Lab 1 – Logic Gates

ITI1100 – Digital Systems

Winter 2017

Objectives

- Construct simple combinational logic circuits from a schematic.
- Experimentally determine the functional operation of simple combinational logic circuits.
- Identify equivalent logic gates to those produced by various circuit configurations from the resulting truth table.
- Connect various gates together to create simple logic functions.
- Analyse combinational logic circuits and predict their operation.
- Construct and test more complex combinational logic circuits.

Equipment & Components

- Quartus II 13.0 Service-Pack I
- Altera DE2-115 Card

Procedure

Refer to the "ITI1100 Laboratory Experiments" lab manual

Circuit Diagrams

One-Chip Logic Circuit (5.1.1)

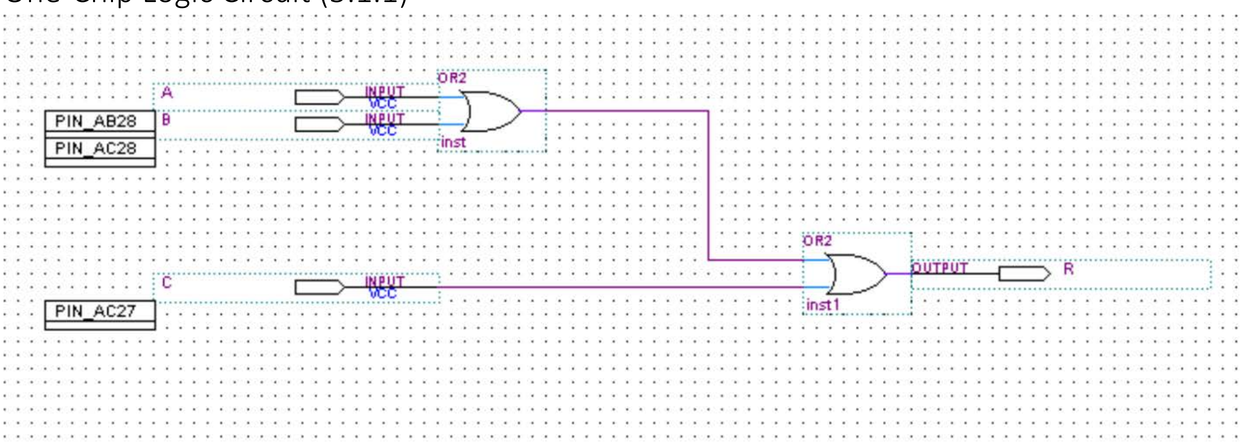

Figure 1: Screenshot of One-Chip Logic Circuit composed of two OR gates

Exclusive OR Circuit (5.1.5)

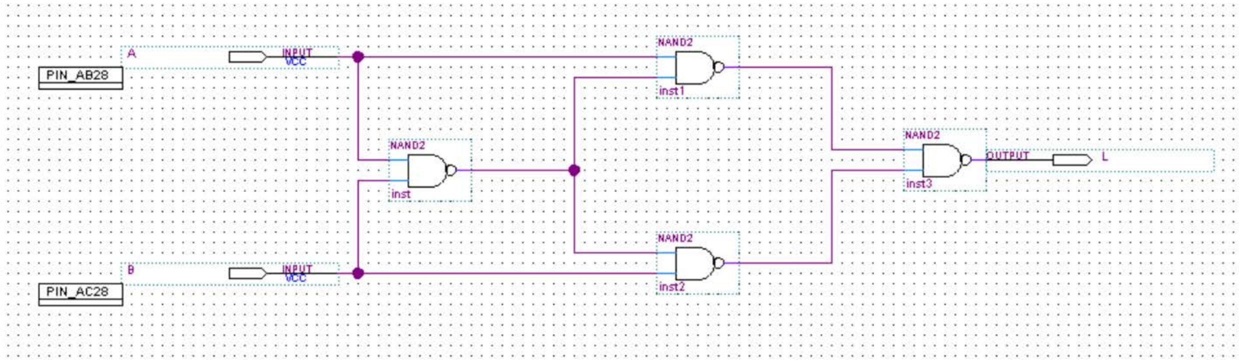

Figure 2: Screenshot of Exclusive OR Circuit composed of four NAND gates

Multiple Output Circuit (5.1.8)

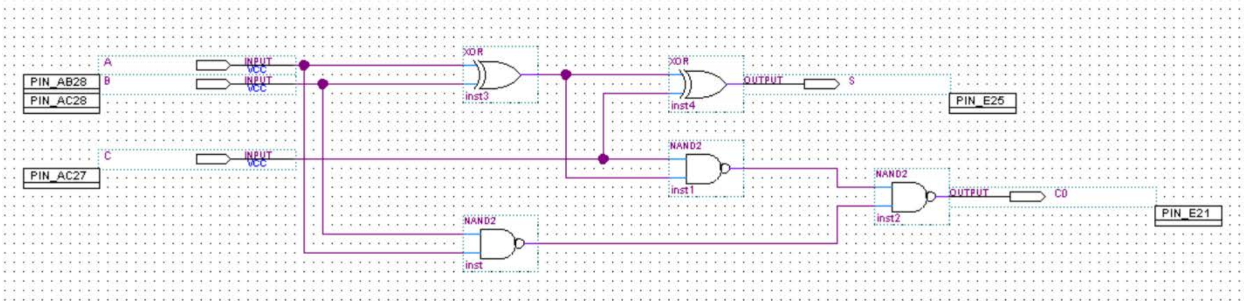

Figure 3: Screenshot of Multiple Output Circuit composed of three NAND gates and two XOR gates

Experiment Data & Data Processing

One-Chip Logic Circuit (5.1.1)

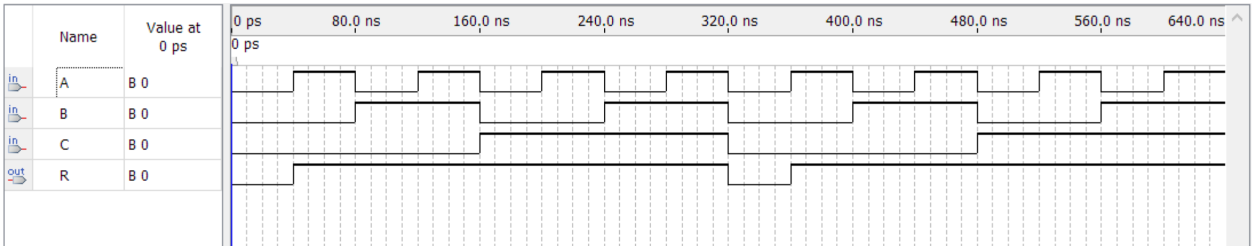

Figure 4: Waveform Simulation for One-Chip Logic Circuit

Table 1: Data Observed from the Altera DE2-115 board for One-Chip Logic Circuit (5.1.1)

Input			Output
A	**B**	**C**	**R**
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Logic Expression: (A + B) + C = R

Exclusive OR Circuit (5.1.5)

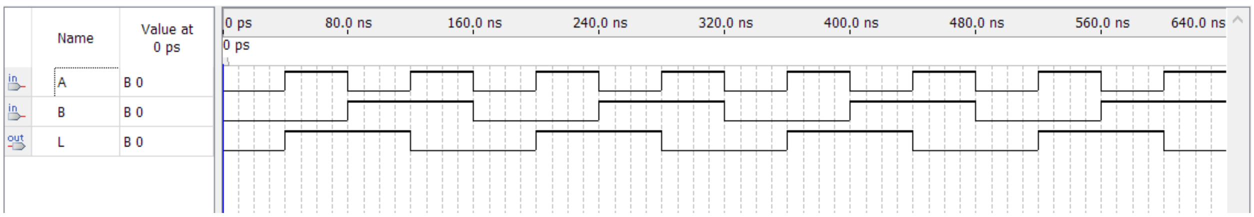

Figure 5: Waveform Simulation for Exclusive OR Circuit

Table 2: Data Observed from the Altera DE2-115 board for Exclusive OR Circuit (5.1.5)

Input		Output
A	**B**	**L**
0	0	0
0	1	1
1	0	1
1	1	0

Logic Expression: (((A·B)′ · A)′ · ((A·B)′ · B)′)′ = L

Multiple Output Circuit (5.1.8)

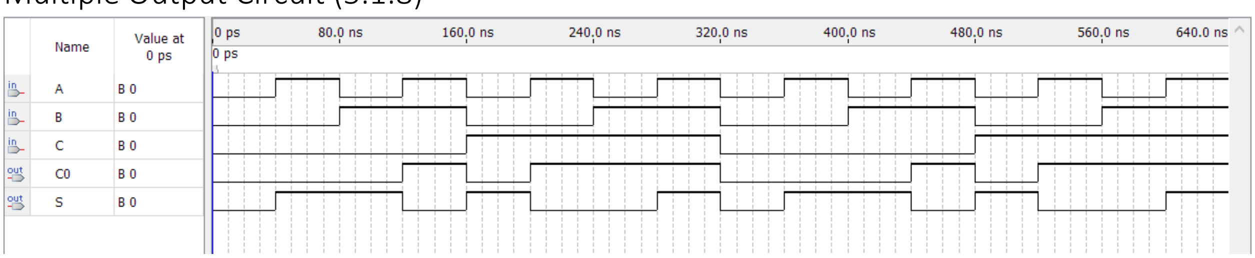

Figure 6: Waveform Simulation for Multiple Output Circuit

Table 3: Data Observed from the Altera DE2-115 board for Multiple Output Circuit (5.1.8)

Input			Output	
A	**B**	**C**	**C_0**	**S**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Logic Expressions: $((A\cdot B)'\cdot((A\oplus B)\cdot C)')' = C_0$

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

Comparison of Theoretical & Experimental Data

Table 4: Comparison of Theoretical & Experimental Results for One-Chip Logic Circuit (5.1.1)

Input			Expected Results	Actual Results
A	**B**	**C**	**R**	**R**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

The theoretical results matched the experimental data obtained from the Altera DE2-115 card.

Table 5: Comparison of Theoretical & Experimental Results for Exclusive OR Circuit (5.1.5)

Input		Expected Results	Actual Results
A	**B**	**L**	**L**
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

The theoretical results matched the experimental data obtained from the Altera DE2-115 card.

Table 6: Comparison of Theoretical & Experimental Results for Multiple Output Circuit (5.1.8)

Input			Expected Results		Actual Results	
A	**B**	**C**	**C_0**	**S**	**C_0**	**S**
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	0
1	1	1	0	1	1	1

The theoretical results did not match the experimental data obtained from the Altera DE2-115 card. From the experimental results, it was determined that output C_0 would be on while the inputs A and B are on.

Discussion

The purpose of this lab was to form Boolean logic expressions from circuit schematics and verify their output using the Altera DE2-115 card.

The experiment required the use of Quartus II 13.0 to design, compile, simulate, and test multiple combinational logic circuits. Truth tables were created for each circuit during testing in order to get the practical results of each circuit. This data was then compared to our prelab truth tables. If our results matched, then we had correctly defined our logic expression for that circuit. However, if the results had discrepancies then either the truth table or logic expression was computed incorrectly.

Part I

All the logic circuits designed in part I of the lab returned the same results as we expected. By comparing our theoretical and experimental data, we concluded that the One-Chip logic circuit (5.1.1) was equivalent to an OR gate, the Two-Chip logic circuit (5.1.2) was equivalent to a NAND gate, and the Three-Chip logic (5.1.3) circuit was equivalent to an XNOR gate.

Part II

While 2 out of the 3 circuits designed in part II matched the theoretical data, we found deviations when testing our Multiple Output circuit (5.1.8). As highlighted in Table 6, our initial expectation was that the C_0 output would be inactive for the last two input combinations. However, when testing the circuit, we found that we got the opposite result and output C_0 was active during the last two input combinations. This was not a major deviation in results as the previous input combinations were predicted correctly. The reason for this discrepancy was a calculation error while creating the prelab truth table. After validating the logic expressions once more we realized that source of error must have been from an inaccurate computation.

Conclusion

In conclusion, we achieved the objectives laid out at the beginning of the lab and learned how to construct circuits, predict their output, and analyse data. We also learned how to use the Quartus II software as well as the Altera DE2-115 card to perform these tests. While most of our experimental results matched our theoretical results, we did have some errors in our last circuit due to invalid computation.

5.1.1: One-Chip Logic Circuit

Logic Expression: (A + B) + C = R

Truth Table

A	B	C	A + B	R
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

5.1.2: Two-Chip Logic Circuit

Logic Expression: ((A·B)·(C·D))′ = U

Truth Table

A	B	C	D	A·B	C·D	U
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	0

5.1.3: Three-Chip Logic Circuit

Logic Expression: (A·B) + (A′·B′) = K

Truth Table

A	B	A·B	A′·B′	K
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

5.1.5: Exclusive OR Circuit

Logic Expression: (((A·B)′ · A)′ · ((A·B)′ · B)′)′ = L

Truth Table

A	B	(A·B)′	(A·B)′ · A	(A·B)′ · B	L
1	1	0	0	0	0
1	0	1	1	0	1
0	1	1	0	1	1
0	0	1	0	0	0

5.1.7: OR Circuit

Logic Expression: ((A·B) + (A′·C)) + (B′·(A + C)) = P

Truth Table

A	B	C	A·B	A′·C	A + C	P
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	1	1	1
1	0	0	0	0	1	1
1	0	1	0	0	1	1
1	1	0	1	0	1	1
1	1	1	1	0	1	1

5.1.6: AND Circuit

Logic Expression: (C·A) + (C·A′) · ((D·B)′·(D·B′)′)′ = V

Truth Table

A	B	C	D	C·A	C·A′	D·B	D·B′	V
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0
0	0	1	0	0	1	0	0	0
0	0	1	1	0	1	0	1	0
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0
0	1	1	1	0	1	1	0	0
1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	1	0
1	1	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0
1	1	1	0	1	0	0	0	0
1	1	1	1	1	0	1	0	1

5.1.8: Multiple Output Circuit

Logic Expression: $((A \cdot B)' \cdot ((A \oplus B) \cdot C)')' = C_0$

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

Truth Table

A	B	C	$(A \cdot B)'$	$A \oplus B$	$(A \oplus B) \cdot C$	C_0	S
0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	1	1	0	0	1
0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	1
1	0	1	1	1	1	1	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	1