National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: _____ **Dated:** _____

Semester: _____ **Section:** _____

Group No.:

EE-221: Digital Logic Design

Lab 10: Excess-3 to Gray Code Conversion using Nand Gates

		PLO4/CLO4	PLO4/CLO4	PLO5/CLO5	PLO8/CLO6	PLO9/CLO7	
Name	**Reg. No**	**Viva / Lab Performance** 5 Marks	**Analysis of data in Lab Report** 5 Marks	**Modern Tool Usage** 5 Marks	**Ethics and Safety** 5 Marks	**Individual and Team Work** 5 Marks	**Total marks Obtained** 25 Marks
AILYA ZAINAB	**523506**						
IMAN NAEEM	**525378**						
LAIBA NASIR	**510419**						
LUQMAN SHEHZAD	**507599**						

Lab 10: Excess-3 to Gray Code Conversion using Nand Gates

This Open ended Lab has been divided into two parts:

In first part you are required to **design** and **implement** a Excess-3 to gray code converter.

The next part is the Verilog Modeling and **Simulation** of the Circuit you implemented in you first part.

Objectives:

- ✓ Understand steps involved in design of combinational circuits
- ✓ Understand binary codes for decimals and their hardware realization
- ✓ Write code for combinational circuits using Verilog Gate Level Modeling
- ✓ Design a circuit in Verilog by calling different modules

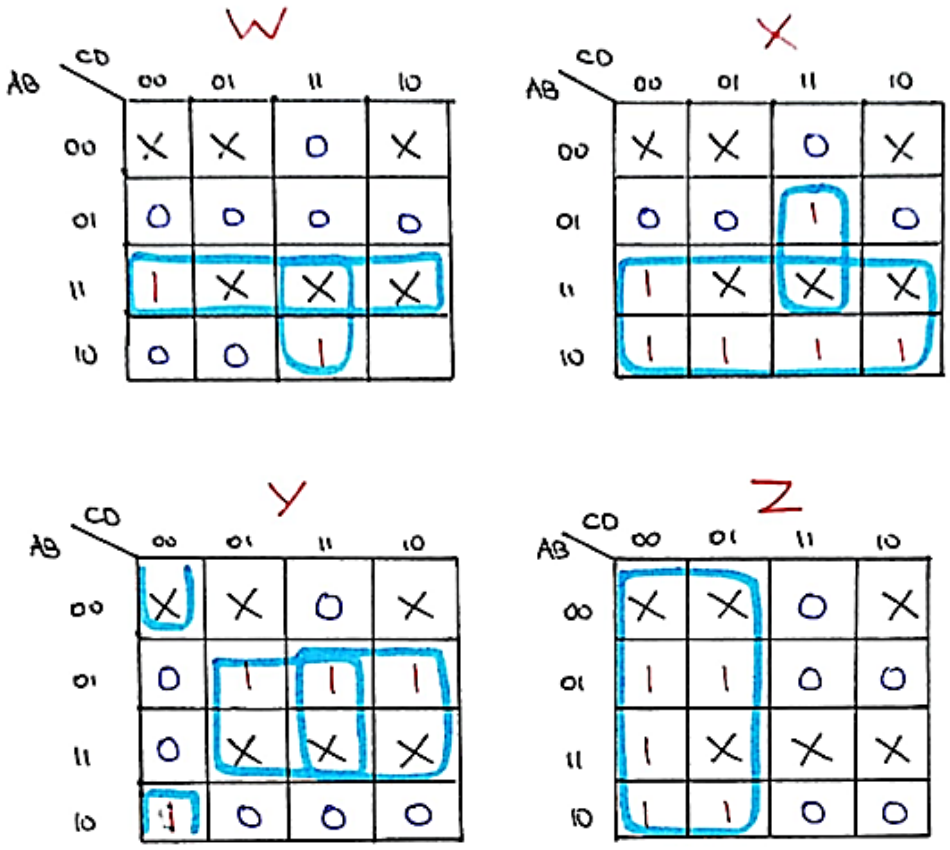
Lab Instructions

- ✓ This lab activity comprises three parts, namely Pre-lab, Lab tasks, and Post-Lab Viva session.
- ✓ The lab report will be uploaded on LMS three days before scheduled lab date. The students will get hard copy of lab report, complete the Pre-lab task before coming to the lab and deposit it with teacher/lab engineer for necessary evaluation.
- ✓ The students will start lab task and demonstrate design steps separately for step-wise evaluation(course instructor/lab engineer will sign each step after ascertaining functional verification).
- ✓ Remember that a neat logic diagram with pins numbered coupled with nicely patched circuit will simplify trouble-shooting process.
- ✓ After the lab, students are expected to unwire the circuit and deposit back components before leaving.
- ✓ The students will complete lab task and submit complete report to Lab Engineer before leaving lab.
- ✓ There are related questions at the end of this activity. Give complete answers.

Pre-Lab Tasks (4)

1. In the lab you would be implementing an Excess-3 to gray code converter. Make a truth table for both the codes by filling in the following tables and Simplify the expressions for W,X,Y,Z in terms of A,B,C,D.(Use backside of the page if necessary). Use unused combinations as don't care conditions. Hint, u can take help from http://engineeringproblemsandanswers.blogspot.com/2015/03/logic-circuits-1-digit-decimal-in-excess-3-to-gray-code-converter.html (1)

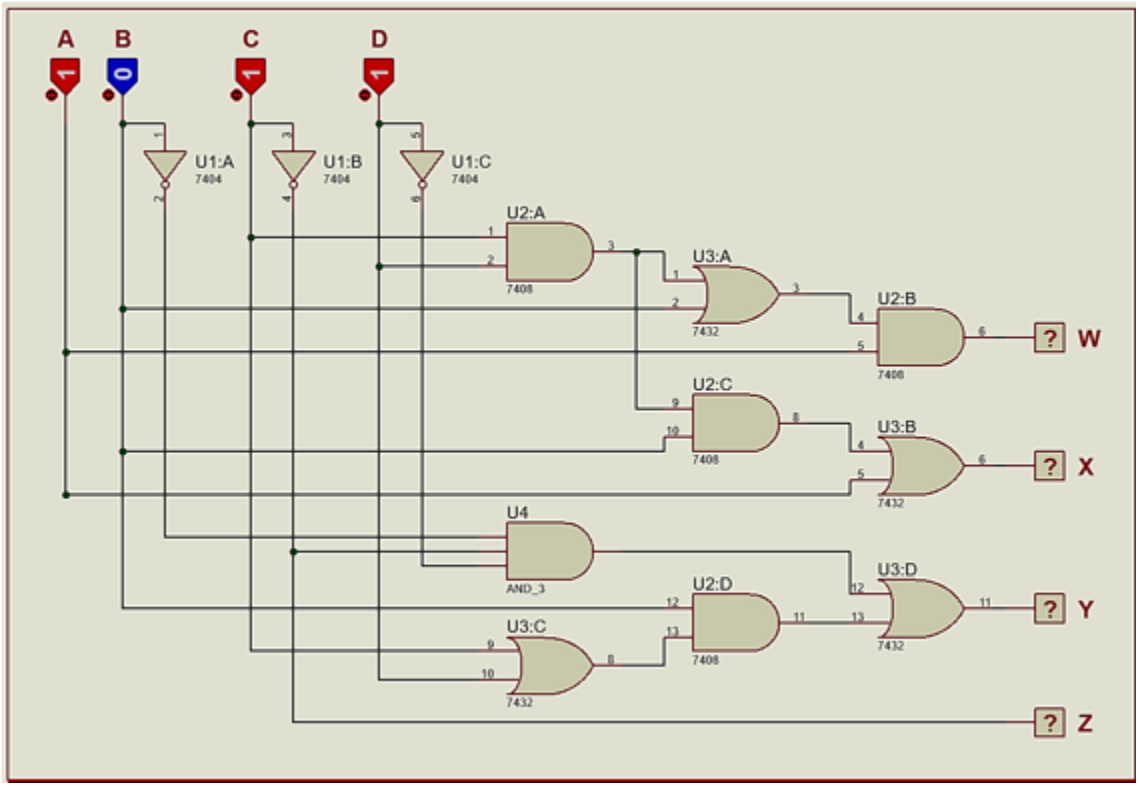
Dec	XS - 3				Gray Code			
	A	B	C	D	W	X	Y	Z
	0	0	0	0	X	X	X	X
	0	0	0	1	X	X	X	X
	0	0	1	0	X	X	X	X
0	0	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0	1
2	0	1	0	1	0	0	1	1
3	0	1	1	0	0	0	1	0
4	0	1	1	1	0	1	1	0
5	1	0	0	0	0	1	1	1
6	1	0	0	1	0	1	0	1
7	1	0	1	0	0	1	0	0
8	1	0	1	1	1	1	0	0
9	1	1	0	0	1	1	0	1
	1	1	0	1	X	X	X	X
	1	1	1	0	X	X	X	X
	1	1	1	1	X	X	X	X



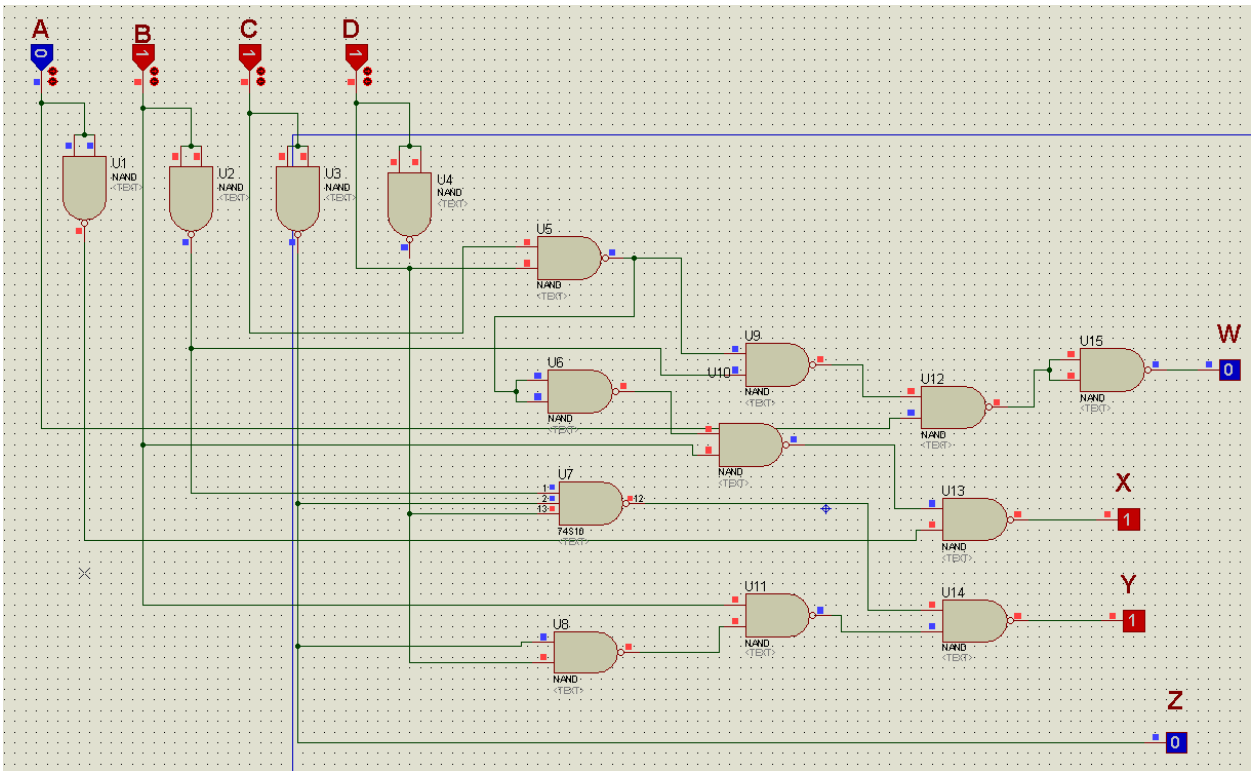
Simplified Expressions

W	**AB + ACD**	**A (B + CD)**
X	**A + BCD**	**A + B (CD)**
Y	**B'C'D' + BD + BC**	**B'C'D' + B (C + D)**
Z	**C'**	**C'**

2. Draw the logic diagram for Excess-3 to gray code converter using AND, OR and NOT gates in the space provided below. You can use 2,3,4 input gates if required. (1)



3. Draw the logic diagram for Excess-3 to gray code converter using only NAND gates in the space provided below, you can use 2,3,4 input Nand gates if required. Simulate in Proteus as well. (2)

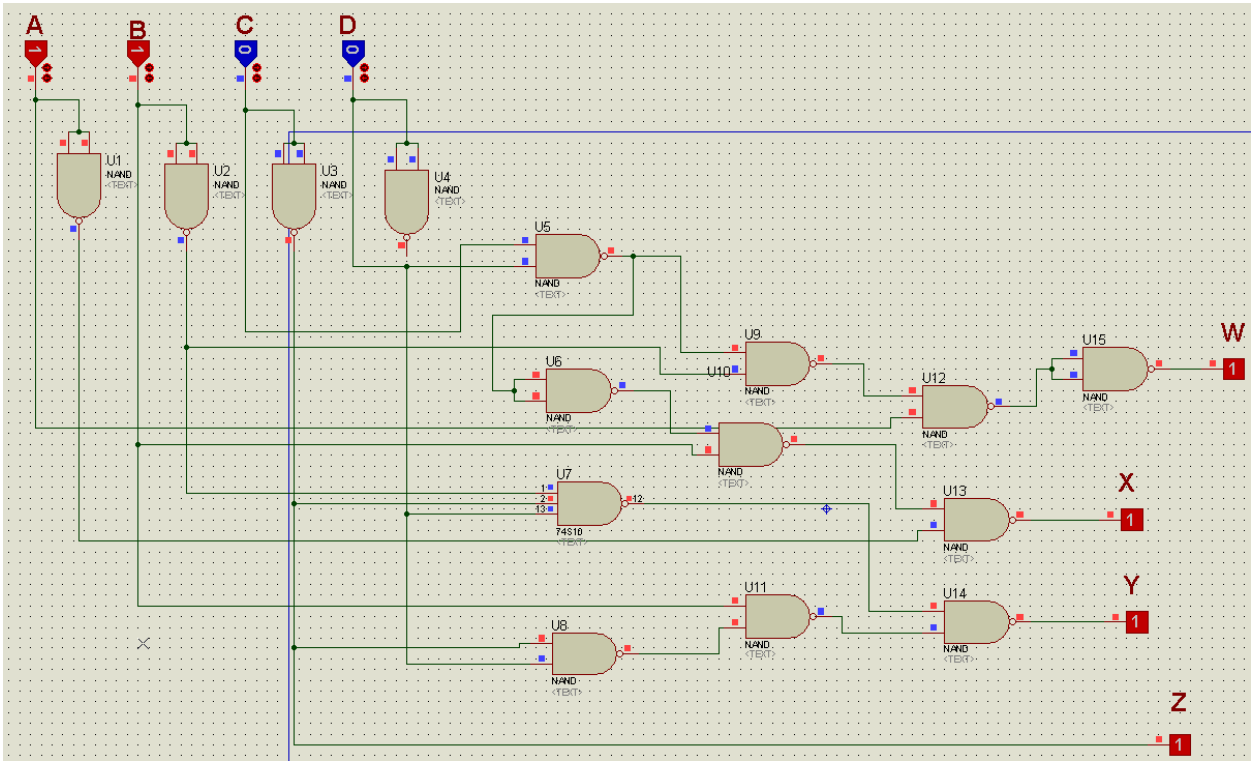

A
B
C
D
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
W
X
Y
Z
NAND
74S10

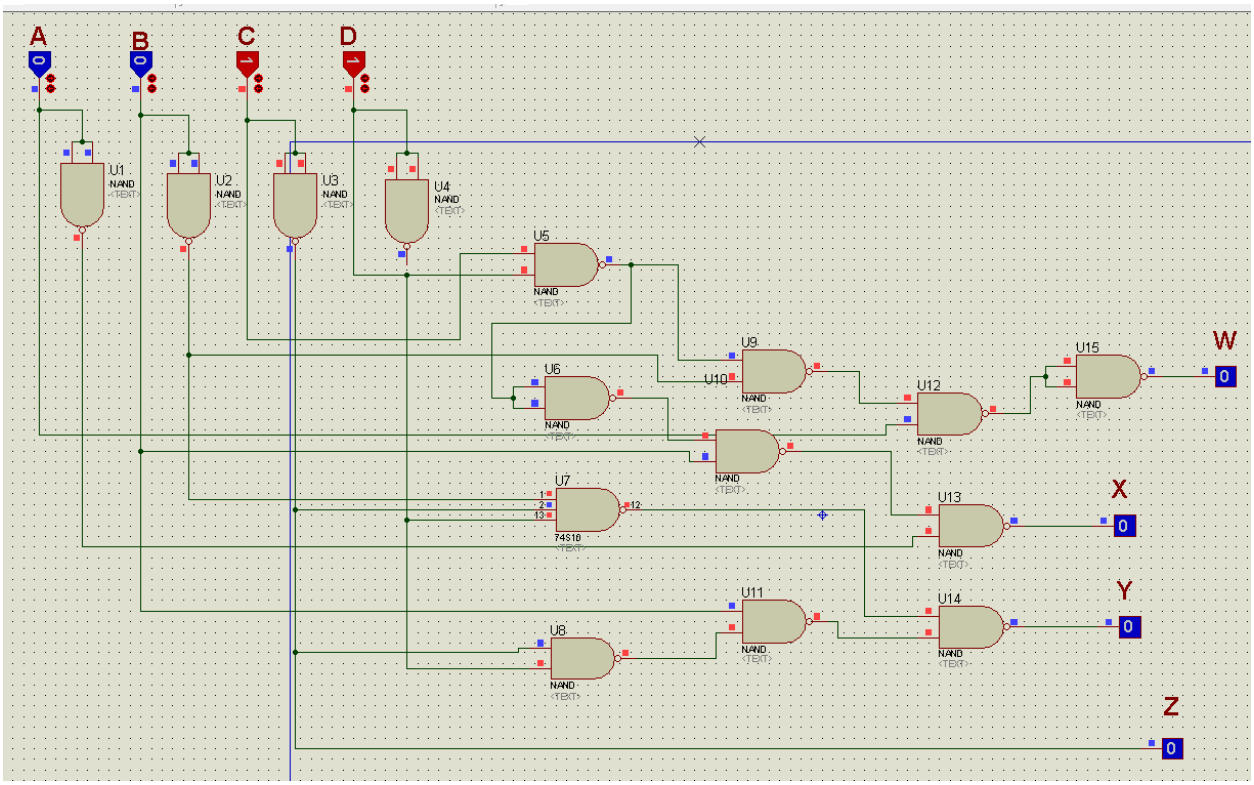
Lab Tasks: (To be completed in the lab)

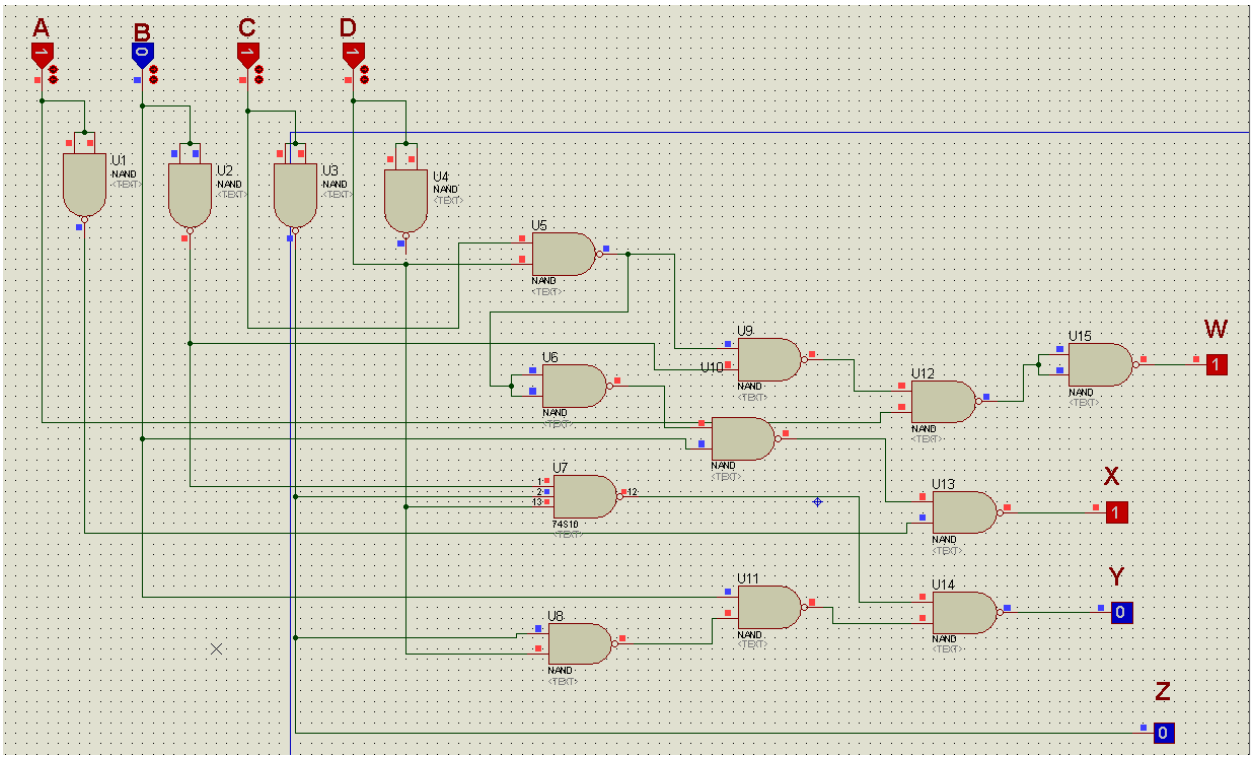
Lab Task 1: (4)

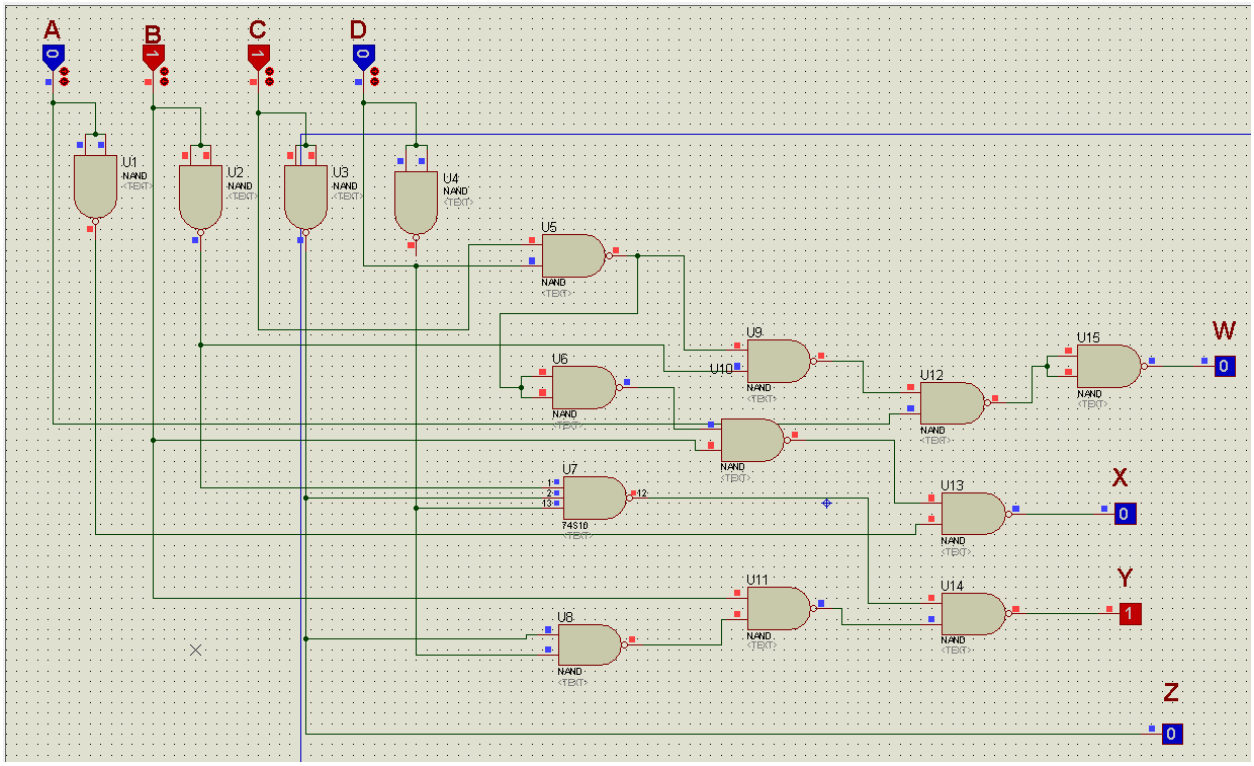
Implement Excess-3 to gray code converter using only NAND gates on hardware. Paste the complete circuit diagram, depicting hardware results.

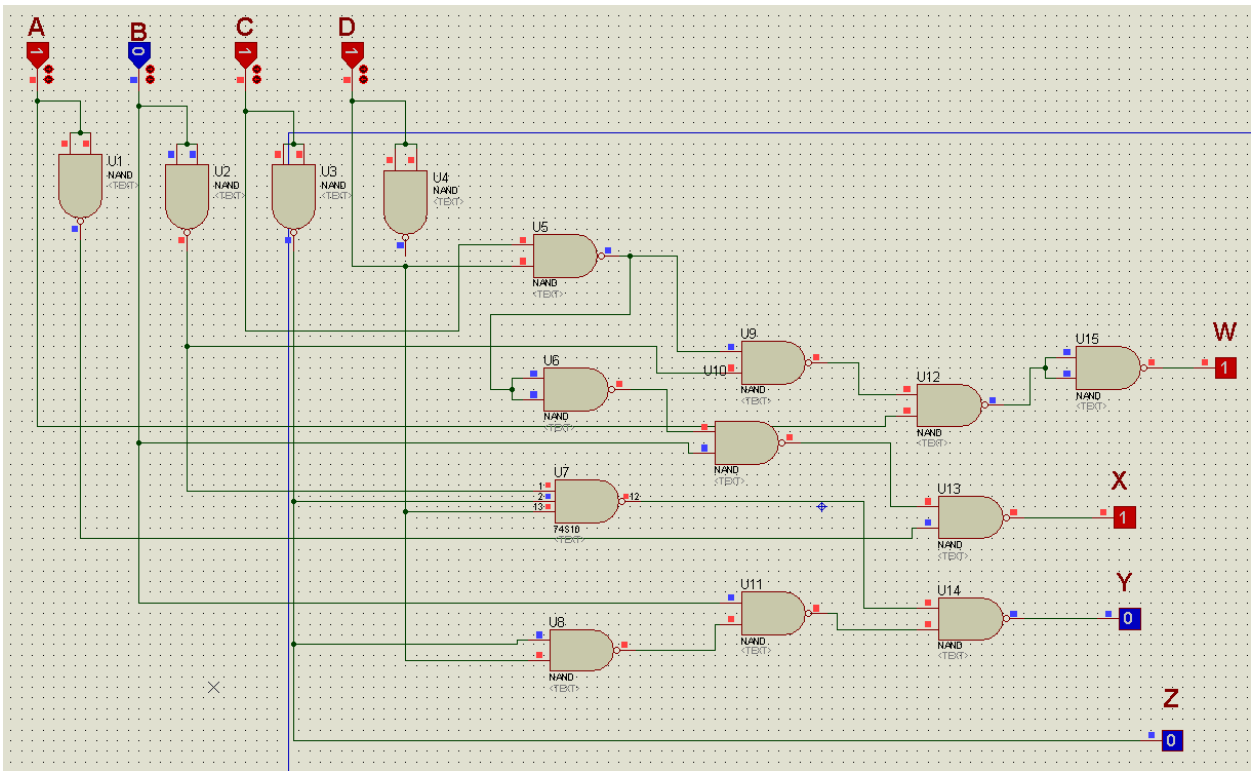
Dec	XS - 3				Gray Code (Hardware)			
	A	B	C	D	W	X	Y	Z
0	0	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0	1
2	0	1	0	1	0	0	1	1
3	0	1	1	0	0	0	1	0
4	0	1	1	1	0	1	1	0
5	1	0	0	0	0	1	1	1
6	1	0	0	1	0	1	0	1
7	1	0	1	0	0	1	0	0
8	1	0	1	1	1	1	0	0
9	1	1	0	0	1	1	0	1

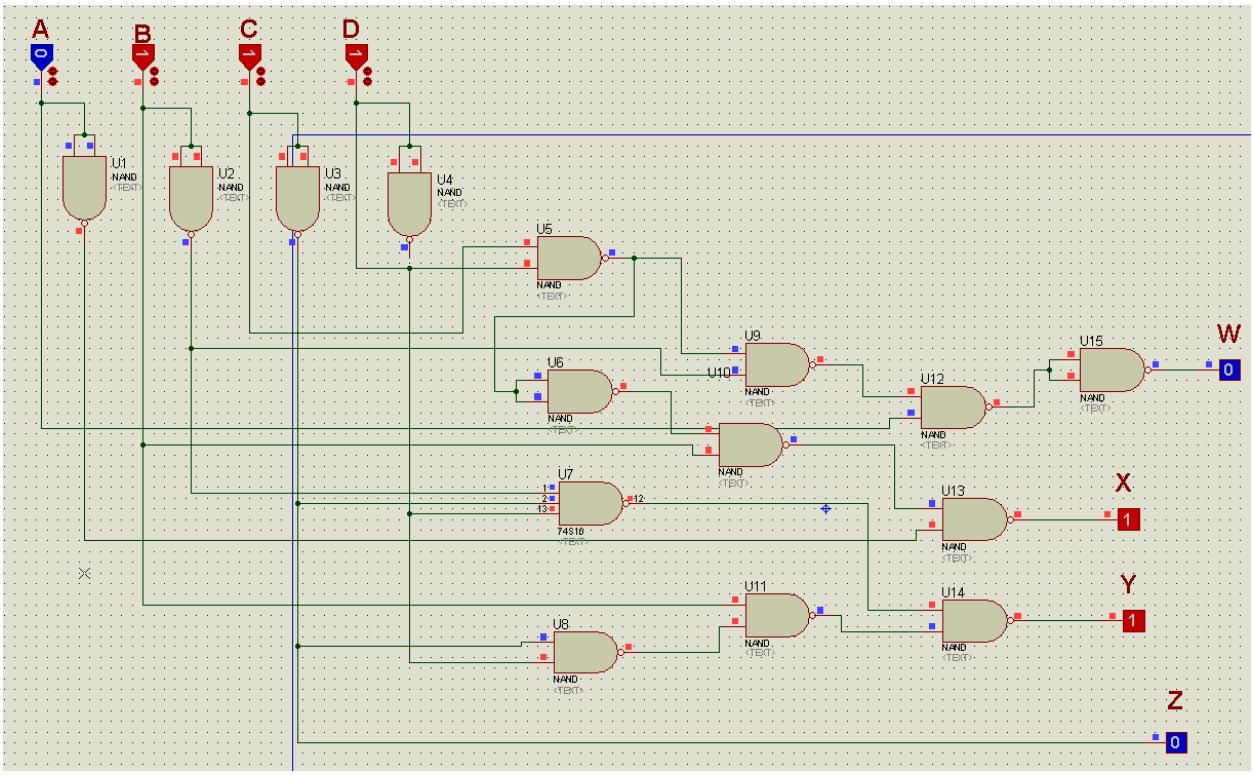

A
B
C
D
1
1
0
0
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
74S10
W
X
Y
Z
1
1
1
1


A
B
C
D
0
0
1
1
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
74S10
W
X
Y
Z
0
0
0
0

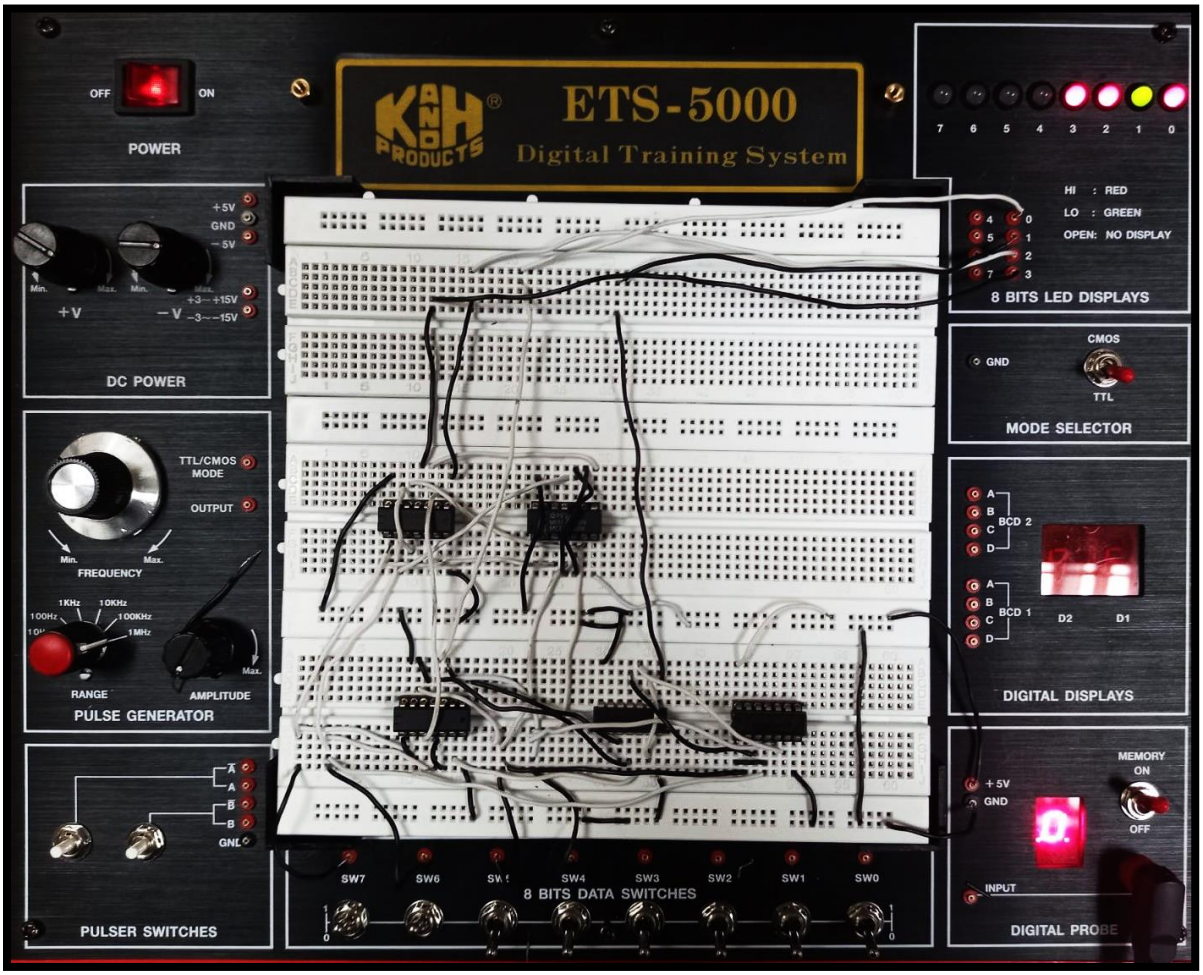

A
B
C
D
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
74S10
W
X
Y
Z
1
0


A
B
C
D
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
74S10
W
X
Y
Z
0
1


A
B
C
D
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
74S10
W
1
X
1
Y
0
Z
0


A
B
C
D
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
74S10
W
0
X
1
Y
1
Z
0

HARDWARE:



Lab Task2: (2)

Design and simulate the circuit k-map equations you obtained in Pre-lab task 1 in Verilog **dataflow** modeling. Give the code and testbench and waveform in the space provided below.
CODE:

```
 1 module xs3_to_gray(w, x, y, z, a, b, c, d);
 2     input a, b, c, d;
 3     output w, x, y, z;
 4
 5     assign w = a & (b | (c & d));
 6     assign x = a | (b & (c & d));
 7     assign y = (~b & ~c & ~d) | (b & (c | d));
 8     assign z = ~c;
 9
10 endmodule
11 module X3TOGRAY;
12
13     reg A, B, C, D;
14     wire W, X, Y, Z;
15
16     xs3_to_gray t1(W, X, Y, Z, A, B, C, D);
17
18     initial begin
19         #100 A = 0; B = 0; C = 1; D = 1;
20         #100 A = 0; B = 1; C = 0; D = 0;
21         #100 A = 0; B = 1; C = 0; D = 1;
22         #100 A = 0; B = 1; C = 1; D = 0;
23         #100 A = 0; B = 1; C = 1; D = 1;
24         #100 A = 1; B = 0; C = 0; D = 0;
25         #100 A = 1; B = 0; C = 0; D = 1;
26         #100 A = 1; B = 0; C = 1; D = 0;
27         #100 A = 1; B = 0; C = 1; D = 1;
28         #100 A = 1; B = 1; C = 0; D = 0;
29     end
30
31 endmodule
32
```