North South University

Department of Electrical and Computer Engineering

LAB REPORT-04

Course name: Digital Logic Design

Course Code: 231

Experiment Number: 04

Experiment name: Combinational Logic Design

Experiment Date: 23rd March, 2021

Report Submission Date: 31st March, 2021

Section: 06

Group no: 04

Student Name	ID
Khalid Bin Shafiq	1911342642
Rafidul Islam	1912152642
Rashiqur Rahman Rifat	1911445652
Towsif Muhtadi Khan	1911576642
Remarks:	**Score:**

OBJECTIVE

- 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

Combinational Logic Circuits are memoryless digital logic circuits whose output at any instant in time depends only on the combination of its inputs. One of example of Combinational Logic Circuits is Binary Coded Decimal or BCD. BCD is simply the 4-bit binary code representation of a decimal digit with each decimal digit replaced in the integer and fractional parts with its binary equivalent. And The excess-3 code (or XS3) is a non-weighted code used to express code used to express decimal numbers. It is a self-complementary binary coded decimal (BCD) code and numerical system which has biased representation. It is particularly significant for arithmetic operations as it overcomes shortcoming encountered while using 8421 BCD code to add two decimal digits whose sum exceeds 9. Excess-3 arithmetic uses different algorithm than normal non-biased BCD or binary positional number system.

K-Map:

In many digital circuits and practical problems, we need to find expression with minimum variables. We can minimize Boolean expressions of 3, 4 variables very easily using Karnaugh Map as known as K-map without using any Boolean algebra theorems. K-map can take two forms Sum of Product (SOP) and Product of Sum (POS) according to the need of problem. K-map is table like representation but it gives more information than TRUTH TABLE. We fill grid of K-map with 0's and 1's then solves it by making groups.

BCD to Excess-3

The Excess-3 binary code is an example of a self-complementary BCD code. A self-complementary binary code is a code which is always complimented in itself. By replacing the bit 0 to 1 and 1 to 0 of a number, we find the 1's complement of the number. The sum of the 1'st complement and the binary number of a decimal is equal to the binary number of decimal 9. The process of converting BCD to Excess-3 is quite simple from other conversions. The Excess-3 code can be calculated by adding 3, i.e., 0110 to each four-digit BCD code of which decimal equivalent is 6. This decimal equivalent will become 9 by adding 3. BCD equivalent of 9 is 1001. And 1001 is Excess-3 code of BCD code 0110.

EQUIPMENT LIST

- Trainer Board
- 1 x IC 4073 Triple 3-input AND gates
- 1 x IC 4075 Triple 3-input OR gates
- 1 x IC 7404 Hex Inverters (NOT gates)
- 1 x IC 7400 2-input NAND gates
- 2 x IC 7408 2-input AND

CIRCUIT DIAGRAM

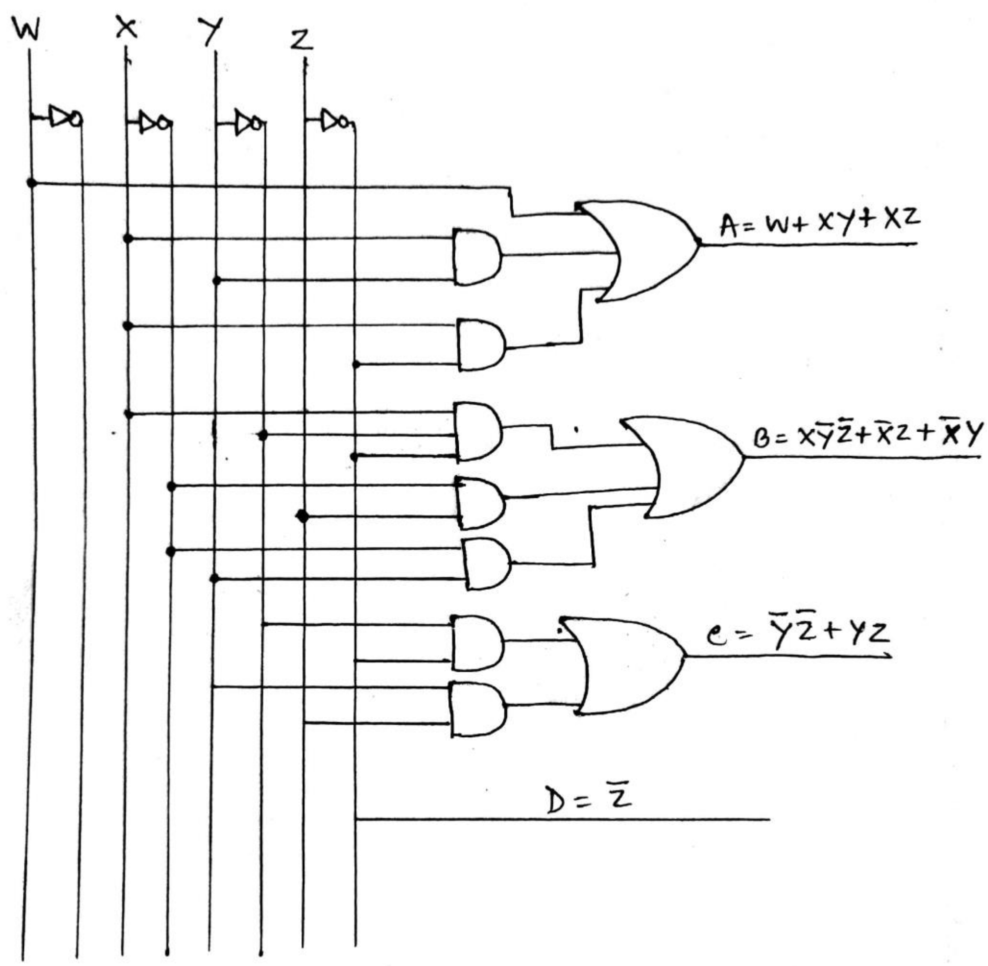

Figure F2: *Minimal 1st Canonical Circuit of BCD to Excess-3 converter*

)

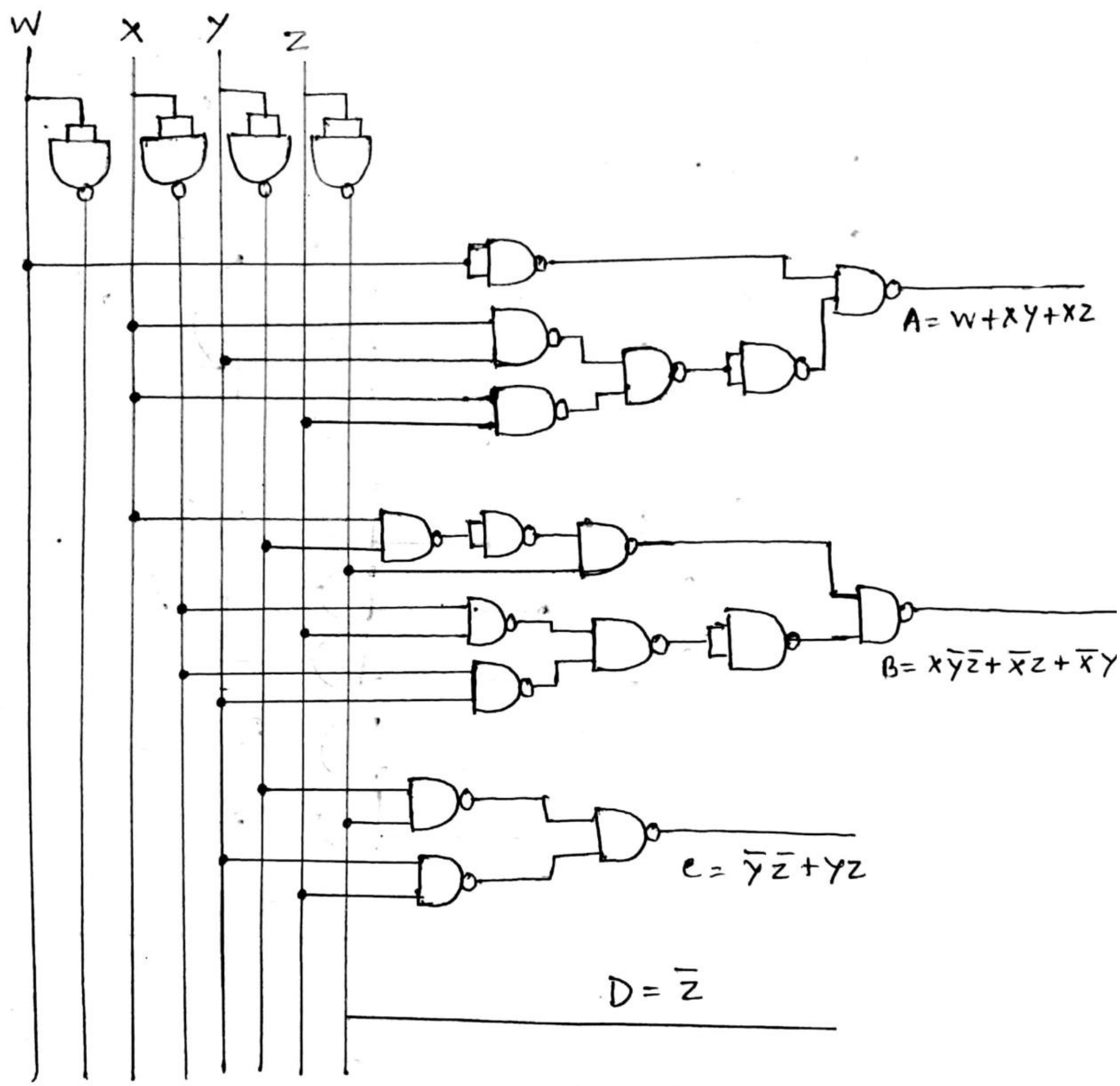

Figure F3: *Minimal Universal Gate implementation of BCD to Excess-3 converter*

DATA & TABLE

Decimal	Binary Coded Decimal (BCD)				Excess - 3			
Digit	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

Table F1: *Truth table - BCD to Excess-3*

Number of input bits:	4 bits	Input variables:	W,X,Y,Z
Number of output bits:	**4 bits**	**Output variables:**	**A,B,C,D**

Table F2: *System analysis*

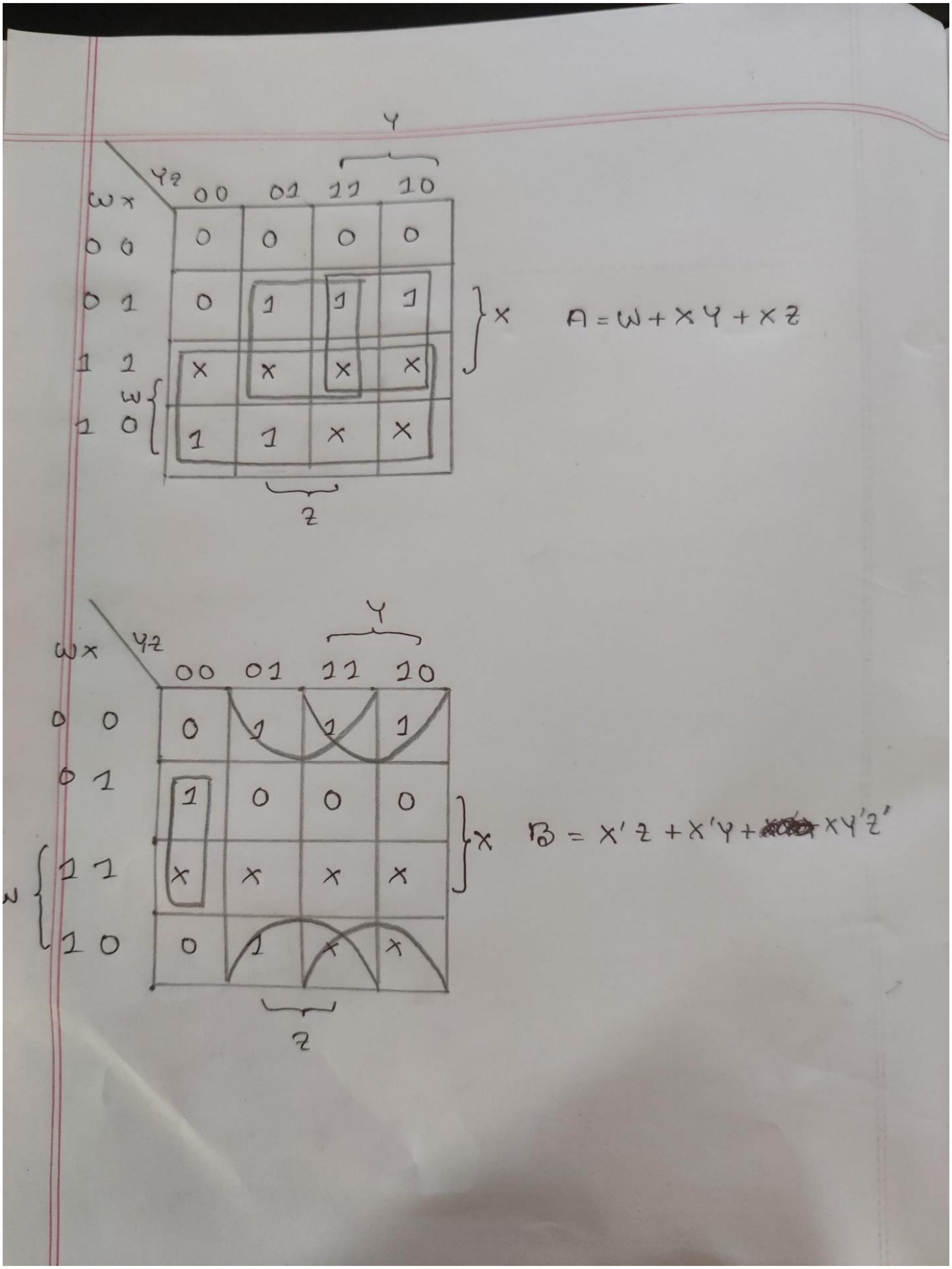

Figure F1: *K-Maps*

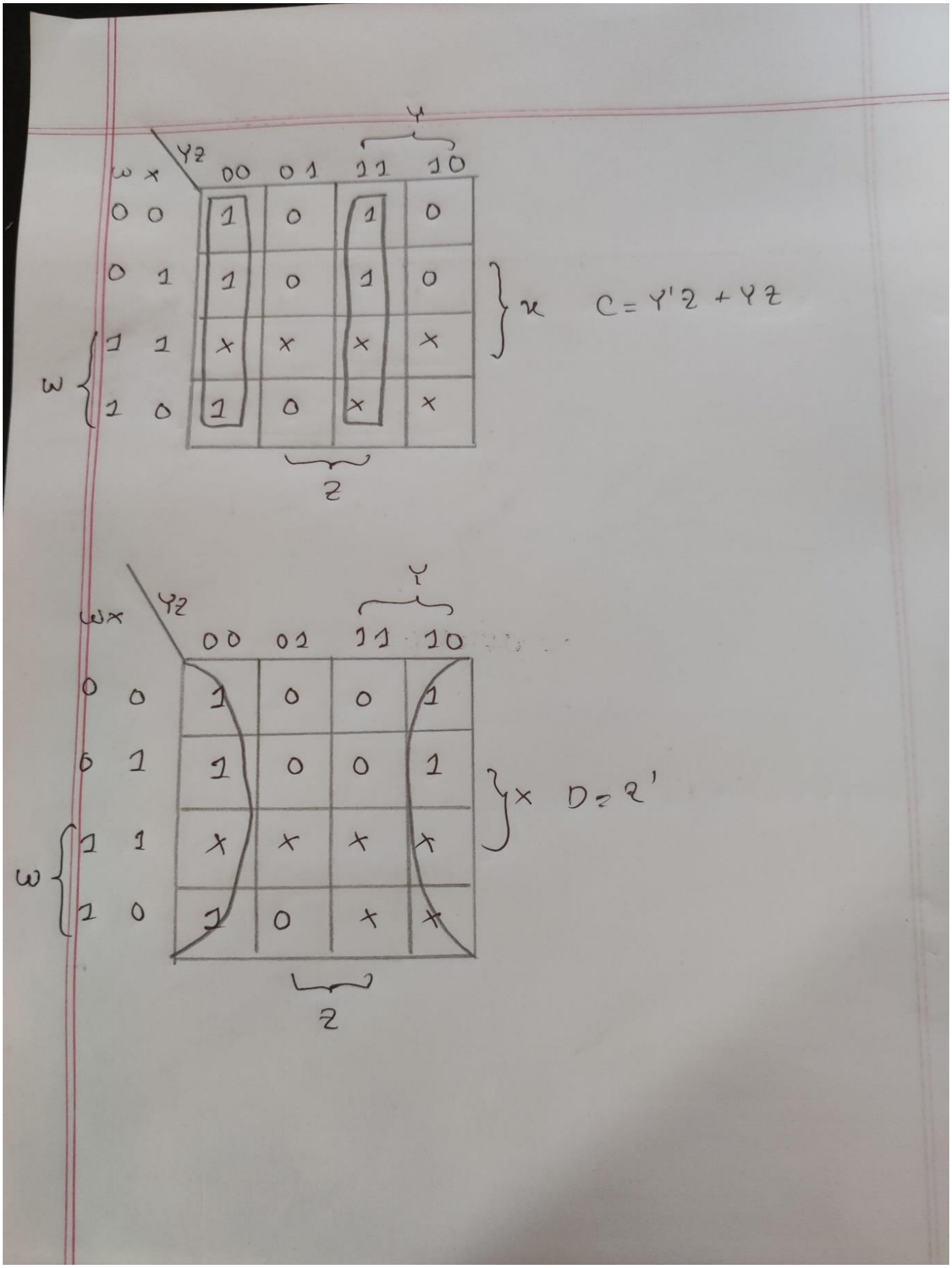

Figure F1: *K-Maps*

RESULT ANALYSIS AND DISCUSSION

The overall lab experiment was about performing the BCD to Excess-3 conversion using Combinational Logic Circuits.

At the beginning of the experiment, we had to create truth table with Excess-3 values for corresponding values of BCD code. Then we needed to draw K-map to find minimized form of four output functions.

Then we implemented combinational logic circuits. For performing BCD to Excess-3 conversion, we implemented minimal AND-OR-NOT gates combinational logic design, minimal universal gates combinational logic design.

From the experiment, we learned about different ways of implementation of combinational logic circuit design for performing BCD to Excess-3 conversion.

CONTRIBUTION

NAME	CONTRIBUTION IN
Khalid Bin Shafiq	Circuit Diagram
Rafidul Islam	Result Analysis and Discussion
Rashiqur Rahman Rifat	Theory
Towsif Muhtadi Khan	Data & table, Coordinator

CSE231L
Class Assignment 04

Task-1: Complete the data tables.

Task -2: Do the K-mapping for the required design in your notebook and attach their pictures here.

Task -3: Design the required BCD to Excess-3 converters in the logisim and attach their screenshots below.

Experimental Data

Decimal Digit	Binary Coded Decimal (BCD)				Excess-3			
	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

Table F1: Truth table - BCD to Excess-3

Number of inputs bits:	4 bits	**Input variables:**	W,X,Y, Z
Number of outputs bits:	4 bits	**Output variables:**	A,B,C D

Table F2: System analysis

CSE231L
Class Assignment 04

0	0	0	0
0	1	1	1
x	x	x	x
1	1	x	x

A=W+XY+XZ

0	1	1	1
1	0	0	0
x	x	x	x
0	1	x	X

B=X’Z+X’Y+XY’Z’

1	0	1	0
1	0	1	0
X	X	X	X
1	0	X	x

C=Y’Z+YZ

1	0	0	1
1	0	0	1
X	x	x	X
1	0	x	x

D=Z’

Figure F1: K-Maps

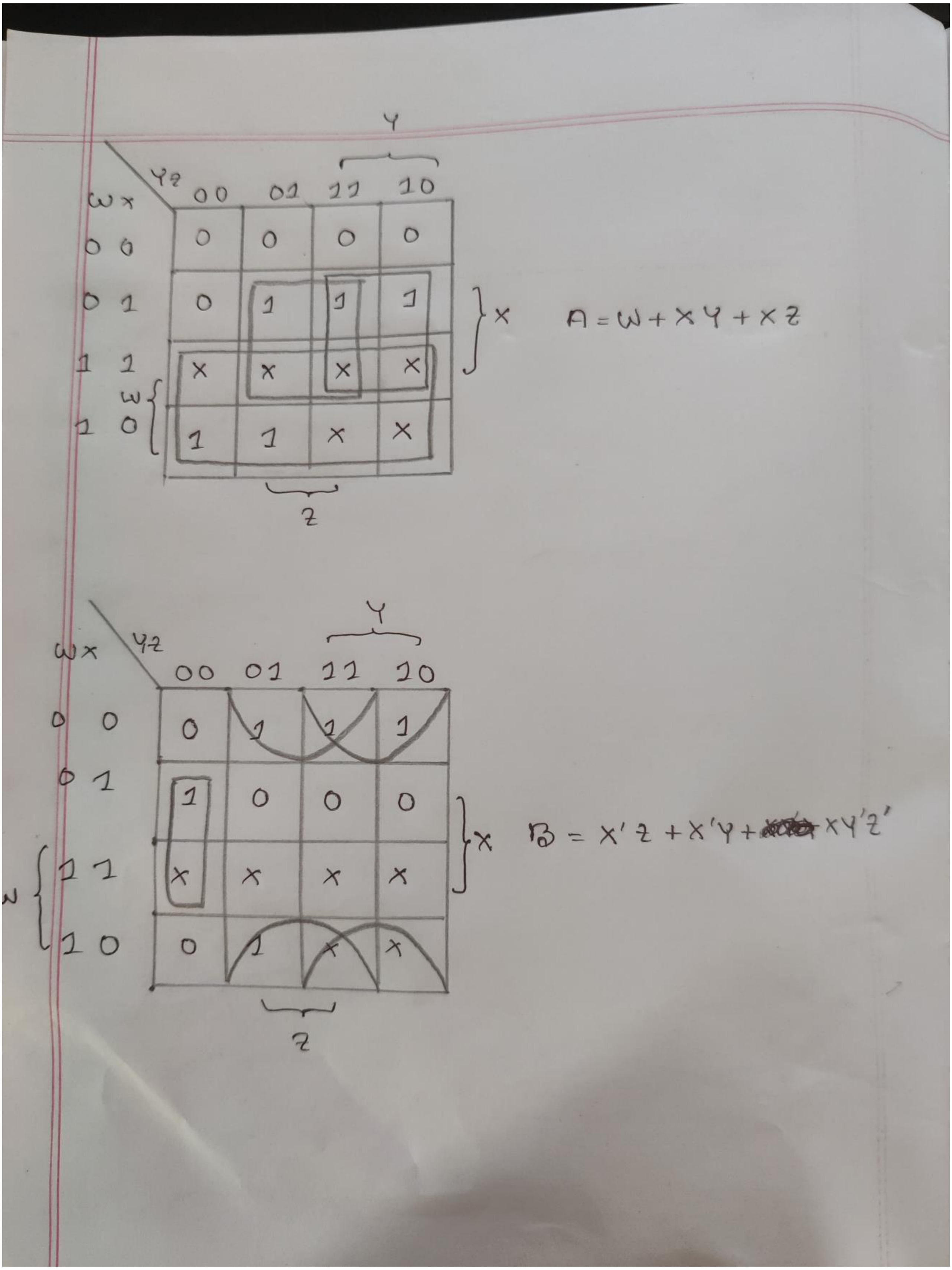

Figure F1: K-Maps

CSE231L
Class Assignment 04

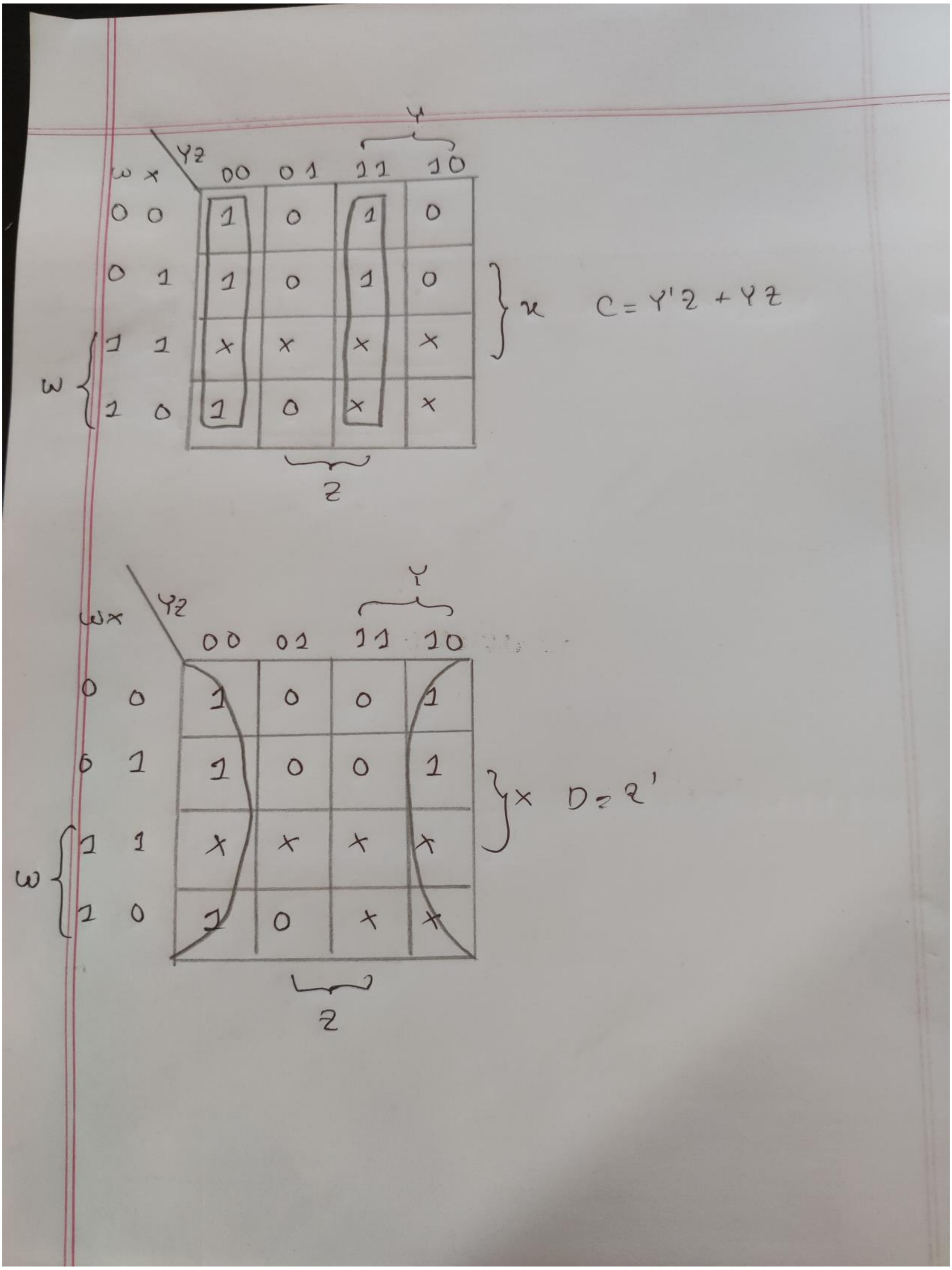

Figure F1: K-Maps

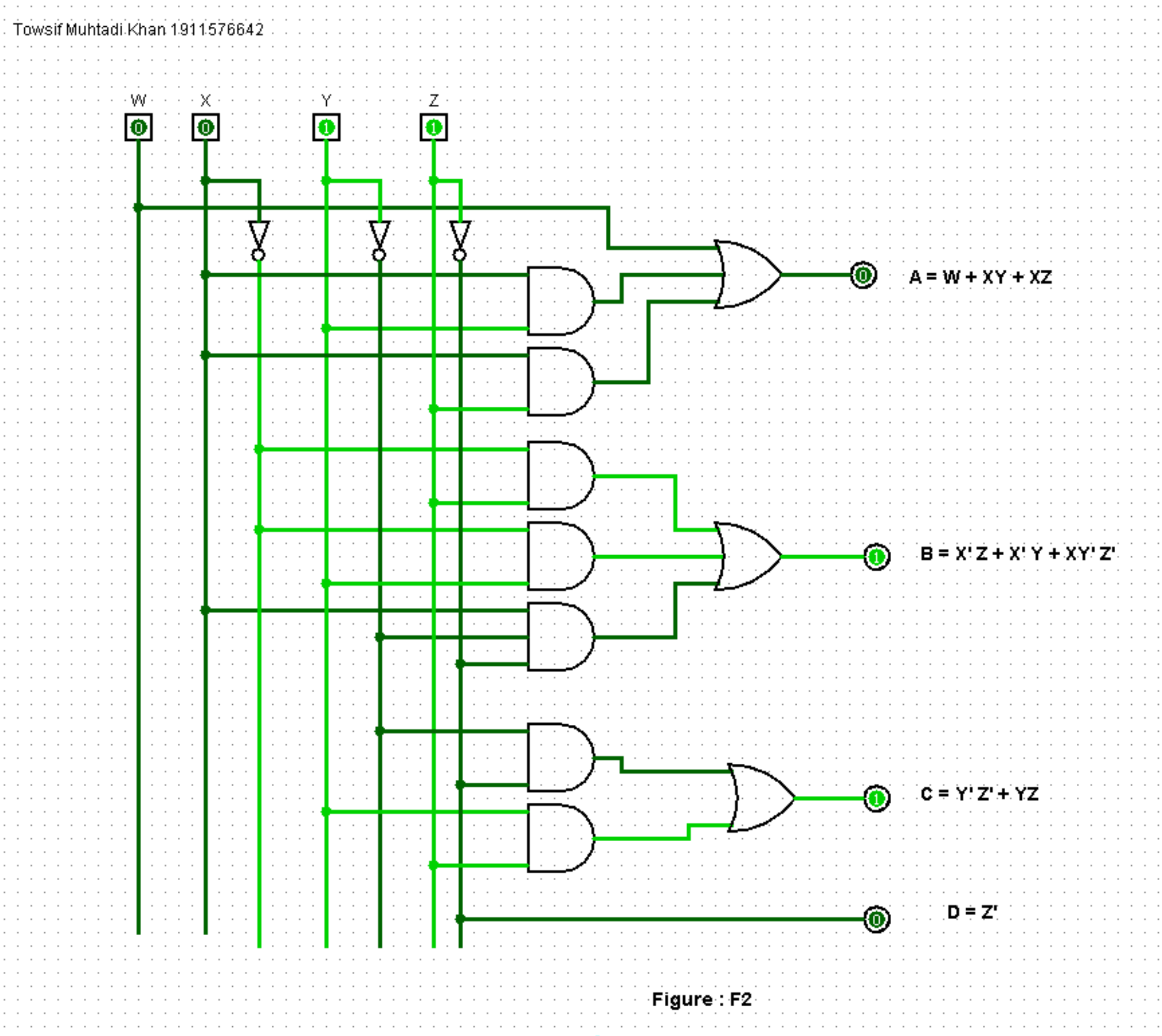

Figure F2: Minimal 1st canonical circuit of BCD to Excess-3 converter

Towsif Muhtadi Khan
1911576642