**Laboratory #2**

Design Combinational Circuits and Hierarchy

Prepared by: Devin Buzzetta and Jesus Ramirez De La Pena

Course: EENG 284

Section: LC

Prepared: February 3, 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 objective of this lab is to design simple digital circuits, as well as self-checking test benches, and introduce the concept of hierarchy for complex system design. This means that when a test bench is ran, a message will be outputted displaying whether or not the test values for the inputs resulted in the expected output. This also means that once a simple digital circuit is designed, a more complex digital circuit can be designed while incorporating the previously created simple circuit.

**Procedure:**

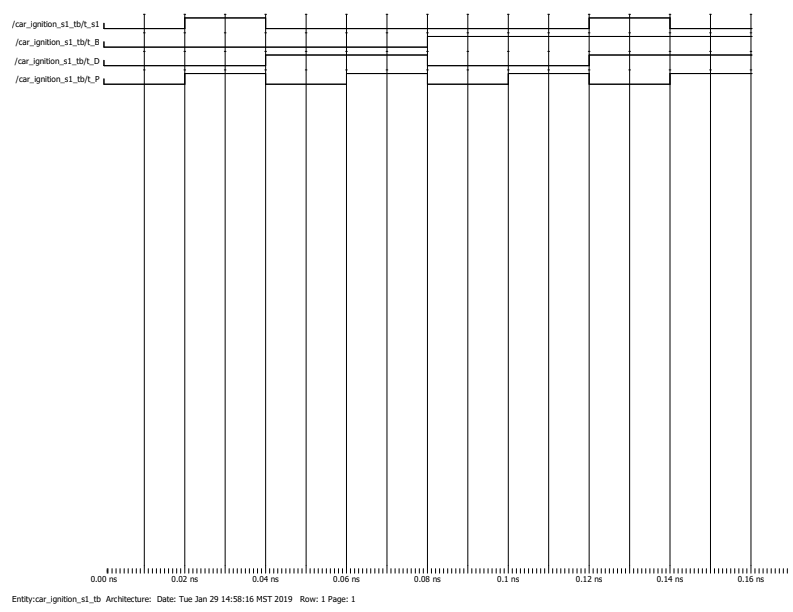
The first step of this procedure will be to read and understand the instructions and problem statements. A new project folder will be created in Quartus II with the *car\_ignition\_s1.v* and *car\_ignition\_s1\_tb.v* files added to the project folder. The *car\_ignition\_s1.v* file will be completed according to the first schematic in the prelab and compiled. The *car\_ignition\_s1\_tb.v* testbench file will be completed to test every input combination and to be self checking by having it display a message on whether or not the test values for the inputs resulted in the expected output and compiled. A simulation will be ran for these files in ModelSim until error free (shown by the absence of red lines in the simulated waveform, and the self checking test bench not displaying error messages). This will then be repeated once again except using the second schematic in the prelab with the files being *car\_ignition\_s2.v* and *car\_ignition\_s2\_tb.v*.

A new project folder will be created in Quartus II called “fulladder” with the *fulladder.v* file added to the project folder. The *fulladder.v* file will be completed so three 1-bit inputs are successfully added together and two 1-bit numbers are outputted representing a two bit sum of the three inputs and compiled. A self checking testbench file called *fulladded\_tb.v* will be created to test every possible input combination and compiled. A simulation will be ran for these files in ModelSim until error free (shown by the absence of red lines in the simulated waveform, and the self checking test bench not displaying error messages).

Finally, A new project folder will be created in Quartus II called “rca4” with the *fulladder.v*, *rca4.v*, and *rca4\_tb.v* files added to the project folder. The code in *rca4.v* will be modified and completed using verilog so that it adds two four bit binary numbers and outputs a five bit binary number made up of a four bit binary number and a one bit binary number in the front to account for overflow. This will be done by having the code in *rca4.v* call upon the *fulladder.v* file four separate times for each separate bit in the two inputs. A table with the different inputs and their expected outputs will be completed, and the self checking testbench file will be written to test for the inputs shown on the table and then compiled. A simulation will be ran for these files in ModelSim until error free (shown by the absence of red lines in the simulated waveform, and the self checking test bench not displaying error messages).

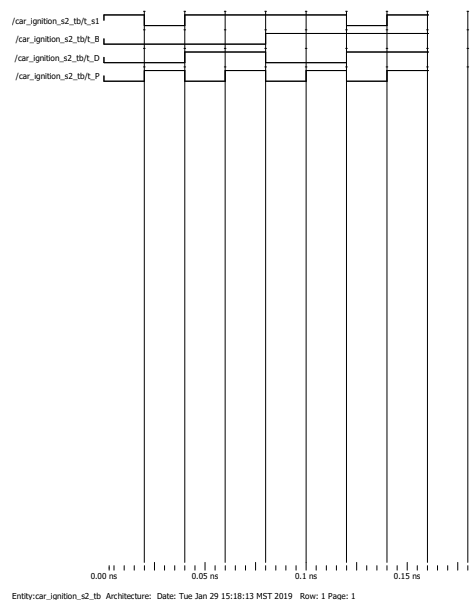
**Results:**

The project team was able to successfully run the lab tests. In part one of the lab, as seen in Figure 1, the team was able to correctly simulate the car when it wouldn’t start. The car wouldn’t start when the doors are closed and seats are unbuckled , if the seatbelt is buckled and the parking brake is on, or if the parking brake is off and the doors are open.

****

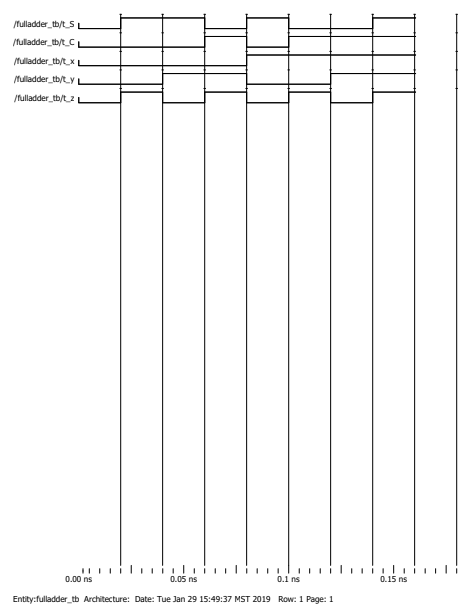
**Figure 1:** This is the result from analyzing the car\_ignition\_s1.

The lab team then went on to simulate the cases when the car did start. This was the result of inverting the code in part 1. As seen in Figure 2, the very top row (the output) is inverted from Figure 1 (every two segments in Figure 1 is the same as a single in Figure 2).



**Figure 2:** This is the result from analyzing car\_ignition\_s2.

The next part of the lab was to create the full adder function. The function was supposed to add numbers together and output a sum and a carry if there needed to be one. As shown in Figure 3, the carry doesn’t always happen.

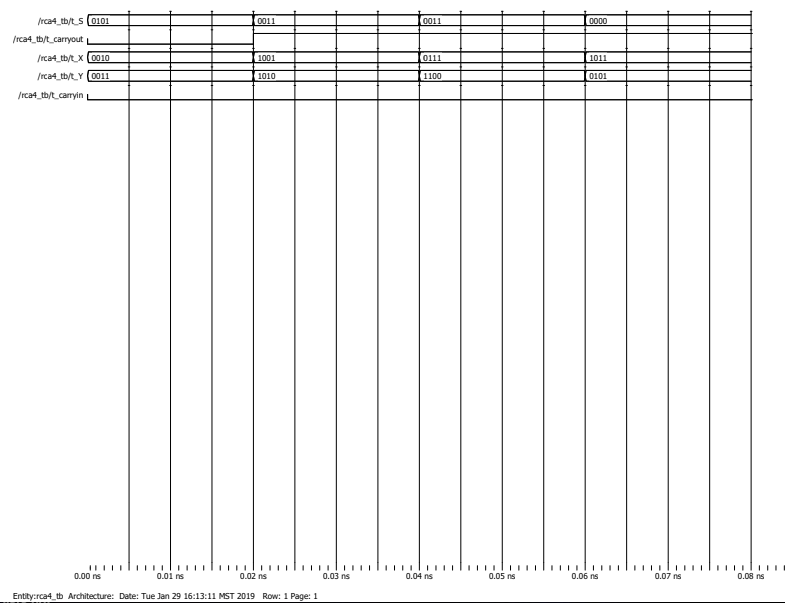


**Figure 3:** This is the result from analyzing fulladder.

The final part of the lab was to make rca4. This was the result of using four of the full adder functions back to back. In order to do this the project team had to make sure that the numbers worked properly. The result is in the “S” row in Figure 4. If there was a carry it is denoted in the “carryout” row to be used in the next calculation. Table 1 was able to be filled out as a result of Figure 4 with the correct data from testing.

**Table 1:** Data table with the results from running rca4. The underlined is filled in data the project team was looking for.

| A3:0 | B3:0 | S3:0 | C4 |
| --- | --- | --- | --- |
| 0010 | 0011 | 0101 | 0 |
| 1001 | 1010 | 0011 | 1 |
| 0111 | 1100 | 0011 | 1 |
| 1011 | 0101 | 0000 | 1 |



**Figure 4:** This is the result from analyzing rca4.

**Conclusion:**

Overall, every objective for this lab was met to a satisfying degree. Simple digital circuits were designed when doing the car\_ignition\_s1 and car\_ignition\_s2 projects. Self checking testbenches were written in verilog on every project. The concept of hierarchy was also a new methodology learned and implemented in this lab when writing the *rca4.v* file since the simpler *fulladder.v* file was used as a building block to create the more complex *rca4.v* file. So, every aspect of the objective was met during this lab and a new methodology was learned.