**Introduction To The EdSim51 Simulation Programming**

Objectives:

This experiment will familiarize the student with the EdSim51 simulator program and running programs in EdSim51.

Procedure:

1) Locate your files previously saved (simple\_add.ASM & simple\_add.HEX)

a) Using the 8051IDE program, modify the main program of simple\_add.ASM as shown below:

ORG 00000H

;PROGRAM

start:

mov 020h,#00h

mov 021h,#00h

mov 022h,#00h

mov A,#00h

mov A,#27h

mov 020h, A

mov 021h,#11h

mov A,020h

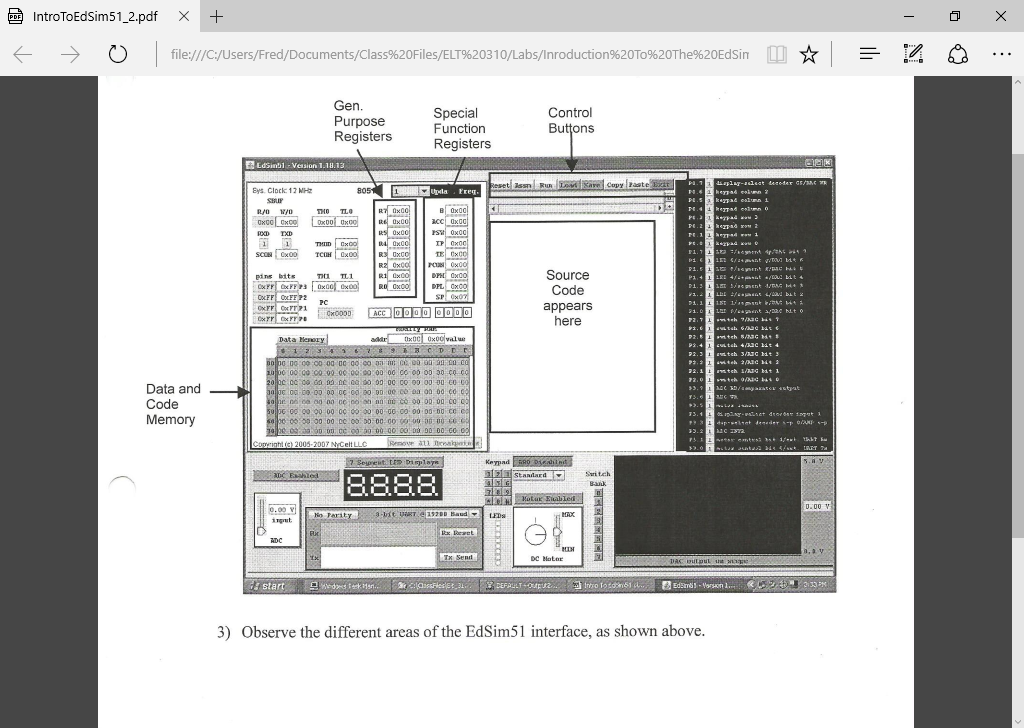
add A,021h

mov 022h,A

ret

b) Save the file as "simple\_add". Be sure to add the .ASM to your program name and change the "Save as type:" to All Files.

c) Assemble the program

2) Open EdSim51

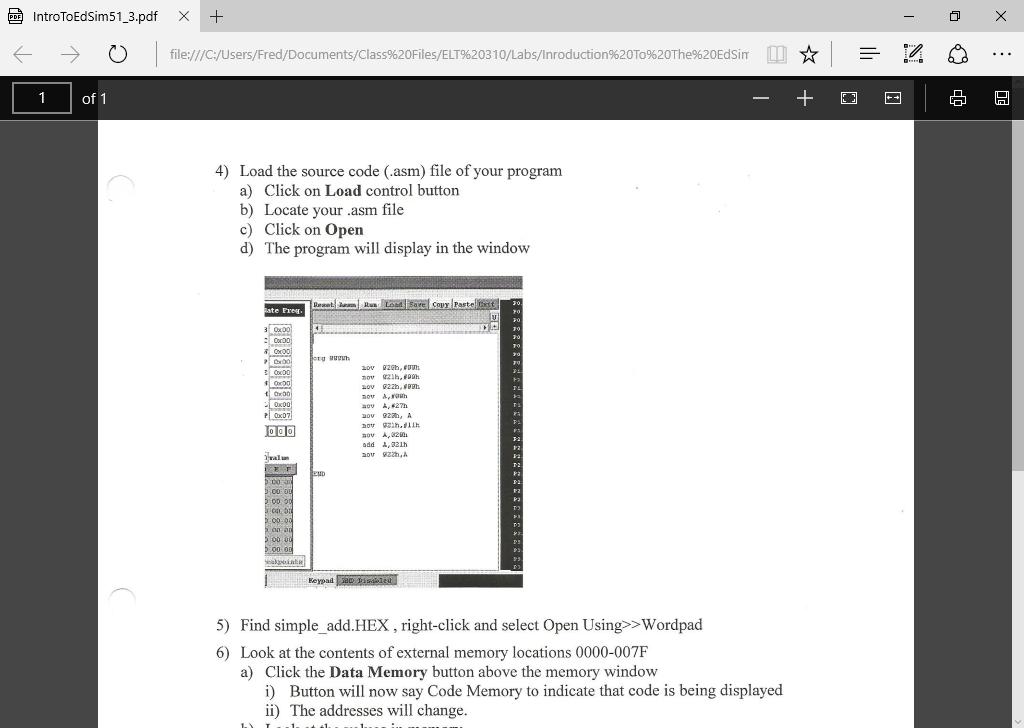
3) Observe the different areas of the EdSim51 interface, as shown above.

4) Load the source code (.ASM) file of your program

a) Click on **Load** control button

b) Locate your .ASM file

c) Click on **Open**

d) The program will display in the window

5) Find simple\_add.HEX , right-click and select Open Using>>Wordpad

6) Look at the contents of external memory locations 0000-007F

a) Click the **Data Memory** button above the memory window

i) Button will now say Code Memory to indicate that code is being displayed

ii) The addresses will change.

b) Look at the values in memory

c) Assemble the program in EdSim51 by pressing the Assm control button above the source code window

d) Reexamine the code memory. Note that in the simulator once the program is assembled it is automatically loaded into memory. This will not happen on the 8051 Development Board! The program must be loaded onto the development board in a separate step.

e) Screen capture the EdSim51 screen.

f) Compare the contents of memory with the hex values for the program in the .HEX file.

7) Examine the contents of Internal Memory

a) Click the **Code Memory** button above the memory window

i) Button will now say Data Memory to indicate that data is being displayed

ii) The addresses will change.

b) Look at the values in memory

c) Look at the first line in the program and determine what it should do.

d) Click on the Step button above the source code window.

e) Look carefully the data values in locations 20-23

f) Could you determine whether or not the instruction was successfully executed?

g) Repeat steps d-f for the remaining lines of code in the program. Be sure to look at the accumulator (ACC) window if the instruction involves register A.

i) The accumulator is shown in two places. Once in binary and once in hex.

h) Look carefully (or copy and past into Word) the data values in locations 20-23

i) Screen capture the EdSim51 to show memory after all steps of the program have executed.

8) Click the **Reset** button above the source code window.

9) Assemble the program again.

10) Observe memory locations 20-23. What, if any changes occurred when the program was reset?

11) Step through the program again.

12) Were the effects of the first three steps seen when the program executed a second time?

13) Using the 8051IDE program modify the program to move the data value 05h into internal memory location 2Ah, data value 50h into internal memory location 2Bh, add the contents of 2Ah to the contents of 2Bh and store in 2Ch. Save as **simple\_add\_ revl.ASM** Assemble the program. Modify the original program. DO NOT add lines to the original program. Your modifications should be saved as a new program,simple\_add\_ revl.ASM

14) Test your modification using EdSim51

15) Close EdSim51 by clicking on the Exit button above the source code window.

Analysis:

1) Describe the process used to enter, assemble and test a program. Be sure to mention the programs used for each step and the instructions used within each program.

2) For the program run in step 11:

a) What was the specific function of the program? What is the program to do, not why are you doing the experiment.

b) Using your carefully annotated memory printouts, explain why the program run in step 9 was successful.

3) Using your carefully annotated memory printouts, explain why the modifications to the program run in step 13 were successful.

3/18/20