Lab 2: BOOLEAN LOGIC

ITI 1100A- Digital Systems
Winter 2024

School of Electrical Engineering and Computer Science
University of Ottawa

Course Coordinator: Dr. Hussein T.Mouftah
Teaching Assistants: Siddhant Tiwari
Emilia Zielinska

Group 29 :
Layth Eleissawi 300330109
Sameed Ahmed 300366534

Experiment Date:02/02/2024
Submission Date:12/02/2024

Objectives

- Simplify logic functions into simpler expressions or truth tables to understand how they behave.
- Creating circuits that perform specific tasks based on the input while using basic logic gates like AND , OR , and NOT gates.
- Trying to simplify logic circuits while maintaining its functionality.
- Implementing NAND gates to create complex circuits while simplifying the design process .
- Testing circuits while providing multiple outputs and verifying if it matches our expectations.

Equipment

- Quartus sp13.0 64bit
- Altera DE2-115 Cyclone IV E EP4CE115F28C7

Part I – Combinational Logic Circuits Minimization by Boolean Algebra

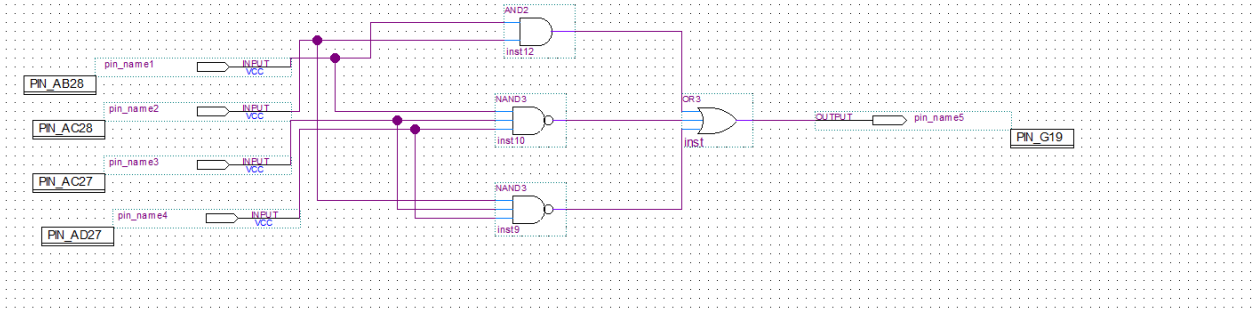


Figure 1: Combinational Circuit comprised of 3 NAND gates and 1 OR gate

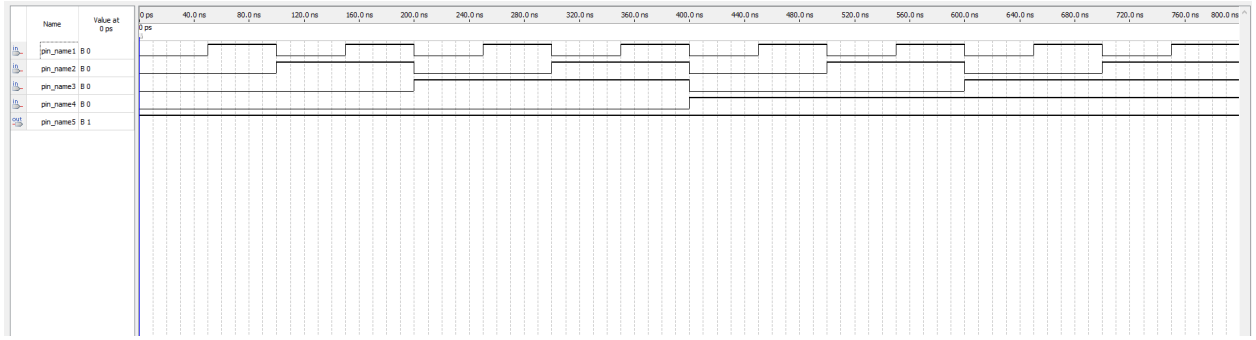


Figure 2: Simulation diagram of a circuit in Figure 1.

	Input			Output	
A	B	C	D	Expected	Actual
0	0	0	0	1	1
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Table 1: Data observed for combinational logic circuit from figure 1

Part II – Combinational Logic Circuits Minimization by the Karnaugh Map Method

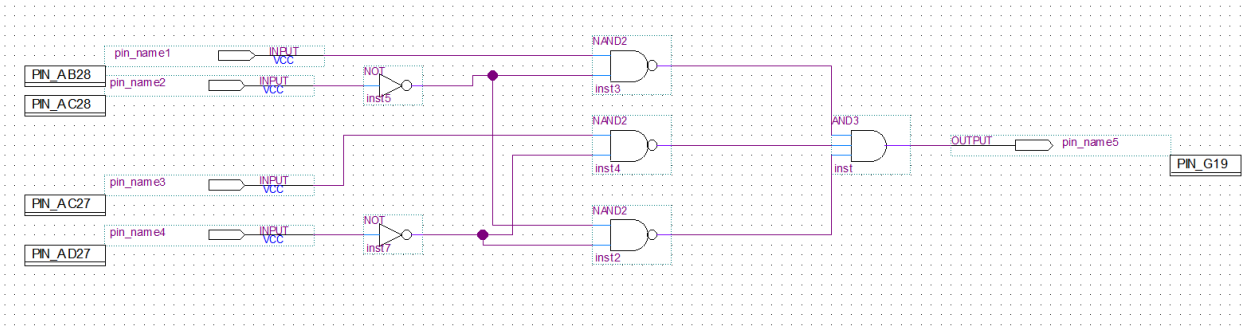


Figure 3: Combinational Circuit comprised of 3 NAND gates, 2 NOT gates and one AND gate

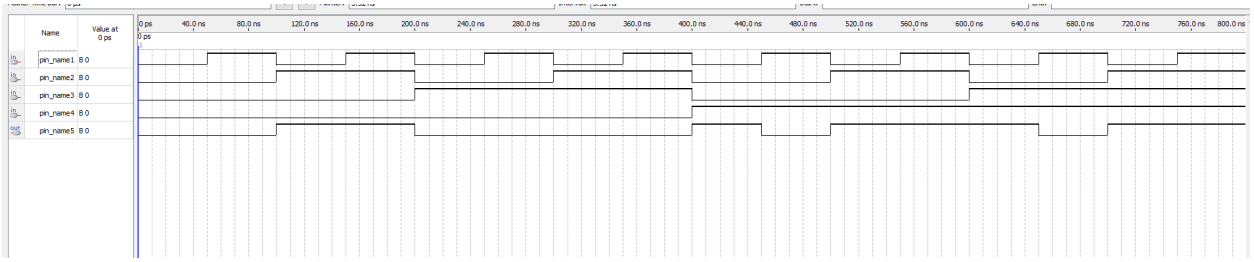


Figure 4: Simulation diagram of a circuit in Figure 3

	Input			Output	
A	B	C	D	Expected	Actual
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	1
1	1	1	1	1	1

Table 2: Data observed for combinational logic circuit from figure 3

Part III – Design of Combinational Logic Circuits

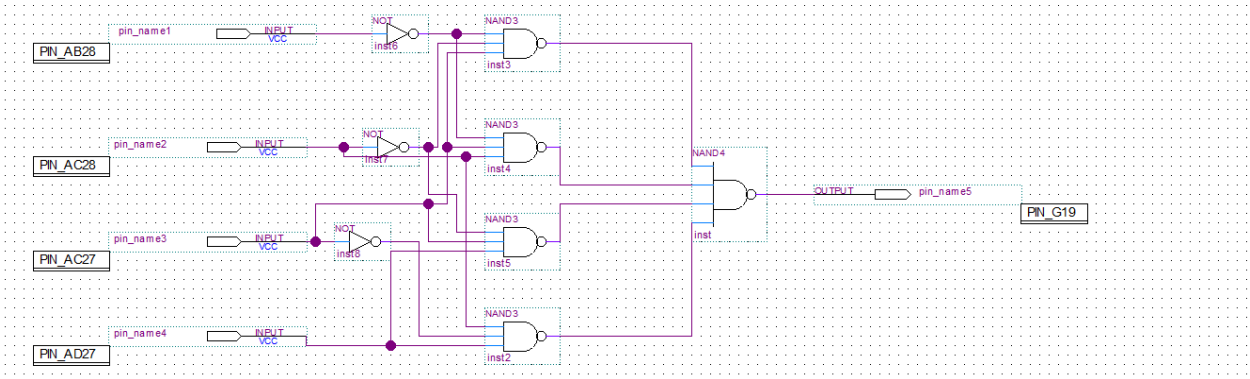


Figure 5: Combinational Circuit comprised of 5 NAND gates and 3 NOT gates

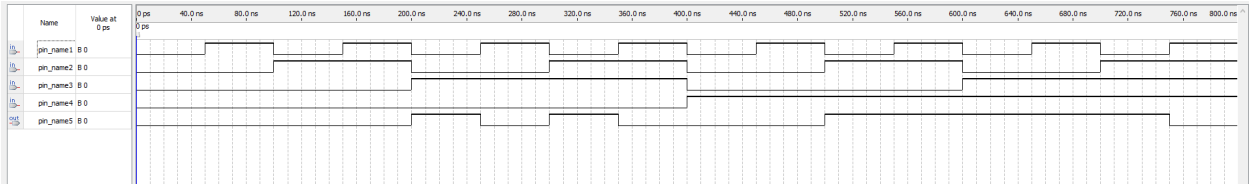


Figure 6: Simulation diagram of a circuit in Figure 5

	Input			Output	
D3	D2	D1	D0	Expected	Actual
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	0	0

Table 3: Data observed for combinational logic circuit from figure 5

Discussion and conclusion

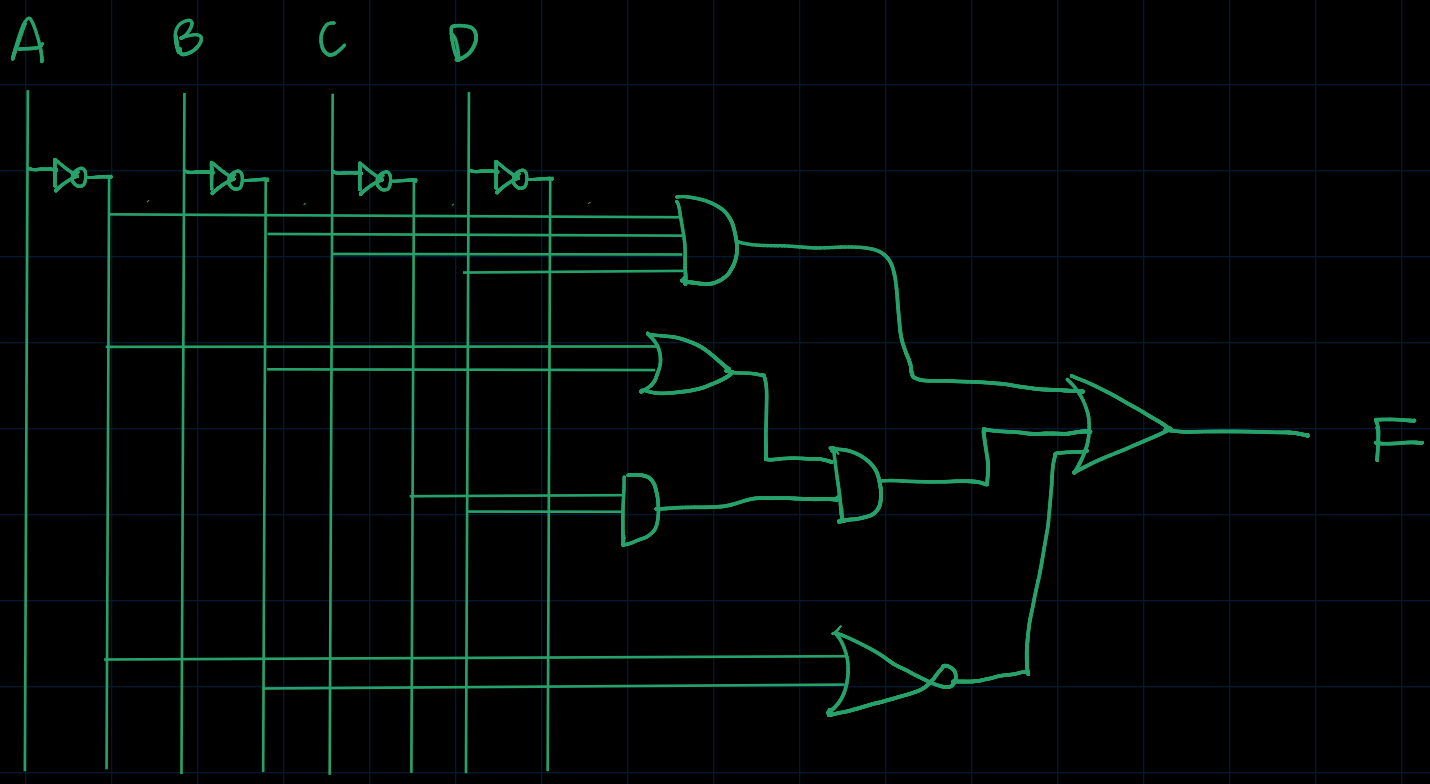
Discussion and Conclusion

The objective of the lab was to build combinational circuits to help understand boolean logic and how certain logic functions can be made exclusively with NOT and NAND gates, as a cost saving measure and simplification. Combinational Logic Circuits can also be simplified, as attempted through boolean algebra and using a karnaugh map, and essentially yielding the same result as was intended by the original expression. The experiment was performed to a high degree of success, with no known errors. After running tests on our circuits and inputting data, we checked to see if our results matched what we expected. Thankfully, all our results turned out to be accurate, just as we predicted.. However, we did encounter a challenge along the way. Ensuring the precision of our circuit designs was a bit challenging. Even small errors in connecting gates or misinterpreting truth tables could throw off our results. We had to be really careful and run multiple simulations to catch and fix any mistakes.

In conclusion, the lab aimed to build combinational circuits to understand Boolean logic, focusing on using NOT and NAND gates for cost-saving and simplification. We successfully experimented with simplifying logic circuits using Boolean algebra and Karnaugh maps, achieving accurate results. However, ensuring precise circuit designs posed a challenge, requiring careful testing and attention to detail.

$$Y = \overline{A\bar{B}CD} + (\bar{A}+\bar{B})(\overline{CD}) + \overline{(\bar{A}+\bar{B})}$$

a)



b) $Y = A'B'CD + (A'+B')(C'D) + (A'+B')'$

$= A'B'CD + (A'+B')(C'D) + AB$ (De Morgans)

$= A'B'CD + A'C'D' + B'C'D' + AB$

$= AB + A'C'D' + B'C'D'$

Truth Table		A'B'CD + (A' + B') (C'D') + (A'B')'						
	A	B	C	D	A'B'CD	(A'+B')(C'D')	(A'B')'	R
	0	0	0	0	0	1	1	1
	0	0	0	1	0	0	0	0
	0	0	1	0	0	0	0	0
	0	0	1	1	1	0	0	0
	0	1	0	0	0	1	0	1
	0	1	0	1	0	0	1	1
	0	1	1	0	0	0	1	1
	0	1	1	1	0	0	1	1
	1	0	0	0	0	1	1	1
	1	0	0	1	0	0	1	1
	1	0	1	0	0	0	1	1
	1	0	1	1	0	0	1	1
	1	1	0	0	0	0	1	1
	1	1	0	1	0	0	1	1
	1	1	1	0	0	0	1	1
	1	1	1	1	0	0	1	1

Pt 2

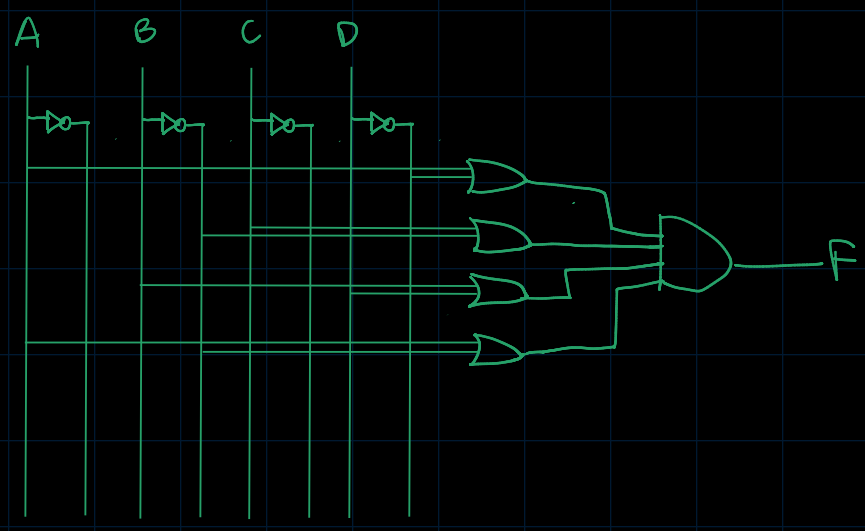
$$A'B'C'D' + A'B'CD' + AB'C'D' + A\bar{B}'C'D + AB'CD' + AB'CD + ABCD'$$

	CD	00	01	11	10
AB					
00		1	0	0	1
01		0	0	0	0
11		0	0	0	1
10		1	1	1	1

$$F = AB' + ACD' + B'D'$$
$$= (AB' + ACD' + B'D')'$$
$$= (AB')'(ACD')'(B'D')'$$

POS

$$(A+D')(C+B')(B+D)(A+B')$$



A	B	C	D	(A+D')	(C+B')	(B+D)	(A+B')	R
0	0	0	0	1	1	0	1	0
0	0	0	1	0	1	1	1	0
0	0	1	0	1	1	0	1	0
0	0	1	1	0	1	1	1	0
0	1	0	0	1	0	1	0	0
0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	1	0	0
0	1	1	1	0	1	1	0	0
1	0	0	0	1	1	0	1	0
1	0	0	1	1	1	1	1	1
1	0	1	0	1	1	0	1	0
1	0	1	1	1	1	1	1	1
1	1	0	0	1	0	1	1	0
1	1	0	1	1	0	1	1	0
1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1

	(D2.D1.D0) + (D3'.D1.D0) + (D3'.D2.D1) + (D2'.D1.D0)								
N	D3	D2	D1	D0	2.1'.0	3'.1.0	3'.2'.1	2'.1.0	P
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0
2	0	0	1	0	0	0	1	0	1
3	0	0	1	1	1	1	1	1	1
4	0	1	0	0	0	0	0	0	0
5	0	1	0	1	0	0	0	0	1
6	0	1	1	0	0	0	0	0	0
7	0	1	1	1	1	1	0	0	1
8	1	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	0
10	1	0	1	0	0	0	0	0	0
11	1	0	1	1	0	0	0	1	1
12	1	1	0	0	0	0	0	0	0
13	1	1	0	1	1	0	0	0	1
14	1	1	1	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0

$(D_3'D_2'D_1D_0') + (D_3'D_2'D_1D_0) + (D_3'D_2D_1'D_0) + (D_3'D_2D_1D_0) + (D_3D_2'D_1D_0) + (D_3D_2D_1'D_0)$

3

	D1 D0	00	01	11	10	
D3 D2						
00		0	0	1	1	
01		0	1	1	0	
11		0	1	0	0	
10		0	0	1	0	

$Y = (D_3'D_2'D_1) + (D_3'D_2D) + (D_2'D_1D_0') + (D_2D_1'D_0)$

$y = [(D_3'D_2'D_1)'(D_3'D_2D_1)'(D_2'D_1D_0)'(D_2D_1'D_0)']'$

$[(A'B'C)'(A'BC)'(B'CD)'(BC'D)']'$