MAKERERE  UNIVERSITY

COLLEGE OF ENGINEERING, DESIGN, ART AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
SCHOOL OF ENGINEERING
BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING
YEAR THREE
ELE3103: APPLIED DIGITAL ELECTRONICS
ASSIGNMENT 1
GROUP

August 2023

NAME	COURSE	REG. NO.
Babirye Immaculate	BELE	21/U/1048
Akirapa Jacky	BELE	21/U/0637
Nabwire Jovia Namulinda	BELE	21/U/1352
Mbalire Shawal	BELE	21/U/0851
Bwire Kennedy	BELE	21/U/0456

1 Question One

Consider a digital control circuit that controls temperature and pressure in a given system. If the temperature and/or pressure is out of the desired range, the system energizes an indicator light S at the output. A temperature value that is within the desired range is indicated by a HIGH at logic signal T. Pressure is measured in two different points of the system. A pressure value out of range is indicated by two logic signals U and V both of which go HIGH at both points.

1.1 Design a logic circuit that energizes the indicator light.

To design a logic circuit that activates the indicator light (S), we first establish the truth table to specify the desired behavior. The truth table is as follows:

Truth Table

T	**U**	**V**	**S**
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

The Karnaugh Map (K-Map) for the output S is as follows:

K-Map for Output S

S	$\overline{T}$	T
$\overline{U}\overline{V}$	1	0
$\overline{U}V$	1	0
UV	1	1
$U\overline{V}$	1	0

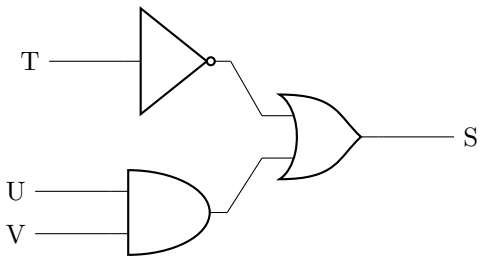
$$S = \overline{T} + UV$$

From the K-Map and the given conditions:

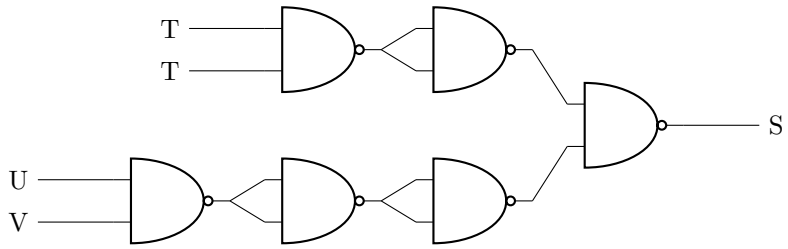
- T is HIGH when the temperature is within the desired range.
- If both U and V are HIGH, the pressure is out of range.
- To energize the light (S), we need T to be LOW, and both U and/or V should be HIGH at the same time.

This leads to the following logical expression for S:

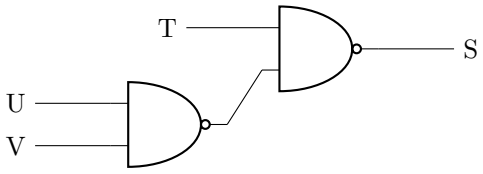
$$S = not\ T\ or\ (U\ and\ V)$$



For implementation with a 74LS00N chip, we must change the AND gate to NAND gates. The circuit diagram is as follows:



This circuit, using NAND gates, effectively activates the indicator light S according to the specified conditions for temperature and pressure. It can be implemented with a 74LS00N chip.



$$S = \overline{T\overline{U}\overline{V}}$$

1.2 Implementing the Circuit with Multisim and 74LS00N Chip

To demonstrate the practical implementation of the logic circuit, we used Multisim and the 74LS00N quad 2-input NAND gate integrated circuit. The following figures (1 and 2) depict the circuit designed in Multisim. In these figures, we showcase two different scenarios by varying the input values of U, V, and T, with the corresponding output S.

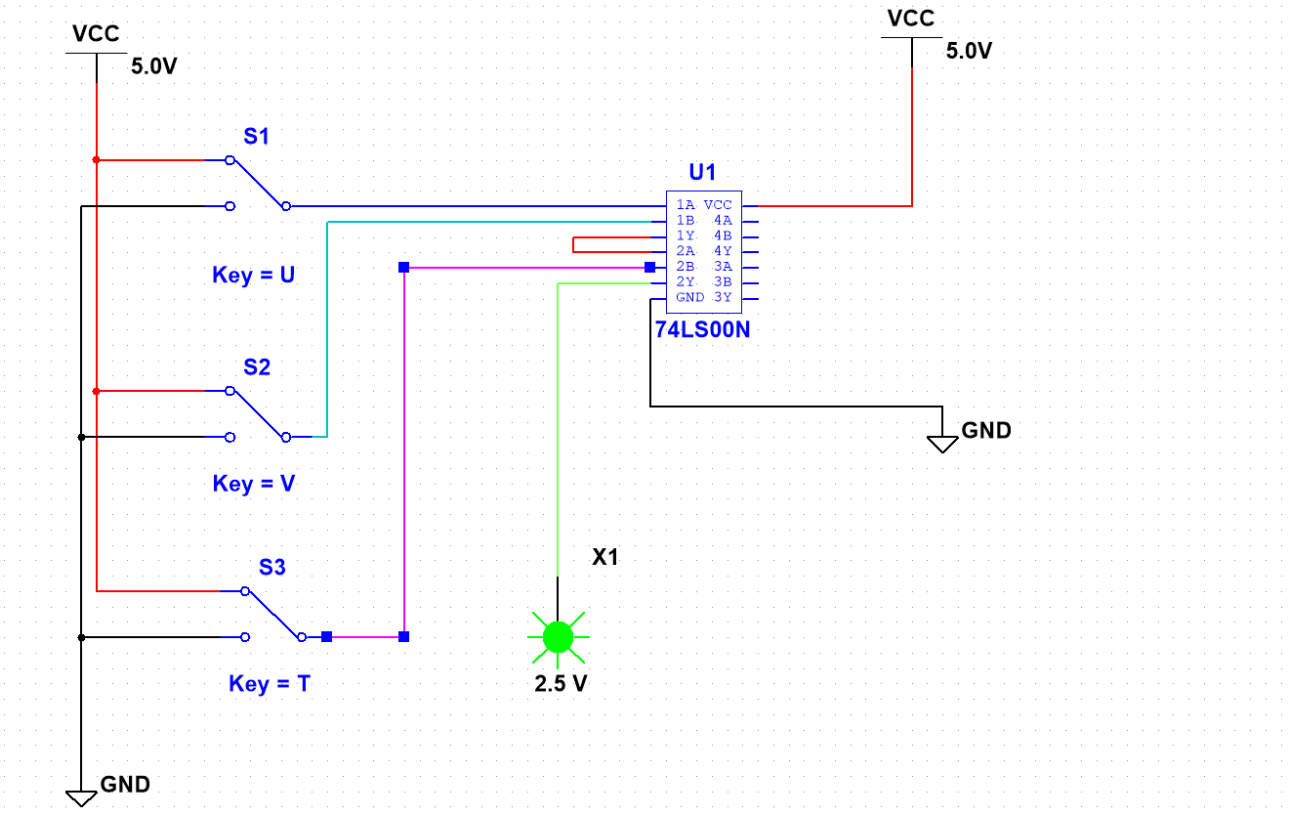


Figure 1: Multisim circuit for question 1a, Scenario 1:$V = 1, U = 1, T = 1$

In Figure 1, we present Scenario 1 where the inputs U, V, and T have specific values, resulting in an output S. $V = 1, U = 0, T = 1$ and $S = 1$. This is also an expected output since the temperature is within the desired range, and the pressure is out of range.

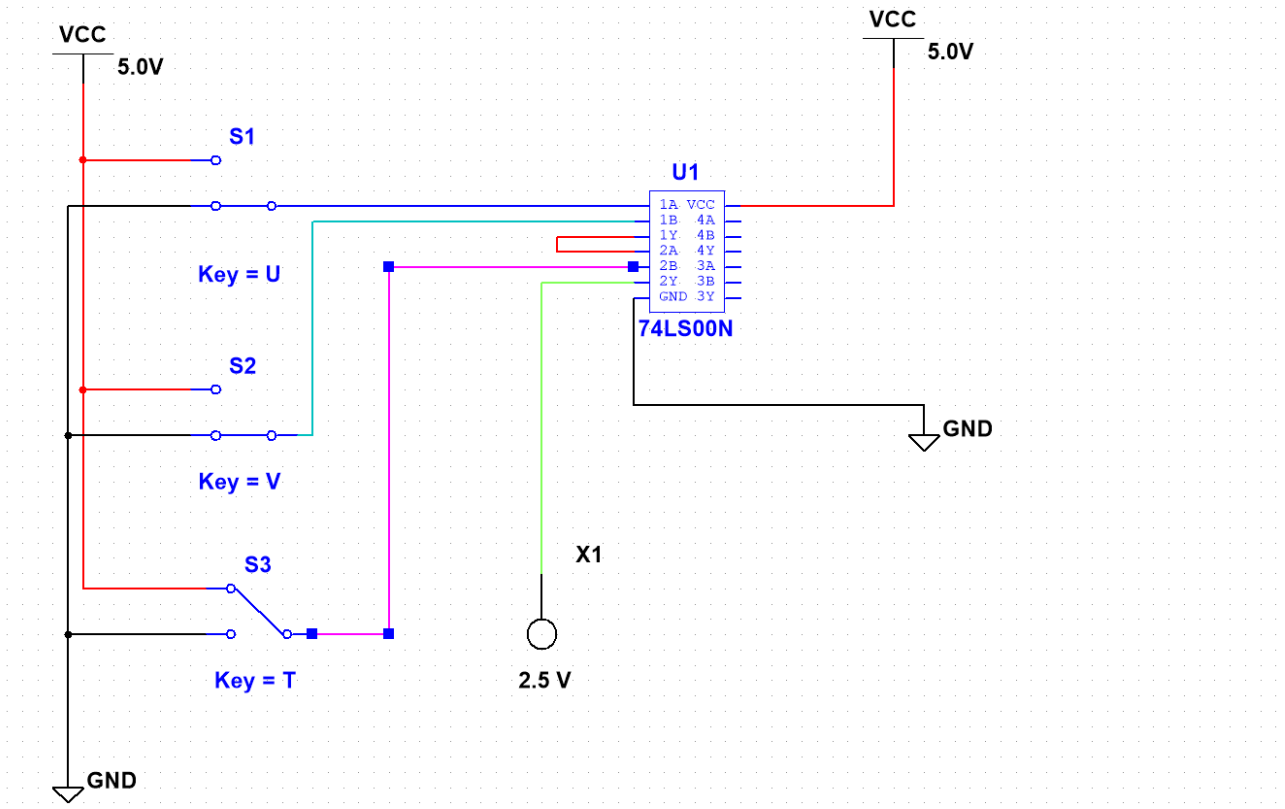


Figure 2: Multisim circuit for question 1a, Scenario 2: $V = 1, U = 0, T = 1$

In Figure 2, we illustrate Scenario 2 with a different set of inputs for U, V, and T, leading to a distinct output S. $V = 1, U = 1, T = 1$ and $S = 0$. This is an expected output since the temperature is within the desired range, and the pressure is out of range. These scenarios demonstrate the practical performance and functionality of the implemented circuit.

1.3 Use Quartus software to implement the circuit in (a). Show the code, RTL viewer schematic, and timing diagram of the circuit performance.

The Figures 4 and 3 show the circuit implemented in Quartus with U,V,T as high inputs and S as output.

```
1
2
3   SUBDESIGN program
4   (
5       U, V, T : INPUT;
6       S : OUTPUT;
7   )
8
9
10  VARIABLE
11      m : NODE;
12
13
14  BEGIN
15      m = U !& V;
16      S = m !& T;
17  END;
```

Figure 3: Quartus Code for question 1a

Figure 3 showcases the Quartus code used for implementing the logic circuit. The code defines the behavior and connections of the circuit using AHDL and NAND gates.

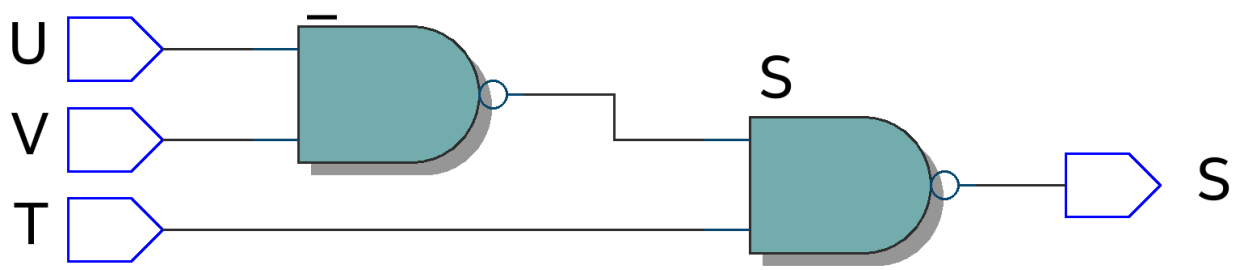


Figure 4: Quartus Circuit Schematic for question 1a

In Figure 4, we present the Quartus circuit schematic, which visually represents the implemented logic circuit. The inputs U, V, and T are set to HIGH, and the output S is displayed.

Furthermore, the RTL viewer schematic in Figure 10 illustrates the doubly implemented NAND gates. They are represented as a NOT gate in this viewer, highlighting the logical structure of the circuit.

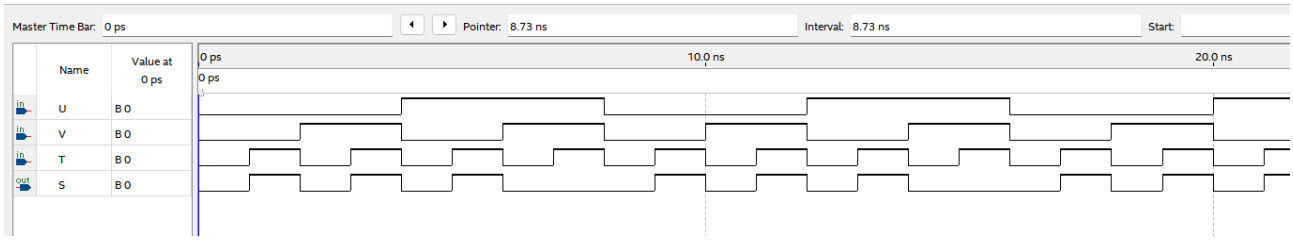


Figure 5: Quartus Timing Diagram (VWF) for question 1a

Figure 10 displays the Quartus Timing Diagram in VWF format, providing insights into the circuit's performance over time. It shows signal waveforms, illustrating how the circuit responds to different input conditions and transitions.

2 Question Two

Implement a MOD-13 synchronous counter. Ensure that the counter correctly counts through the full cycle. Use a HEX component to display the counting sequence. Add the XLA to view your output. Include only two screenshots in your report. **Note: I have provided a simple tutorial about using the XLA in a separate document.**

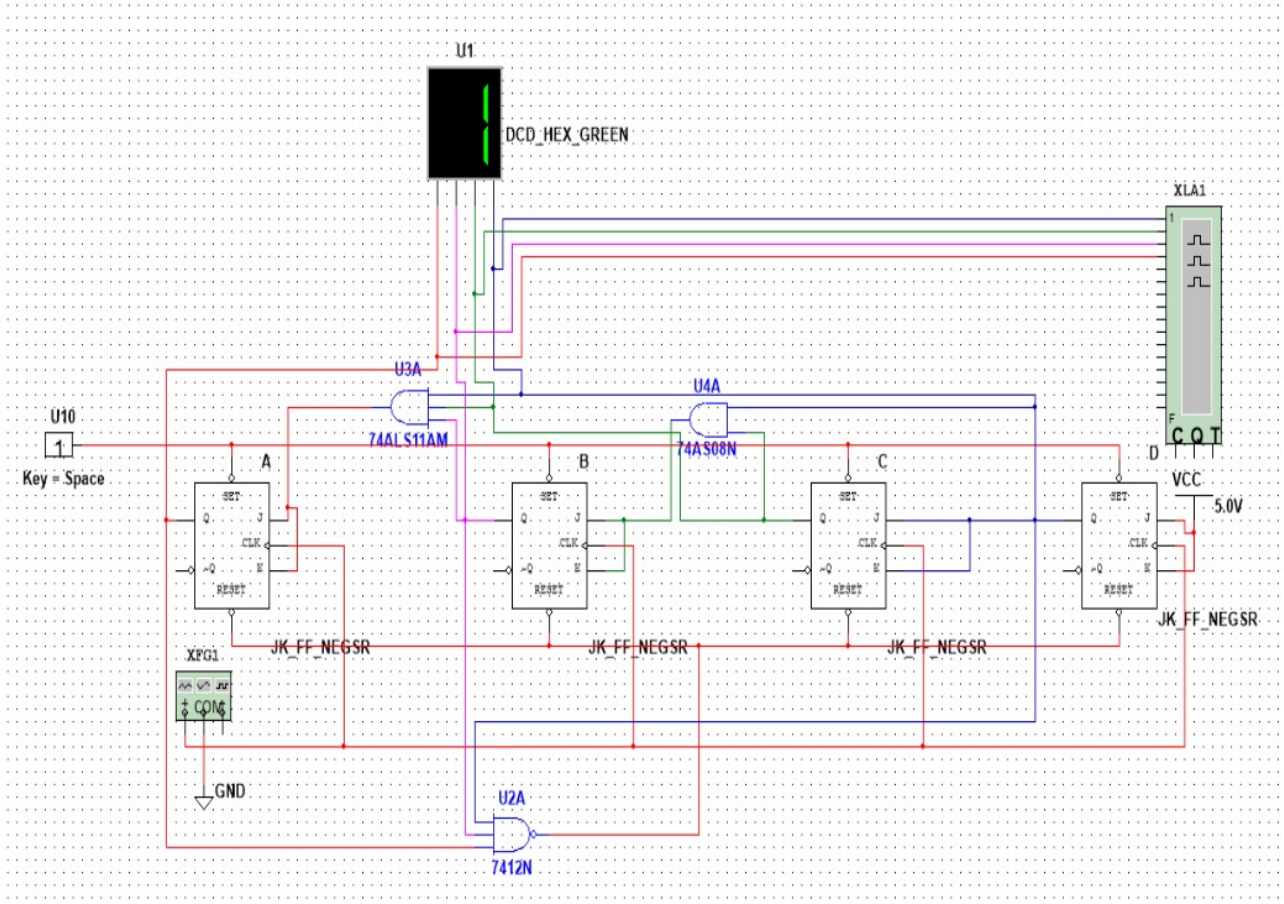


Figure 6: Scenario one

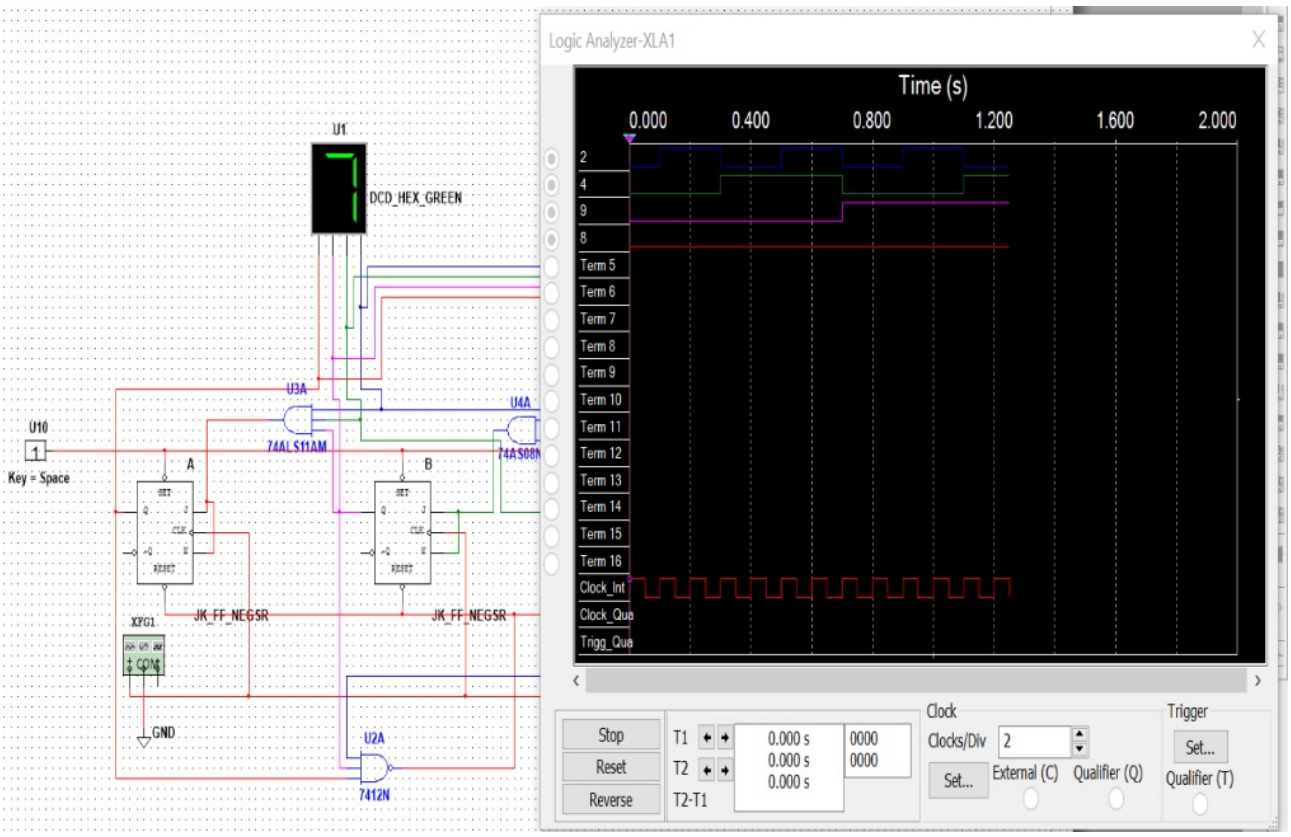


Figure 7: Scenario two

3 Question Three

Demonstrate the counting performance of a 74ALS161 IC counter. Use a numeric display (e.g., the SEVEN SEG COM A) to display its count states. Use Multisim.

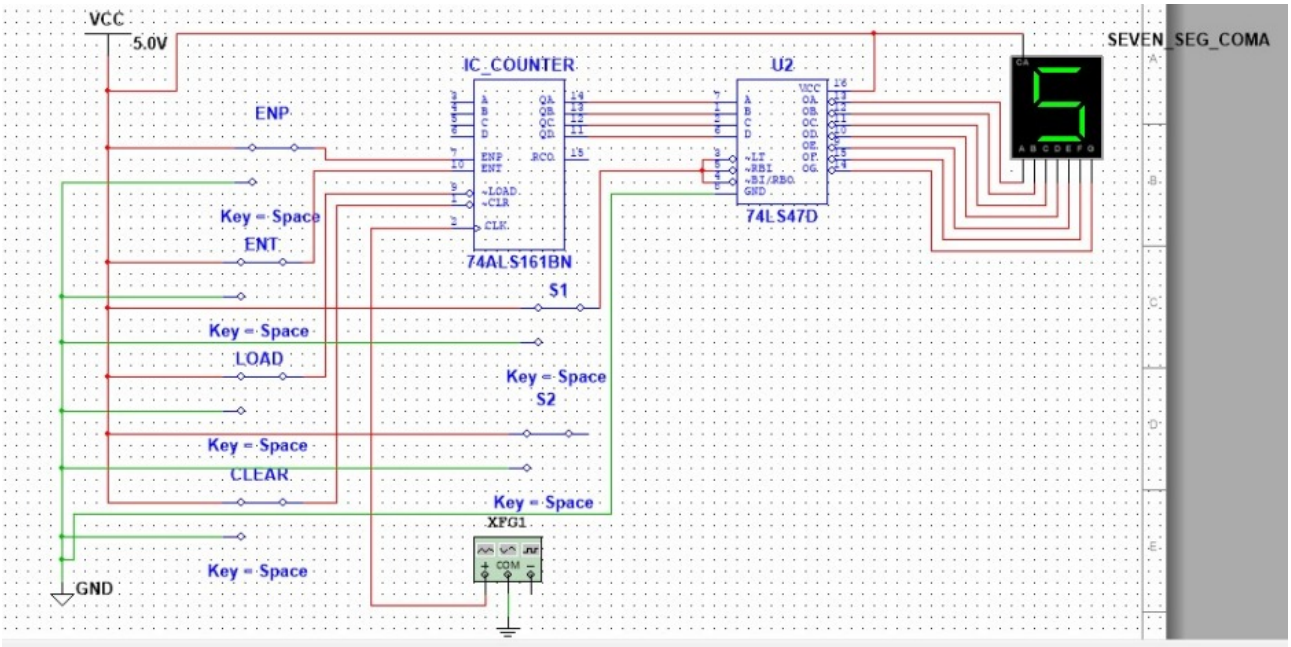


Figure 8: Scenario one

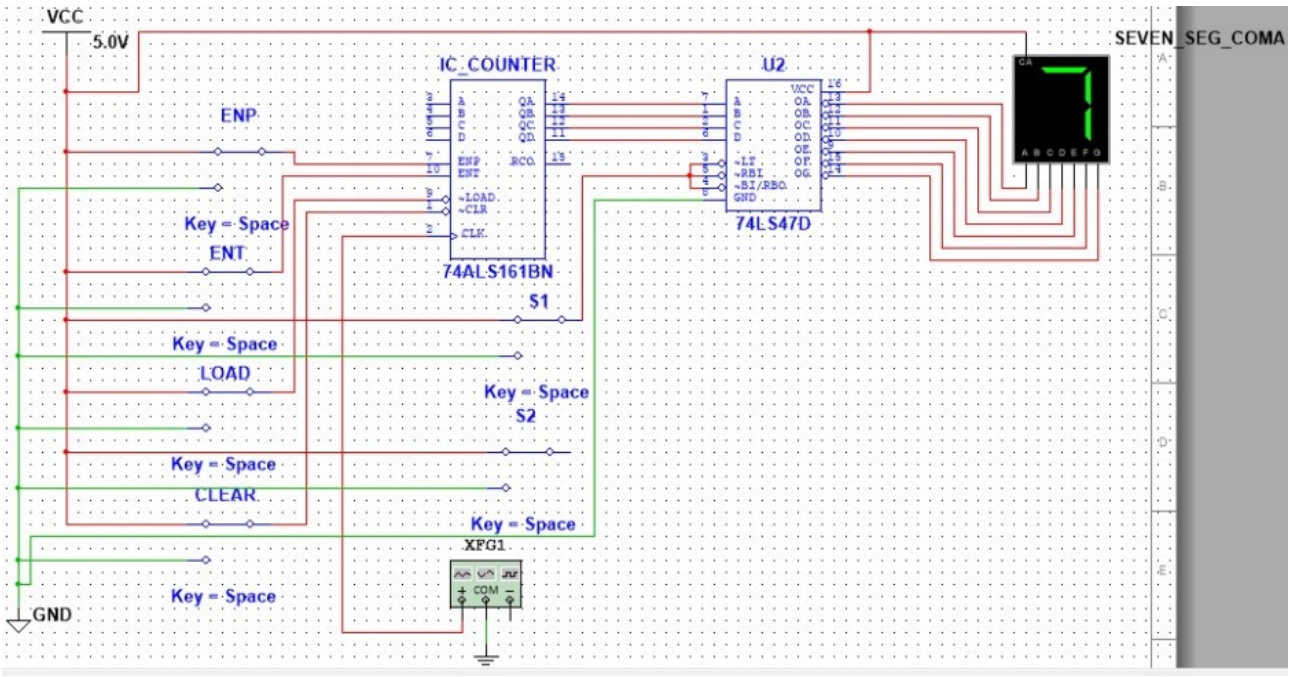


Figure 9: Scenario two

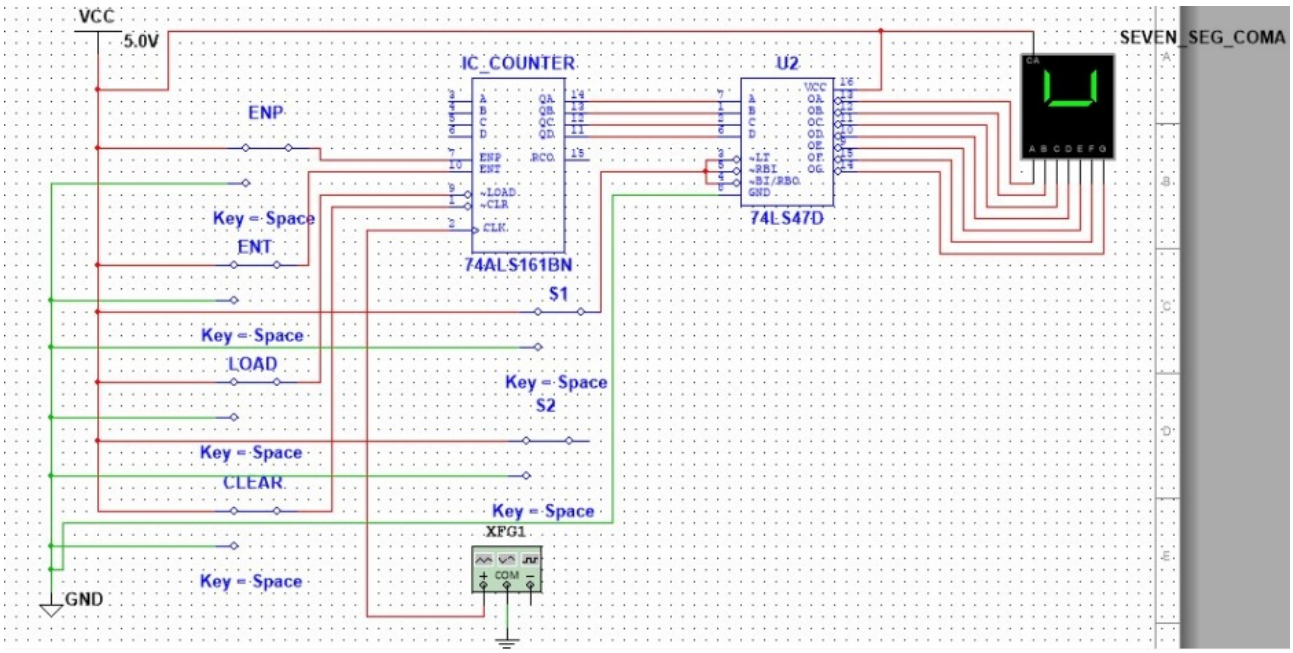


Figure 10: Scenario three

4 Question Four

The 7442 BCD-to-Decimal decoder (Fig 9.5) can be used as a 3-line-to-8-line decoder, by using the D input as an ENABLE input (see Problem 9-7). Explain briefly how this arrangement works as an enabled 1-of-8 decoder, and state how the level on D either enables or disables the outputs (show in a truth table). **Note: This number does not require implementation in Multisim.**

Solution The 7442 BCD-to-Decimal decoder can be used as a 3-line-to-8-line decoder when the D input serves as an ENABLE input. In this configuration, D will be held active LOW since it is considered as an ENABLE input. When D is set to HIGH (1), it enables the decoder, allowing it to function as a 3-line-to-8-line decoder.

Since D is considered as an ENABLE input, C, B, and A will be the three inputs, and the outputs will range from 0 to 7 while outputs 8 and 9 will be inactivated since D will be in its INACTIVE state. In this state, the inputs A, B, and C (the 3-line input) are decoded, and one of the eight outputs (Y0 to Y7) is selected based on the binary combination at A, B, and C.

Conversely, when D is LOW (0), it disables the decoder. In this condition, all of the outputs (Y0 to Y7) are rendered inactive. This effectively disconnects the A, B, and C inputs from the decoder, resulting in all outputs remaining at an inactive state.

Truth Table:

Here is a truth table illustrating how the level on D either enables or disables the outputs of the 7442 decoder:

D	C	B	A	Output
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	None
1	0	0	1	None
1	0	1	0	None
1	0	1	1	None
1	1	0	0	None
1	1	0	1	None
1	1	1	0	None
1	1	1	1	None