North South University

Department of Electrical & Computer Engineering

LAB REPORT

Course Code: CSE231L.07

Course Title: Digital Logic Design

Faculty: DR. M ABDUR RAZZAK

Experiment Number: 01

Experiment Name:

Digital Logic Gates and Boolean Functions

Experiment Date: 19-10-2022

Date of Submission: 26-10-2022

Section: 07

Group Number: 02

Submitted To: MD. AL – AMIN BHUIYAN

Submitted By:

Score

Serial No	Student Name	ID
9	Al Zammee	2021648642
12	Umme Suraia Haque Setu	2031278642
18	Humayra Rahman Nipa	2121128642
24	Saif Mohammed	2121913642
35	Afra Saiara Saima	2132242642

Lab Report 1: Digital Logic Gates and Boolean Functions

Experiment Names

1. Introduction to Basic Logic Gates
2. Constructing 3-input AND & OR Gates for 2-input & OR Gates
3. Implementation of Boolean Functions

Objectives

- Study the basic logic gates - AND, OR, NOT, NAND, NOR, XOR.
- Get acquainted with the representation of Boolean functions using truth tables, logic diagrams and Boolean Algebra.
- Prove the extension of inputs of AND and OR gates using the associate law.
- Become familiarized with combinational logic circuits.

Theory

Logic Gates

Logic gates are the components which are used to design a logical circuit or digital circuit. These logic gates are used to perform logical operations from logical inputs to get a single logical output. Logic gates only consider two discrete values of voltage level to determine the inputs. Those values also can called binary inputs. The binary value 1 is to represent logical high or True and the value 0 is to represent the logical low or false. Logic gates are named as AND, OR, NOT, NAND, NOR, XOR to perform logical operations and design digital circuit.

Truth Table

A truth table is a binary input-output table, used to represent the corresponding outputs by a function using logic gates. It describes the relationship between the input and output of a logic circuit. For each and every possible combination of inputs, a truth table show all possible outputs.

Integrated Circuit (IC)

The basic rule for most ICs is that there is polarity mark, such as the half- moon notch shown in the figure. Another common polarity mark is a small dot, triangle or tab by pin 1. The rule is to move counter-clockwise around the chip from the polarity mark while numbering the pins starting at 1. Sometimes no direct mark may be present, in which case the pin numbers can be inferred

simply from the orientation of the text inscribed on the IC. Among these pins the 7^{th} pin is designed to connect the ground and the 14^{th} pin is designed to connect the +5V as Vcc.

7400 Series Integrated Circuits

TTL - Transistor-Transistor Logic

The 7400 series of digital logic ICs represents the most popular family of TTL ICs. Most such modern ICs have been replaced with CMOS. To find the IC number on the chip, simply read the numbers off it ignoring the letters. For example, 74HC04N is the 7404 Hex Inverter IC where the HC denotes it is a high-speed CMOS variant of the TTL circuit.

Boolean Algebra

Boolean algebra is a branch of mathematical logic that formalizes the relation between variables that take the truth values of *true* and *false*, denoted by 1 and 0 respectively. It is fundamental in the development of digital electronics. Digital electronics networks are generally expressed as Boolean functions. Discrete voltage levels are used to represent the truth values.

Combinational Logic

Combination logic refers to digital networks where the output is solely dependent on the current input(s) and is not affected by previous states. The analysis of combination logic requires writing the Boolean functions for each element of the circuit, producing their truth tables, and subsequently combining each function for the final output and truth table.

Apparatus List

- IC 7400 Quadruple 2-input NAND gates
- IC 7402 Quadruple 2-input NOR gates
- IC 7404 Hex Inverters (NOT gates)
- IC 7408 Quadruple 2-input AND gates
- IC 7432 Quadruple 2-input OR gates
- IC 7486 Quadruple 2-input XOR gates
- Trainer Board
- Wires

Circuit Diagram

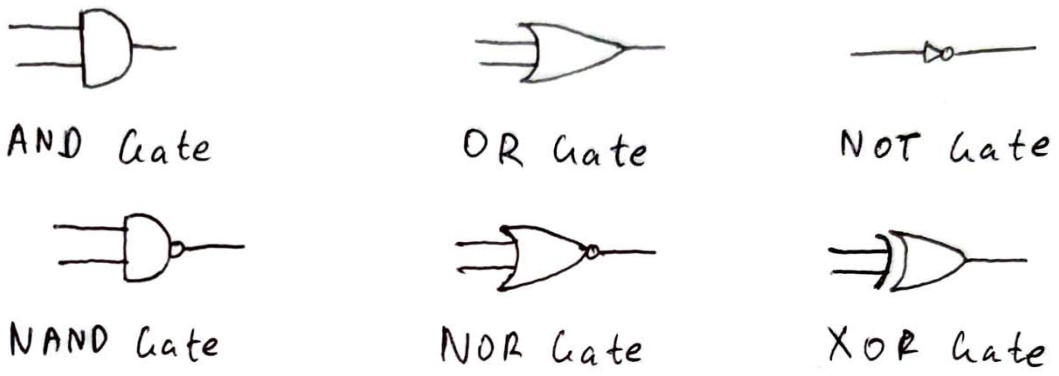


Figure 1.1: Pin Configuration of gates in ICs

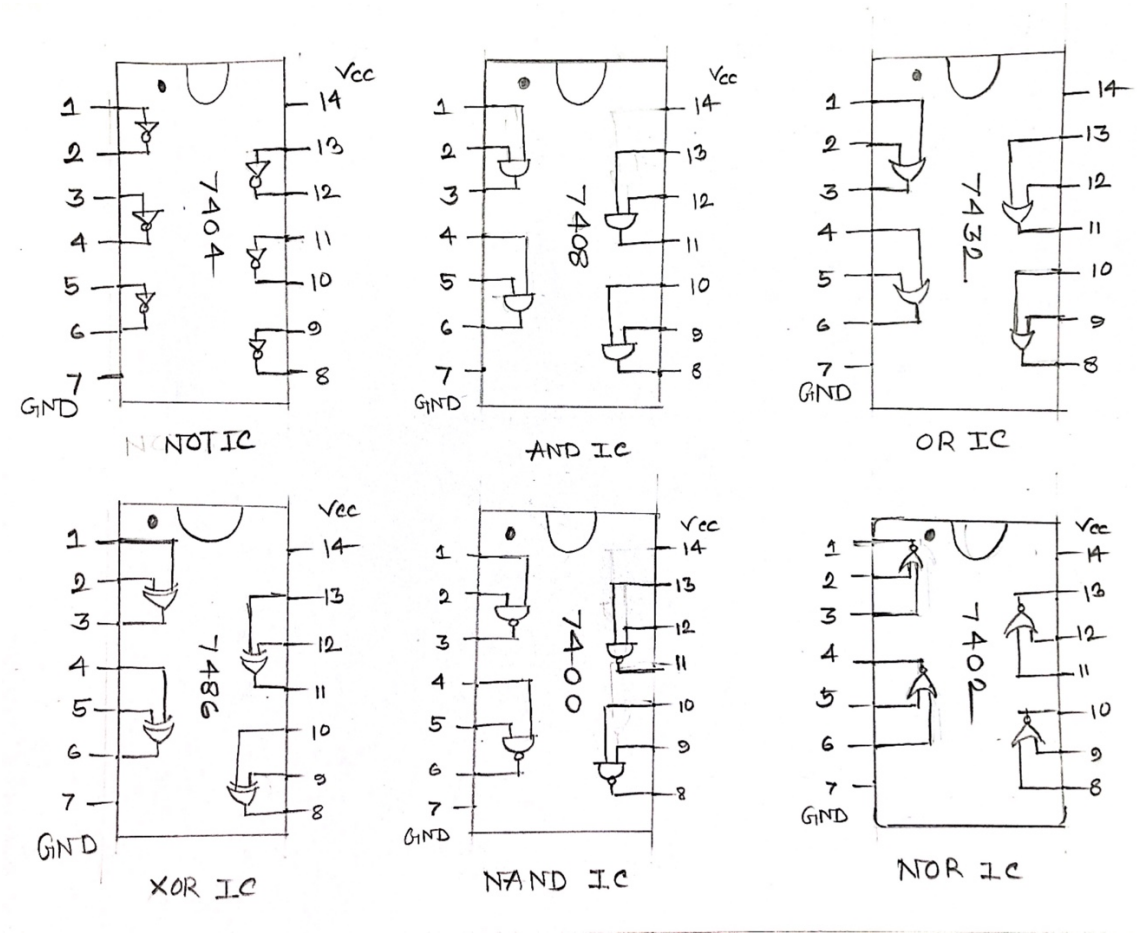


Figure 1.2: Pin Diagram of 7400 Series IC

Figure2.1: Extension of inputs AND and OR gates

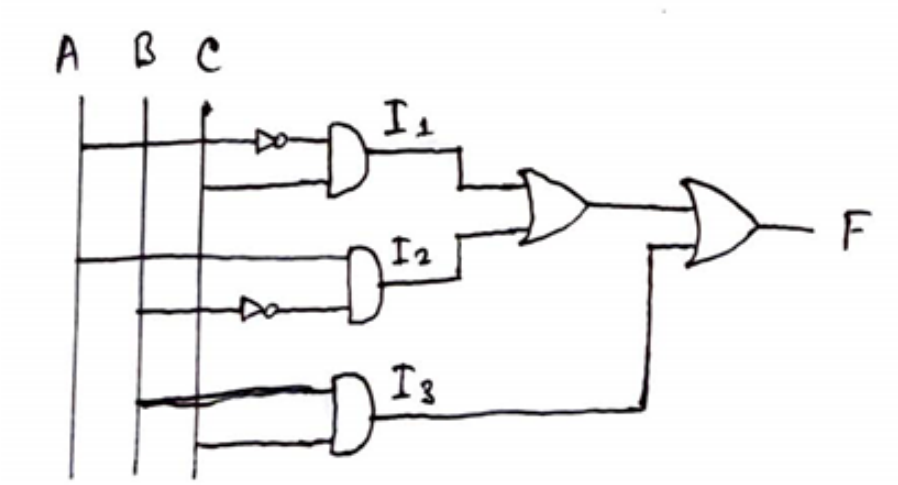
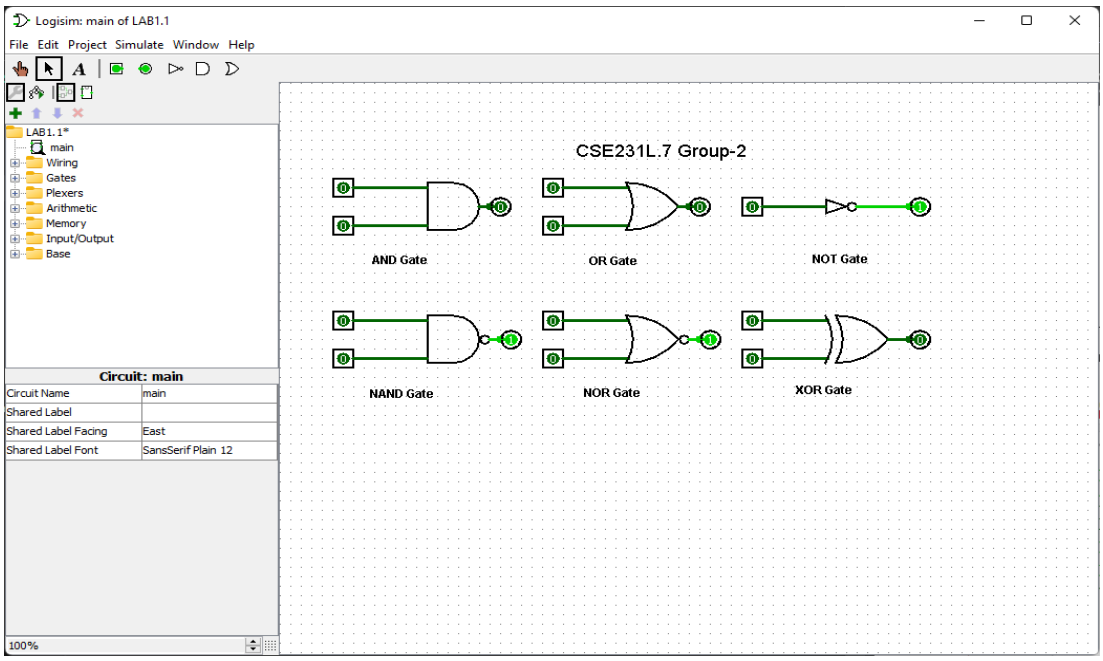
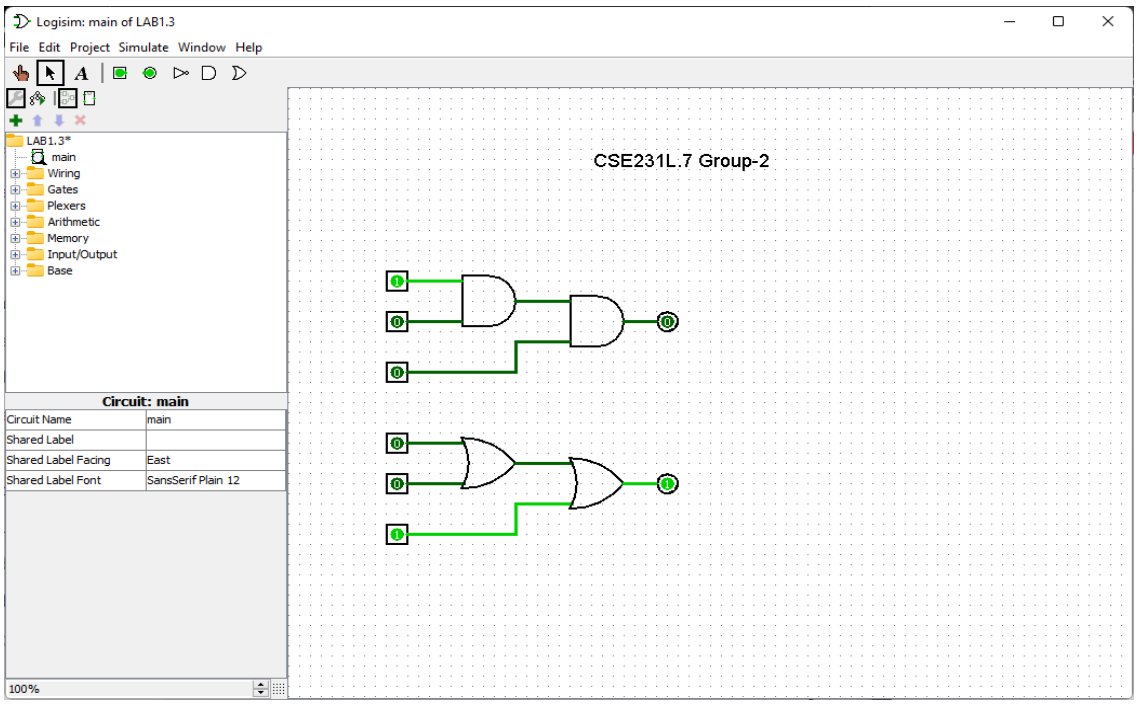


Figure 3.1: Logic Diagram for Driven Boolean Function

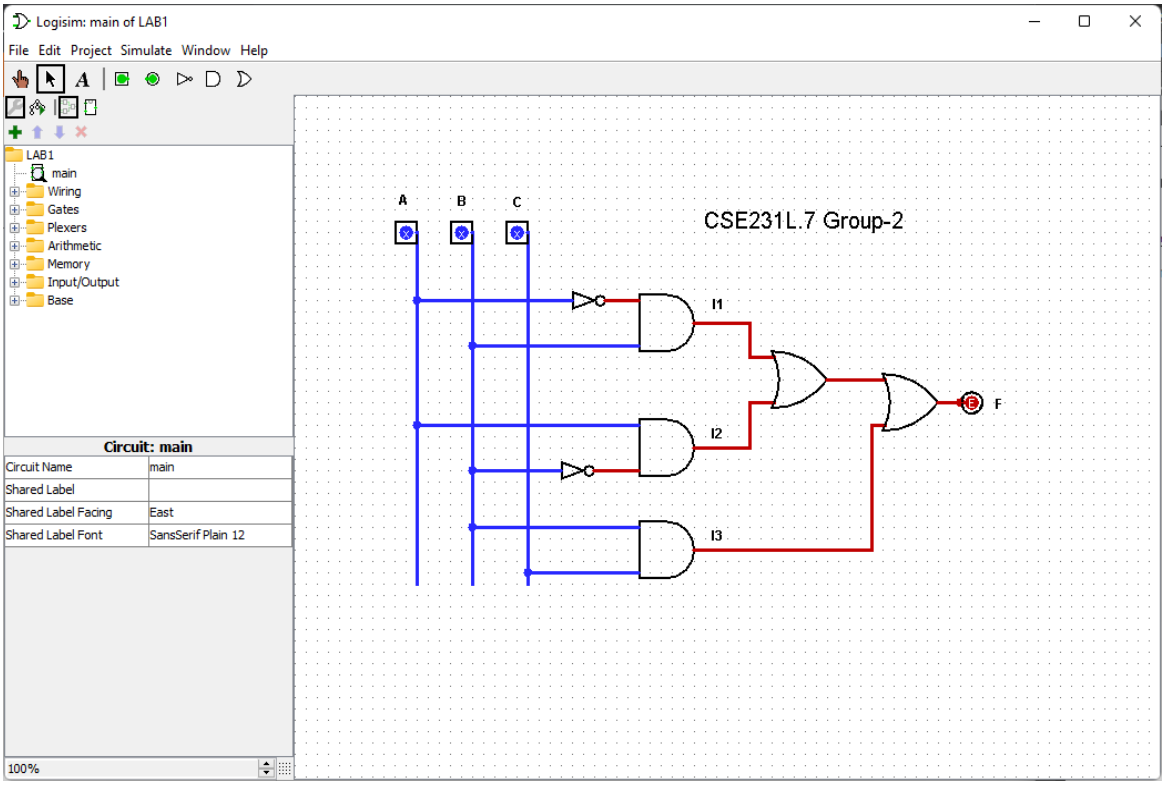
Logisim Simulation



Screenshot 1: Experiment 1



Screenshot 2: Experiment 2



Screenshot 3: Experiment 3

Data Table

Input A	B	AND F= A·B	OR F=A+B	NAND F=$\overline{A \cdot B}$	XOR F=A⊕B	NOR F=$\overline{A + B}$
0	0	0	0	1	0	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	1	0	0	0

Input A	NOT F= $\overline{A}$
0	1
1	0

Table F.1.1: Truth Table of Logic Gate

A	B	C	F= ABC	F=A+B+C
0	0	0	0	0
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	1	1

Table F.2.1: Truth Table for 3-input AND or OR

F = ABC = A (BC) = (AB) C
F = A+B+C= A+ (B+C) = (A+B) + C

Table F.2.2: Expressing 3-input gates as 2-input gates using associative law

A	B	C	$I_1 = \overline{A}C$	$I_2 = A\overline{B}$	$I_3 = BC$	F= I_1+I_2+I_3
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	0
0	1	1	1	0	1	1
1	0	0	1	0	1	1
1	0	1	0	1	0	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Table F.3.1: Truth table for given Boolean Expression

Q&A

1. What are the names of the ICs that you would need if you wanted to use 13 AND gates, 12 NOT gates and 15 NOR gates in a circuit? How many of each IC would you need?

   Answer: For 13 AND gates, we would need 4 7408 ICs. Each 7408 IC contains 4 AND gates. So, we need [13/4] = 4 ICs of 7408 IC.

   For 12 NOT gates, we would need 2 7404 ICs. Each 7404 IC contains 6 NOT gates. So, we need [12/6] = 2 ICs of 7404 IC.

   For 15 NOR gates, we would need 4 7402 ICs. Each 7402 IC contains 4 NOR gates. So, we need [15/4] = 4 ICs of 7402 IC.

2. How can you power your logic IC if the +5V port of your trainer board stops working?

   Answer: We can power our logic ICs form input switches of trainer board. For +5 volts connections we need to switch the switch to 1 and for GND connection we need to set the switch at 0.

3. Explain the Associative Law of Boolean algebra.

   Answer: Associative Law states that the operation can be performed in any order when the variables priority is the same. As “AND” and “OR” have same priority.

- A+(B+C) = (A+B)+C = A+B+C (ORAssociateLaw)
- A (B.C) = (A.B) C = A . B . C (AND Associate Law)

4. What is a Truth Table? Draw the Truth Table for an XNOR gate.

   Answer: The truth table displays the logical operations on input signals in a table format. Every Boolean expression can be viewed as a truth table. The truth table identifies all the possible input combinations and the output for each. It is common to create the table so that the input combination can produce an unsigned binary up-count.

   Truth table of 2-input Exclusive NOR gate:

A	B	XNOR $F = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

5. Let us assume you have two logical inputs A and B. If you pass A and B through a NAND gate and then pass the output of the NAND gate through a NOT gate, what logical operation will your final output represent? What is the name of the Boolean Algebra theorem that can be used to find this answer?

   Answer: Logical expression of the final output will represent AB. The name of the Boolean Algebra Theorem is Involution Law that can be used to find the answer.

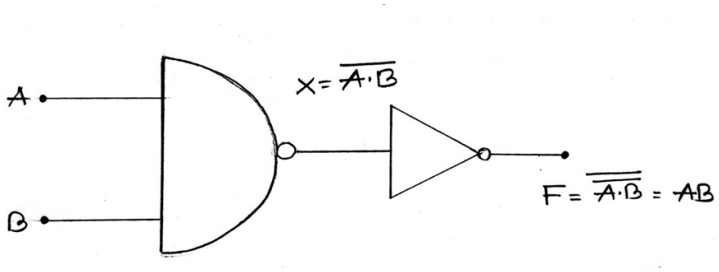
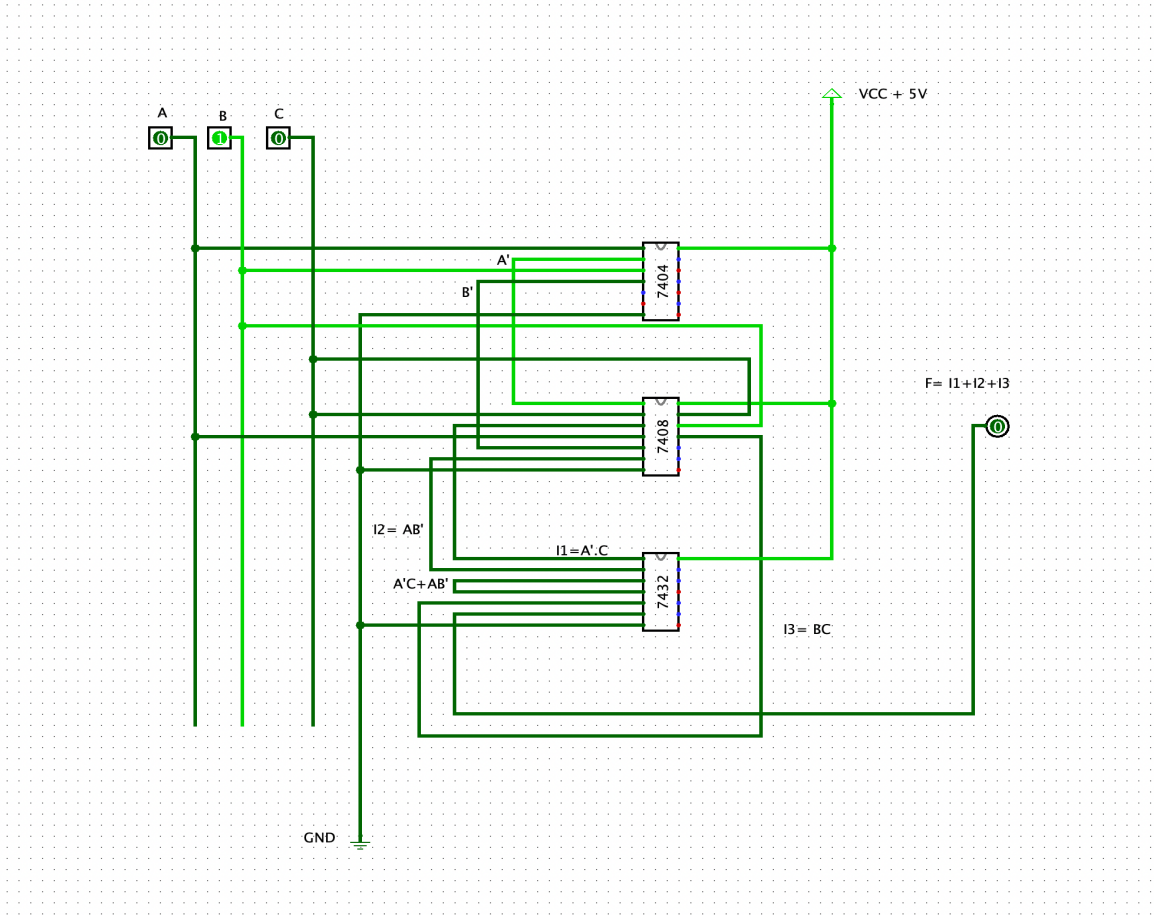


Figure: Application of Boolean Involution Law

6. Draw the IC diagram for the circuit in Figure F.3.1. In place of the logic gates, draw the ICs and all the connections required to make the circuit work.

Answer:



7. How can you use a 3 input AND gate as a 2-input AND gate? Can you use the same method to use a 3-input OR gate as a 2-input OR gate?

   Answer: By using the Associative Law theorem, if we pass A and B through a AND gate and then pass the output of the AND gate through another AND gate and also pass C as the second input, then we can use a 3-input AND gate as a 2-input AND gate.

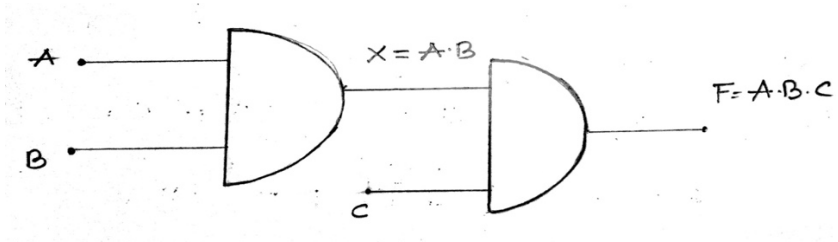


Figure: 3-input AND gate as a 2-input AND gate

Yes, we can use the same method to use a 3-input OR gates as a 2-input OR gate. If we pass A and B through a OR gate and then pass the output

of the OR gate through another OR gate and also pass C as the second input, then we can use a 3-input OR gate as a 2-input OR gate.

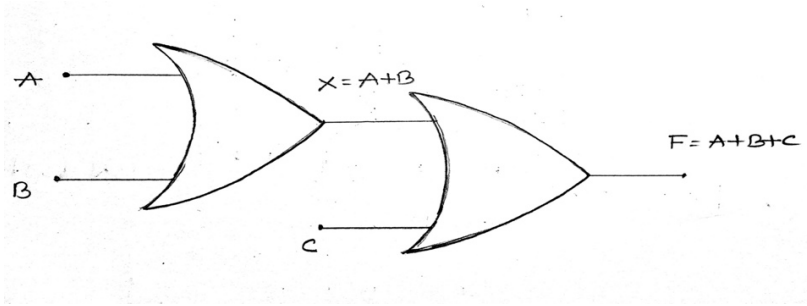


Figure: 3-input OR gate as a 2-input OR gate

Discussion

In the Lab-1, we did three experiments to understand the digital logic gates and Boolean functions. In the First Experiment, our prime goal was to learn the primary characteristics of the AND, OR, NOT, XOR, NAND & NOR gates and how to build the gates theoretically with the Logisim. Then we learned how to use gates to simulate a circuit diagram. In the case of the NOT gate, if the input is 0, then the output is 1. If the circuit input is 1, the circuit output will be 0. So, the NOT gate performs logical negation on its circuit input. Following these basic operations, we did our first experiment.

In our Second Experiment, our primary goal was to prove Associative Law. We construct a 3-input AND gate or OR gate from 2-input AND or OR gates. So, we drew our desired circuit for experiment 2. The circuit output truth table matched our truth table. So, after analyzing this truth table, we learned that the associative law is the same for both practical simulation and theoretical proof. Then we simulated 3-input AND gate in Logisim using only 2-input AND gates.

In the Third Experiment, our goal was the implementation of Boolean functions. We have taken a Boolean Equation, $F = A'C + AB' + BC$ and theoretically solved it with basic logic gates AND, OR & NOT using Logisim. Practically when we tried to build the logic circuit on the trainer box, we faced a problem with the circuit output. We didn't get the desired circuit output of the truth table. The output LED light was always on whenever we switched on the trainer box. So, we checked the IC placements, Vcc and GND connections, logic gate connections, and wired connections. But every circuit connection on the trainer box was perfect. The reason behind this problem could be damage to the IC pins, a defective hole in the breadboard or a technical issue in the trainer box.