Report For Lab 1

Connor McCullough

EEN 312

6 February 2013

**Objective:**

The objective of this lab was to learn how to create a virtual program in edsim51.jar by inputting source code and assembling the program into machine code. The program created in this lab moved hex and decimal values into data memory locations, then reading the info from these memory locations into the accumulator.

**Equipment Used:**

Software:

* Text editor and 8051 ASM assembler
* Step debugger to execute program one instruction at a time
* Data memory, accumulator, code memory

**Flow Chart:**

Read Data From Memory Location 1 to Accumulator

Store EFH in Data Memory Address 6D

Store 38 in Data Memory Address 8

Store 20 in Data Memory Address 1

Store 20H in Data Memory Address 0

**START**

Read Data From Memory Location 8 to Accumulator

Read Data From Memory Location 6DH to Accumulator

Read Data From Memory Location 24 to Accumulator

**END**

Read Data From Memory Location 0 to Accumulator

**Test Results:**

Memory After First Five Steps:



Memory After Step 6:



Memory After Step 7:



Memory After Step 8:



Memory After Step 9:



Memory After Step 10:



**Conclusion:**

This lab showed how to store values in the data memory using the MOV function, as well as showing how moving values to the accumulator will cause the value in that register to continuously get larger. It also showed how debugging can be done using the Step function to run one line of code at a time and check the results in the memory. Finally, the lab taught how to reset the value in the data memory at the conclusion of the program.

**Code:**

org 0h ; Starts code at code memory location 0

mov 0, #20H ; Store hex value of 20 in data memory address 0

mov 1, #20 ; Store decimal value of 20 at data memory address 1

mov 8. #38 ; Store decimal value of 28 at data memory address 8

mov 24, #0AH ; Store hex value of AH at data memory address 24

mov 6DH, #0EFh ; Store hex value of EFH at data memory address 109

mov A, 1 ; Read data from memory address 1 to Accumulator

mov A, 24 ; Read data from memory address 24 to Accumulator

mov A, 6DH ; Read data from memory address 109 to Accumulator

mov A, 8 ; Read data from memory address 8 to Accumulator

mov A, 0 ; Read data from memory address 0 to Accumulator

end