ECE 242 – Programming Assignment #2 **Spring 2021**

Due Date: Tuesday, March 2nd 2021, 11:59PM

PART I:

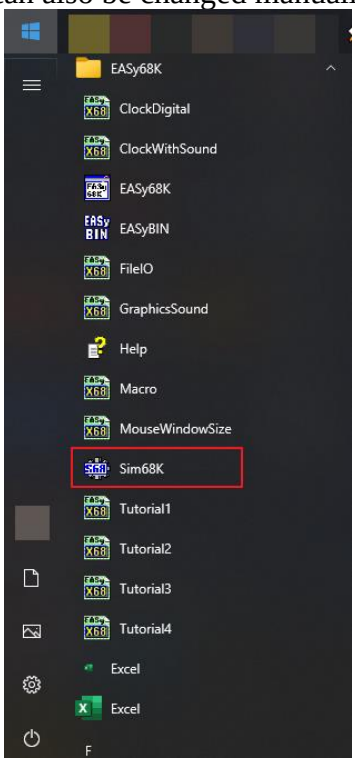
1. Write the following values in the memory locations below as the microprocessor would store them starting at address $1000.

$C532_{16}$
$A6B0_{16}$
89_{16}
F_{16}
$B175E3D7_{16}$
$5B03548A_{16}$
$005C_{16}$

	Memory Contents	
Address	Bits 15-8	Bits 7-0
$1000		
$1002		
$1004		
$1006		
$1008		
$100A		
$100C		
$100E		
$1010		

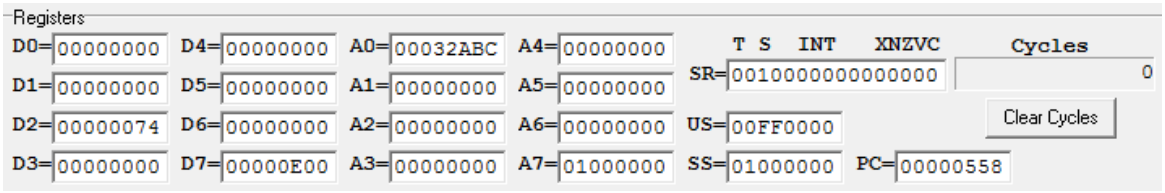
Simulation is having one computer act like (simulate) another. In this way, the need for the actual computer is avoided. This is beneficial in development of a new system since hardware and software can be developed together based on the simulation of the computer. The MC68000 simulator program will allow us to execute MC68000 instructions without actually needing the MC68000 microprocessor.

When the simulator is started, the contents of all registers and memory locations (except A7 and SR) are cleared to zeroes. The contents of these registers will change after execution of each instruction to reflect the updated status of the microprocessor. The contents of these registers can also be changed manually.



2. Start the simulator from Windows Explorer by navigating to Start > EASy68K > Sim68K.

3. Change the contents of the following registers:
    a. In register D2, put 74_{16}
    b. In register A0, put $32ABC_{16}$
    c. In register D7, put $E00_{16}$
    d. In the PC register, put 558_{16}



4. Verify that your data registers match those above.
5. Go to View > Memory and enter the following memory values at the specified addresses (starting memory locations are enclosed in <> brackets)
    a. <1000>=C532
    b. <1002>=A6B0
    c. <1004>=89
    d. <1005>=F
    e. <1006>=B175E3D7
    f. <100A>=5B03548A
    g. <100E>=005C
6. Print out what is now on the screen using the key combination Alt + Print Screen.
7. Using the table below, copy the memory contents into the corresponding memory locations in the chart below. Note that the simulator displays bytes of memory, one after the other, in rows.

Memory Contents

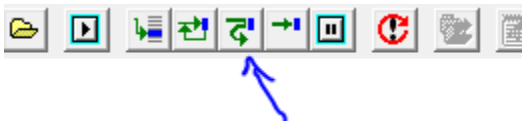
Address	Bits 15-8	Bits 7-0
$1000		
$1002		
$1004		
$1006		
$1008		
$100A		
$100C		
$100E		
$1010		

8. Compare the contents of the simulator memory (in Step 7) with the predicted values from Step 1. They should be the same. If not, you may have made error in Step 5. Find your errors, if any, and be sure the values agree.

PART II:

We have already practiced entering data into the microprocessor registers and memory locations via the simulator. Now we will enter a program, containing seven instructions, into the memory. Then we will run the program on the simulator, stopping after each instruction.

1. Start the simulator as you did for the preceding exercise and open the Memory view.
2. Store the following instructions in memory, starting at address $3100.
    a. <3100>=3A3C0039
    b. <3104>=323C07CE
    c. <3108>=3C3C00F4
    d. <310C>=363C0113
    e. <3110>=DA41
    f. <3112>=D606
    g. <3114>=4E4E
3. Set the program counter to the starting address $3100.
4. Press the Step-Over button or press F9 to execute the first instruction in the program.



5. Examine the top part of your screen. Record in the box below, the names and contents of any registers that contain non-zero values. (Most registers will contain zeroes).

6. Examine the bottom part of your screen. Record the output by filling in the table below.

PC	Op-code	Instruction

7. To execute each instruction, press the Step-Over button or press F9. After each instruction is executed, record the updated values of only the data registers referenced above and the number of cycles. Repeat until all seven instructions are executed.

Instruction	Data Registers	Number of Cycles

8. Verify that the program instructions are still in memory and could be executed again.

9. For each machine instruction above (in Step 2), convert the hexadecimal code to binary, partition the bits into appropriate fields. (reference MC68K Programmer's Manual or Appendix IV of the Textbook)

10. Decode each instruction into its assembler-language mnemonic from Step 9. (reference MC68K Programmer's Manual or Appendix IV of the Textbook)

11. Submit all of the above results on the due date to the Blackboard