Lab Assignment 4: Unsigned and Two's Complement Comparator Design

ECE/CS 3700
Spring 2018
Assigned during the week Feb 27 - March 5. Due date: Simulation and Synthesis demo during your respective lab sessions during the week March 6 - March 12. Final report due by Monday March 12.

I. Laboratory Objectives

The purpose of this *short, 1-week lab* is to give you some practice with designing combinational circuits using **always @()** statements in Verilog. So far, you have learnt how to design combinational logic using gate-level primitives or using continuous assign statements. Now you are asked to use only always blocks to design combinational logic.

In this lab, you will design and implement a **configurable comparator** using Verilog. You will perform Verilog Design, Simulation and Synthesis. *However, there is no need to map and download the circuit onto FPGAs (we'll do that again in the next set of labs).*

II. Design Specifications

You are asked to design a circuit that takes two data inputs, a 4-bit vector $A[3:0]$ and another 4-bit vector $B[3:0]$. There is another *binary* control input c. When $c = 0$, the circuit will treat both vectors A, B as **unsigned integers**. When $c = 1$, then the circuit will treat both vectors as **two's complement** numbers. The circuit will produce three (one-bit, Boolean) outputs $F1, F2, F3$.

- When $A > B, F1 = 1, F2 = F3 = 0$.
- When $A == B, F2 = 1, F1 = F3 = 0$.
- When $A < B, F3 = 1, F2 = F1 = 0$.
- Depending upon whether the input $c = 0$ or 1, the comparisons have to be made for unsigned or two's complement schemes, respectively.

III. The Assignment

- Design the circuit describe above. Think it through. Think in terms of hardware designs.
- Write a **behavioural Verilog description** for the same using the always statement. Be careful when writing the behavioural code. I have shown you in class how a poorly written Verilog code may generate poor quality circuit, or maybe even a wrong circuit for the design.
- Write a testbench to simulate your design "exhaustively".
- Synthesize the design and observe the schematic and the synthesis reports. Trace the schematic and convince yourself that the design is indeed combinational and there are no latches at the output nodes!
- How many LUTs and slices have been used? What is the delay of your circuit?

Feel free to use if-statements, case-statements, comparators, MUXes, whatever; but write the whole thing in one always block. For your reference, I have uploaded some sample Verilog files on the class website (Lecture 14, Feb 22) that we studied in class last week. Also, Section 4.6 in the textbook gives a description of all Verilog operators that you have at your disposal.

Deadlines: The entire class should demonstrate the final circuit operation (simulation and synthesis results) to the TAs during their respective lab sessions next week (3/6-3/12). You may then write your report and submit it on Canvas before midnight Monday 03/12 (11:59pm).

Report: As usual, your report should have a **brief** write-up of your objective, computations (if any), your approach (how did you think about the logic?), the Verilog Code, observations, results and conclusions. Attach appropriate code, test benches, waveforms, or whatever you deem necessary.

In the next lab, we'll move onto sequential circuit design (circuits with memory) using always blocks.