North South University

Department of Electrical & Computer Engineering

Course Code: CSE 231L

Course Title: Digital Logic Design

Faculty: DR. M ABDUR RAZZAK

Project Proposal of "Design a Combinational Logic Circuit to display "IUBFALL" in a Seven Segment Display"

Date of Submission: 07/12/2022

Section: 07

Group Number: 02

Submitted To: MD. AL – AMIN BHUIYAN

Submitted By		
S.N.	Student Name	ID
9	Al Zammee	2021648642
12	Umme Suraia Haque Setu	2031278642
18	Humayra Rahman Nipa	2121128642
24	Saif Mohammed	2121913642
35	Afra Saiara Saima	2132242642

<u>**CSE-231 Lab Project Proposal**</u>

2. Executive Summary

This project is about displaying English letters "IUBFALL" with the help of seven segment device. In this project, we are going to use a cathode seven segment display by using combinational and sequential circuit. For the combinational circuit we built SOP, POS, NAND, NOR and MUX Expression form to display the letters. We created a Truth Table to find out what pins need to be active for each letter. After completing that, we implemented Boolean function by using K-Map and then design our combinational circuits accordingly. We applied the simplified expressions for each circuit and figured out which circuit design was the most efficient in terms of cost and power consumption.

3. Background

The need of the project is to fulfill the main objectives of the course. To ensure it we need to use Boolean logic expression, K-Map, logic design etc. To develop project, we need to make BCD Truth tables for letters "IUBFALL".
Then from that table the SOP, POS, NAND, NOR and MUX expression will develop.

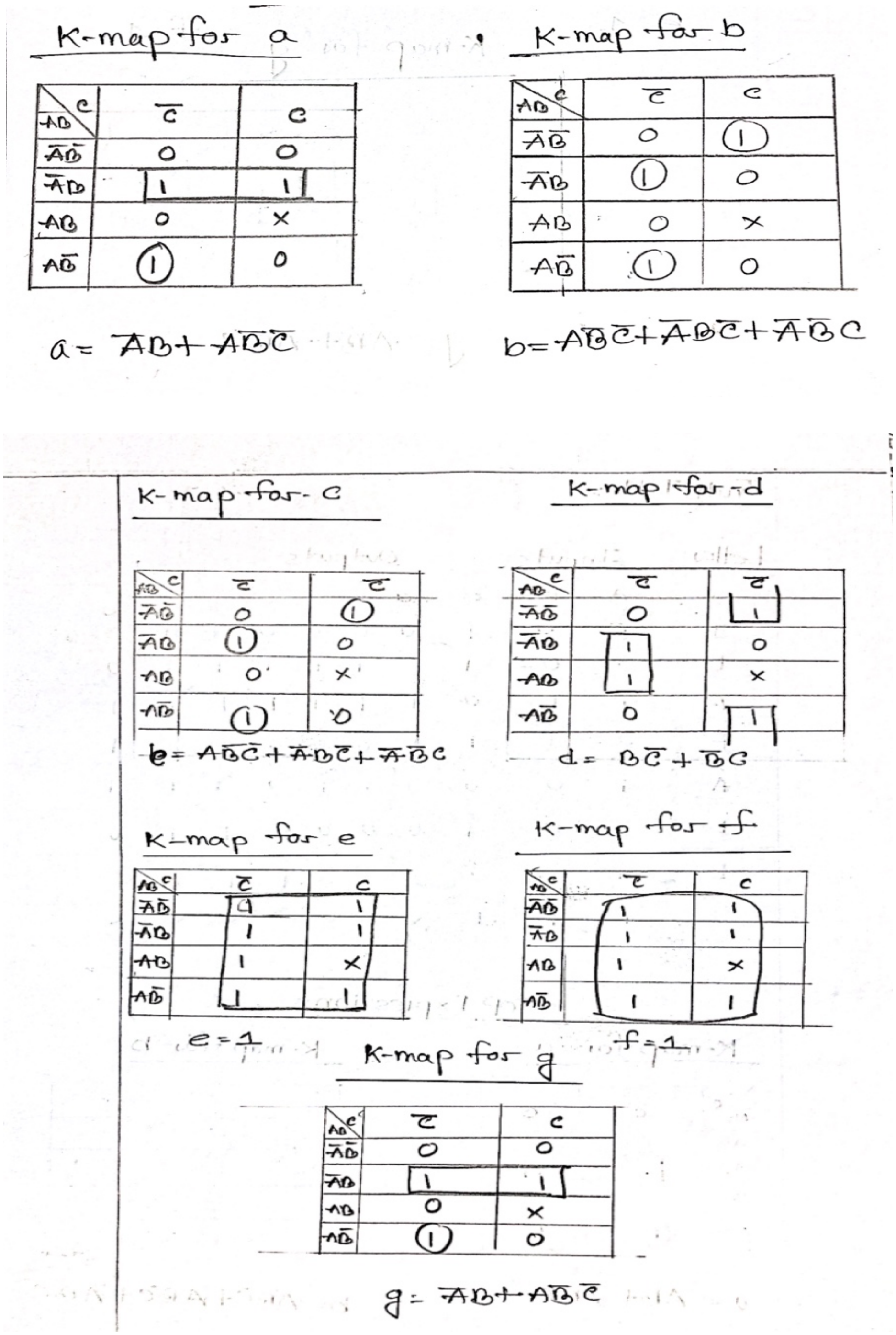
- Truth Table for "IUBFALL"

Letters	Inputs			Outputs						
	A	B	C	a	b	c	d	e	f	g
I	0	0	0	0	0	0	0	1	1	0
U	0	0	1	0	1	1	1	1	1	0
B	0	1	0	1	1	1	1	1	1	1
F	0	1	1	1	0	0	0	1	1	1
A	1	0	0	1	1	1	0	1	1	1
L	1	0	1	0	0	0	1	1	1	0
L	1	1	0	0	0	0	1	1	1	0
	1	1	1	X	X	X	X	X	X	X

- **SOP Expressions**

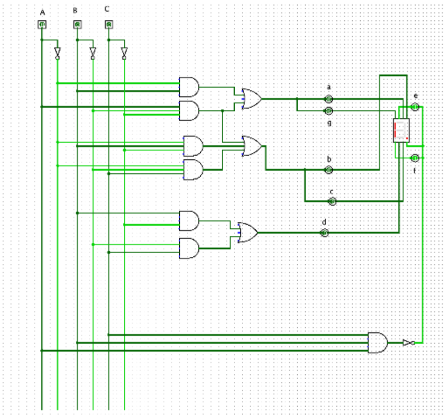
Sum of Product is the abbreviated form of SOP. Sum of product form is a form of expression in Boolean algebra in which different product terms of inputs are being summed together. We know that in SOP, it evaluates with 1 for each combination of x and y. ‘1’ means the variable is ‘Not complemented’ and ‘0’ means the variable is ‘Complemented’. Following the method of SOP form we drew the kmaps for our project and got the following equations:

The 3-variable Karnaugh map is an array of eight cells. For our project Kmap Using SOP is given below:

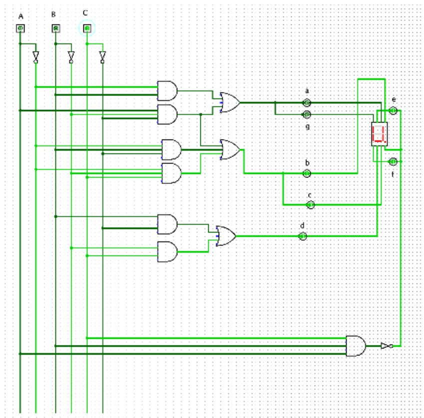


K-Map for SOP Expression

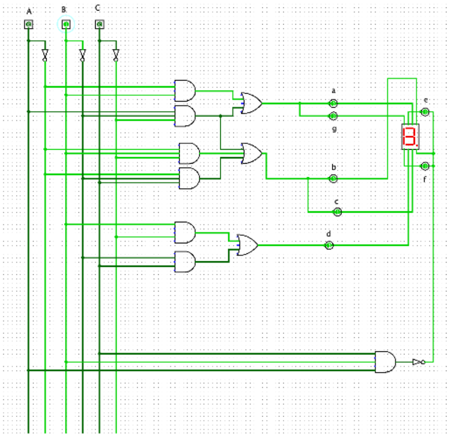
Circuit Diagrams (SOP)



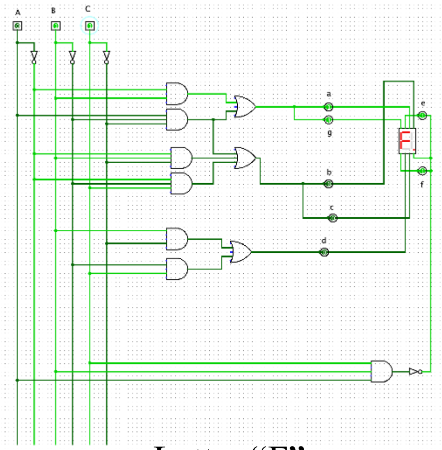
Letter “I”



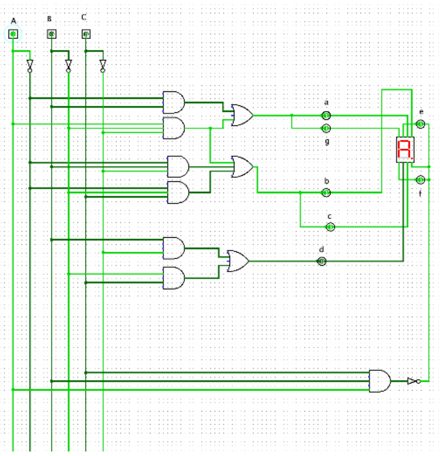
Letter “U”



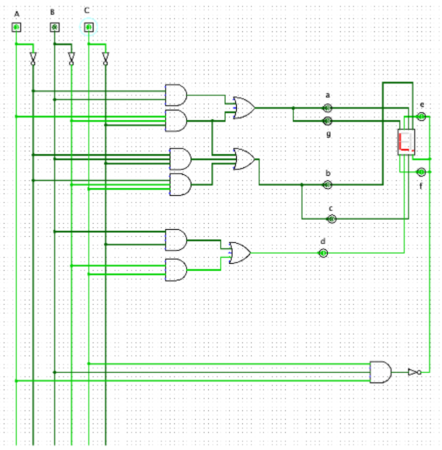
Letter “B”



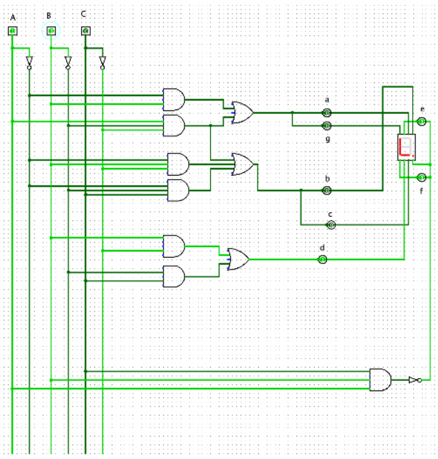
Letter “F”



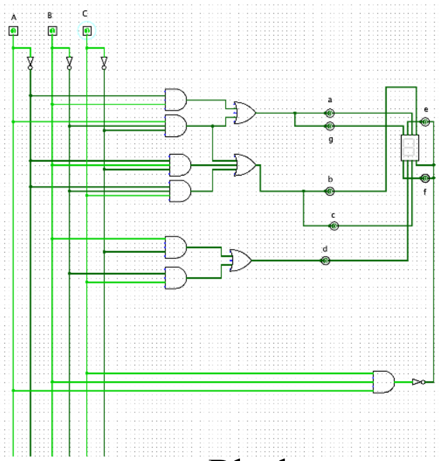
Letter “A”



Letter “L”



Letter “L”



Blank

- **POS Expression**

The product of sums form is a method (or form) of simplifying the Boolean expressions of logic gates. All these sum terms are multiplied together to get the product-of-sum form. This form is exactly opposite to the SOP form. Here the sum terms are defined by using the OR operation and the product term is defined by using AND operation. When two or more sum terms are multiplied by a Boolean OR operation, the resultant output expression will be in the form of product-of-sums form or POS form.

The 3-variable Karnaugh map is an array of eight cells. For our project Kmap Using POS is given below:

Pos Expressions

K- map for-a

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	1	1
AB	0	×
$A\bar{B}$	1	0

$\bar{a} = (\bar{A}\bar{B} + \bar{B}\cdot C + A\cdot B)$

$\bar{\bar{a}} = \overline{\bar{A}\bar{B} + \bar{B}\cdot C + A\cdot B}$

$\bar{a} = \overline{\overline{(A+B)}}\cdot\overline{\overline{(B+\bar{C})}}\cdot\overline{\overline{(\bar{A}+\bar{B})}}$

$a = (A+B)\cdot(B+\bar{C})\cdot(\bar{A}+\bar{B})$

K-map for b

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	1	0
AB	0	×
$A\bar{B}$	1	0

$\bar{b} = BC + AC + AB + \bar{A}\bar{B}\bar{C}$

$\bar{\bar{b}} = \overline{B\cdot C + AC + AB + \bar{A}\cdot\bar{B}\cdot\bar{C}}$

$b = (\bar{B}+\bar{C})\cdot(\bar{A}+\bar{C})\cdot(\bar{A}+\bar{B})$

$(A+B+C)$

K-map for - c

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	1	0
AB	0	×
$A\bar{B}$	1	0

$c = (\bar{B}+\bar{C})(\bar{A}+\bar{C})(\bar{A}+\bar{B})$

$(A+B+C)$

K-map for - d

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	1	0
AB	1	×
$A\bar{B}$	0	1

$\bar{d} = \bar{B}\bar{C} + BC$

$\bar{\bar{d}} = (B+C)\cdot(\bar{B}+\bar{C})$

$d = B$

K-map for e

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	1	1
AB	1	x
$A\bar{B}$	1	1

$e = 1$

K-map for f

	$\bar{C}$	C
	1	1
	1	1
	1	x
	1	1

$f = 1$

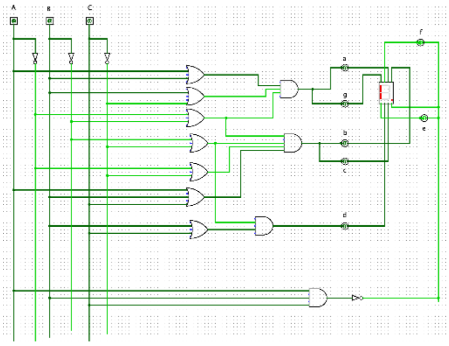
K-map for g

$AB \backslash C$	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	1	1
AB	0	x
$A\bar{B}$	1	0

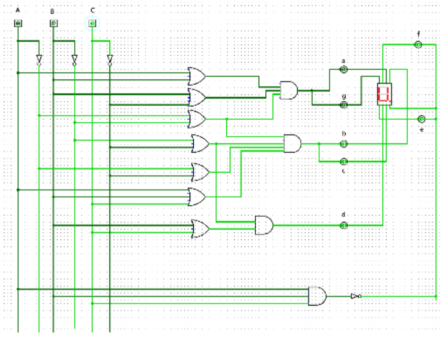
$g = (A+B) \cdot (B+\bar{C}) \cdot (\bar{A}+\bar{B})$

K-Map for POS Expressions

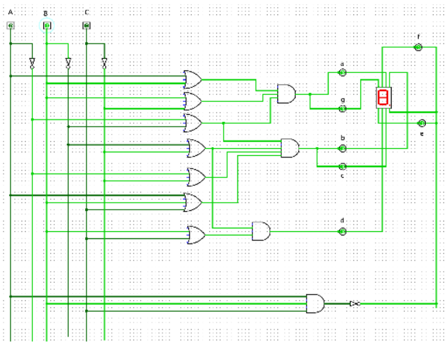
Circuit Diagrams (POS)



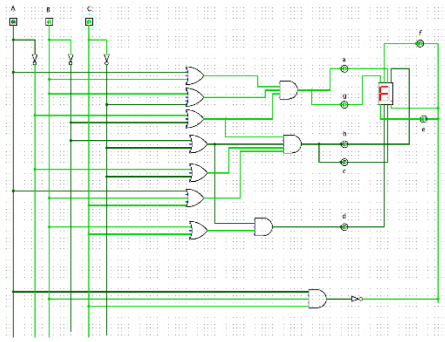
Letter “I”



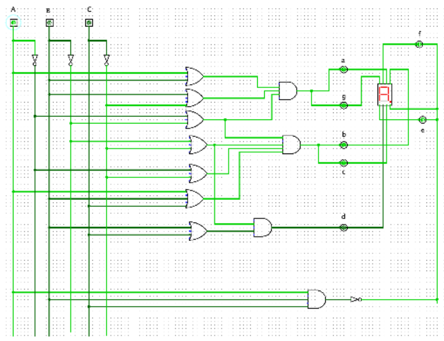
Letter “U”



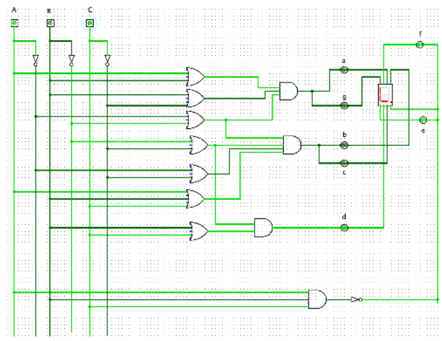
Letter “B”



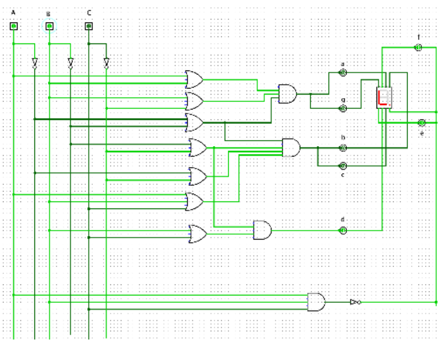
Letter “F”



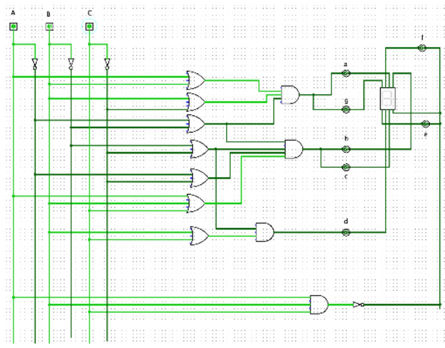
Letter “A”



Letter “L”



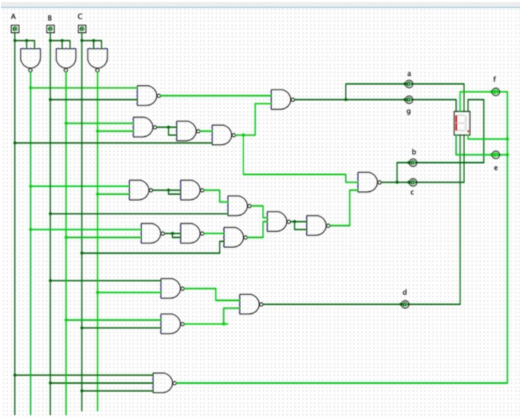
Letter “L”



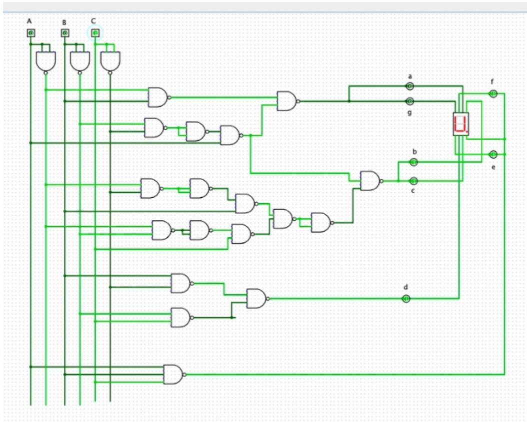
Blank

- **NAND Gate**

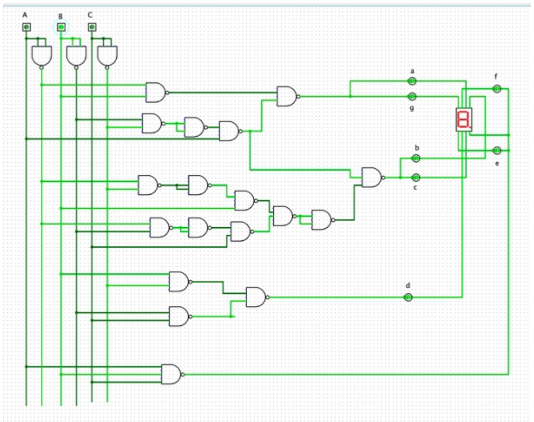
An operator which gives the value zero if and only if all the operands have a value of one, and otherwise has a value of one (equivalent to NOT AND). For using NAND gate, we replaced all the basic gates with NAND gates. We constructed NOT gate, OR gate and AND gates using NAND gates.And after that we got the following circuit:



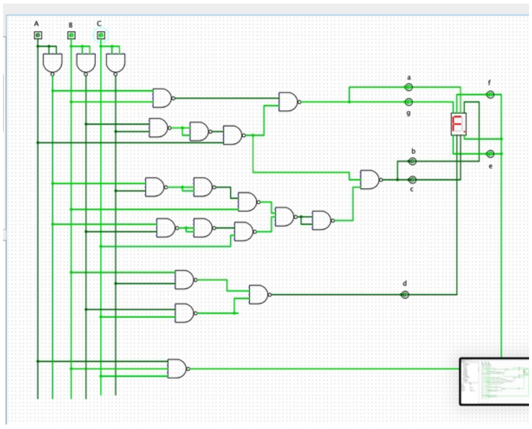
Letter “I”



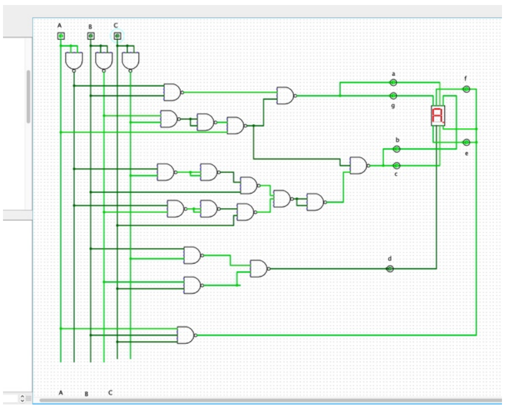
Letter “U”



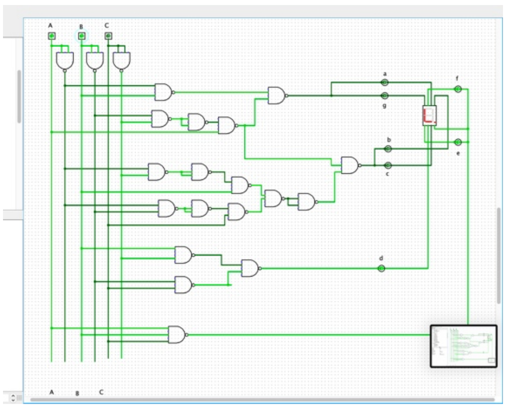
Letter “B”



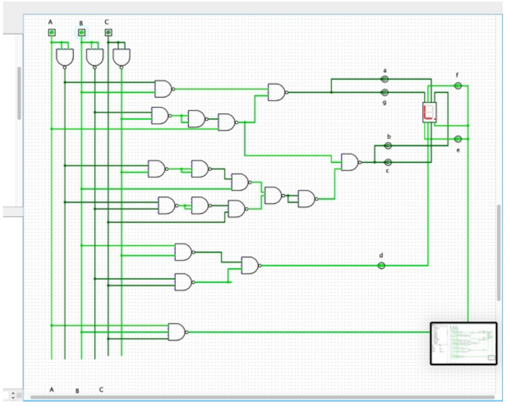
Letter “F”



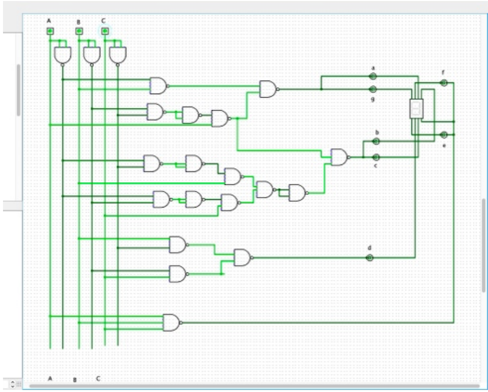
Letter “A”



Letter “L”



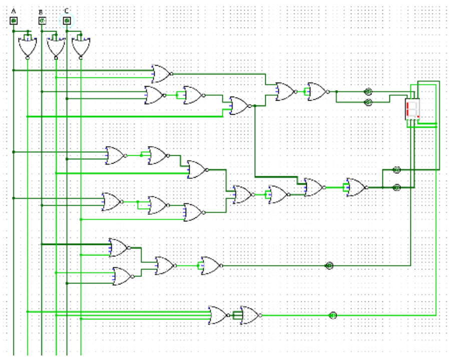
Letter “L”



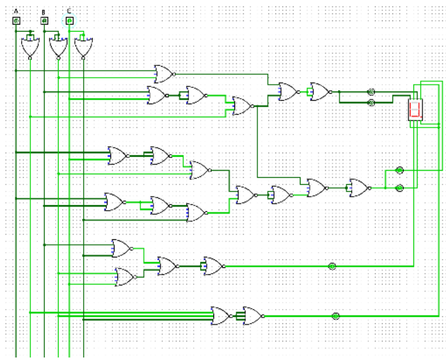
Blank

- **NOR Gate**

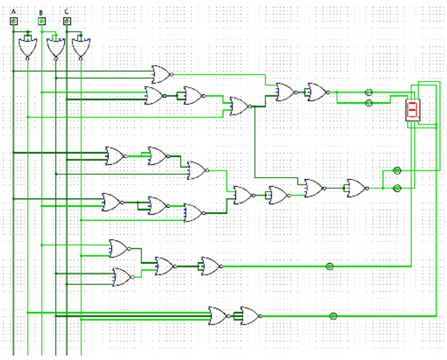
  A Boolean operator which gives the value one if and only if all operands have a value of zero and otherwise has a value of zero. For using NOR gate, we replaced all the basic gates with NOR gates. We constructed NOT gate, OR gate and AND gates using NOR gates. And after that we got the following circuit:



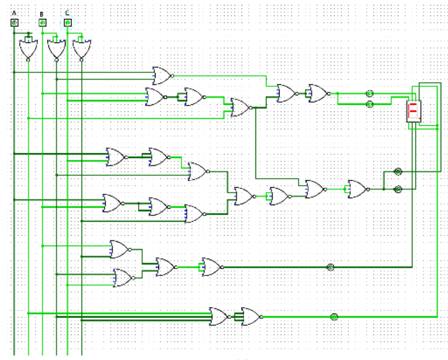
Letter “I”



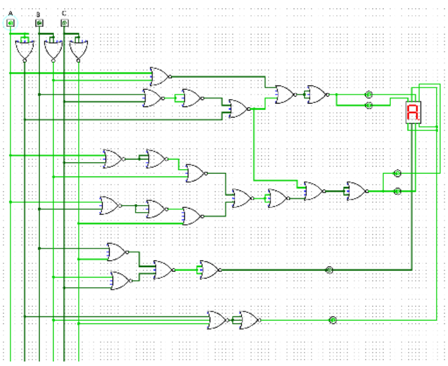
Letter “U”



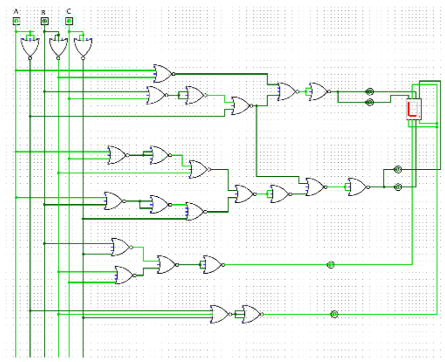
Letter “B”



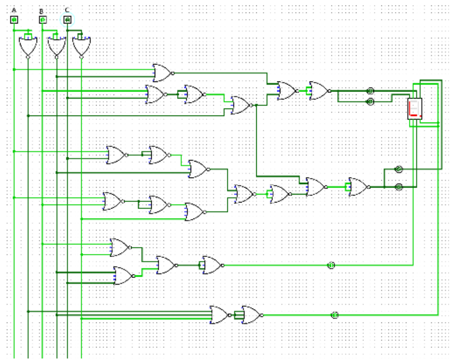
Letter “F”



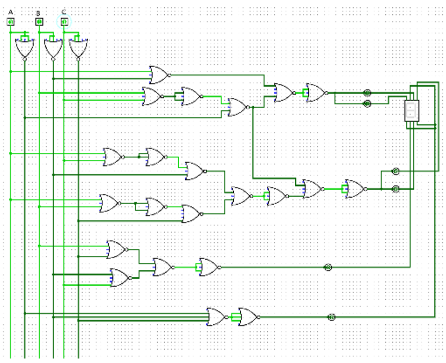
Letter “A”



Letter “L”



Letter "L"



Blank

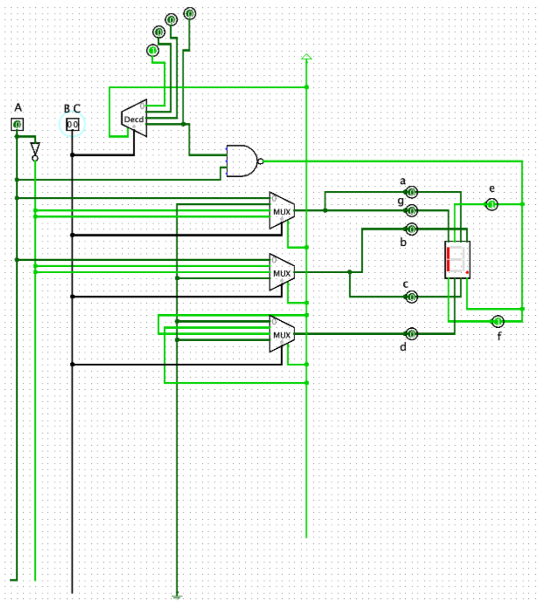
- **Multiplexer**

Multiplexer is a combinational circuit that has 2^n input lines and a single output line. Simply, the multiplexer is a multi-input and single-output combinational circuit. The binary information is received from the input lines and directed to the output line. On the basis of the values of the selection lines, one of these data inputs will be connected to the output. Unlike encoder and decoder, there are n selection lines and 2^n input lines. So, there is a total of 2^N possible combinations of inputs. A multiplexer is also treated as Mux.

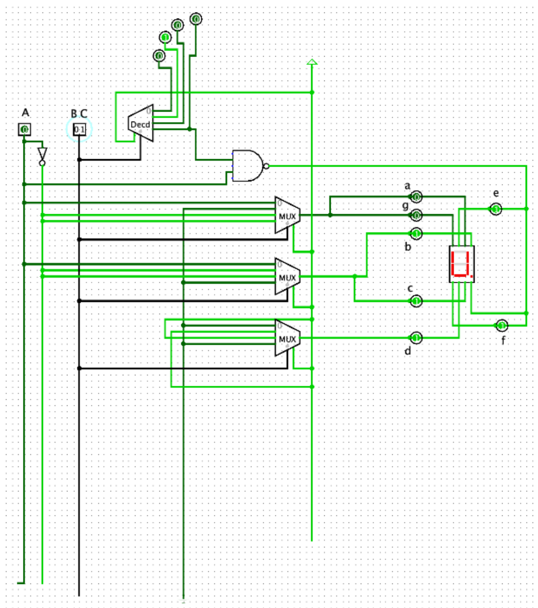
4:1 MUX

In the 4×1 multiplexer, there is a total of four inputs, i.e., A_0, A_1, A_2, and A_3, 2 selection lines, i.e., S_0 and S_1 and single output, i.e., Y. On the basis of the combination of inputs that are present at the selection lines S^0 and S_1, one of these 4 inputs are connected to the output.

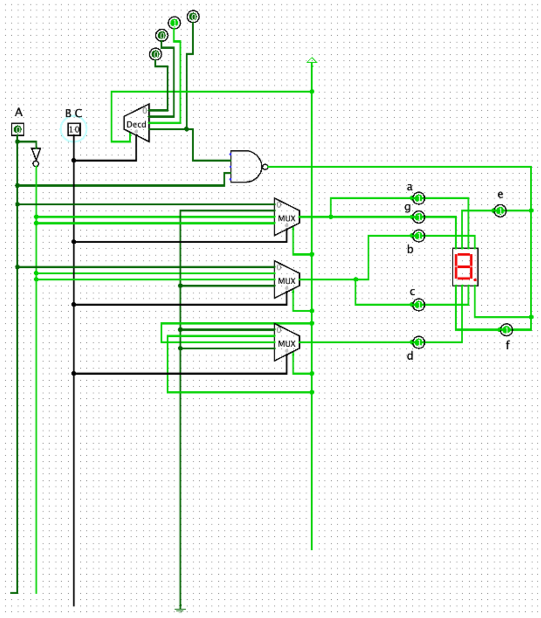
Circuit Diagram (Multiplexer)



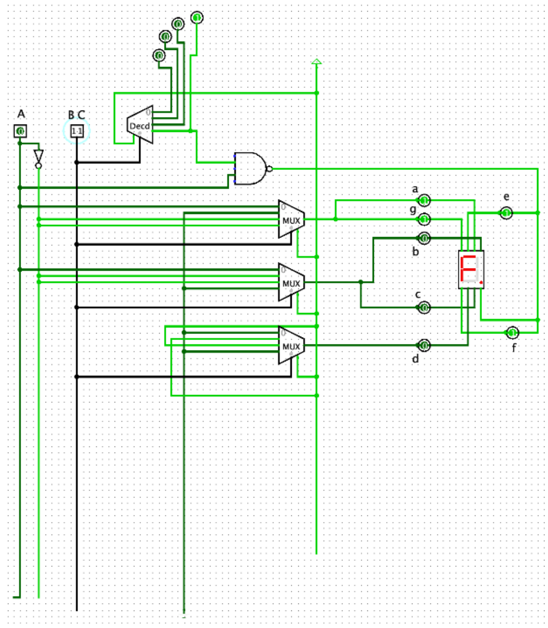
Letter "I"



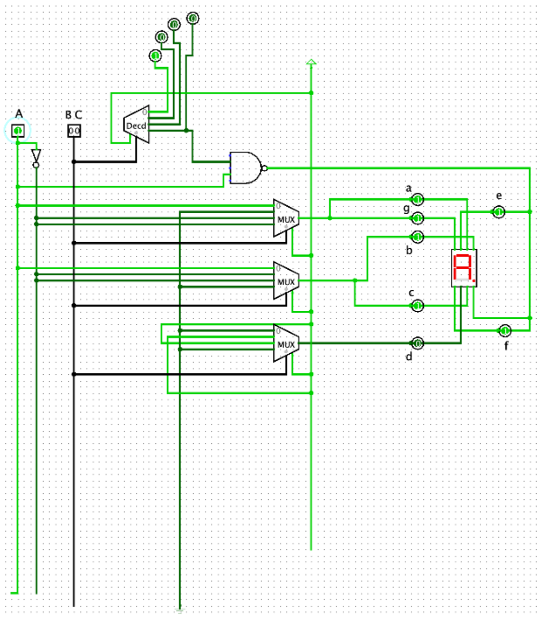
Letter "U"



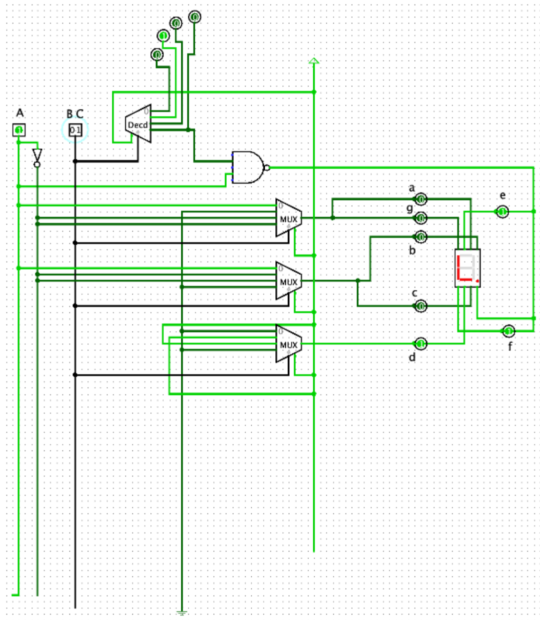
Letter “B”



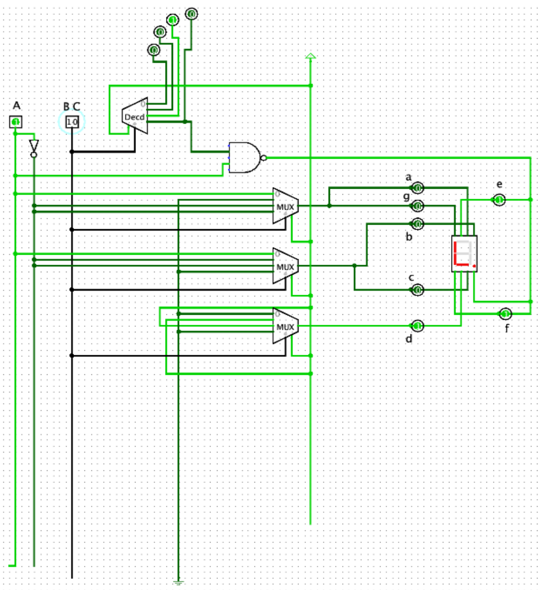
Letter “F”



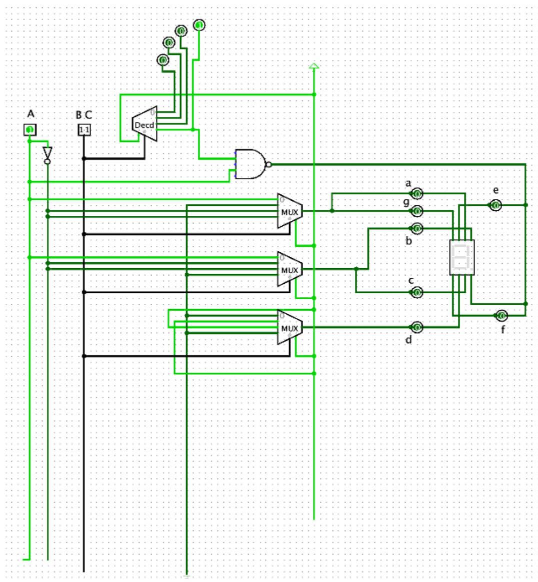
Letter “A”



Letter “L”



Letter “L”



Blank

4. Objective of the project

The purpose of the proposed seven segment display project is to understand the concept of the Boolean logic operation and Boolean Algebra, to learn K-maps to realize two-level minimal/optimal combinational circuits, to understand design process of combinational and sequential circuits, to learn operation of latches and flip-flops also analyze and design the sequential circuits by using this. And to combine all of this we are using simulation tool Logisim for digital system design.

5. Project Work Plan and Human Resources

- **Work Plan**

  At first, we need to create a truth table for "IUBFALL" then, from that table we have to create SOP, POS, NAND, NOR, and MUX Expression by using K-Map. Which satisfy the objective to understand the concept of Boolean function and K-Map to minimizing the expression. Then we have to build the minimized expression form into Logisim simulation which will satisfy logic design purpose. After that, we have to decide in which form the project is efficient and less costly. Then we have to build according to the plan and execute the letters. By doing this project we can satisfy all the outcomes of the project objectives.

- **Human Resources**

Work Distribution (Topics)	Work done by
1. Truth Table (IUBFALL) 2. Combinational Circuit (MUX)	Saif Mohammed 2121913642 (Coordinator)
1. K-Map (NOR) 2. Combinational Circuit (NOR)	Al Zammee 2021648642
1. K-Map (SOP) 2. Combinational Circuit (SOP)	Humayra Rahman Nipa 2121128642
1. K-Map (NAND) 2. Combinational Circuit (NAND)	Umme Suraia Haque Setu 2031278642
1. K-Map (POS) 2. Combinational Circuit (POS)	Afra Saiara Saima 2132242642

6. Proposed Budget

As we are using Multiplexer to display “IUBFALL” letters we need
2-IC 74LS153 (4:1 MUX) cost= 90 Tk
1-IC 7400 (2-input NAND gate) cost = 30 Tk
1-IC 7404 (2-input NOT gate) cost = 30 TK
2-Breadboard cost = 300 Tk
1-9V Battery cost = 100 TK
1-IC 7805 (Register) = 15 Tk
Wires Set cost = 90 Tk
1-Cathode 7 Segment Display = 15 Tk
Total cost = 670 Tk
So, the proposed budget for the combinational circuit is 670 Tk.

7. Conclusion

As we can do this project with all minimized SOP, POS, NAND, NOR and MUX form of expression. We choose the 4:1 MUX form to display the letters because of the efficiency of bits and fast execution of timing. That’s why we proposed to build this project based on the Multiplexer.