

Due June 3 (Friday)!

Purpose

In this project you will integrate everything that you have been learning and building in the problems. So far you have learned how to simulate parts of a computer, the ALU and the CPU, and the Controller (FSM). You have also learned how to produce a debug monitor software that allows you to 'see' inside the computer as it operates. This has all been done with respect to a tiny computer specification, but the architecture of the simulation software is the same for any CPU you would like to simulate. Now it is time to apply this architecture to simulate a portion of the LC-3 to see how possible it is to do this.

Requirements

Use the textbook for your guidance to instructions, their formats, addressing modes, and CPU architecture. Applying the same basic approach that you used in the problems, implement the following LC-3 instructions such that your simulator can run a program assembled by the LC-3 editor. You will be able to run the program on the LC-3 book simulator to check the correctness of your results.

Teamwork

You are working in pairs but may divide the work as you think necessary. Make sure, however, that you are in constant communication, especially as regards the interfaces between partners' work when it comes time to integrate. You don't want any last minute surprises!

Instructions

ADD, ADD(i), AND, AND(i), BR, JMP, JSR, LD, LEA, NOT, RET, ST, TRAP

Extra credit:

LDR, STR, LDI, STI

The first set of instructions should allow your simulator to run interesting programs.

In addition to the instructions, you will need to implement simulations of several of the TRAP routines as described in Appendix A.

GETC, OUT, PUTS, HALT

Additional Implementations

Input/Output

The monitor screen should be modified to allow a few lines at the bottom of the screen where the simulator can get characters and print to the screen so that the trap functions can perform input and output.

Loading an Object File

You will need to write a routine that loads an object program from disk to your main memory. We will cover the file format and how to go about this in class.

Assembly Program

In class we will be developing a test program in assembly language that should be capable of running on both the book simulator and your simulator. You will also develop your own program (I will provide a list of program requirements from which you can choose which one to implement).

Turn In

1. Source code files
2. Screen shots of running program - 3 shots at different stages of the run of the two machine program