



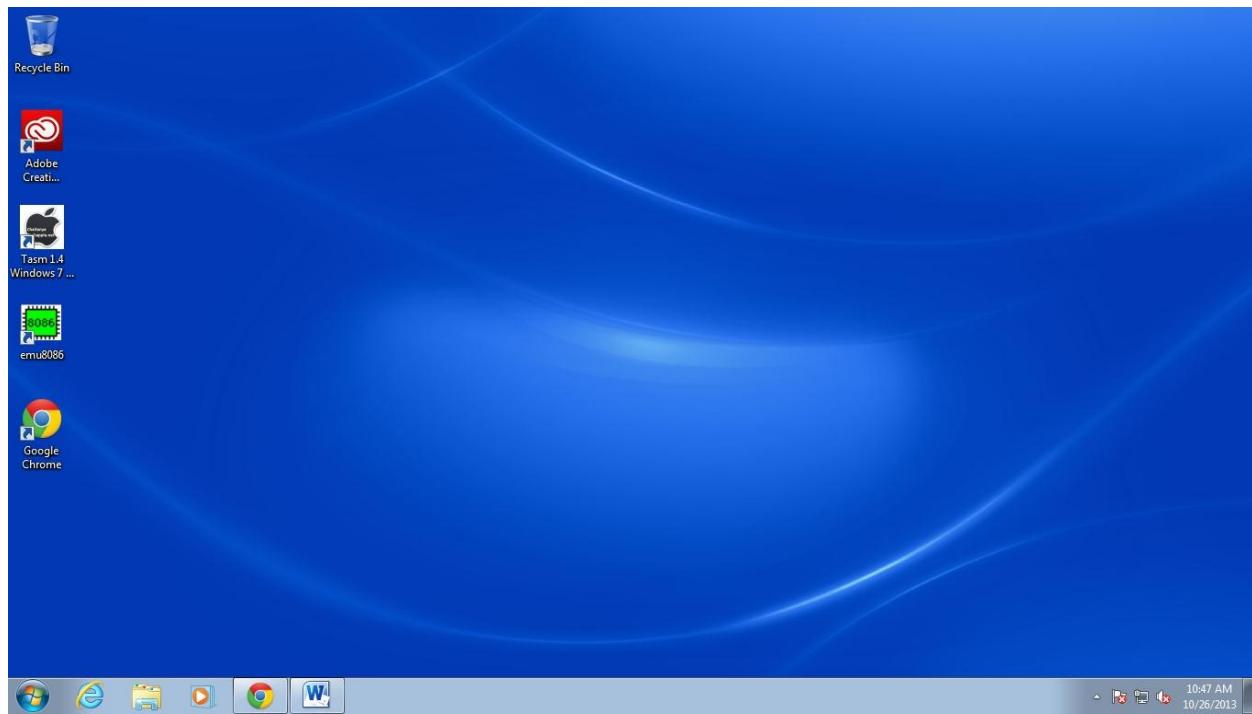
# **University of Central Punjab**

## **Introduction to Emulator**

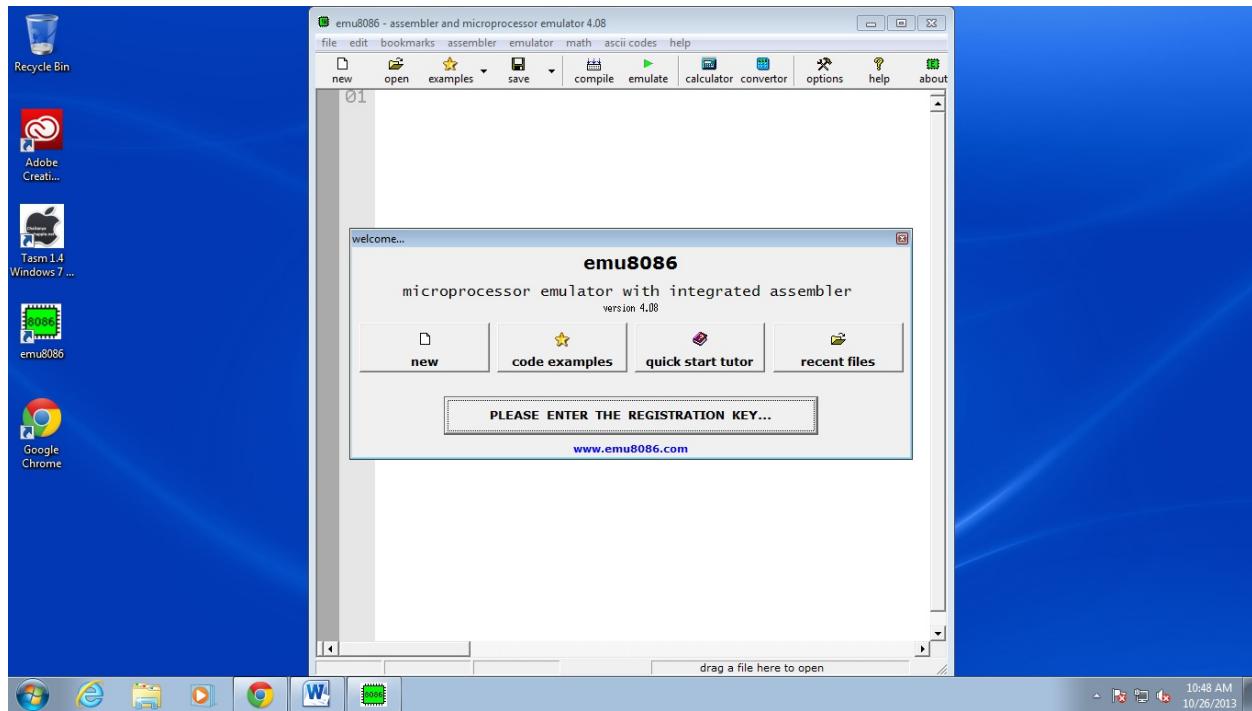
**Emu8086**

**Loading a Program and Emulating**

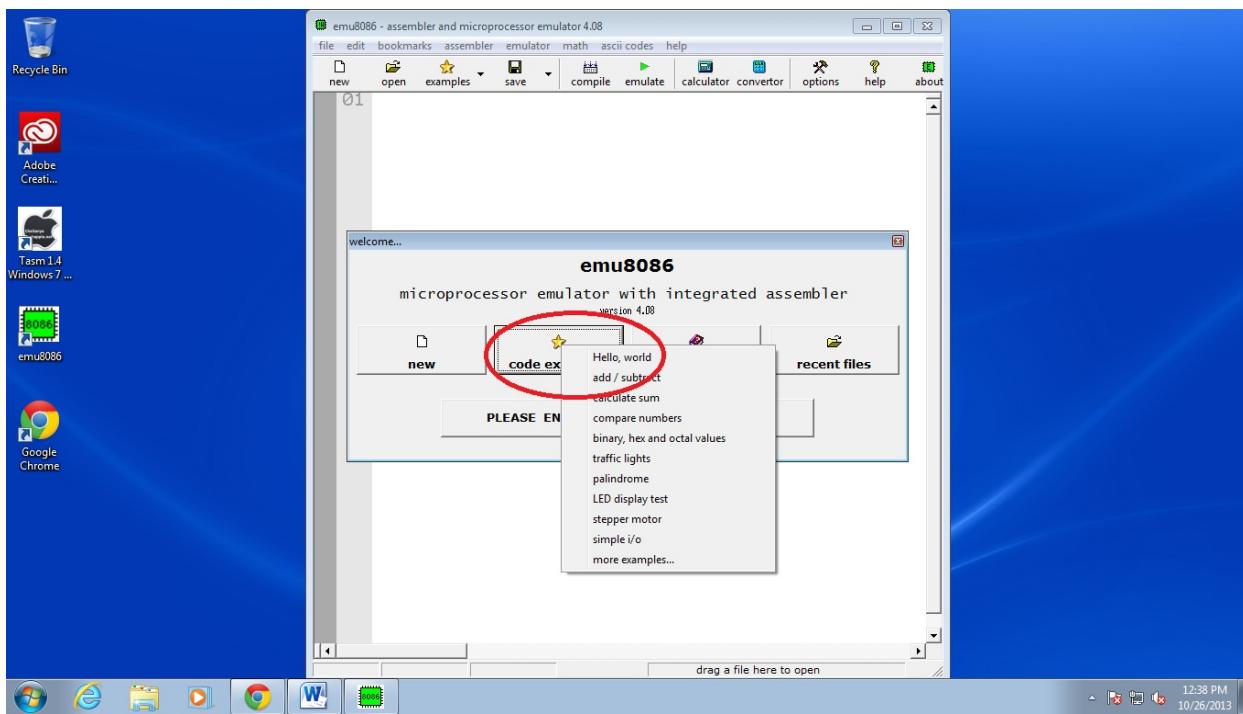
1. Locate “emu8086” on your desktop and double click to open the emulator.



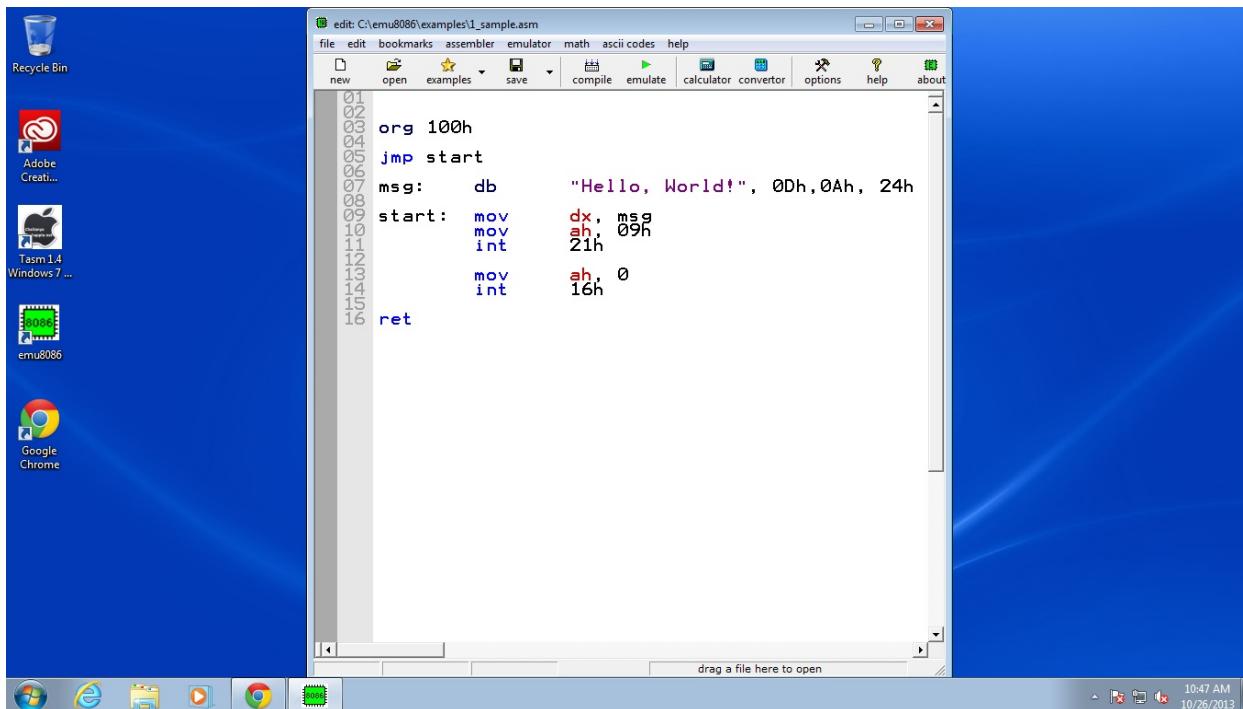
2. Click on the code examples.



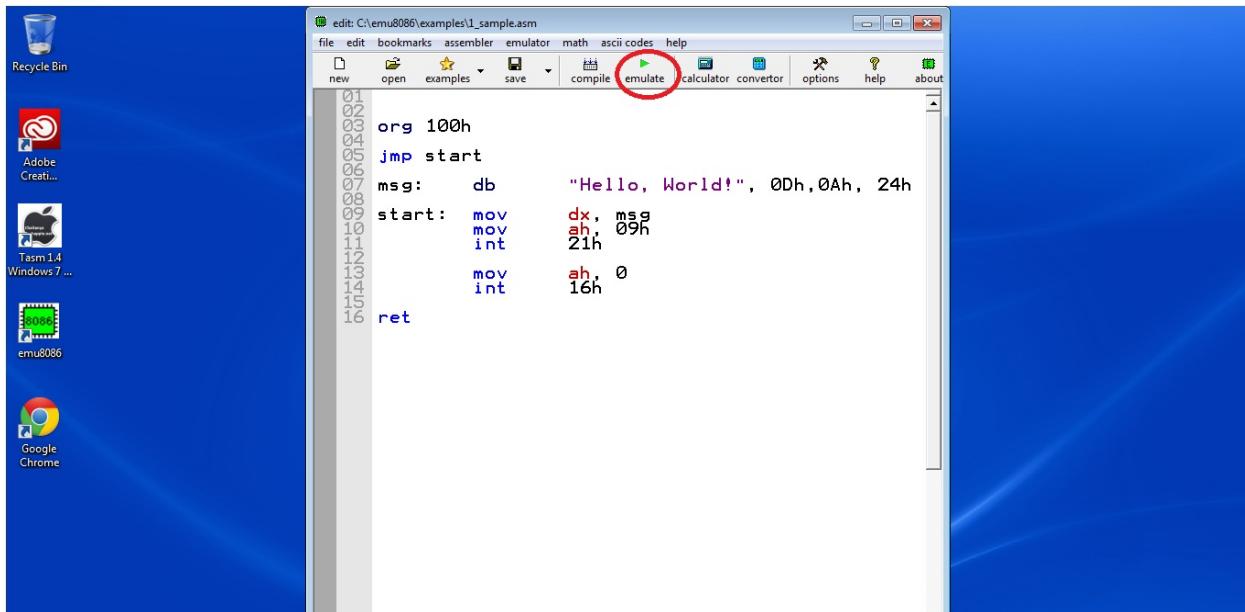
3. Click on code example and then load Hello, world.



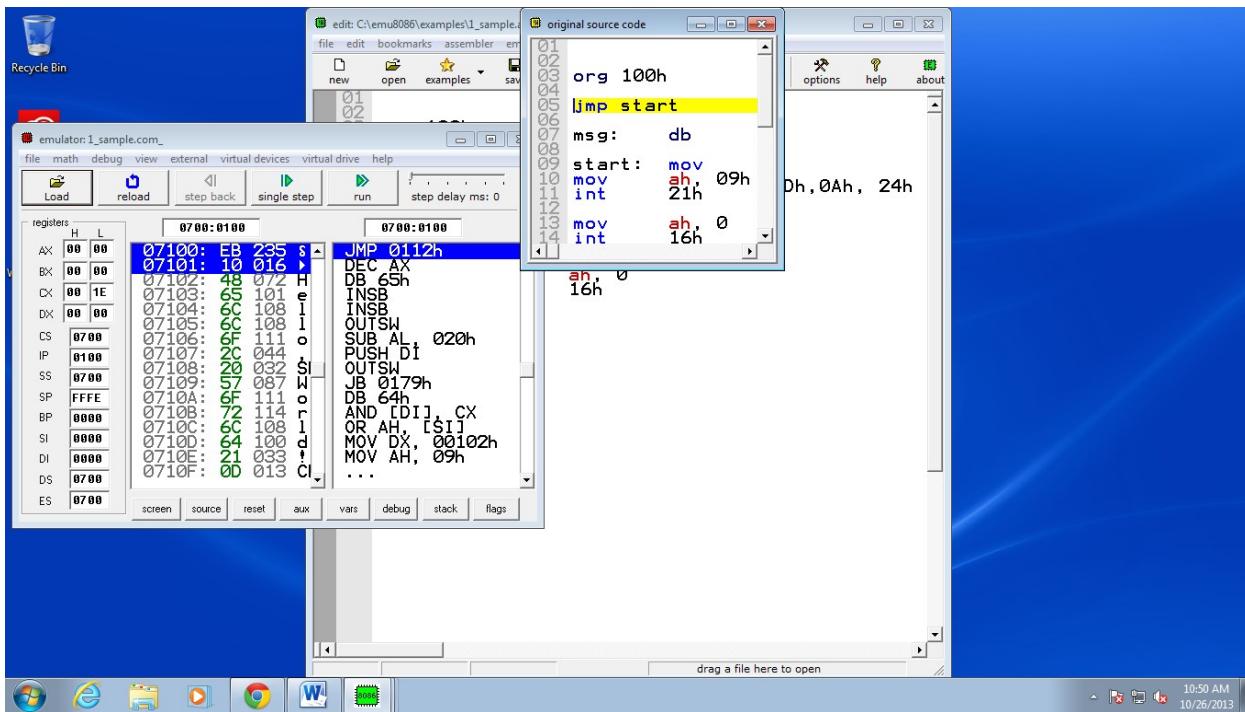
4. If you all have followed the steps, I hope now you all can see interface of emu8086.



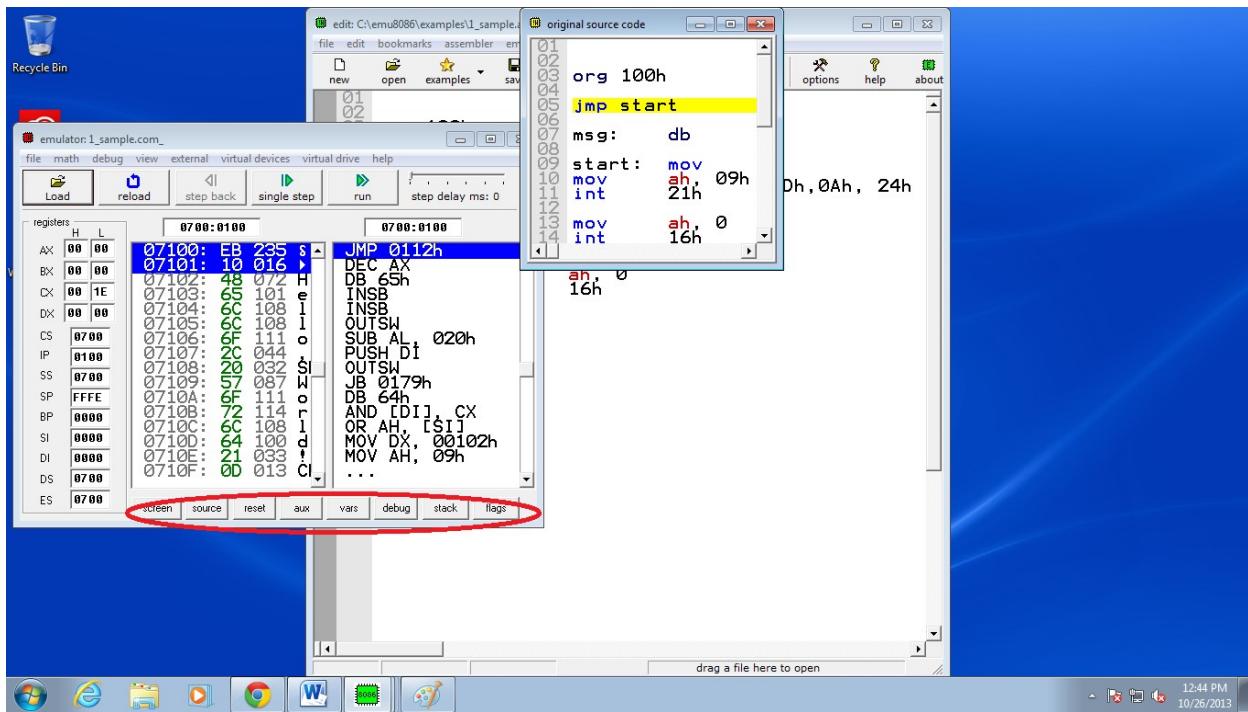
5. Click on emulate.



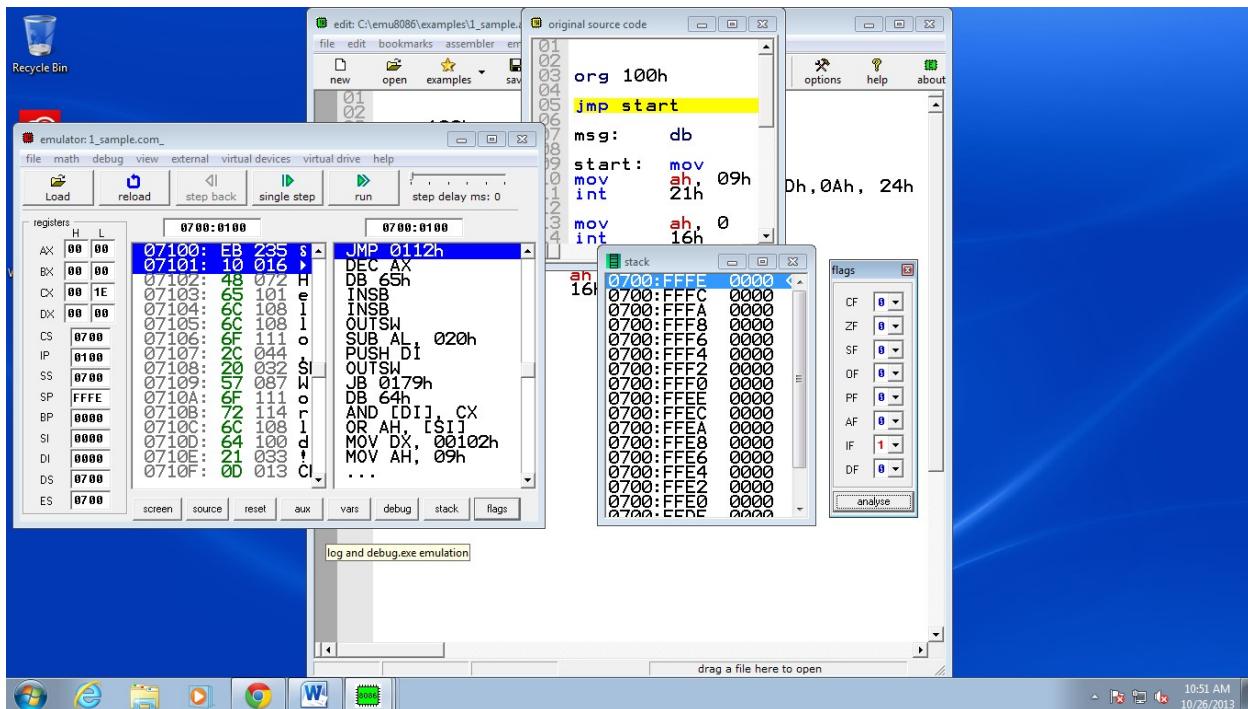
6. An interface like shown below will appear on your screen



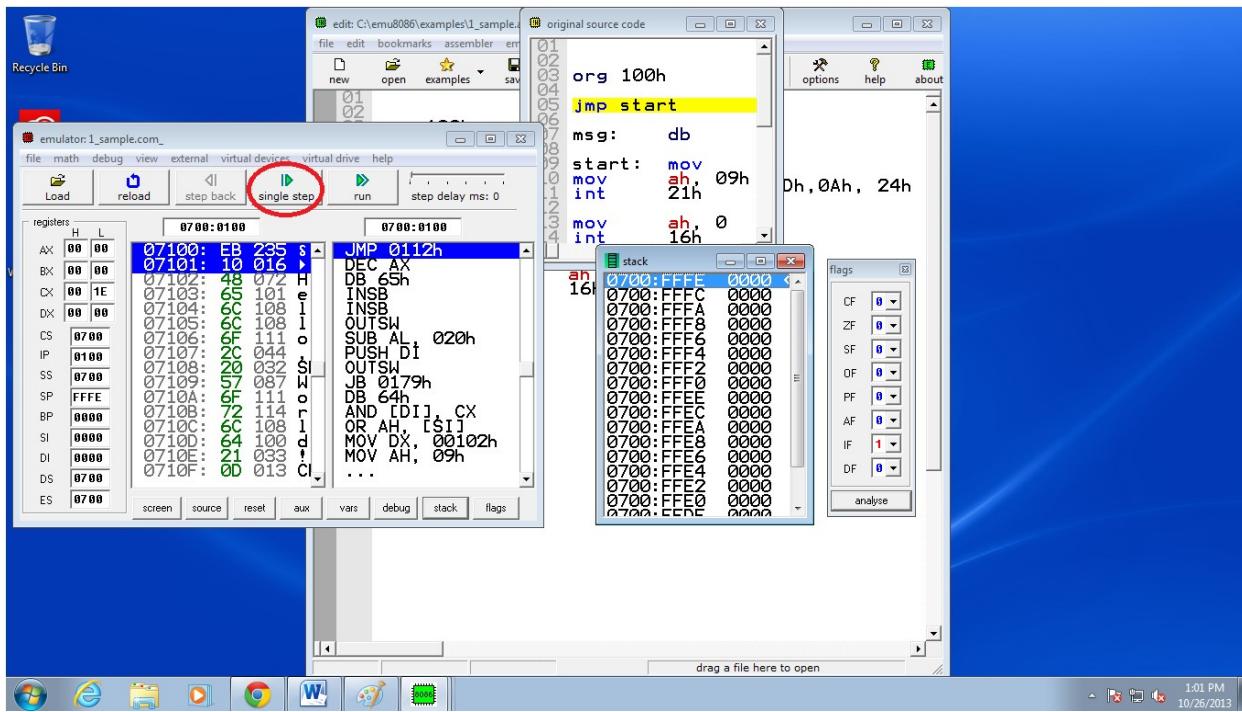
7. Locate the stack and flags icon and click to open these windows.



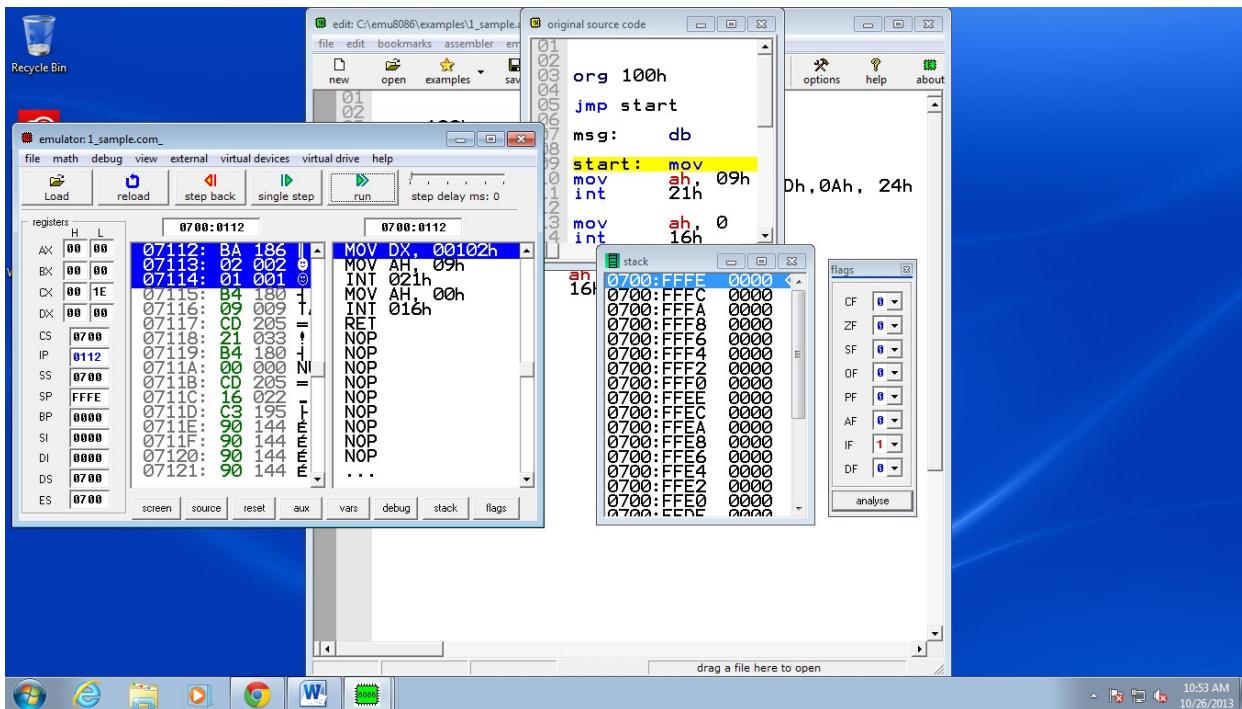
8. If you have followed the steps you can see the following screen on your desktop.



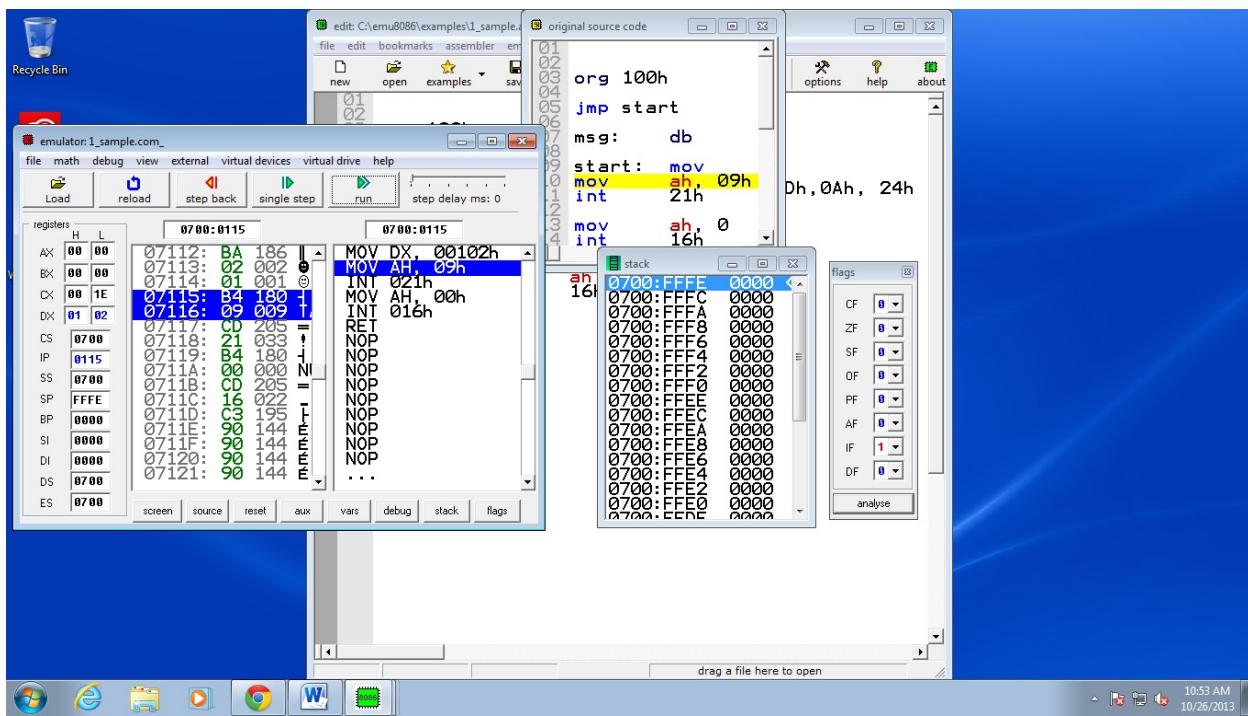
9. Click on single step to start emulation.



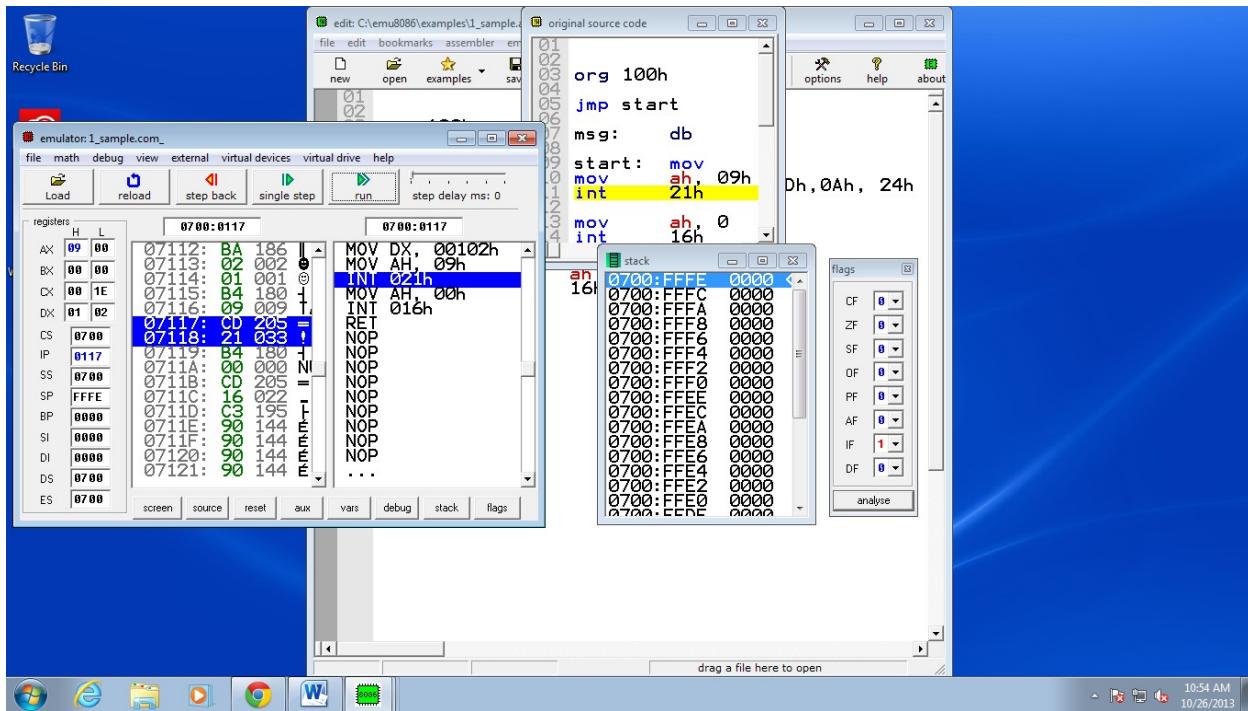
10. Now, you can see the step by step emulation of the program. The highlighted text in the following window is indicating that we are right here in the program execution.



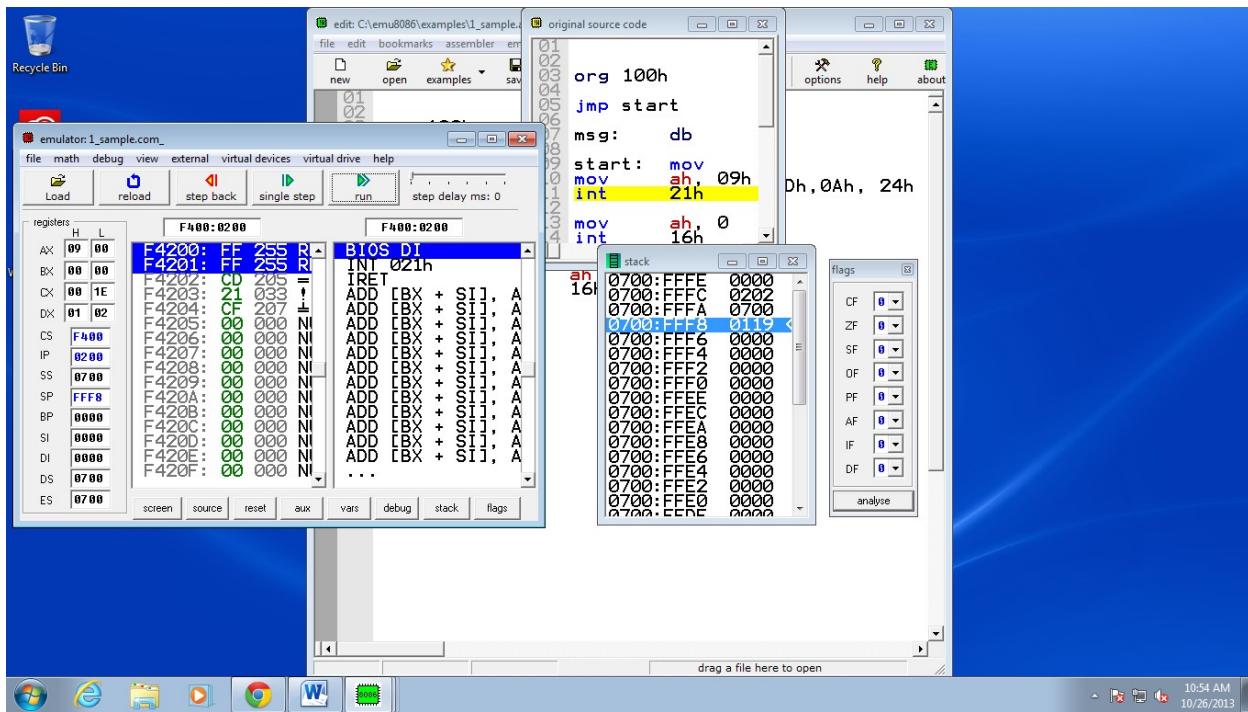
11. Note what values are in CS, SS and SP.



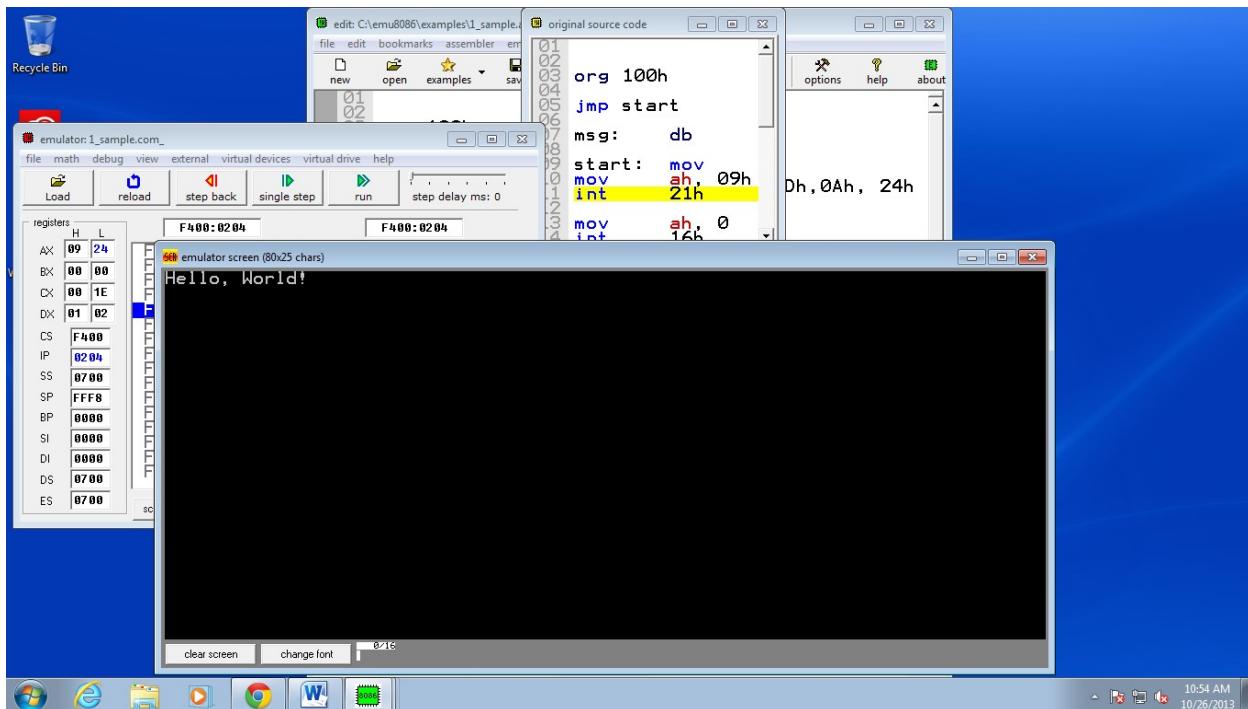
12. Note the values in auxiliary registers.



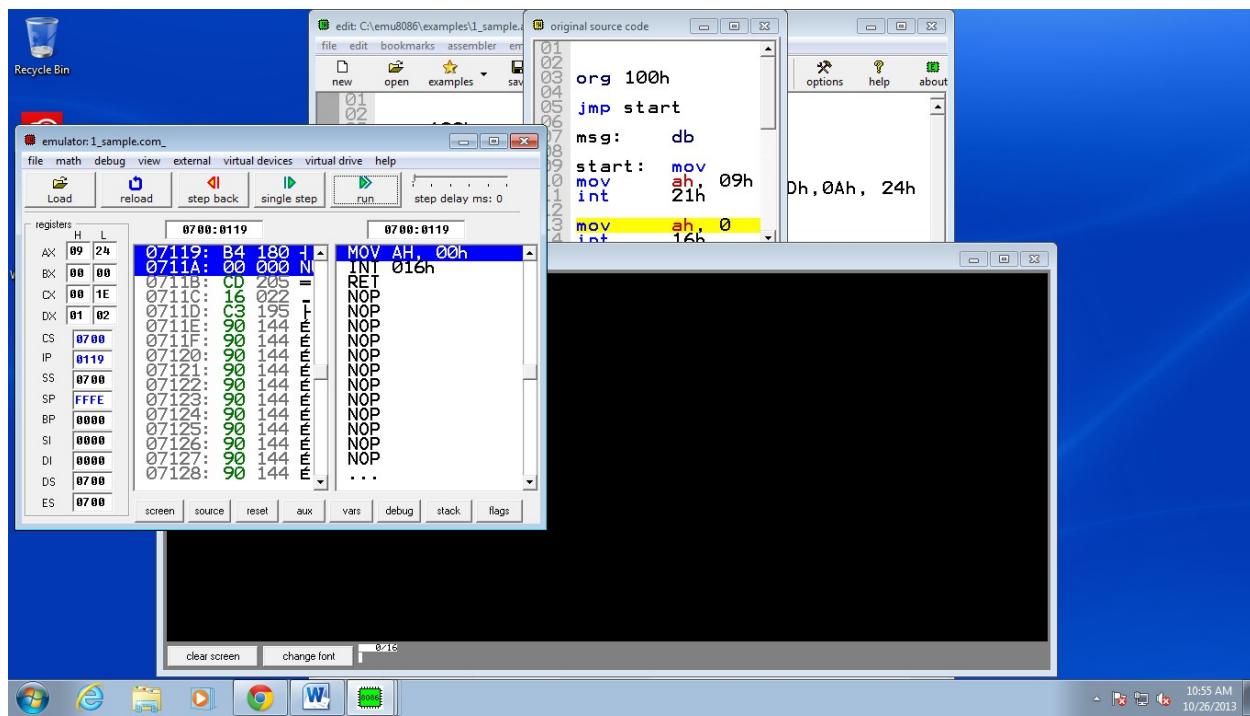
13. As you can see interrupt 21h is called here. The reason is that we want to display the “Hello, world!”



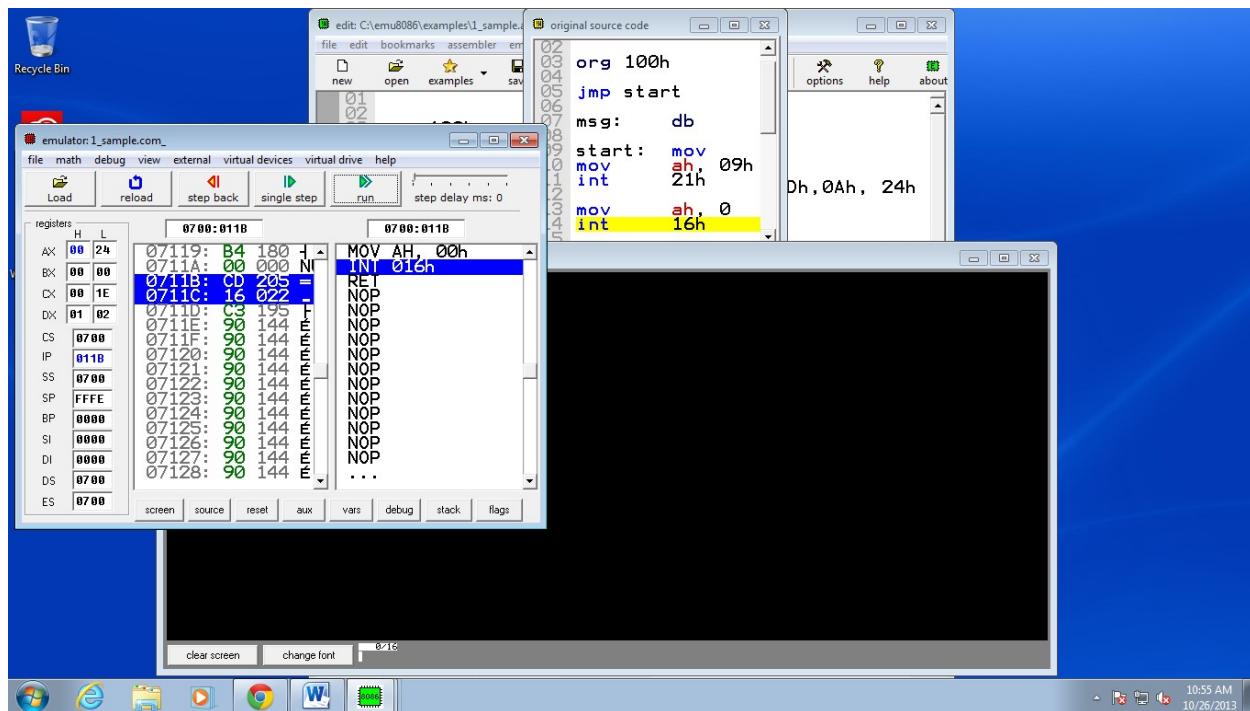
14. You can see the Hello, World! Printed on the screen.



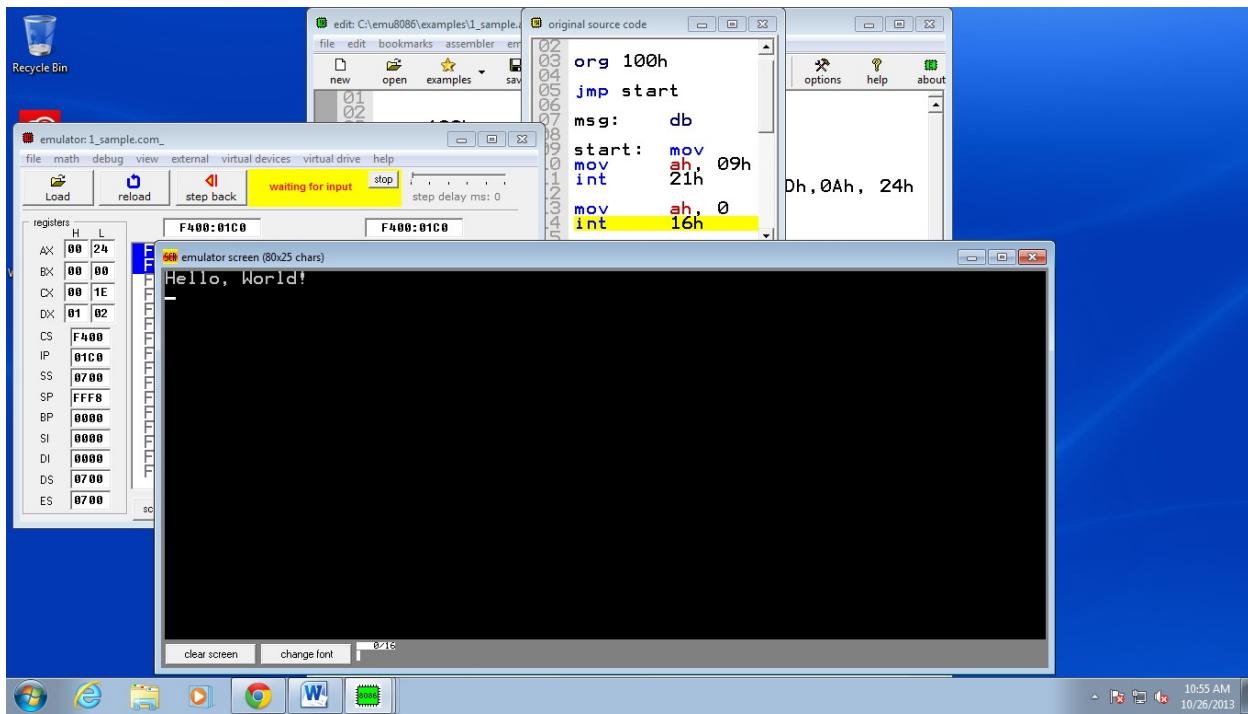
15. Now, we want the program to return the control to the OS.



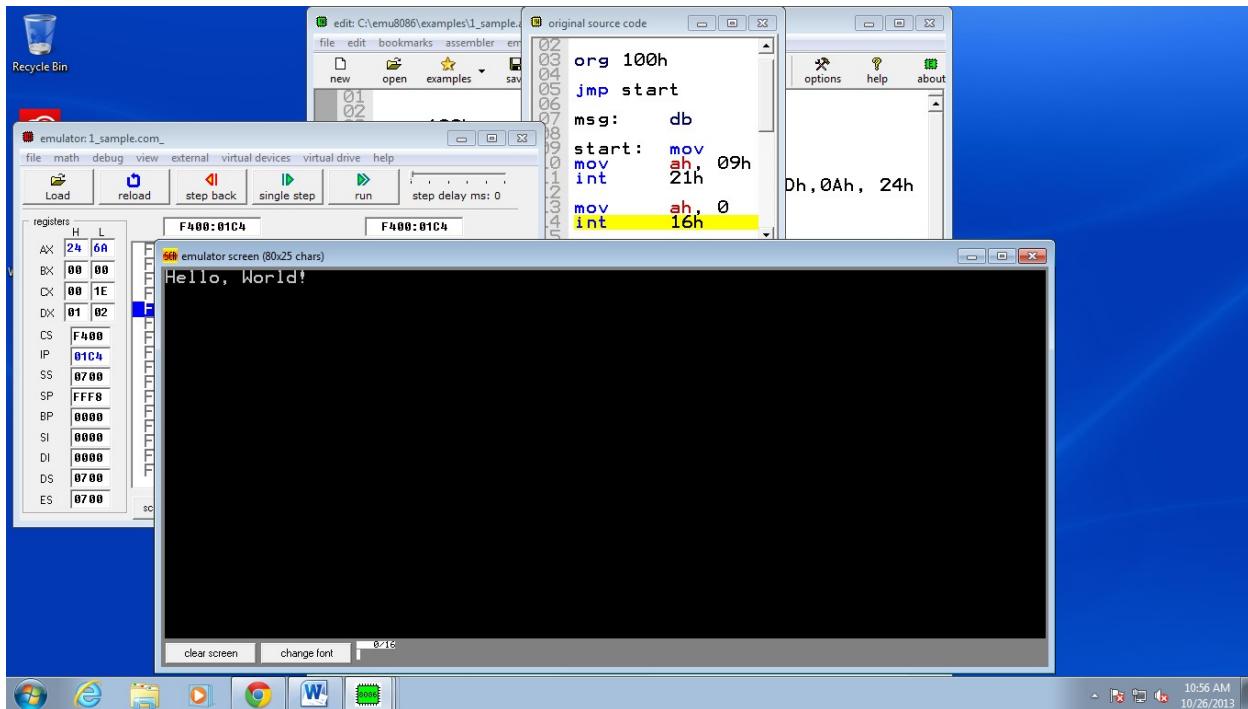
16. Keep clicking on the single step.



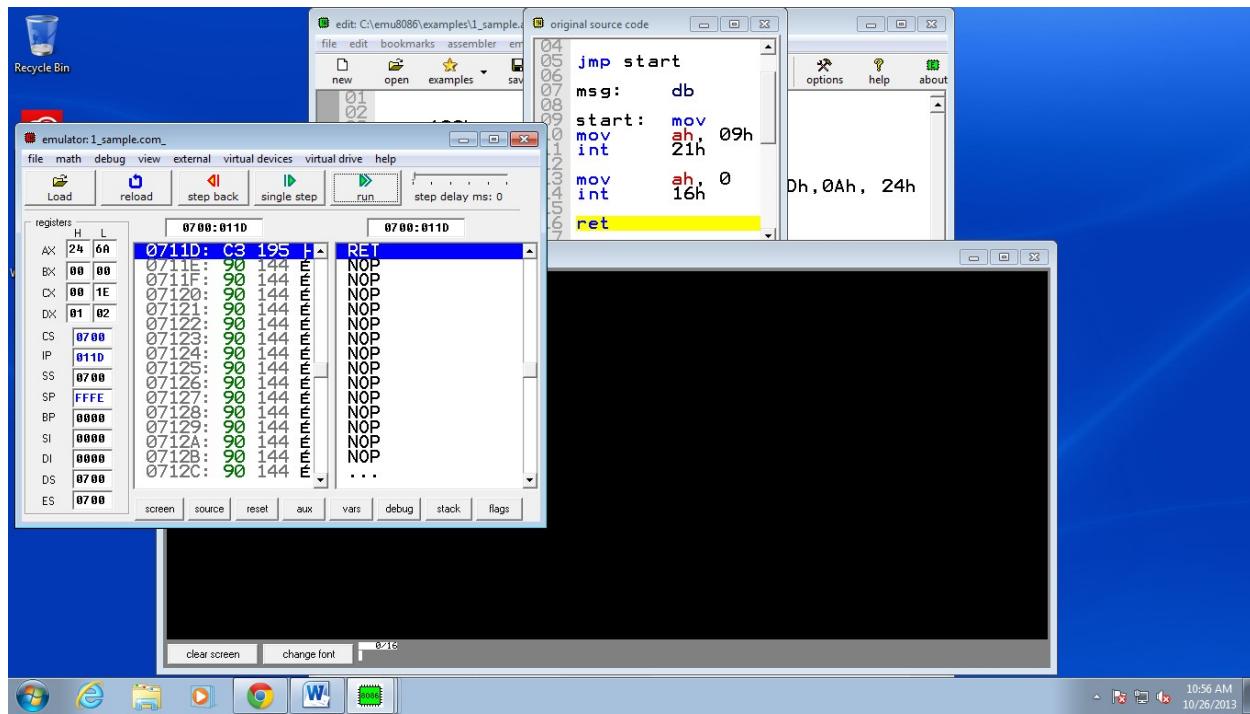
17. Waiting for key stroke.



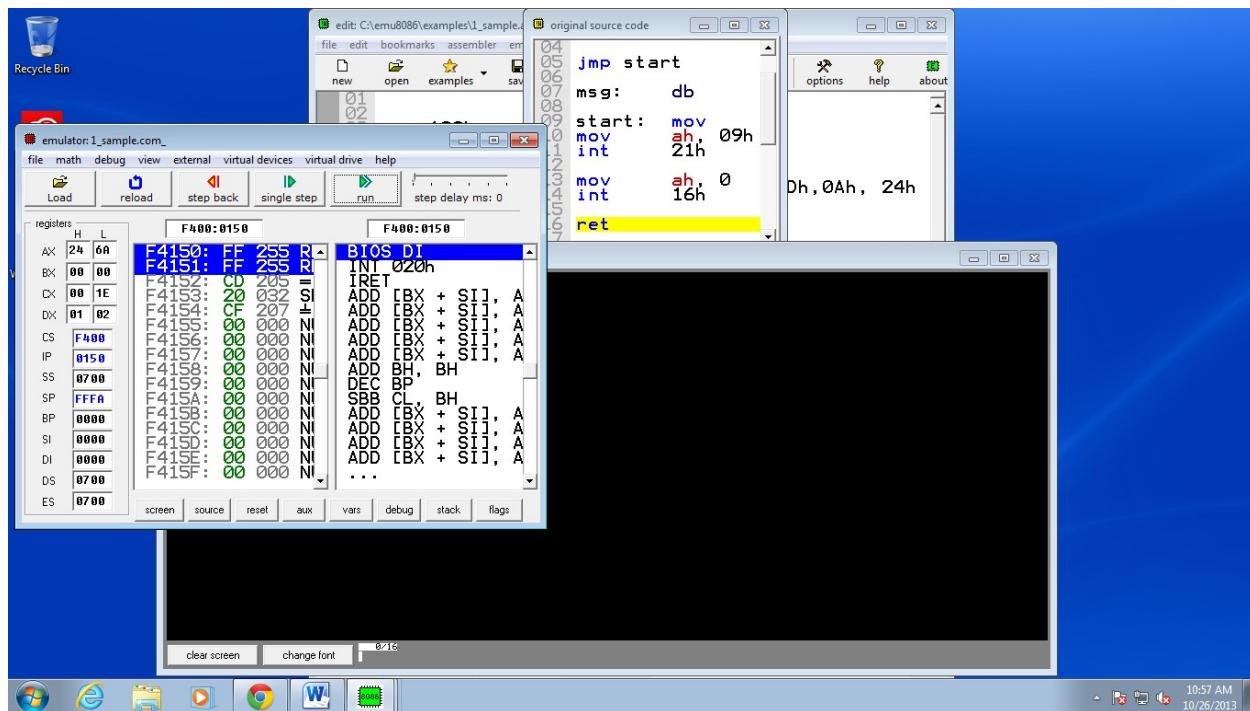
18. After pressing any key, you will have a window like this.



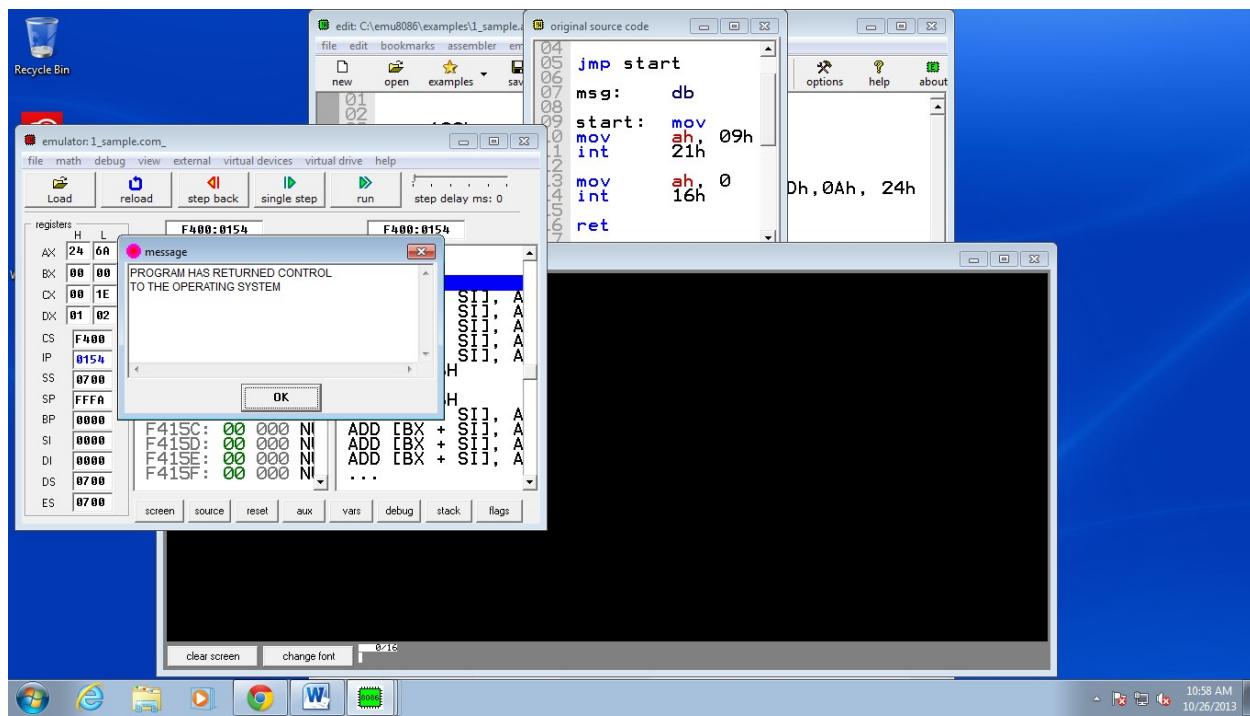
19. Now you can see the program is on ret.



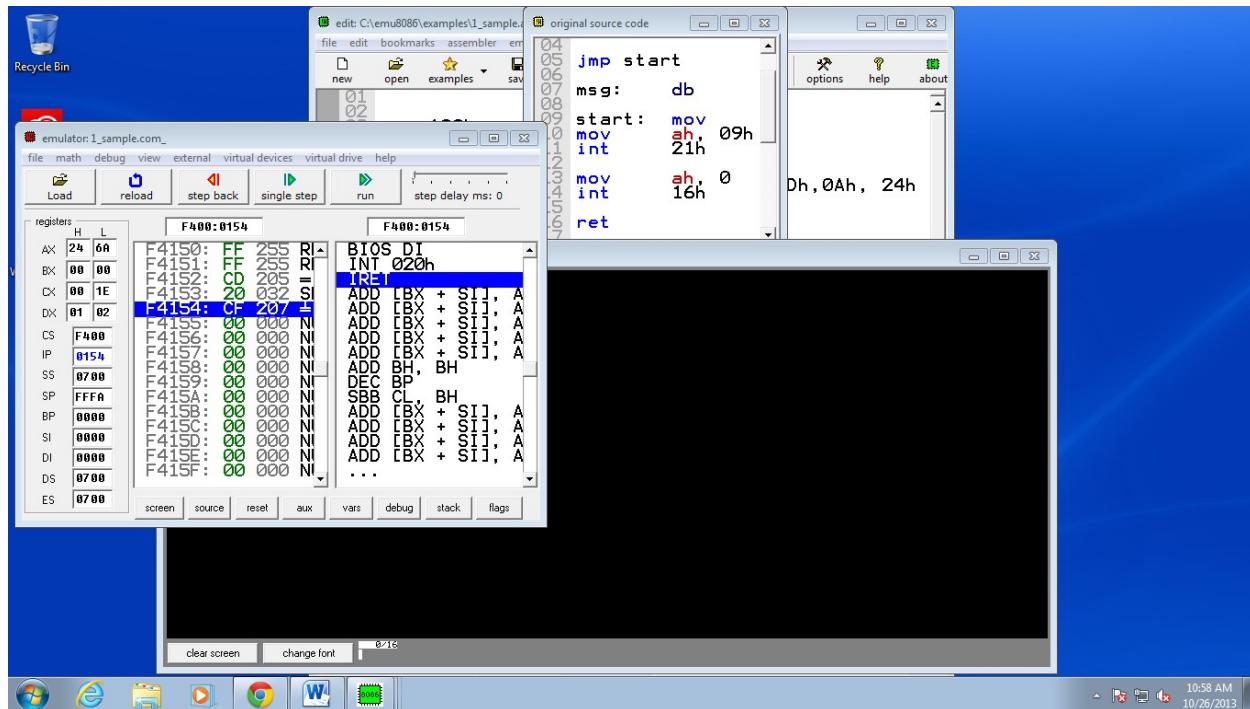
20. Return is called here.



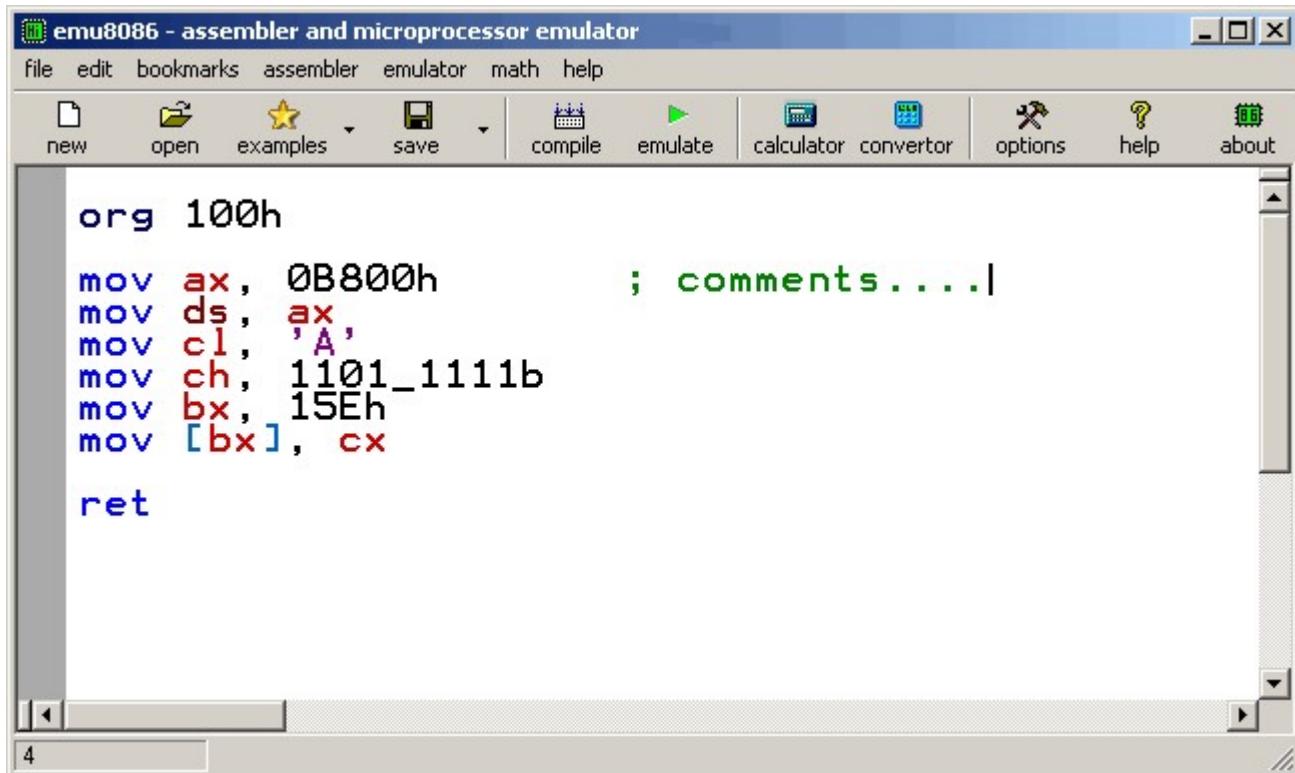
21. Program has returned the control to the OS. Click OK to proceed.



22. Congrats you have successfully debugged the first program.



## Writing your own ALPs



The screenshot shows the emu8086 software window. The menu bar includes File, Edit, Bookmarks, Assembler, Emulator, Math, and Help. The toolbar contains icons for New, Open, Examples, Save, Compile, Emulate, Calculator, Convertor, Options, Help, and About. The main text area contains the following assembly code:

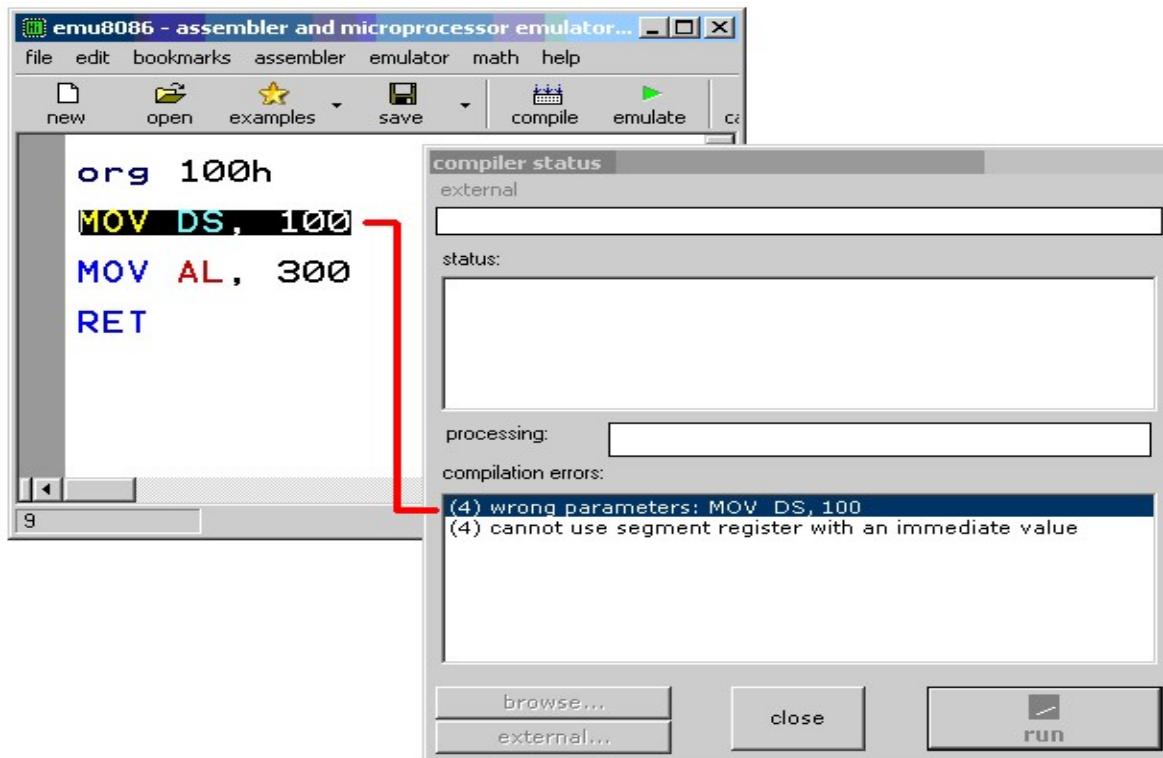
```
org 100h
mov ax, 0B800h      ; comments.....
mov ds, ax
mov cl, 'A'
mov ch, 1101_1111b
mov bx, 15Eh
mov [bx], cx
ret
```

Type your code inside the text area, and click **compile** button. You will be asked for a place where to save the compiled file.

After successful compilation you can click **emulate** button to load the compiled file in emulator.

## Error processing

Assembly language compiler (or assembler) reports about errors in a separate information window:



MOV DS, 100 - is illegal instruction because segment registers cannot be set directly, general purpose register should be used, for example

MOV AX, 100

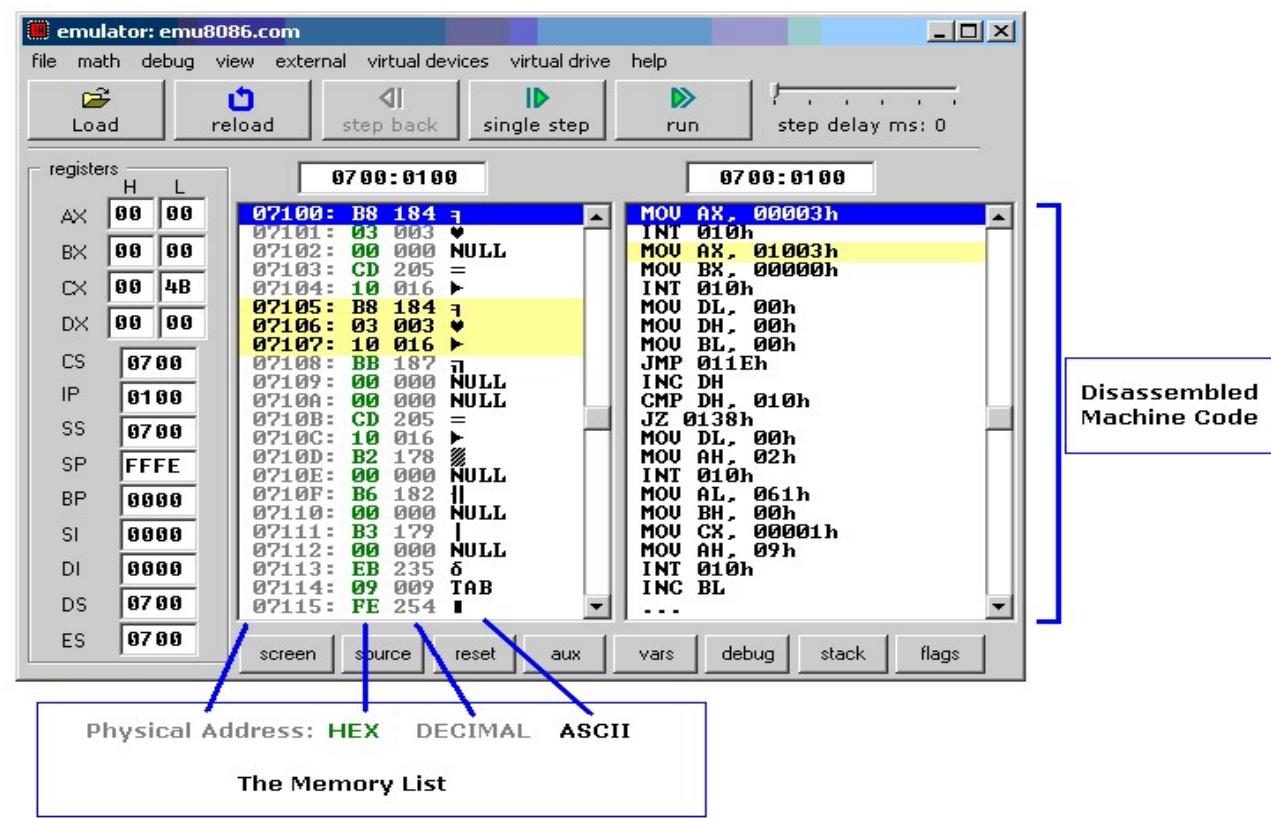
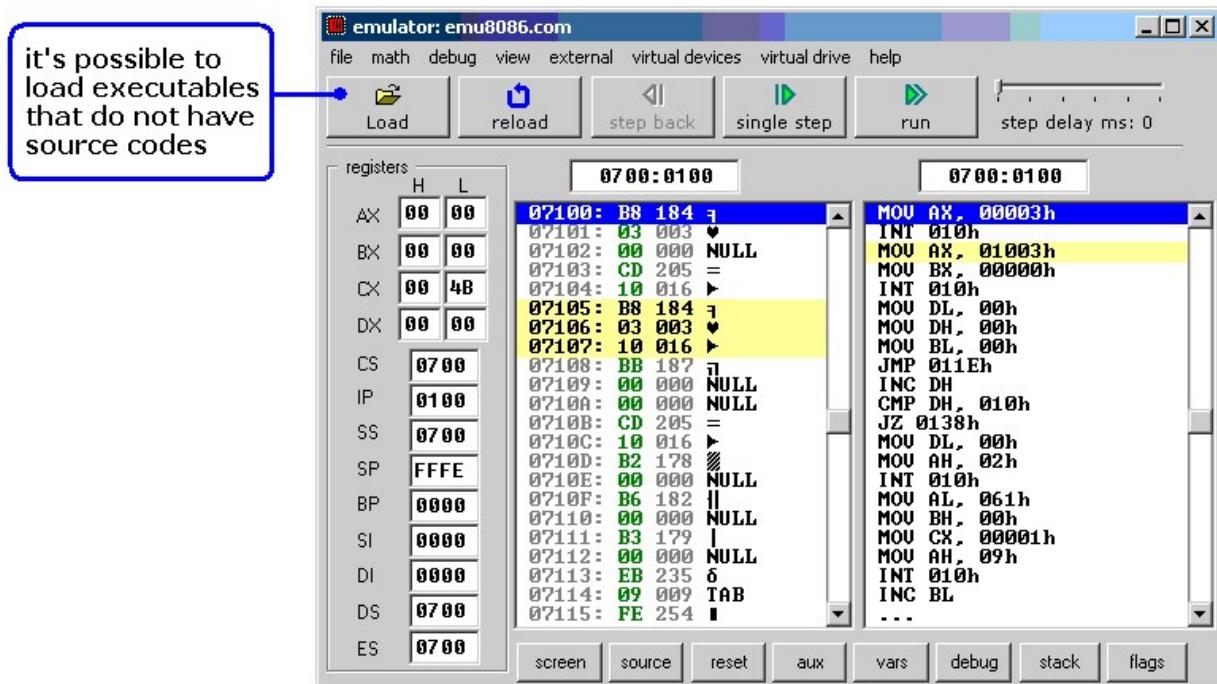
MOV DS, AX

MOV AL, 300 - is illegal instruction because **AL** register has only 8 bits, and thus maximum value for it is 255 (or 11111111b), and the minimum is -128.

## A few notes about the emulator

If you want to load your code into the emulator, just click "**Emulate**" button

But you can also use emulator to load executable even if you don't have the original source code. Select **Show emulator** from the **Emulator** menu.



[**Single Step**] button executes instructions one by one stopping after each instruction.

[**Run**] button executes instructions one by one with delay set by **step delay** between instructions.

Double click on register text-boxes opens "**Extended Viewer**" window with value of that register converted to all possible forms. You can modify the value of the register directly in this window.

Double click on memory list item opens "**Extended Viewer**" with WORD value loaded from memory list at selected location. Less significant byte is at lower address: LOW BYTE is loaded from selected position and HIGH BYTE from next memory address. You can modify the value of the memory word directly in the "**Extended Viewer**" window; you can modify the values of registers on runtime by typing over the existing values.

## Lab1 Tasks

1. Compile the code example1.asm which is provided to you.
2. Print out your Name.
3. Print out your Registration Number.
4. Come up with an extension to the program.

