AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Experiment Title:	Deriving logic equations and truth table from a given statement or expression and construction of combinational circuits		
Experiment No:	02	**Date of Submission:** 04\02\24	
Course Title:	DIGITAL LOGIC AND CIRCUITS LAB		
Course Code: 00647		**Section:** R	
Semester:	Spring 2023-2024	**Course Teacher:**	MD. ASHIQUZZAMAN

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

** Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 03

No	Name	ID	Program	Signature
1	Sakib, Md. Showkat Islam	22-49858-3	CSE	
2	Bhuiyan, Takbir Zaman	22-49852-3	CSE	
3	Kazi Imtiaz	22-49857-3	CSE	
4	Tutul Majumder	23-51364-1	CSE	

Faculty use only

FACULTY COMMENTS		
	Marks Obtained	
	Total Marks	

American International University- Bangladesh

Department of Electrical and Electronic Engineering

EEE3102:Digital Logic & Circuits Laboratory

Title: Deriving logic equations and truth table from a given statement or expression andconstruction of combinational circuit.

Abstract:

This experiment is designed to-

1. Help students implement the logic circuits derived from a given statement in the breadboard using gate ICs and observe whether the output verifies the truth table of the given logic statement or not.

2. Perform relevant theoretical work by deriving the logic circuit and truth table from the given logic equation/statement and get familiarized with Boolean algebra and De Morgan’s law.

3. Simplify the logic expressions with K-Map and verify accuracy by breadboard implementation.

Introduction:

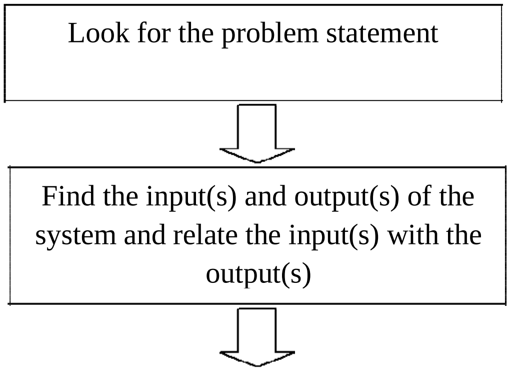
From any given logic statement, it is possible to construct a digital logic circuit. The first step in this process is to construct a truth table and then determine a standard SOP (sum of products) or POS (product of sums). At the same time, it is also possible to derive a logic expression from a given combinational circuit diagram by observing the individual logic operations performed in the circuit and matching them with their corresponding logic gates. Expressions are simplified using Boolean algebra and De Morgan's law or K-Map to reduce the number of gates used. Then the circuit is implemented in the breadboard using gate ICs and observed whether the output verifies the truth table of the given statement.

This experiment shows the students a practical verification of deriving logic equations and truth table from combinational circuits. Knowing how to derive logic equations and truth table from combinational circuits helps a person with detecting the output logic expressions from any unknown logic circuit.

Theory and Methodology:

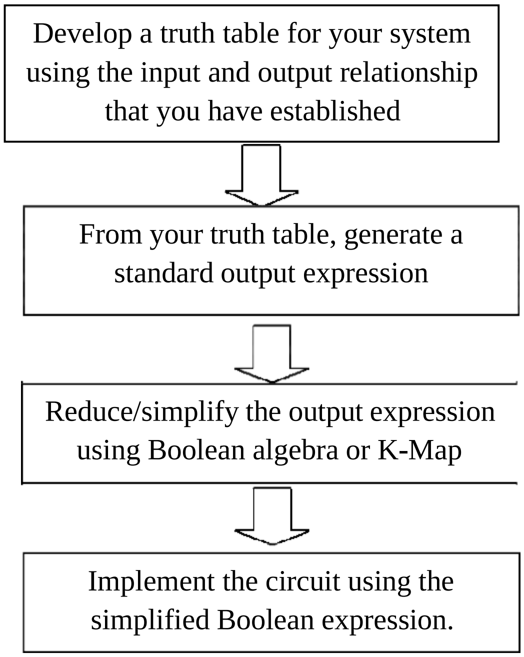
Combinational circuits are built with logic gates and other components. It does not include any values to be taken from a previous state of the circuit. Designing such a combinational digital system requires use of one of the following methods:

1. If a problem statement is given, the following steps will help designing the system



Look for the problem statement

Find the input(s) and output(s) of the system and relate the input(s) with the output(s)



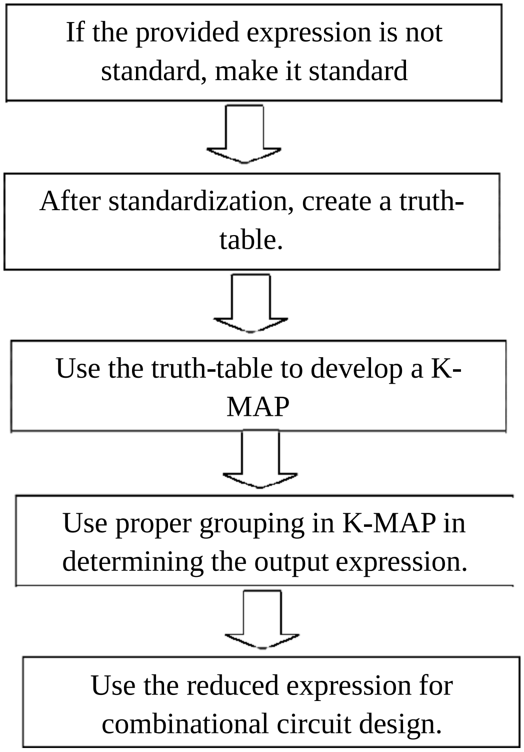
Develop a truth table for your system using the input and output relationship that you have established

From your truth table, generate a standard output expression

Reduce/simplify the output expression using Boolean algebra or K-Map

Implement the circuit using the simplified Boolean expression.

2. Or if an expression is given, the following steps will help in designing the system



If the provided expression is not standard, make it standard

After standardization, create a truth-table.

Use the truth-table to develop a K-MAP

Use proper grouping in K-MAP in determining the output expression.

Use the reduced expression for combinational circuit design.

Some useful definitions related to these procedures are given below:

Boolean algebra: In Boolean algebra, a variable is a symbol used to represent an action, a condition, or data. A single variable can only have a value of 1 or 0.

1. **Variable:** A symbol used to represent a logical quantity that can have a value of 1 or 0, usually designated by an italic letter.
2. **Complement:** The inverse or opposite of a number. In Boolean algebra, the inverse function, expressed with a bar over the variable.
3. **Sum term:** The Boolean sum of two or more literals equivalent to an OR operation
4. **Product term:** The Boolean product of two or more literals equivalent to an AND operation.
5. **Sum of Products (SOP):**

When two or more product terms are summed by boolean addition, the resulting expression is a sum of product. Ex.

Implementing an SOP expression simply requires ORing the outputs of two or more AND gates. A product term is produced by an AND operation, and the sum (addition) of two or more product terms is produced by an OR operation. Therefore, an SOP expression can be implemented by AND-OR logic in which the outputs of a number (equal to the number of product terms in the expression) of AND gates connect to the inputs of an OR gate.

A standard SOP expression is one in which all the variables in the domain appear in each product term. Ex.

Standard SOP expressions are important in constructing truth-tables and in Karnaugh map simplification method.

The SOP expression is equal to 1 only if one or more of the product terms in the expression is equal to 1.

6. **Product of Sums (POS):**

When two or more sum terms are multiplied, the resulting expression is a product of sums (POS). Ex.

Implementing a POS expression simply requires ANDing the outputs of two or more OR gates. A sum term is produced by an OR operation, and the product of two or more sum terms is produced by an AND operation. Therefore, a POS expression can be implemented by logic in which the outputs of a number (equal to the number of sum terms in the expression) of OR gates connect to the inputs of an AND gate.

A standard POS expression is one in which all the variables in the domain appear in each sum term in the expression. Ex.

A POS expression is equal to 0 only if one or more of the sum terms in the expression is equal to 0.

7. **Karnaugh Map:**

A Karnaugh map provides a systematic method for simplifying Boolean expressions and, if properly used, will produce the simplest SOP or POS expression possible le, known as the minimum expression.

A Karnaugh map is similar to a truth table because it presents all of the possible values of input variables and the resulting output of each valued. Instead of being organized into columns and rows like truth table, the Karnaugh map is an array of cells in which each cell presents binary value of the input variables. The cells are arranged in a way so that the simplification of a given expression is simply a matter of properly grouping the cells. Karnaugh maps can be used for expressions with two, three, four and five variables. The number of cells in a Karnaugh map is equal to the total number of possible input variable combinations as is the number of rows in a truth table.

Problem1. A Building has 4 floors which share the same water tank for water supply. In order to start the motor, each floor has a designated switch- Ground Floor with switch A, 1st Floor with switch B, 2nd Floor with switch C and 3rd Floor with switch D. The motor starts if someone presses the switch from the 3rd floor or from both ground and 2nd floor or from 1st and 2nd floor. Your job is to design the system.

Problem2. For the expression ***(AB+AC)'+ A'B'C,*** find the truth-table and the logic gate diagram, reduced expression using K-MAP.

Apparatus:

SL	Apparatus	Picture	Quantity	Remarks
1.	Digital trainer board.		1	Good condition.
2.	Integrated Circuits (ICs) 7400 7402 7404 7408 7432 7486		 1 1 1 1 1 1	Good Condition
3.	Power Supply		1	Good condition
4.	Connecting wires		15	Good condition

Precaution: The IC contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages. For proper operation, V_{in} and V_{out} should be constrained to the range GND to V_{CC}. Connect the ICs according to their pin configuration carefully and use connecting the wires with the ICs to make sure that they are firmly connected. Check whether all the data switches and output showing LEDs are working.

Experimental Procedure:

Problem1:

a) We have drawn a truth table from problem-1where we represent the output with Y.

Ground Floor	1st Floor	2nd Floor	3rd floor	Output Motor Start
A	**B**	**C**	**D**	**F**
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b) Using the truth table outputs to form standard SOP expressions.
SOP expressions:
Y=(A’B’C’D)+(A’B’CD)+(A’BC’D)+(A’BCD’)+(A’BCD)+(AB’C’D)+(AB’CD’)+(AB’CD)+(ABC’D)+(ABCD’)+(ABCD)

c) To Minimize the SOP expression using Boolean algebra and K-Map. Perform hardwareimplementation of the circuit and compare with your truth table output.

CD AB		C’D’	C’D	CD	CD’
		00	01	11	10
A’B’	00	0 0	1 1	3 1	2 0
A’B	01	4 0	5 1	7 1	6 1
AB	11	12 0	13 1	15 1	14 1
AB’	10	8 0	9 1	11 1	10 1

Minimized SOP expression: Y= D+BC+AC

Hardware Implementation for problem-1:

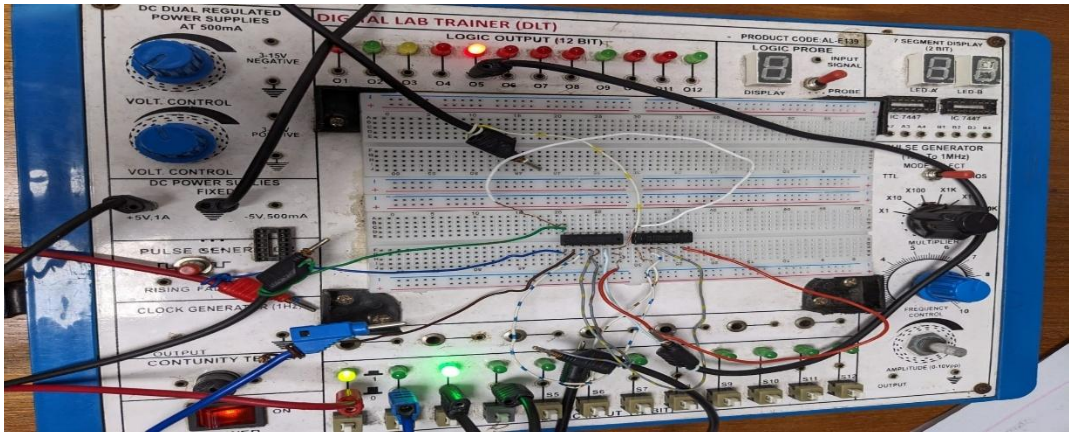


Fig:1-Hardware Implementation

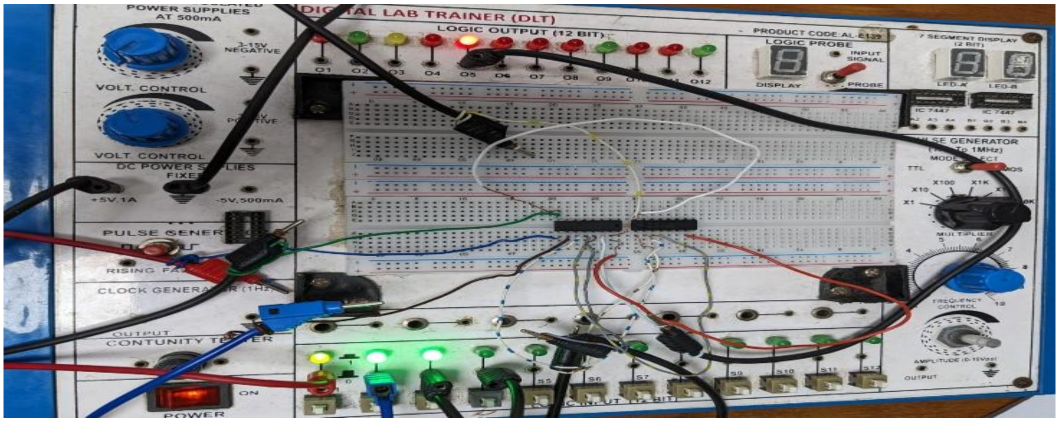


Fig:2- Hardware Implementation

AB C		C’	C
		0	1
A’B’	00	0 1	1 1
A’B	01	2 1	3 1
AB	11	6 0	7 0
AB’	10	4 1	5 0

Problem2:

a) Draw a step-by-step truth table to represent the outputs at each gate and then thefinal output at Y

b) We have used Y as output to form standard SOP expression.

Y=(A’B’C’)+(A’B’C)+(A’BC’) +(A’BC) +(AB’C’)

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

C) Minimized the SOP expression using Boolean algebra and K-Map. Perform hardwareimplementation of the circuit and compare with your truth table output.

The minimized SOP expression: Y=A’+AB’C’

Hardware Implementation for Problem-2

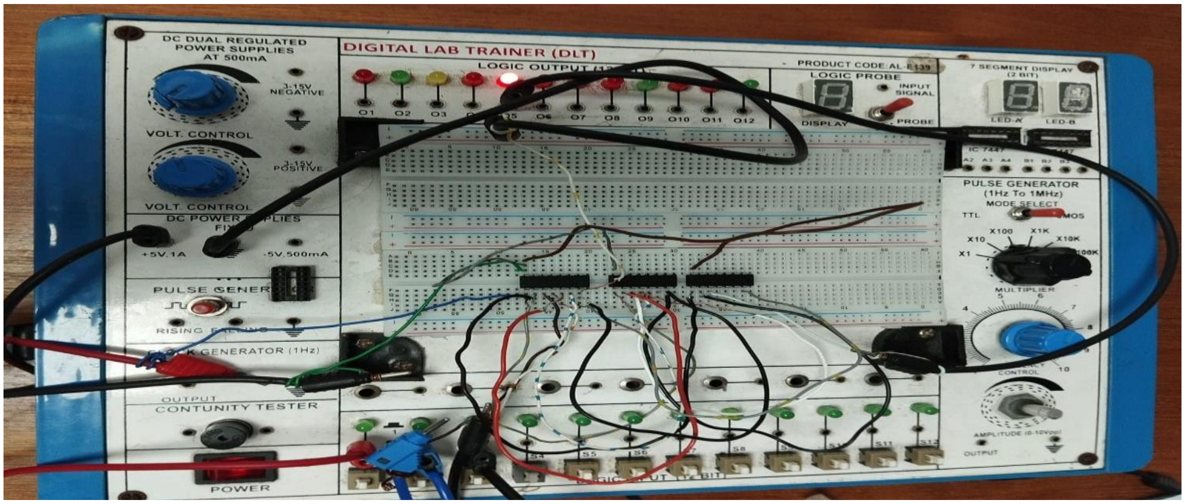

DIGITAL LAB TRAINER (DLT)
DC DUAL REGULATED POWER SUPPLIES AT 500mA
LOGIC OUTPUT (12 BIT)
LOGIC PROBE
7 SEGMENT DISPLAY (2 BIT)
PULSE GENERATOR (1Hz To 1MHz)
CONTUNITY TESTER
POWER

Fig:3- Implementating for **Y=A’+AB’C’**

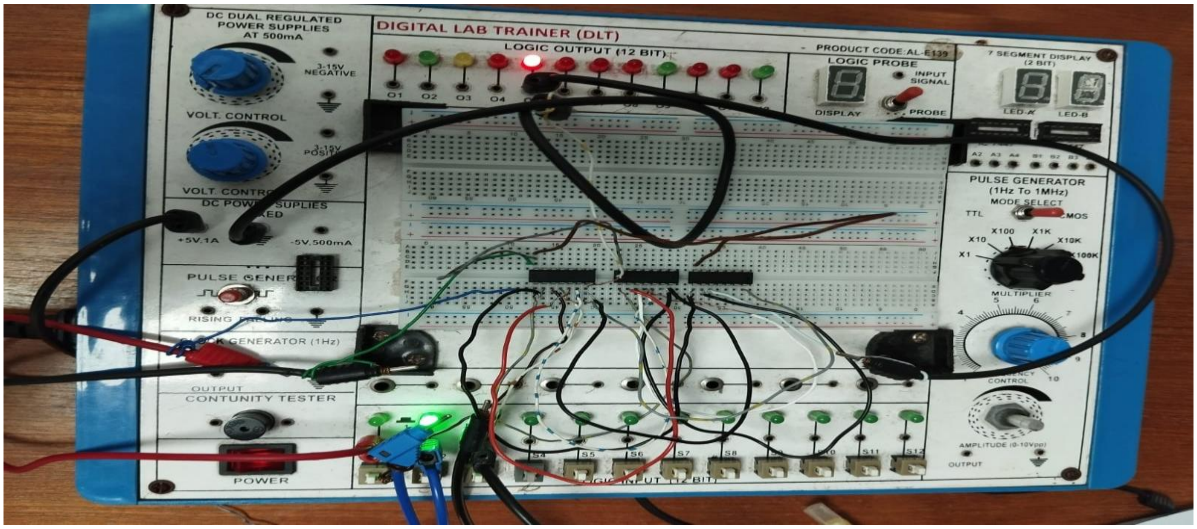

DIGITAL LAB TRAINER (DLT)
DC DUAL REGULATED POWER SUPPLIES AT 500mA
LOGIC OUTPUT (12 BIT)
LOGIC PROBE
7 SEGMENT DISPLAY (2 BIT)
PULSE GENERATOR (1Hz To 1MHz)
CONTUNITY TESTER
POWER

Fig:4- Hardware Implementating for **Y=A’+AB’C’**

Simulation and Measurement:

For Problem-1

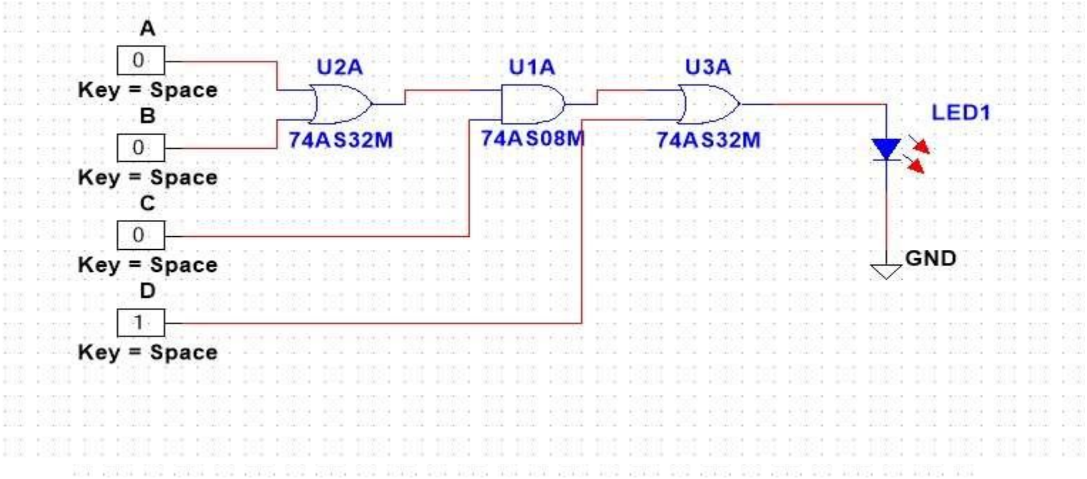


Fig:5-Simulation diagram for fig-1

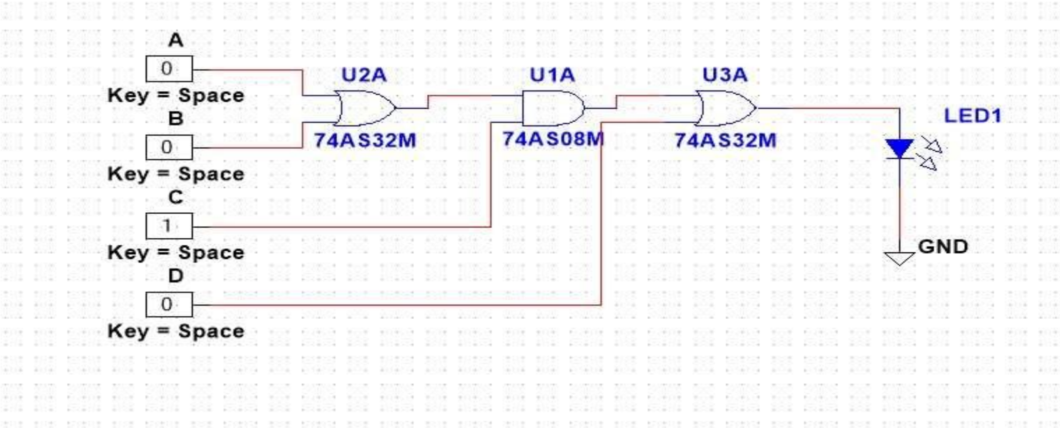


Fig:6-Simulation diagram for fig-2

For Problem-02

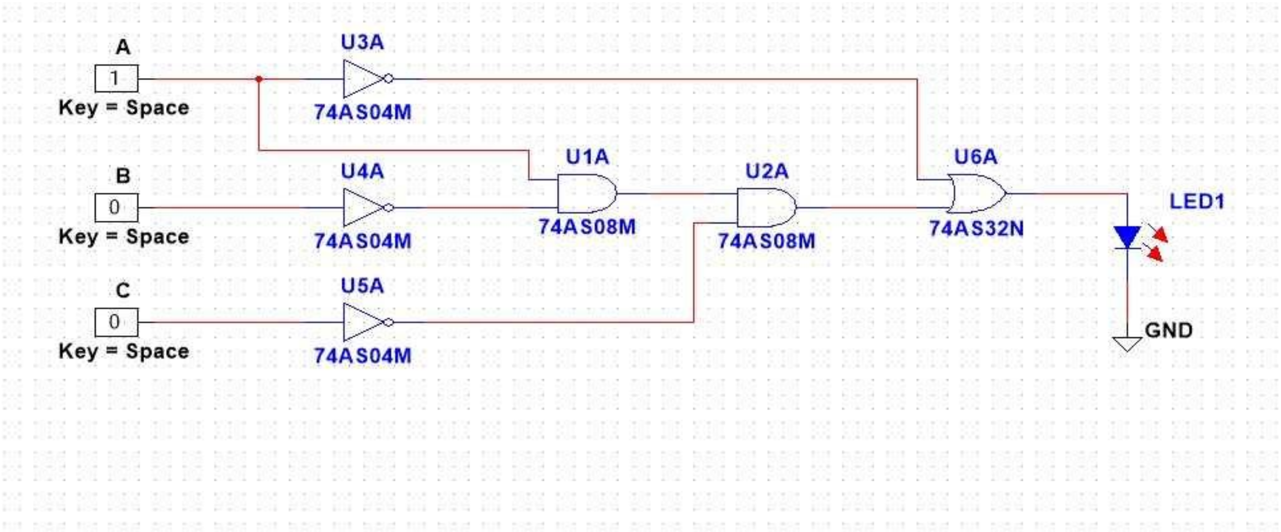


Fig:7-Simulation diagram from fig 5

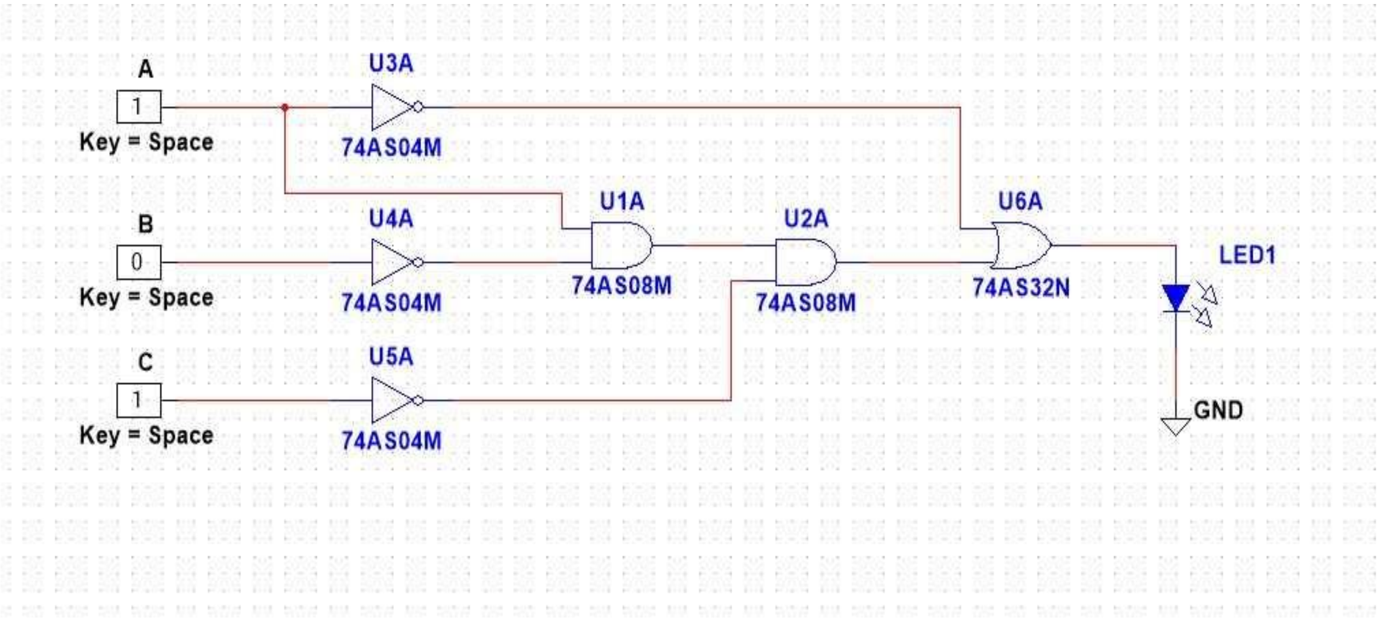


Fig:8-Simulation diagram from fig-6

Discussion:

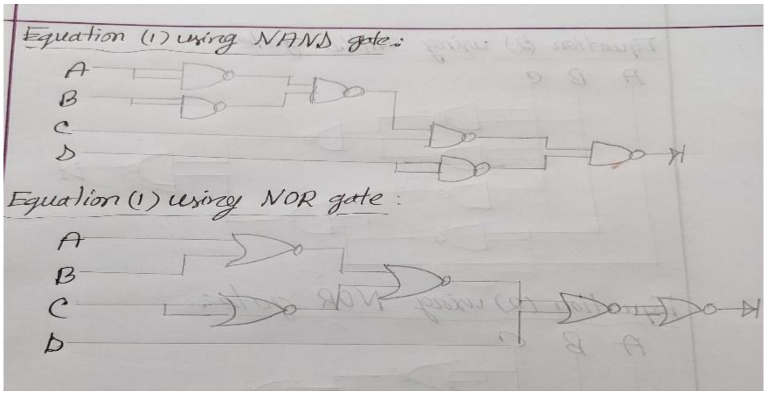
We have derived the logical expression from the provided statement and translated it into a hardware implementation on the breadboard & software implementation on Ni Multisim version-14.3 . The entire process unfolded seamlessly, devoid of any instrument errors, ensuring a flawlessly executed experiment. Subsequently, we verified the truth table, meticulously scrutinizing every detail to confirm the precise functionality of our constructed logic gate.

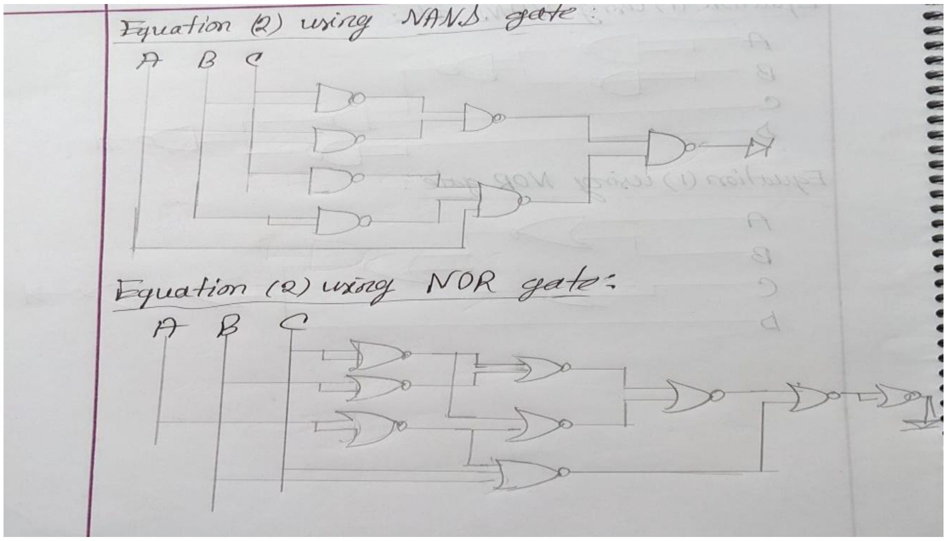
Conclusion:

The lab experiment facilitated practical experience in deriving logic equations, constructing combinational circuits, and verifying their accuracy using truth tables. Through the dissection of statements and expressions, a deeper understanding of logic gate operations was achieved, culminating in successful implementation on a breadboard. The seamless execution of the experiment affirmed a grasp of digital electronics concepts, furnishing a robust foundation for future endeavors in the field of digital logic design.

Answer to the question

1.Construct the derived equations (i) and (ii), using Universal gates (both NAND andNOR).





2.Develop the truth table for a certain three-input logic circuit with the output expression $Y=ABC+(AB)'C+A'BC+AB'C+A(B'+C)$.

Ans:

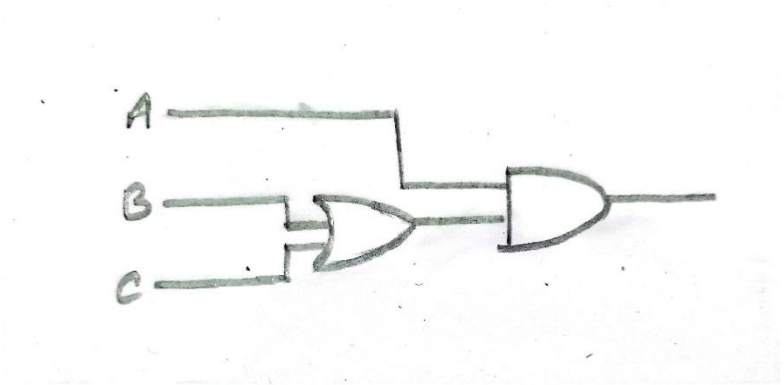
A	B	C	AB	(AB)'	(AB)'C	A'	A'BC	AB'C	A(B'+C)	ABC	Y
0	0	0	0	1	0	1	0	0	0	0	0
0	0	1	0	1	1	1	0	0	0	0	1
0	1	0	0	1	0	1	0	0	0	0	0
0	1	1	0	1	1	1	1	0	0	0	1
1	0	0	0	1	0	0	0	0	1	0	1
1	0	1	0	1	1	0	0	1	1	0	1
1	1	0	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	1	1	1

3.Implement the following logic expressions with logic gates $Y=ABC+AB+AC$

Ans: $Y=AB(C+1)+AC$

$=AB+AC$

$=A(B+C)$



References:

[1]. Thomas L. Floyd, “Digital Fundamentals”, available Edition, Prentice Hall International Inc.

[2]. www.electronics-tutorials.ws

[3].American International University–Bangladesh (AIUB) Digital Logic And Circuits Lab Manual.