CPE/EEE 64: Lab One

Introduction to Verilog

I. Introduction

Verilog is a powerful way to describe circuits. Logic diagrams of that are being used in lecture can become cumbersome as designs become large. Text based design can be less prone to error because it is easier to track differences in large designs. Verilog is commonly used for ASIC design as well as FPGAs. We will be using the Terasic DE0-Nano development board with an Altera Cyclone IV FPGA on board. Altera provides a comprehensive solution for programming and debugging their FPGAs called Quartus. These labs will explore Quartus' and use it to program the FPGA on the Nano. In this lab we will:

- Instantiate a System Verilog module
- Use a System Verilog Testbench
- Bink some LED's

A. Verilog Modularity

One of the most important features of Verilog is it's ability to reuse a design. This section will involve building logic gate modules which will be reused in later labs to build more complex structures. Reusing these modules is very similar to how you would reuse code in the workplace to be more productive. The lab documentation comes with Verilog implementations of the four logic gates in **Source.zip** The demo for the lab will be implementing these modules with the DE0-Nano development board and testing the design on a breadboard. You could think of this as the source libraries that would be available at the company that you might work for.

B. Test Bench

Verilog roughly breaks into two halves synthesizable and non-synthesizable. FPGAs synthesis can take a very long time, using a simulator to verify individual modules can be much faster than resynthesizing the entire design. The Testbench also offers a unique ability to check expected outputs generate test stimulus. We will use a test bench to check the provided verilog modules are providing the desired operation in part C of the procedure.

C. Altera tutorials

Altera offers a complete traning course for the interested. You have to have a free account on Altera.com **protip: do it.**

II. Lab Procedure

A. Install Quartus

Download most recent version of Quartus and Cyclone device drivers from Altera. Screencast 1 is a walk through for windows.

B. Expand Source.Zip

The logic gate modules and test bench are contained within this archive, first one's free.

C. Run testbench

Open Modelsim and compile the test bench. Start a new simulation and add the waveforms as shown in screencast 2.

D. Program and test the DEO-0 nano

1) Create top module and assign pins: refer to screencast 3 for a walk through. Verilog wires will need to be assigned to output pins on the FPGA package. The DE0-Nano instruction manual has tables that you can look up the values you need for the assignments.

2) Use Quartus to program the Nano: refer to screencast 4 for a walkthrough. After synthesis Quartus will generate a .SOF file that can be used to program the FPGA using Quartus' programmer.

3) Test behavior against expected truth table: prepare a table to expected outputs from the known properties of the logic gate.

III. Lab Report

The lab report must be typed and submitted as a .PDF. Look to IEEE's guidelines for publishing for directions on format. Include