LAB MANUAL 01

**Introduction to 8086 Architecture and Register Set, Use of Emulator and executing Assembly Language programs**

**What is a Microprocessor?**

A device that takes data as its input, process the data and outputs the result.

The important characteristics of a microprocessor are the widths of its internal and external address bus and data bus (and instruction), its clock rate and its instruction set.

**What is a Register?**

Small numbers of high-speed memory locations in a microprocessor.

## Microprocessor’s Differences:-

Different microprocessors have different specifications, the main differences they hold are:

**Instruction Set**: operations, addressing modes.

**Registers**: size, number

**Address bus**: size (number of bits).

**Data bus**: size

**Storage of Data Structures of a Microprocessor:-**

There are two ways to store the data structures. For multiple-byte data items stored in memory, the order needs to be specified:

Most Significant 8 bits at lowest address ("Big Endian"),

OR

Least Significant 8 bits at lowest address ("Little Endian")



**Registers:-**

The four data registers labeled **AX, BX, CX and DX** may be further subdivided for 8-bit operations into a high-byte or low-byte register. Thus, for byte operations, the registers may be individually addressed. So, each of these are 16 bits wide, but can be accessed as a byte or a word.

AX: known as an accumulator is used in arithmetic and logical operations.

BX: refers to the 16-bit base register.

CX: is used as a counter.

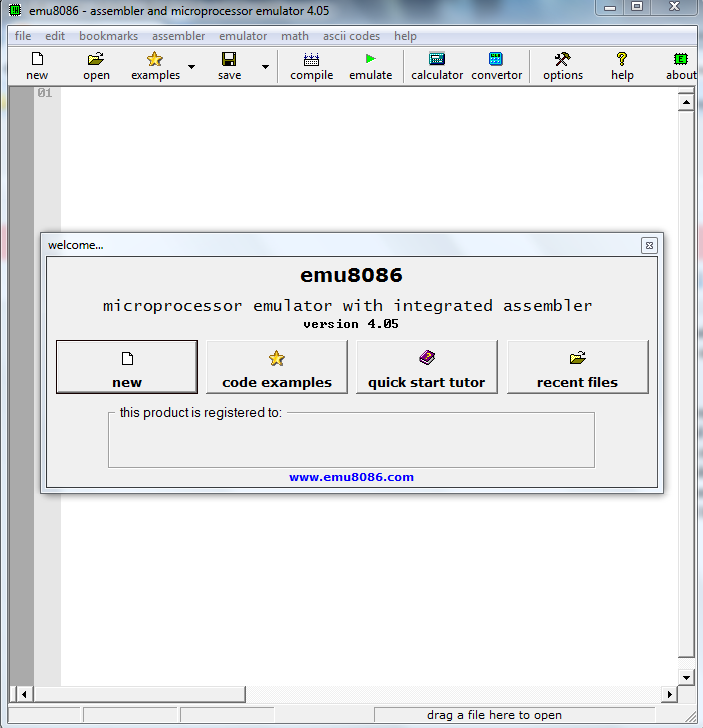
DX: Data Register.

Basic Instructions and its Use:

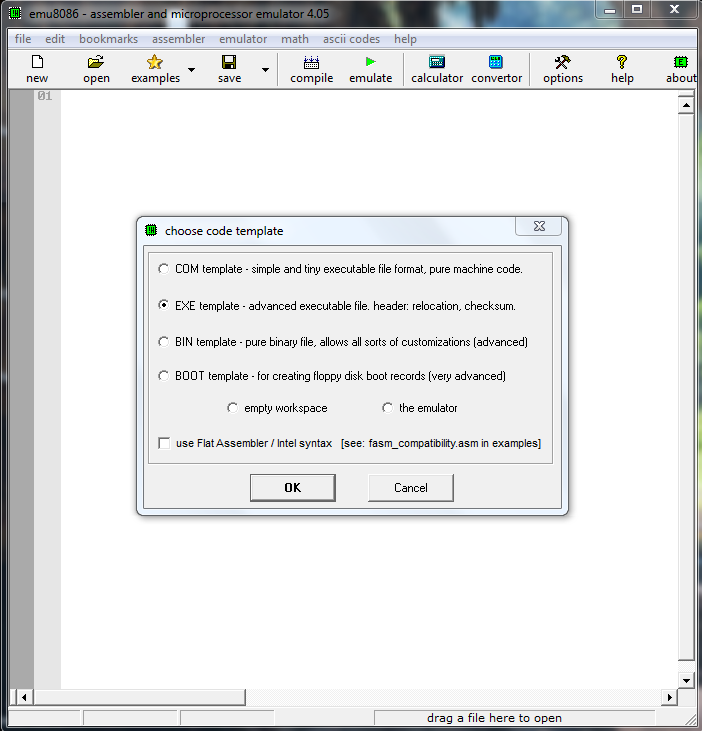
Mov and Add explained

**How to use Emulator:**

Once you open the emu8086 emulator you should see the following screen

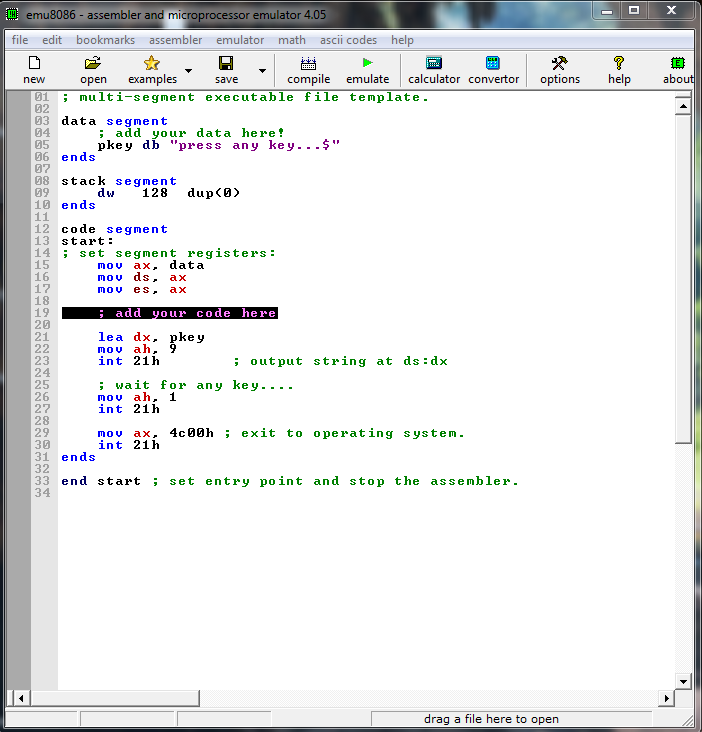


From here on you can create a new project with following options-



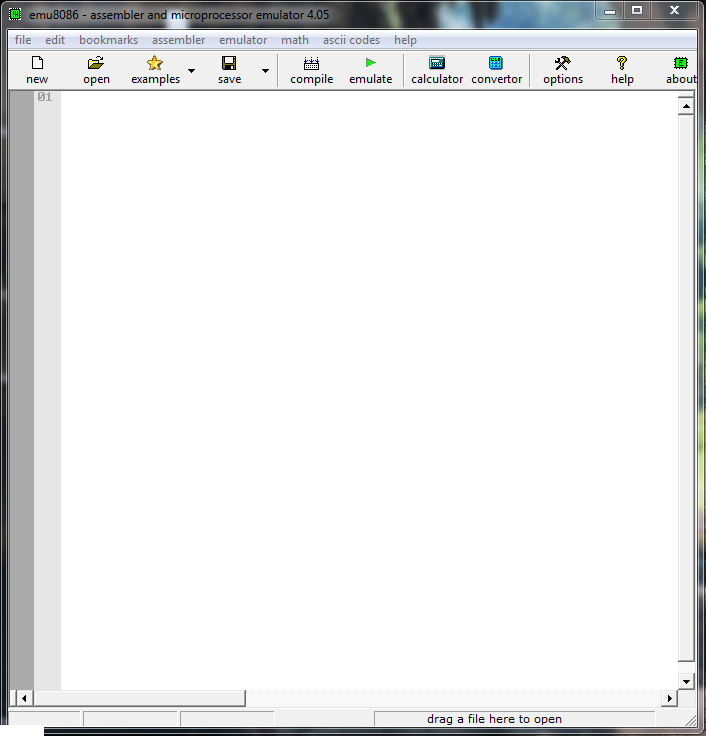
This screen allows you select what kind of file you want create- either a bin file with .com extension, an executive file with .exe extension, a pure binary file or a boot file. Once you select the type the emu8086 opens the editor with a readymade template of the kind of file you wanted to create.

Consider that you selected .exe template then the editor opens with default template as shown below,



From here on you can write the code in the add your code here section, change the code, data segment part and define constant or variables.

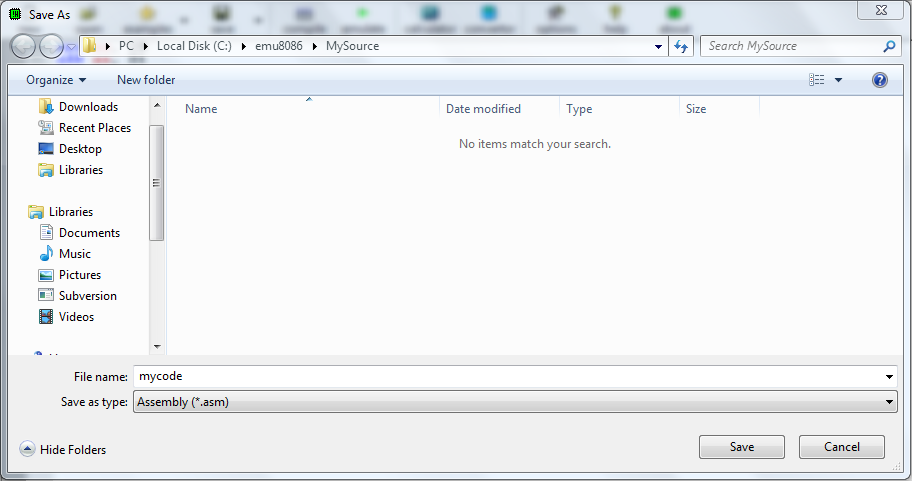
If you select cancel in the above default template selection then it opens just the blank editor as shown below,



From here you can write you assembly program. For example, consider the simple program that loads the accumulator with some values, add new value to the content of the accumulator and so on.

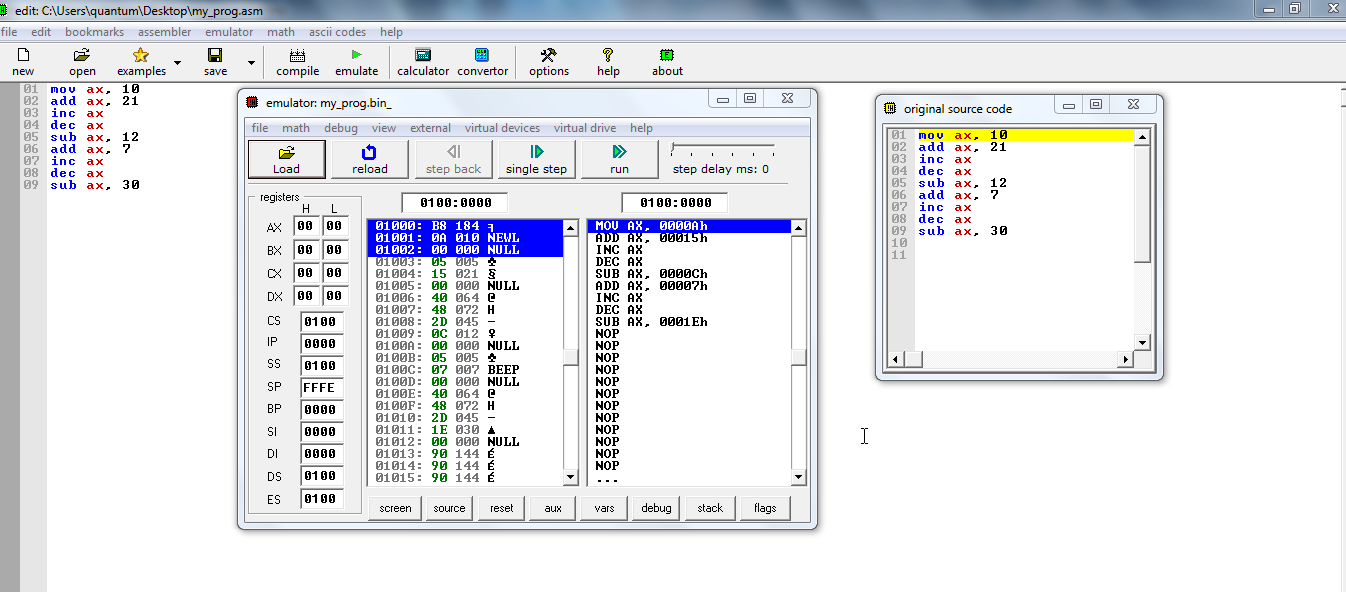
mov ax, 10   
add ax, 21   
inc ax       
dec ax       
sub ax, 12   
add ax, 7   
inc ax       
dec ax      
sub ax, 30

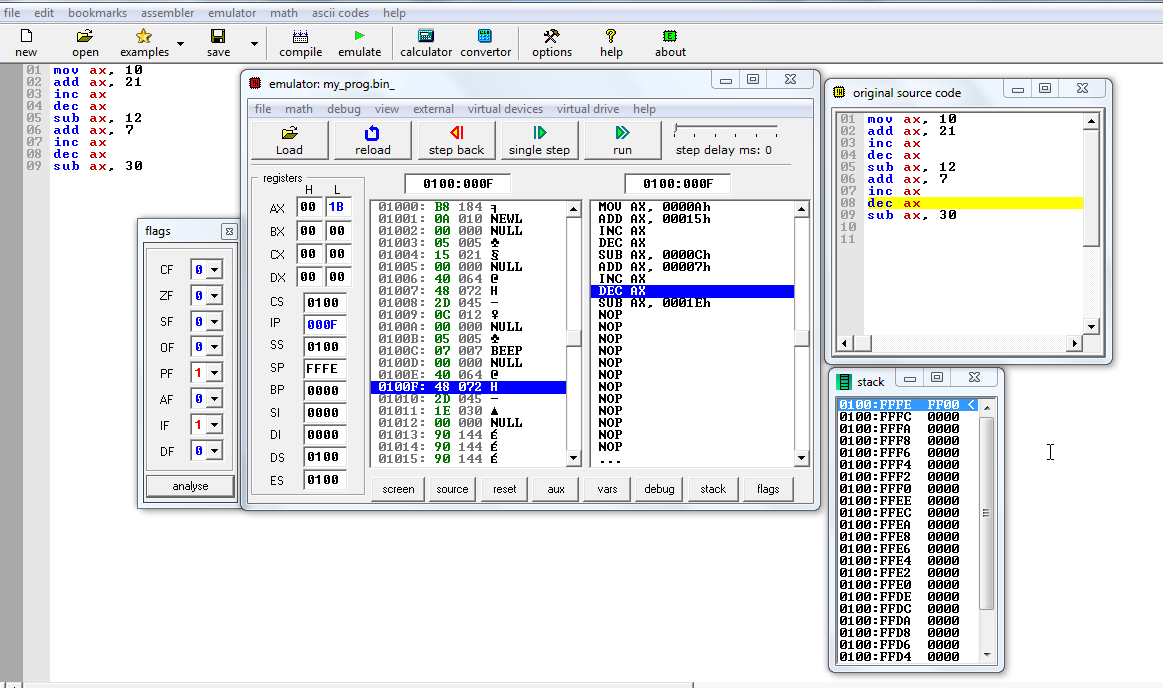
Once you have typed in your program, you should save the program by going to File>Save or File>save as or just by clicking the save button on the toolbar. Emu8086 will save it the newly created file with .asm extension in its default location(C:\emu8086\MySource) but you can browse to a folder where you want to save the file.



Once you have saved the file, you can emulate the program by clicking on the emulate button on the toolbar to see the behavior of the program. That is what register has what content, the flag resisters that gets effected by the program and so on.

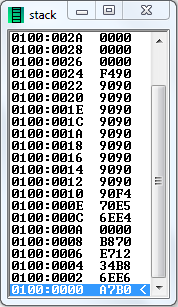
The screenshot below shows this process.



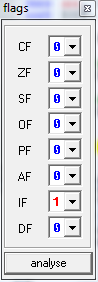


From this emulator window we can just run the program or go through the program step by step by clicking on the single step button. For the example program above, when we step through the program we can see the content of the accumulator and how it gets modified and what new values are stored. Not only can we know the content of the accumulator but also the other registers, the stack and the flag register.

The stack content is shown below,



The Flag Register is shown below,





**Lab Tasks**

**Execute the following tasks and note down all changes you observe. CLO [1]**

**TASK 1:**

Write the following code in emulator and examine the contents of registers by single stepping

MOV AL, 57H

MOV DH,69H

MOV DL, 72H

MOV BX, DX

MOV BH, AL

MOV BL, 9FH

MOV AH, 20H

ADD AX, DX

ADD CX, BX

ADD AX, 1F35H

**Task 02:**

Write a program to subtract the content of register DX from the content of register AX, then add the result to the content of CX. Set the registers to 4, 0A and 1F respectively.

**Task 03:**

Add three binary numbers to get the result of 30 in assembly ?

**Task 04:**

MOV AL, ‘9’

ADD AL, ‘5’

Have a close look at results?