Digital Electronics & Computer Engineering (E85)

Lab 11: Multicycle Processor (Part 2)

Multicycle Processor Completion

In this lab, you will complete your own multicycle MIPS processor! Please refer to your Lab 10 handout for an overview of the processor. You may also refer to Lab 10 solutions. By this point, you should be a competent digital designer so few directions will be provided.

Design the eunit, iunit, and munit taking the inputs and outputs specified in Lab 10. Reuse hardware from earlier labs (such as the ALU, sgnext, and register file) wherever possible. Use a 16-word LogiBLOX RAM in the munit. Connect all four units together into a top-level design. The design should take Clk and Reset as inputs. All of your flip-flops should take a reset input into the Async Control terminal to reset the initial value to a known state. The Instruction Register and PC also require CLK_EN enable inputs. You should label the internal signals and busses so you can view them during simulation. Pay careful attention to bus connections; they are an easy place to make mistakes. For example, remember that the IorD mux needs bits [5:2] of the PC or ALUOut values as the address so don't try to connect the 32-bit bus directly to the 4-bit mux inputs. Be sure to enter your test program from Lab 10 into the munit RAM .mem file.

Simulate your processor for 370 ns. Use a 10 ns clock and a Reset signal that is high for the first 10 ns. Display, at a minimum, the PC, Instruction, FSM state (from within your cunit), SrcA and SrcB (from within your eunit), ALUResult, and Zero. You will likely want to add other signals to help debug. Check that your results match the expectations from Lab 10. If there are any mismatches, debug your design.

What to Turn In

Please turn in each of the following items:

1. Please indicate how many hours you spent on this lab. This will not affect your grade, but will be helpful for calibrating the workload for next semester's labs.
2. Schematics of:
   - eunit
   - cunit
   - munit
   - your top level design
3. Simulation waveforms of the processor showing PC, Instruction, FSM State, SrcA, SrcB, ALUResult, and Zero while running the test program. Be sure the results match your expectations.