ECE 3700 Lab 01: Two Bit Computer

September 12, 2024

Rubric

Tasks	Point value
You came up with a Boolean function and/or collection of logic gates that solves the problem	2.5
You generated Verilog from a block diagram and included two screenshots (.bdf and .v files)	5
Structural code explanation and screenshot	2.5
Functional code explanation and screenshot	2.5
Testbench code, discussion, and simulation (with screenshots)	10
Checkoff	5
Overall report readability	2.5
Total	30

Objectives:

The point of lab 1 is to teach you how to:

1. Verify a circuit does what you expect by writing a testbench for it and simulating that testbench.

2. Download a program (.v file) onto your FPGA to verify it works.

3. Describe hardware in three different ways (You will create the same circuit three different times):

   i A block diagram (.bdf file)

   ii Structural code

   iii Functional code

   Your reports do not need to follow any format or meet a length criteria. They just need to describe the process you followed to complete the lab while showing me that you've met the objectives.

Some things I expect to see in the report for lab1:

1. A picture/screenshot of your .bdf file.
2. The Verilog code generated by the .bdf file.
3. Your structural code
4. Your functional code
5. Your testbench code
6. A screenshot of your simulation showing the waveform and the console output so I know you got it to work.

   If the entire console output doesn't fit in your screenshot upload another picture or text file showing the full output.
7. The lab instructions ask you to mention the number of logic blocks in your circuit and find the critical path. You can ignore this. We will discuss timing and area concerns in Lab 2.

Your submission should include a report written as a word or PDF file. You can either insert pictures/text of all the things I've asked for into a single file or you can submit multiple files and refer to each one in your report.

Reports

You are not limited to this structure but can use it as a guideline:

Lab Objectives

....

Introduction to Assignment

Describe the problem as you understand it. ALL submissions that earn full credit will demonstrate that the student understood what they were being asked to do. If I can tell you don't know what the assignment asks for but you have a valid solution then I will probably think you copied your friend's solution.

Approach

Given the specifications for the circuit, explain how you came up with your Boolean Function and/or collection of logic gates to describe F_0 and F_1.

Implementation (Verilog Code)

Prove you wrote code by including screenshots. If your code is complex enough that someone who knows Verilog can't understand from looking then describe the code (but don't go overboard in your explanations. I am grading a lot of reports and well-written code with comments can speak for itself).

Testbench and Simulation

Show me that you know what a testbench is and that you did the simulation (include code and screenshots). When it comes to your Verilog skills, writing testbenches is one of the most important things you will take from this class and so they will be a part of every single lab and I will take them particularly serious when I grade. Read and understand this summary I've written:

- A Test Bench is when you instantiate one or more modules to be tested within another module that does the testing. You simulate various input valuations and observe the outputs caused by each one. In a self-checking testbench, you will perform some action depending on if the output is what you desired. If your testbench is not ”self-checking” then you are simply observing the outputs and not reacting to them. Self-checking test benches are generally preferred for complex designs because there will be a lot of test cases to cover and it is impractical to manually verify the output of each one. Note: I will not grade you differently if your testbench is self-checking or not. This is only a concern of style and good practice.

- In Verilog, data can exist in two places. A wire and a register. The difference between the two is basically memory. Registers can remember the values they are given and wires cannot. Therefor wires must be constantly 'driven'. In A testbench, the module you're testing is constantly driving its output so you will connect that output to a wire. The inputs however must be assigned test values that they can remember long enough for the 'Unit Under Test' to produce outputs based on that input. Therefore the testbench will use registers to hold test cases and feed those registers to the input of the units under test.

Programming the FPGA

If you got checked off then I know you programmed your board. But for the sake of completeness, you can say something about how you assigned the pins or show me a picture of your completed "Pin Planner".

Conclusion

Review and reinforce the objectives. You can also discuss errors and difficulties you had. Some of you will have implemented your circuits wrong and it will show in your testbench. Others will have approached the problem completely wrong. For lab 1, I will allow you to be incorrect and still earn full points as long as you know why you're wrong and I know you know why just from reading your report. If you were wrong, don't just state it. Tell me what the right thing you should have done is.

In future labs, correctness will be more important.