Experiments-1 Name: Introduction to Basic Logic Gates

Objective:

- Knowing the logic gates AND, OR, NOT, NAND, NOR, XOR and their truth table.
- Identifying an IC and its PIN order.
- How to test an IC working or not.

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

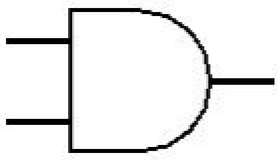
Theory:

Digital logic gates operate at two discrete voltage levels representing the binary values 0 (logical LOW) and 1 (logical HIGH). A brief description of each gate is given below:

AND Gates:

A logic gate produces a HIGH output only when all of the inputs are HIGH.

Symbol:



Truth Table:

IC: 7408 Quadruple 2-input AND gates

OR Gates:

A logic gate that produces a HIGH output when one or more inputs are HIGH.

Symbol:

Truth Table:

IC: 7432 Quadruple 2-input OR gates

NOT Gates:

A logic gate that inverts or complements its input.

Symbol:

Truth Table:

IC: 7404 Hex Inverters (NOT gates)

NAND Gates:

A logic gate that produces a LOW output only when all the inputs are HIGH.

Symbol:

Truth Table:

IC: 7400 Quadruple 2-input NAND gates

NOR Gates:

A logic gate in which the output is LOW when one or more inputs are HIGH.

Symbol:

Truth Table:

IC: 7402 Quadruple 2-input NOR gates

XOR Gates:

A logic gate produces a HIGH output only when its two inputs are at opposite levels.

Symbol:

Truth Table:

IC: 7486 Quadruple 2-input XOR gates

Identifying IC:

For IC numbers, we need to read the number of them, ignoring other letters. For example, **74**HC**04**N is the 7404 Hex Inverter IC, where the HC denotes a high-speed TTL circuit CMOS variant.

Identifying IC PIN Numbers:

The basic rule for most ICs is that there is a polarity mark, such as the half-moon notch shown in the figure. Another standard polarity mark is a tiny 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.

Circuit Diagram:

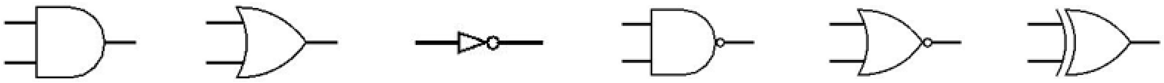


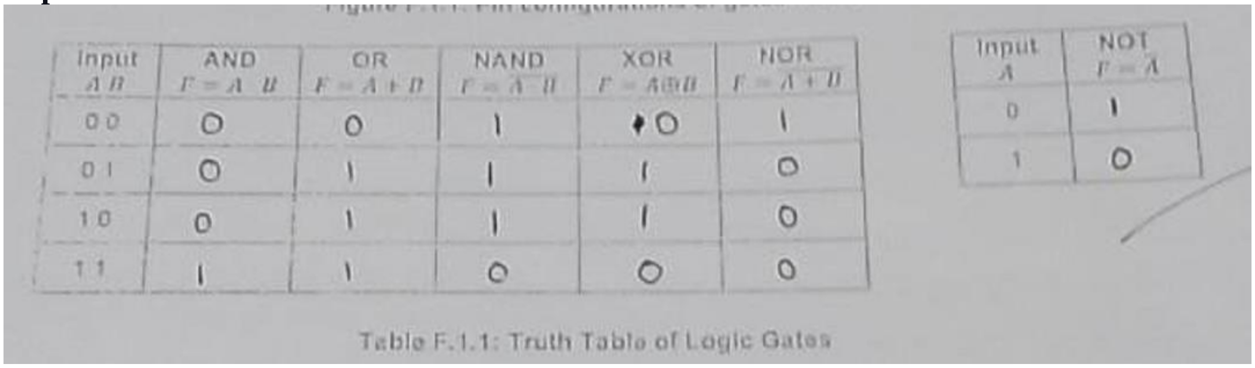
Figure F.1.1: Pin configurations of gates in ICs

Experimental Procedure:

1. First, we place the 7408 AND IC on the breadboard as every pin is in a separate node.
2. Then we connect the VCC and GND pins of the IC to the +5 V and GND ports of the trainer board, respectively.
3. After that, we connect each input of the gate to a toggle switch and connect the output to an LED on the trainer board.
4. Then, we test all the combinations of inputs by turning the toggle switches on (1) and off (0) and recording if the LED is on (1) or off (0) as the output of the gate.
5. Then we replace the AND IC with OR, NOT, NAND, NOR, and XOR ICs. Then we repeat step 4 for each ICs also maintain the input and output order by repeating step 3.

Simulation: N/A

Experimental Data Table:



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 F.1.1: Truth Table of Logic Gates

Results:
All the data are correct according to the truth table of each gate.

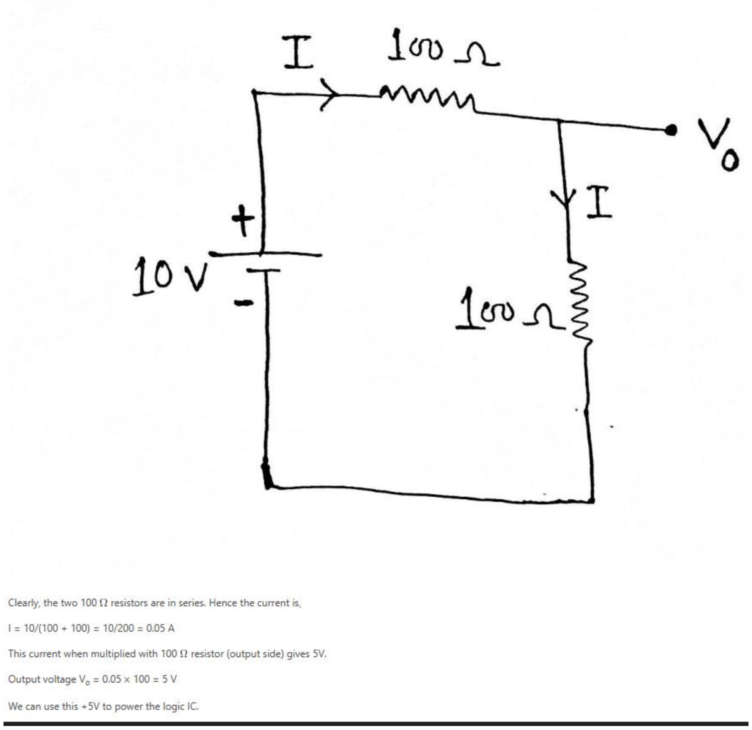
Questions and Answers (Q/A):

1.

We need
7408 Quadruple 2-input AND gates IC - 5x
7404 Hex Inverters (NOT gates) IC - 2x
7402 Quadruple 2-input NOR gates IC - 4x

2.

If the +5V port is not working, we can design a +5V power supply very easily, as below. We only require a 10V battery and two 100 Ohm resistors so that the output voltage is +5V. This way, we can design a voltage divider network, as shown below.



Clearly, the two 100 Ω resistors are in series. Hence the current is,

I = 10/(100 + 100) = 10/200 = 0.05 A

This current when multiplied with 100 Ω resistor (output side) gives 5V.

Output voltage V_o = 0.05 × 100 = 5 V

We can use this +5V to power the logic IC.

Discussion:
From the first test run, our ICs were working perfectly. Then we tried to make NAND Gate by combining the AND and NOT Gates ICs. But we couldn't do that; then, we figured out that the breadboard was not tight enough to connect all the pins of ICs. Then for faster work, we bring NAND IC and complete our other test. Finally, we completed all our test runs within the time.

Experiments-2 Name: Constructing 3-input AND & OR gates from 2-input AND & OR gates.

Objectives:

- Prove the extension of inputs of AND and OR gates using the associate law
- Making three inputs AND and OR gates using two inputs AND and OR Gates ICs.

Apparatus:

- IC 7408 Quadruple 2-input AND gates
- IC 7432 Quadruple 2-input OR gates
- Trainer Board
- Wires

Theory:
We know that 7408 IC has four AND gates with two inputs. But if we need to take three inputs, we can combine two AND gates of this IC. Just connect the output pin with an input pin of another gate using a wire. Then connect the third input to the second input pin of the second gates. Then the output pin of the second gate will give us the output of three inputs AND gates.

Circuit Diagram:

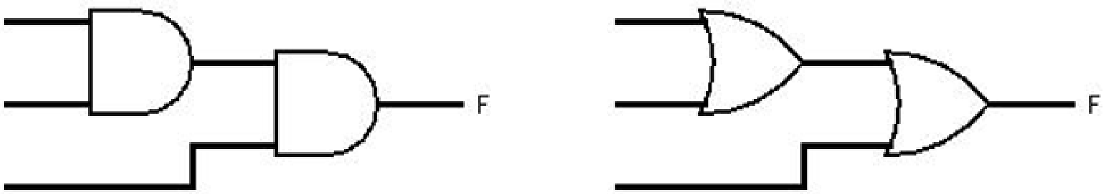


Figure F.2.1: Extension of inputs of AND and OR gates

Experimental Procedure:

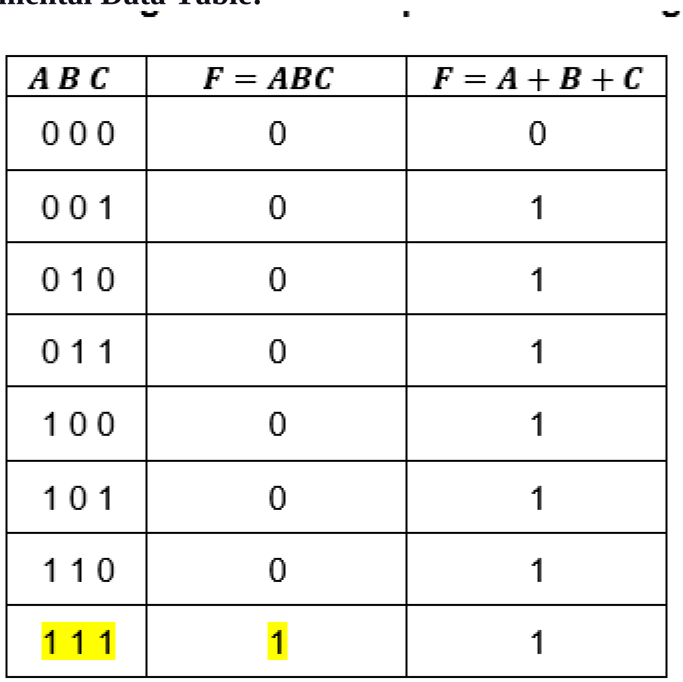
1. First, we complete the truth table for the 3-input AND gate in Truth Table.
2. Then we place the 7408 AND IC on the breadboard as every pin is in a separate node, and then we connect the VCC and GND pins of the IC to the +5 V and GND ports of the trainer board, respectively.

3. After that, we connect two inputs in PINs 1 and 2. Then we connect the output pin 3 with input pin 4. After that, we connect the third input to the input pin 5.
4. Then we connect the output pin to the LED.
5. Then, we test all the combinations of inputs by turning the toggle switches on (1) and off (0) and recording if the LED is on (1) or off (0) as the output of the gate. Then match the output to the truth table.
6. Then we replace the AND IC with OR IC and repeat step 5.

Simulation:

Attached.

Experimental Data Table:



$A\ B\ C$	$F = ABC$	$F = A + B + C$
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 Tables for 3-input AND and OR

Results:

All the data table's output results match the truth table of three inputs AND gate. Also, matched with OR Gate.

Questions and Answers (Q/A):

N/A

Discussion:

After the first experiment, we were so excited to build our first circuit. Then we make a three-input AND gate using the two-input AND gate very quickly. We don't face any difficulty in this experiment.

Experiment-3 name: Implementation of Boolean Functions

Objectives:

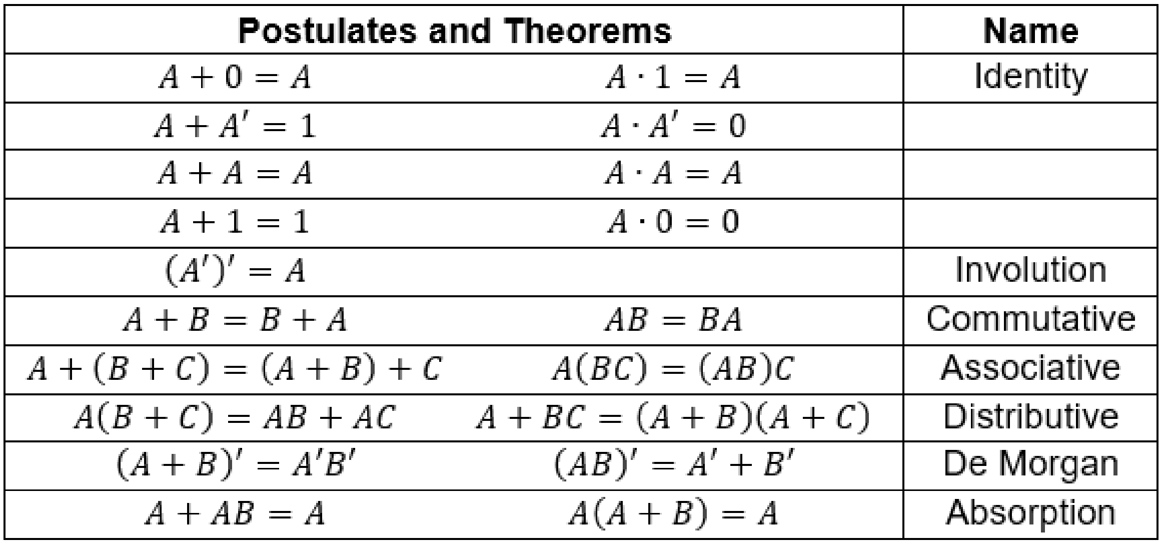
- Get acquainted with the representation of Boolean functions using truth tables, logic diagrams and Boolean Algebra.
- Become familiar with combinational logic circuits.

Apparatus:

- IC 7408 Quadruple 2-input AND gates
- IC 7432 Quadruple 2-input OR gates
- IC 7404 Hex Inverters (NOT gates)
- Trainer Board
- Wires

Theory:

Boolean algebra is the mathematics of logic circuits. The Boolean variables are represented as binary numbers to represent truths: 1 = true and 0 = false. Elementary algebra deals with numerical operations, whereas Boolean algebra deals with logical operations.
Some Postulates and Theorems are given below:



Postulates and Theorems		Name
$A + 0 = A$	$A \cdot 1 = A$	Identity
$A + A' = 1$	$A \cdot A' = 0$	
$A + A = A$	$A \cdot A = A$	
$A + 1 = 1$	$A \cdot 0 = 0$	
$(A')' = A$		Involution
$A + B = B + A$	$AB = BA$	Commutative
$A + (B + C) = (A + B) + C$	$A(BC) = (AB)C$	Associative
$A(B + C) = AB + AC$	$A + BC = (A + B)(A + C)$	Distributive
$(A + B)' = A'B'$	$(AB)' = A' + B'$	De Morgan
$A + AB = A$	$A(A + B) = A$	Absorption

Table B.3: Laws of Boolean algebra

Circuit Diagram:

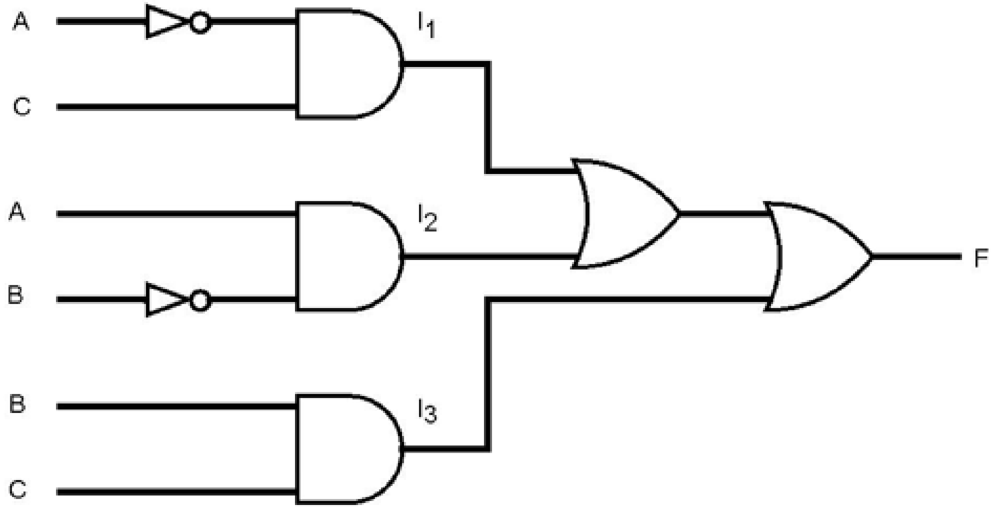


Figure F.3.1: Logic Diagram for the given Boolean Function

Experimental Procedure :

Consider the following Boolean Equation:

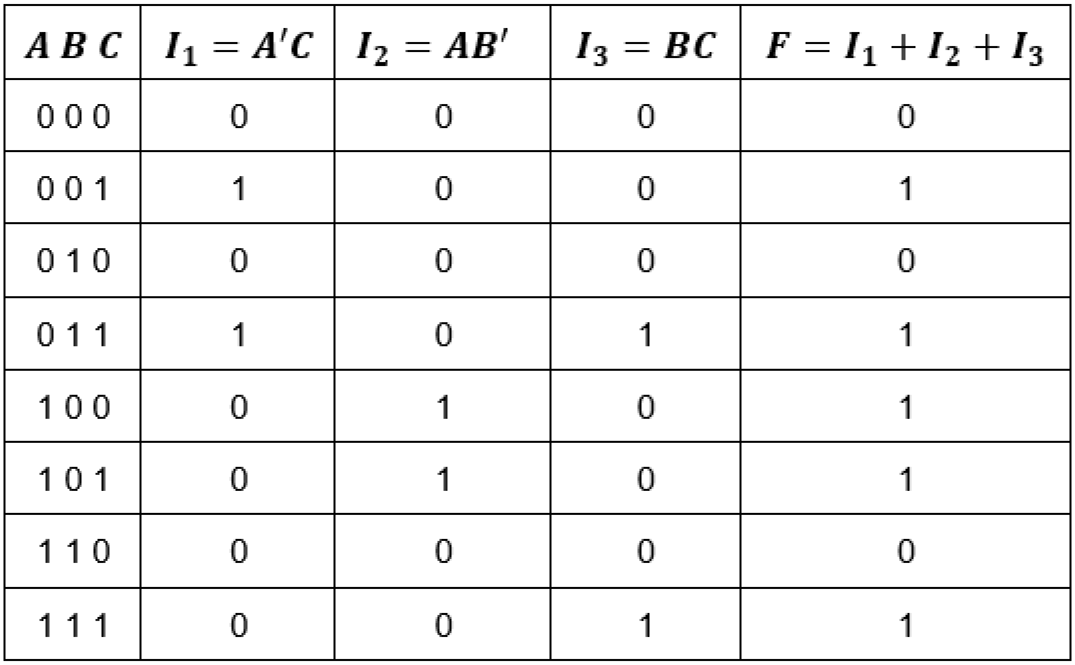
$$F = A'C + AB' + BC$$

1. Then we complete the truth table for the implicants , $I_1 = A'C, I_2 = AB'$ and $I_3 = BC$
2. Then, we make three input branches separately for A, B, and C to take multiple inputs from one input switch.
3. Then we implicant I1. First, we take an input from A and connect it with NOT gate PIN-1 and connect the output from PIN-2 to the AND gate PIN-1. Then we connect another input from C with the AND gate PIN 2. And connect the Output pin of AND gate PIN-3 to a LED for testing purposes. After a successful test, we connect the output with OR gate PIN-1.
4. After that, we implemented I2. First, we take input from A and connect with AND Gate input PIN-4. Then we take another input from B and connect with NOT gate input PIN-3. Then take the output from PIN-4 and connect with AND gate input PIN-5. Then, we take the output from PIN-6 and connect it with LED for testing purposes. After a successful test, we connect the output with OR gate input PIN-2.
5. After that, we implemented I3. First, we take two inputs from B and C and connect with AND gate input PINs 9 and 10. Then take the output from PIN-8 and connect it with a LED for testing purposes. After a successful test, we connect the output with OR gate input PIN-5.

6. We connect the OR gate output PIN-3 with input PIN-4. Then we connect the output PIN-6 with an LED. And test for every possible combination and verify with the truth table.

Simulation:
Attached.

Experimental Data Table:



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

Figure F.3.1: Truth Table for the given Boolean Function

Results:
All data collected from step 6 are accurate with the truth table of this Boolean function. And we completed this experiment successfully.

Questions and Answers (Q/A):

1. **Draw the IC diagram for the first implicant I1. In place of the logic gates, draw the ICs and all the connections required to make the circuit work.**

Draw step 3 with IC

Discussion:
When we are doing step 6, we face some difficulties. The result was different from the truth table at the first output. Then we tried to figure out why it was not working and checked the wire connection from step 2. And we found that in the OR gate, we give the wrong input in VCC and GND. Then we reverse the inputs then test again. And then it's working perfectly.