Simple 8-Bit Assembler Simulator

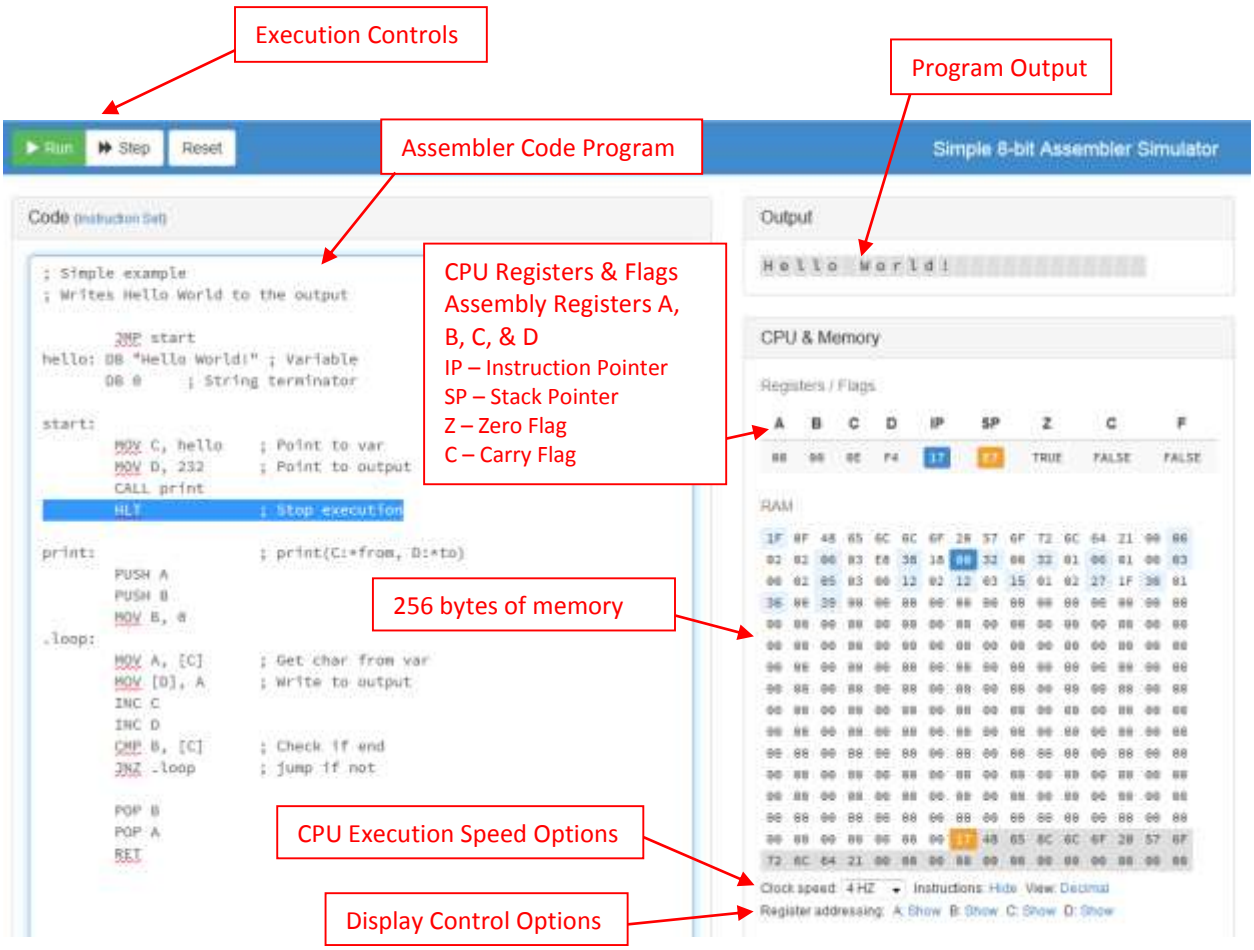
Instructions

Open a web browser and navigate to https://schweigi.github.io/assembler-simulator/index.html

This simulator provides a simplified assembler syntax (based on Netwide Assembler NASM) and is simulating a x86 like cpu. In depth documentation and introduction to assembler can be found on the following websites:

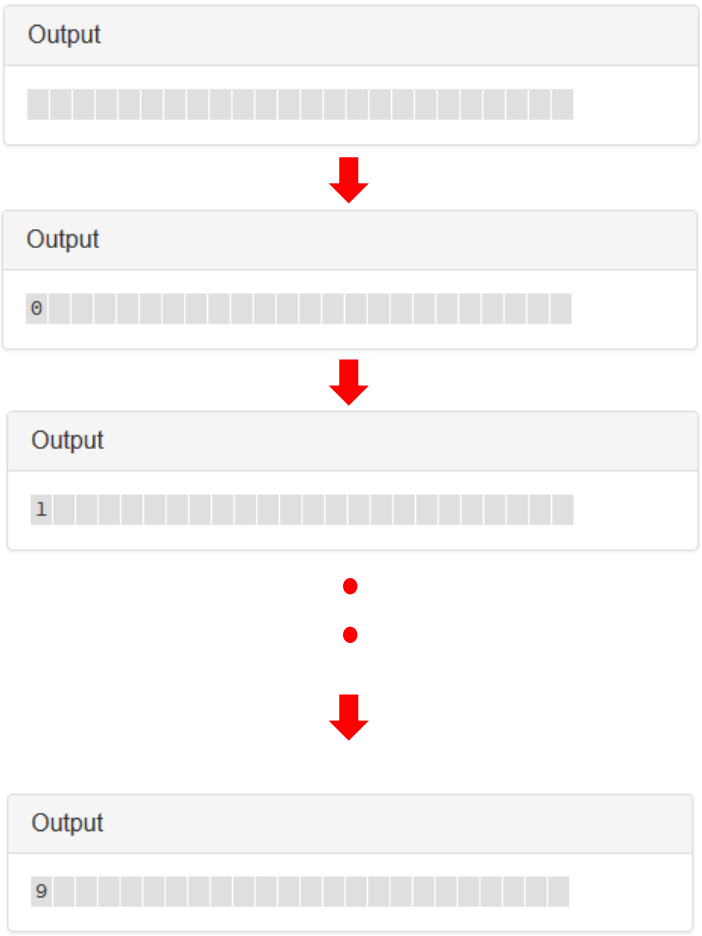
- Assembly - Wikipedia
- The Art of Assembly Language Programming
- NASM Language Documentation
- The simulator consists of a 8-bit cpu and 256 bytes of memory. All instructions (code) and variables (data) needs to fit inside the memory. For simplicity every instruction (and operand) is 1 byte. Therefore a MOV instruction will use 3 bytes of memory. The simulator provides a console output which is memory mapped from 0xE8 to 0xFF. Memory mapped means that every value written to this memory block is visible on the console.

User Interface



Part A

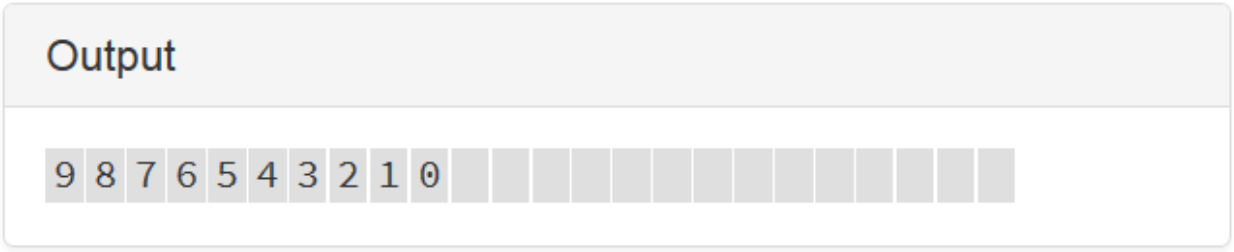
Write an assembler program that, when executed, will output the accumulated value of a counter starting at 0 and ending at 9. Write the counter value to byte **0xE8** so that it will appear in the simulator output window as shown below.



Thoroughly test your solution to see that it works as required. When you have finished the testing, copy the assembler code, paste it into a text file named HW8a.txt, save it, and attach it to the assignment.

Part B

Modify your assembler program from Part A so that it will output the accumulated value of a counter starting at 9 and ending at 0. This time have the output write to consecutive output bytes starting at **0xE8**. The resulting output will look like the figure below which shows the simulator output window.



Thoroughly test your solution to see that it works as required. When you have finished the testing, copy the assembler code, paste it into a text file named HW8b.txt, save it, and attach it to the assignment.