|  |
| --- |
| EE260 Lab 1 |

**Submitted by:**

**Kenny Uehara**

|  |  |
| --- | --- |
|  |  |
|  | Lab partner: Sam Jung |
|  |  |
|  |  |
|  |  |
| EE260 Lab Section | 2 |

|  |
| --- |
| **Introduction** |

As we begin our adventure into digital design, we are introduced to many components that are used to engineering circuits. Many of which we saw for the first time and had the chance to experiment with in our first lab. Lab 1 allowed us to familiarize ourselves with Vivado, which is a program used to help us with the process of design entry, synthesis, implementation, and verification. This lab also helped us become familiar with the more general process of digital design which starts from ‘application’ and ends with a ‘physical’, tangible result. In our case, the HDL implemented in the FPGA used certain logics that allowed us to flip slider switches on the board so that the LEDs on the board illuminate in a predetermined way according to the logic.

|  |
| --- |
| **Methodology** |

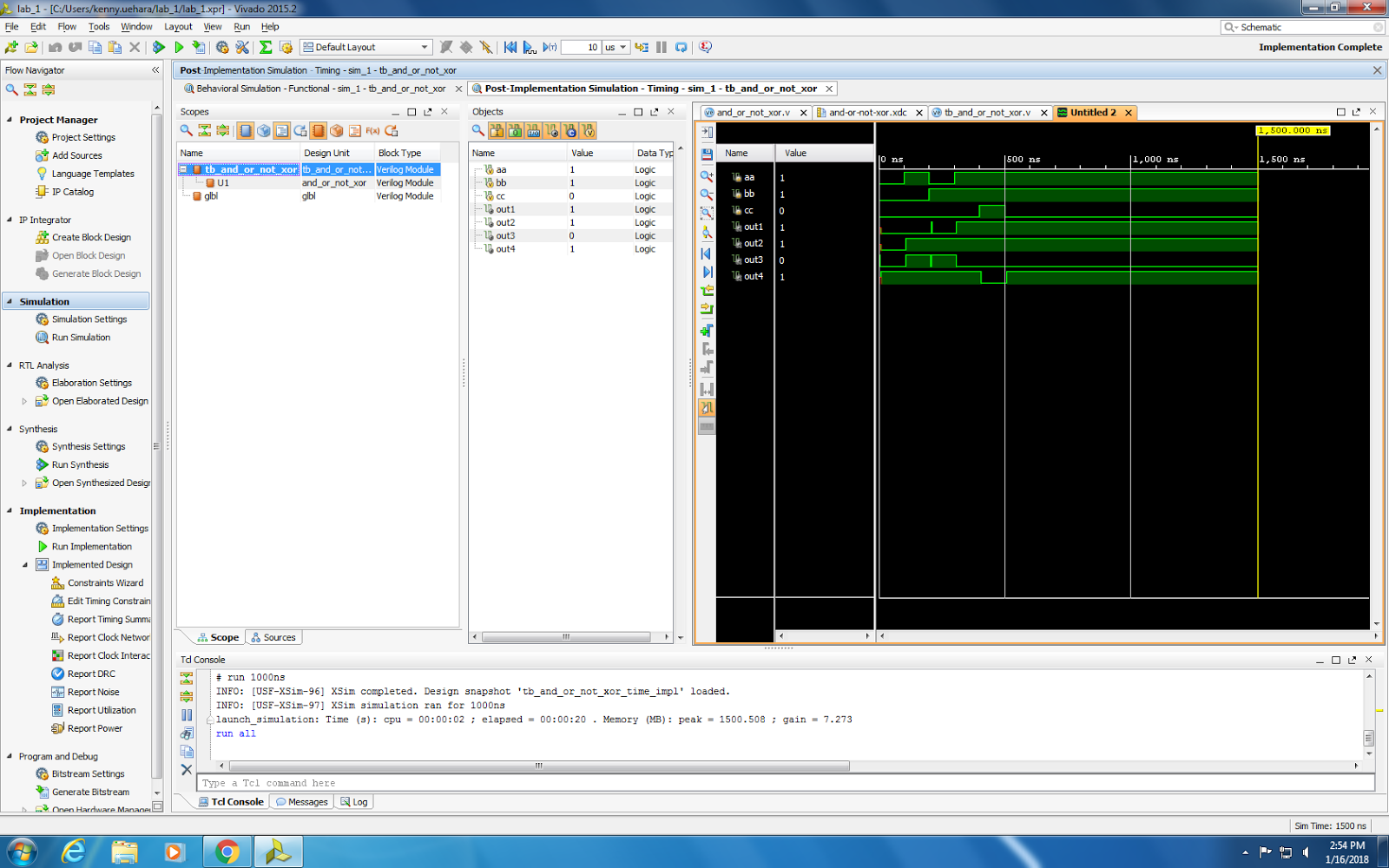
1. Run Vivado Design Suite and create a new project
2. After naming your project, select the RTL project type and make sure to check the “Do not specify sources at this time”.
3. Select the correct Xilinx Artix 7 FPGA (XC7A35T-1CPG236C)
4. Start your design entry by selecting Add Sources under project management and create file
5. The file made is to specify the inputs and outputs to create AND, OR, NOT, and XOR gates
6. Enter the code in the lab manual into the file
7. Run synthesis
8. Add constraints by selecting Add sources then add or create constraints
9. Type in the code into the constraint file
10. Next we need to generate a test-bench for verification
11. Create a file with “tb” in the name in the simulation sources
12. Set the file as a top module
13. Type the code into the file
14. Select the test bench file and click run simulation then run behavioral simulation
15. A timing diagram of the simulation should appear
16. Now we need to synthesize the design for the Basys3 FPGA device
17. Click run implementation
18. Select the test bench and run simulation/run post implementation timing simulation
19. Next we need to download the file into the Basys3 FPGA
20. Click generate bitstream (this will generate a downloadable bitstream file)
21. On the Basys3 board, make sure the JP1 is set to the JTAG mode and the board is connected to the computer via USB cable
22. Turn the board on
23. Xilinx xc7a35\_0 should appear after auto connecting
24. After downloading the file into the board, you can test the the design by toggling SW0, SW1, and SW2 and by looking at which LED lights turn on
25. We need to write the program to the QSPI serial flash on the Basys3 board so we can run it offline

|  |
| --- |
| **Discussion** |

The objective of familiarizing ourselves with the Vivado program and overall design process was completed thanks to the straightforward instructions. Although carrying out the lab took some time because of the “search and peck” method we used on the somewhat, overwhelming Vivado interface, the unequivocal instructions of the lab left us with almost no questions. We found that most of ambiguity comes from the lack of experience using the Vivado program. Another item that was vague at that time was the charts of the test bench simulation which was later clarified by the informative lab proctor and will be later discussed in this report. As we progressed through the lab we found ourselves going through the motions of the digital design. It was not apparent at first but in retrospect we can break down the steps of the lab into the general form of design entry, synthesis, implementation and verification. The first few steps of choosing which board we are using, and entering the logic ‘code’ as a source can be considered design entry. Next, we synthesized using the constraints we put in and also tested our code in a virtual setting via Vivado. Implementation included steps such as selecting the specific FPGA that we used and loading the our sources into the board. And finalizing the process included verifying our work by generating a bit stream, and making sure our board behaves properly even while not connected to the computer.

|  |
| --- |
| **Results** |

Although we found much of the lab to be fascinating, working hands-on with the Vivado program and obtaining a physical result was also very interesting. Another intriguing part of the Vivado program was the result of the test bench.



We expected the aa input to equal 0 and bb input to equal 1 simultaneously which would theoretically have no output on output1, but instead there is a small glitch. This small flare up on the output was explained by the lab proctor as the delay within the circuit due to physical means. In application, it would probably be negligible because the scale is in nanoseconds or it could have been prevented by adding a wait time between the aa and bb input signals. Other satisfying results we found was the successful implementation of the design on the board and interacting with the LEDs on the board to confirm a predicted result.

|  |
| --- |
| **Conclusion** |

Conclusion convincingly describes what has been learned in the lab.

Overall, we believe this was a very informative and successful lab. By executing this lab we were able to familiarize ourselves with the general process of digital design, and learned how to use the Vivado program. The lab guided us through defining modules and designing logic gates, then synthesizing and viewing virtual schematics, also adding constraints based on the specific board that we used. We also benched tested the logic design via Vivado and was also able to see the results on a time scaled graph. We were then able to implement the digital design into the board to obtain a physical output and result. The experience and skills we gained from carrying out this lab are definitely going to be the solid foundation which we will use throughout this semester as we delve deeper into the digital design world.