| MOUNT C D://COAL//NASM  C: |
| --- |

(We are mounting C drive to our folder where we have saved AFD and we will save our .asm file in this directory)

You may follow links given below for **Video Tutorial of Setup:**

Tutorial part

1: https://drive.google.com/file/d/1N3lWL8hsN0ZbhF3tlNwCWWwjJ\_eHQqk6/view?usp=sharing

Tutorial part 2: https://drive.google.com/file/d/10p8qyaOVOwF5lDighrMKE-uNYQX c3bL/view?usp=sharing

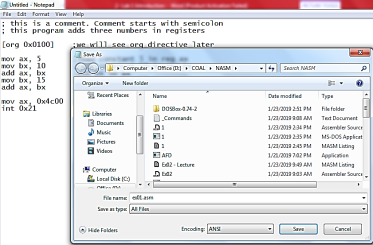
AFD: https://drive.google.com/file/d/1eXnD3JEwBelFiJT6iIk7gluudV2Fu\_iX/view?usp=sharing NASM: https://drive.google.com/file/d/1ZoeE2MxjNaK7DdJKCacYfAJyn006MI\_F/view?usp=sharing Dosbox: https://drive.google.com/file/d/1DnaDIk4RoGBFDP1y4Dr3q89xwM3gx1d1/view?usp=sharing **Activity 2: Running your First Program**

Follow these step in order to run your first program:

1- Copy/paste following code in notepad

| ; this is a comment. Comment starts with semicolon  ; this program adds three numbers in registers  [org 0x0100] ;we will see org directive later  mov ax, 5 ; AX = 5  mov bx, 10 ; BX = 10  add ax, bx ; AX = AX + BX  mov bx, 15 ; BX = 15  add ax, bx ; AX = AX + BX  mov ax, 0x4c00 ;terminate the program  int 0x21 |
| --- |

2- Save this file as “ex01.asm” in your NASM folder e.g. “D:\COAL\NASM”:



3- Go to NASM installation directory ( e.g. “D:\COAL\NASM”). Double click **nasmpath.bat** (batch file) and type following command there. (Your .asm file and nasm should be in one folder) **nasm ex01.asm -o ex01.com -l ex01.lst**

4- Above command will assemble your code and create ex01.com and ex01.lst files. Open ex01.lst file in notepad.

a. What is opcode of instruction “mov ax, someConstant”

**“B8{someConstant}”**

b. Verify the above opcode everywhere the instruction has been used.

c. What does “B80500” mean?

**Moving or Storing the constant value 5 in the Accumulator Register AX.**

d. Verify the opcode of instruction “mov bx, someConstant” throughout the machine code.

**“BB{someConstant}”**

e. What is the offset of first instruction?

**00000000**

f. Why are offsets of second and third instructions 3 and 6?

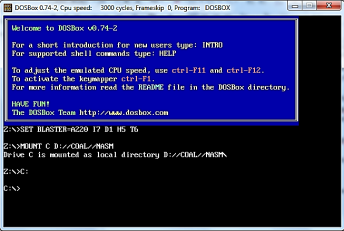
**Since each instruction takes up 3 bytes in the memory. The instruction B80500 is divided into B8, 05, 00 and is stored in 3 bytes respectively, making the offset of the next instruction 3/**

g. What should be the size of ex01.com file?

**The .lst file ends at 00000010 offset with the last instruction being of 2 bytes. Thus the total size of the file is 18 bytes.**

h. Right click ex01.com and verify its size.

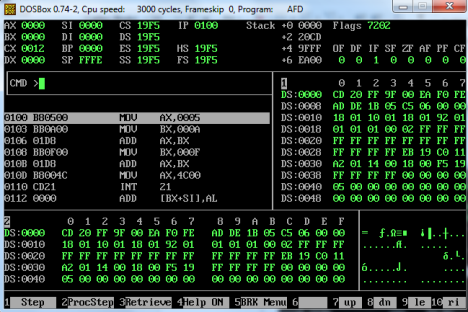
5- Open DOSBox (by double clicking dosbox.exe), following window will appear:



6- Write following command and press enter.

**Afd ex01.com**

(Your AFD.exe should be in same directory where we have installed everything)



7- Above command will open the debugger and load your ex01.com file in it.

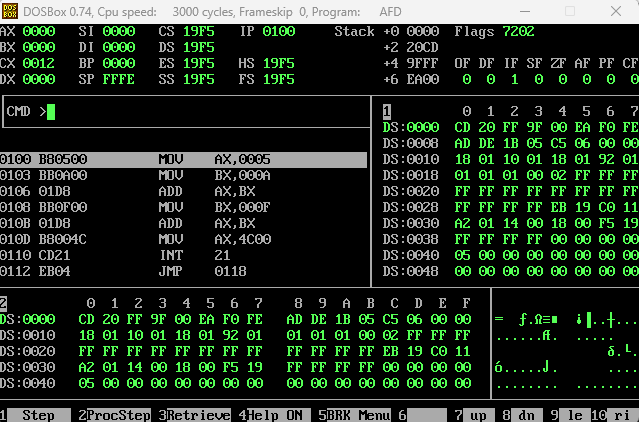
a. What is the value of IP register? And what will be its effect?

**0100 in hexadecimal**

b. Note the initial values of data registers

c. Press F1 and watch the values of data registers

**Solution:**

****

**Activity 3: Explore different functions available in debugger** (after completing activity 4 and 5).

**Activity 4:** Modify this program to generate the sum of first five entries of table of 3, using registers, and watch its execution in the debugger.

**Code:**

| [org 0x0100]  MOV AX, 0  MOV BX, 3  ADD AX, BX  MOV BX, 6  ADD AX, BX  MOV BX, 9  ADD AX, BX  MOV BX, 12  ADD AX, BX  MOV BX, 15  ADD AX, BX  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Activity 5:** Write a program in assembly language that calculates the square of six by adding six to the accumulator six times.

**Code:**

| [org 0x0100]  MOV AX, 0  ADD AX, 6  ADD AX, 6  ADD AX, 6  ADD AX, 6  ADD AX, 6  ADD AX, 6  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Activity 6:** Do exercise 4 and 5 with Byte size operations (for example, AL, AH etc.)

**Code for Ex#4:**

| [org 0x0100]  MOV AX, 0  MOV BL, 3  ADD AL, BL  MOV BL, 6  ADD AL, BL  MOV BL, 9  ADD AL, BL  MOV BL, 12  ADD AL, BL  MOV BL, 15  ADD AL, BL  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Code for Ex#5:**

| [org 0x0100]  MOV AX, 0  ADD AL, 6  ADD AL, 6  ADD AL, 6  ADD AL, 6  ADD AL, 6  ADD AL, 6  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Practice Problems**

1- Listing files of two programs are given below. What will be the size of their com files? (\_\_\_\_\_\_\_\_\_\_\_\_)16 and (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)10

Listing File 1:

| 1 ; a program to add ten numbers  2  3 [org 0x0100]  4  5 00000000 BB[1D00] mov bx, num1  6 00000003 B90A00 mov cx, 10  7 00000006 B80000 mov ax, 0  8  9 00000009 0307 l1: add ax, [bx]  10 0000000B 81C30200 add bx, 2  11 0000000F 81E90100 sub cx, 1  12 00000013 75F4 jnz l1  13  14  15  16 00000015 A3[3100] mov [num1], ax  17  18 00000018 B8004C mov ax, 0x4c00 ;terminate the program 19 0000001B CD21 int 0x21  20  21 0000001D 010002000300040005- num1: dw 1, 2, 3, 4, 5, 1, 2, 3, 4, 5 22 00000026 000100020003000400-  23 0000002F 0500 |
| --- |

**Size:**

(32)16 and (50)10

Listing File 2:

| 1 ; Multiplication  2  3 [org 0x0100] |
| --- |

| 4  5 00000000 E90600 jmp start  6  7 00000003 4E9D a: DW 0x9D4E  8 00000005 C3A54E9D b: DD 0x9D4EA5C3  9  10 00000009 B8A9FC start: mov ax, 0xFcA9  11 0000000C C1E004 shl ax,4  12 0000000F C1C004 rol ax,4  13  14 ;mov [a],0x9D4E ; Error op size not specified  15 00000012 C706[0300]4E9D mov word[a],0x9D4E  16  17 00000018 C026[0300]04 shl [a],4; by default this will operate on 1 byte 18 0000001D C006[0300]04 rol [a],4  19  20 00000022 C706[0300]4E9D mov word[a],0x9D4E  21 00000028 C126[0300]04 shl word[a],4  22 0000002D C106[0300]04 rol word[a],4  23  24 00000032 B8004C mov ax, 0x4c00 ;terminate the program  25 00000035 CD21 int 0x21 |
| --- |

**Size:**

(38)16 and (56)10