



North South University
Department of Electrical & Computer Engineering
LAB REPORT-1

Course Code: CSE231L

Course Title: Digital Logic Design

Section: 8

Experiment Number: 1

Experiment Name:

Digital Logic Gates and Boolean Functions

Experiment Date: 26/6/2021

Date of Submission: 3/7/2021

Submitted By: Md. Rifat Ahmed - 1931725042

Course Instructor: Md. Shahriar Hussain

Submitted To: Md. Anisur Rahman Asif

Objectives:

- We have to study the basic logic gates - AND, OR, NOT, NAND, NOR, XOR.
- We also have to get acquainted with the representation of Boolean functions using truth tables, logic diagrams and Boolean Algebra.
- Then we need to prove the extension of inputs of AND and OR gates using the associate law.
- And finally we have to become familiarized with combinational logic circuits.

Apparatus:

- 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

Theory:

Logic Gates:

Logic gates are called the elementary building blocks of digital circuits. They perform logical operations of one or more inputs to give a single output. I.e., if we give two inputs 1 and 0 in an AND gate it'll give us the output 0.

There are seven basic digital logic gates each having their own distinct IC number and symbol. So, its important to know the IC numbers of the basic gates and the IC numbers of the basic gates are,

AND gate IC no. 7408, OR gate IC no. 7432, NOT gate IC no. 7404, NAND gate IC no. 7400, NOR gate IC no. 7402, XOR gate IC no. 7486 & XNOR gate IC no. 74266.

Truth Tables:

Truth tables gives us all the output of a logic circuit for every possible combination of inputs. If we look at Table 3 in "Data Sheet" part we can see that the table gave the output F for the input combination $A'C+AB'+BC$ with 0s and 1s indicating only a single output for the combination.

Boolean Algebra:

Boolean algebra is also a branch of mathematics which we work with variables that has only two values true and false or mathematically indicated as 1 and 0 respectively. It's mainly used in the digital electronics field where networks are generally given in Boolean functions. However, there are some laws and theorems that Boolean algebra follows such as Associative law, De-Morgan's law, Distributive law etc.

Experimental Procedure:

Experiment 1:

At first, we need to place the 7408 AND IC on a breadboard then we need to connect the Vcc and GND pins to the +5V and GND ports of the trainer board respectively. Then we need to label the pin numbers of the inputs and output of the gates in Figure 1. After that we need to connect each input of the logic gate into a toggle switch on the trainer board and connect the output to an LED on the trainer board. Then apply all combinations of inputs by toggling the switches on and off and record if the LED is on or off as the output of the gate, 1 if on and 0 if off and record the results in Table 1. Then we need to replace the AND IC with OR, NAND and XOR ICs without changing the connections and repeat the previous steps for each of the gate and record the results in Table 1 and then again repeat the same process for NOT and NOR ICs.

Experiment 2:

We need to complete the truth table for 3-input AND gate in Table 2.1. Then we need to express the 3-input function using 2-input AND gates in Table 2.2 using the associative laws, $A + (B + C) = (A + B) + C$ & $A(BC) = (AB)C$. Then we have to label the pin numbers in Figure 2 and then connect the circuit. Then connect the output to an LED and verify it using the truth table. And finally, we need to repeat all those steps for the 3-input OR gate.

Experiment 3:

For the following Boolean Equation, $F = A'C + AB' + BC$ we need to complete the truth table for the implicants, $I_1 = A'C$, $I_2 = AB'$ and $I_3 = BC$ and using the values of the implicants we need to complete the truth table for F in Table 3. Then we need to label the pin numbers for the NOT, AND & OR gates of the function F in Figure 3. After that we have to connect the input A to a NOT gate using the pins assigned and check the output using the LED. Then wire up implicant I_1 and connect the output of I_1 to an LED and verify the result using the truth table. Then again do the same thing for input B and C and then wire up I_2 and I_3 respectively connect their output to an LED and verify them using the truth table. Then finally connect the outputs of the three implicants as inputs to the OR gates like Figure 2 and connect the output F to an LED to verify the function using the truth table.

Question/Answer:

Answer to the Question No. 1 of Experiment 1:

The names of the ICs that I would need if I wanted to use 13 AND gates, 12 NOT gates and 15 NOR gates in a circuit are "IC#7408" for AND gates, "IC#7404" for NOT gates and "IC#7402" for NOR gates.

Now normally there are 4 gates in an AND gates IC so for 13 AND gates we would need $13/4=3.25\approx 4$ IC#7408.

Then for NOT gates IC there are 6 gates so for 12 NOT gates we would need $12/6=2$ IC#7404.

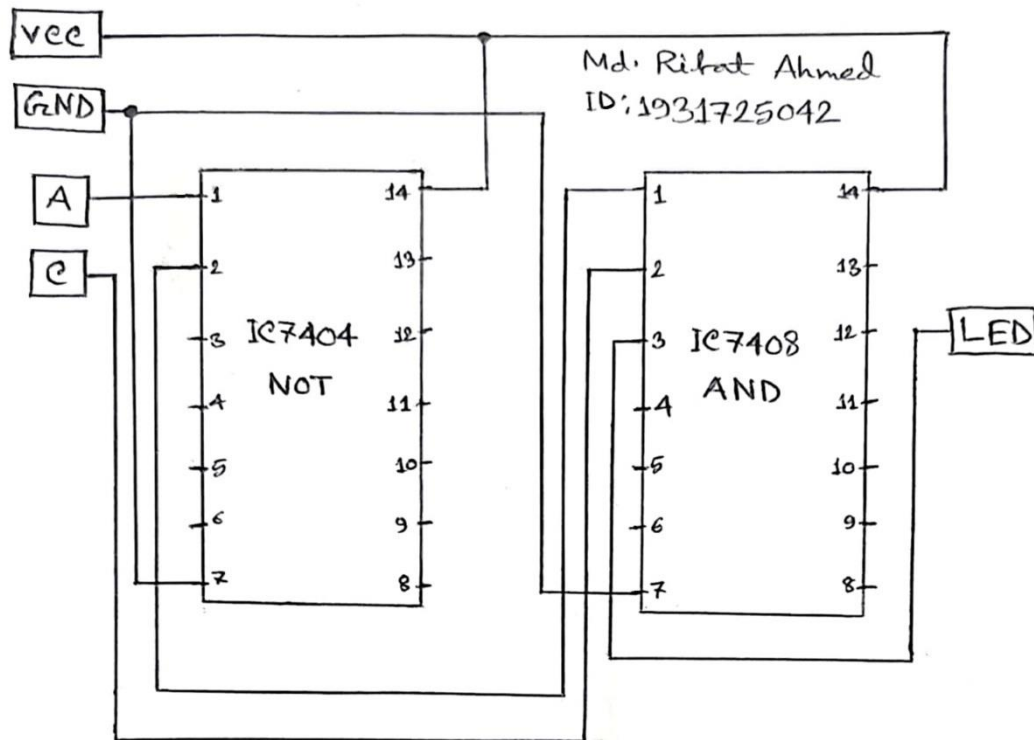
And then for the NOR gates IC there's 6 gates in them so for 15 NOR gates we would need $15/4=3.75\approx 4$ IC#7402.

Answer to the Question No. 2 of Experiment 1:

If the +5V port of our trainer board stops working then we can use any of the toggle switch from the board to power our logic ICs. Toggling the switches up will give us the +5V.

Answer to the Question No. 1 of Experiment 3:

IC diagram for the first implicant $I_1 = A'C$:



IC Diagram of Experiment 3 1st Implicant $\rightarrow A'C$

Discussion:

Through this lab we learned that logic gates are the most important element of a digital circuit. And then we learned the different IC numbers of the basic gates. Then we learned how to draw an IC diagram. We also learned how to simulate circuits using Logisim. After that in experiment 1 we verified the output of different gate ICs via LED using the truth table in Table 1. Then in the 2nd experiment we learned how to construct a 3-input AND & OR gates from 2-input AND & OR gates. We had to connect the output of one gate as another ones input and then connect the LED with the output of the latter to get the final result and then verify it using the truth table in Table 2.1. Then we learned to express 3-input gates as 2-input gates using the associative law. And then in experiment 3 we took a Boolean function with 3 implicants made their truth table using the data from the simulation where we had to use 2 NOT gates and 3 AND gates for the 3 implicants and then 2 OR gates for the 3 outputs of the 3 implicants. Here we had to use 2 OR gates because we are only given 2-input OR gates. However, as we didn't do the experiment in real life physical lab so the experience will vary a little. But overall, it was a fun experiment where we learned a lot of new things which are going to be very helpful for us in the future.

Data Sheet & Circuit Diagrams:

Introduction to Basic Logic Gates:

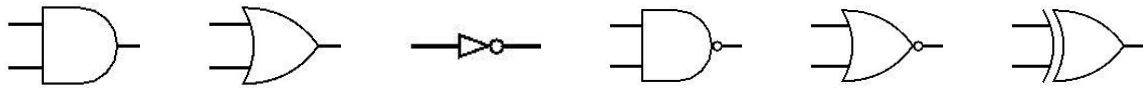


Figure 1: Pin configurations of gates in ICs

Input $A \ B$	AND $F = A \cdot B$	OR $F = A + B$	NAND $F = \overline{A \cdot B}$	XOR $F = A \oplus B$	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 1: Truth Table of Logic Gates

Constructing 3-input AND & OR gates from 2-input AND & OR gates:

$A B C$	$F = ABC$	$F = A + B + C$
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 2.1: Truth Tables for 3-input AND & OR

$F = ABC = (A.B).C$
$F = A + B + C = (A+B)+C$

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

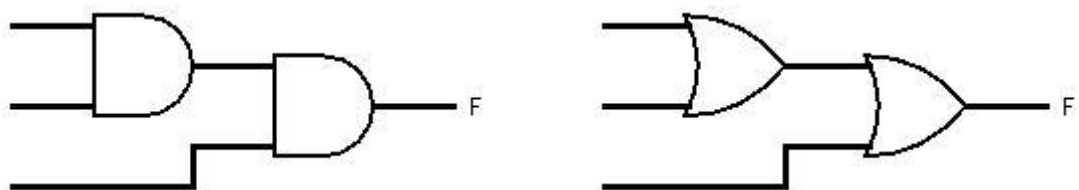


Figure 2: Extension of inputs of AND & OR gates

Implementation of Boolean Functions:

$A B C$	$I_1 = A'C$	$I_2 = AB'$	$I_3 = BC$	$F = I_1 + I_2 + I_3$
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	0	1	0	1
1 0 1	0	1	0	1
1 1 0	0	0	0	0
1 1 1	0	0	1	1

Table 3: Truth Table for the given Boolean Function

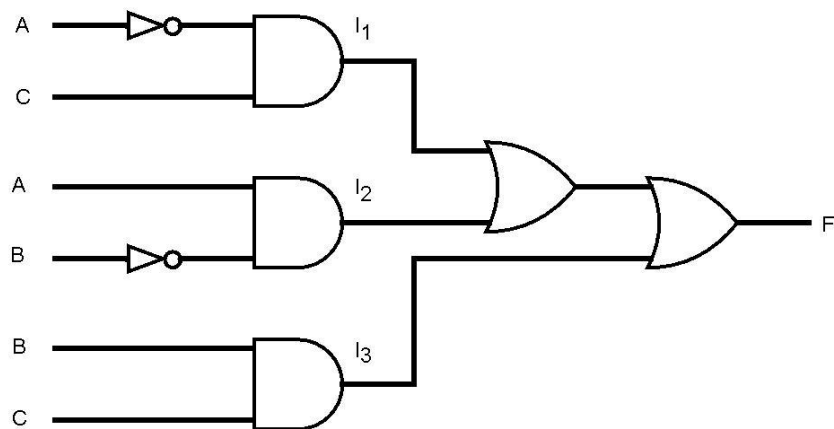
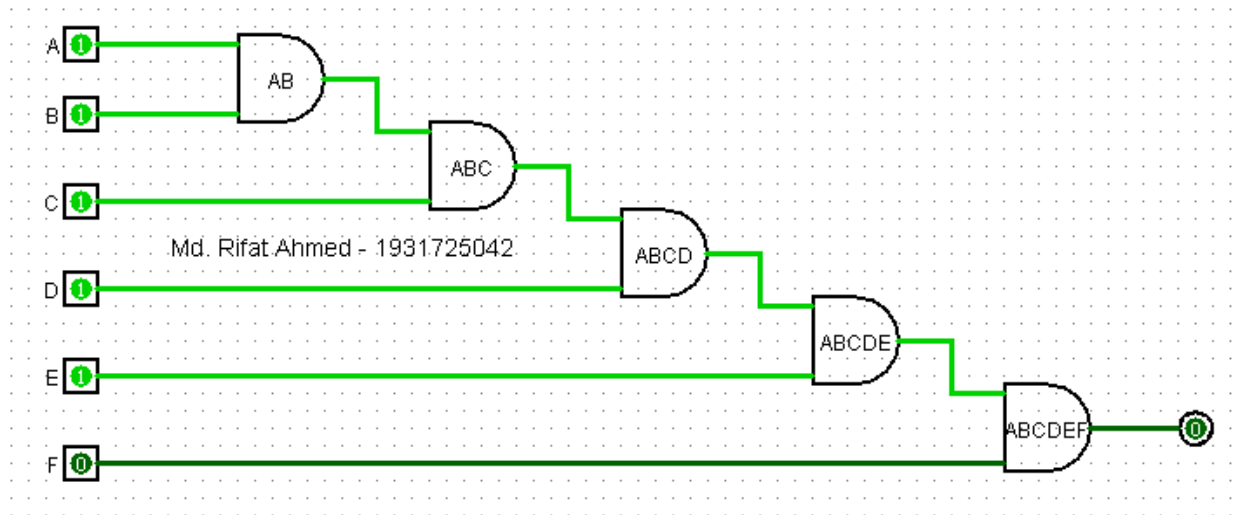


Figure 3: Logic Diagram for the given Boolean Function

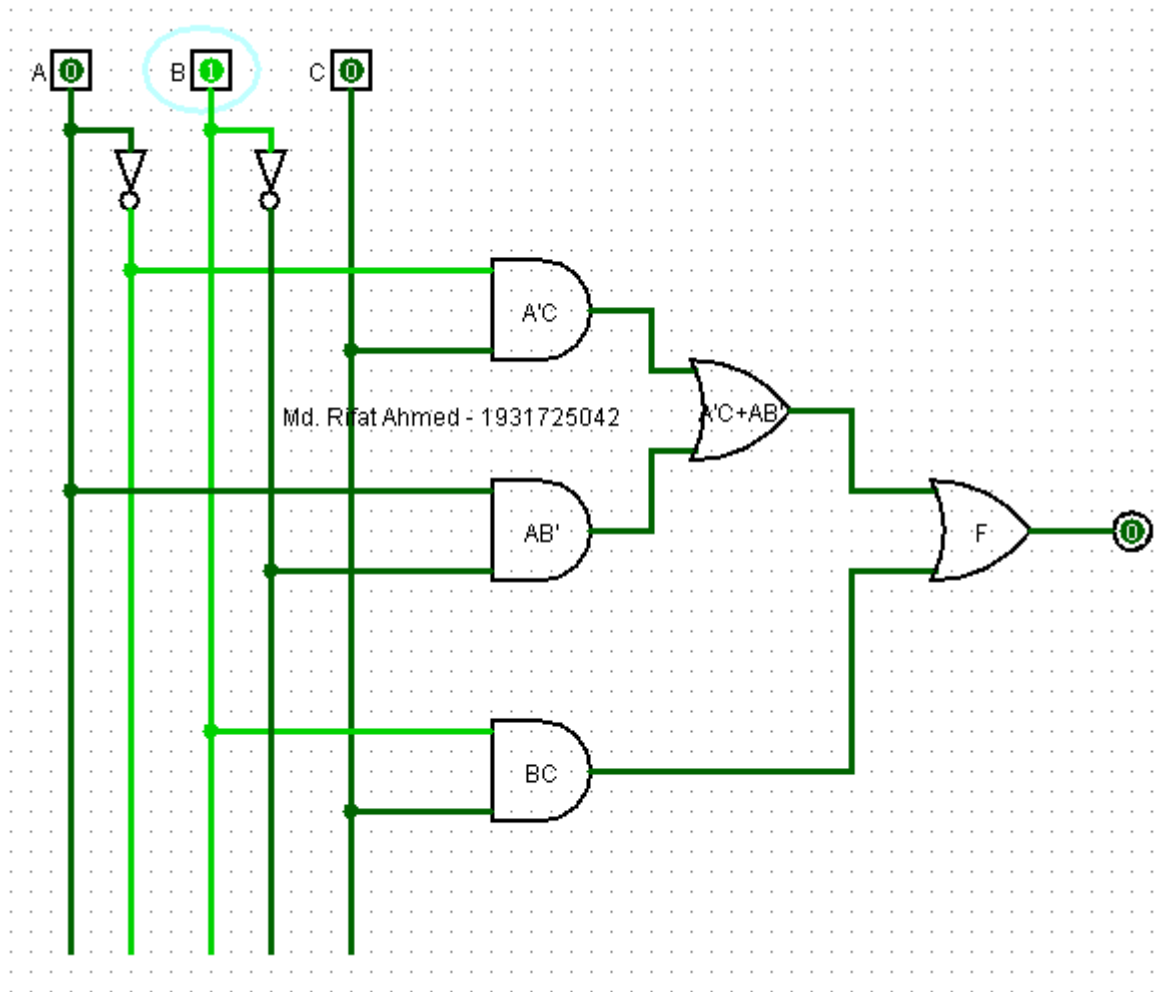
Simulation:

Simulating a 6-input AND gate in Logisim using only 2-input AND gates:



Simulation of Experiment-2

Simulating the combinational logic circuit constructed in experiment-3 in Logisim and showing the instance when the input is ABC = 010:



Simulation of Experiment-3