Group name: The-Boss

Topic: Introduction to compiler (assembler language).

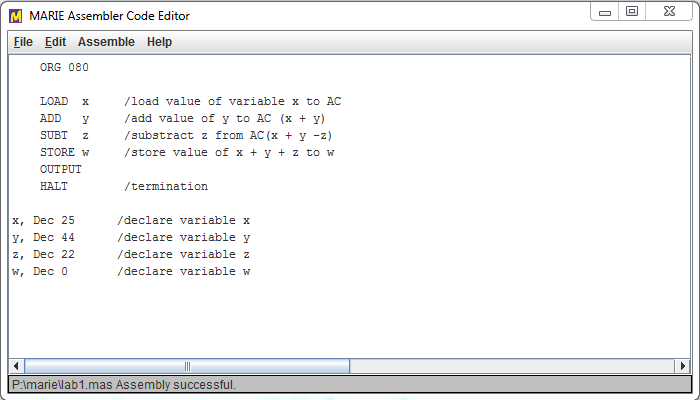
Aim: To convert assembly language (using mnemonics) into machine language (which consists entirely of binary values, or strings of 0s and 1s). Assemblers take a programmer’s assembly language program, which is really a symbolic representation of the binary numbers, and convert it into binary instructions, or the machine code equivalent.

Introduction: A Complier: A compiler translates something from one language into another, for example, a Java program into machine code.

The MARIE is a very simple assembly language that is used in class rooms. It comes with a graphical simulator/debugger and a graphical Data Path simulator.

MARIE’s Architecture: The machine running MARIE has AC: The accumulator (AC) is special memory cell that numbers are loaded into and out of and adding takes place. PC: The program counter (PC) is a special memory cell that contains an address/index to the memory cell containing the current instruction being executed. After an instruction is executed, this index is incremented so that it indexes the next instruction. A jump involves changing the PC to the index of a different instruction.

Results:

Figure 1: The Assembler editor containing the commands showing step by step how the program will run.

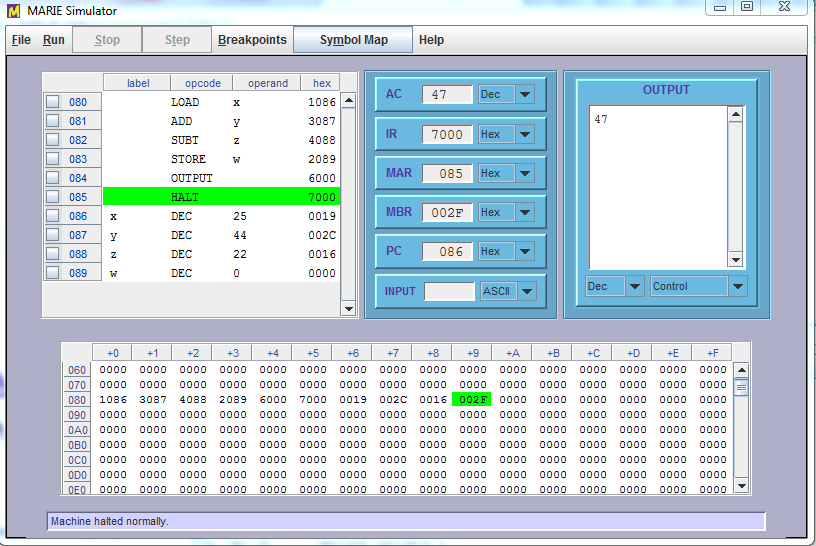


Figure 2: The program : w = x + y – z is loaded on the simulator from Marie assembler code editor and it is suppose to show the output of the declare decimal variables of the code written on the editor.

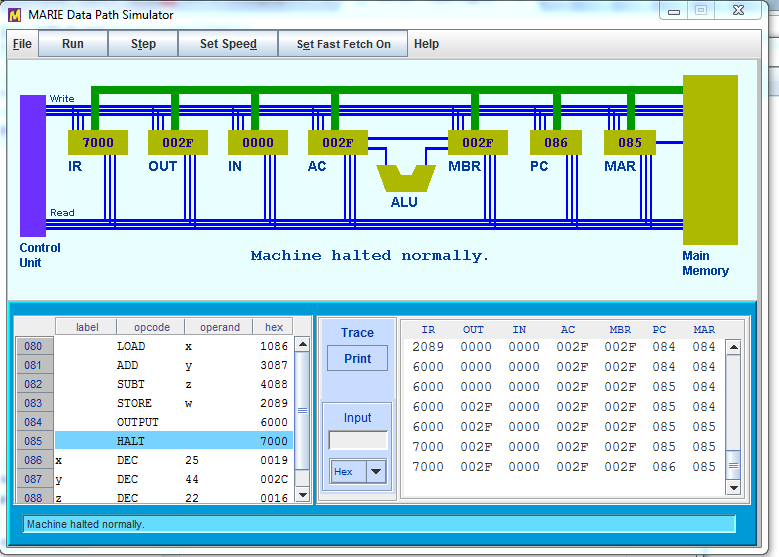


Figure 3: A data path simulator showing visuals of how instructions relate to sequence of signals.