Laboratory #1

Introduction to CAD tools and Verilog

Prepared by: Devin Buzzetta and Jesus Ramirez De La Pena

Course: EENG 284

Section: LC

Prepared: January 26, 2019

Instructor: Professor Sager

Evaluation: Devin Buzzetta

|  | Percentage | Score |
| --- | --- | --- |
| Purpose | 10 |  |
| Procedure | 20 |  |
| Results/Successful Demonstration/ Correct Simulated Results | 40 |  |
| Conclusions | 10 |  |
| Pre-Lab | 10 |  |
| Post-Lab | 10 |  |
| Final Score | --- |  |

Evaluation: Jesus Ramirez De La Pena

|  | Percentage | Score |
| --- | --- | --- |
| Purpose | 10 |  |
| Procedure | 20 |  |
| Results/Successful Demonstration/ Correct Simulated Results | 40 |  |
| Conclusions | 10 |  |
| Pre-Lab | 10 |  |
| Post-Lab | 10 |  |
| Final Score | --- |  |

**Purpose:**

The purpose of this lab is to get acquainted with the Quartus II software, to design entries for the software using Verilog, and to run circuit simulations with them.

**Procedure:**

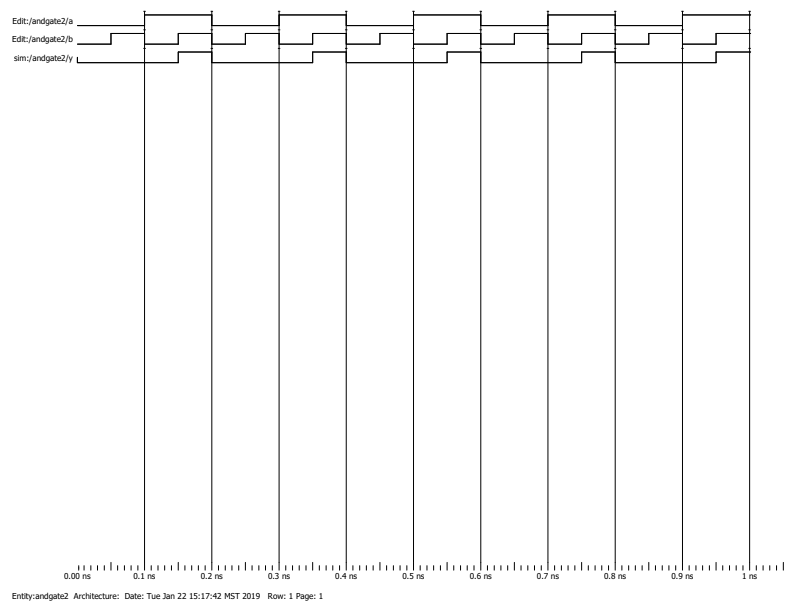
First the Quartus II software will be opened and a new project folder named *andgate2* will be created. Then, the files andgate2.v and andgate2\_tb.v will be added. The andgate2.v file will be filled out in Verilog and compiled. Then it will be ran as a simulation in ModelSim until it is error free. The andgate2\_tb.v file will then be filled out in Verilog as well and compiled. This file will also be ran as a simulation in ModelSim until error free (shown by the absence of red lines in the simulated waveform).

A new project folder will be created in Quartus II with the *simplefunction.v* and *simplefunction\_tb.v*. These files will be completed according to the schematic in the prelab and compiled. A simulation will be ran for the *simplefunction\_tb.v* testbench file in ModelSim until error free (shown by the absence of red lines in the simulated waveform).

Finally, a new project folder will be created using the *counter.v* file, this file will be ran as a simulation in ModelSim without a test bench until error free (shown by the absence of red lines in the simulated waveform).

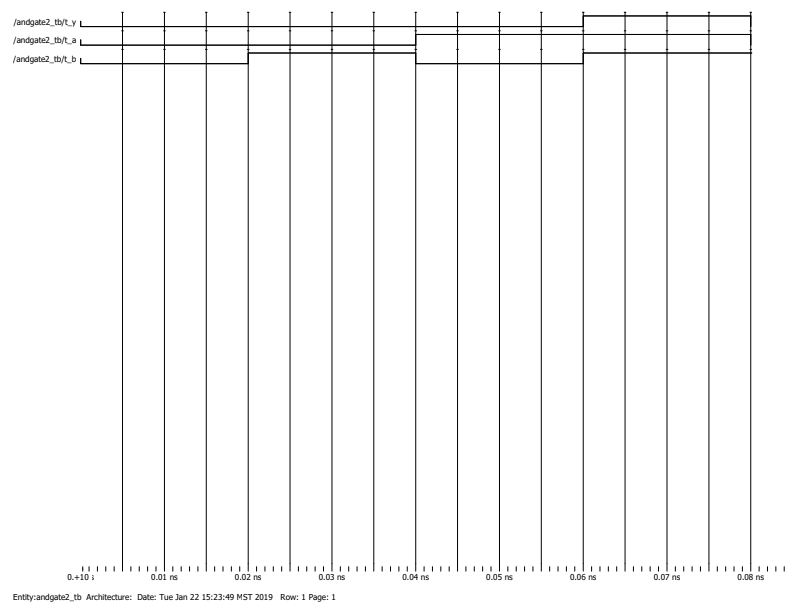
**Results:**

After the andgate2.v file was simulated without a test bench, an error-free simulation waveform was created (As shown in Figure 1).

****

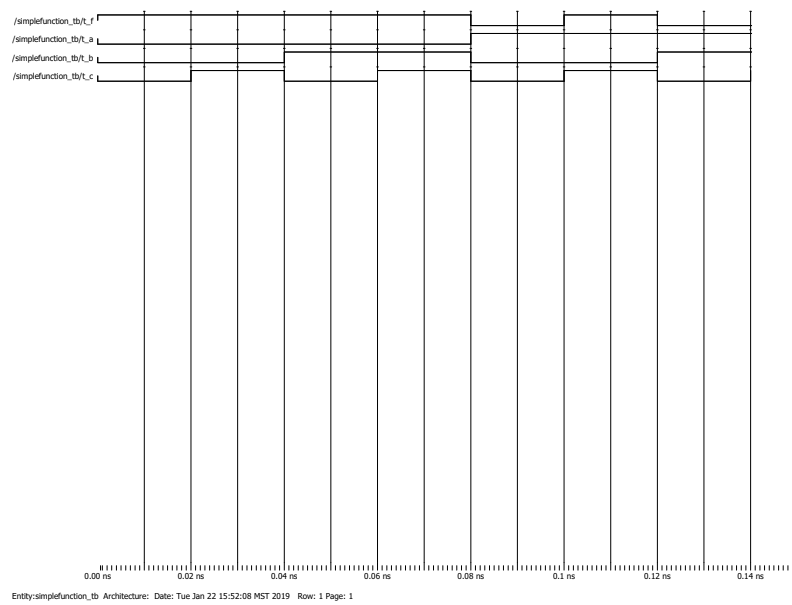
**Figure 1:** This is the simulation waveform of andgate2.v.

After the andgate2.v file was simulated with the test bench andgate2\_tb.v, an error-free simulation waveform was created (As shown in Figure 2).



**Figure 2:** This is the simulation waveform of andgate2.v using the test bench.

After the simplefunction.v file was implemented with the project team’s code (As seen in Listing 1), it was simulated with the test bench simplefunction\_tb.v. The simplefunction\_tb.v had to be modified to include each test case. This was done by writing the verilog code making eight different input cases so that all possible combinations of the inputs were tested (As seen in Table 1). After the code was implemented, an error-free simulation waveform was created (As shown in Figure 3).

****

**Figure 3:** This is the simplefunction.v simulation waveform.

**Table 1:**

| Input: | 1st Test | 2nd Test | 3rd Test | 4th Test | 5th Test | 6th Test | 7th Test | 8th Test |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| b | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| c | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

The inputs to simplefunction\_tb.v that were implemented.

**module** simplefunction(F, A, B, C);

**output** F;

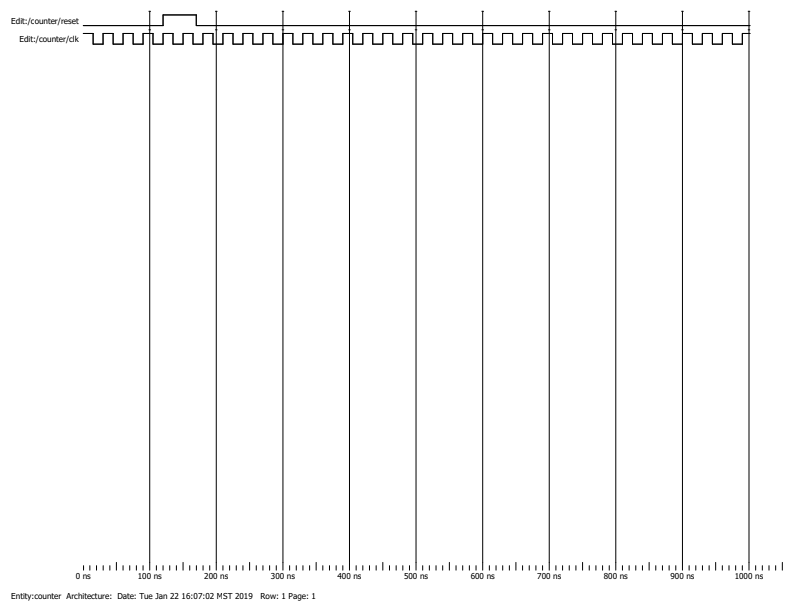
**input** A,B,C;

**assign** F = (!A) || (!B && C);

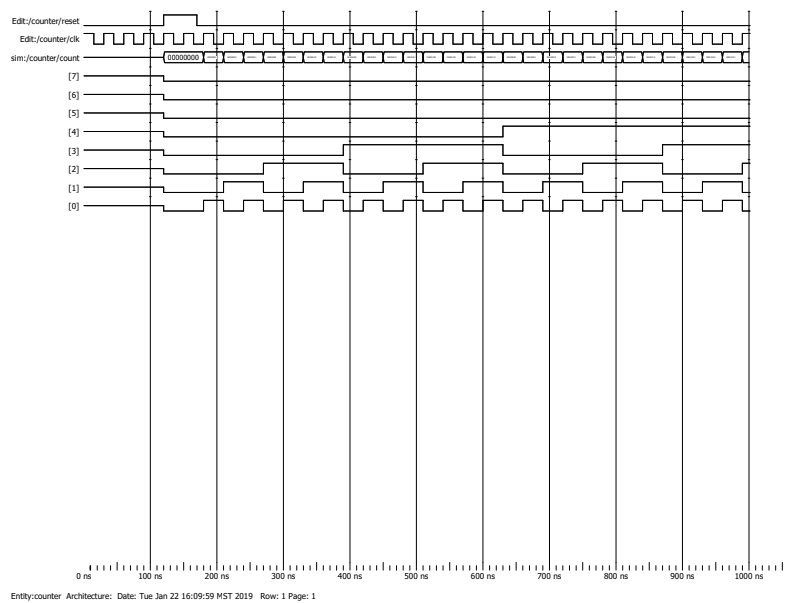
**endmodule**

**Listing 1:** This code allows the simplefunction.v to test the different A, B, and C inputs through the function.

After the counter.v file was simulated without a test bench, an error-free simulation waveform was created without the counter (As shown in Figure 4), and another error-free simulation waveform was created with the counter (As shown in Figure 5).



**Figure 4:** This is the counter.v function simulation waveform without the count.



**Figure 5:** This is the counter.v function simulation waveform with the counter.

**Conclusion:**

After some modifications to the code in andgate2.v, counter.v, and the test bench for simplefunction.v the software was able to run all of the programs. All of the outputs of each function came out correct with the needed changes. This was seen in each of the outputs by not having red lines as seen previously in figure 1, figure 2, figure 3, figure 4, and figure 5. The Quartus II software was a new software to the project team conducting the lab. The project team learned how to use the software to enable them to visualize the results of logic gates represented in verilog code.