Lab 9 – LC-3 FSM Implementation

1 Objective

To understand and to exercise the control logic of the LC-3.

2 Introduction

In this lab you will design the finite state machine controller for a subset of the LC3 instructions. You will then link the controller to the Datapath from the last lab to form a complete CPU. You will then simulate the entire structure.

The LC3 has 16 or so instructions. You will implement at least the AND, ADD, NOT, JMP, JSR, BR, LD, ST and RET instructions.

3 Preparation

3.1 Design FSM Controller

The Datapath you designed in the last lab has many control signals as inputs. These will come from the Controller block that you will design in this lab. The input to the Controller block will come from the LC-3 Datapath. Specifically, these inputs will be the N, Z, P flags and the 16-bit IR.

The FSM in the Controller block needs to be able to do the following:

1. Reset itself in response to the the reset signal.
2. Fetch the next instruction into the IR.
3. Decode the instruction in the IR.
4. Execute the instruction in the IR if it is any of AND, ADD, NOT, JSR, JMP, BR, LD, ST or RET. If it is not one of those instructions, ignore it and go fetch the next instruction

You may use any state machine design technique desired. However, it is STRONGLY suggested you use one-hot encoding. It is the easiest to implement and the next state equations can be derived by inspection. It is also STRONGLY suggested that you use VHDL data-flow coding whenever possible.

Include a copy of your state graph in your lab write-up.

Some other suggestions:

1. Figure out how to execute AND, ADD, and NOT in the same execute state in the machine. You don't need to use 3 different states and will be docked points if you do so.

2. Don't forget to include a DECODE state in your machine.

3. Carefully review the lecture notes on Designing the LC3 Control.

3.2 Modify the Memory module

Load a new copy of the Memory Contents module into your project. This new memory block contains the program for this lab which your LC3 will execute.

NOTE: This is not the same memory file as the one used in the previous lab. You will need to download this new file. I will let you know when the new file is available. You can use the previous lab's file to test and ensure proper function of your FSM.

4 Procedure and Functionality

Now, create a testbench file that watches all the important registers inside the LC3 (PC, NZP, IR, R0-R7, MAR, MDR, Buss, etc.) and then simulate for long enough to execute the entire program contained in the Memory module. You will know when the program is done when it just loops and keeps executing the same single instruction over and over. Verify that the program does what you expected.

As before, print out your simulation results and write all over them to explain to the TA's what you did. In your write-up, be sure to describe, in your own words, just what this program is doing (is it calculating artillery tables, is it sorting numbers, ???). Then, mark up your simulation output to explain that it correctly does that thing.

There is a functional pass-off for this lab. Have a TA go over your simulation waveforms to verify functionality. Export the waveforms and include copies of the VHDL files, testbench files, simulation files and state graph. You must turn them in box by Wednesday, December 6th. You have until then to get the work done. No late labs will be accepted.