https://github.com/kylewold/ECE230LAB-lab2

Lab Group 51: Brenen Elliott and Kyle Wold

Lab 02 - First Verilog

In this lab, you've learned how to do an initial and simple design in Verilog to learn the Vivado tooling and process involved in RTL/FPGA design.

Rubric

Item	**Description**	**Value**
Summary Answers	Your writings about what you learned in this lab.	25%
Question 1	Your answers to the question	25%
Question 2	Your answers to the question	25%
Question 3	Your answers to the question	25%

Lab Summary

Today in the lab, we were able to learn a few new things. First of all, it was a good opportunity to get acquainted with Vivado and its user interface, which turns out to have a lot of features, maybe even an overwhelming amount. Also, having no experience with Verilog, it was good to get a brief look at its syntax and the control that this language gives the user. Of the features that we learned, some important ones to note are how to set up the project, simulating, and connecting the hardware once we finished doing the coding portion through Generate Bitstream.

Lab Questions

1 - Describe the stages of building a Verilog project in Vivado.

To build a project, we first need to click "Create Project" on the Vivado homepage. We then proceed to give the file of the project a name and a directory to be placed in, select it as a RTL project, and add both of our source files and constraint files. The last thing to do before starting the project is select the hardware we are going to use.

Once the project has been loaded, we can modify the code of the source files in order to assign switches to different parts of the circuit. Given test code, we can run a simulation to ensure that our code is working and meeting the standards given to us. Once passing the test, we are able to Generate Bitstream and use the Hardware Manager to add whatever hardware we are using. We finally choose a Program Device and run the code on the hardware.

2 - What is the value in looking at the elaborated design schematic?

The elaborated design schematic allows us to get a visual graph behind the logic of the design we are making. As our designs get bigger and our logic gets more complex, the elaborate design schematic gives us an easier way to showcase and understand what is happening. Within this lab, because we were directly assigning our switch to the led light, this graph is simply a line between the input and the output. But, as we add AND, OR, XOR, NAND, and NOR logic to our designs, those will also show up.

3 - Why should we simulate our designs frequently? What does the simulation do?

As the lab guide says, simulating our designs is like debugging code before shipping it off. When a simulation is run, it is running our test file. We can find any shortcomings or errors in the design on the software-based simulation before trying it out on the hardware. Because of the simulation we are able to find bugs faster, allowing us to spend less time debugging.

Code Submission

Upload a .zip of all your code or a public repository on GitHub.