ECPE 174 – Lab 7
ALU

1. **Objectives**
   Finalize our ALU to include a logic unit, a FIFO and a reset. Verify using a randomized test suite with complete code coverage.

2. **Problem Statement**
   We want to finalize our ALU. Over two weeks, we will finish the datapath and control, add memory, include a reset, and complete our verification procedure.

   The complete ALU will operate on two 6-bit twos complement numbers (A and B). It will:
   - Add
   - Subtract
   - Compare the 2 values to determine if A equals B, is strictly greater than B, or is strictly less than B
   - Check if A==0

   The ALU must be a hierarchical design, utilizing correct digital design principles. Each component needs to be individually tested and verified. At a minimum, you should have a 6-bit adder, a 6-bit logic unit, and a FSM-based control unit. In implementing your system, you can use any technique you want.

   The ALU will use the following Instruction Set:

```
Add = 000
Subtract = 001
Equal = 100
Greater than = 101
Less than = 110
A equal 0 = 111
```

   Our FIFO system from Lab 6 will allow us to store a set of operations and then run these operations. You should store the Instruction, A, and B together into a single memory entry.

   The complete system should have a reset. For the FFs, ensure this reset is asynchronous assert, synchronous deassert reset.

   We will enter Instructions, 'A' values, 'B' values, FIFO control, and reset on toggle switches (your choice of switches). KEY3 will indicate that the Instruction and Data are ready. Output the Instruction on the red LEDs. Output 'A' values, 'B' values, and the result on the 7-segment displays. Display 'A' and 'B' magnitudes on the 7-segment and use red LEDs below the 7-segment to indicate the sign of the value (light up when the result is negative). For arithmetic operations, display the correct magnitude result on the 7-segment and use a red LED below the 7-segment to indicate the sign of the result. For the logic operations, you should display '1' if the case is true and '0' if false on the 7-segment display.

3. Pre and In Lab

For the first week, you need to turn in a pre-lab but no in-lab. For the second week, you need to turn in an in-lab, but no pre-lab.

For the first week, complete the usual prelab section as well as the following steps before lab and turn in by 1:00pm on Canvas:

1. Required: Design a block diagram of your full ALU system. Label all signals.
2. Required: Describe the current state of your system (based on Lab 4 and Lab 6). This should cover both design and testbenches.
3. Required: Outline your plan for completing the design and verification of the ALU. Explicitly detail what you will do in the week leading up to the first lab, in the first lab, in the week leading up to the second lab, and in the second lab. This should be several paragraphs of text along with bulleted lists of procedures.
4. Required: For the code you complete before the first lab, explain your design choices, provide the SystemVerilog or Verilog, and include simulation results demonstrating the working system.

Complete the design and the following steps in lab by 6:00pm of the second week:

5. Submit all code for your ALU design.
6. Submit all code and scripts for a testbench suite that completely tests your design using randomization, tasks, and self-checking. Submit screenshots of results verifying component and full system operation. Submit screenshots showing the final code coverage results.
7. Implement the complete design on the Cyclone 4. Demonstrate for check-off.
8. Draw a system diagram including the main components. Create a constraints file to set the clock to a reasonable clock speed. Using this constraints file and the Timing Analyzer, analyze the timing of your overall system. Determine the longest path, the propagation delay through the add/subtract unit, the propagation delay through the logic unit, and the maximum clock frequency. Outline the equation for the maximum clock frequency using the information from the analysis. Submit screenshots and figures.

4. Post-Lab

Discuss the following your lab report (everything should be based on the final design):

- How did your test procedure work? What if any modifications were necessary to either the design or the test procedure?
- How much space do the units require? What is the clock information for your different units? Include setup time, hold time, and maximum clock rate.
- Compare and contrast the space and time requirements between the add/subtract and logic units. Which requires the most space? Which has the longest setup and hold times? Do these answers make sense to you? Explain.
- What, if any, assumptions did you need to make in order to complete this design? What, if any, additional logic or functional extensions would improve the system?
- Outline the information determined from the timing analyzer. Describe your computation of the maximum clock frequency and longest propagation delay path. Are these reasonable values? How might you modify the design to increase the clock frequency? How might you modify it to reduce the propagation delay?