Experiments Name: Combinational Logic Design (K-Map), BCD to Excess-3 Converter

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.

Apparatus:

- 1x IC 7404 Hex Inverters (NOT gates).
- 3x IC 7411 Triple 3-input AND gates.
- 1x IC 4075 Triple 3-input OR gates.
- 1x IC 7432 Quadruple 2-input OR gates.
- 3x IC 7400 Quadruple 2-input NAND gates.
- 1x IC 7410 Triple 3-input NAND gates.
- Trainer Board
- Wires

Theory:

BCD (Binary Coded Decimal) is a way of representing decimal numbers using binary digits. In BCD, each decimal digit is represented by a four-bit binary code, allowing each BCD digit to have one of 16 possible values (0000 to 1001). BCD is commonly used in digital systems where decimal arithmetic is required, such as in calculators, counters, and displays.

Excess-3 is a decimal code that represents decimal numbers using binary digits by adding 3 to the corresponding binary value of each decimal digit. xcess-3 is commonly used in digital systems where decimal arithmetic is required, such as in calculators and digital displays.

A K-map is a diagram made up of squares, with each square representing one minterm of the function that is to be minimized. In fact, the map presents a visual diagram of all possible ways a function may be expressed in standard form. By recognizing various patterns, it is possible to derive alternative algebraic expressions for the same function, from which the simplest can be selected.

Example:

W	X	Y	Z	A
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1

Table B1

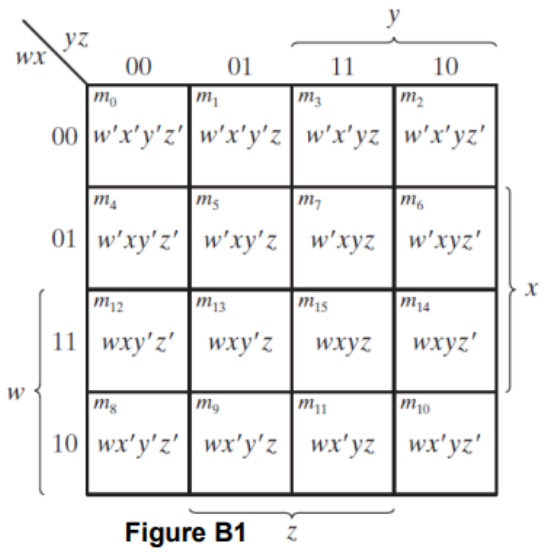


Figure B1

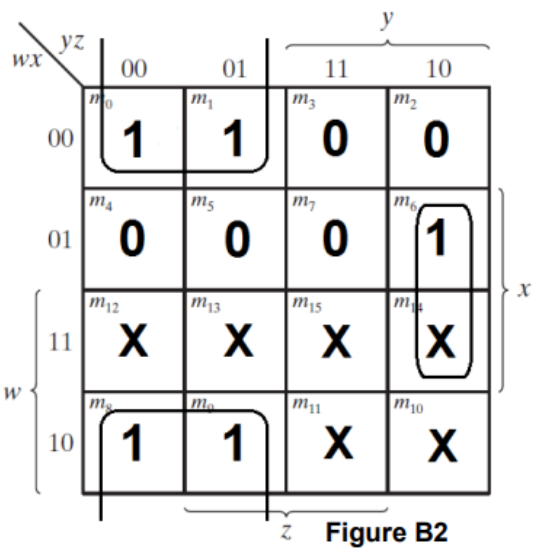
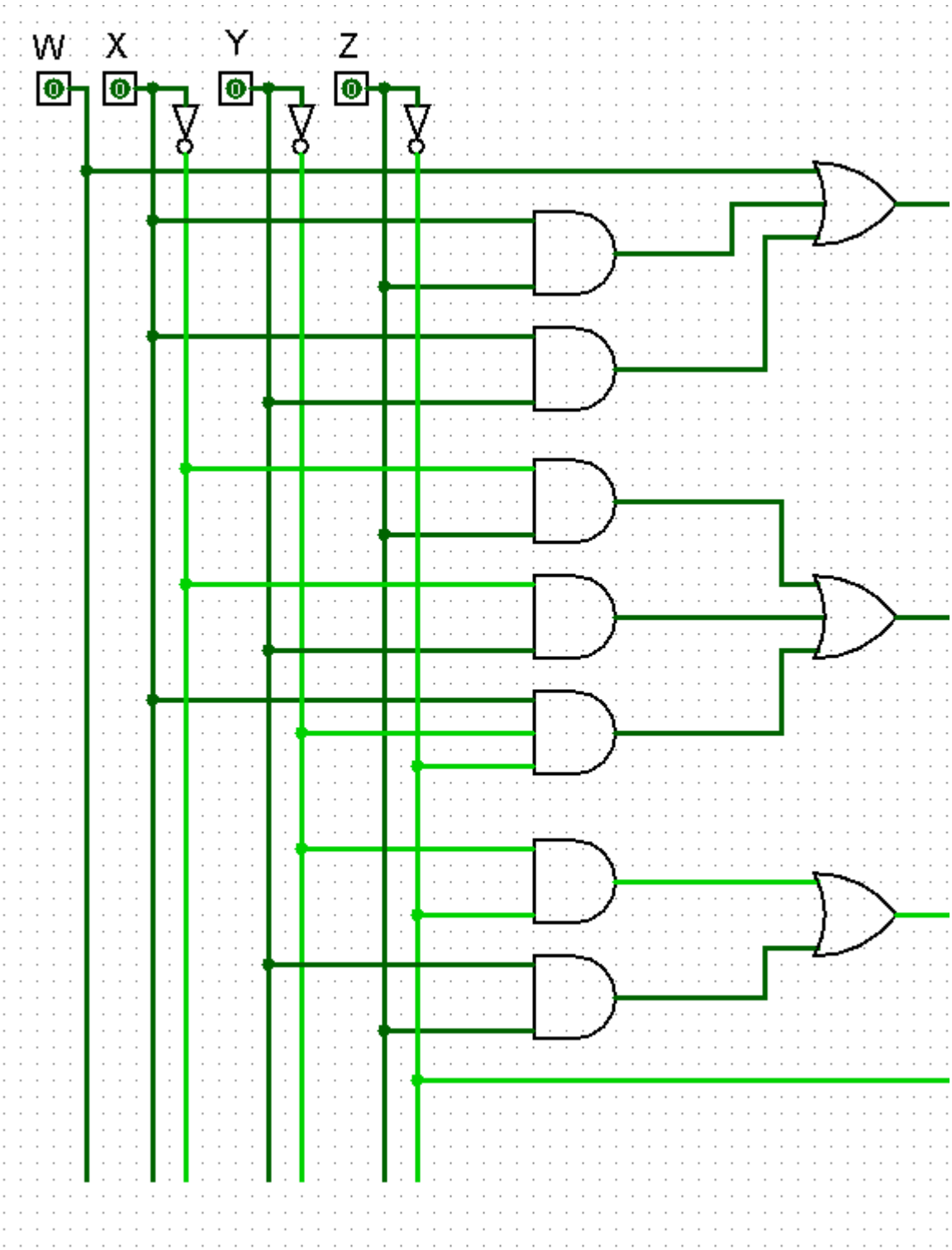


Figure B2

Figure B1 shows the minterm positions on the K-map for 4 input variables and 1 output variable.
Table B1 shows the truth table for the function A = WX + XYZ' + WZ + X'Y' where WXYZ goes from binary 0 to 9.
Figure B2 shows the K-map for the function. We can use the K-map to minimize the function to A = X'Y' + XYZ'

Circuit Diagram:



Experimental Procedure:

01. First we complete the truth table of Table F1 for the BCD to Excess-3 converter.
02. Then, we complete the K-Maps for output of A, B, C, and D to find the minimal 1st canonical functions.
03. After that we draw the circuit diagram with pin configurations in Figure F2. Then, we start to implement this circuit according to the drawing.
04. First, we implement for A and test it then B and so on. After all we connect 4 outputs at once and test it again.
05. Then we start to minimize the circuit diagram using universal NAND gate. First, we draw the minimal Circuit in Figure F3 with pin configurations. Then we start to implement it.
06. Here, we also implement and test it one by one. After complete we connect 4 outputs at once and again test it.

Simulation:
Attached.

Experimental Data Table:

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	Input variables:	4
Number of outputs bits:	4	Output variables:	4

Table F2: System analysis

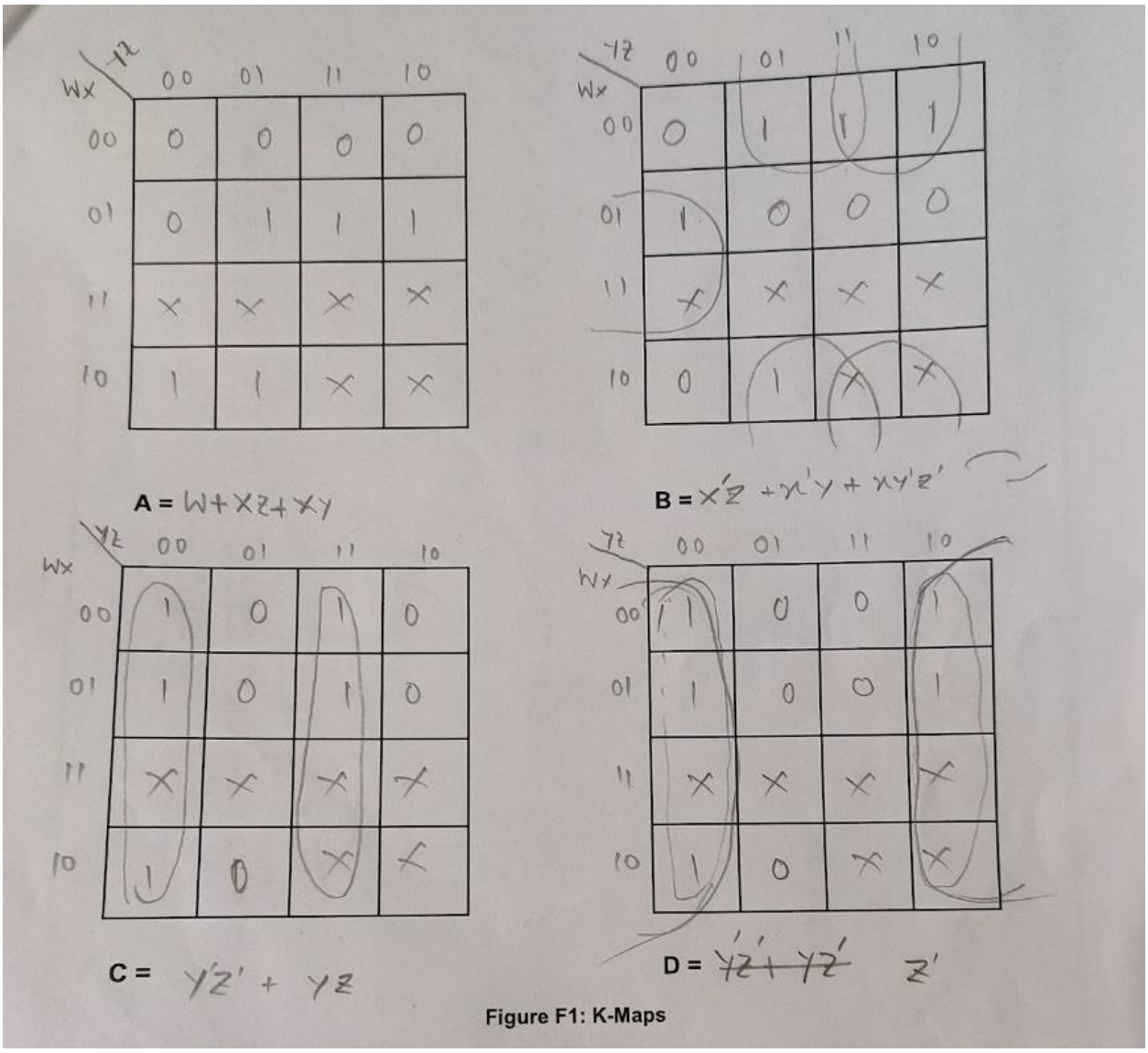


Figure F1: K-Maps

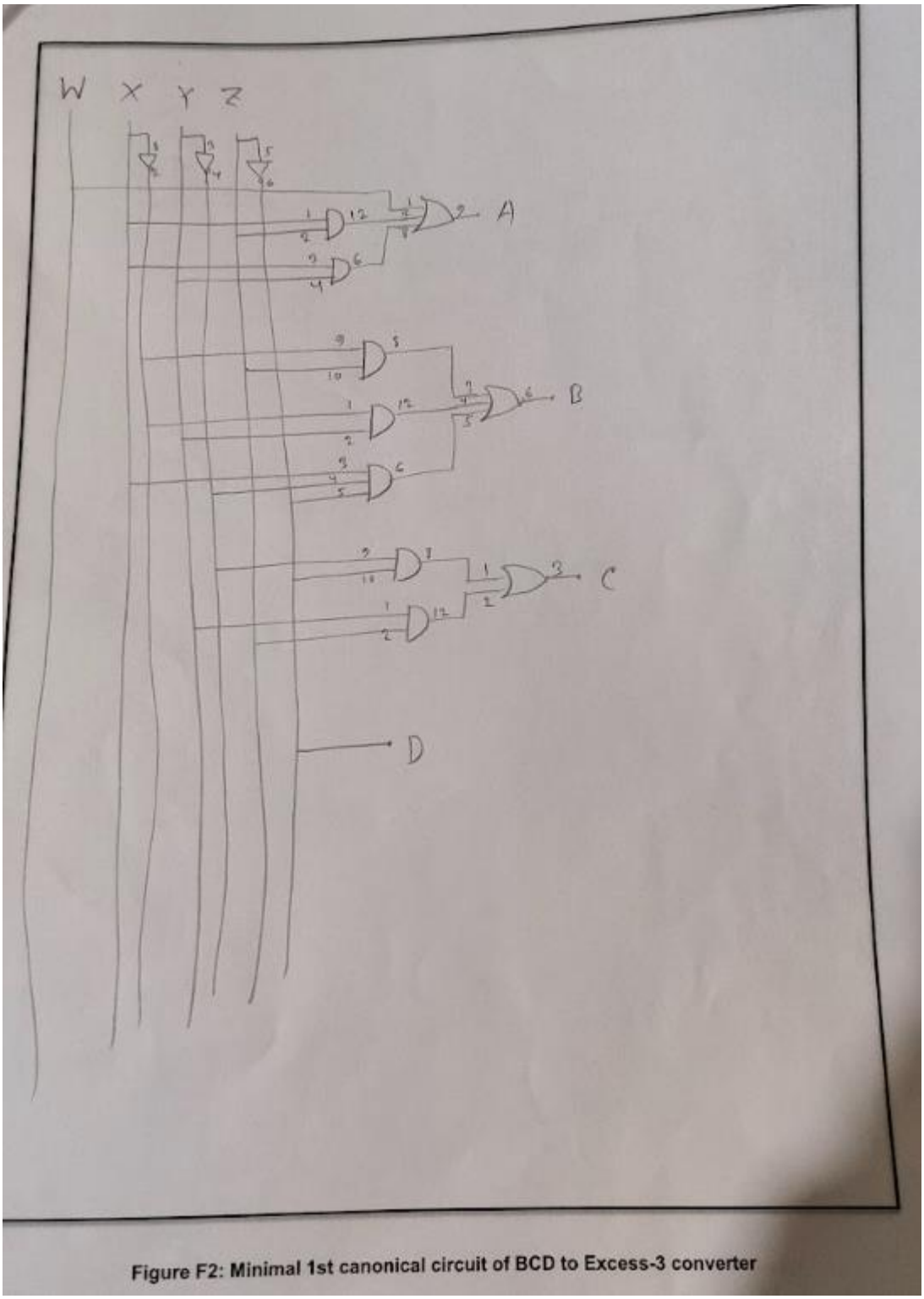


Figure F2: Minimal 1st canonical circuit of BCD to Excess-3 converter

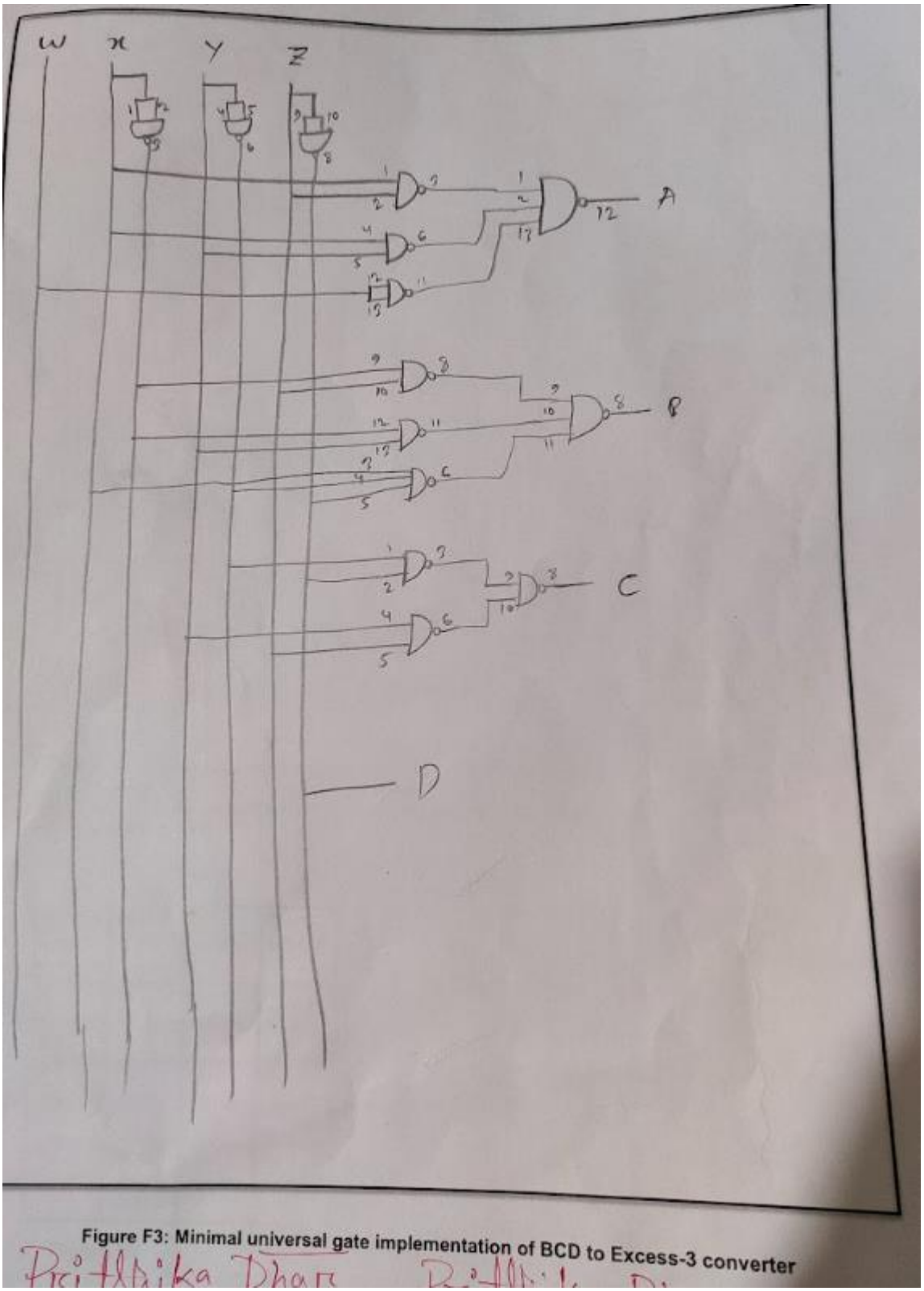


Figure F3: Minimal universal gate implementation of BCD to Excess-3 converter

Prithika Dhar

Results:
After following all the procedures, we successfully build the circuit of BCD to Excess-3 Conversion. And we also make it by using universal NAND gates, which reduces the ICs count.

Questions and Answers (Q/A):

01. Draw the IC diagram for output B from Figure F2
02. Simulation Attached.

Discussion:
From this experiment, we learn a lot about the K-Maps, BCD, and Excess-3. We learn how to minimize a combinational circuit using K-maps. We can now design a minimal combinational logic system from specification to implementation. In this experiment, when we do the first part, we face many problems. NOT IC was not working correctly; at the end, three inputs OR IC was not working with two inputs. Then we replace the three inputs OR IC with two inputs OR IC and complete the circuit. In the second part, we don't face any problems. In short, we learn a lot about the K-Maps.