Lab 1 – Introduction to Quartus II Design Software

COOP 2901 – Rapport premier stage

Summer 2023

University of Ottawa

Jasmin Cartier 300160723

Experiment Date: 09/22/2022

Table of Contents

Introduction

The purpose of this lab is to understand how to implement logic circuits on an Altera DE2-115 board by first designing, compiling, stimulating and testing circuit diagrams using Quartus II software. This lab will provide an interactive review of designing a circuit while also creating a deeper understanding of how to use Quartus II software in a physical lab setting. These skills will be very important for future lab experiments in this course.

Before completing the lab, truth tables and logic equations were derived from a given circuit diagram representing a full adder. The function of this adder was to take 3 inputs, A, B, and Cout, and return the sum of A and B along with the carry value from this operation. In this case, S represents the sum and Cout represents the carry value. In order to fulfill this function, the full adder required 1 OR gate, 2 AND gates, and 2 XOR gates. Using this information, the circuit for this full adder needed to be designed in Quartus II and programmed onto the Altera board where the output would be produced.

Design

The full adder design was given by the lab instructions and there is a reproduction of it in Figure 3. It implements the functions of S and Cout and the K-maps for these functions can be seen in Figure 1 and 2 respectively. These K-maps were based on the information in Table 1. By simplifying the equations, the resulting functions are; S = Cin$\oplus$(A$\oplus$B) and Cout = AB+Cin(A$\oplus$B). As seen in Figure 3, XOR gates ‘inst’ and ‘inst1’ were used for the implementation of S. Cout is the result of the OR2 gate ‘inst4’ that combines (AB) from the AND2 gate ‘inst2’ and Cin(A$\oplus$B) from the AND2 gate ‘inst3’ that reuses the XOR gate ‘inst’.

In the bonus section of the lab, a clock input and the module ‘test_counter’ were used to generate all circuit inputs. This second design also implements a seven-segment BCD to show what minterm the program was currently testing. An error of communication caused segment 5 to be assigned to PIN_G22 instead of PIN_J22 which affected the output of the circuit. This didn’t hinder the understanding of the seven-segment BCD output. This tool is really convenient to avoid having to physically make all input combinations with the different switches and allows the circuit to do it independently.

Simulation and Verification of Implemented Design

After compiling the design without any error, a waveform was created and then compiled to simulate the outputs of the design in Figure 3. The simulated outputs, shown in Figure 5, were checked to ensure they matched truth table outputs, found in Table 1. Afterwards, pins were assigned to each input and output, the design was recompiled and uploaded to the Altera board. By flipping the switches linked with the inputs, all input combinations were tested to see if the output on the Altera board matched the truth table. Figures 6-8 show a couple of switch configurations and the respective outputs.

For the bonus circuit, each minterm number given by the seven-segment BCD was checked to ensure it matched the correct outputs which can be seen in Figure 10.1 and 10.2.

Discussion

In this lab, Quartus was used to build a full adder and was programmed to produce the outputs on an Altera board. Although many of the processes used to complete this task on Quartus were reviewed in previous courses, programming the outputs to display on the Altera board did include some new concepts.

Once the circuit diagram was completed, compiled, and simulated using the waveform editor, the pins needed assignments in order to produce an output on the Altera board. The pin assignment process was a skill learnt in this lab and some minor issues were encountered. It was learnt that in order to properly program the board, the circuit needed to be recompiled after pin assignment otherwise the board would not be properly programmed. This was a step that was forgotten before the first programming attempt which produced no output on the board. This issue was quickly resolved after carefully retracing the steps that were taken to program the board. It was also learnt after successfully programming the circuit, to pay close attention to where the inputs and outputs were assigned on the board. The order in which the inputs and outputs appear on the Altera board may not be in the same order from left to right as the truth table derived in preparation for the lab. This was important to consider to ensure that the testing of our board revealed accurate results and could be properly compared to the expected values from our truth table.

The bonus was also completed for this lab which incorporated a clock and a seven-segment BCD into the circuit designed in the main part of the lab. This allowed dynamic testing of the circuit and included the use of sequential logic. During our testing, segment 5 didn’t light up and after asking one of the TAs in the lab, it was assumed that this segment was broken. While reviewing the lab report, it was noticed that the wrong pin had been assigned to the output corresponding to segment 5. In the future, better communication should be used to avoid these types of mistakes and pin assignments should be double checked before compiling. This showed that most of the time the material is working properly and the program should be checked before assuming that something is broken. Overall, this bonus circuit provided a basic review of sequential circuit logic which will be used in the next lab.

In conclusion, the main circuit design was successful and encountered no major problems during the experiment. The minor errors that did occur were quickly solved and had no effect on the overall outcome of the circuit. The knowledge applied to this lab provided a good review of basic logic theories learnt in other courses while also providing the necessary skills for Quartus and the Altera board in preparation for future labs.

Appendix

Table 1: Truth table derived from the given circuit diagram outlining the expected results of the circuit

Inputs			**Outputs**	
A	**B**	**Cin**	**S**	**Cout**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

B Cin A	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Figure 1: Karnaugh map for output S derived from Table 1

$S = AB'Cin' + A'B'Cin + ABCin + A'BCin'$

(Distributive Law)

$S = Cin'(AB'+A'B) + Cin(A'B'+AB)$

(De Morgan's Law)

$S = Cin'(AB'+A'B) + Cin((A+B)(A'+B'))'$

(Distributive Law)

$S = Cin'(AB'+A'B) + Cin(AA'+AB'+BA'+BB')'$

(Complement)

$S = Cin'(AB'+A'B) + Cin(0+AB'+BA'+0)'$

(Identical element)

$S = Cin'(AB'+A'B) + Cin(AB'+BA')'$

(Xor property)

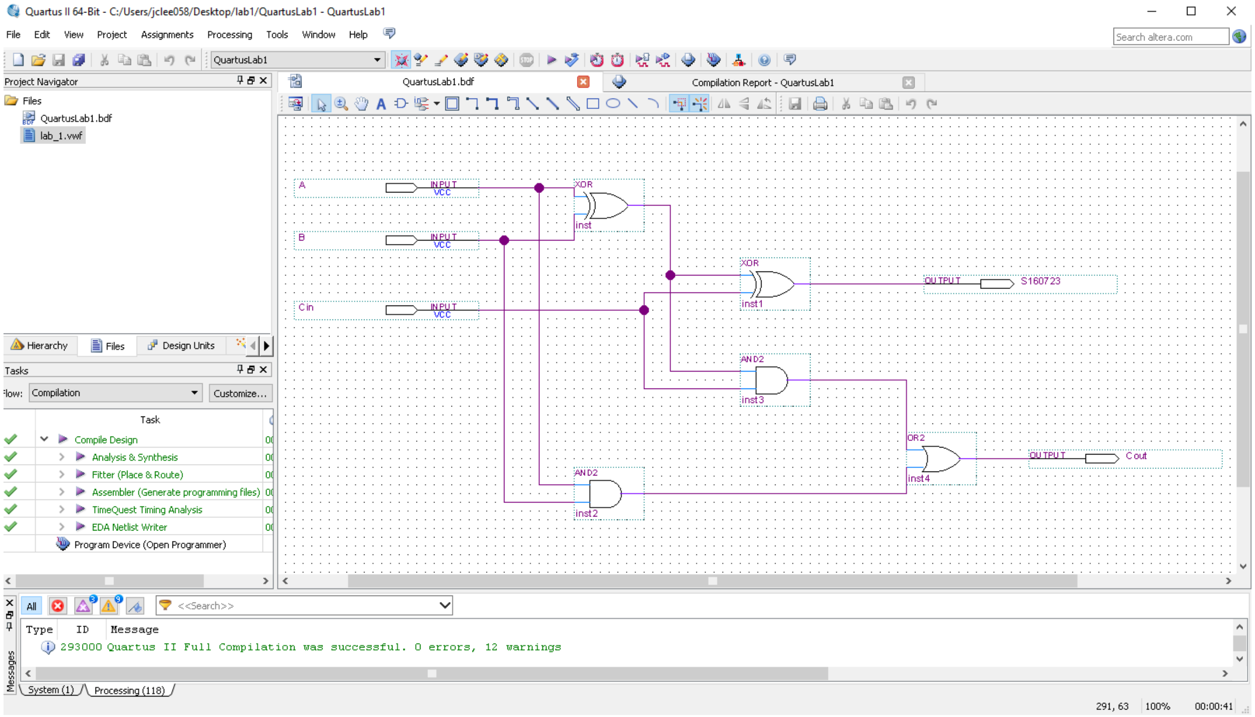
$S = Cin'(A \oplus B) + Cin(A \oplus B)'$

$S = Cin \oplus (A \oplus B)$

B Cin A	00	01	11	10
0	0	0	1	0
1	0	1	1	1

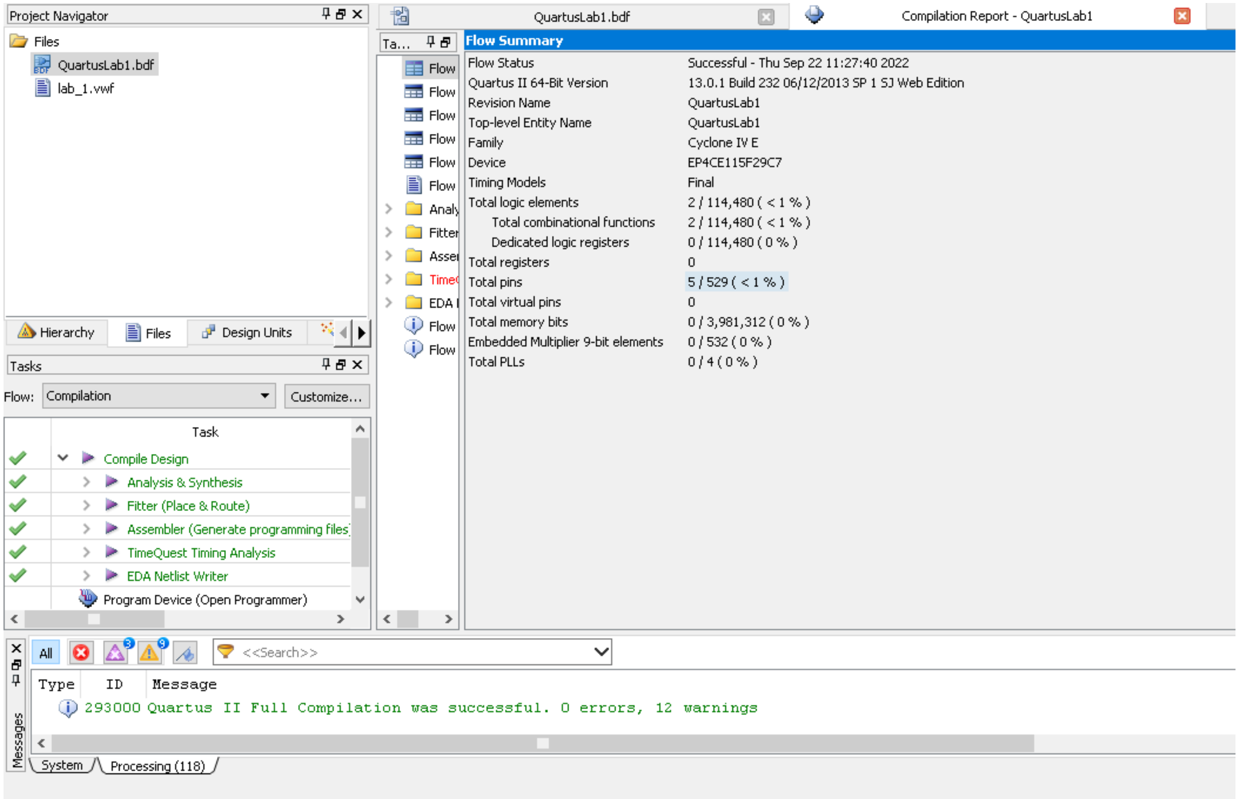
Figure 2: Karnaugh map for output Cout derived from Table 1

Cout = AB+AB’Cin+A’BCin
(Distributive Law)
Cout = AB+Cin(AB’+A’B)
(Xor property)
Cout = AB+Cin(A⊕B)



```
Type  ID    Message
      293000 Quartus II Full Compilation was successful. 0 errors, 12 warnings
```

Figure 3: Circuit diagram representing a full-adder



Flow: Compilation

Customize...

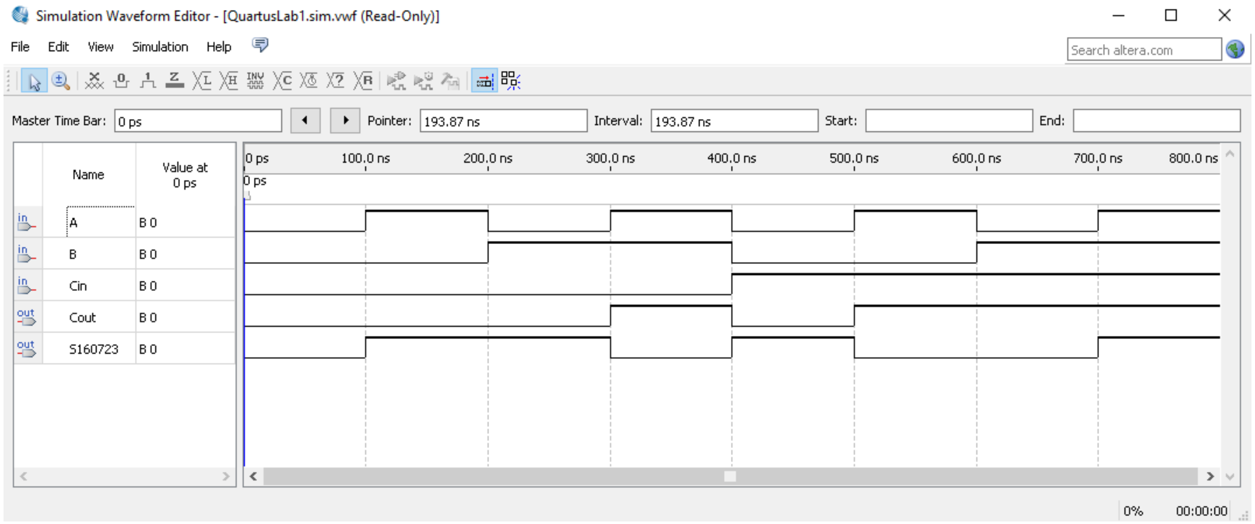
Task
✓ Compile Design
✓ Analysis & Synthesis
✓ Fitter (Place & Route)
✓ Assembler (Generate programming files)
✓ TimeQuest Timing Analysis
✓ EDA Netlist Writer
Program Device (Open Programmer)

Flow Summary

Flow Status	Successful - Thu Sep 22 11:27:40 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	QuartusLab1
Top-level Entity Name	QuartusLab1
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	2 / 114,480 (< 1 %)
Total combinational functions	2 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)
Total registers	0
Total pins	5 / 529 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

```
Type   ID     Message
   ⓘ 293000 Quartus II Full Compilation was successful. 0 errors, 12 warnings
```

Figure 4: Compilation result of the circuit shown in Figure 3



Master Time Bar: 0 ps

Pointer: 193.87 ns

Interval: 193.87 ns

Start:

End:

	Name	Value at 0 ps
in	A	B 0
in	B	B 0
in	Cin	B 0
out	Cout	B 0
out	S160723	B 0

Figure 5: Simulation results of the waveform representing the circuit in Figure 3

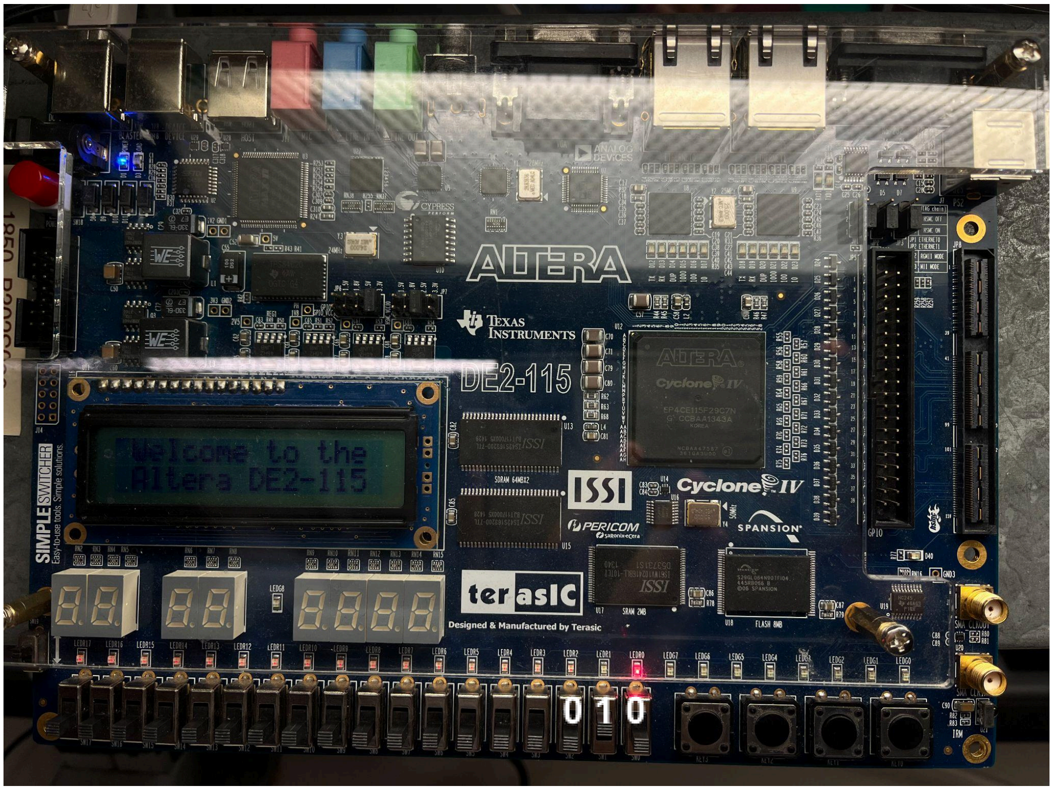


Figure 6: Experimental results of the circuit in Figure 3 (A=0, B=1, Cin=0, S=1, Cout=0)

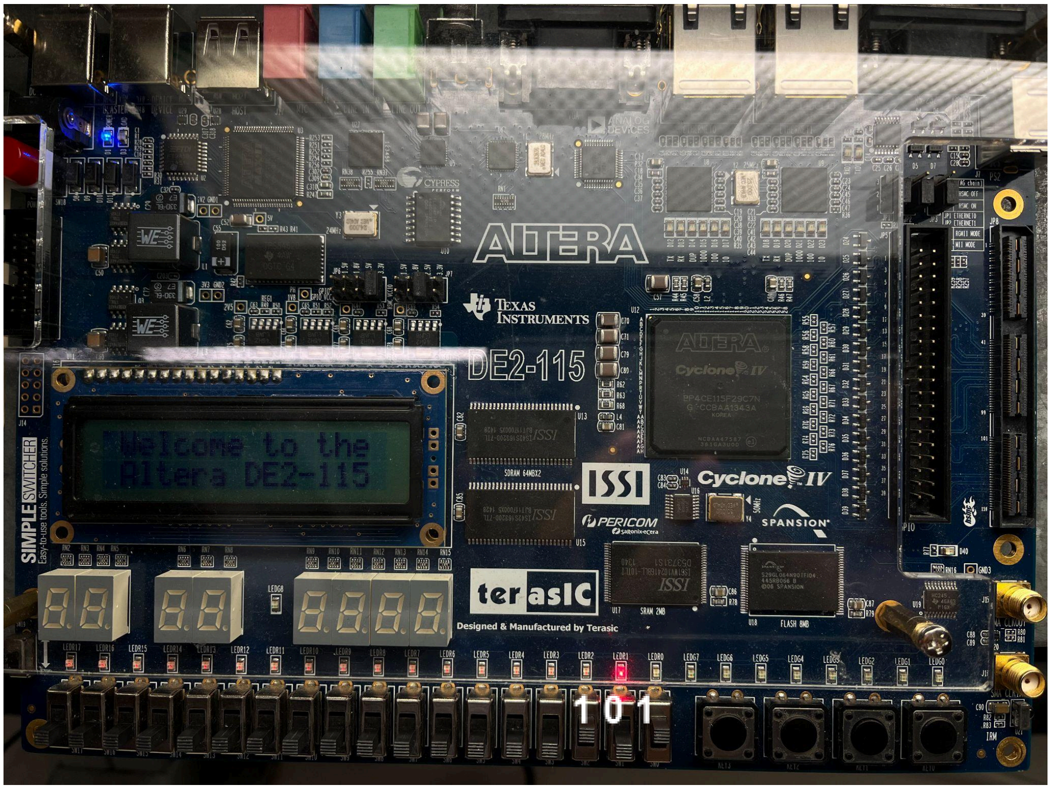


Figure 7: Experimental results of the circuit in Figure 3 (A=1, B=0, Cin=1, S=0, Cout=1)

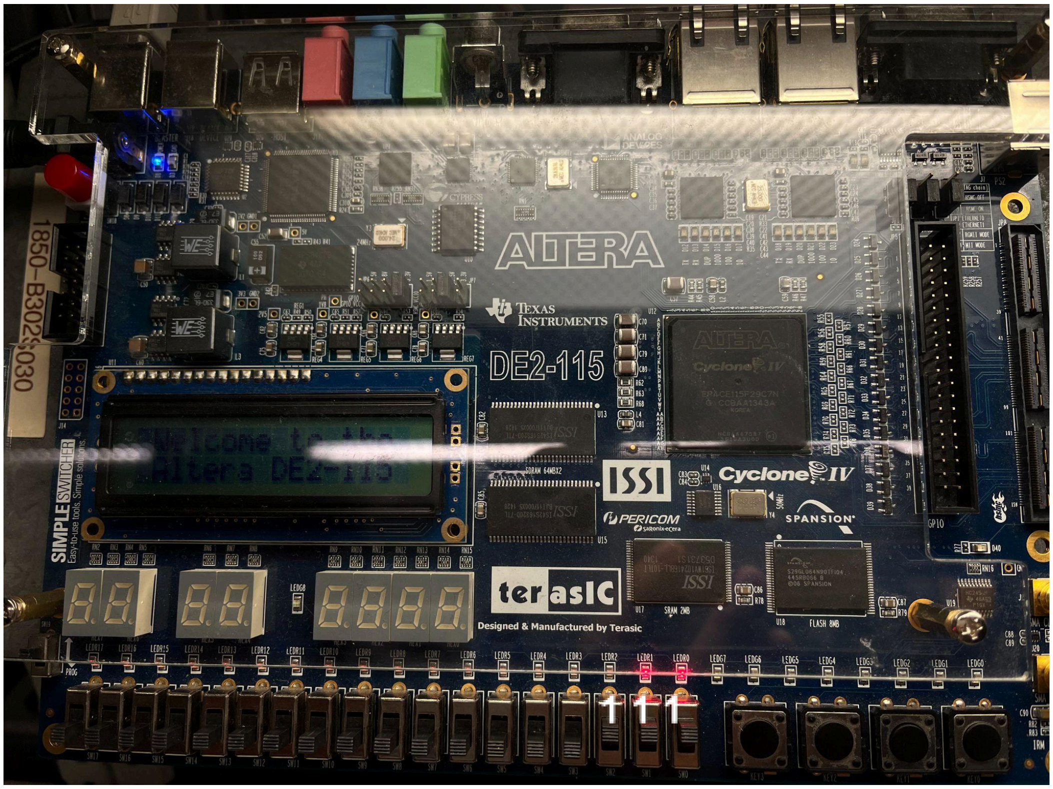


Figure 8: Experimental results of the circuit in Figure 3 (A=1, B=1, Cin=1, S=1, Cout=1)

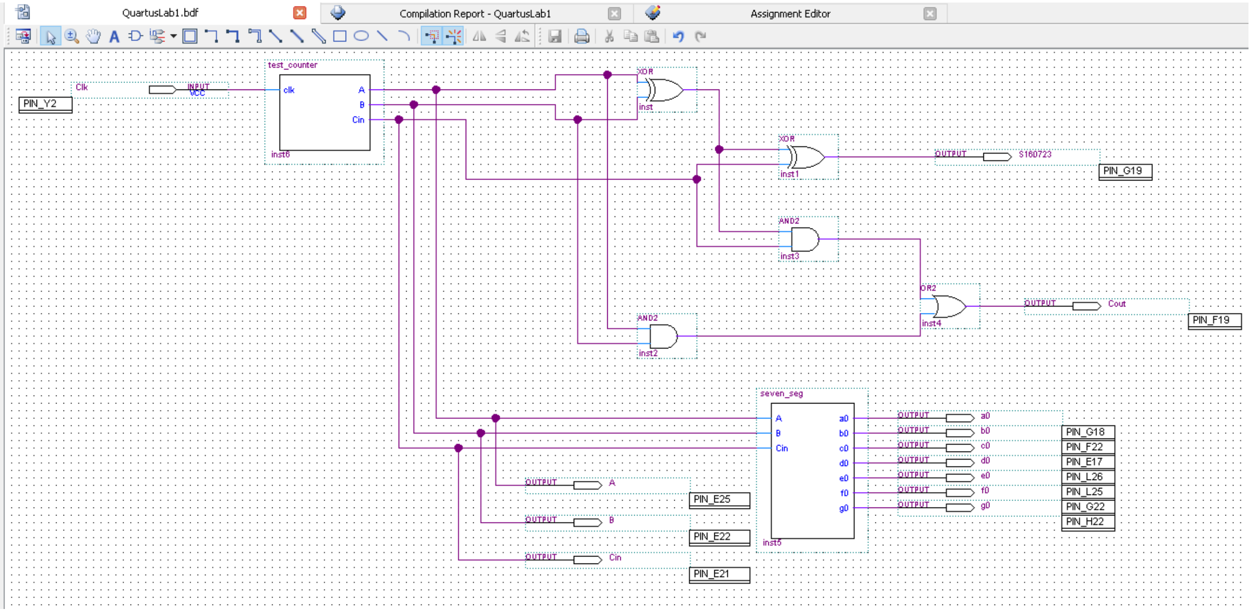


Figure 9: Bonus circuit diagram representing a full adder with a seven-segment BCD

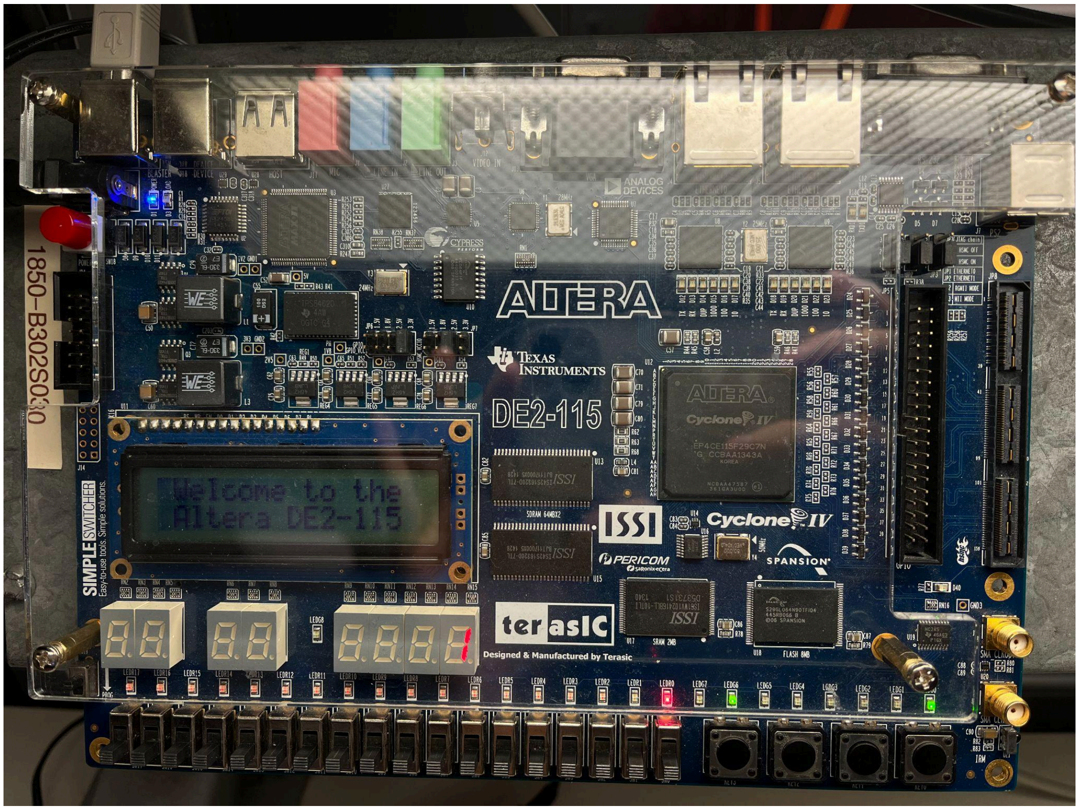


Figure 10.1: Experimental results of the circuit in Figure 9

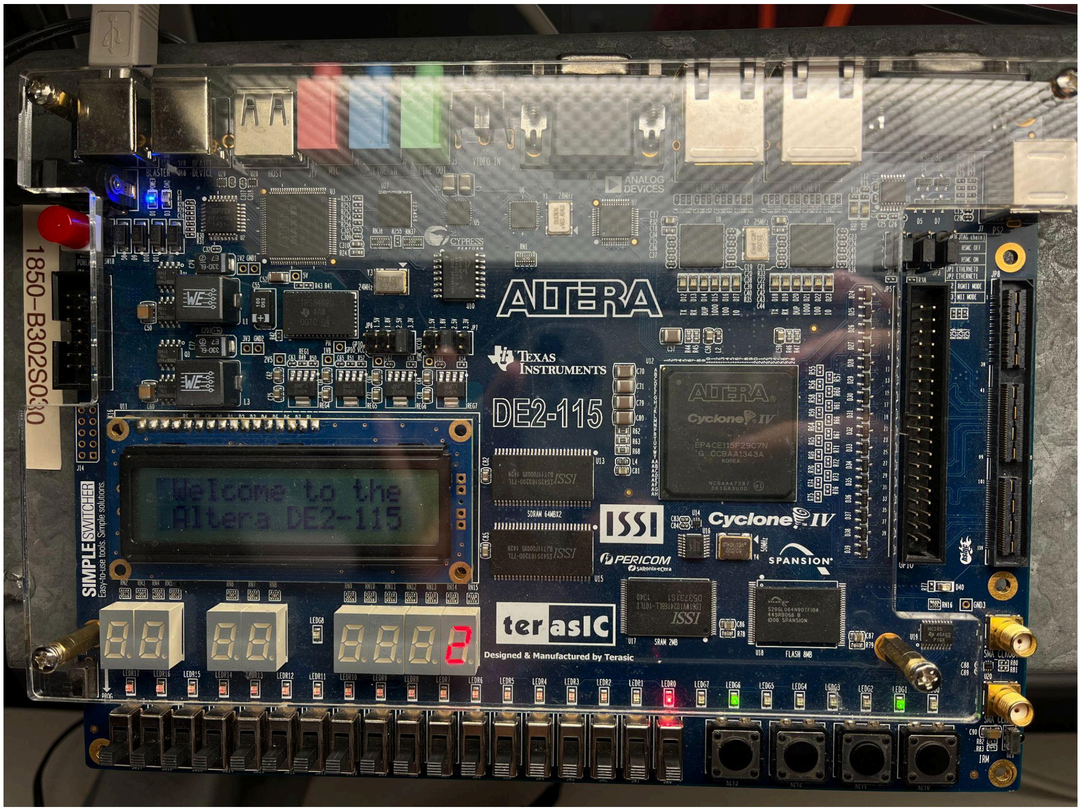
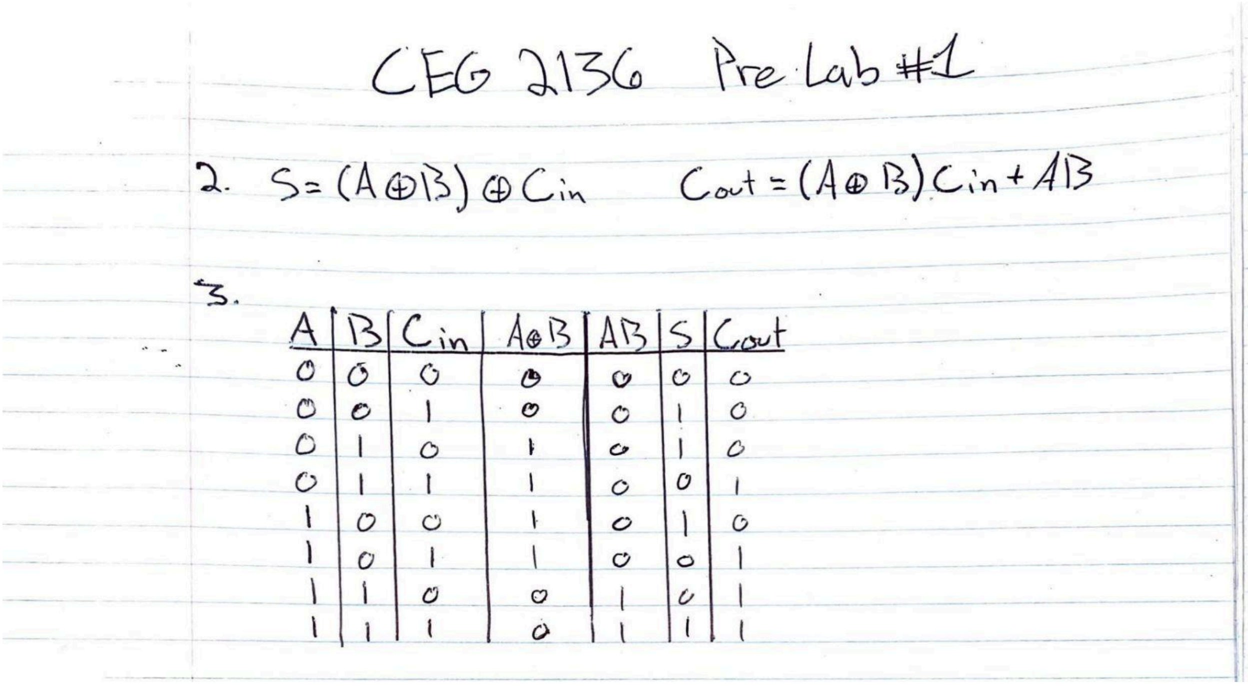


Figure 10.2: Experimental results of the circuit in Figure 9



CEG 2136 Pre Lab #1

2. $S=(A\oplus B)\oplus C_{in}$ $C_{out}=(A\oplus B)C_{in}+AB$

3.

A	B	C_{in}	$A\oplus B$	AB	S	C_{out}
0	0	0	0	0	0	0
0	0	1	0	0	1	0
0	1	0	1	0	1	0
0	1	1	1	0	0	1
1	0	0	1	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	1
1	1	1	0	1	1	1

Figure 11: Pre-lab