AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH
Faculty of Engineering

Laboratory Report Cover Sheet

Please submit all reports to your subject supervisor or the office of the concerned faculty.

Laboratory Title: Deriving logic equations and truth table from a given statement or expression and construction of combinational circuits

Experiment Number: 02 **Due Date:** 11/10/2022 **Semester:** Fall, 22-23
Subject Code: EEE3102 **Subject Name:** DLC Lab **Section:** E
Course Instructor: Abrar Fahim Liaf **Degree Program:** BSc CSE/EEE

Declaration and Statement of Authorship:

1. I/we hold a copy of this report, which can be produced if the original is lost/ damaged.
2. This report 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 report has been written for me/us by any other person except where such collaboration has been authorized by the lecturer/teacher concerned and is clearly acknowledged in the report.

I/we understand that

4. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. 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 the material used is not appropriately cited.
5. Enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy your work.

Group Number: 02	Group Submission

Group Members:

No.	Student's Name	Student ID	Comments
01	Hossen, Md Al Amin	19-39535-1	
02	Amin, Abdullah Al	19-39502-1	
03	Karim, Rezaul	19-39521-1	
04	Singho, Argho Proshad	20-42906-1	
05	Sarker, Dipunkor	18-37976-2	

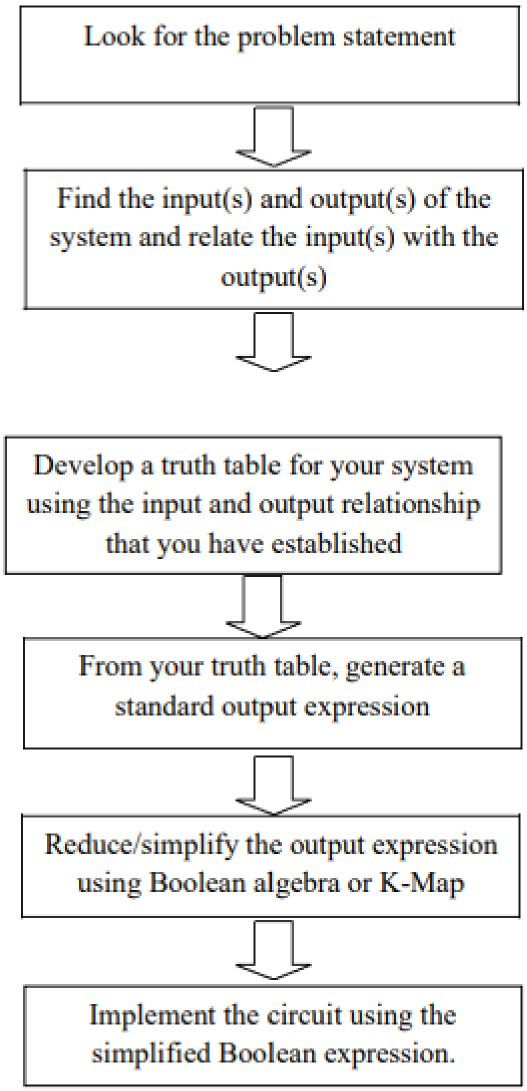
Abstract:

This experiment is designed to 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. Also 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. And finally, simplify the logic expressions with K-Map and verify accuracy by breadboard implementation.

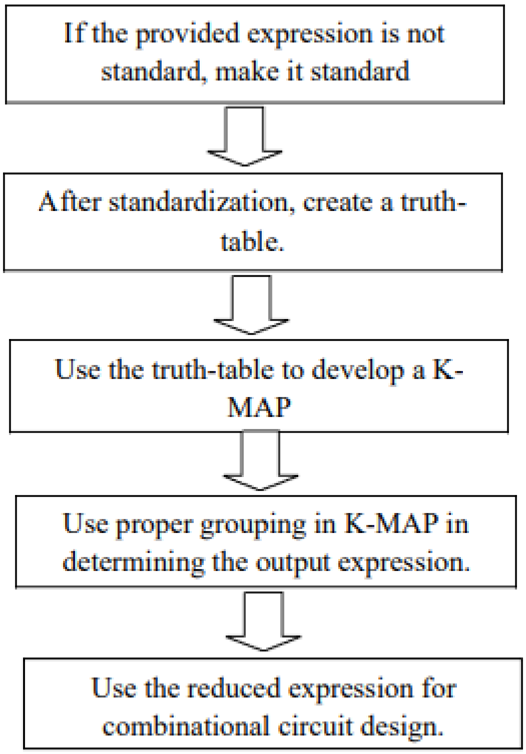
Theory:

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



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



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.

<u>Apparatus:</u>

1. Digital trainer board
2. IC 7432:1 pcs
3. IC 7408:1 pcs
4. IC 7404:2 pcs
5. IC 7402:1 pcs
6. IC 7400:1 pcs
7. IC 7486:1 pcs
8. Connecting wires

<u>Experimental Procedure:</u>

Problem1:

a) A truth table has been drawn to represent the output Y.
b) Using the truth table outputs, standard SOP and POS expressions were formed.

c) Minimized the SOP expression using Boolean algebra and K-Map. After that, hardware implementation of the circuit was performed and compared with the truth table output.

Simulation:

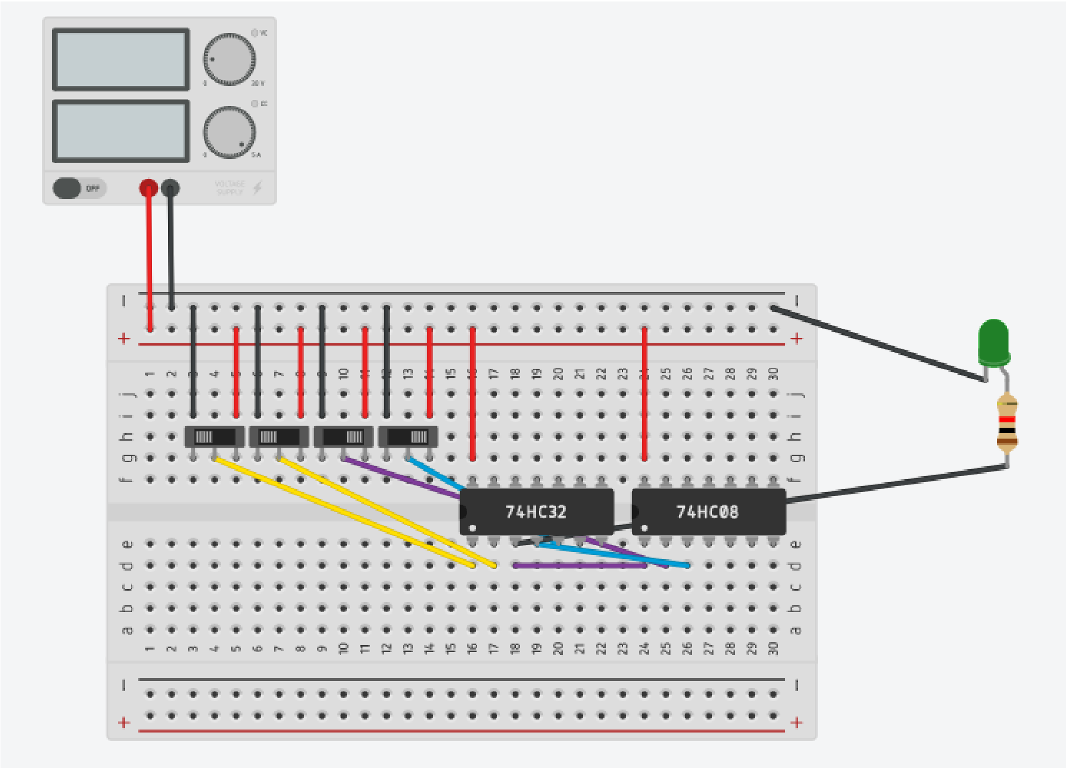


Fig-01: Simulation setup (Tinkercad)

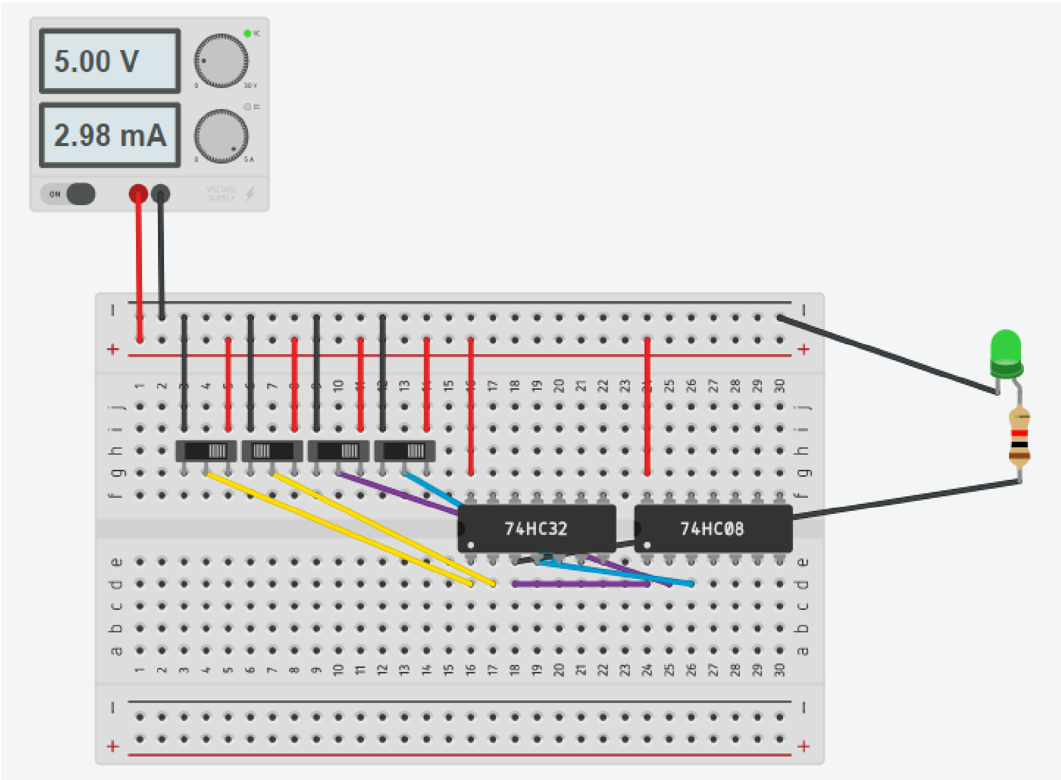


Fig-02: Simulation result (Tinkercad)

Experiment Result:

Consider,
Ground floor has switch A,
1st floor has switch B,
2nd floor has switch C and
3rd floor has D

Truth table for representing the output Y:

Ground Floor (A)	1st Floor (B)	2nd Floor (C)	3rd Floor (D)	Output (Y)
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

Table 01: Truth Table.

Standard SOP:

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

Standard POS:

Y = (A’+B’+C’+D). (A’+B’+C+D). (A’+B+C’+D'). (A’+B+C+D’). (A’+B+C+D). (A+B’+C’+D). (A+B’+C+D’). (A+B’+C+D). (A+B+C’+D). (A+B+C+D’). (A+B+C+D)

SOP Minimization:

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

= A’B’CD + A’BCD + AB’CD + ABC’D + ABCD

= A’B’CD(1+B’) + AB’C’D(C’+B’) + ABCD … … (i)

K-MAP:

CD\AB	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	1

Table 02: K-MAP Table.

Using K-MAP,

Output, Y = D + BC + AC

= D+C(A+B) … … (ii)

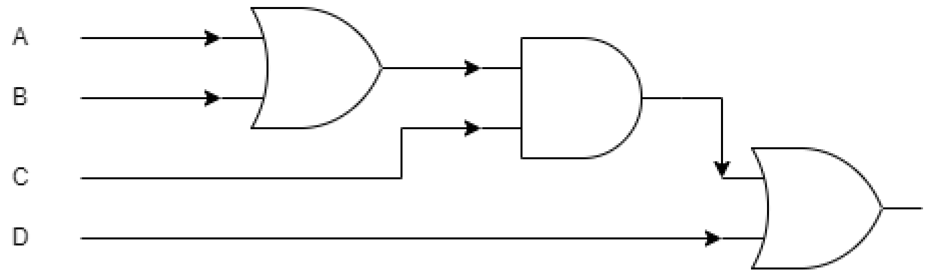


Fig-03: Circuit Diagram of Y = D + BC + AC.

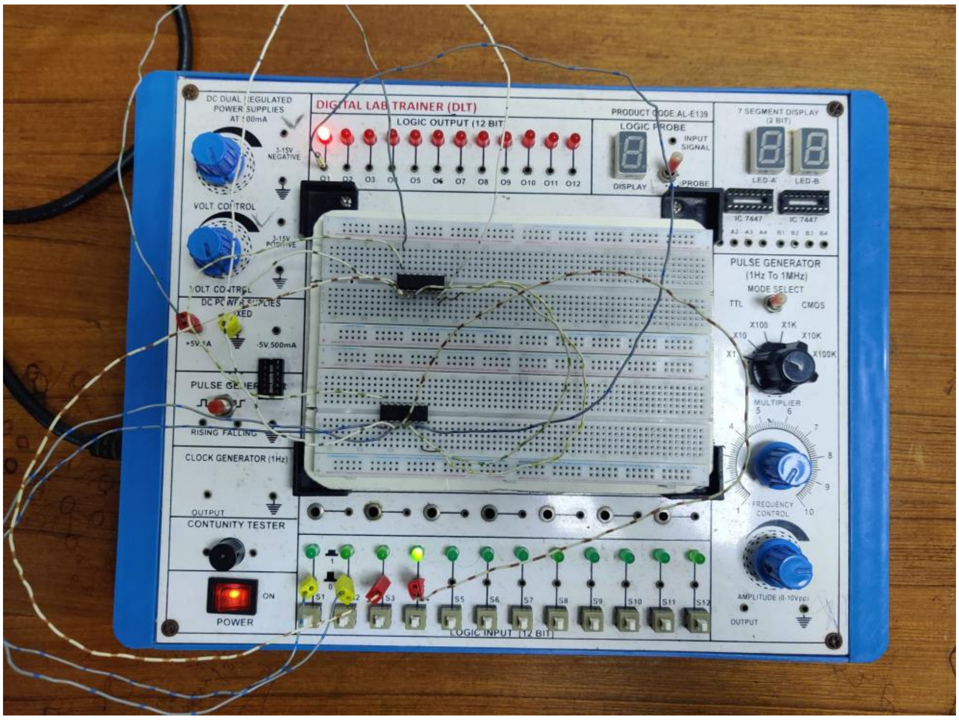


Fig-04: Simulation Result (Hardware).

Report Questions:

1. Construct the derived equation (ii), using Universal gates (both NAND and NOR).

Here,
Derived equation, Y = D+C(A+B).

Construct using NAND Gate:

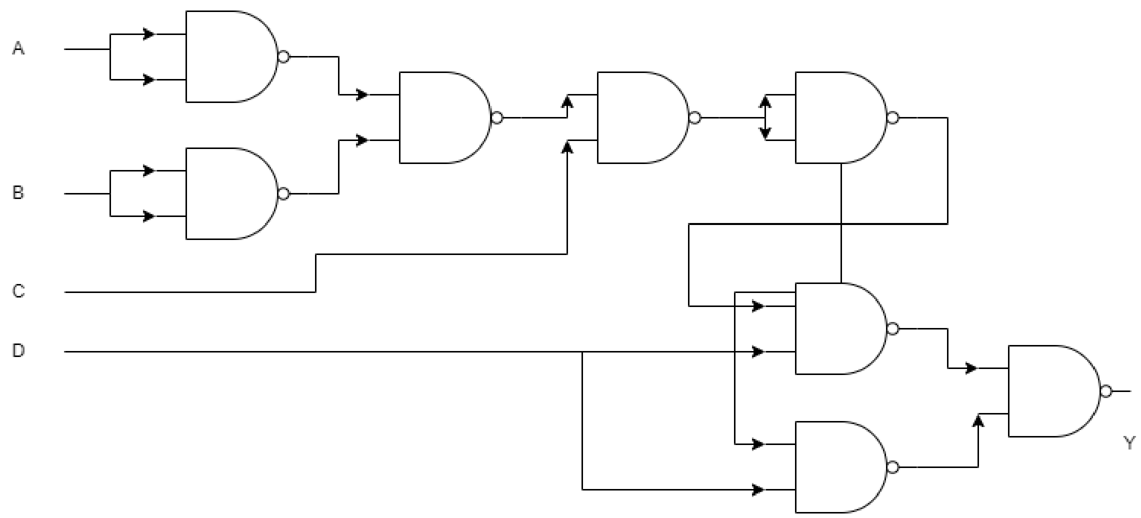


Fig-05: Circuit Diagram using NAND Gate.

Construct using NOR Gate:

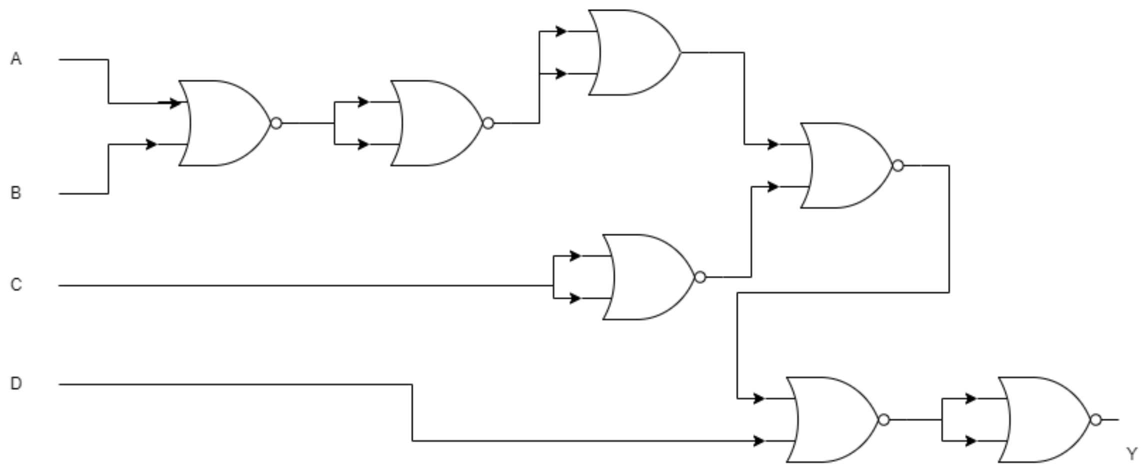


Fig-06: Circuit Diagram using NOR Gate.

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).

Here,

Y=ABC+(AB)’C+A’BC+AB’C+A(B’+C).

=ABC+A’C+B’C+A’BC+AB’C+AB’+AC

=BC(A+A’) +AB’(C+1) +C(A+A’) +BC’

=BC+AB’+C+B’C

=C+AB’

Truth table for representing the output Y:

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

Table-03: Truth Table.

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

Here,
Y = ABC + AB + AC
= AB(C+1) + AC
= AB + AC
= A(B+C)

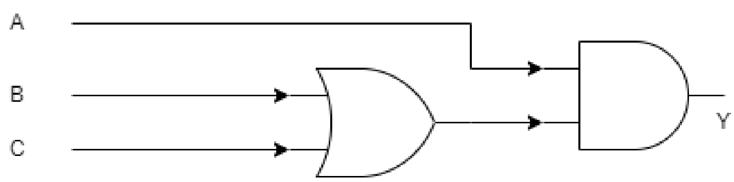


Fig-07: Circuit Diagram of Y = A(B+C).

<u>Discussion and conclusions:</u>

The primary goal of this experiment was to implement the logic circuits derived from a given statement in the breadboard

using gate ICs and observing whether the output verifies the truth table of the given logic statement or not. In this experiment, we performed 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. Breadboard, LEDs, Resistors, and connecting wires were used to design the systems. While experimenting, there were some problems that we faced due to the loose connection of ICs on the breadboard. The experiment covered a couple of different control systems as well as the usage of simulation tools. Hence, providing a proper interpretation.

We design the logic circuit & implement the system. And lastly, we have implemented a logical circuit based on logic by using the online simulation tool 'Tinker cad'. The simulation was done successfully and the system was run properly. By doing this experiment, we have learned to create logical equations step by step based on any problem statement and the implementation of logical circuits. So, we can easily say that our goal of the experiment is fulfilled.

References

[1] T. L. Floyd, Digital Fundamentals 11th Edition, Pentice Hall International Inc., 2014.

[2] EEE3102:Digital Logic & Circuits Laboratory, "Deriving logic equations and truth table from a given statement," Department of Electrical and Electronic Engineering, 2022.