Experiment Name:

Digital Logic Gates and Boolean Functions.

Objectives:

1) Study the basic logic gates – AND, OR, NOT, NAND, NOR, XOR.

2) Acquaint with the representation of boolean functions using truth tables, logic diagrams and Boolean algebra.

3) Prove the extension of inputs of AND and OR gates using the associate law.

4) Familliarize with combinational logic circuit.

Theory:

Logic Gates – Logic gates are the elementary building blocks of digital circuits. They perform logical operations of one or more logical inputs to produce a single output. Digital logic gates operate at two discrete voltage levels representing the binary values – 0 (logical LOW) and 1 (logical HIGH). The basic logic gates with their corresponding IC numbers and circuit symbols are given below –

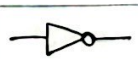
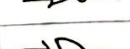
Gate	IC #	Symbol
AND	7408	
OR	7432	
NOT	7404	
NAND	7400	
NOR	7402	
XOR	7486	

Table C.1 – Logic Gates

Truth Table:

A truth table shows all the output logic levels of a logic circuit for every possible combination of inputs. Following table shows the truth table for a two-input AND Gate.

AB	F = A.B
00	0
01	0
10	0
11	1

Table C.2 – Truth table of AND

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 and is not affected by previous states. The analysis of combination logic requires writing the boolean functions for each element of the circuit, producing

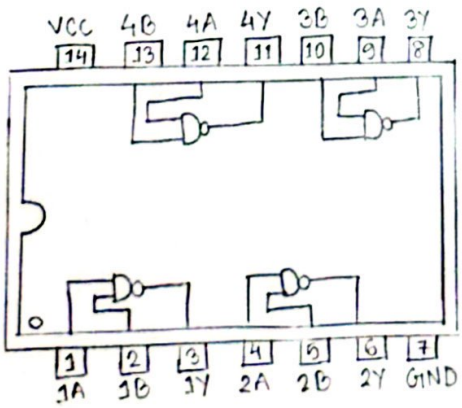
their truth tables, and subsequently combining each function for the final output and truth table.

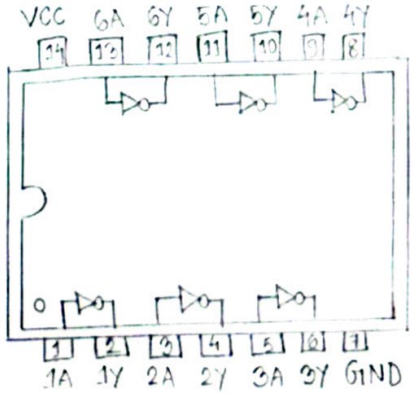
Integrated Circuits - IC

ICs typically feature a polarity mark like a half-moon notch or a dot, triangle or tab by pin 1. The convention is to number the pins counterclockwise from this mark, starting at 1. In absence of a direct mark, pin numbers can be inferred from the orientation of the next inscribed on the IC.

7400 Series IC

The 7400 series of TTL (Transistor-Transistor Logic) IC are widely used in digital logic circuits, although many have been replaced by CMOS technology. To identify the IC number, simply read the number on the chip, ignoring the letters. Pin configurations for basic logic gates like NAND, AND, OR, XOR are same. Pin 7 is GND (logical LOW) and pin 14 is connected to +5V as VCC (logical HIGH).





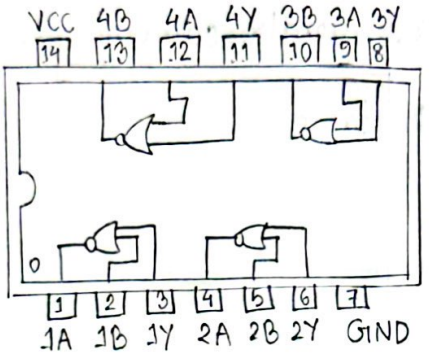


Figure C.2 – Schematic of a) 7400 NAND, 7408 AND, 7432 OR and 7486 XOR, b) 7404 NOT, and c) 7402 NOR ICs

Apparatus List:

1. IC 7400 Quadruple 2-input NAND gates
2. IC 7402 Quadruple 2-input NOR gates
3. IC 7404 Hex Inverters (NOT gates)
4. IC 7408 Quadruple 2-input AND gates
5. IC 7432 Quadruple 2-input OR gates
6. IC 7486 Quadruple 2-input XOR gates
7. Trainer Board

Experimental Data:

AB	F = A.B	F = A+B	F = (A.B)′	F = A⊕B	F = (A+B)′
00	0	0	1	0	1
01	0	1	1	1	0
10	0	1	1	1	0
11	1	1	0	0	0

Table D.1 – Truth table of logic gates

A	F = A′
0	1
1	0

ABC	F = ABC	F = A+B+C
000	0	0
001	0	1
010	0	1
011	0	1
100	0	1
101	0	1
110	0	1
111	1	1

Table D.2 – Truth table of 3-input AND and OR

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

Table D.3 – Associative Law

ABC	$I_1 = A'C$	$I_2 = AB'$	$I_3 = BC$	$F = I_1 + I_2 + I_3$
000	0	0	0	0
001	1	0	0	1
010	0	0	0	0
011	1	0	1	1
100	0	1	0	1
101	0	1	0	1
110	0	0	0	0
111	0	0	1	1

Table D.4 – Truth table of logic gates

Circuit Diagram:

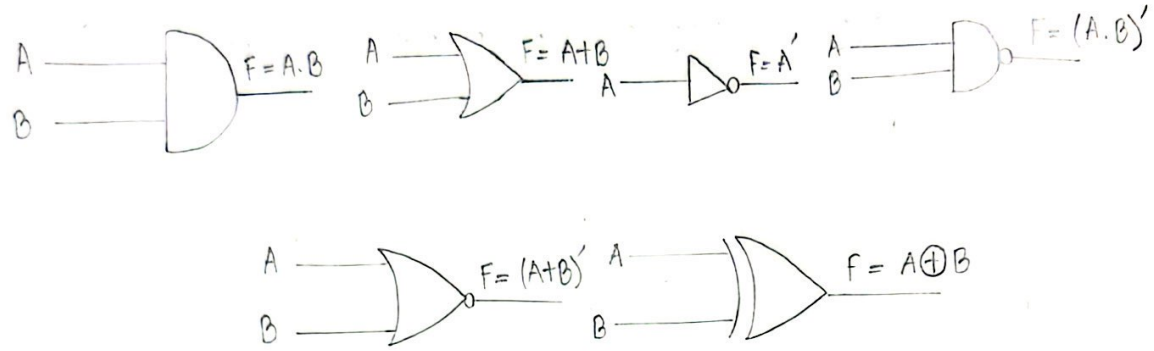


Figure D.1 – Pin configuration of gates in ICs

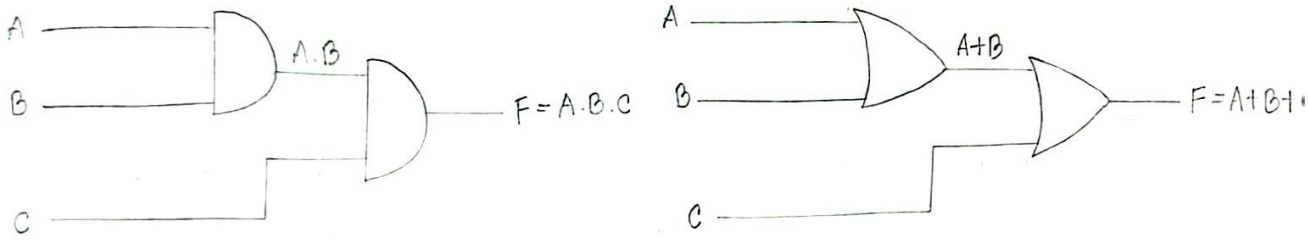


Figure D.2 – Extension of inputs of AND and OR gates

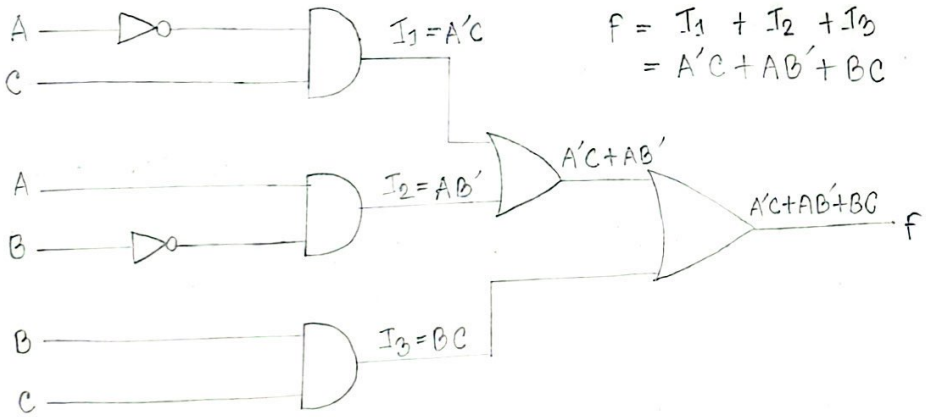


Figure D.3 – Circuit diagram for the boolean function F

Questions & Answers:

1| Is it possible to make a 3-input NAND or NOR gate with 2-input NAND or NOR gates? Justify your answer.

Ans: Yes, it is possible to make a 3-input NAND or NOR gate with 2-input NAND or NOR gates. When we want to make a 3-input NAND or NOR gate, we can't connect 3 inputs directly. To connect 3 inputs, we have to connect 2 inputs first. After that, we can connect the 3rd input with the output of first 2 inputs. Then the 3rd input will be the second input of the second NAND or NOR gate. In this way, we can perform much more logical operations.

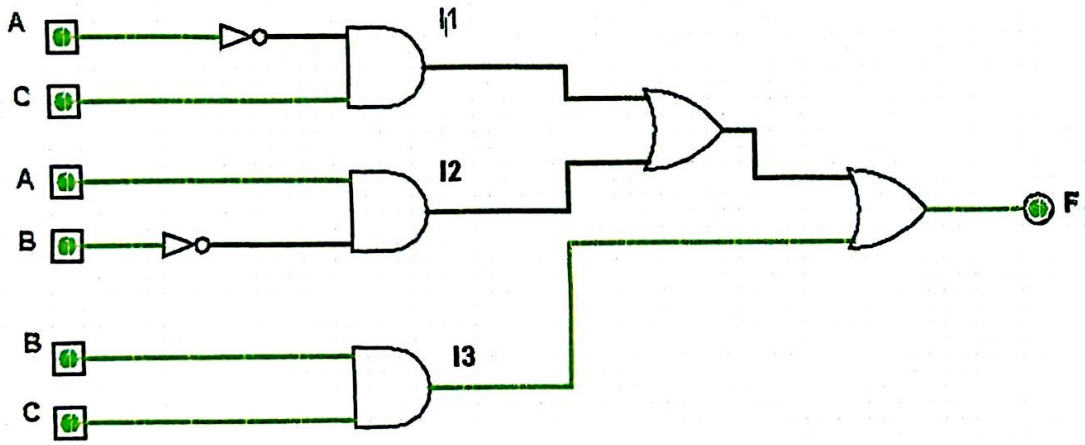
Discussion:

From this experiment, we could the basic logic gates and it's behaviour such as – AND, OR, NAND, NOR, XOR, NOT. We could understand how their truth tables work. How different logic gates have different ICs, how to put IC in the breadboard, how to connect inputs and outputs – we could learn that. We also observed how different input combinations resulted in different outputs.

By using truth table, we can check the outputs theoritically. But sometimes, the simulated results were not matching the theoritical values. We faced this issue when the IC was not set right, setting the IC ~~can~~ in wrong direction can get us wrong results. Other

reasons of not getting proper outputs can be— loose connections, inconsistencies in the trainer broad's voltage supply. Placing the notch of the IC in right direction, ensuring secure connection and proper voltage supply could improve the reability of our results. This way the theoritical and simulated results will have more similarity.

1.Tamanna Islam (2233172642)
2.Nudrat Rahman Tushin (2231058642)
3.Md Sadid Ahmed Safin (2233394642)



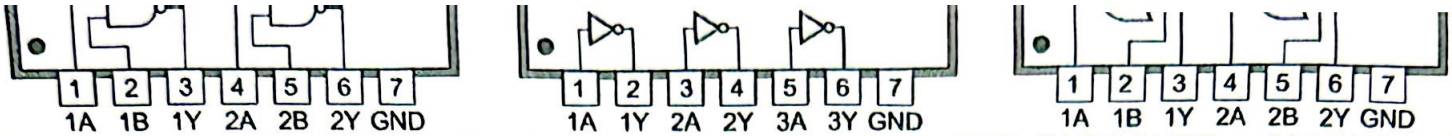


Figure C.2 Schematic of (a) 7400 NAND, 7408 AND, 7432 OR and 7486 XOR, (b) 7404 NOT, and (c) 7402 NOR ICs

D. Procedure

D.1. Introduction to Basic Logic Gates

1. Place the 7408 AND IC on the breadboard.
2. Connect the V_{CC} and GND pins of the IC to the +5 V and GND ports of the trainer board respectively.
3. Label the pin numbers of the inputs and output of the gate in Figure D.1, using the pin configurations in Figure C.2.

Figure D.1 Pin configurations of gates in ICs

4. Connect the gate.
   a. Connect each input of the gate to a toggle switch on the trainer board.
   b. Connect the output of the gate to an LED on the trainer board.

5. Apply all combinations of inputs by turning the toggle switches on (1) and off (0), and record if the LED is on (1) or off (0) as the output of the gate. Record your results in Table D.1

AB	$F = A \cdot B$	$F = A + B$	F=(A.B)'	$F = A \oplus B$	F=(A+B)'
00	0	0	1	0	1
01	0	1	1	1	0
10	0	1	1	1	0
11	1	1	0	0	0

A	$F = A'$
0	1
1	0

Table D.1 Truth table of logic gates

Repeat steps 1-5 for the NOT and NOR ICs.

Extension of Inputs of AND and OR

Complete the truth table for the 3-input AND gate in Table D.2.

A B C	F = ABC	F = A + B + C
000	0	0
001	0	1
010	0	1
011	0	1
100	0	1
101	0	1
110	0	1
111	1	1

Table D.2 Truth table of 3-input AND and OR

2. Using the associative law given in Table C.3, express the 3-input function using two 2-input AND gates in Table D.3.

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

Table D.3 Associative law

3. Label the pin numbers in Figure D.2, using the pin configurations in Figure C.2.

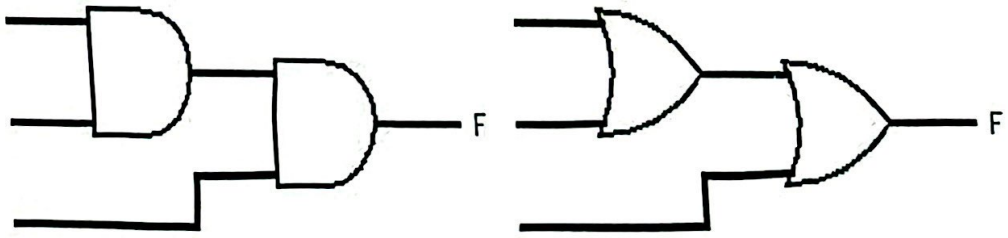


Figure D.2 Extension of inputs of AND and OR gates

4. Connect the circuit.
5. Connect the output to an LED and verify it using the truth table.
6. Repeat steps 1-5 for the 3-input OR gate.

D.3. Implementation of Boolean Functions

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

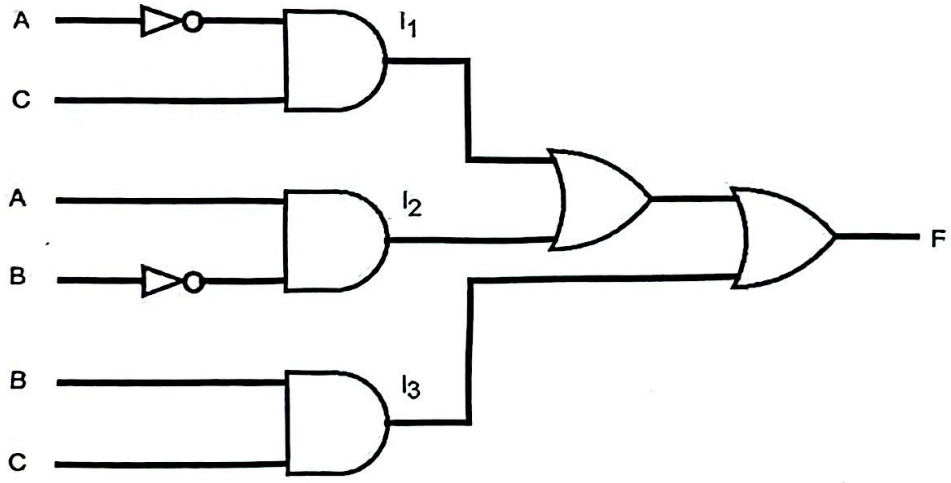
1. Complete the truth table for the *implicants* $I_1 = A'C$, $I_2 = AB'$ and $I_3 = BC$ in Table D.4.

plete the truth table for the function F in Table D.4.

ABC	$I_1 = A'C$	$I_2 = AB'$	$I_3 = BC$	$F = I_1 + I_2 + I_3$
000	0	0	0	0
001	1	0	0	1
010	0	0	0	0
011	1	0	1	1
100	0	1	0	1
101	0	1	0	1
110	0	0	0	0
111	0	0	1	1

Table D.4 Truth table of logic gates

Label the pin numbers for the NOT, AND and OR gates of the function Fin Figure D.3, using the pin configurations in Figure C.2.



10-09-24

Figure D.3 Circuit diagram for the Boolean function F

Connect the input A to a NOT gate using the pins assigned in step 3 and check the output via an LED.
Wire upimplicant I_1.
Connect the output of $I_1$to an LED and verify it using the truth table.
Connect the input B to a NOT gate using the pins assigned in step 3 and check the output via an LED.
Wire up implicant I_2.
Connect the output of I_2 to an LED and verify it using the truth table.
Wire up implicant I_3.
Connect the output of I_3 to an LED and verify it using the truth table.
Connect the outputs of the three implicants as inputs to the OR gates (using the associative law).
Connect the output F to an LED and verify the function using the truth table.