LAB 1:
INTRODUCTION TO QUARTUS II SOFTWARE DESIGN
CEG 2136 B: Computer Architecture I

Date of experiment: 09/24/2015
Report submitted: 09/30/2015

Teaching Assistant: Osama Aloqaily

Heidi Fahim 6350557
Youssef El-Khalili 7923376

TABLE OF CONTENTS

1.0 LAB THEORY

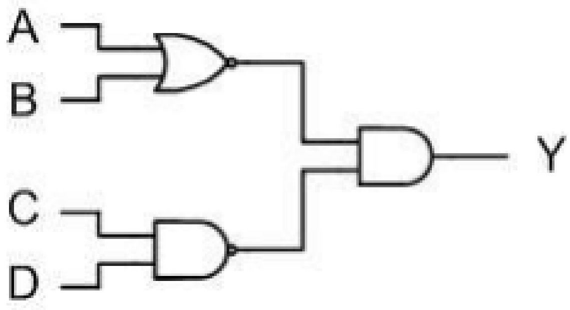
1.1 Introduction

The purpose of this lab experiment was to become familiar with the Altera Quartus II Software Design. After creating a project, designing a simple logic circuit, compiling, simulating and finally testing the design, the relationship between Quartus II FPGA design software and the Altera DE2-115 FPGA board was better understood. The Altera DE2-115 is a field-programmable gate array (FPGA) device that contains an integrated circuit that is based around a matrix of Configurable Logic Blocks (CLBs). The CLB is the basic logic unit in an FPGA, which consists of configurable switches for inputs and can be configured to handle different types of circuits.[1]

1.2 Discussion of Problem

The purpose of this lab was to implement a simple combinational circuit from a given circuit diagram which uses AND, NAND and NOR logic gates. Each of these gates implements its corresponding Boolean function, performs the logical operation on one more inputs, and then produces a single output. This output will then be passed to another gate depending on the size of the circuit, or will become the result of the function output. The output of a combinational circuit is therefore a function of the inputs and only depends on the present state of its inputs. The state of each input is set to either 0 or 1, and the respective signals of each input will be combined and processed through the logic gates yielding an output (also represented by 0 or 1). Figure 1 shows the given circuit that was given to be implemented, designed and tested in this lab experiment.

Figure 1: *A combinational circuit using AND, NAND and NOR logic gates*



In order to correctly implement, design and test this circuit the corresponding Boolean logic function was first determined. From the Boolean function, the truth table was derived and the expected outputs were determined. This circuit was then designed and tested using the Quartus II software, along with the Altera DE2-115 board, and the experimental outputs were recorded.

1.3 Discussion of Algorithmic Solution

In order to implement, design and test the given circuit correctly, four fundamental steps were taken:

1) <u>Determining the corresponding logic function</u>

   From the given circuit diagram, it is possible to define the corresponding logic function. A Boolean function is of the form $f: B^k \rightarrow B$ where B={0,1}. As seen in the circuit diagram, A, B, C, and D represent the four distinct inputs, and Y represents the single output that results after the input signals are processed through the entire circuit. In the circuit diagram, the logic gate connected to inputs A and B is a NOR gate. This represents the function $(\overline{A+B})$. The logic gate connected to inputs C and D is a NAND gate. This represents the function $(\overline{CD})$. As seen in the circuit diagram, the outputs of each of these gates are then passed to an AND gate and the result is processed, yielding the output Y.

2) <u>Deriving the truth table</u>

   From the logic function determined, the truth table was then derived. Table 1 represents the truth table corresponding to a two-input NOR function. Table 2 represents the truth table corresponding to a two-input NAND function.

Table 1: *NOR function truth table*

	Inputs		Output
	x	y	z = $(\overline{x+y})$
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0

Table 2: *NAND function truth table*

	Inputs		Output
	x	y	z = $(\overline{xy})$
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

   In order to correctly derive the truth table for the entire logic function, the number of inputs is considered and used to determine the number of different combinations of input signals that can be entered into the circuit. For a combinational circuit with n inputs, there are 2^n possible combinations of the input signals. In this case, there are 16 possible combinations (2^4), which means a truth table consisting of 16 rows is required. Using standard Boolean algebra, the expected output of the circuit under each of the 16 input signal combinations can be determined.

3) <u>Creating the circuit in Altera Quartus II Software Design</u>

   This step consists of the design of the circuit diagram in Quartus' Graphic Editor, compiling the project, assigning inputs and outputs to pin locations and LEDs, simulating the project, and finally, downloading it to the Altera

DE2-115 board in order to test and verify the experimental results. This step is discussed in more depth in section 2.3.

4) Testing

   This step consists of applying each of the 16 possible combinations of input signals and verifying that it matches the expected output results. The four inputs were assigned to four of the 18 slide switches on the Altera board and the output to one LED. The LED lights on the Altera board are active-high, which means that the output LED will illuminate if the output state is 1, and turn off if the state is 0.

✓

2.0 LAB DESIGN

2.1 Presentation of Design Methodology

As discussed in section 1.3, the logic function of the circuit was determined to be:

Y = (A+B)'(CD)'

The truth table derived from this function is seen in table 4:

***Table 4**: Truth table corresponding to Y = (A+B)'(CD)'*

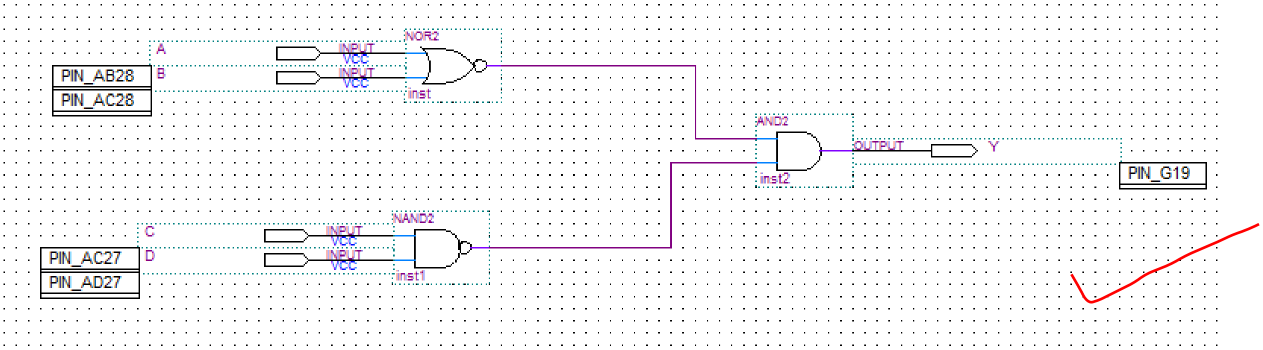
	INPUTS				GATES		OUTPUT
	A	**B**	**C**	**D**	**(A+B)'**	**(CD)'**	**Y =(A+B)'(CD)'**
0	0	0	0	0	1	1	1
1	0	0	0	1	1	1	1
2	0	0	1	0	1	1	1
3	0	0	1	1	1	0	0
4	0	1	0	0	0	1	0
5	0	1	0	1	0	1	0
6	0	1	1	0	0	1	0
7	0	1	1	1	0	0	0
8	1	0	0	0	0	1	0
9	1	0	0	1	0	1	0
10	1	0	1	0	0	1	0
11	1	0	1	1	0	0	0
12	1	1	0	0	0	1	0
13	1	1	0	1	0	1	0
14	1	1	1	0	0	1	0
15	1	1	1	1	0	0	0

✓

2.2 Discussion of Used Components

For this circuit, one NOR-2 gate, one NAND-2 gate and one AND-2 gate were used and were connected as seen in the block diagram file in figure 5.

Figure 5: *Circuit schematic designed on Quartus II Graphic Editor*



The orthogonal node tool was used to connect all the gates together as well as the generated pins for the inputs and outputs.

2.3 Discussion of the Solution

The purpose of this lab experiment to test the given circuit required creating a project in Quartus II to implement the circuit. The first part is to design the circuit schematic using Quartus Graphic Editor. This editor provides the user with symbol blocks that represent the different types of logic gates. In this experiment, one NAND, one NOR and one AND symbol block were used to design the circuit schematic in the graphic editor. These can be seen in figures 1,2 and 3.

Figure 2*: Quartus NAND symbol block*

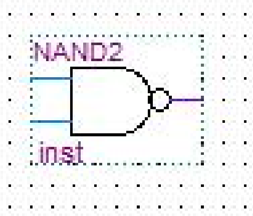


Figure 3: *Quartus NOR symbol block*

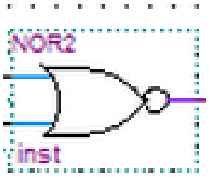
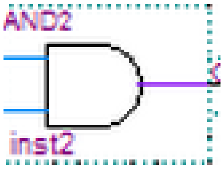
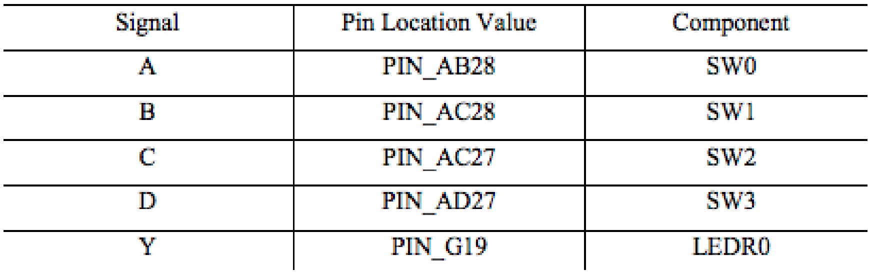


Figure 4*: Quartus AND symbol block*



Input and output pins were connected to the gates using the Orthogonal Node tool in the graphic editor. This circuit schematic was then compiled and inputs and output signals were assigned to pins according to table 3.

Table 3: *Device pin connections and names*



Signal	Pin Location Value	Component
A	PIN_AB28	SW0
B	PIN_AC28	SW1
C	PIN_AC27	SW2
D	PIN_AD27	SW3
Y	PIN_G19	LEDR0

The project was then simulated using the Waveform Editor in Quartus which allows the user to input test vectors as input to the circuit as to observe the outputs generated in response. There are two types of simulations that Quartus allows the user to do on the designed circuit: functional simulation and timing simulation. Functional simulation is done to verify the functional correctness of the designed circuit, and was not used in this lab experiment. Timing simulation takes into account propagation delays due to logic elements and interconnecting wires.[2] After the simulation was completed, the project was then downloaded to the Altera DE2-115 board in order to test the experimental outputs.

2.5 Discussion of Challenges

No challenges were encountered for this lab experiment as we have previous experience using Quartus II and the Altera boards.

3.0 REAL IMPLEMENTATION

3.1 Simulation Results

Figures 6 and 7 show the waveform files before and after simulation, respectively.

Figure 6: Waveform file before simulation

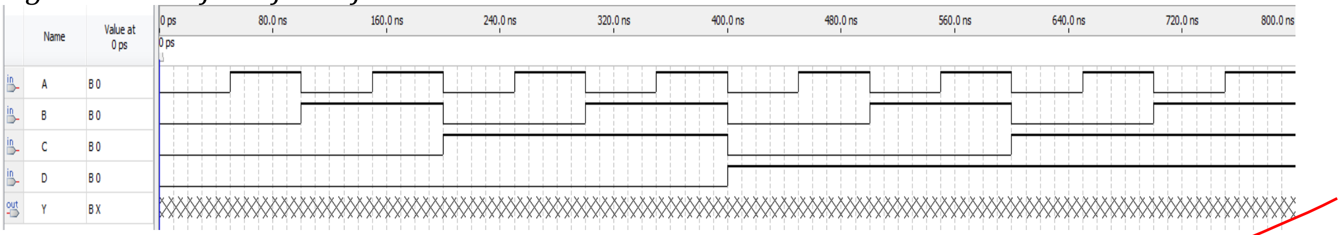
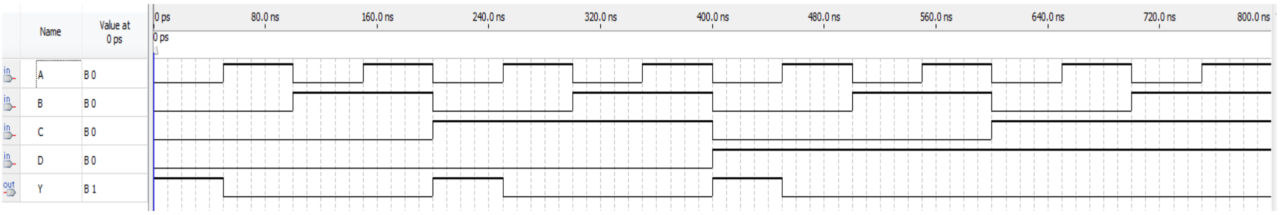


Figure 7: Waveform file after simulation



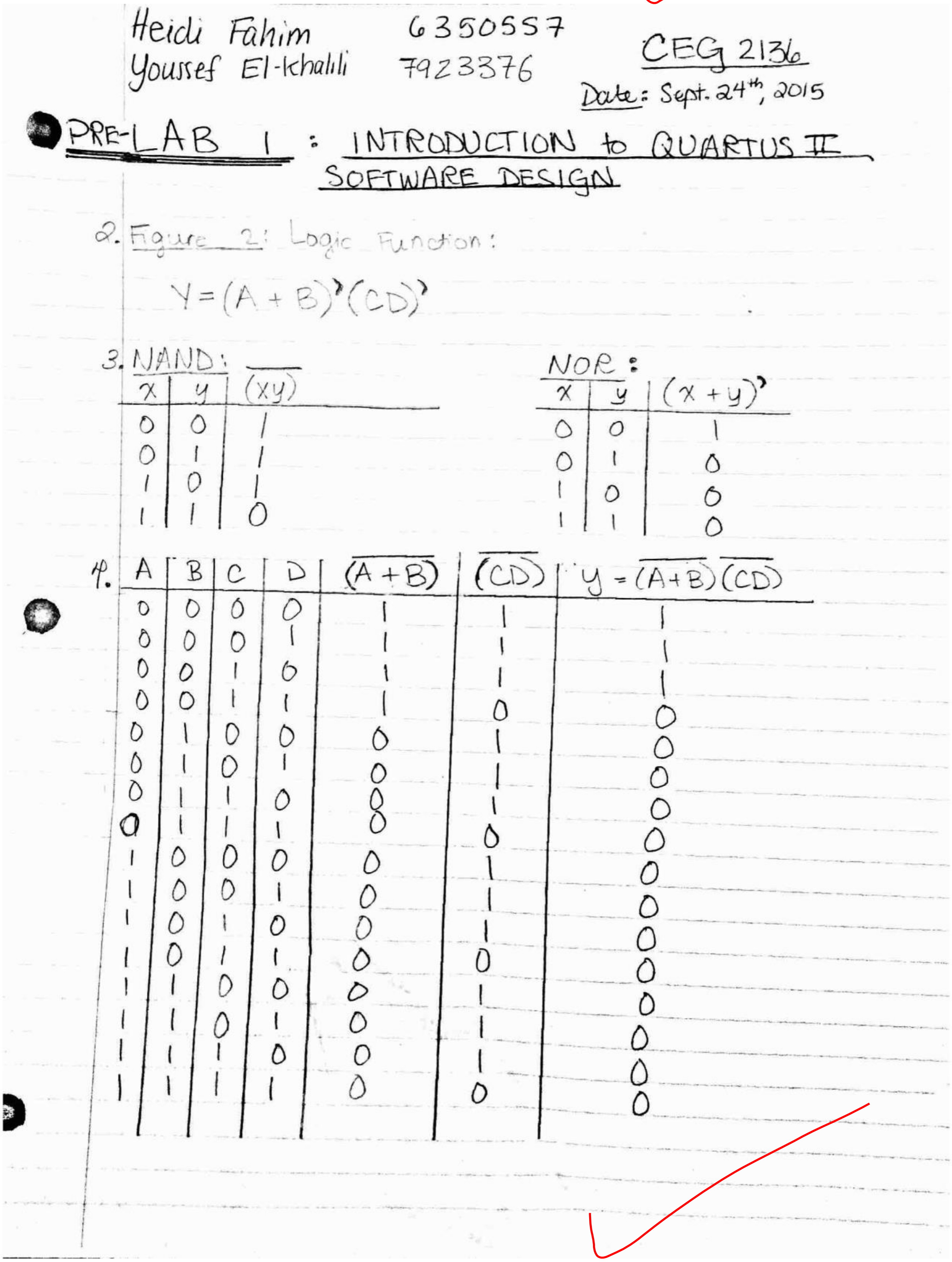
3.2 Verification

We ended up with a working design and the results we obtained through the Altera board were exactly like the theoretical predicted results we obtained in the attached pre-lab.

4.0 DISCUSSION

No errors were found in our design and everything went as expected, this lab was a good exercise for us to re-familiarize ourselves with the new Quartus and Altera software and hardware.

5.0 PRE-LABORATORY



Heidi Fahim 6350557

Youssef El-khalili 7923376

CEG 2136

Date: Sept. 24th, 2015

PRE-LAB 1 : INTRODUCTION to QUARTUS II SOFTWARE DESIGN

2. Figure 2: Logic Function:

$$Y = (A+B)'(CD)'$$

3. NAND: $\overline{}$

x	y	$\overline{(xy)}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR:

x	y	$(x+y)'$
0	0	1
0	1	0
1	0	0
1	1	0

4.

A	B	C	D	$\overline{(A+B)}$	$\overline{(CD)}$	$y = \overline{(A+B)}\,\overline{(CD)}$
0	0	0	0	1	1	1
0	0	0	1	1	1	1
0	0	1	0	1	1	1
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	0	0	0

6.0 REFERENCES

1) "What Is a FPGA?" *What Is a FPGA?* N.p., n.d. Web. 27 Sept. 2015. <http://www.xilinx.com/fpga/>
2) Corporation, Altera. *Getting Started with Quartus II Simulation Using the ModelSim-Altera Software* (n.d.): n. pag. Web. <https://www.altera.com/en_US/pdfs/literature/ug/ug_gs_msa_qii.pdf>

✓