Lab 1

Brief Introduction of COAL and Software Installation

Computer Organization refers to the way in which the hardware components of a computer system are arranged and interconnected. Computer Organization and Assembly Language (COAL) Programming deals with lower-level computer programming—machine or assembly language, and how these are used in the typical computer system. Assembly language programming develops a very basic and low-level understanding of the computer. In higher level languages there is a distance between the computer and the programmer. This is because higher level languages are designed to be closer and friendlier to the programmer, thereby creating distance with the machine. This distance is covered by translators which are called compilers and interpreters. The aim of programming in assembly language is to bypass these intermediates and talk directly with the computer.

Different architectures of the computer system are adopted to understand the rules, methods, and procedures that describe the execution and functionality of the entire computer system. John von Neumann coined and developed an architecture named **von Neumann architecture** . The computer we are using nowadays is based on the von Neumann architecture. It has some concepts. It is also known as Princeton architecture. It renders a unique design for the electronic digital systems having the following components:

- A Central Processing Unit (CPU) with arithmetic and logic unit (ALU) and processors with attached registers.
- A memory that can store data and instructions.
- External mass storage or secondary storage.
- A Control Unit (CU) with the ability to hold instructions in the program counter (PC) or instruction register (IR).
- Input and output mechanisms and peripherals.

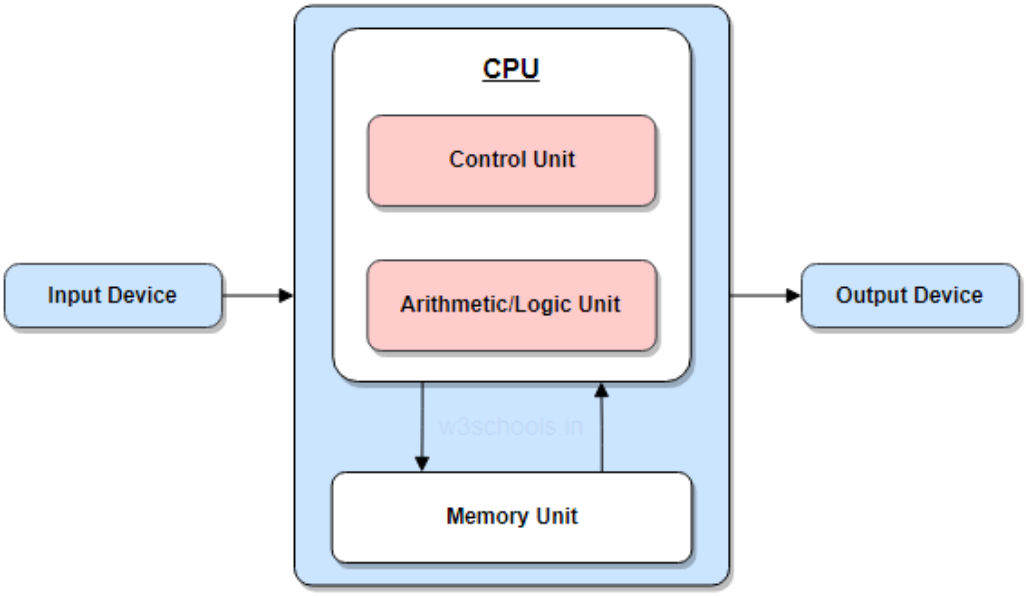


Figure 1: Von Neumann Architecture

The von Neumann design thus constitutes the foundation of modern computing. The Harvard architecture, a similar model, had committed data addresses and buses for reading and writing to memory. It won because von Neumann's architecture was easier to execute in real hardware.

Software Required

- DOSBox (An Emulator)
- Notepad ++/Visual Studio Code for Assembly Coding

Objective of Using DOSBox

DOSBox is an emulator and the purpose of using it here is to study a specific 16-bit architecture of Intel IBM PC using assembly language on a 32-bit or 64-bit machine.

Figure 2: DOSBox Icon

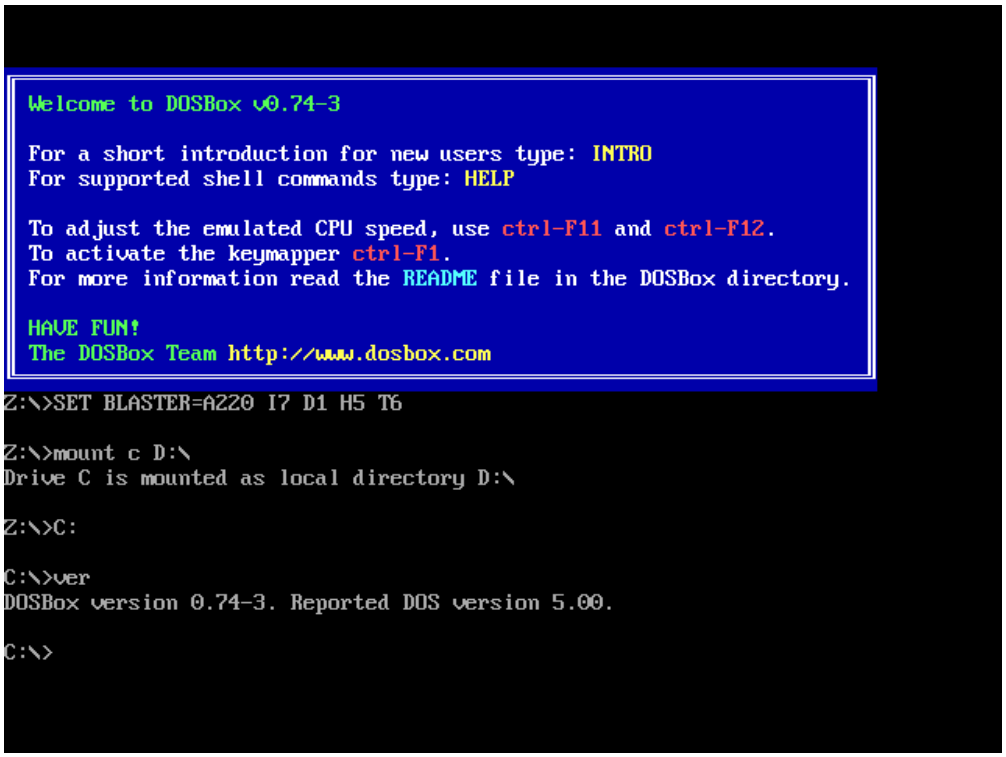


Figure 3: Interface of the DOSBox

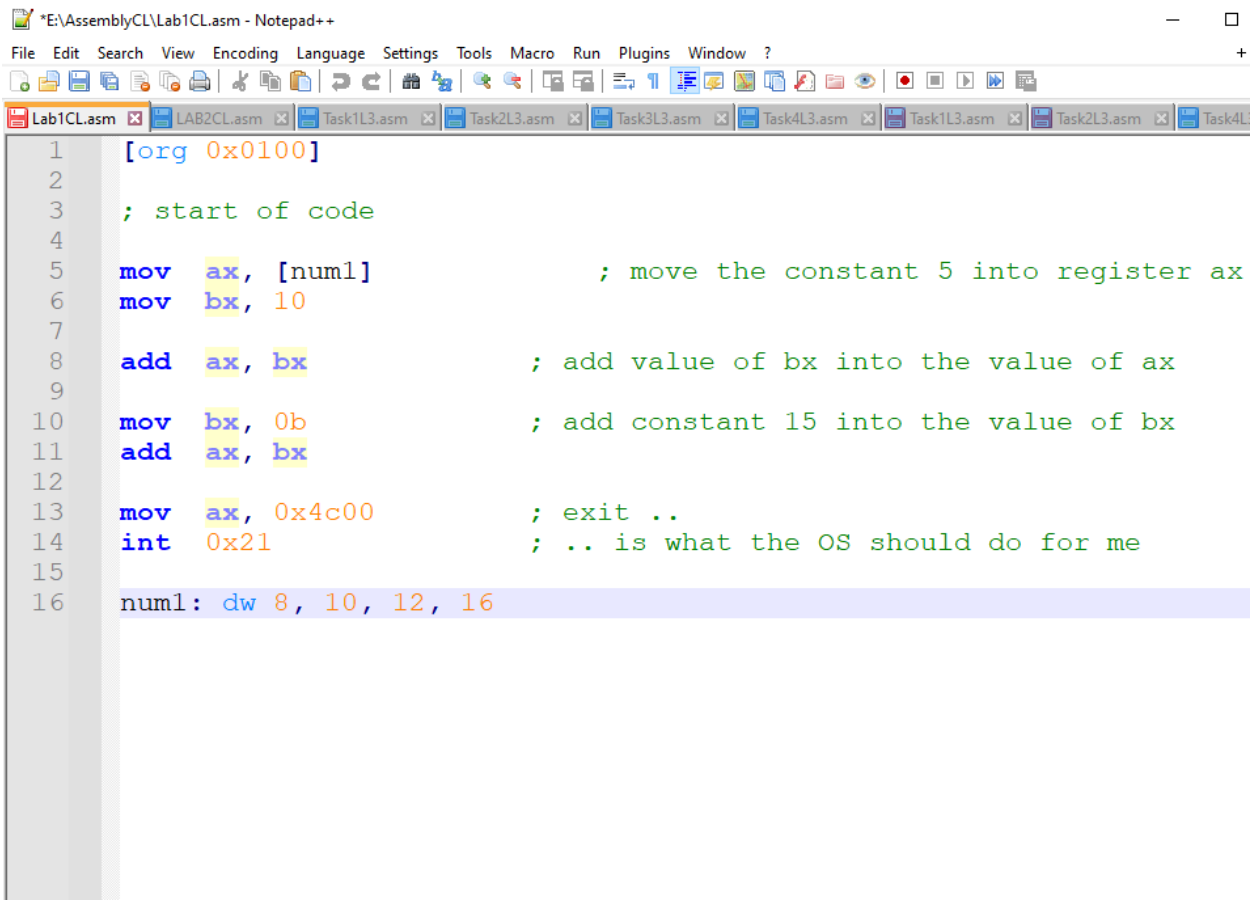
Steps of Installation for Windows

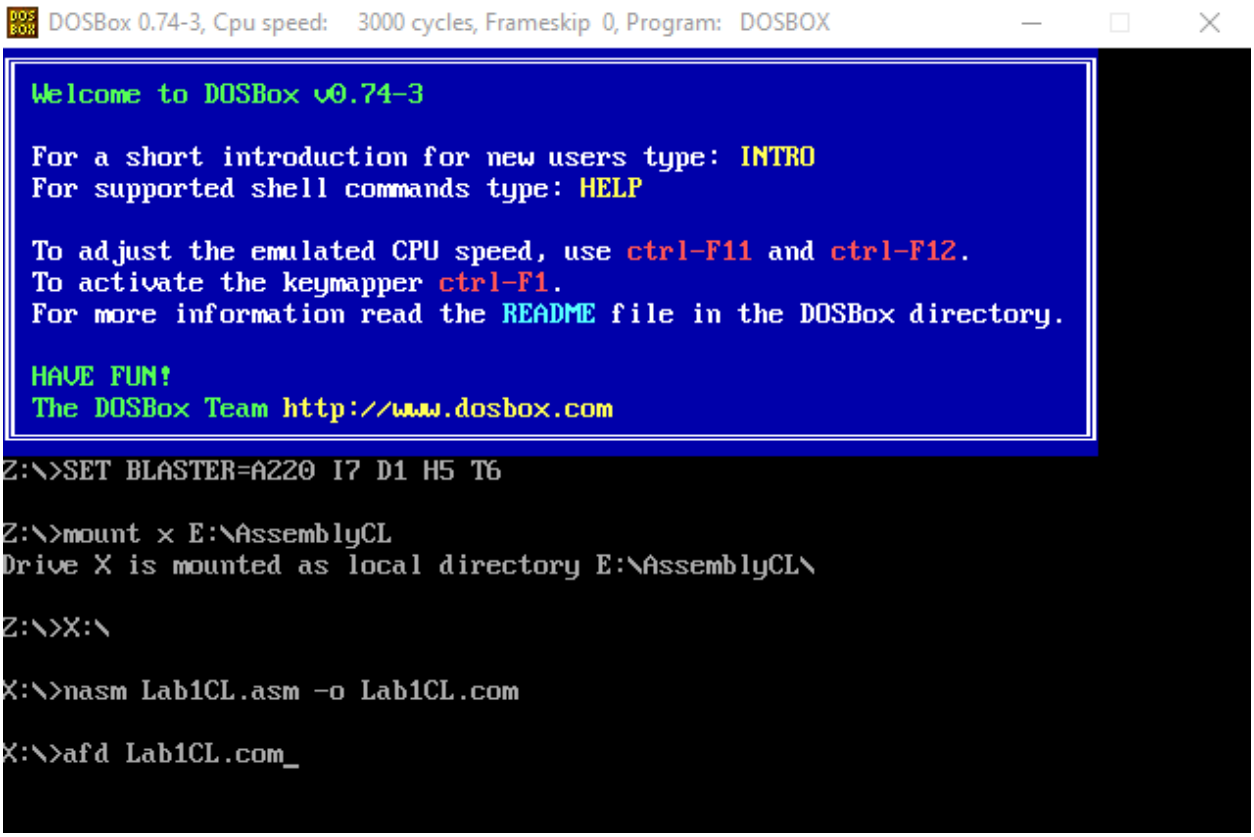
Follow these steps if you have a Windows PC to download and install DOSBox:

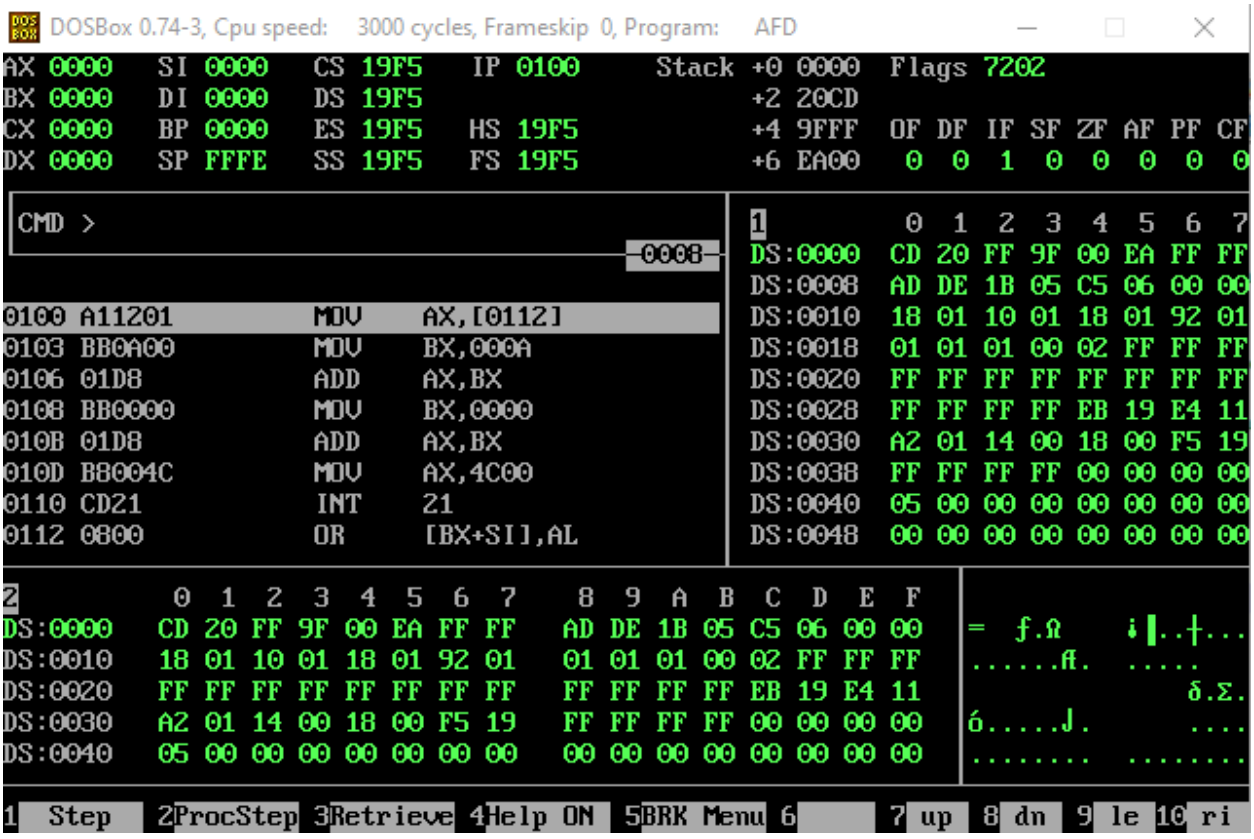
- Open a browser and go to the DOSBox download page.
- Locate the latest Windows installer download and select it.
- After the download is completed, run the installer.
- Follow the on-screen directions. It's fairly simple, and the default options should suffice in most situations.

Lab Work

- Installation of the Softwares
- Getting familiar with the interface of DOSBox and making **.asm** files in NotePad++




DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Welcome to DOSBox v0.74-3
For a short introduction for new users type: INTRO
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount x E:\AssemblyCL
Drive X is mounted as local directory E:\AssemblyCL\
Z:\>X:\
X:\>nasm Lab1CL.asm -o Lab1CL.com
X:\>afd Lab1CL.com


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0008
0100 A11201 MOV AX,[0112]
0103 BB0A00 MOV BX,000A
0106 01D8 ADD AX,BX
0108 BB0000 MOV BX,0000
010B 01D8 ADD AX,BX
010D B8004C MOV AX,4C00
0110 CD21 INT 21
0112 0800 OR [BX+SI],AL
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E4 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri