Lab Assignment 3: Unsigned and Two's Complement Comparator Design

ECE/CS 3700
Fall 2024

Assigned: Oct 2 Due: Oct 27

Students should be able to finish this Lab, including their checkoff, in one lab session.

I. Laboratory Objectives

The purpose of this short, 1-week lab (not counting fall break) 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 signed 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 signed integers or two's complement schemes, respectively.

III. The Assignment

- Design the circuit described above.

- Write a behavioural Verilog descripton for the same using the always statement. Be careful when writing! Poorly written behavioral code may generate a poor quality circuit, or maybe even a wrong circuit for the design.

- Write a testbench to simulate your design to demonstrate all comparison cases, and all corner cases.
- 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 (memories) at the output nodes!

Feel free to use if-statements, case-statements, comparators, MUXes, whatever; but write the whole thing in ONE always block.

Note: Section 4.6 in the textbook gives a description of all Verilog operators that you have at your disposal.

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 schematic, the Verilog Code, observations, results and conclusions. Attach appropriate code, test benches, waveforms, or whatever you deem necessary.

In the next lab, we'll expand on your knowledge of behavioral constructs (always blocks) by implementing a sequential circuit (a circuit with memory).