E85: Digital Design and Computer Architecture

Lab 10: Multicycle Controller

Objective

In Labs 10 and 11, you will design a multicycle ARM processor in SystemVerilog and test it on a simple machine language program. This will tie together everything that you have learned in E85 about digital design, hardware description languages, assembly language, and microarchitecture. In Lab 10, you will build and test the controller. In Lab 11, you will build the datapath and test the whole system. You will need a working multicycle processor for your final exam.

1. Multicycle ARM Controller

Figure 7.31 of the textbook shows the controller for the multicycle processor implementing a subset of ARMv4. Figure 7.41 shows the Main FSM. Table 7.6 defines the Instruction Decoder. Table 7.3 and HDL Example 7.3 define the ALU Decoder. Page 400 and HDL Example 7.3 show the PC Logic. Table 6.3 and HDL Example 7.4 have the Condition Check logic.

Write a hierarchical Verilog description of the multicycle controller. The controller should have the following module declaration and should follow the hierarchy of Figure 7.31. Remember that Op, Funct, and Rd are bitfields of Instr.

```
module controller(input  logic         clk,
                  input  logic         reset,
                  input  logic [31:12] Instr,
                  input  logic [3:0]   ALUFlags,
                  output logic         PCWrite,
                  output logic         MemWrite,
                  output logic         RegWrite,
                  output logic         IRWrite,
                  output logic         AdrSrc,
                  output logic [1:0]   RegSrc,
                  output logic [1:0]   ALUSrcA,
                  output logic [1:0]   ALUSrcB,
                  output logic [1:0]   ResultSrc,
                  output logic [1:0]   ImmSrc,
                  output logic [1:0]   ALUControl);
```

2. Test Bench

Develop a self-checking testbench for the multicycle controller. It should exercise all of the logic in the controller. Run your testbench and debug any errors you find.

For greater confidence and to reduce the risk of discovering new bugs during Lab 11, optionally trade testbenches with a classmate. Test your controller with your colleague's testbench and debug any errors you find.

What to Turn In

1. Please indicate how many hours you spent on this lab. This will be helpful for calibrating the workload for next time the course is taught.
2. Hierarchical SystemVerilog for your controller module matching the declaration given above.
3. Self-checking testbench. Explain how you chose your test cases to test all of the logic in the controller. Does your controller pass your testbench?

If you have suggestions for further improvements of this lab, you're welcome to include them at the end of your lab.