NORTH SOUTH UNIVERSITY

School of Engineering & Physical Sciences

CSE-231

Digital Logic Design

Submitted To:

Saifuddin Md. Tareeq
Associate Professor
Department of Electrical
& Computer Engineering

Submitted By:

Md.Nur-E-Azam
1512268042

Sahid khan
1421432042

Al-Amin
1520179642

Joy saha
1520715642

ACKNOWLEDGEMENTS

At first we will like to thank Almighty, by his grace everything went smooth. We are mostly grateful to our course instructor Saifuddin Md. Tareeq who was in charge of our whole project. Under his supervision and with assist of our lab instructor Miss. Diya Islam we made it happen. The lab staffs also were so co-operating.

North South University gave us plenty of privileges regarding laboratory facilities. Our laboratory was well equipped. The lab experiments were very helpful. So we will also like to thank our authority.

Again Thanks to M. Morris Mano and Ronald J. Tocci, authors of our textbook“Digital Design - With an Introduction to the Verilog HDL”. The book came in handy during the project.

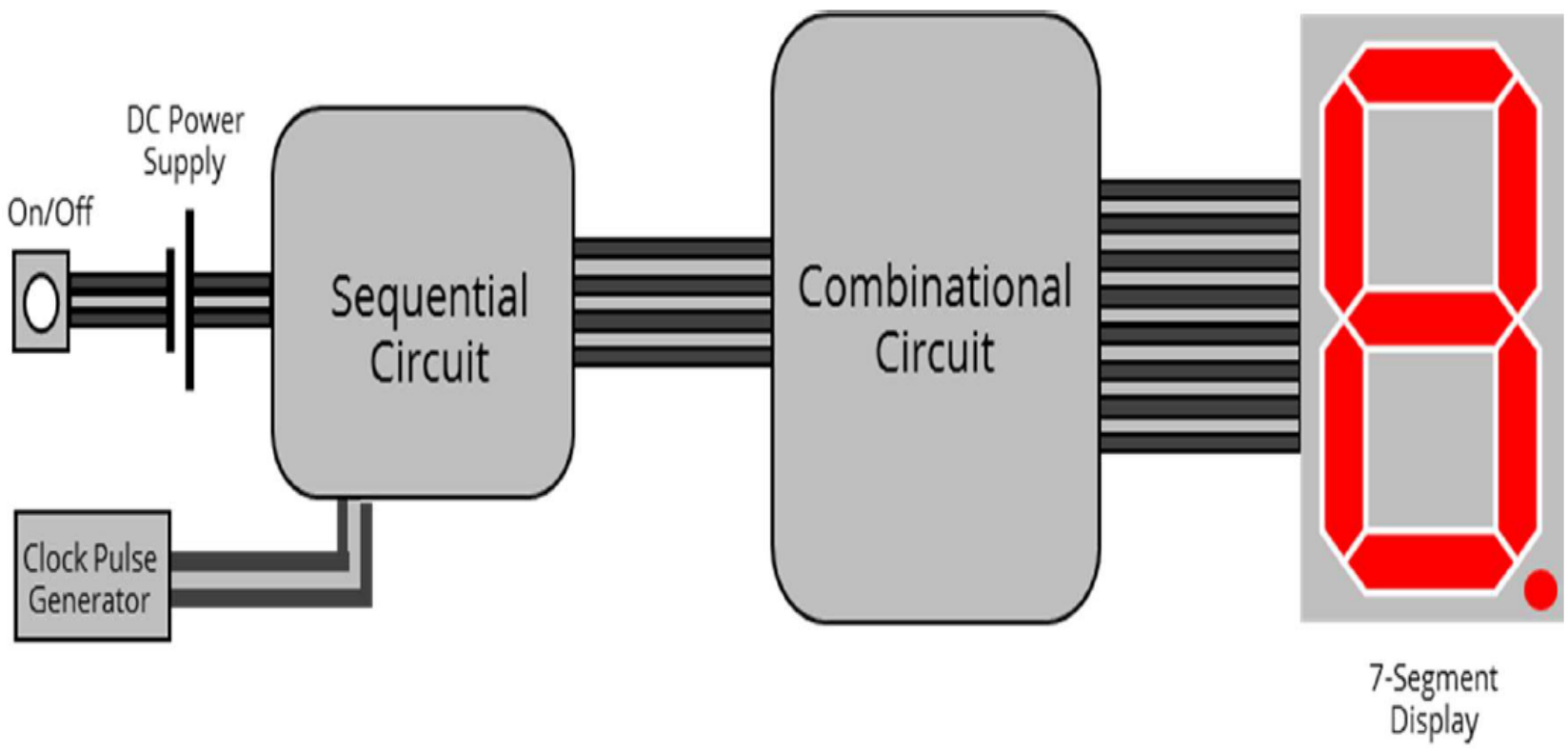
Finally, we will like to express our heartiest gratitude to our dear parents who supported us all the moments.

TABLE OF CONTENT

1 INTRODUCTION

1.1 Project Description

This project is all about the digital system shown in Fig. 1 which displayed the string of characters on the seven segment display, one character at a time, at a time interval of approximately 1.386 seconds, and at the end of the sequence string is repeated. According to the string format “CSE-LAB Sec no”, our desired string is
“CSE-LAB 5”.



2 DESIGN

2.1 Design of Combinational Logic Circuit

We have total 10 characters and 8 unique characters. So we need 3 bits A, B, C to design the combinational circuit.

		Input			Output							
Ind.	Char	A	B	C	a	b	c	d	e	f	g	Minterms
0	C	0	0	0	1	0	0	1	1	1	0	A’B’C’
1	S	0	0	1	1	0	1	1	0	1	1	A’B’C
2	E	0	1	0	1	0	0	1	1	1	1	A’B C’
3	–	0	1	1	0	0	0	0	0	0	1	A’B C
4	L	1	0	0	1	0	0	1	1	1	0	A B’C’
5	A	1	0	1	1	0	1	1	1	1	1	A B’C
6	B	1	1	0	1	1	1	1	1	1	1	A B C’
7	5	1	1	1	1	0	1	1	0	1	1	A B C

Expressions for a, b, c, d, e, f, g are given below,

a(A, B, C) = Σ(0, 1, 2, 4,5, 6)

b(A, B, C) = Σ(6)

c(A, B, C) = Σ(1, 5, 6,7)

d(A, B, C) = Σ(0, 1, 2, 4,5, 6,7)

e(A, B, C) = Σ(0, 2,4,5, 6)

f(A, B, C) = Σ(0, 1, 2,4,5, 6,7)

g(A, B, C) = Σ(1, 2, 3, 5, 6, 7)

2.1.1 1st Canonical Form:

Functions of a, b, c, d, e, f, g as 1st canonical form are given below,

a = A'B'C'+A'B'C+A'BC'+AB'C'+AB'C+ABC'+ABC
b = ABC'
c = A'B'C+AB'C+ABC'+ABC
d = A'B'C'+A'B'C+A'BC'+AB'C'+AB'C+ABC'+ABC
e = A'B'C'+A'BC'+AB'C'+AB'C+ABC'
f = A'B'C'+A'B'C+A'BC'+AB'C'+AB'C+ABC'+ABC
g = A'B'C+A'BC'+A'BC+AB'C+ABC'+ABC

2.1.2 Minimal 1st Canonical Form (SOP):

To find the minimal 1st canonical form (SOP), we need to draw K-Maps,

2.1.3 Minimal Logic Implemented Using Universal Gates:

In order to build the combinational circuit with universal gate, we replaced the minimal 1st canonical circuit with NAND and NOR gates.

2.1.4 Decoder:

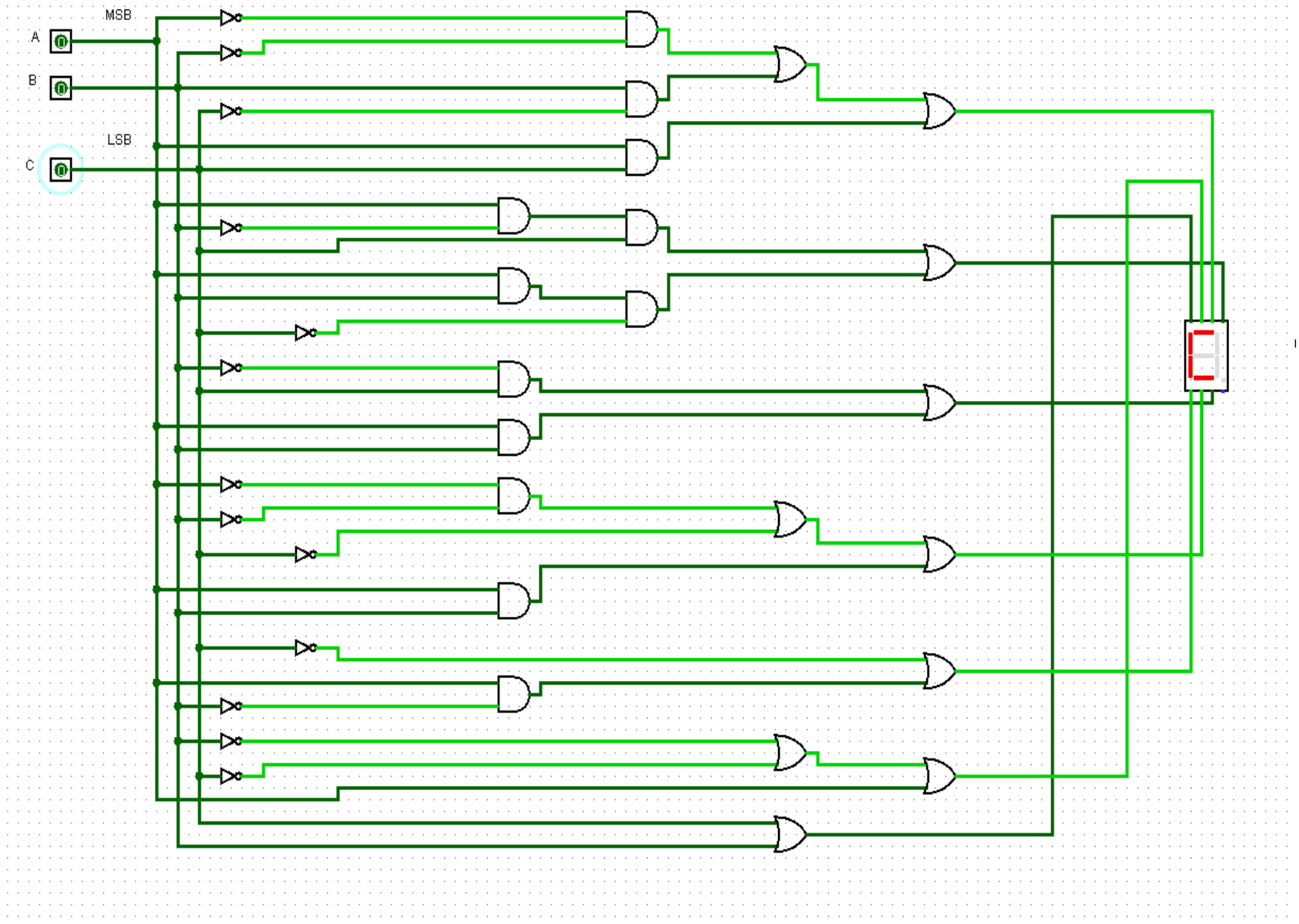
We have 3 input bits. So we use 3 to 8 line decoder to design the circuit. Table of
decoder given below:

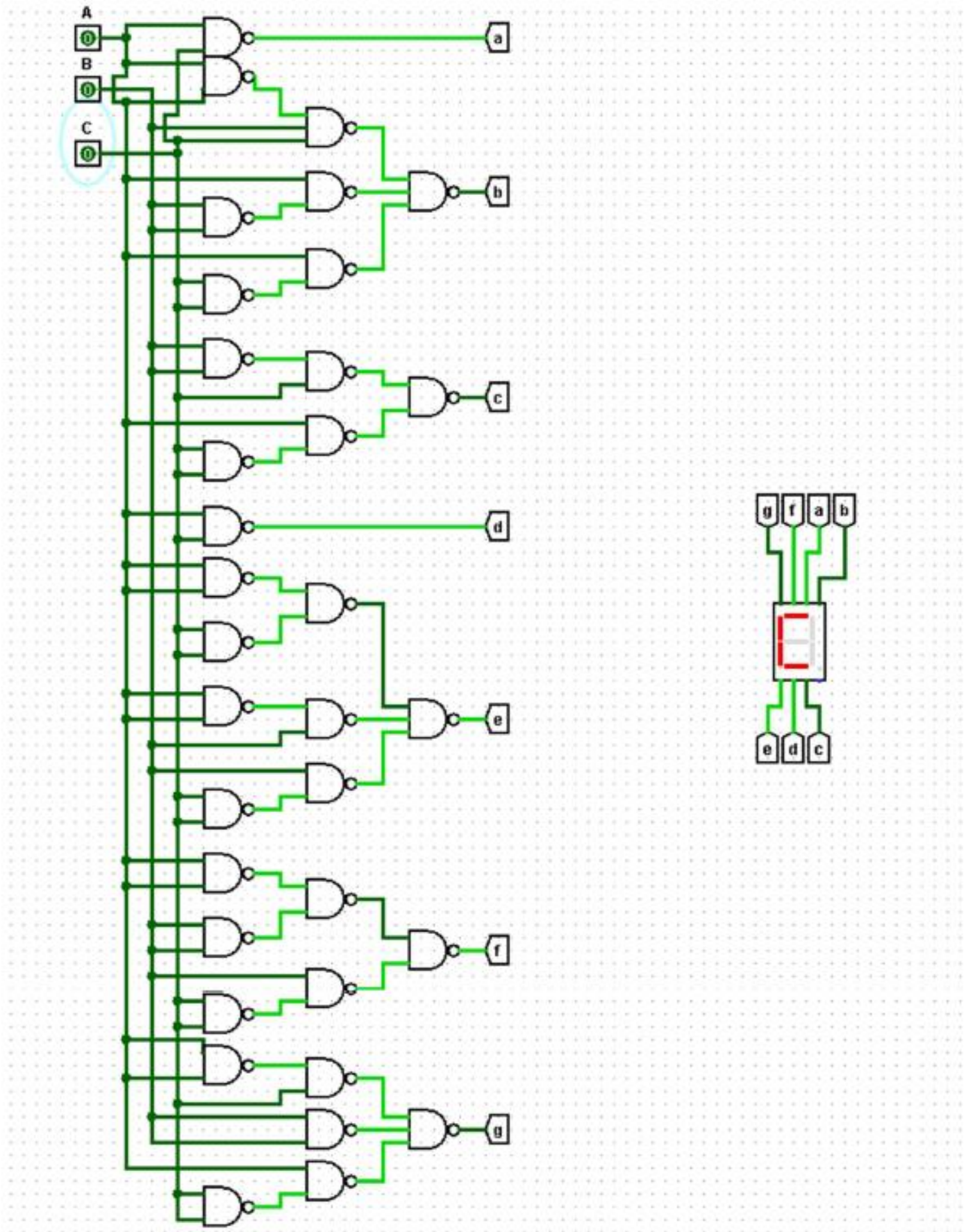
Enable	Input			Output							
	A	B	C	D0	D1	D2	D3	D4	D5	D6	D7
0	x	X	x	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

2.2 Justification of Choice of Design

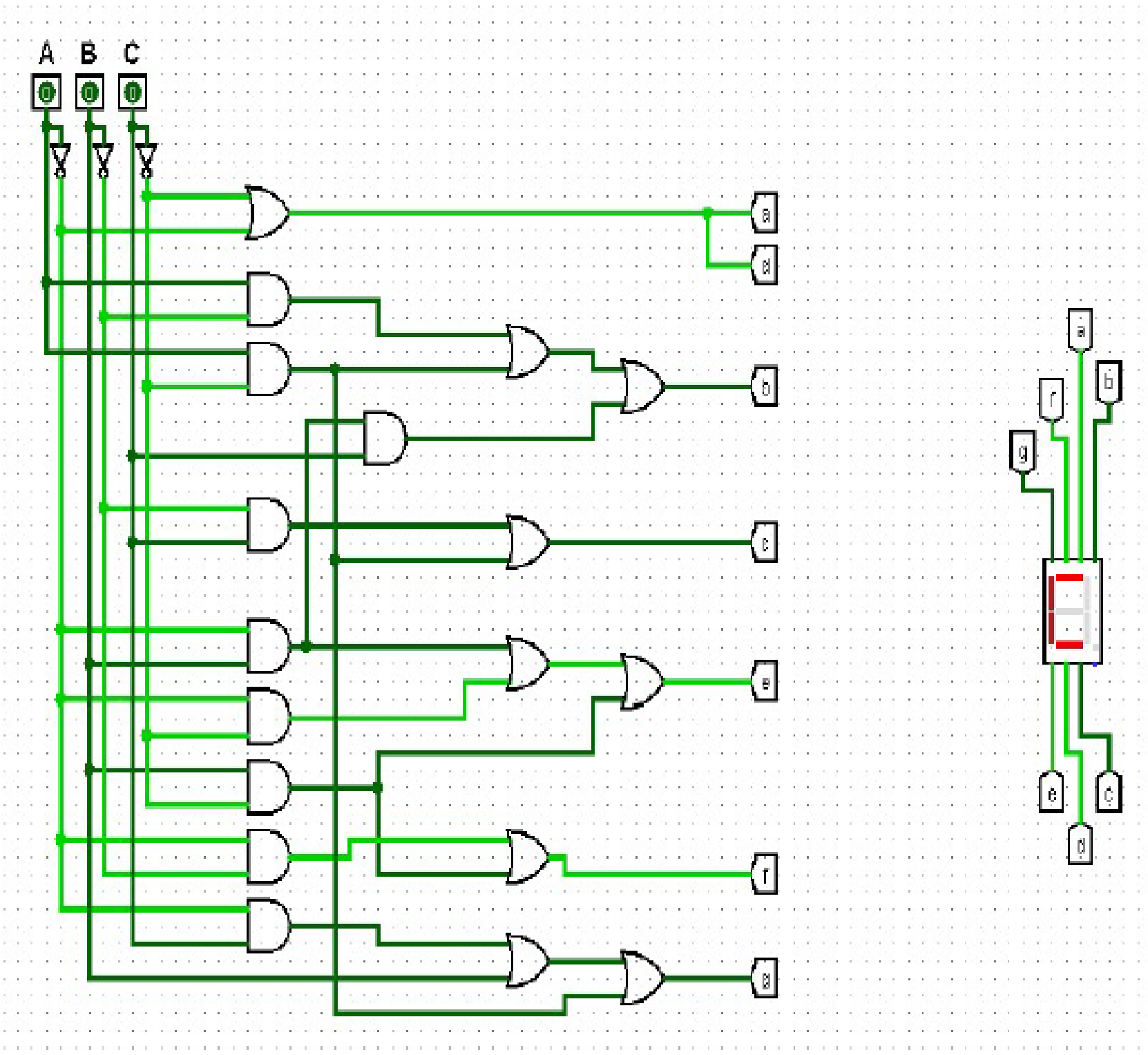
In the final implementation we used minimal 1st canonical form. We had definite reasons behind it. Since it was built with basic gates, it was easier not only to build but also to understand. The IC diagram of this was simpler than the IC diagrams of Decoder and Multiplexer. As a result implementing the circuit in breadboard wasquite easy.

2.3 Simulation of Combinational Circuit

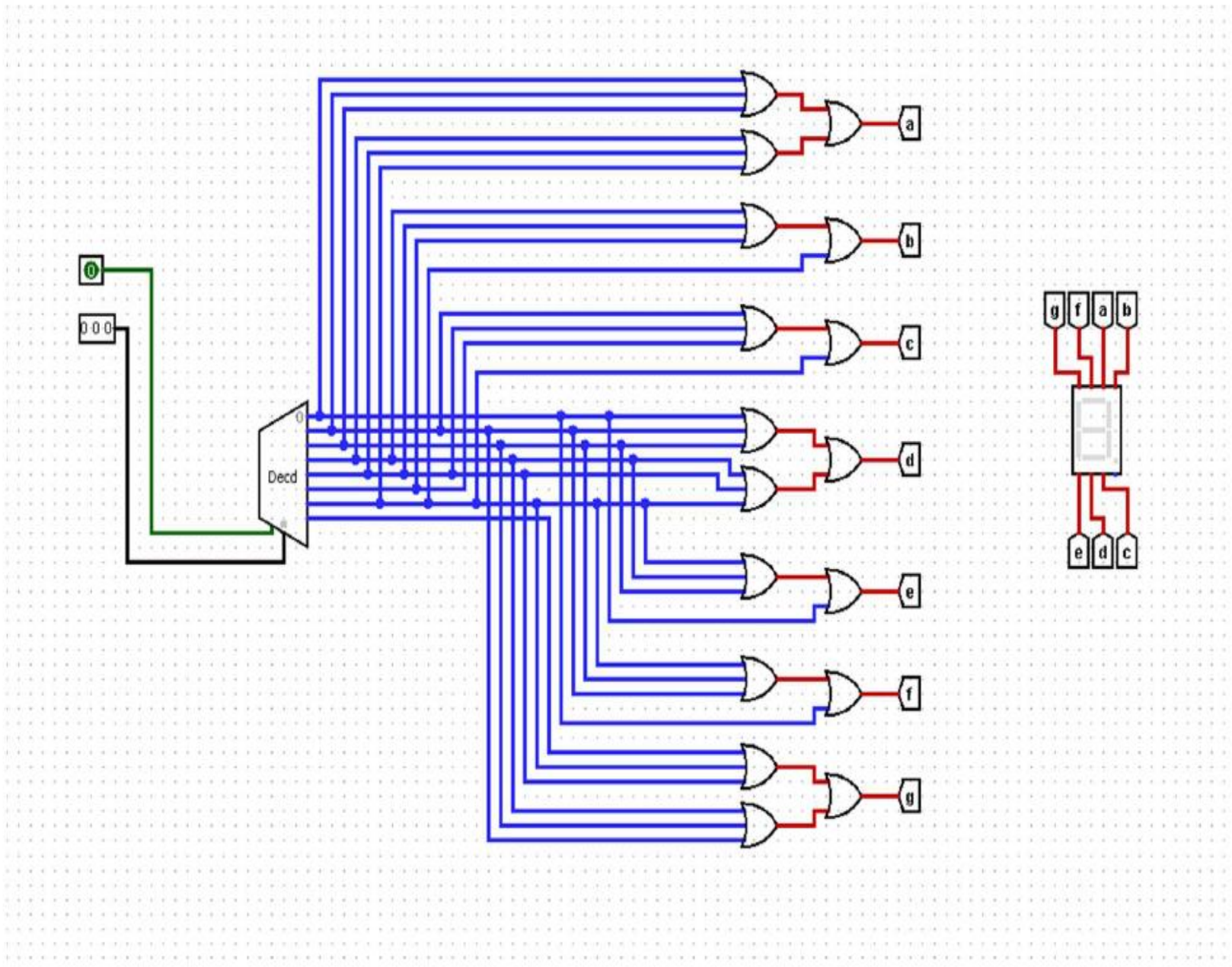




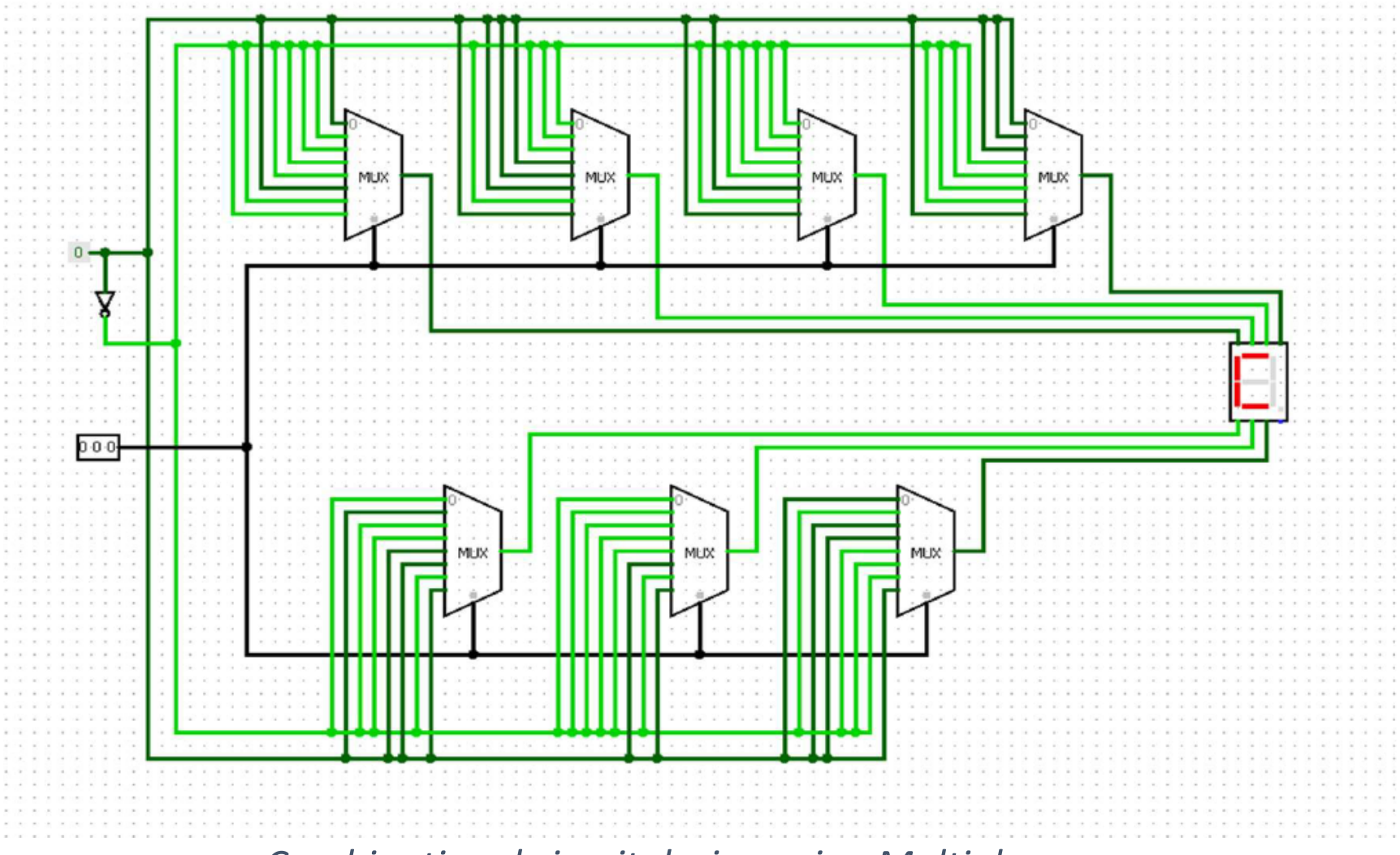
Minimal logic implemented using universal gates



Logisim schematic of minimal 1st canonical form



Combinational circuit design using Decoder



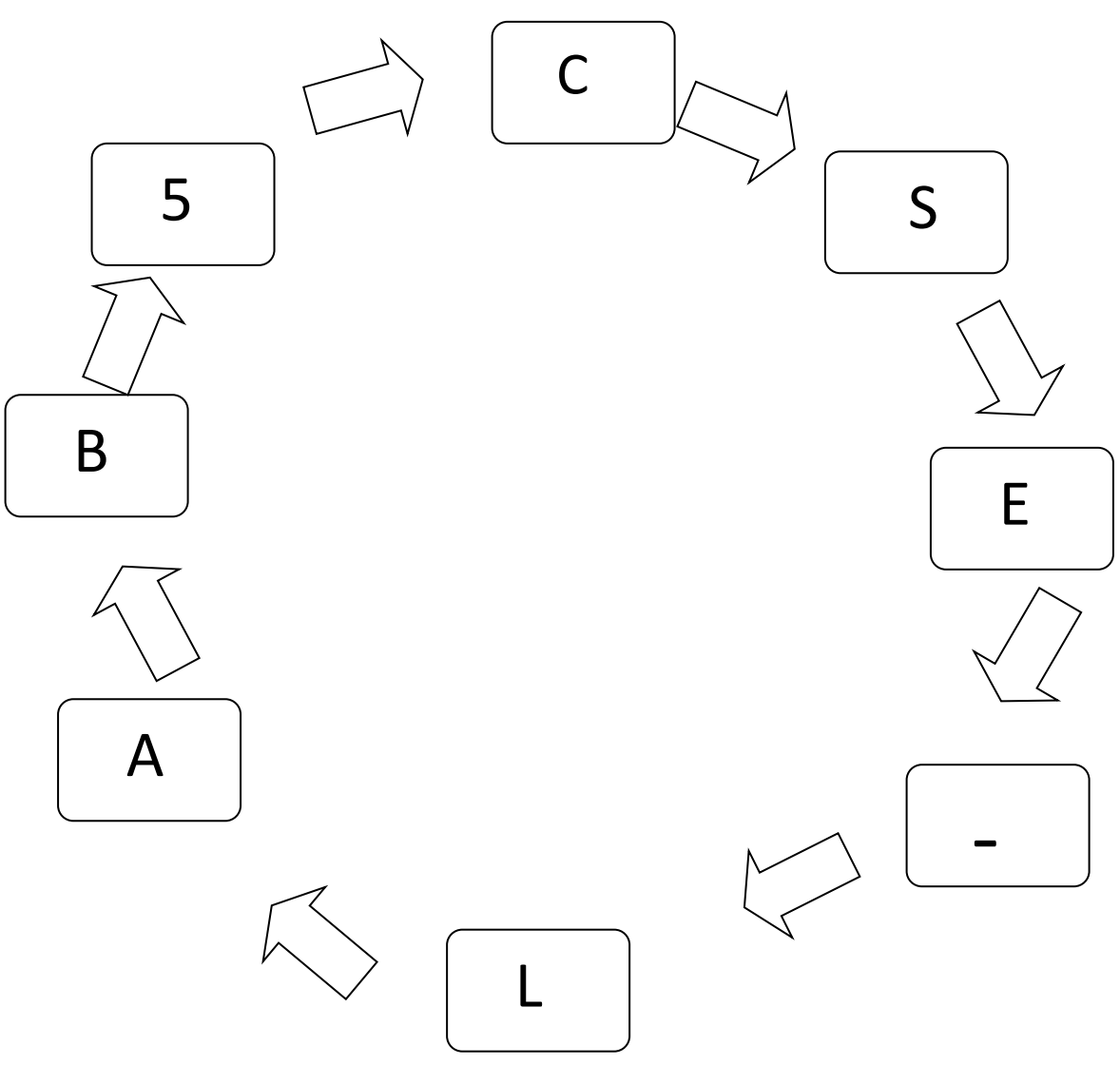
Combinational circuit design using Multiplexer

2.4 Design of Sequential Logic Circuit

We have total 8 characters that means 8 states. So we need 3 bits to represent all state.

Character	X	Y	Z
C	*0*	*0*	*0*
S	*0*	*0*	*1*
E	*0*	*1*	*0*
-	*0*	*1*	*1*
L	*1*	*0*	*0*
A	*1*	*0*	*1*
B	*1*	*1*	*0*
5	*1*	*1*	*1*

State Diagram of sequential circuit is given below,



State diagram of sequential circuit

We uses D flip-flops to design the sequential circuit. Now we need state transition table which is given below,

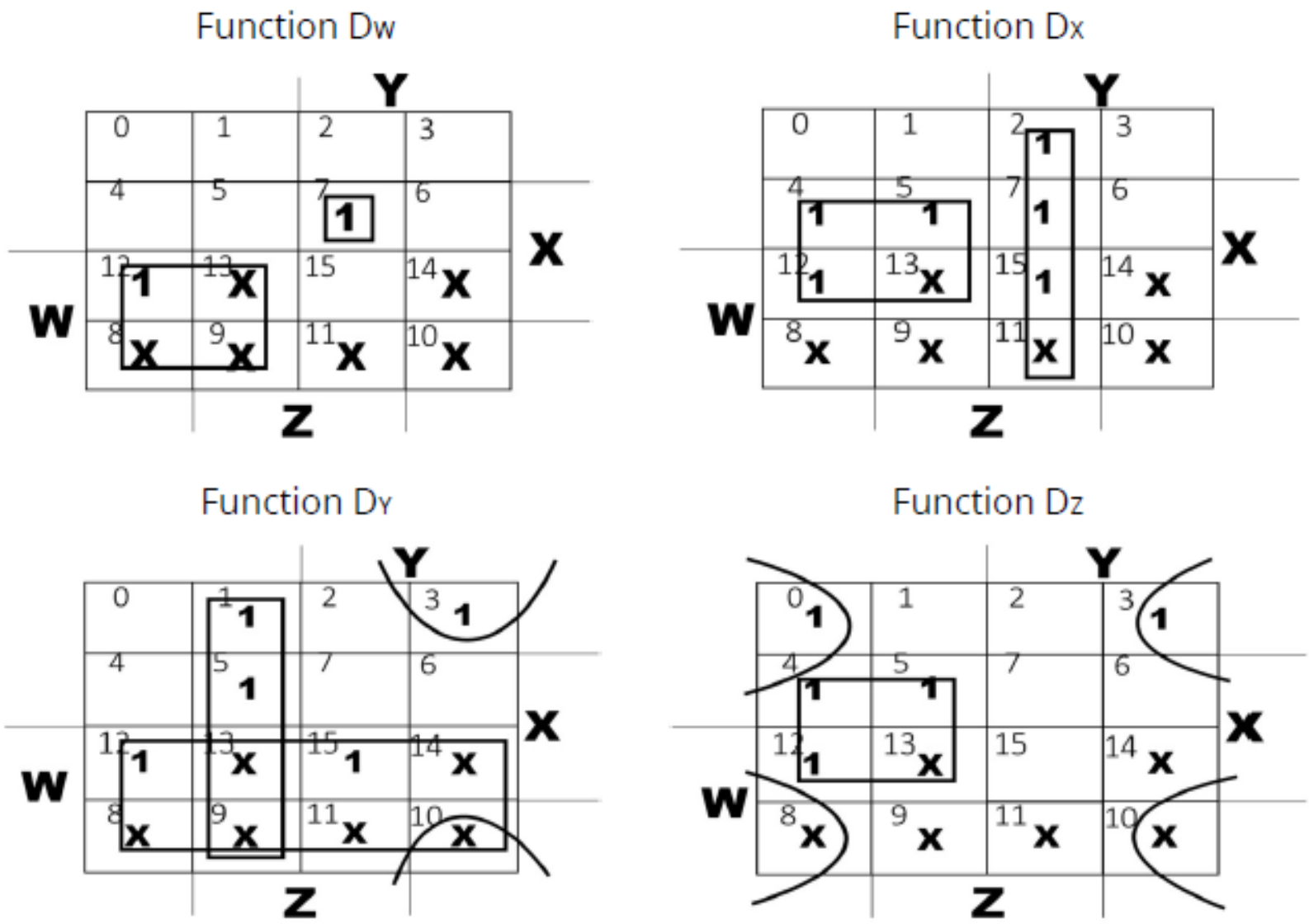
Ind.	Present State				Next state				F. F. Input			
	W(t)	X(t)	Y(t)	Z(t)	W(t+1)	X(t+1)	Y(t+1)	Z(t+1)	Dw	Dx	Dy	Dz
0	0	0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	0	1	0	0	0	1	0
2	0	0	1	0	0	0	1	1	0	0	1	1
3	0	0	1	1	0	1	0	0	0	1	0	0
4	0	1	0	0	0	1	0	1	0	1	0	1
5	0	1	0	1	0	1	1	1	0	1	1	1
6	0	1	1	0	0	0	0	0	0	0	0	0
7	0	1	1	1	1	1	0	0	1	1	0	0
8	1	0	0	0	x	x	x	x	x	x	x	x
9	1	0	0	1	x	x	x	x	x	x	x	x
10	1	0	1	0	x	x	x	x	x	x	x	x
11	1	0	1	1	x	x	x	x	x	x	x	x
12	1	1	0	0	1	1	1	1	1	1	1	1
13	1	1	0	1	x	x	x	x	x	x	x	x
14	1	1	1	0	x	x	x	x	x	x	x	x
15	1	1	1	1	0	1	1	0	0	1	1	0

State Transition table of sequential circuit

Input functions of DFF are given below,

$DW = \Sigma(7, 12) + \Sigma d(8, 9, 10, 11, 13, 14)$
$DX = \Sigma(3, 4, 5, 7, 12, 15) + \Sigma d(8, 9, 10, 11, 13, 14)$
$DY = \Sigma(1, 2, 5, 12, 15) + \Sigma d(8, 9, 10, 11, 13, 14)$
$DZ = \Sigma(0, 2, 4, 5, 12) + \Sigma d(8, 9, 10, 11, 13, 14)$

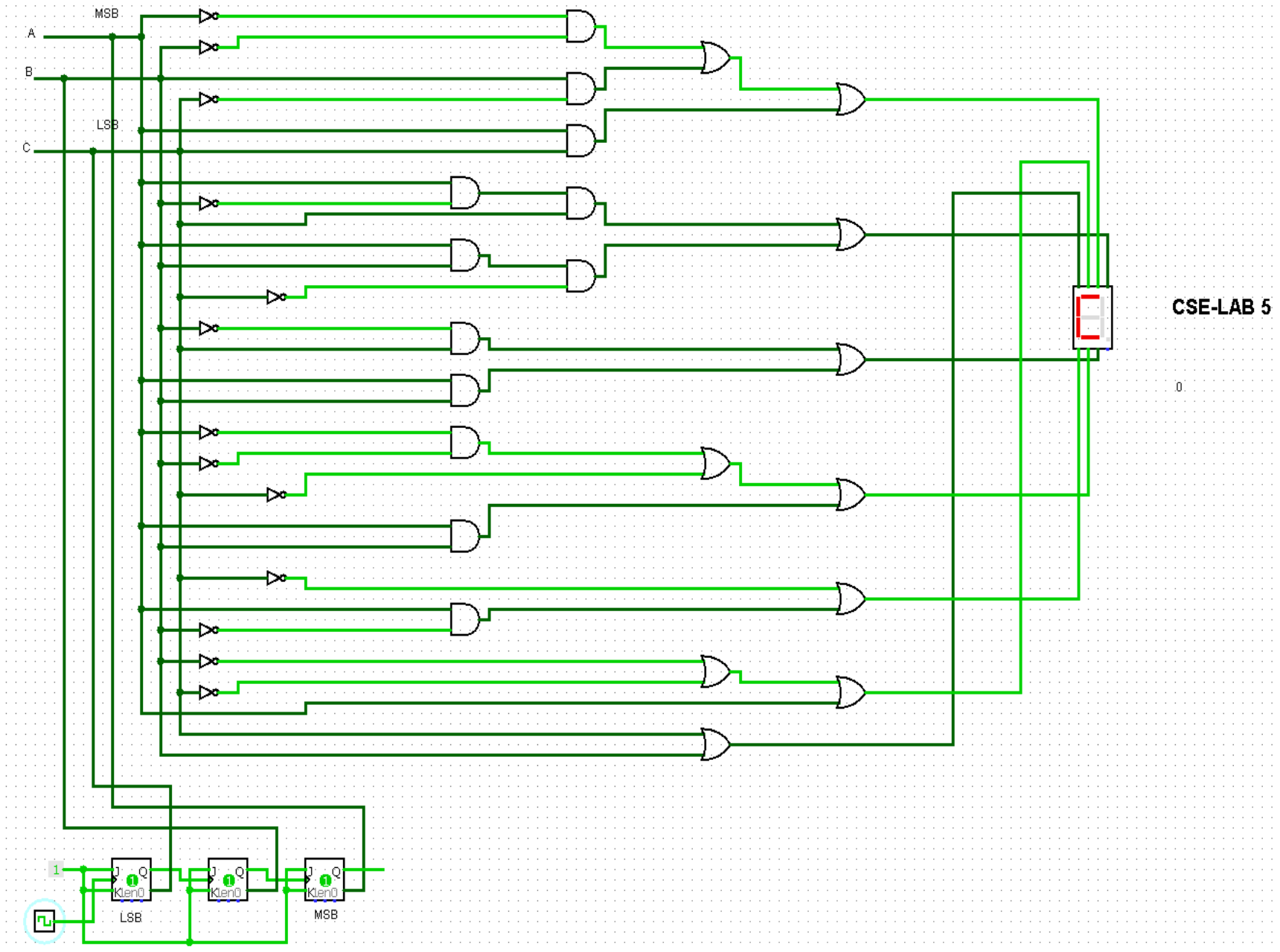
K-Maps of DW, DX, DY, DZ are given below,



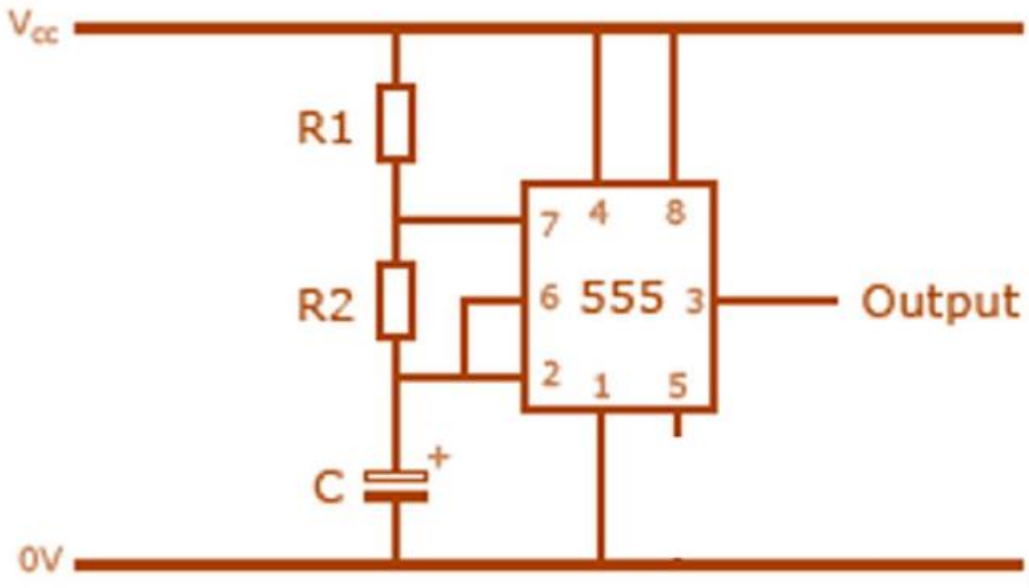
Therefore, functions of DFF inputs
DW = W’XYZ + WY’
DX = YZ + XY’
DY = Y’Z + X’YZ’ + W
DZ = X’Z’ + XY’

2.5 Justification of Choice of Design We know that sequential circuits can be designed in two ways which are Moore model and Mealey model. In Moore model all the outputs are synchronized with the clock because they depend only on flip-flop outputs that are synchronized with clock. Since all the outputs of our project depend on flip-flops, we used Moore model.

2.6 Simulation of Sequential Circuits



2.7 Clock pulse generator The 555 has three main operating modes, Monostable, Astable, and Bistable. We use Astable circuit to make the clock pulse generator. An Astable
Circuit has no stable state - hence the name "astable". In an astable circuit, the output continually switches state between high and low without any intervention from the user. The duration of the high and low states are based on what values we chose for R1, R2 and C. For 2.079 seconds time delay, we used 10kΩ as R1, 10kΩ as R2 and 100μF capacitor as C.
The formulas used are:
Time High (secs) = 1.386 s
Time Low (secs) = 693.000ms
Frequency = 0.481 Hz



An Astable circuit using 555 timer IC

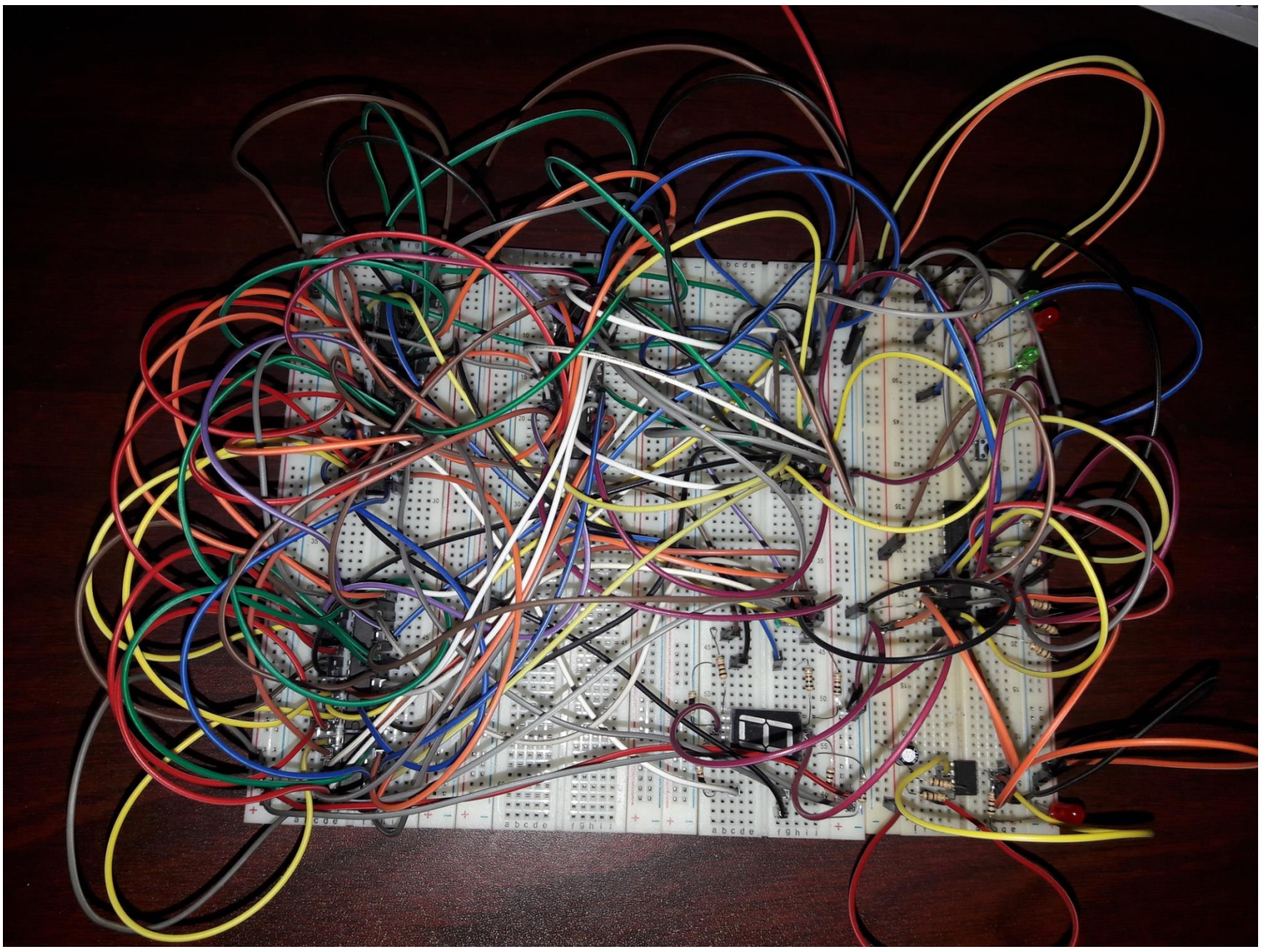
3 IMPLEMENTATION

In the combinational part we implemented the Minimal SOP design among our 4 circuit designs. Although the design was not part of Phase I, it turned out to be more feasible in terms of cost. And, we have implemented the Moore model in the sequential part, using all D Flip Flops.

Throughout the project, we used 5 different types of ICs. Specifications are as follows,

IC model	Brief Description	Number of components	Quantity of ICs used
7404	This is a NOT gate IC	6 Inverters	
7408	This is a 2 inputs AND gate IC	4 AND gates	5
7432	This is a 2 inputs OR gate IC	4 OR gates	5
7474	This is D flip-flop IC which can store at most 2 bits	2 DFF	2
555	This is a timer IC	--------	1

ICs specifications



Final Hardware Implementation

4 CONCLUSION

4.1 Budget & Cost When we designed and implemented this project, we tried to minimize the cost as much as possible. The overall cost incurred for the project is summarized as below,

Project cost summary:

Equipment name	Quantity	Rate (Tk.)	Amount (Tk.)
7404 IC	3	25.59	76.77
7408 IC	5	23.59	117.95
7432 IC	5	25.59	127.95
7474 IC	2	25.9	51.8
555 IC	1	12.67	12.67
wires	5 set	43	215
Resistors	22	1.25	27.5
Capacitor	1	3.1	3.1
Display	1	9.9	9.9
Switch	1	3.5	3.5
Bread Board	4	103.9	415.6
Led	4	1.31	5.24
		Total Costs	1066.98

Table : Project cost summary

4.2 Difficulties Faced

Throughout the project implementation phase, we had come across a few challenges. We had to set up our circuit, both combinational and sequential parts, 3 times on the breadboard due malfunction of IC chips and/or faulty wiring. And, evaluating the timer section of the circuit was a little challenging – it took us a while to learn that RC circuit was an essential part of the timer.

4.3 Future Recommendations

In our opinion, this circuit can be made far more robust, easier, less time consuming, and portable using VLSI design as discussed in class.

5 REFERENCES

Logisim: http://www.cburch.com/logisim/download.html
555 timer: http://www.555-timer-circuits.com/
Gantt chart: https://en.wikipedia.org/wiki/Gantt_chart
Writing reports: http://gradschool.unc.edu/academics/thesisdiss/guide/ordercomponents.html