CEG2136 LAB 1
Group 21

Ryan Fleck - 8276723
Xiuzhu Li - 8571645

September 27, 2017

Contents

List of Figures

1 Theoretical Implementation

1.1 Lab Objectives

The primary goal of this lab is to familiarize the student with the *Altera DE2-115 FPGA* and it's companion software, *Altera Quartus II.* The following objectives should be met:

1. Understand the basic of the Altera environment.
2. Design a simple logic circuit using the Graphic editor.
3. Compile, simulate, debug, and test their design.
4. Download and run their design on the Altera DE2-115 board.

1.2 Discussion of Requirements

A circuit with the following logic function is to be implemented in *Altera Quartus II* and run on the *Altera DE2-115 FPGA*:

$$(A \cdot B)' + (C + D)' \tag{1}$$

A derived truth table for this function can be found at the end of this document, **Figure 4**. The goal of the lab is to implement this circuit and have it run without errors on the *Altera DE2-115 FPGA*, with every set of inputs resulting in the output predicted by the logic funciton and truth table.

1.3 Proposed Algorithmic Solution

As the equation **(1)** is already minimized, the equivalent logic gates will be used to construct the circuit.

1. A NOR gate will be used to channel A and B.
2. A NAND gate will be used to channel C and D.
3. An AND gate will be used to process the output from the NAND and NOR gates.

In total, three logical elements will be used. One each of NAND, NOR and AND gates. Please refer to *Figure 2* in the *CEG2136 Lab 1 Manual* for the original schematic the circuit is to be modeled after, and *Altera Quartus II* part numbers.

2 Design

2.1 QUARTUS II Circuit Diagrams

The circuit was implemented in *Altera Quartus II* as shown in **Figure 1**.

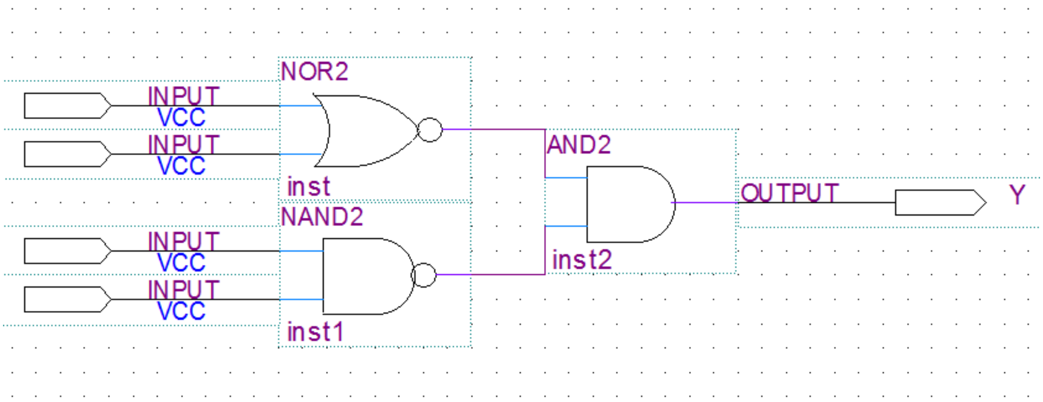


Figure 1: Implemented Logical Circuit

2.2 Components Utilized

Our design, as shown in as shown in **Figure 1**, was implemented as intended with the components outlined in the *Proposed Algorithmic Solution* subsection. No alternative components, that is, similar components with different part numbers (than those specified in the lab manual,) were used to create the logical circuit.

2.3 Implemented Solution

When implementing our solution, we took the following steps:

1. *Altera Quartus II* was opened and the file was initialized.
2. The *nor2, nand2* and *and2* gates were selected and placed in the simulation without any problems, and joined together to adhere to the logical expression from the lab requirements **(1)**.
3. Pins were added and assigned correctly.
4. The file was compiled.

5. Pin numbers for the *Altera DE2-115 'Cyclone' FPGA*, corresponding to logical inputs *A, B, C, and D* were saved to the equivalent digital pins and LEDs within *Altera Quartus II.*

2.4 Challenging Problems Encountered

Some minor compilation errors occured at the beginning of the circuit's construction, but did not occur after the pin elements were placed on the digital schematic. We also had the classic error of being unable to see the *USB Blaster* when initially trying to upload to the *Altera DE2-115 FPGA* later in the experiment. No truly challenging problems were encountered when implementing **Figure 1**. It should be noted that the search function, though sometimes hidden in *Altera Quartus II*, is very useful and came in handy during this lab. It is problematic that this program function was not used more often in ITI1100.

3 Implementation

3.1 Simulation Results

Figure 2 shows a single set of 16 clock cycles, enough for PIN A to achieve a LOW voltage for 8 cycles, and a HIGH voltage for 8 more. As predicted by the KMAP derived in the prelab, and displayed in the appendices as **Figure 4**. This series of waveforms did not deviate from the waveform defined by our truth table and logical equations.

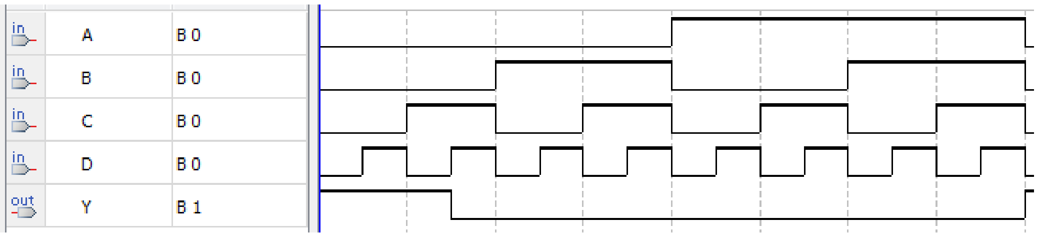


Figure 2: Logical Circuit Waveform Output

3.2 Simulation Verificaiton

After recompiling, our *Altera DE2-115 FPGA* had no issues simulating the circuit that had been designed. All 16 combinations of the 4 inputs were tested, and it was verified that the LED on the *Altera DE2-115 FPGA* only responded with a HIGH voltage to the following combinations of the four inputs:

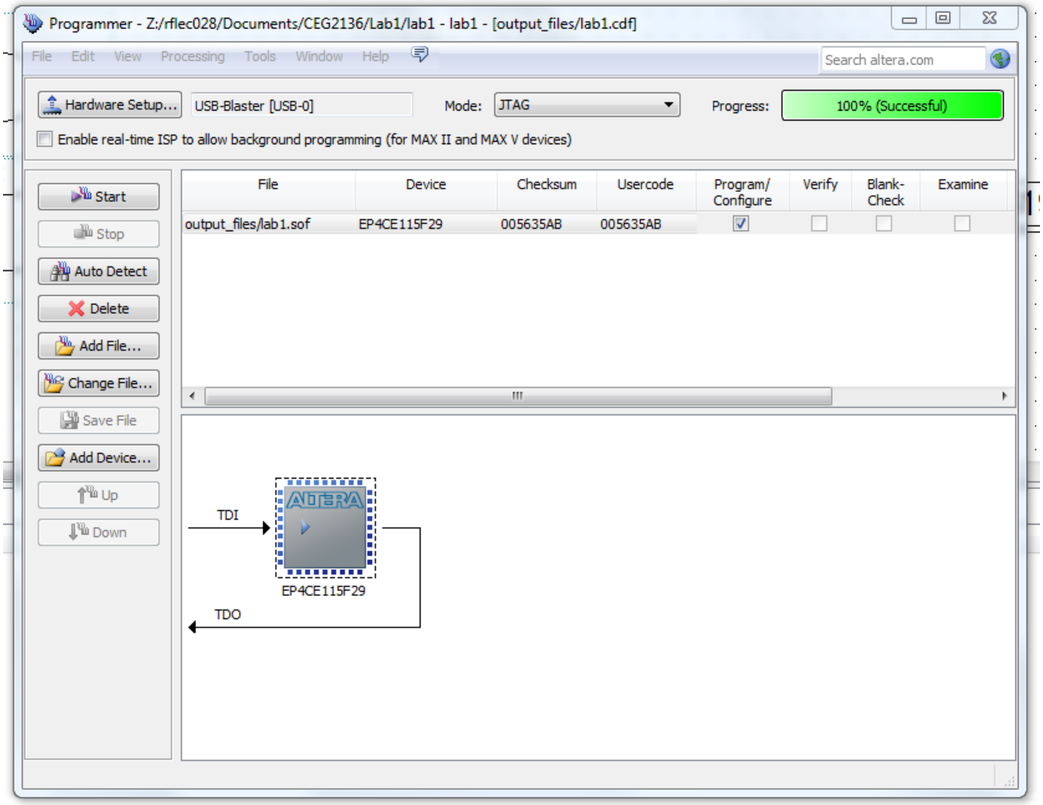
$$A'B'C'D', A'B'C'D, A'B'CD' \tag{2}$$

4 Discussion

After exhaustive testing of our running logical circuit, it was determined that it responded predictively to all possible input. There was no deviation from the behavior defined in **Figure 4** or our logic equations. There was no need to modify our circuit to fix deviant behaviour in the final waveform or physical implementation on the *Altera DE2-115 FPGA*. It was fun to reminisce about ITI1100, and hopefully knowledge of the quirks in Altera's *Quartus II* development environment will make the labs in this course smoother than those in that previous course.

4.1 Errors Encountered

Figure 3 shows our successful board upload. We had to reattempt the upload a few times, and had difficulty getting the lab computer to recognize the active *Altera DE2-115 FPGA* connected via USB.



Programmer - Z:/rflec028/Documents/CEG2136/Lab1/lab1 - lab1 - [output_files/lab1.cdf]

USB-Blaster [USB-0]

Mode: JTAG

Progress: 100% (Successful)

□ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
output_files/lab1.sof	EP4CE115F29	005635AB	005635AB	☑	□	□	□

Figure 3: Successful Upload to EP4CE115F29 Device.

4.2 Conclusions

During lab 1 we recapped the course material of ITI1100, and once again found ourselves familiar with the *Altera DE2-115 FPGA* along with its software, *Quartus II*. Combinational circuits are a fundamental component of a robust computer engineering education. Mastering combinational circuits will allow more complex topics and digital circuits to be broken down more easily due to a depthful knowledge of the small-scale components. Lab 1 emphasizes the importance of logic gates' functionality, which are crucial to comprehending upcoming sequential circuits and more. Though straightforward, CEG2136 Lab 1 was a nice refresher and starting point to what we will encounter later in Computer Architecture I. We are looking forward to the challenge.

Figure 4: Truth Table for Logic Function 1

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0