North South University

Department of Electrical & Computer Engineering

LAB REPORT

Course Name:CSE231L

Section: 01

Experiment Number: 03,04

Experiment Name: Combinational Logic Design

Experiment Date: 14/03/2021

Report Submission Date:21/03/2021

Group Number:

Student Name: Ishrat Jahan	Score
Student ID: 1921909042	
Remarks:	

Combinational Logic Design

Objectives

- Becoming familiarized with the analysis of combinational logic networks.
- Implementation of circuits using the two canonical forms.
- Design a complete minimal combinational logic system from specification to implementation.
- Minimize combinational logic circuits using Karnaugh maps.
- Learn various numerical representation systems.
- Implement circuits using canonical minimal forms.

Theory

Canonical Form

In Boolean algebra, Boolean function can be expressed as Canonical Disjunctive Normal Form known as **minterm** and some are expressed as Canonical Conjunctive Normal Form known as **maxterm.**
In Minterm, we look for the functions where the output results in "1" while in Maxterm we look for function where the output results in "0".
Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form.

- **Sum of minterms –**
  The minterms whose sum defines the Boolean function are those which give the 1's of the function in a truth table. Since the function can be either 1 or 0 for each minterm, and since there are 2^n minterms, one can calculate all the functions that can be formed with n variables to be (2^(2^n)). It is also known as Sum of Products(SOP).
- **Sum of maxterms –**
  The maxterms whose sum defines the Boolean function are those which give the 0's of the function in a truth table. Since the function can be either 1 or 0 for each maxterm, and since there are 2^n maxterms, one can calculate all the functions that can be formed with n variables to be (2^(2^n)). It is also known as Product of Sum(POS).

Binary-coded Decimal (BCD)

BCD is a way of representing a decimal number as a string of bits suitable for use in electronic systems. We must represent the decimal digits by means of a code that contains1'sand 0's in order for computers to understand the values and perform operations using them as they accept only binary values whereas we usually work with decimal numbers. A decimal number in BCD is the same as its equivalent binary number only when the number is between 0 and 9.

For example-The BCD of 5 is 0101

Excess-3

Excess-3, is a non-weighted code used to express decimal numbers. It is another important binary code. The Excess-3 code for a given decimal number is determined by adding '3' to each decimal digit in the given number and then replacing each digit of the newly found decimal number by its four bit binary equivalent. For example-The Excess-3 of 24:

(2+3),(4+3)->(5),(7)

-->(0101)(0111)

=0101 0111

Gate-level minimization

Gate-level minimization is the task of designing an implementation of Boolean functions that describes a digital circuit. However, the technique of minimization lacks guidelines to proceed.

The Map Method

The map method is also known as the Karnaugh map or K-map. It provides a straight forward procedure for minimizing Boolean functions. The simplified expressions are always in one of the two standard forms:-

Sum of Products (SOP) or Product of Sums (POS)

A K-Map is a diagram made up of squares, with each square representing one minterm of the function that is to be minimized. Since any Boolean function can be expressed as a sum of minterms, it follows that a Boolean function is recognized graphically in the map from the area enclosed by those square whose minterms are included in the function. It is sometimes possible to find two or more expressions that satisfy the minimization criteria. The simplest algebraic expression is the one with minimum number of terms and with the smallest possible number of literals in each term. This expression produces a circuit diagram with a minimum number of gates and the minimum number of inputs to each gate. The number of squares in a K-Map varies according to the number of variables.

Using K-Map to minimize function

Step 1: Truth Table / Canonical Expression with Min- or Max-Terms

The first step in designing any digital circuit is to see the variables involved, along with their values. Further, we have to look at the outputs of each variable for each and every combination of the input literals, which is represented in the form of a truth table. From the truth table, we select the outputs with 1's which are for Minterms.

Step 2: Populate K-Map

From Step 1, we know the number of input variables involved in the logical expression from which we can determine the size of the K-map. Next, we have to fill the K-map cells with one for each minterm and zero for each maxterm.

Step 3: Forming the Groups

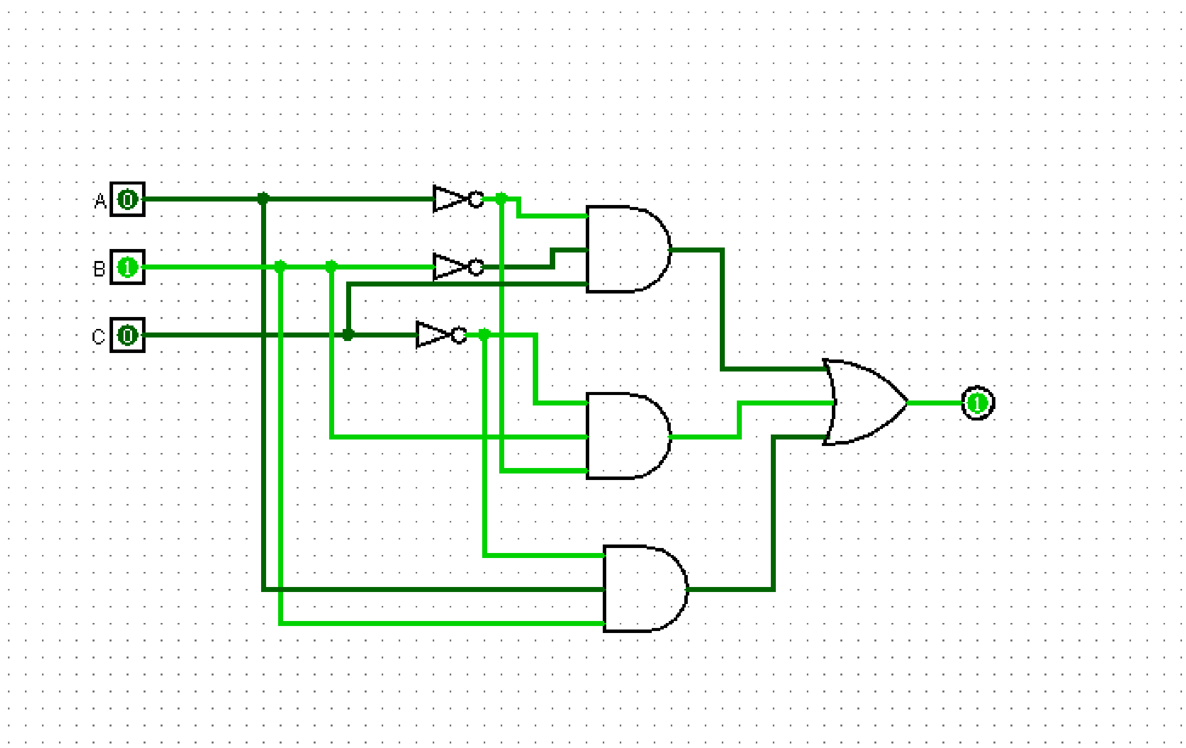
K-map simplification relies on the formation of groups. The groups include the cells that contain's 1,it can be horizontal or vertical,the groups may overlap as well as wrap and the groups should be as large as possible.

Step 4: Simplified Logical Expression

For each of the groups, we obtain logical expression in terms of the input-variables. Finally, we will combine all these group-wise logical expressions with OR gate to form the simplified Boolean equation from the K-map.

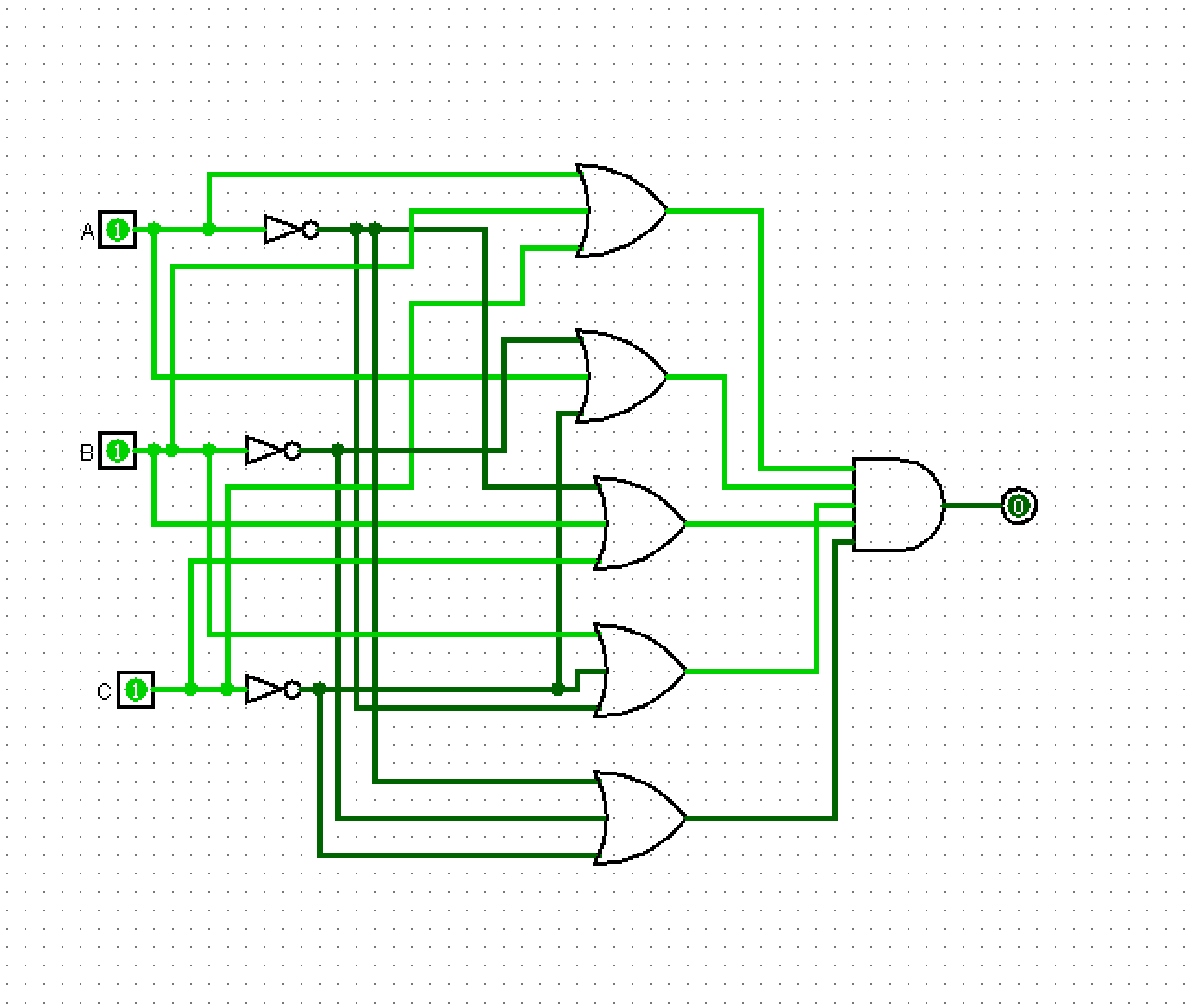
Circuit Diagrams

1st Canonical Form



FigureF.1:1st Canonical circuit diagram of the combinational circuit of TableF.1

2^{nd} Canonical Form



FigureF.1:2^{nd} Canonical circuit diagram of the combinational circuit of TableF.1

Figure F1: K-Maps

Wx/yz	Y`Z`=00	Y`Z=01	YZ=11	YZ`=10
W`X`=00	0	0	0	0
W`X=01	0	1	1	1
WX=11	X	X	X	X
WX`=10	1	1	X	X

Output A

A=W+XY+XZ

Wx/yz	Y`Z`=00	Y`Z=01	YZ=11	YZ`=10
W`X`=00	0	1	1	1
W`X=01	1	0	0	0
WX=11	X	X	X	X
WX`=10	0	1	X	X

Output B

B=X’Y+X’Z+XY’Z’

Wx/yz	Y`Z`=00	Y`Z=01	YZ=11	YZ`=10
W`X`=00	1	0	1	0
W`X=01	1	0	1	0
WX=11	X	X	X	X
WX`=10	1	0	X	X

Output C

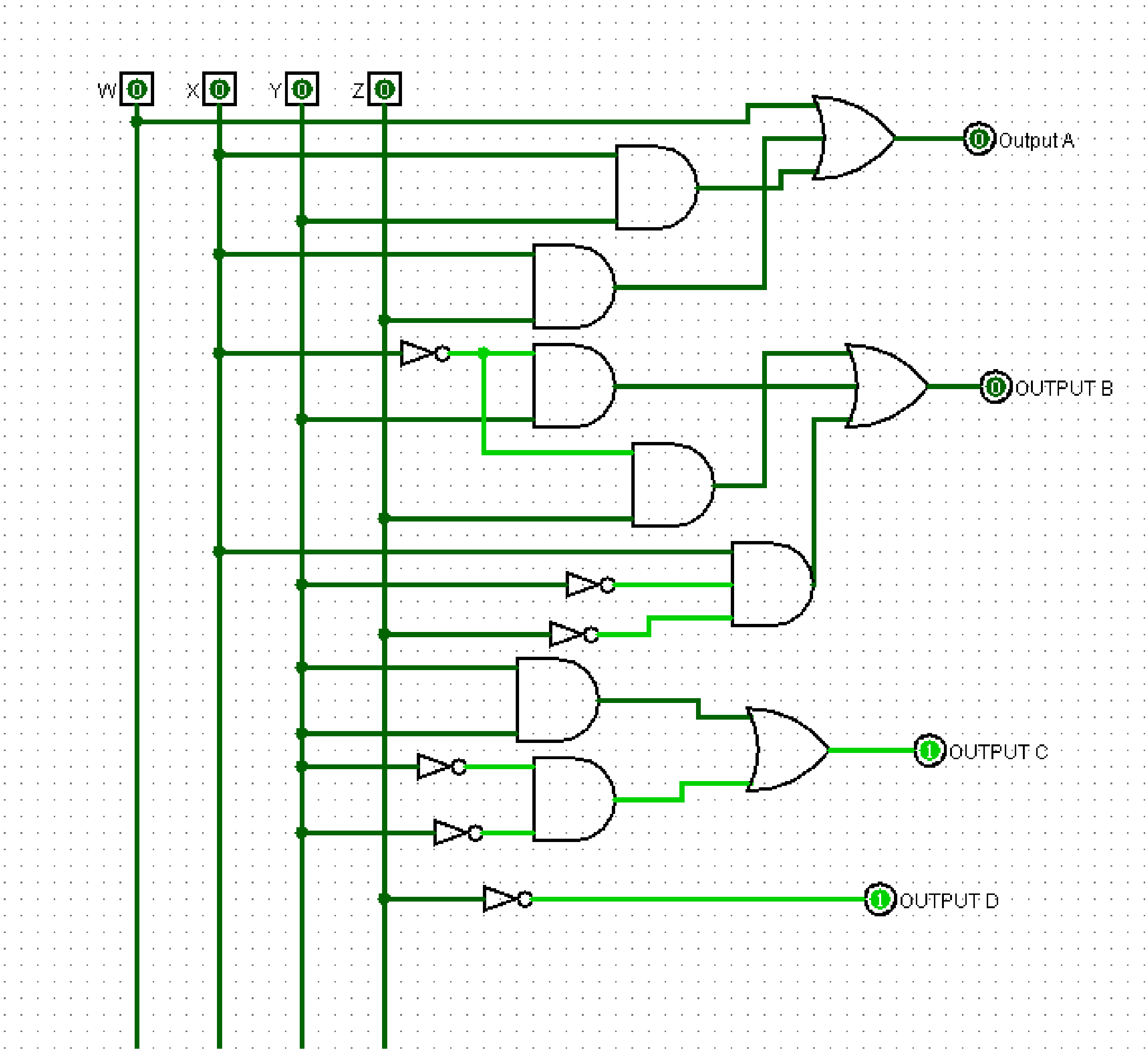
C=YZ+Y'Z'

Wx/yz	Y`Z`=00	Y`Z=01	YZ=11	YZ`=10
W`X`=00	1	0	0	1
W`X=01	1	0	0	1
WX=11	X	X	X	X
WX`=10	1	0	X	X

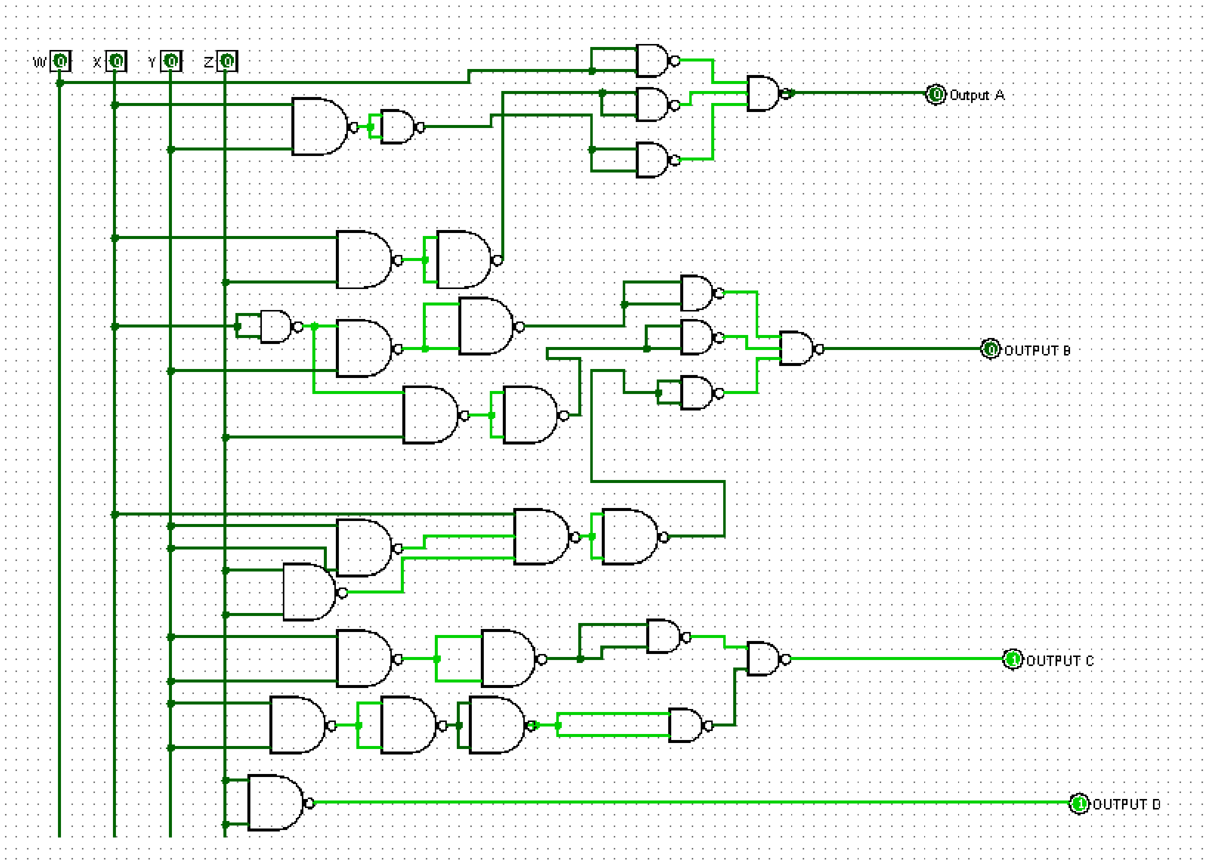
Output D

D=Z'

Figure F2: Minimal 1st canonical circuit



FigureF3: Minimal universal gate implementation of BCD



Truth Table

Input Reference	A B C	F	Min term	Max term
0	000	0		A+B+C
1	001	1	A’B’C	
2	010	1	A’B C’	
3	011	0		A+B’+C’
4	100	0		A’+B+C
5	101	0		A’+B+C’
6	110	1	A B C’	
7	111	0		A’+B’+C’

TableF.1 Truth table to a combinational circuit

	Shorthand Notation	Function
1st Canonical form	m=(1,2,6)	F=A’B’C+A’B C’+A B C’
2nd Canonical Form	m=(0,3,4,5,7)	F=(A+B+C)(A+B’+C’)+(A’+B+C)(A’+B+C’)(A’+B’+C’)

TableF.2 1st and 2nd canonical forms of the combinational circuit of TableF.1

TableF1: Truth table-BCD to Excess-3

DECIMAL	W	X	Y	Z	A	B	C	D
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Number of inputs bits:	**4**	**Input variables:**	**w, x , y ,z**
Number of outputs bits:	**4**	**Output variables:**	**A,B,C,D**

Table F2: System analysis

Discussion

The first topic of the lecture was Canonical Form. We studied the two types of canonical form, Sum of Minterms (SOM) and Product of Maxterms(POM) .Secondly, we learned how to write a Boolean function as SOM and POM from a truth table. The output of the inputs that are 1's are used for Sum of Minterms and the 0's are used to write the function in Product of Maxterm. If A=0, A'=1.We can also find the product of maxterm by complementing the minterms. For example: **Input->100**

->Sum of Minterm: A B' C'

->Product of Maxterm: (100)'= (011) =A'+B+C

To find Sum of Minterms and Product of Maxterm from truth table we can use the output the table gives For example: **Input->001, 010,110 gives Output->1**

->A' B' C, A' B C', A B C'

Sum of Product/Minterm= A' B' C+ A' B C'+ A B C'

Following that, we used this knowledge and completed the truth table F.1 of the combinational circuit where we wrote the Boolean equations for every input in terms of both Minterm and Maxterm. We further completed F.2 table where we wrote in Shorthand notation, which is denoted by m(input reference) and then wrote the whole Minterm and Maxterm function. Lastly, using the Boolean functions we drew each of the canonical circuits in Logisim.

In Lab-04 we learned about BCD, Excess-3 and K-Maps. In BCD, we studied how to write a number in BCD form and in Excess-3 we learned how the number is added with 3 and then converted to its binary form for the output. We proceeded on leaning about K-Maps. Firstly, we learnt how to draw the K-maps and how to determine the number of squares in the K-map. Secondly, we learned how to write Boolean function from K-map using groups, which must be multiples of two. The larger the groups the less inputs in the equation. The groups does not include the cells that contains 0, it can be horizontal or vertical but not diagonal, the cell number is determined by 2^(number of variables),the groups may overlap as well as wrap and the groups should be as large as possible. Using the knowledge, we completed the truth table F1 where we wrote decimal numbers 0-9 in BCD and excess 3 form. Then, we drew the K-maps using the values of Excess-3 which resulted in four outputs A, B, C and D. Using the group method we found the Boolean equations for each of the output and constructed the Minimal 1st canonical circuit. Lastly, we constructed Minimal universal gate of Excess-3 by replacing the basic gates with universal gate using Logisim.

.