

EXPERIMENT #4

DESIGN OF CODE CONVERTERS

OBJECTIVE:

1. Design and build gray code to binary converter.
2. Design and build BCD-to-7 segment converter.

APPARATUS:

- Seven segment display.
- SN 7400 quad 2-input NAND gates (1)
- SN 7410 triple 3-input NAND gates (4)
- SN 7420 dual 4-input NAND gates (4)
- SN 7404 HEX inverter (1)
- SN 7446 BCD-to-seven segment decoder.

THEORY:

The conversion from one code to another is common in digital systems. Sometimes the output of a system is used as the input to the other system. A conversion circuit is necessary between 2 systems if each system uses different codes for the same information.

In this experiment we will design and construct 3-combinational circuit converters:

See section 4-5 in your book for further information.

Procedure:1. *Gray code to Binary converter:*

Gray code is one of the codes used in digital systems. It has the advantage over binary numbers that only one bit in the code word changes when going from one number to the next.

The following table illustrates the corresponding code representations of Decimal numbers in Binary Number System (Base 2 number system) and Gray Code Convention (Non-weighted)

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Design a combinational circuit with 4 inputs and 4 outputs that converts a four-bit gray code number into an equivalent four-bit Binary number.

Let, the 4 Bit Gray Code number be ABCD and Binary number be WXYZ.

Decimal	Gray Code (ABCD)				Binary Code (WXYZ)			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	1	0	0	1	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	1	1	0	0	1	0	0	0
9	1	1	0	1	1	0	0	1
10	1	1	1	1	1	0	1	0
11	1	1	1	0	1	0	1	1
12	1	0	1	0	1	1	0	0
13	1	0	1	1	1	1	0	1
14	1	0	0	1	1	1	1	0
15	1	0	0	0	1	1	1	1

Table 1: Gray code with corresponding Binary and Decimal Equivalent values

- Use Karnaugh map technique for simplification.
- Use LogicWorks OR OrCAD for pre-lab demonstrations.
- Select the library "7400dev.clf" in the Parts Palette.
- Then select the XOR chip 74-86. This would give you a set of 4 XOR's as shown in Fig. 1, just like the hardware chip 74-86. You could use as many as needed from these XOR gates in your design.
- Get back to ALL LIBRARIES and select switches for the inputs and Binary Probes as indicators of the outputs.
- Verify your design in the pre-Lab. During the Lab construct the circuit and verify its operation

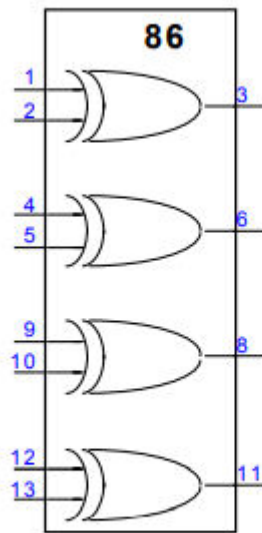


Figure. 1 XOR chip 74-86

Following is the K Map simplification for Gray code to Binary Conversion.
Each digits of the Binary number form a separate Karnaugh Map where each digit is a direct function of all the gray code digits.

For W,

AB \ CD					
		00	01	11	10
AB	00				
	01				
	11	1	1	1	1
	10	1	1	1	1

$$W=A$$

For X,

AB \ CD		CD			
		00	01	11	10
AB	00				
	01	1 ₄	1 ₅	1 ₇	1 ₆
	11				
	10	1 ₈	1 ₉	1 ₁₁	1 ₁₀

$$X = A'B + AB'$$

$$= A \oplus B$$

For Y,

AB \ CD		CD			
		00	01	11	10
00	00			1	1
		0	1	3	2
01	01	1	1		
		4	5	7	6
11	11			1	1
		12	13	15	14
10	10	1	1		
		8	9	11	10

$$\begin{aligned}
 Y &= A'BC' + AB'C' + A'B'C + ABC \\
 &= A(B'C' + BC) + A'(BC' + B'C) \\
 &= A(BC' + B'C)' + A'(BC' + B'C) \\
 &= A(B \oplus C)' + A'(B \oplus C) \\
 &= A \oplus B \oplus C
 \end{aligned}$$

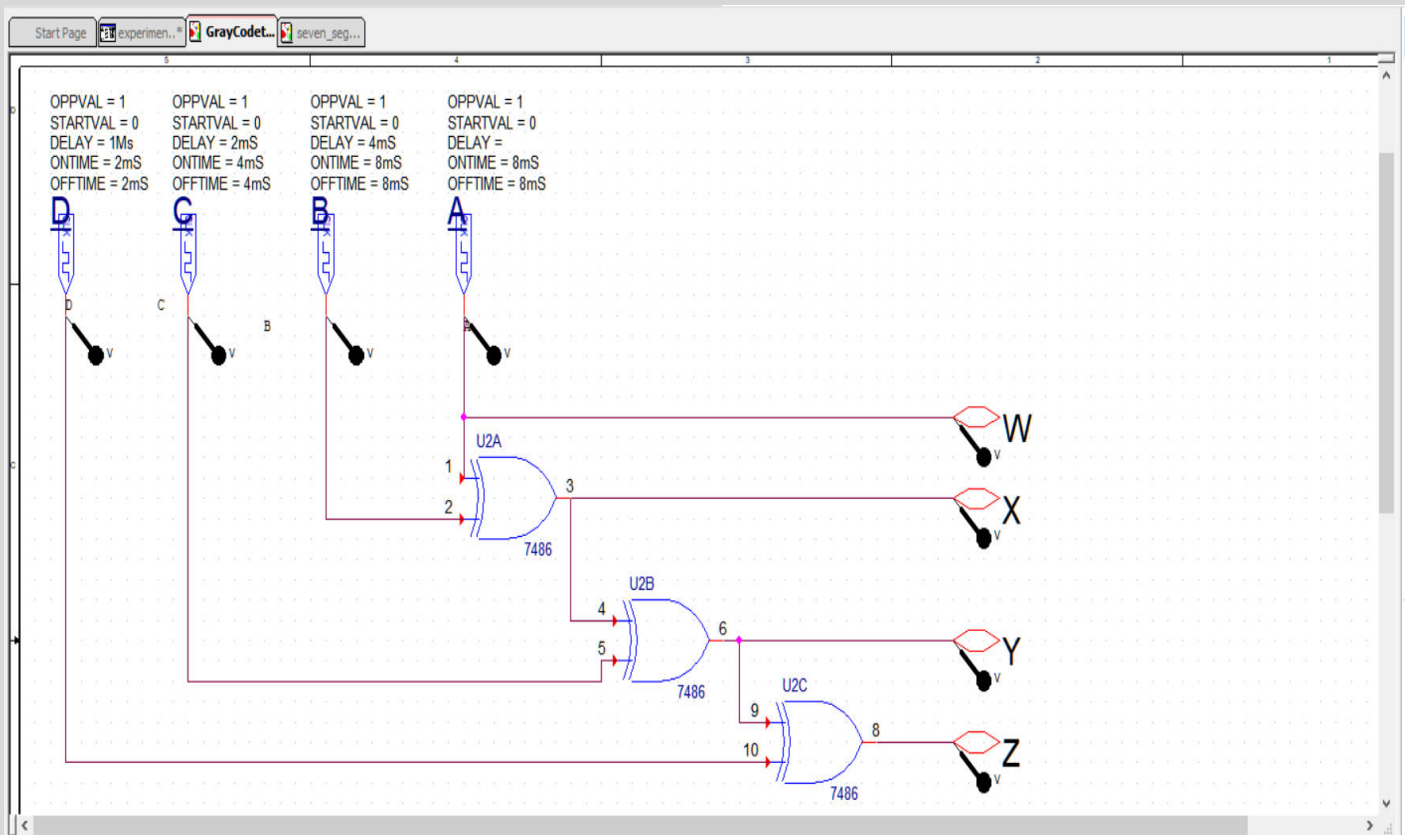
For Z,

AB \ CD		CD			
		00	01	11	10
AB	00		1		1
	01	1		1	
	11		1		1
	10	1		1	

$$Z = A'B'C'D + A'B'CD' + A'BC'D' + AB'C'D' + A'BCD + AB'CD + ABC'D + ABCD'$$

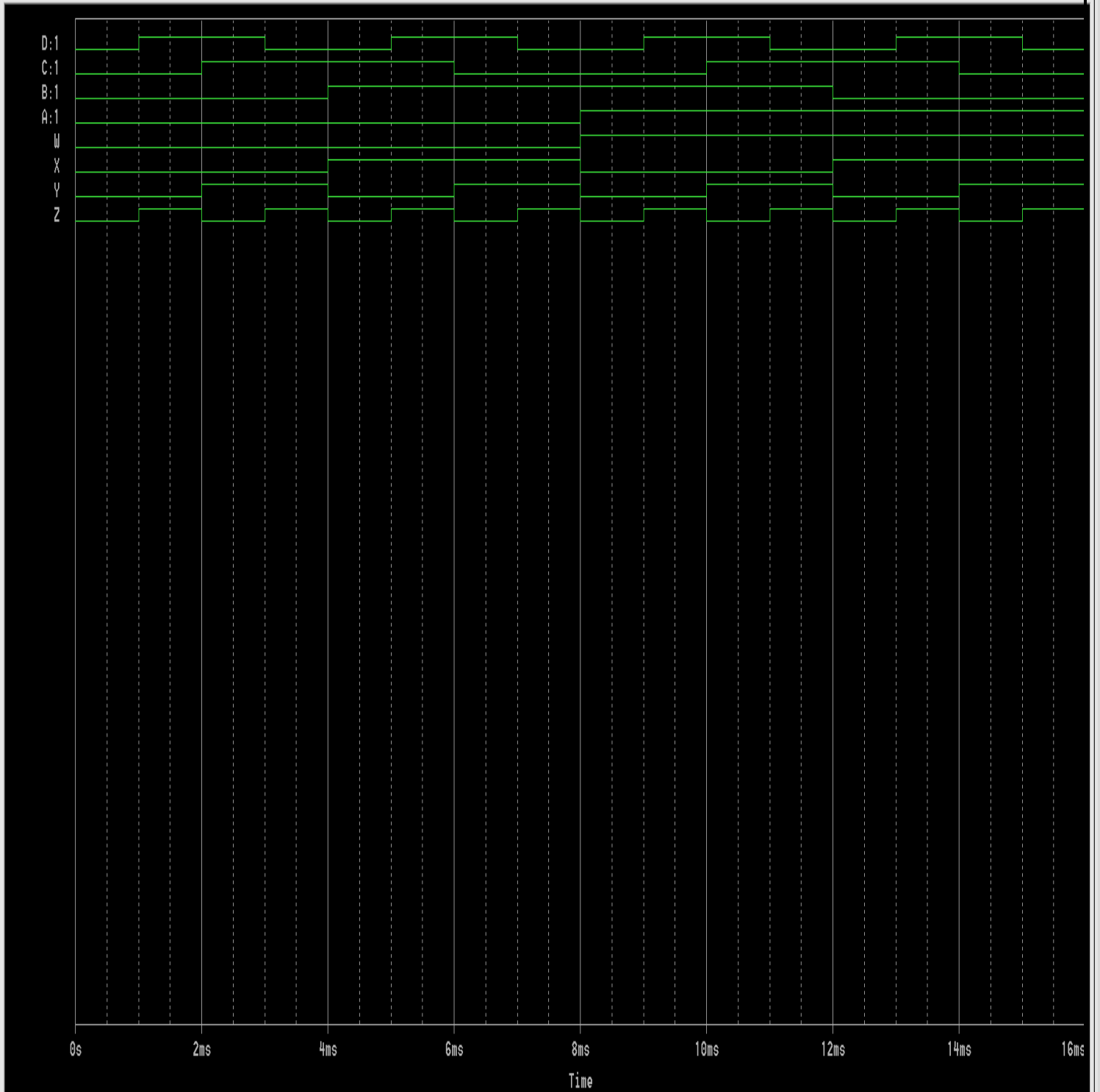
$$Z = A \oplus B \oplus C \oplus D$$

Construction and simulation of Logic diagram to Convert Gray code Input to Binary Number.:



20BDS0405

Inputs: A, B, C, D (Gray Code)
Outputs: W, X, Y, Z (Binary number)



2. BCD-to-seven Segment converter:

A light emitting Diode (LED) is a PN junction diode. When the diode is forward biased, a current flows through the junction and the light is emitted.

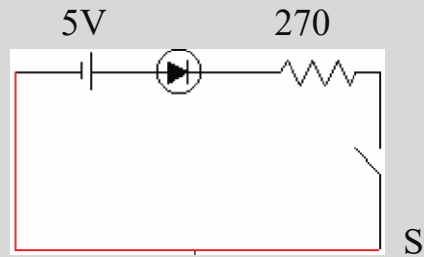


Figure.2

A seven segment LED display contains 7 LEDs. Each LED is called a segment and they are identified as (a, b, c, d, e, f, g) segments. Figure 3.

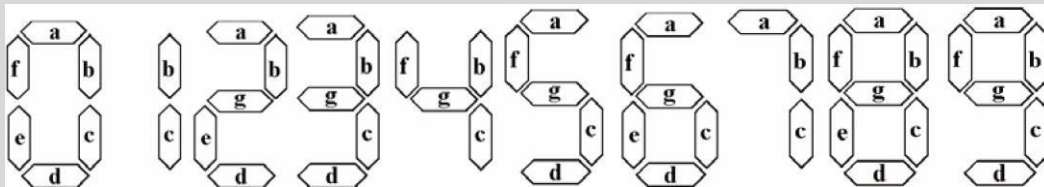


Figure 3. Digits represented by the 7 segments

The display has 7 inputs each connected to an LED segment. All anodes of LEDs are tied together and joined to 5 volts (this type is called common anode type). A limiting resistance network must be used at the inputs to protect the 7-segment from overloading.

BCD inputs are converted into 7 segment inputs (a, b, c, d, e, f, g) by using a decoder, as shown in Fig.4.

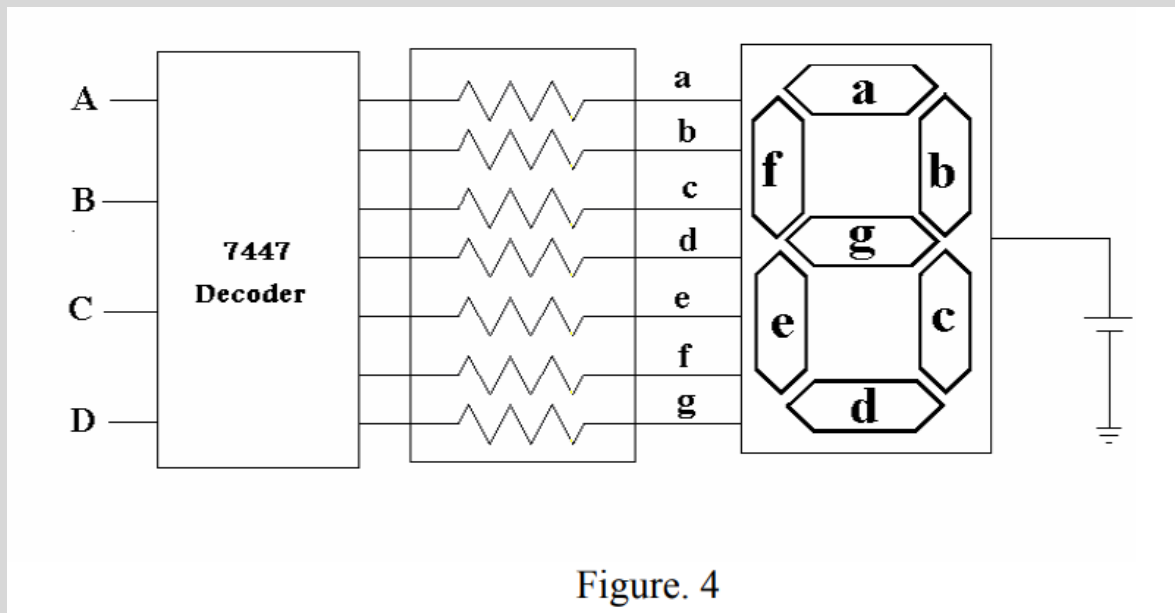


Figure. 4

A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n output lines. The input to the decoder is a BCD code and the outputs of the systems are the seven segments a, b, c, d, e, f, and g. For further information and pin connections, consult the specification sheet for decoder and 7-segment units.

First design a combinational circuit which would simulate the decoder function for only the segment "a", of the display. This can be done in the following steps:

a) Write down the truth table with 4 inputs and 7 outputs (Table 2)

Dec	A	B	C	D		a	b	c	d	e	f	g
0	0	0	0	0		0	0	0	0	0	0	1
1	0	0	0	1		1	0	0	1	1	1	1
2	0	0	1	0		0	0	1	0	0	1	0
3	0	0	1	1		0	0	0	0	1	1	0
4	0	1	0	0		1	0	0	1	1	0	0
5	0	1	0	1		0	1	0	0	1	0	0
6	0	1	1	0		0	1	0	0	0	0	0
7	0	1	1	1		0	0	0	1	1	1	1
8	1	0	0	0		0	0	0	0	0	0	0
9	1	0	0	1		0	0	0	0	1	0	0

- b) For only the output "a", obtain a minimum logic function. Realize this function using NAND gates and inverters only. For example if decimal 9 is to be displayed a, b, c, d, f, g must be 0 and the others must be 1 (For common anode type display units), if decimal 5 is to be displayed then a, f, g, c, d must be 0 and the others must be 1.

K Map for a

CD		00	01	11	10
AB	00	0 1	1 1	3	2
	01	4 1	5	7	6
	11	12 X	13 X	15 X	14 X
	10	8	9	11 X	10 X

$A'.B'.C'.D + B.C'.D'$

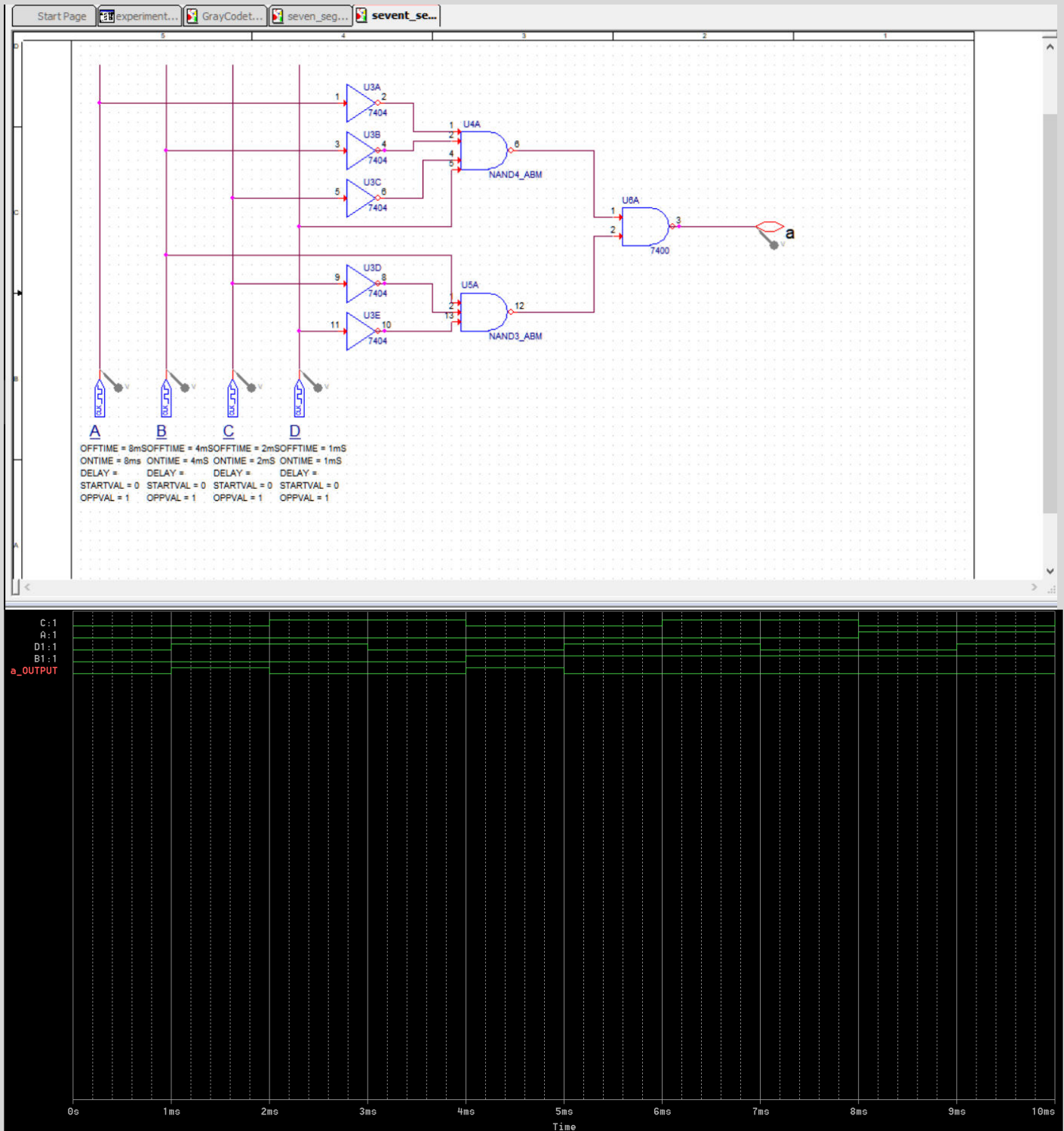
By Using De Morgan's Law,

$$a = A' B' C' D + B C' D'$$

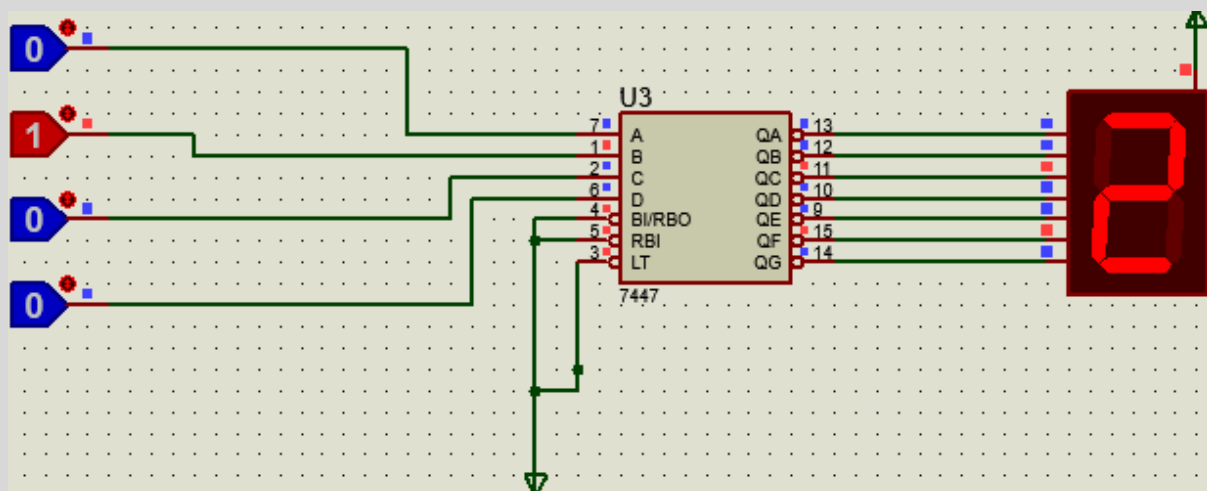
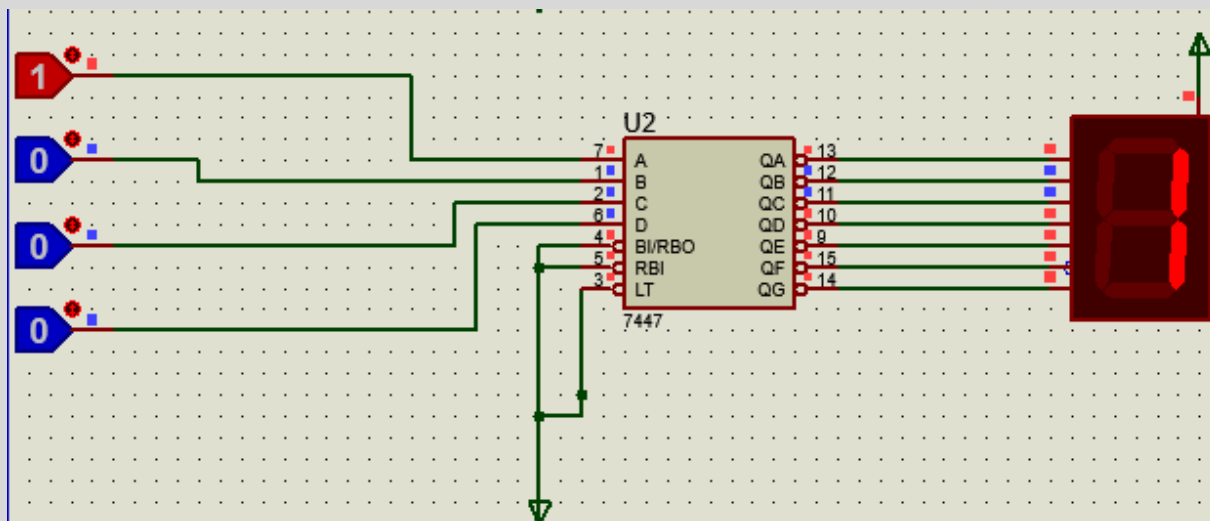
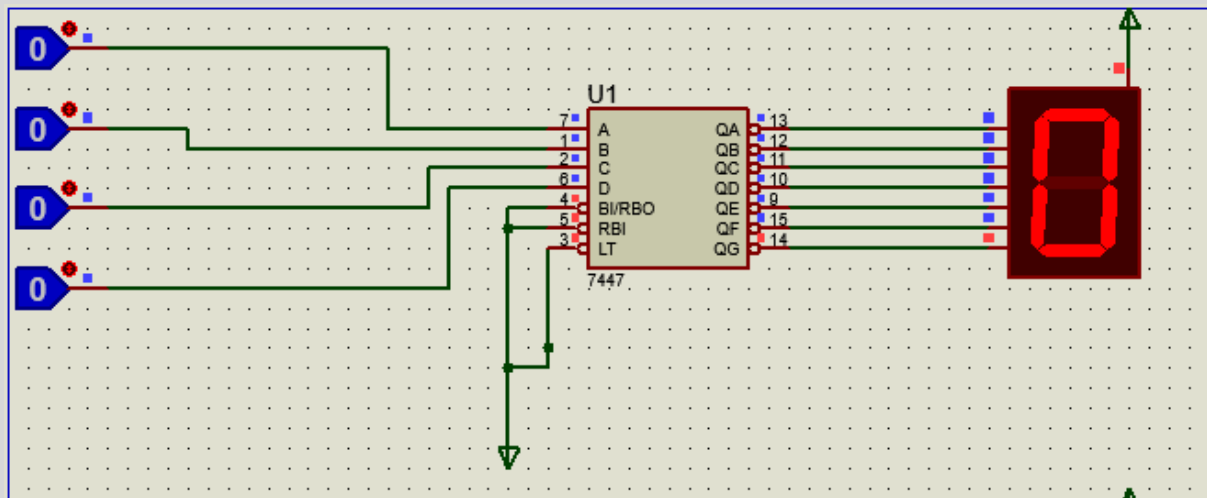
$$= ((A' B' C' D)' . (B C' D')')$$

20BDS0405

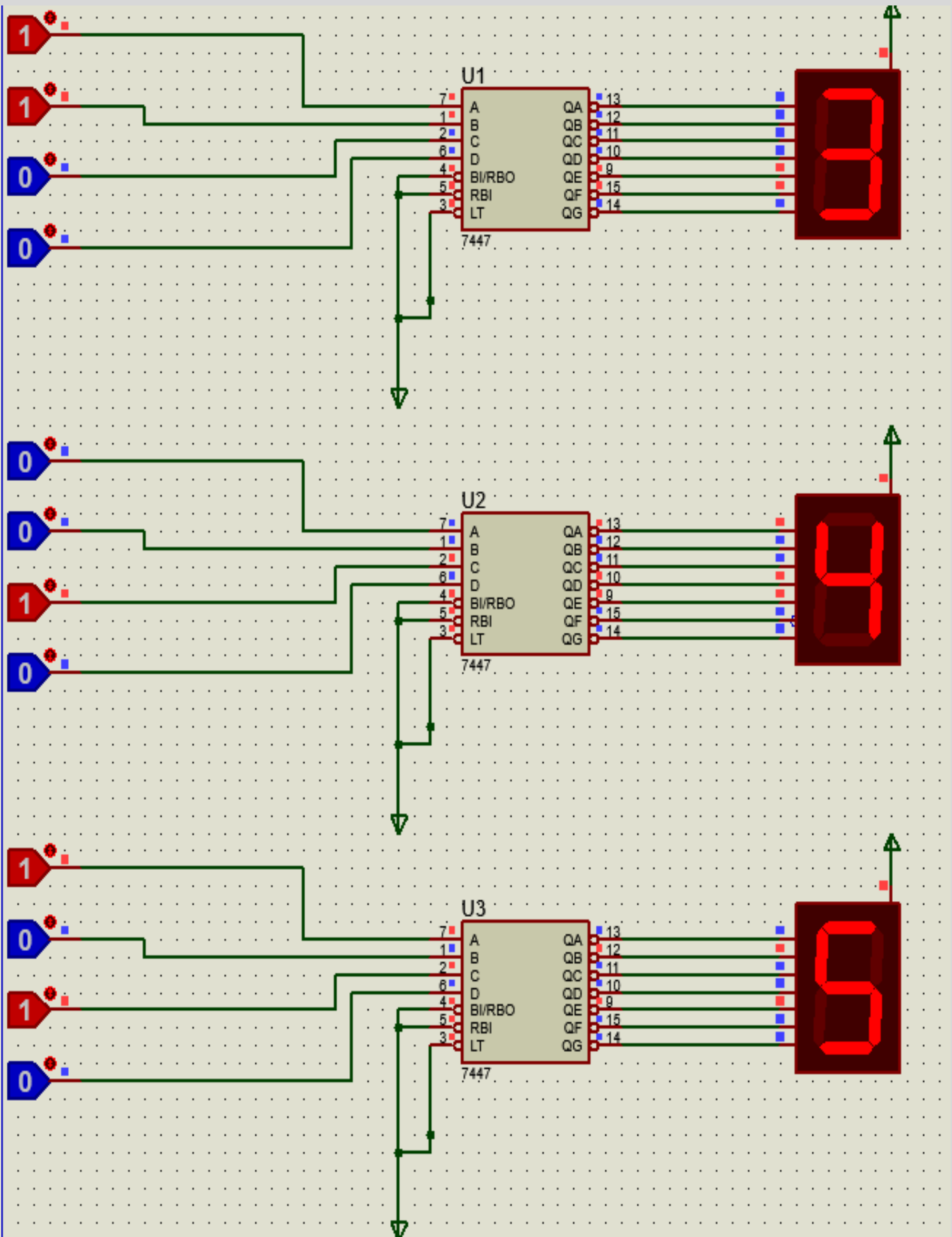
- c) Connect the output "a" of your circuit to appropriate input of 7-segment display unit. By applying BCD codes verify the displayed decimal digits for that segment for "a" of the display.

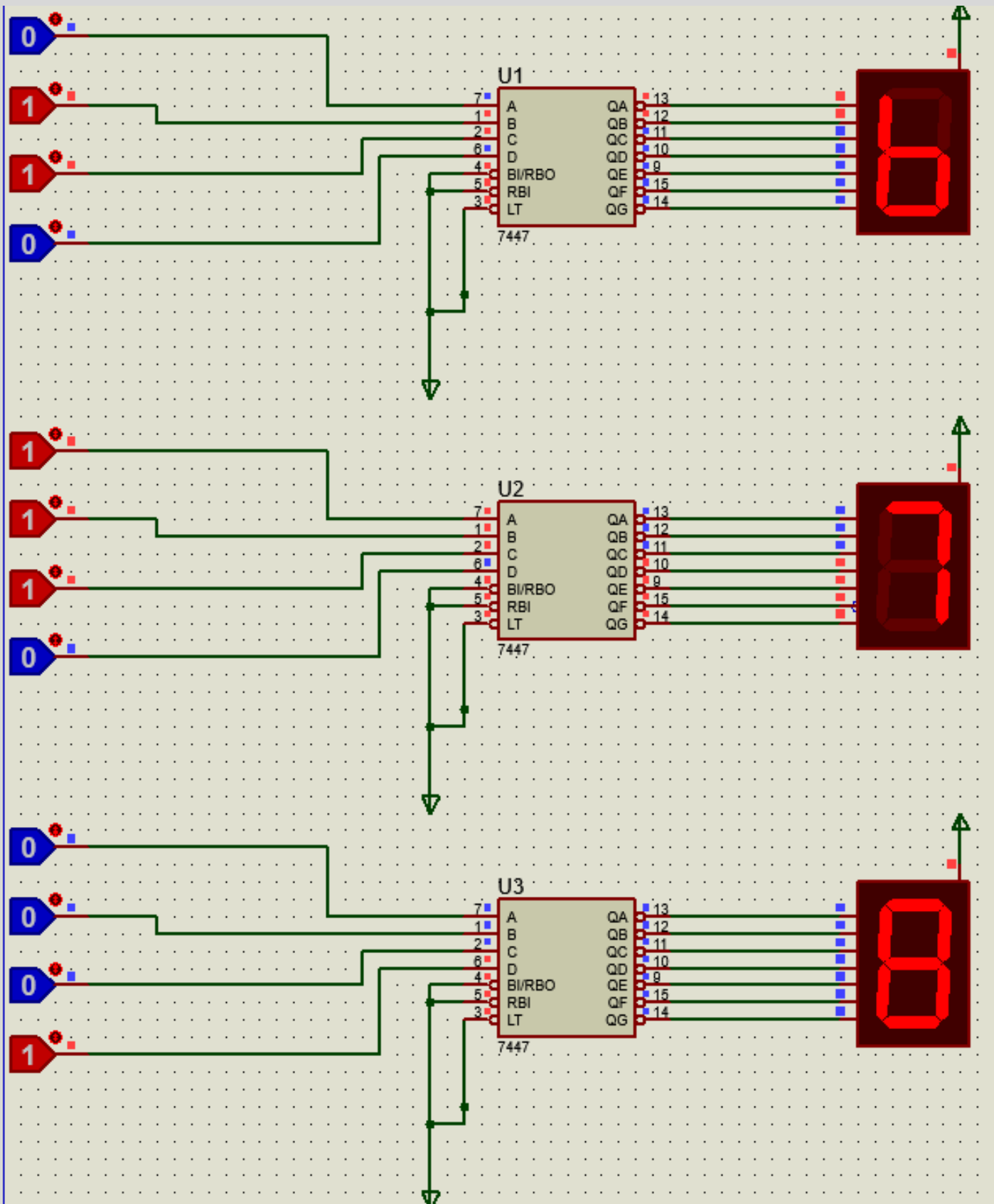


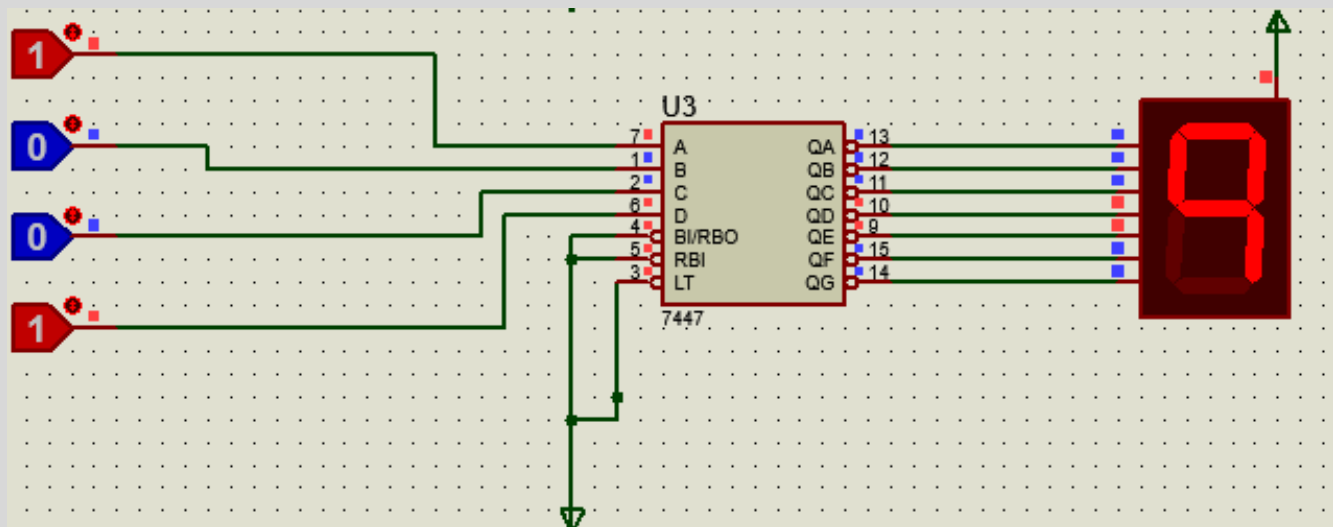
- d) Replace your circuit by a decoder IC 7447 for all of the seven segments. Observe the display and record the segments that will light up for invalid inputs sequence.



20BDS0405







NOTE:

As observed from the simulation, the actual behavior of the 7447 IC circuit is slightly different from the expected ones.

In case of 6, which shows  instead of  and in case of 9, which

shows  instead of .

However, still they are sufficient to distinguish from the remaining numbers

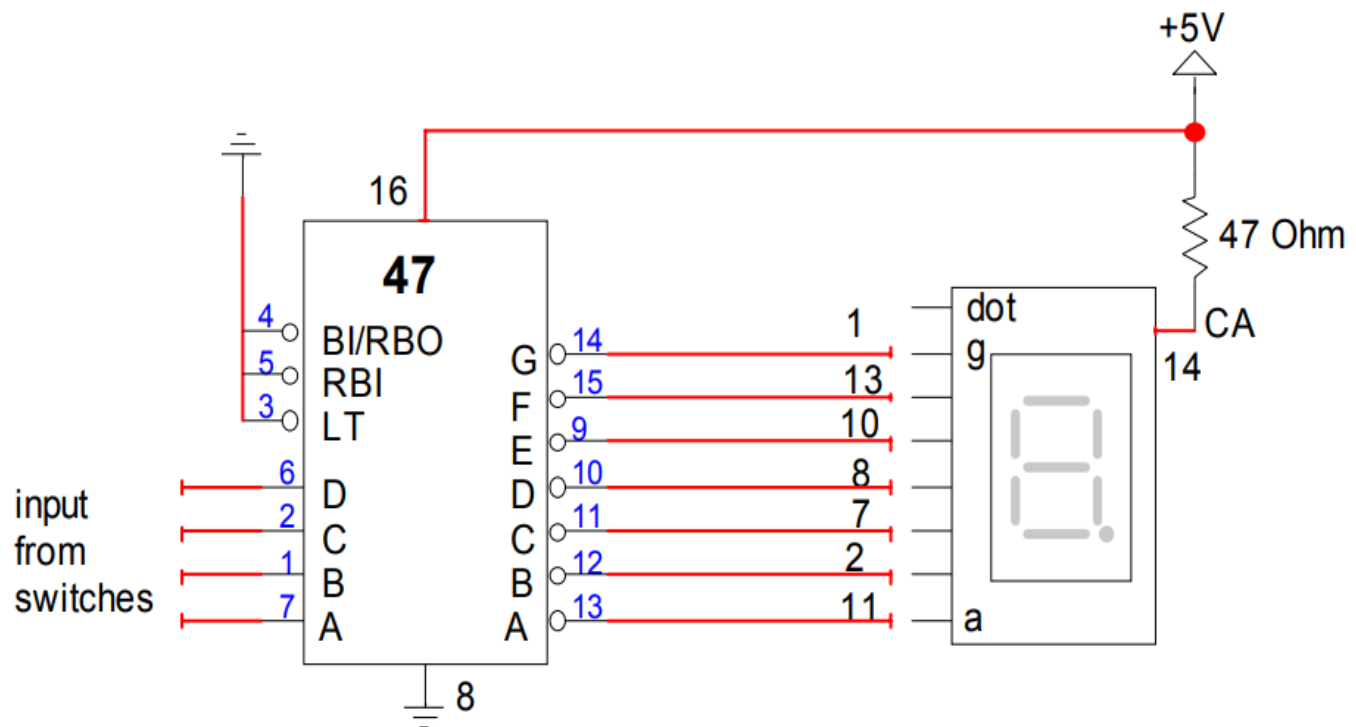
- e) Comment on the design if you don't want to see any digit for invalid input sequence.

While implementing the 7 segment decoder circuit, the value above 9 are considered as don't care and hence can have really random output that we don't care. However, if we want to get rid of that output, we can have the following ways:

- I. Limit the simulation time to only 10 units of time(0-9), instead of 16 units of time(0-16)
- II. Add a combinational block before the decoder circuit such that it sets all the outputs from the clock to 0.

Table 2

Dec.	BCD				Outputs						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	0	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X



BCD-to-Seven Segment Decoder and 7-segment display

Note: In an actual 7-segment display the dot is on the left