

Title: Studying different digital logic gates and designing of basic logic gates using Universal gates

Abstract:

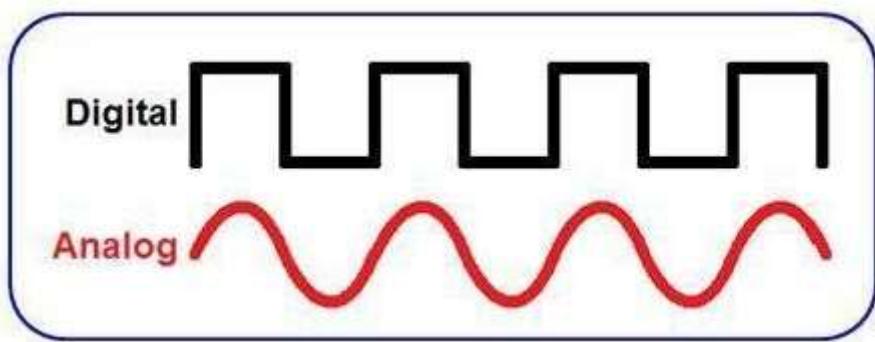
To learn the characteristics of several logic gates and to get familiar with the digital trainer board and digital ICs

Part I (Basic Logic IC's):

An integrated circuit (also referred to as an IC, a chip, or a microchip) is a set of electronic circuits on one small plate ("chip") of semiconductor material, normally silicon. This can be made much smaller than a discrete circuit made from independent components. Different integrated circuits are used to implement different logical operations in the trainer board which will be introduced in this experiment.

Theory and Methodology:

In analog signals, information is translated into electric pulses of varying amplitude but in case of digital, translation of information is in binary format (zero or one) where each bit is representative of two distinct amplitudes.



The main advantage of digital signals over analog signals is that the precise signal level of the digital signal is not vital. This means that digital signals are fairly immune to the imperfections of real electronic systems which tend to spoil analog signals. Codes are often used in the transmission of information. These codes can be used either as a means of keeping the information secret or as a means of breaking the information into pieces that are manageable by the technology used to transmit the code. It can convey information with greater noise immunity, because each information component (byte etc) is determined by the presence or absence of a data bit (0 or one). Analog signals vary continuously and their value is affected by all levels of noise. Digital signals can be processed by digital circuit components, which are cheap and easily produced in many components on a single chip. It uses typically less bandwidth with less electromagnetic interference. Moreover, Information storage can be easier in digital systems than in analog ones. The noise-immunity of digital systems permits data to be stored and retrieved without degradation.

There are two sorts of circuits which are known as integrated circuit and discrete circuit. The two main advantages of ICs over discrete circuits are cost and performance. Cost is low because the chips, with all their components, are printed as a unit by photolithography rather than being constructed one transistor at a time. Furthermore, much less material is used to construct a packaged IC die than to construct a discrete circuit. Performance is high because the components switch quickly and consume little power (compared to their discrete counterparts) as a result of the small size and close proximity of the components.

A logic gate is an elementary building block of a digital circuit. Most logic gates have two inputs and one output. At any given moment, every terminal is in one of the two binary conditions low (0V) and high (5V), represented by different voltage levels. The logic state of a terminal can, and generally does, change often, as the circuit processes data. In most logic gates, the low state is approximately zero volts (0 V), while the high state is approximately five volts positive (+5 V).

There are seven basic logic gates: AND, OR, NOT, NOR, NAND, XOR and XNOR. Different logic operations of different IC's will be introduced which perform the following characteristics:

Operation	Expression
AND	$Y=AB$
OR	$Y=A+B$
NOT	$Y= \bar{A}$
NOR	$Y= \bar{A} + \bar{B} = \bar{A} \bar{B}$
NAND	$Y= \bar{A} \bar{B} = \bar{A} + \bar{B}$
XOR	$Y=A \oplus B = \bar{A}B + A \bar{B}$
XNOR	$Y= AB+ \bar{A} \bar{B} = \bar{A}B + A \bar{B}$

AND operation:

The AND operation produces a high if and only if all the inputs are high. An AND gate can have two or more inputs and performs AND operation or logical multiplication.



Fig1.1: Symbol of AND gate

Truth Table:

Input, A	Input, B	Output, F
0	0	0
0	1	0
1	0	0
1	1	1

Pin configuration for IC-74HC08N :

For a quadrature 2input AND gate HC08 davice code is used. 74HC series devices are designed to work with a 5 V power supply, voltages from 2 V to 5 V are allowed and most circuits work well using 5 V.

OR operation:

The OR operation produces a high output when any of the inputs are high. It has two or more inputs and one output which performs OR operation or logical addition.

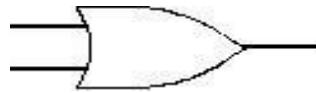


Fig 1.2: Symbol of OR gate

Truth Table:

Input, A	Input, B	Output, F
0	0	0
0	1	1
1	0	1
1	1	1

Pin configuration for IC-74HC32N:

HC32 is the device code. 74HC32 is a Quad 2-input OR gate (High Speed CMOS version) which has lower current consumption/wider Voltage range from 2 to 5V. It requires low input current of $1\mu\text{A}$ with high noise immunity characteristics of CMOS devices.

NOT operation:

The NOT operation changes one logic level to the opposite logic level. It is implemented by a logic circuit known as an inverter.



Fig1.3: Symbol of NOT gate

Truth Table:

Input, A	Output, F
0	1
1	0

Pin configuration for IC-74HC04N :

The 74HC04 is a hex inverter which consists of six inverters which perform logical invert action. The inputs include clamp diodes that enable the use of current limiting resistors to interface inputs to voltages in excess of Vcc. The Input level for 74HC04 is CMOS level .

NAND operation:

The NAND gate operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "AND" followed by negation. The output will be low if both inputs are high. Otherwise, the output is high .



Fig 1.4: Symbol of NAND
gate Truth Table:

Input, A	Input, B	Output, F
0	0	1
0	1	1
1	0	1
1	1	0

Pin configuration for IC-74HC00N :

HC00 is the device code. The device inputs are compatible with Standard CMOS outputs; with pullup resistors. The operating voltage range is 2.0 to 5.0 V and low input current is 1.0 μ A.

NOR operation:

The NOR gate is a combination OR gate followed by an inverter. Its output is high if both inputs are low. Otherwise, the output is low.



Fig 1.5: Symbol of NOR gate

Truth Table:

Input, A	Input, B	Output, F
0	0	1
0	1	0
1	0	0
1	1	0

Pin configuration for IC-74HC02N :

The 74HC02 is a high speed Si-gate CMOS device that provides a quadrature 2 –input NOR function. CMOS level is the input level for this sort of IC's. The operating Voltage Range is 2.0 to 5.0 V and low input current is 1.0 μ A.

XOR operation:

The XOR (exclusive OR) gate acts in the same way as the logical "either/or" .The output is high if either, but not both, of the inputs are high. The output is low if both inputs are low or if both inputs are high. Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same.

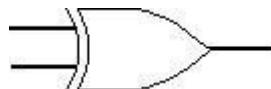


Fig 1.6: Symbol of XOR

gate Truth Table:

Input, A	Input, B	Output, F
0	0	0
0	1	1
1	0	1
1	1	0

Pin configuration for IC-74HC86N :

HC86 is the device code for a quad 2-input xor gate which utilizes advanced silicon gate CMOS technology . It maintains low power consumption and high noise immunity characteristic of standard CMOS integrated circuits. The 74HC logic family has a voltage range of 2V to 5V and the operating temperature is -40°C to 125°C with input current of 1 μ A.

XNOR operation:

The XNOR (*exclusive-NOR*) gate is a combination XOR gate followed by an inverter. Its output is high if the inputs are the same, and low if the inputs are different.

gate Truth Table:



Fig 1.7: Symbol of XNOR

Input, A	Input, B	Output, F
0	0	1
0	1	0
1	0	0
1	1	1

Using combinations of logic gates, complex operations can be performed. Arrays of logic gates are found in digital integrated circuits (ICs). As IC technology advances, the required physical volume for each individual logic gate decreases and digital devices of the same or smaller size become capable of performing ever more complicated operations at ever-increasing speeds.

Apparatus:

1. Digital trainer board.
2. Integrated Circuits (ICs).
3. Power supply.
4. Connecting wires.

Integrated Circuits (ICs):

7400 : 1 pcs

7402 : 1 pcs

7404 : 1 pcs

7408: 1 pcs

7432 : 1 pcs

7486 : 1 pcs

Precautions:

The IC contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages. For proper operation, Vin and Vout should be constrained to the range GND (Vin or Vout) to VCC.

IC configurations:

<pre> 01 1A V_{CC} 14 02 1B 4B 13 03 1Y 4A 12 04 05 2A 4Y 11 06 2B 3B 10 >V >A 09 </pre> <p>7400</p>	<pre> 01 1Y V_{CC} 14 02 1A 4Y 13 03 1B 4B 12 04 05 2Y 4A 11 06 2A 3Y 10 >V >A 09 </pre> <p>7402</p>	<pre> 01 1A V_{CC} 14 02 1Y 6A 13 03 2A 6Y 12 04 05 2Y 5A 11 06 3A 5Y 10 >V >A 09 </pre> <p>7404</p>
<pre> 01 1A V_{CC} 14 02 1B 4B 13 03 1Y 4A 12 04 05 2A 4Y 11 06 2B 3B 10 >V >A 09 </pre> <p>7408</p>	<pre> 01 1A V_{CC} 14 02 1B 4B 13 03 1Y 4A 12 04 05 2A 4Y 11 06 2B 3B 10 >V >A 09 </pre> <p>7432</p>	<pre> 01 1A V_{CC} 14 02 1B 4B 13 03 1Y 4A 12 04 05 2A 4Y 11 06 2B 3B 10 >V >A 09 </pre> <p>7486</p>

PART 1

AND Gate

Truth table:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

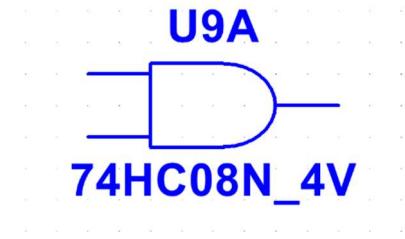
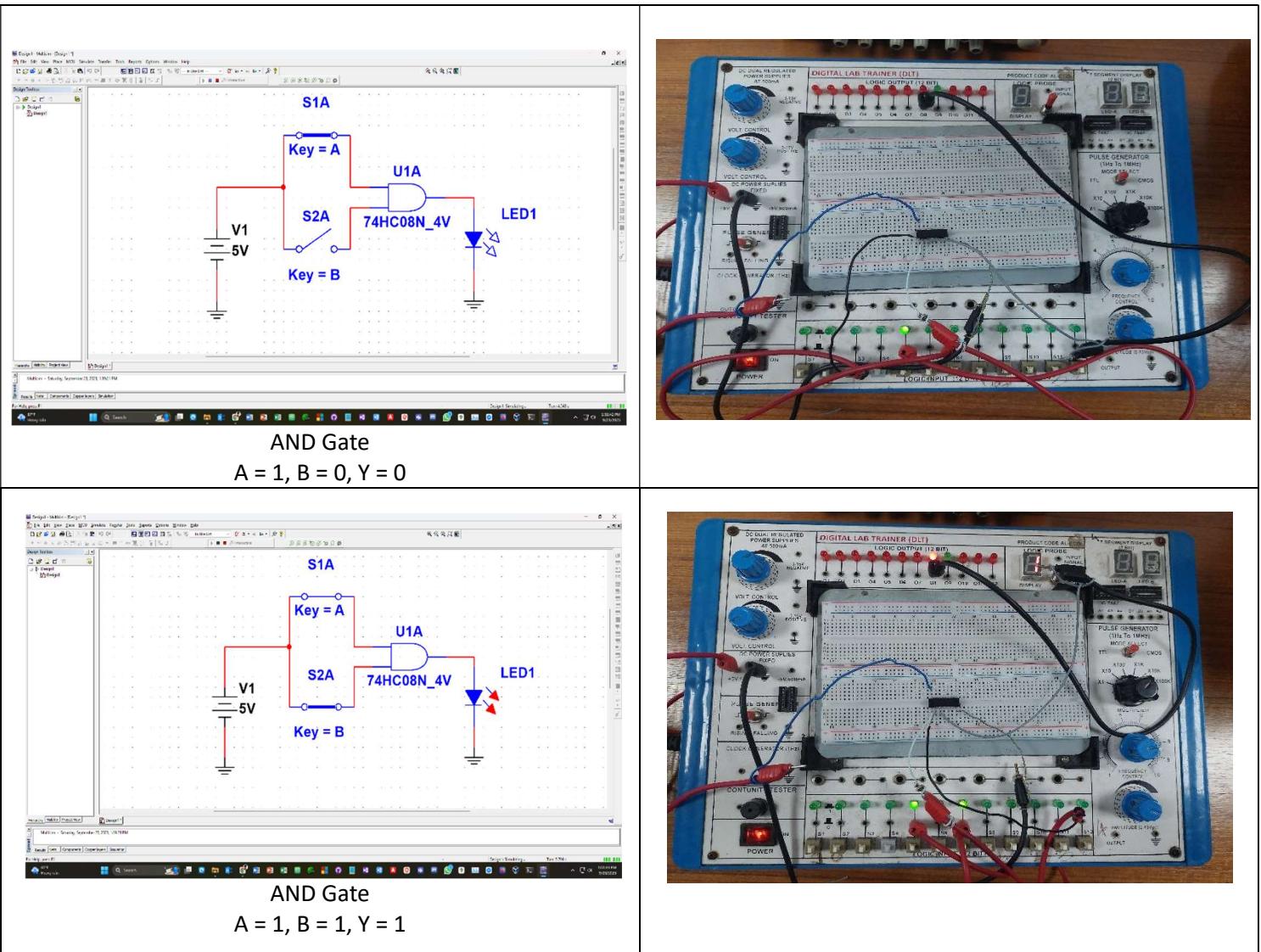


FIG – 1: Symbol of AND gate

SIMULATION	HARDWARE
<p style="text-align: center; font-weight: bold;">AND Gate</p> <p style="text-align: center;">A = 0, B = 0, Y = 0</p>	
<p style="text-align: center; font-weight: bold;">AND Gate</p> <p style="text-align: center;">A = 0, B = 1, Y = 0</p>	



OR Gate

Truth table:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

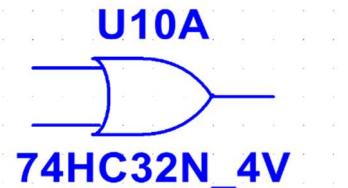
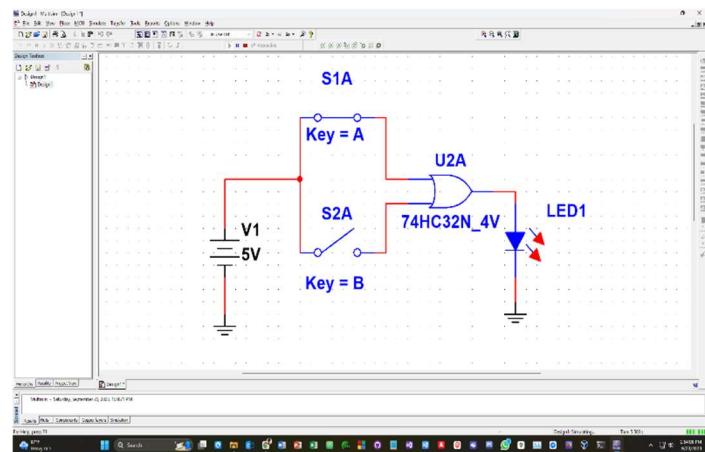


FIG – 2: Symbol of AND gate

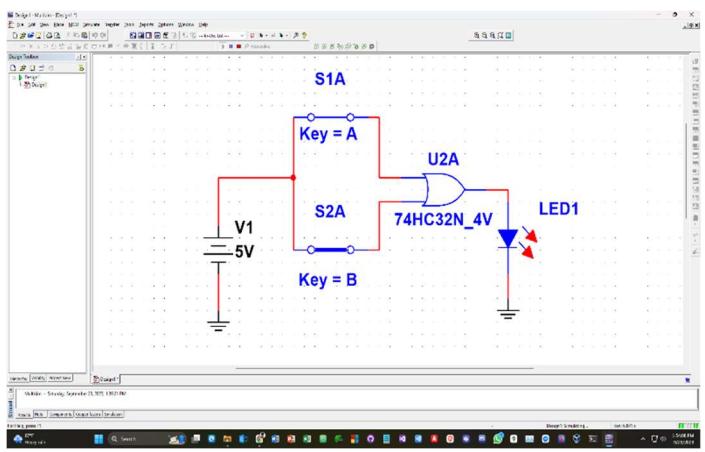
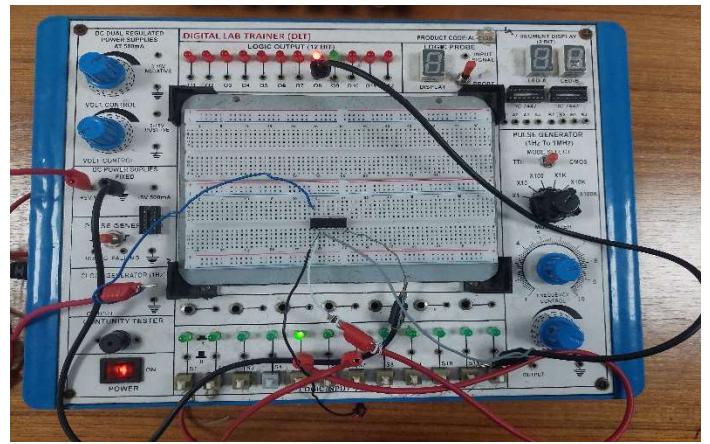
SIMULATION	HARDWARE
<p>OR Gate A = 0, B = 0, Y = 0</p>	<p>A photograph of the breadboard setup for the OR gate. The circuit is connected to a digital lab trainer board. The inputs A and B are connected to pins 1 and 2 of the 74HC32N_4V chip, respectively. The output of the chip is connected to pin 3 of U2A, which is an inverter. The output of U2A is connected to the base of a transistor, which drives an LED. The collector of the transistor is connected to ground. The digital lab trainer board has various knobs and switches for power and signal generation.</p>
<p>OR Gate A = 0, B = 1, Y = 1</p>	<p>A photograph of the breadboard setup for the OR gate. The circuit is connected to a digital lab trainer board. The inputs A and B are connected to pins 1 and 2 of the 74HC32N_4V chip, respectively. The output of the chip is connected to pin 3 of U2A, which is an inverter. The output of U2A is connected to the base of a transistor, which drives an LED. The collector of the transistor is connected to ground. The digital lab trainer board has various knobs and switches for power and signal generation.</p>

SIMULATION

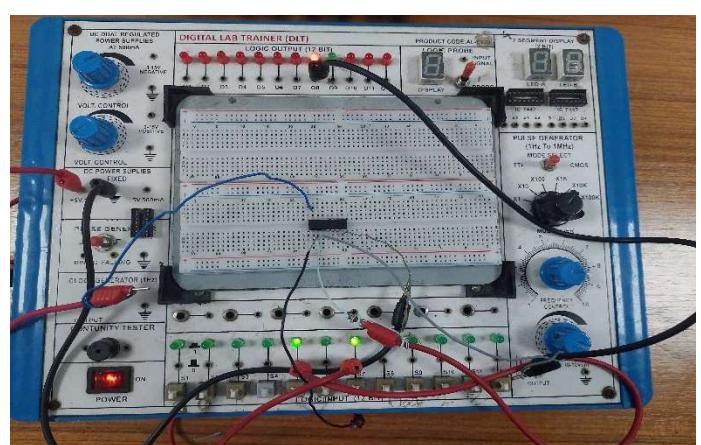


OR Gate
 $A = 1, B = 0, Y = 1$

HARDWARE



OR Gate
 $A = 1, B = 1, Y = 1$



NOT Gate

Truth table:

A	Y
0	1
1	0

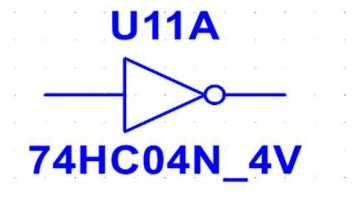


FIG – 2: Symbol of NOT gate

SIMULATION	HARDWARE
<p>NOT Gate A = 0, Y = 1</p>	
<p>NOT Gate A = 1, Y = 0</p>	

NAND Gate

Truth table:

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

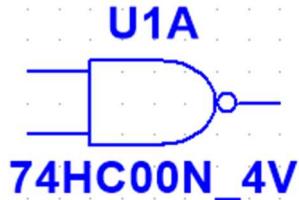
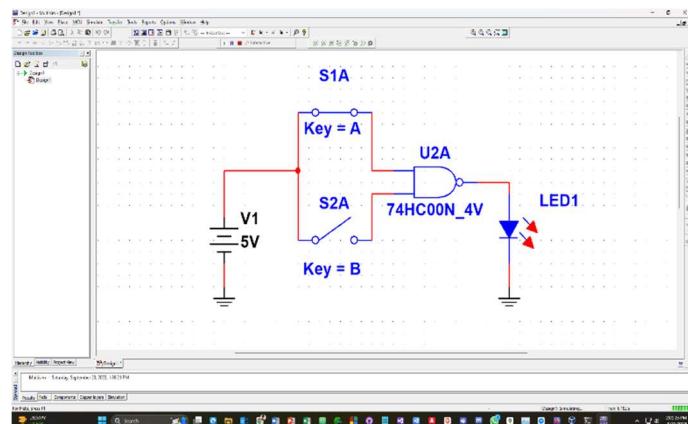


FIG – 1: Symbol of NAND gate

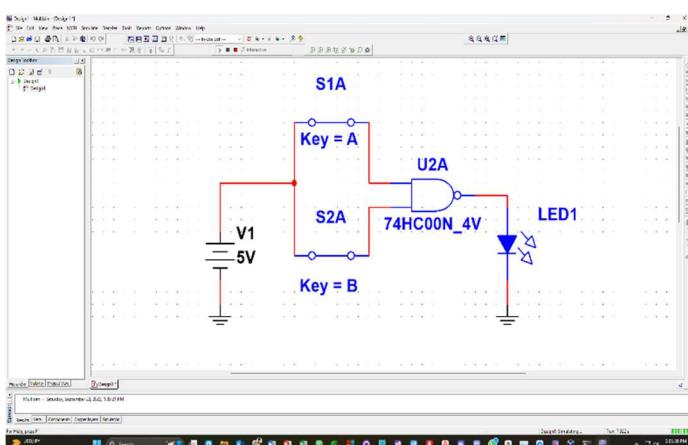
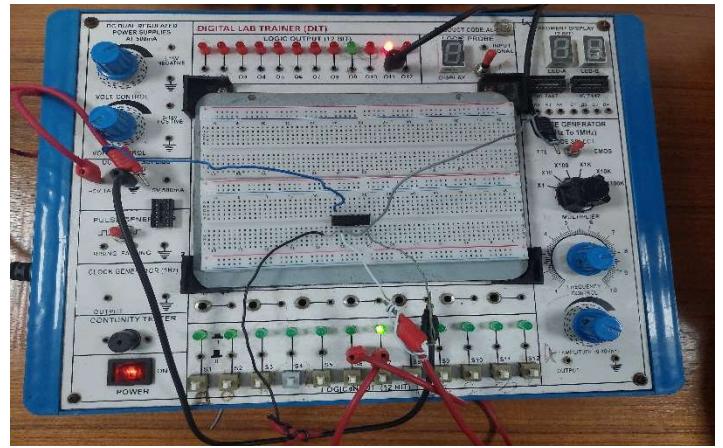
SIMULATION	HARDWARE
<p>NAND Gate A = 0, B = 0, Y = 1</p>	
<p>NAND Gate A = 0, B = 1, Y = 1</p>	

SIMULATION

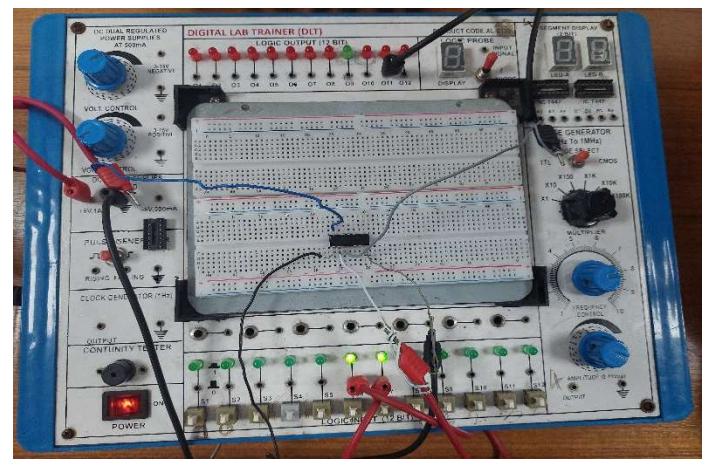


NAND Gate
A = 1, B = 0, Y = 1

HARDWARE



NAND Gate
A = 1, B = 1, Y = 0



NOR Gate

Truth table:

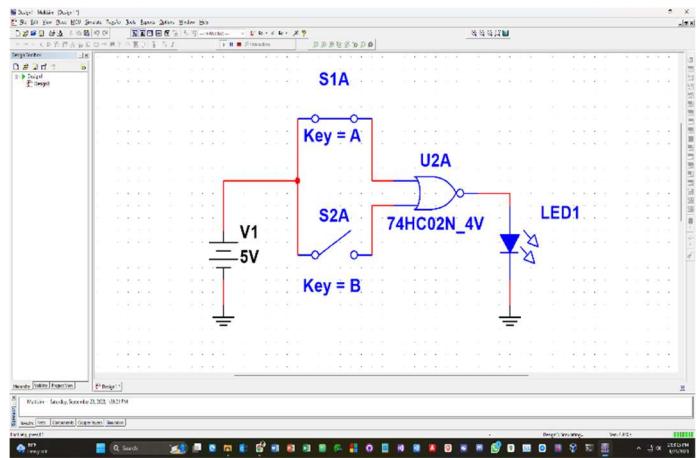
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



FIG – 1: Symbol of NOR gate

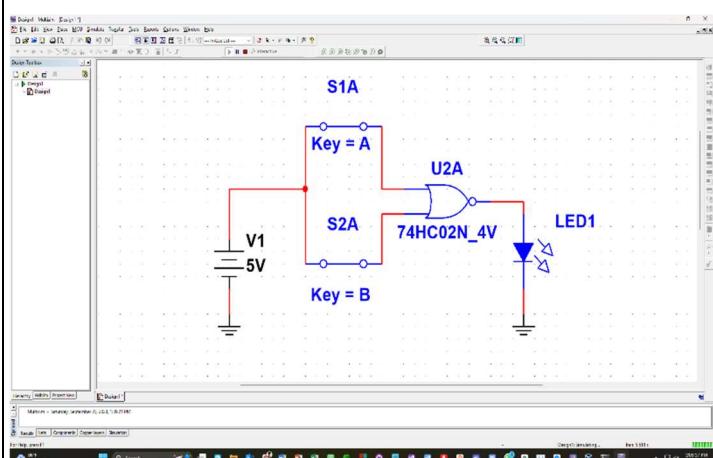
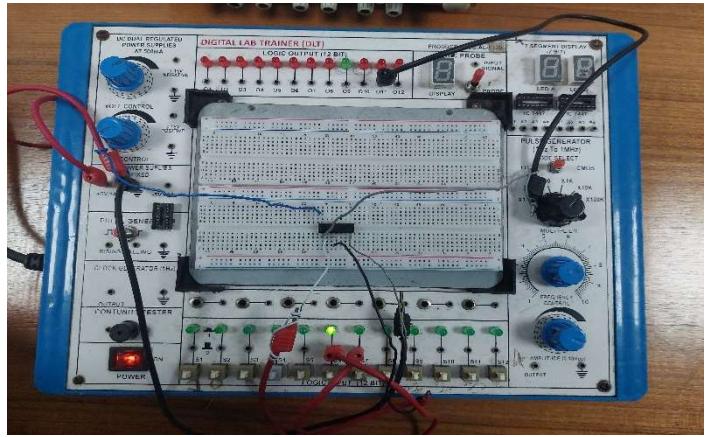
SIMULATION	HARDWARE
<p>NOR Gate A = 0, B = 0, Y = 1</p>	
<p>NOR Gate A = 0, B = 1, Y = 0</p>	

SIMULATION

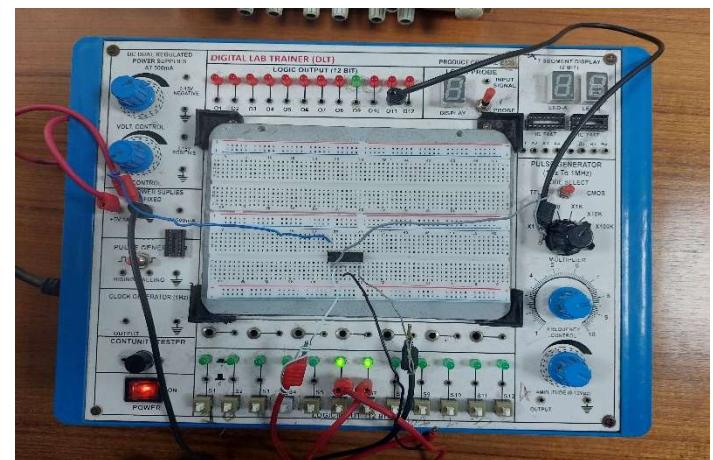


NOR Gate
A = 1, B = 0, Y = 0

HARDWARE



NOR Gate
A = 1, B = 1, Y = 0



XOR Gate

Truth table:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

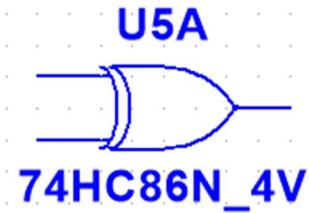
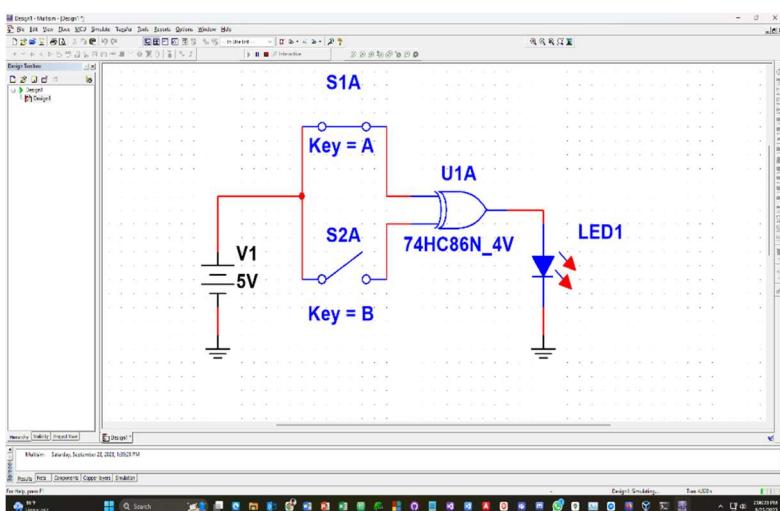


FIG – 1: Symbol of XOR gate

SIMULATION	HARDWARE
<p style="text-align: center; margin-top: 5px;"> XOR Gate $A = 0, B = 0, Y = 0$ </p>	
<p style="text-align: center; margin-top: 5px;"> XOR Gate $A = 0, B = 1, Y = 1$ </p>	

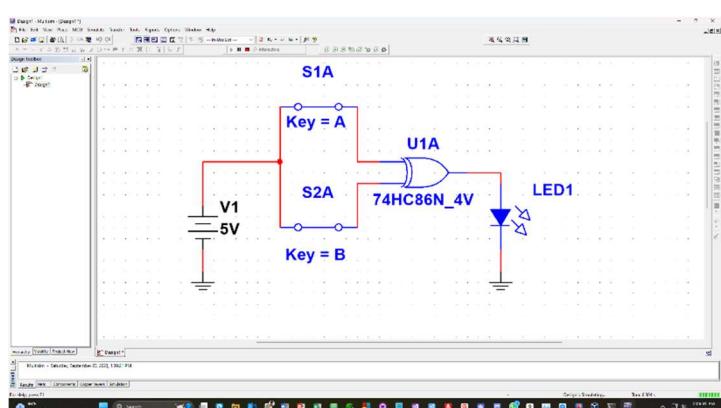
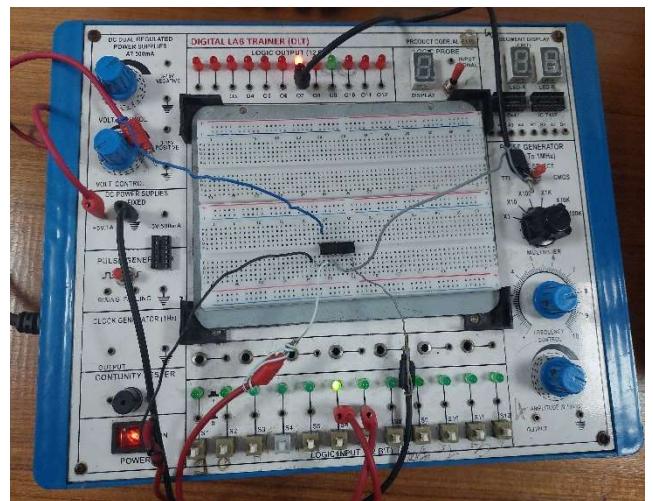
SIMULATION



XOR Gate

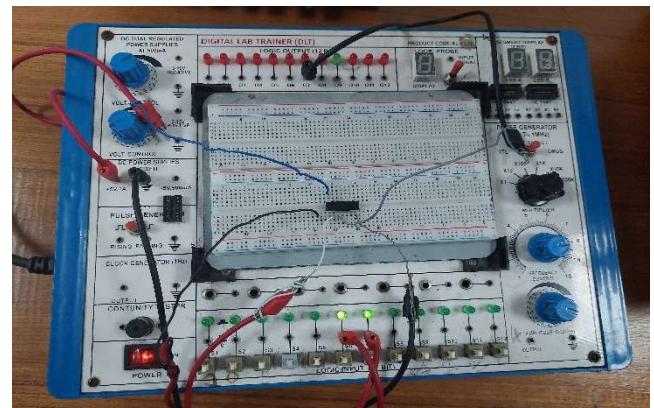
A = 1, B = 0, Y = 1

HARDWARE



XOR Gate

A = 1, B = 1, Y = 0



X-NOR Gate

Truth table:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

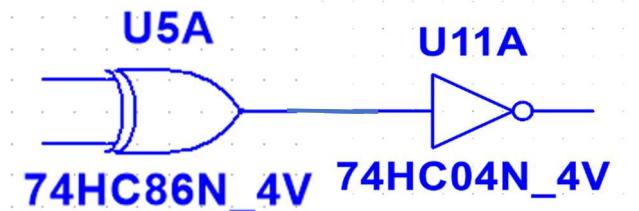
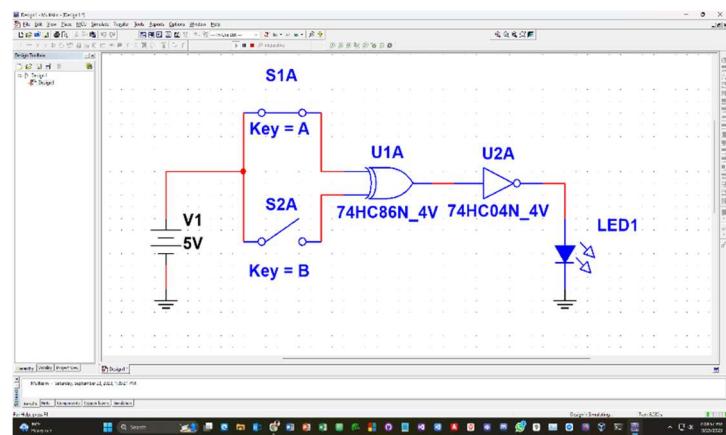


FIG – 1: Symbol of XNOR gate

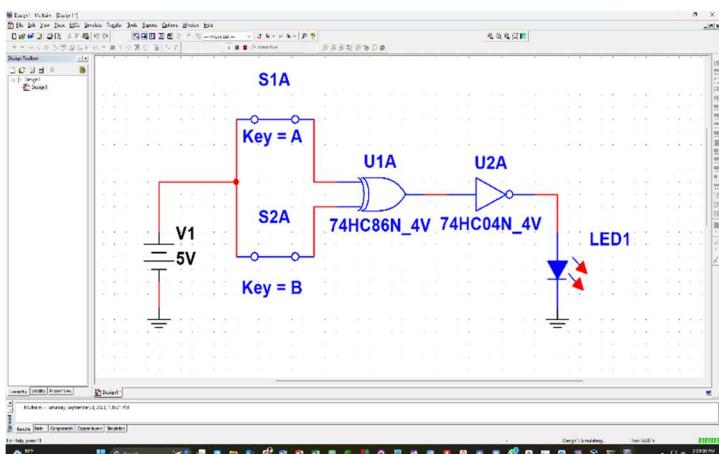
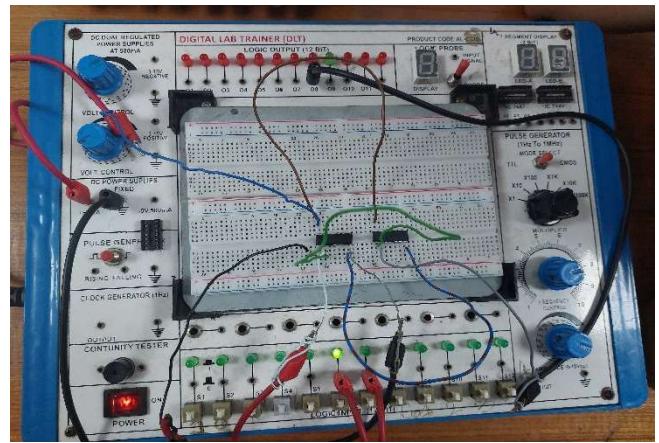
<p>XNOR Gate $A = 0, B = 0, Y = 1$</p>	<p>A photograph of the XNOR gate circuit built on a blue Digital Lab Trainer (DLT) breadboard. The circuit is identical to the one in the screenshot above. It uses a 5V power source (V1), two switches (S1A and S2A) labeled "Key = A" and "Key = B" respectively, and an LED labeled "LED1". The logic gates are labeled U1A and U2A, corresponding to the 74HC86N_4V and 74HC04N_4V parts shown in the symbol. Wires connect the components according to the schematic.</p>
<p>XNOR Gate $A = 0, B = 1, Y = 0$</p>	<p>A photograph of the XNOR gate circuit built on a blue Digital Lab Trainer (DLT) breadboard. This configuration is different from the top one because switch S1A is closed while S2A is open. The logic gates U1A and U2A are present, along with the 5V power source V1 and the LED LED1. The connections are similar to the first setup but reflect the change in input conditions.</p>

SIMULATION

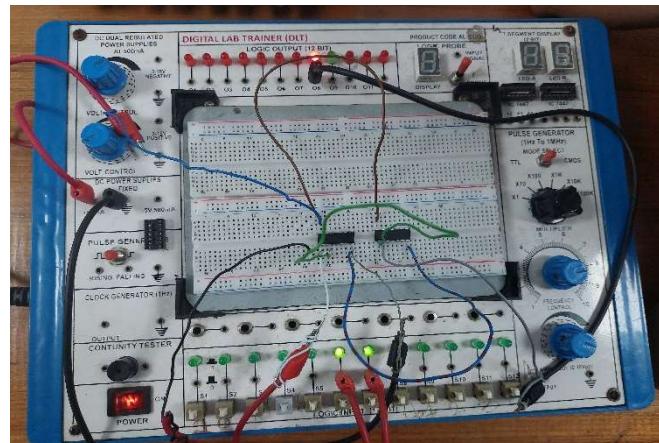


XNOR Gate
A = 1, B = 0, Y = 0

HARDWARE



XNOR Gate
A = 1, B = 1, Y = 1



Part II: Study of Universal Gates

A Logic Gate which can infer any of the gate among Logic Gates or a gate which can be used to create any Logic gate is called Universal Gate. **NAND** and **NOR** Gates are called Universal Gates because all the other gates such as NOT, AND, OR, XOR, XNOR etc can be created by using these gates.

The Objective of this lab is to implement different logic functions using universal gates.

Theory and Methodology:

NAND gate:

The graphic symbol for the NAND gate consists of an AND symbol with a bubble on the output, denoting that a complement operation is performed on the output of the AND gate.



Fig 2.1: Symbol of NAND

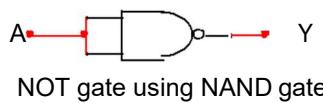
$$\begin{array}{c} \text{gate Output, } Q = \\ + \end{array}$$

Truth Table		
Input A	Input B	Output Q
0	0	1
0	1	1
1	0	1
1	1	0

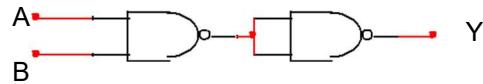
It is possible to construct other gates using NAND gates which are shown in Experimental procedure part.

Implementing various logic functions using NAND Gates:

- 1) Implementing NOT gate using NAND gate:

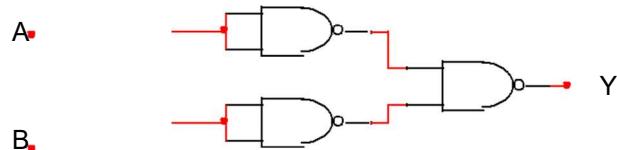


2) Implementing AND gate using NAND gate:



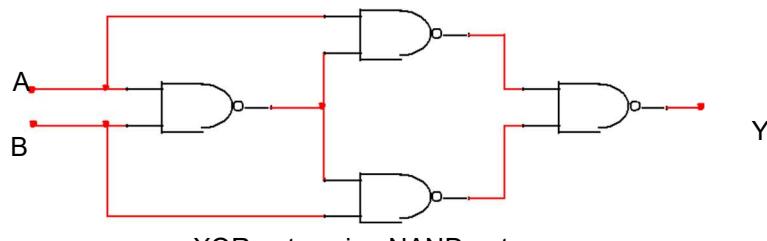
AND gate using NAND gates

3) Implementing OR gate using NAND gate:



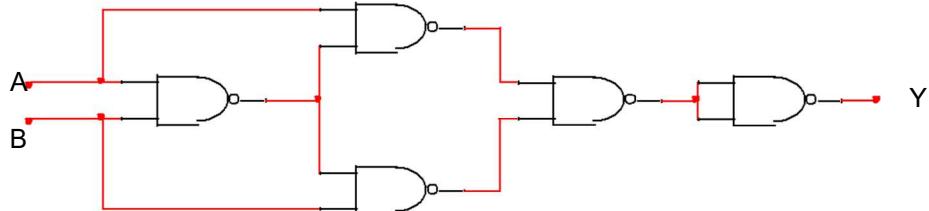
OR gate using NAND gates

4) Implementing XOR gate using NAND gate:



XOR gate using NAND gates

5) Implementing XNOR gate using NAND gate:



XNOR gate using NAND gates

NOR gate:

The **NOR** gate represents the complement of the OR operation. Its name is an abbreviation of **NOT OR**. The graphic symbol for the NOR gate consists of an OR symbol with a bubble on the output, denoting that a complement operation is performed on the output of the OR gate. The truth table and the graphic symbol of NOR gate is shown in the figure.



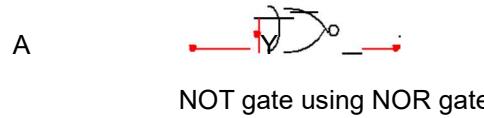
Fig 2.2: Symbol of NOR

gate Output,Q= =

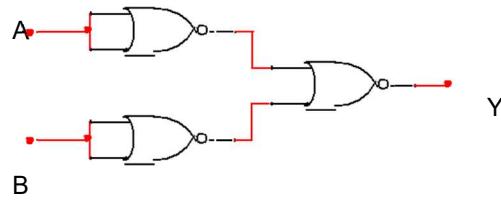
Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	0

Implementing various logic functions using NOR Gates:

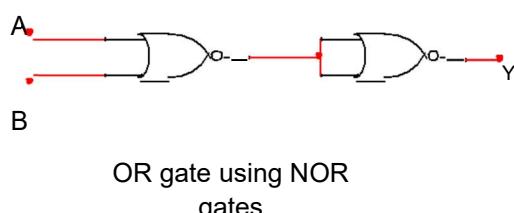
- 1) Implementing NOT gate using NOR gate:



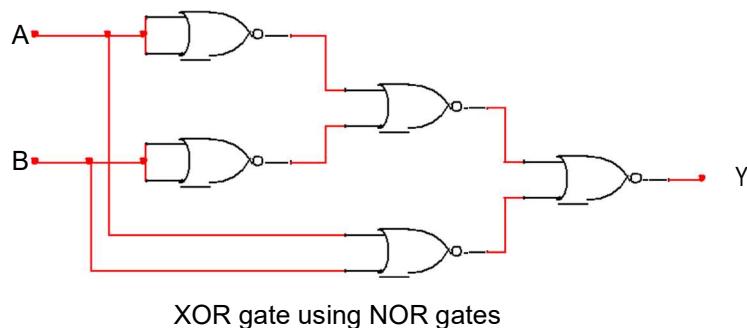
- 2) Implementing AND gate using NOR gate:



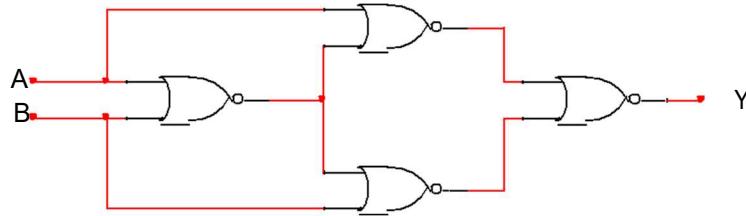
- 3) Implementing OR gate using NOR gate:



- 4) Implementing XOR gate using NOR gate:



- 5) Implementing XNOR gate using NOR gate:



XNOR gate using NOR gates

Pre-Lab Homework:

Students must study the Boolean algebra rules and universal gates, perform simulation of the circuits shown in the circuit diagram section using Power Sim 9.1.1 (PSIM) and MUST present the simulation results to the instructor before the start of the experiment.

Apparatus:

1. Digital trainer board.
2. Integrated Circuits (ICs).
3. Power supply.
4. Connecting wires.

Precautions:

Have your instructor check all your connections after you are done setting up the circuit and make sure that you apply only enough voltage to turn on the chip, otherwise it may get damaged.

Experimental Procedure:

1. Construct an X-OR and X-NOR gate in your trainer board by using NAND gates only. Use required IC to construct the circuit.
2. Find out the equivalent NOT, OR and AND gate by using NOR gates only. Now construct an X-OR and X-NOR gate in your trainer board by using NOR gates only. Use required IC to construct the circuit.
3. Convert the following expressions using universal gates and implement them in the trainer board. Compare the results with the truth table of the equations.
 - i) $A (+) B$
 - ii) $(A(+B)) + C$
 - iii) $(AB + CD)'$

PART 2

$A \oplus B$

Truth table:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

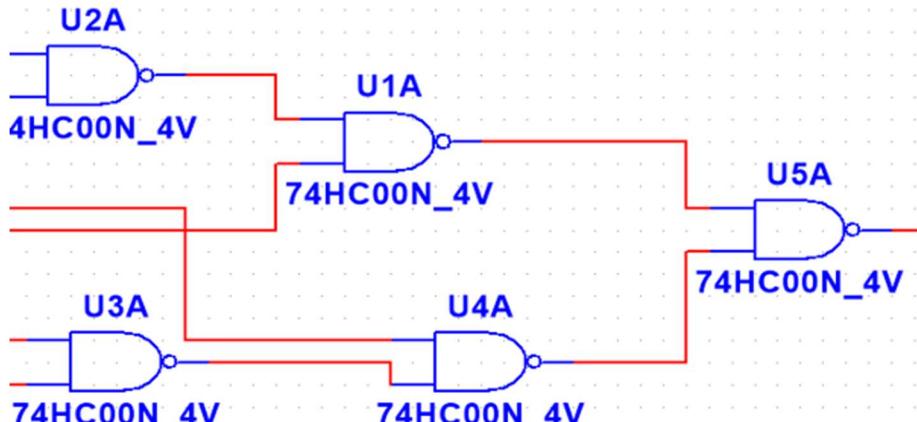
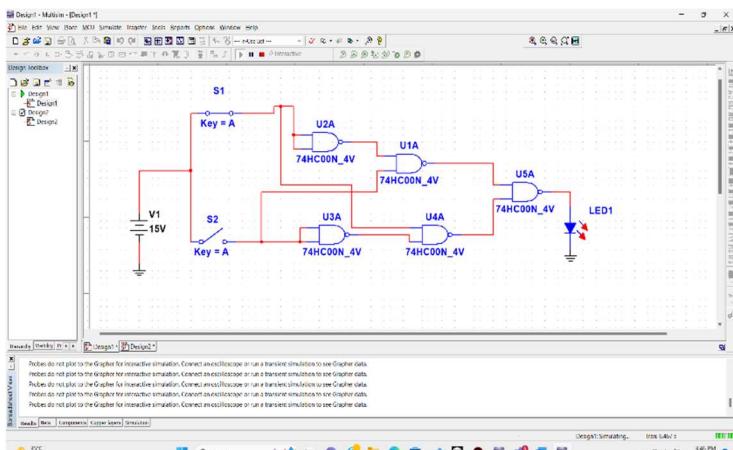


FIG – 1: Symbol of $A \oplus B$ gate

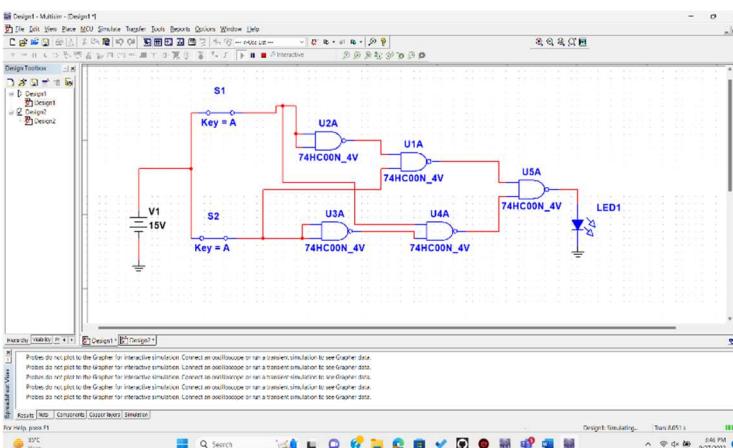
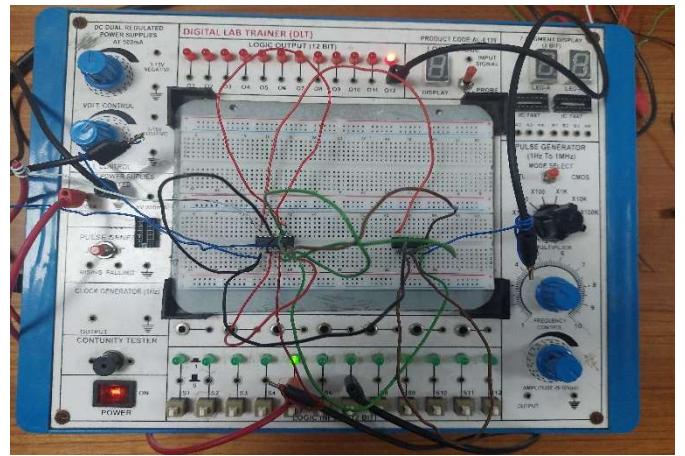
SIMULATION	HARDWARE
<p>$A \oplus B$ Gate A = 0, B = 0, Y = 0</p>	
<p>$A \oplus B$ Gate A = 0, B = 1, Y = 1</p>	

SIMULATION

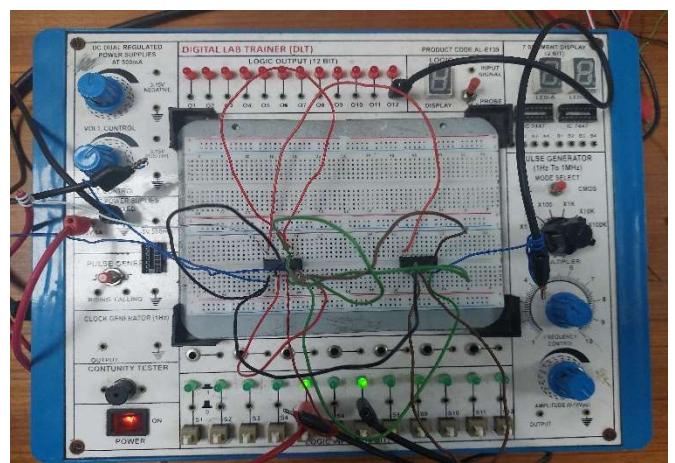


$A \oplus B$ Gate
 $A = 1, B = 0, Y = 1$

HARDWARE



$A \oplus B$ Gate
 $A = 1, B = 1, Y = 0$



$$(A \oplus B) + C$$

Truth table:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

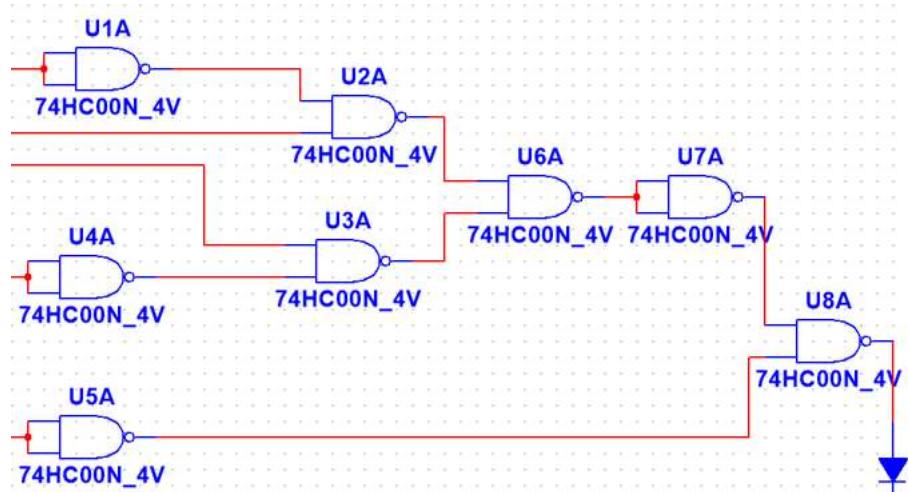
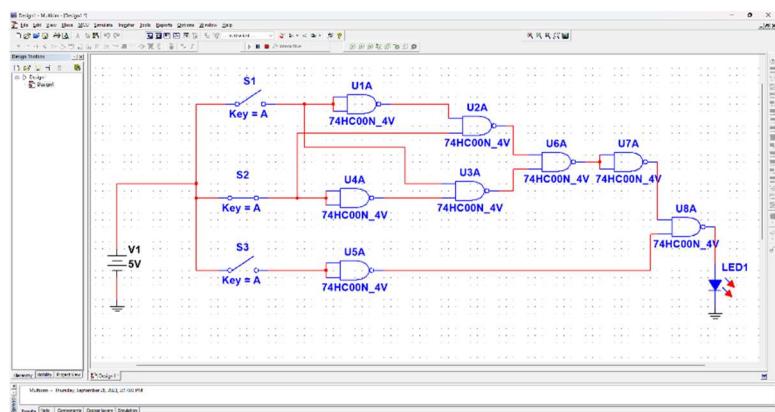


FIG – 1: Symbol of $(A \oplus B) + C$ gate

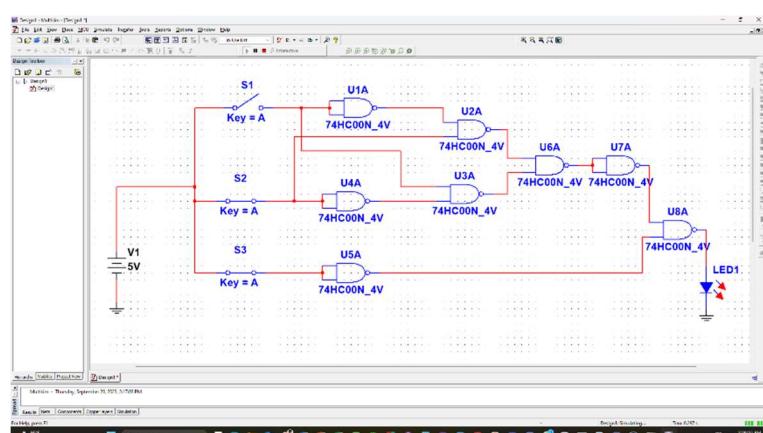
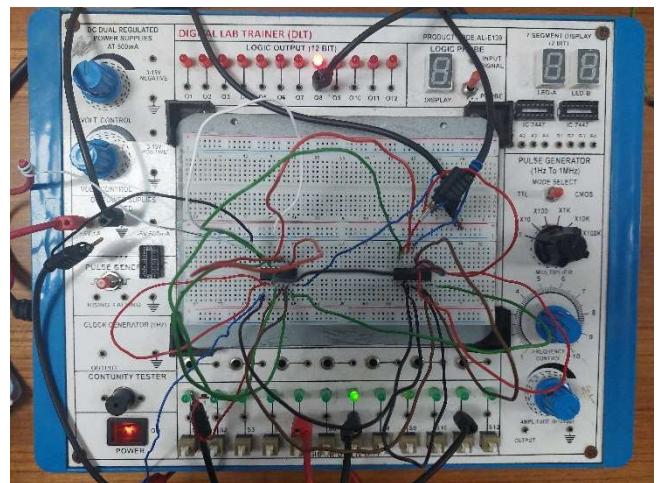
SIMULATION	HARDWARE
<p>(A ⊕ B) + C gate A = 0, B = 0, C=0, Y=0</p>	
<p>(A ⊕ B) + C gate A = 0, B = 0, C=1, Y=1</p>	

SIMULATION

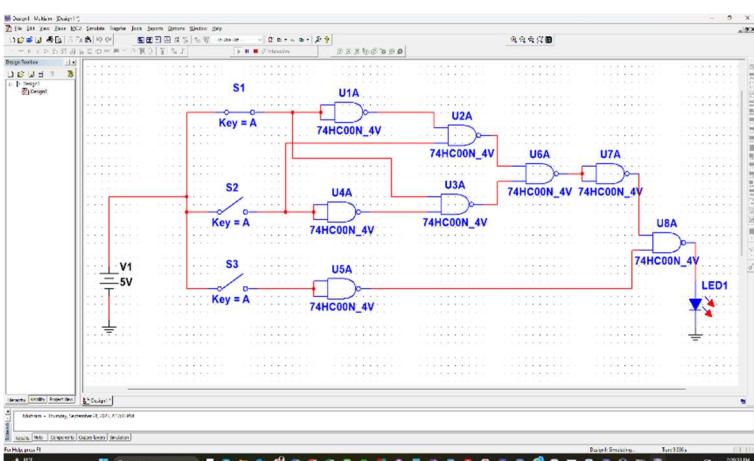
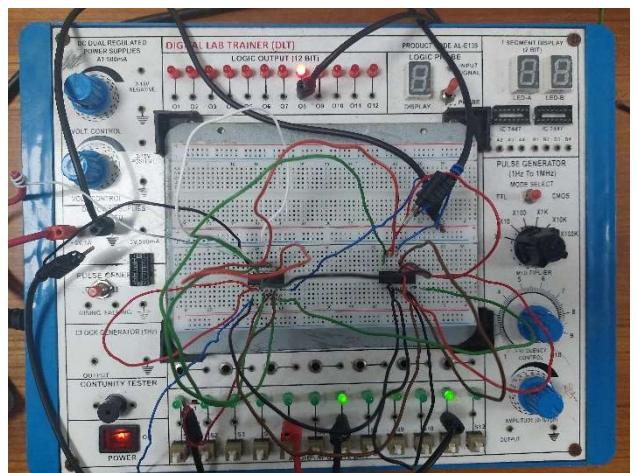


$(A \oplus B) + C$ gate
A = 0, B = 1, C=0, Y =1

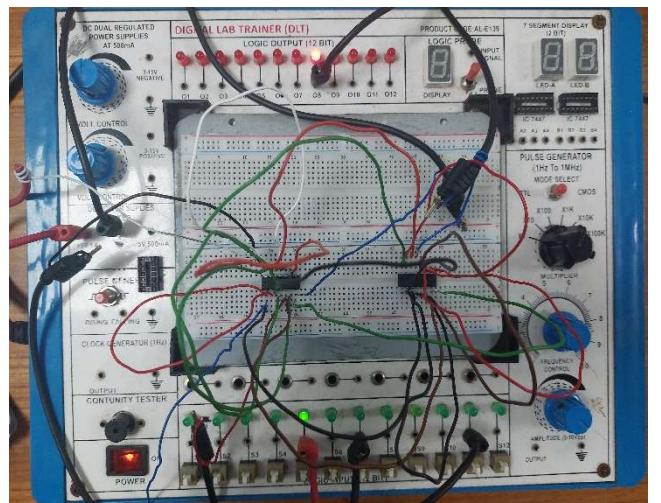
HARDWARE



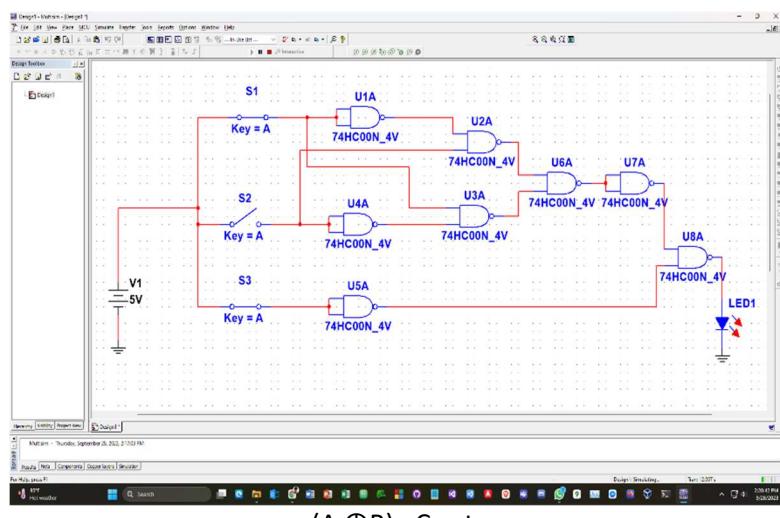
$(A \oplus B) + C$ gate
A = 0, B = 1, C=1, Y =1



$(A \oplus B) + C$ gate
A = 1, B = 0, C=0, Y =1

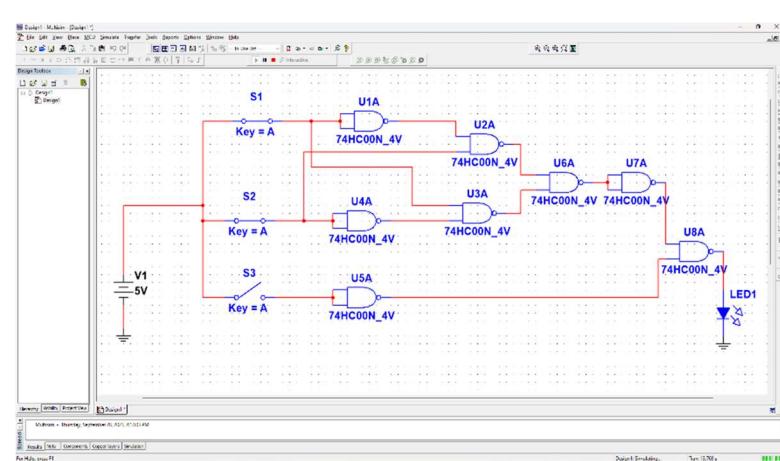
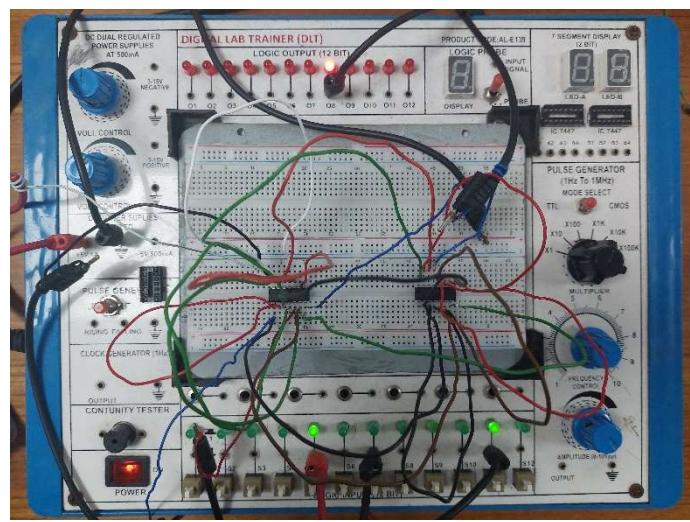


SIMULATION

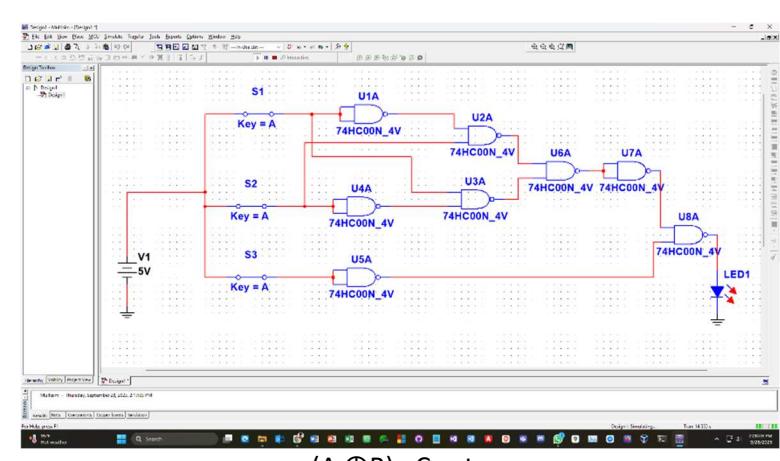
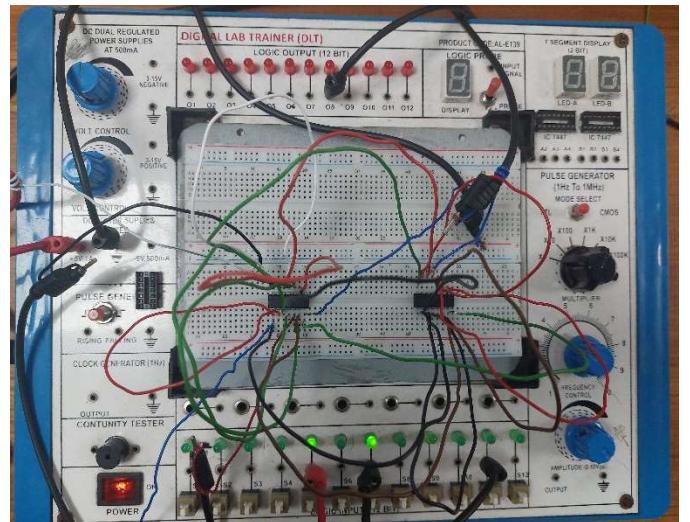


$(A \oplus B) + C$ gate
A = 1, B = 0, C=1, Y=1

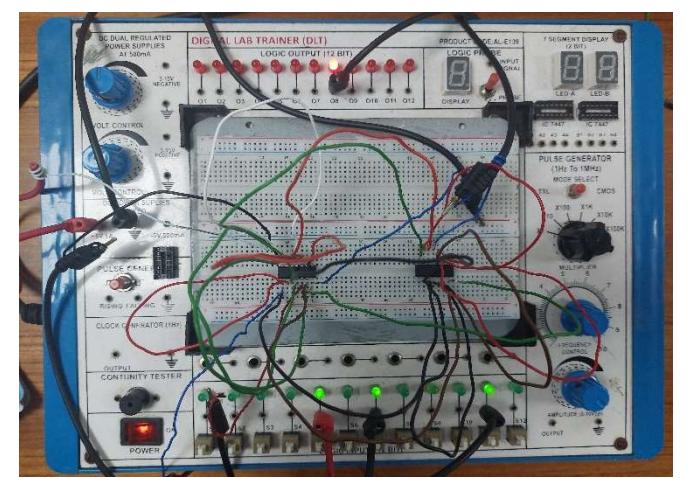
HARDWARE



$(A \oplus B) + C$ gate
A = 0, B = 1, C=0, Y=1



$(A \oplus B) + C$ gate
A = 1, B = 1, C=1, Y=1



$$\overline{AB} + DC$$

Truth table:

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

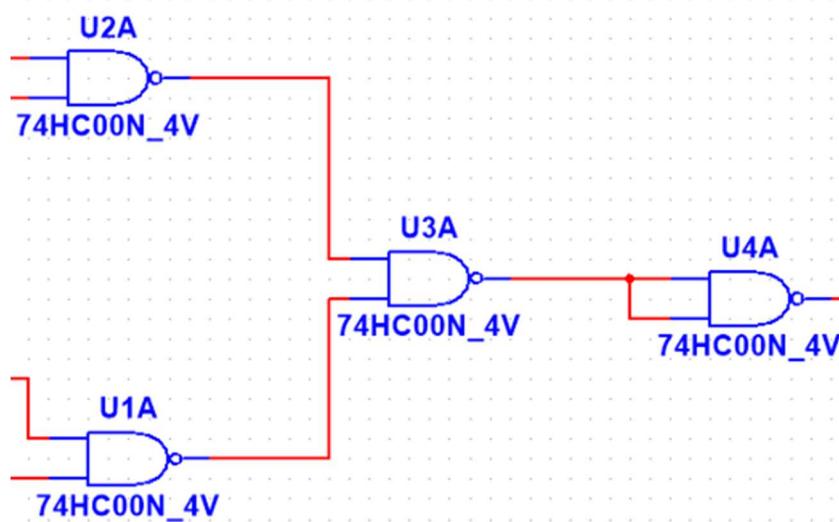
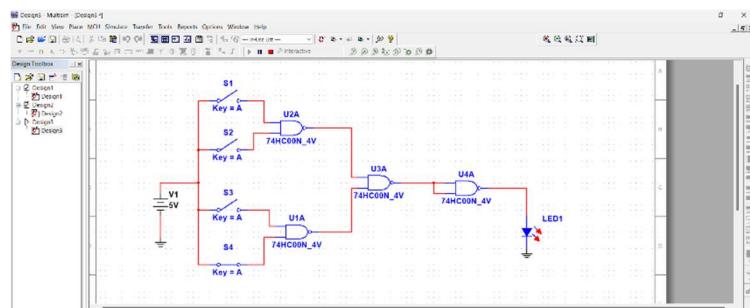


FIG – 1: Symbol of $\overline{AB}+DC$ gate

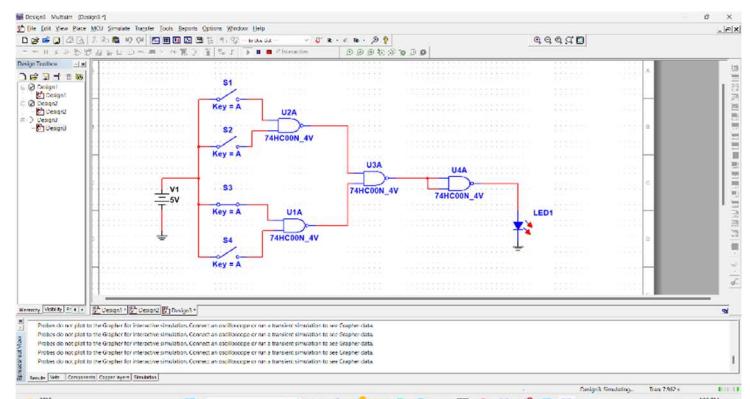
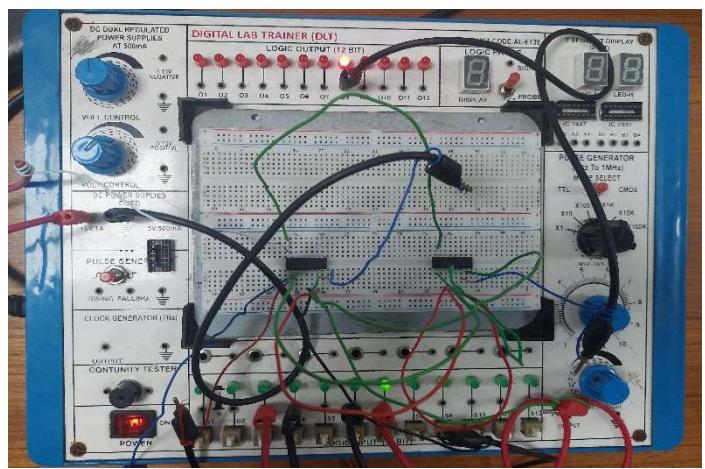
SIMULATION	HARDWARE
<p>AB + DC Gate</p> <p>A = 1, B = 1, C=1, D=0, Y =1</p>	<img alt="Photograph of a breadboard setup on a digital lab trainer. The circuit is built using 74HC00N_4V logic chips. A 5V power source is connected to the breadboard. Four AND gates (U2A, U3A, U1A, U4A) are connected in series, followed by a fifth AND gate. The output of the fifth AND gate drives an LED. Various control keys (S1-S4) are connected to the inputs of the first four AND gates.</p>

SIMULATION

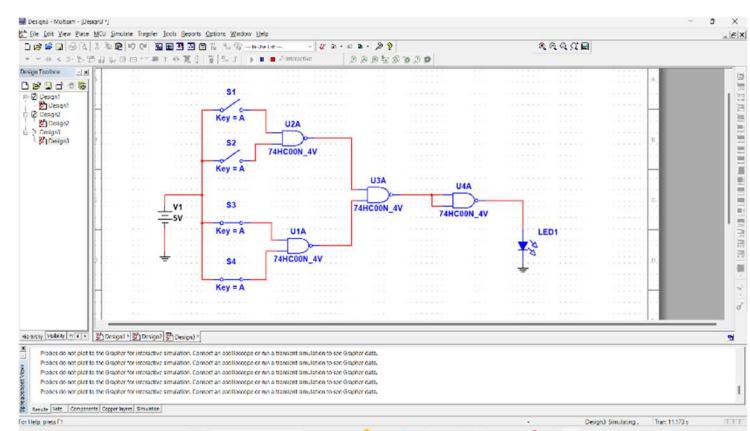
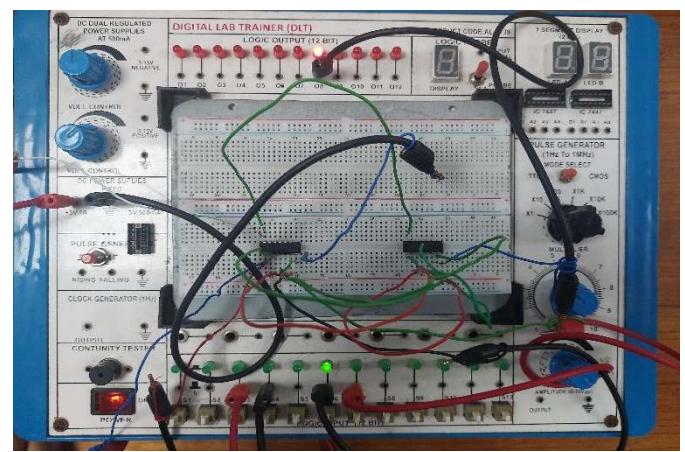


$AB + DC$ Gate
 $A = 0, B = 0, C=0, D=1, Y = 1$

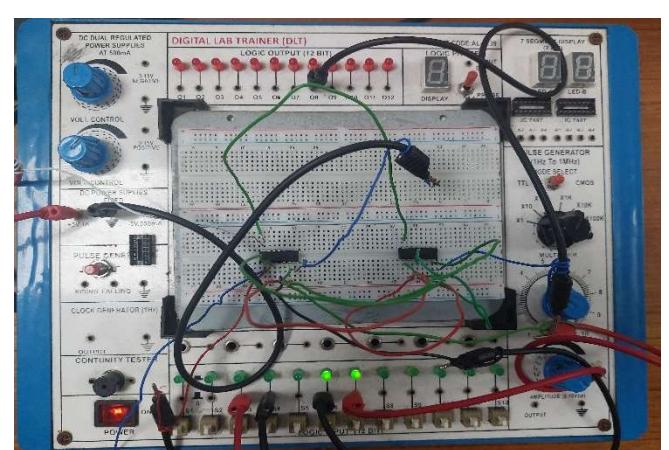
HARDWARE



$AB + DC$ Gate
 $A = 0, B = 0, C=1, D=0, Y = 1$

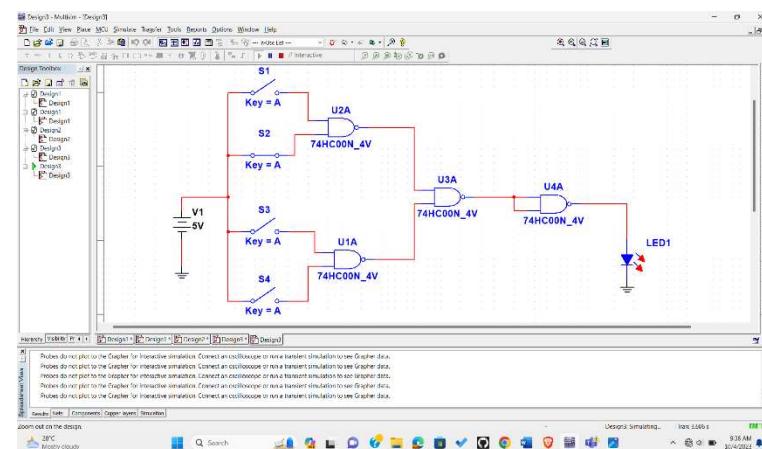


$AB + DC$ Gate
 $A = 0, B = 0, C=1, D=1, Y = 0$



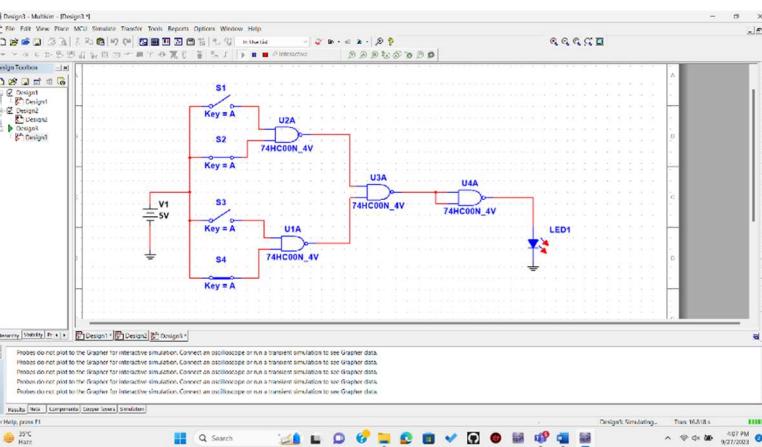
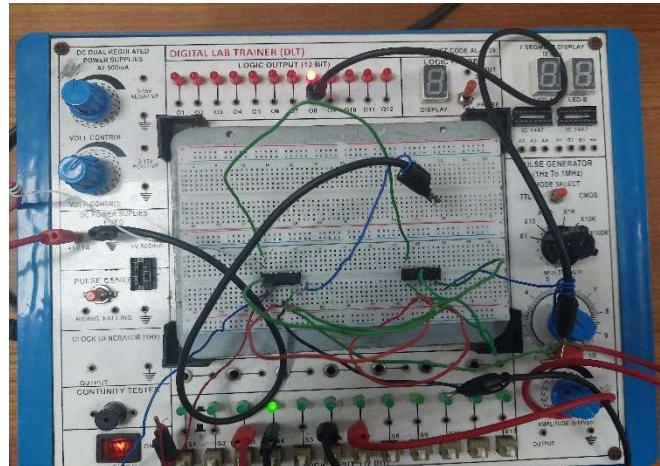
SIMULATION

HARDWARE



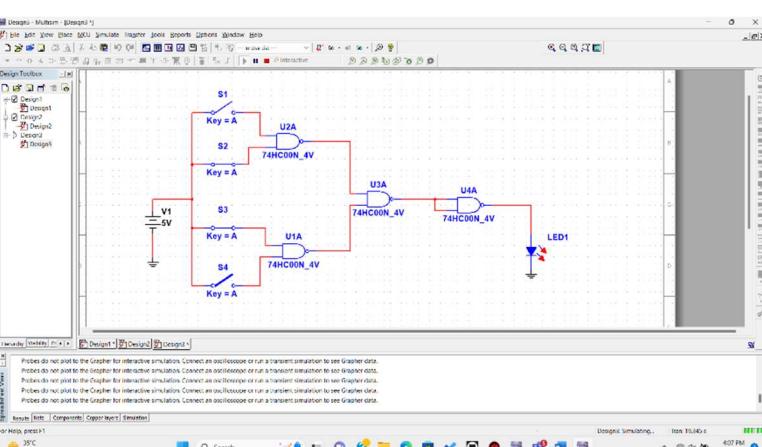
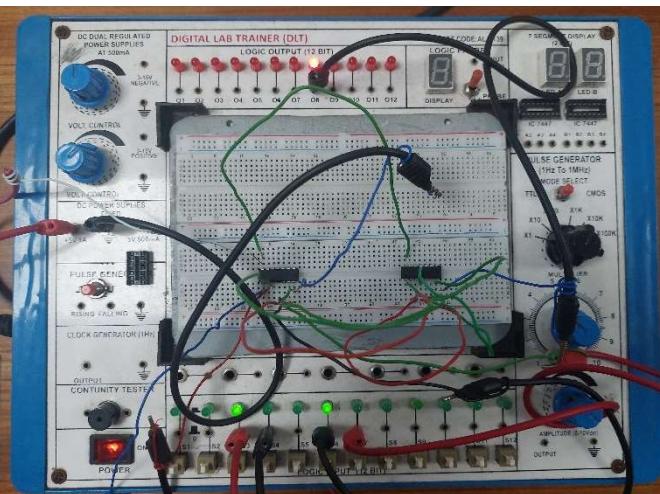
$AB + DC$ Gate

A = 0, B = 0, C=0, D=0, Y = 1



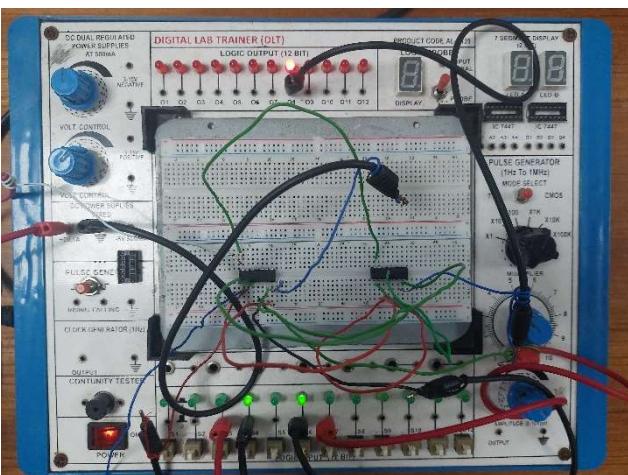
$AB + DC$ Gate

A = 0, B = 1, C=0, D=1, Y = 1

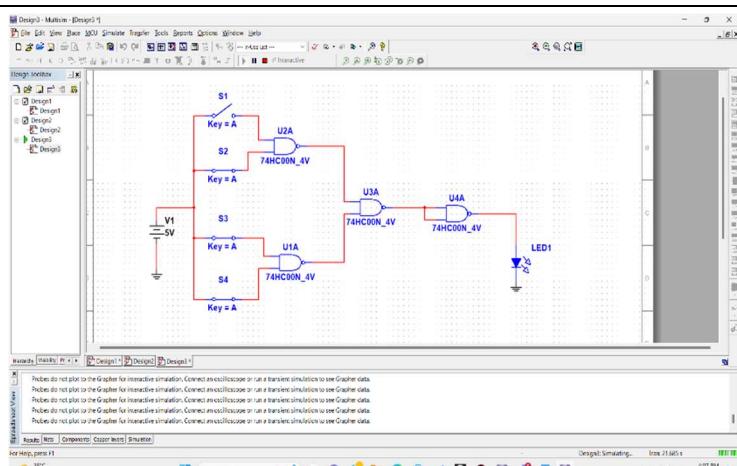


$AB + DC$ Gate

A = 0, B = 1, C=1, D=0, Y = 1

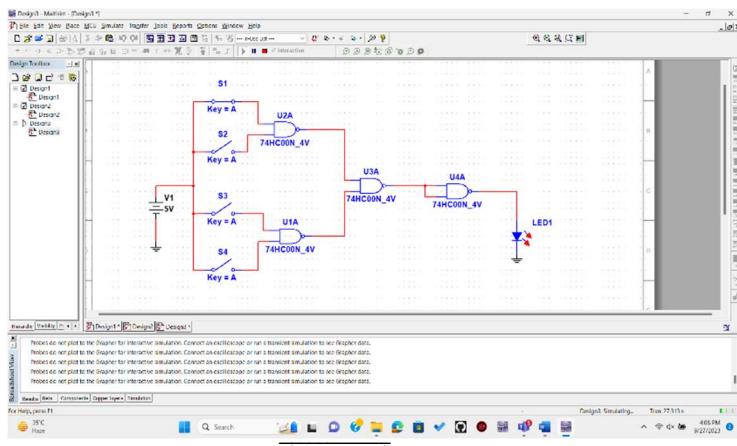
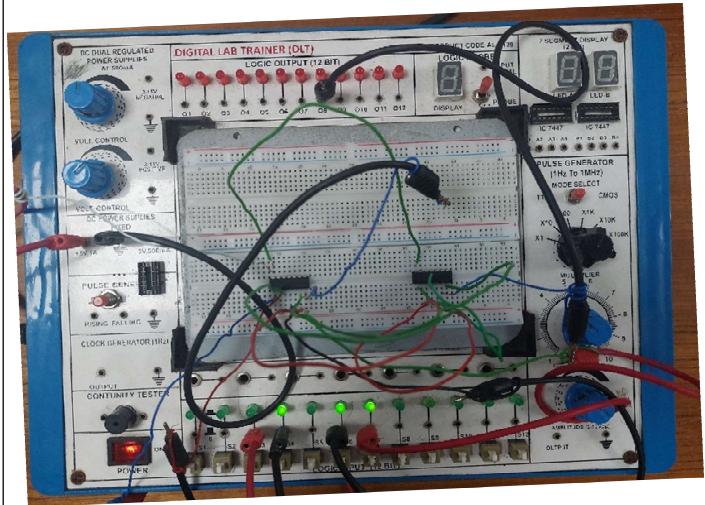


SIMULATION

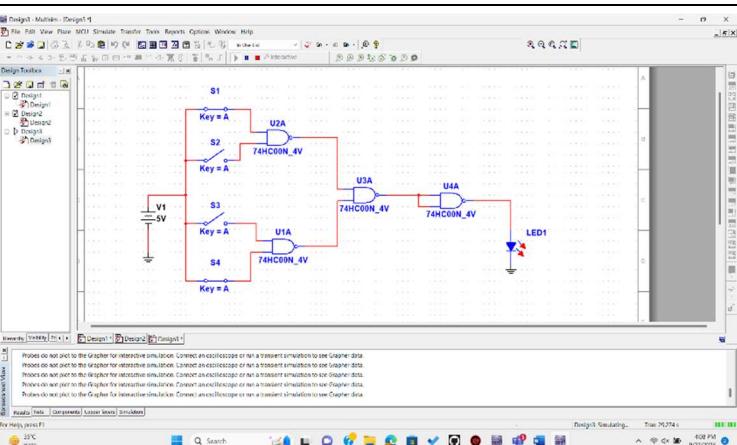
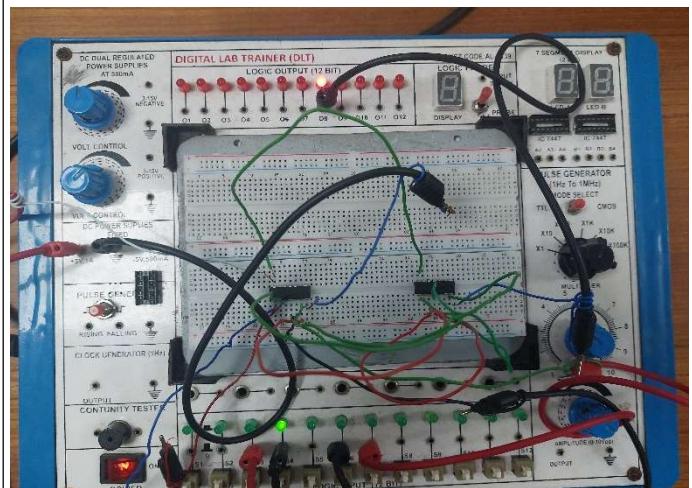


$AB + DC$ Gate
 $A = 01, B = 0, C=0, D=0, Y = 1$

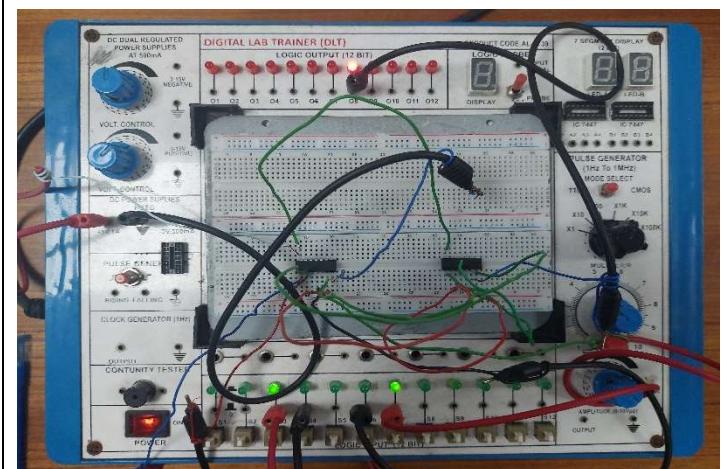
HARDWARE



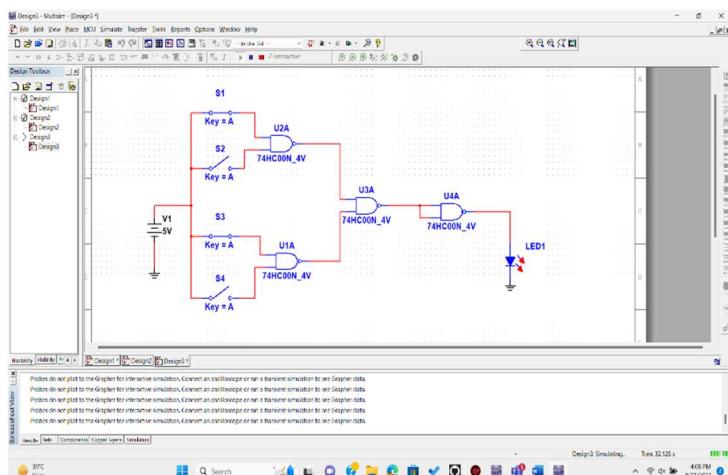
$AB + DC$ Gate
 $A = 1, B = 0, C=0, D=0, Y = 1$



$AB + DC$ Gate
 $A = 1, B =0, C=0, D=1, Y = 1$



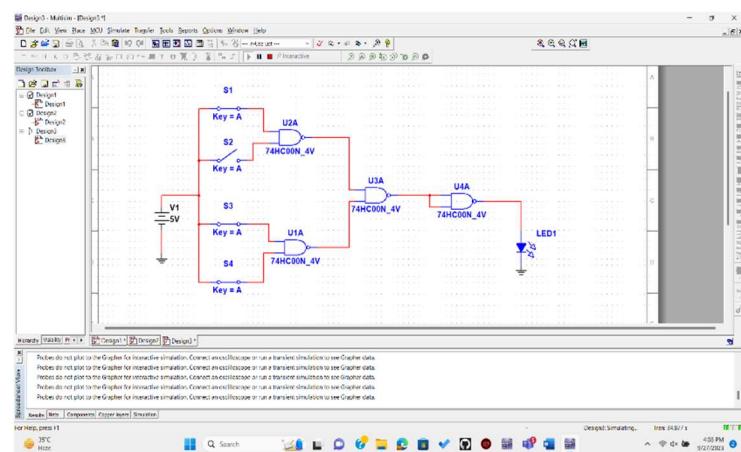
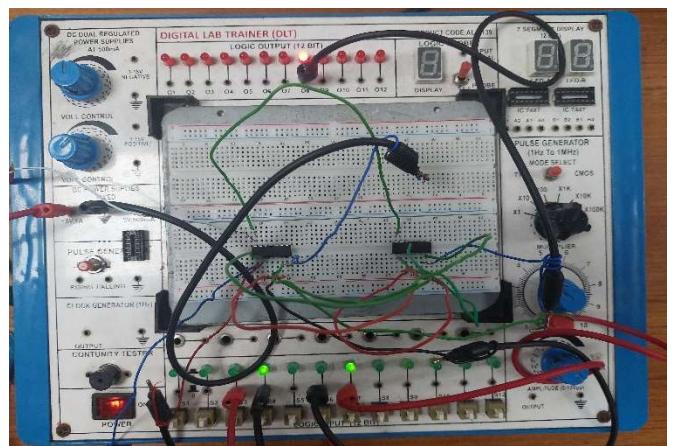
SIMULATION



$AB + DC$ Gate

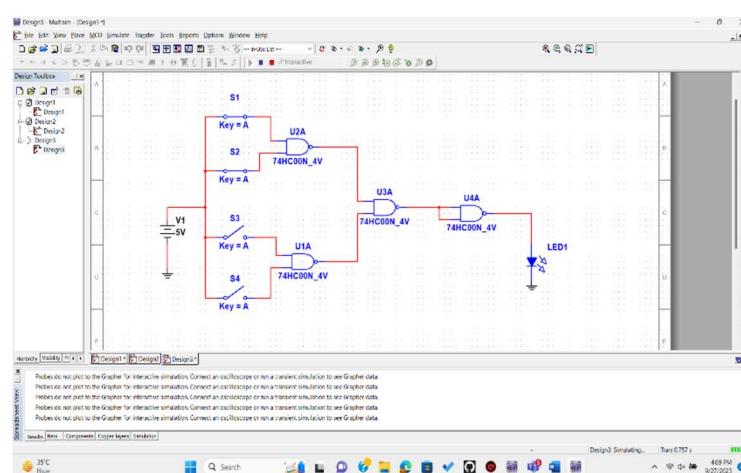
A = 0, B = 1, C=0, D=1, Y =1

HARDWARE



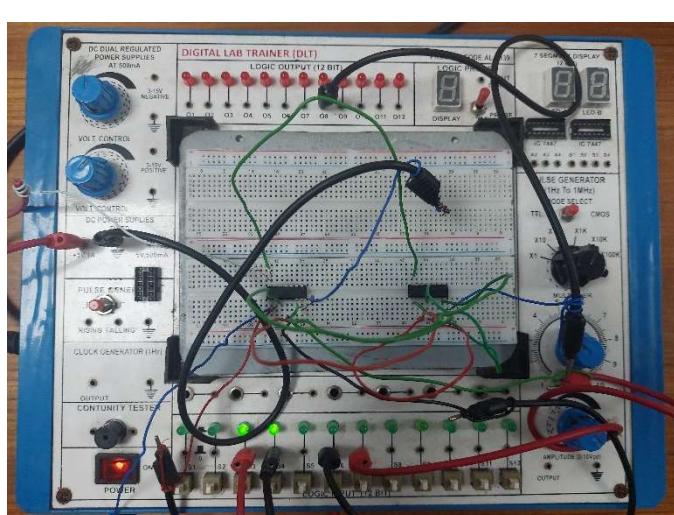
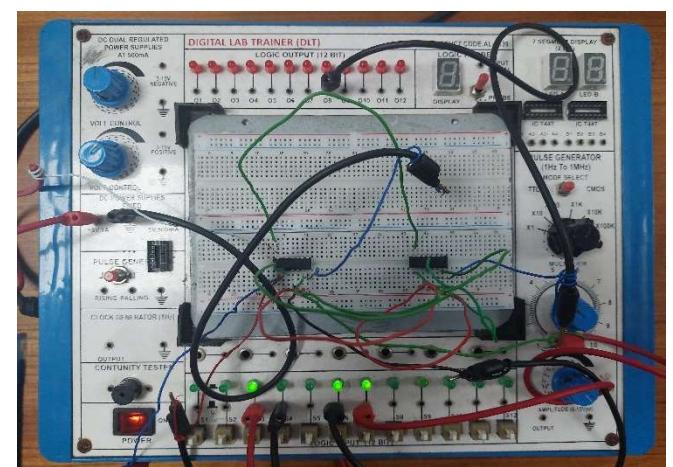
$AB + DC$ Gate

A = 1, B = 0, C=1, D=1, Y =0

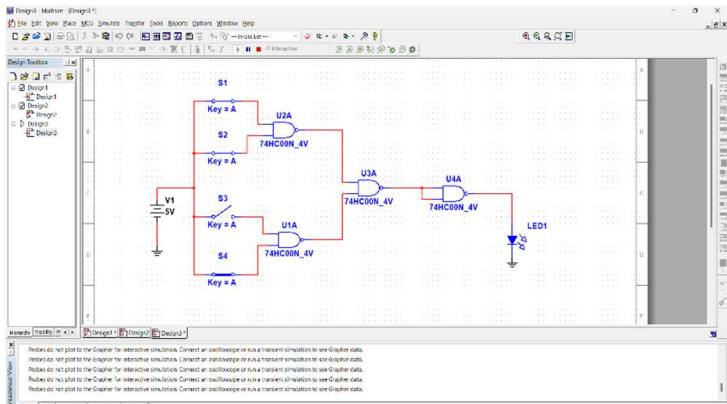


$AB + DC$ Gate

A = 0, B = 0, C=0, D=0, Y =0



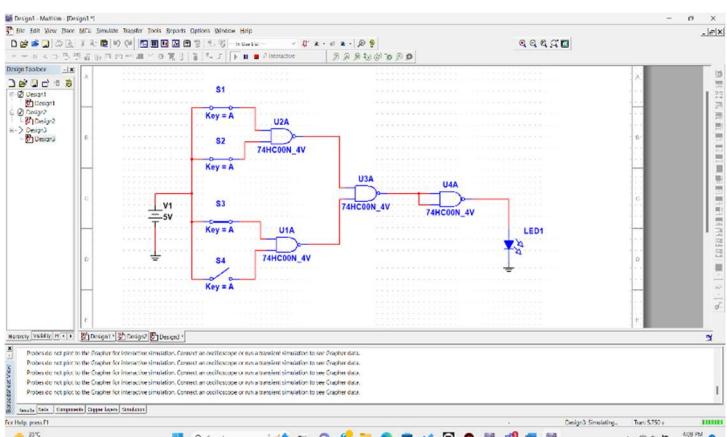
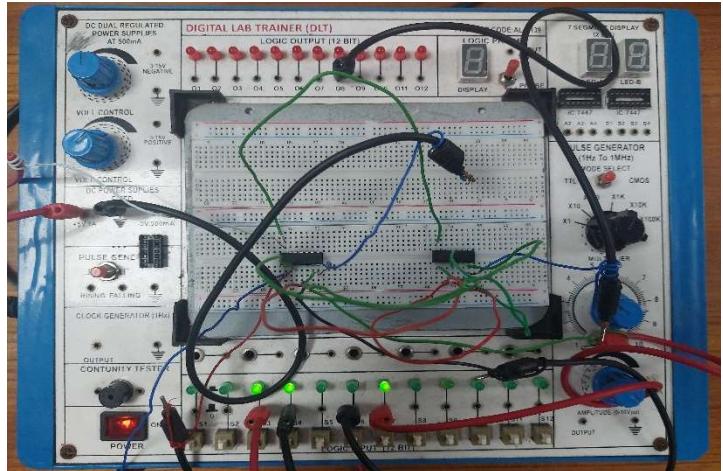
SIMULATION



$AB + DC$ Gate

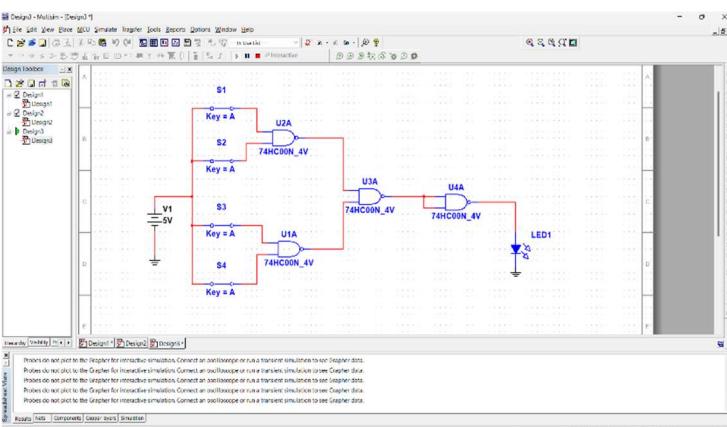
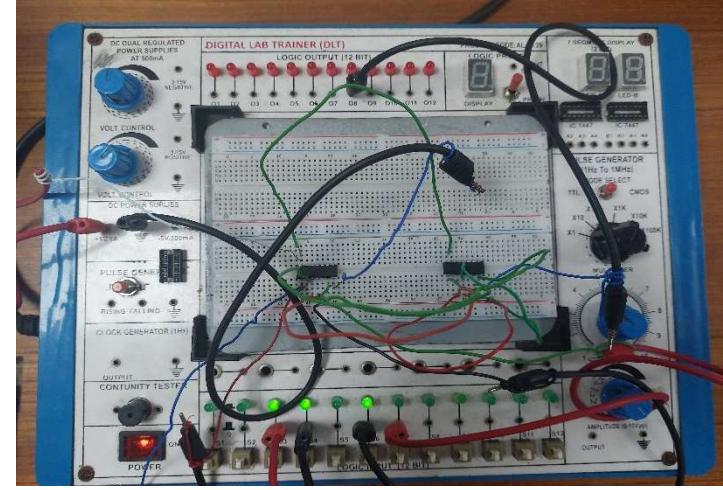
A = 1, B = 1, C=0, D=0, Y = 0

HARDWARE



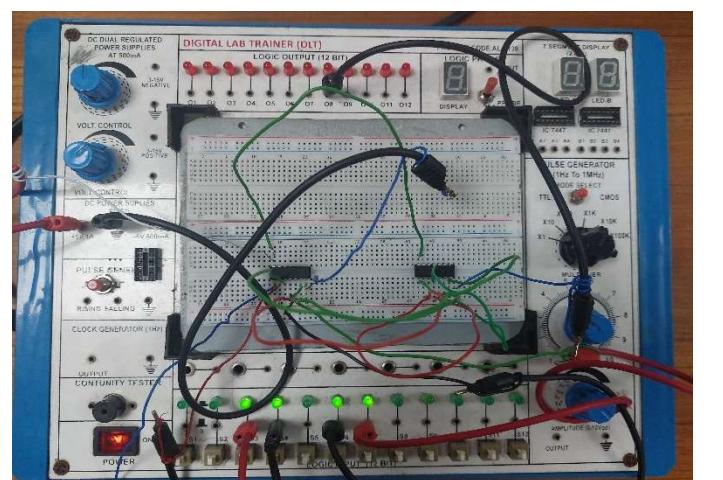
$AB + DC$ Gate

A = 1, B = 1, C=0, D=0, Y = 0



$AB + DC$ Gate

A = a, B = 1, C=1, D=1, Y = 0



$$A \oplus B$$

Truth table:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

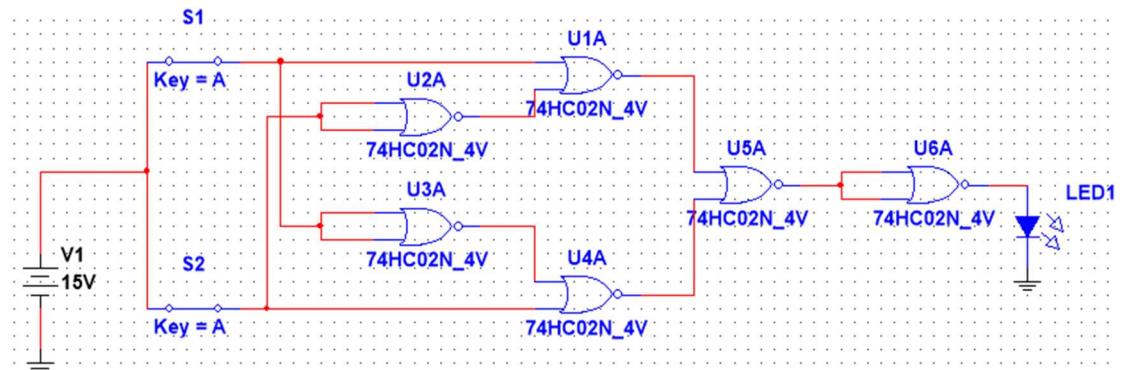


FIG – 1: Symbol of $A \oplus B$ gate

SIMULATION

 A⊕B Gate $A = 0, B = 0, Y = 0$	 A⊕B Gate $A = 0, B = 1, Y = 1$
 A⊕B Gate $A = 1, B = 0, Y = 1$	 A⊕B Gate $A = 1, B = 1, Y = 0$

$$(A \oplus B) + C$$

Truth table:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

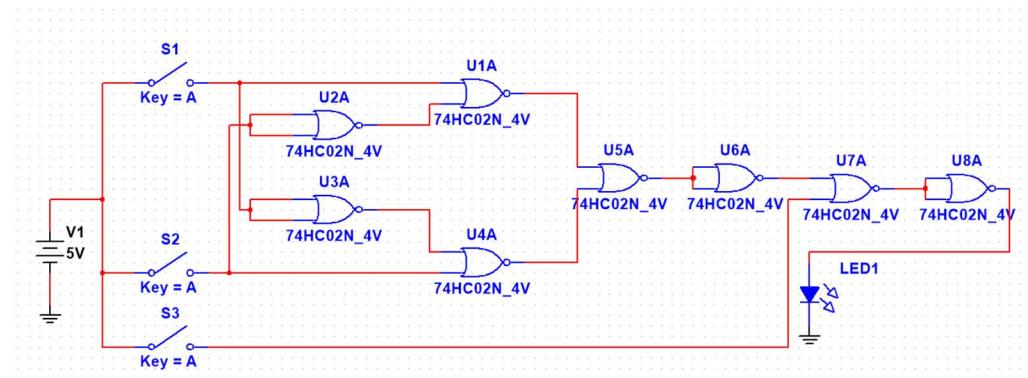
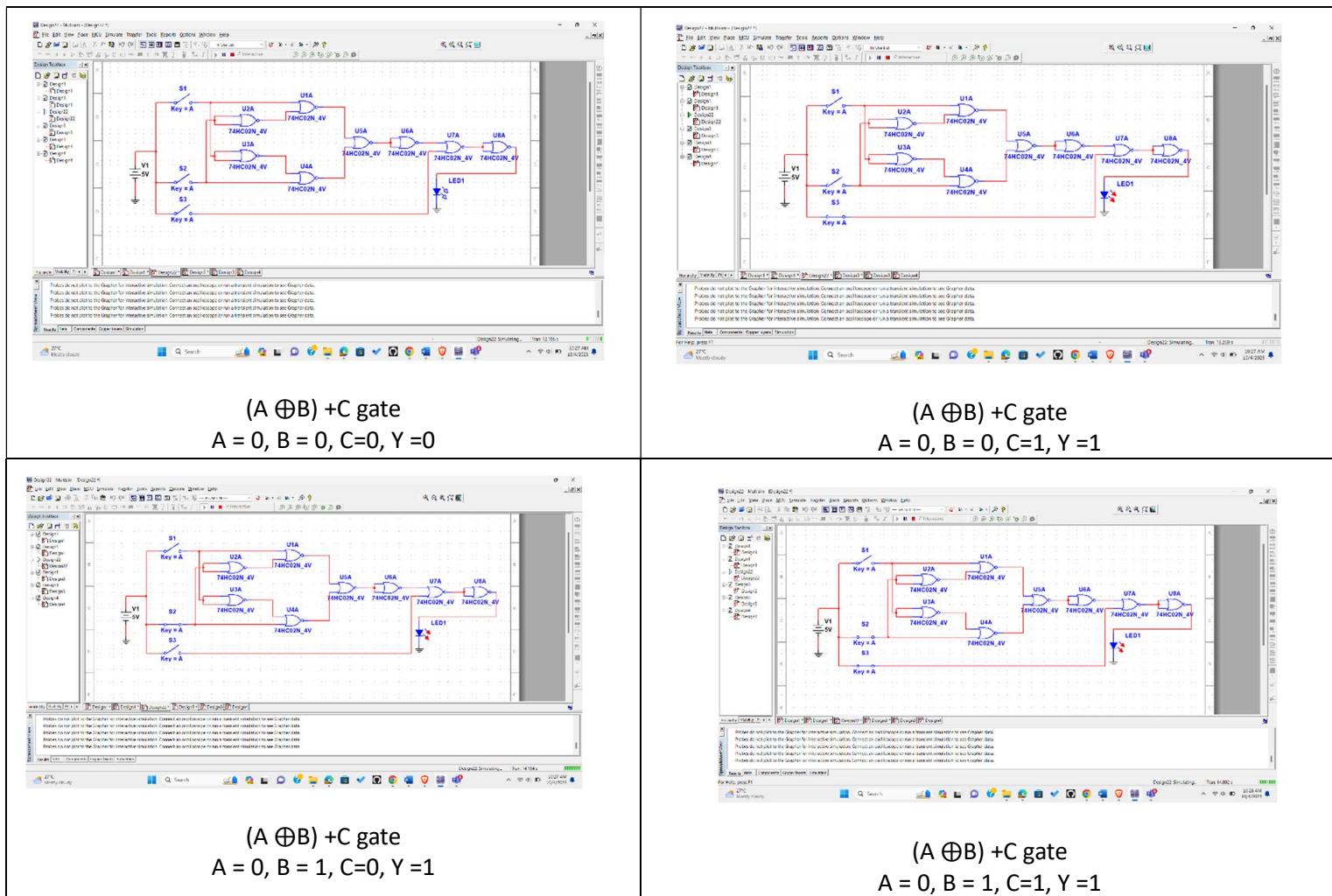
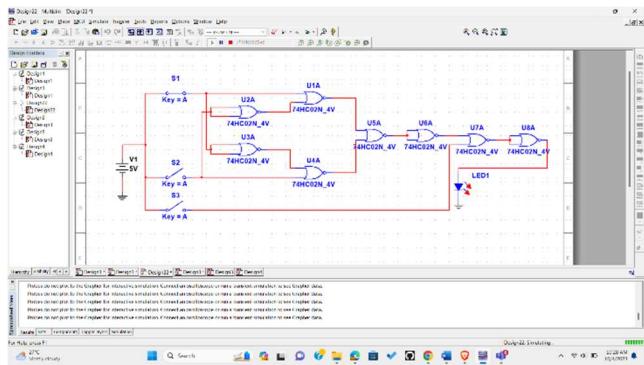


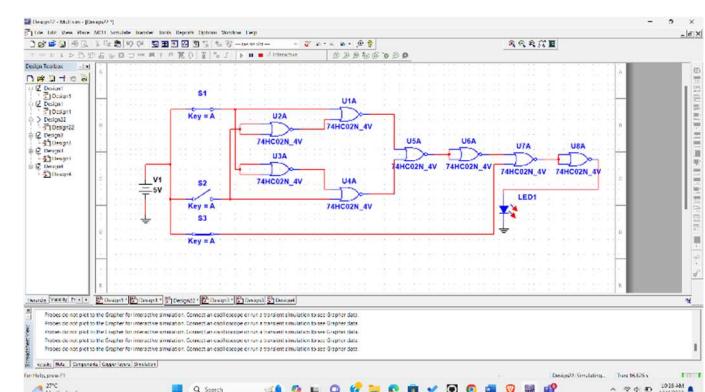
FIG – 1: Symbol of $(A \oplus B) + C$ gate

SIMULATION

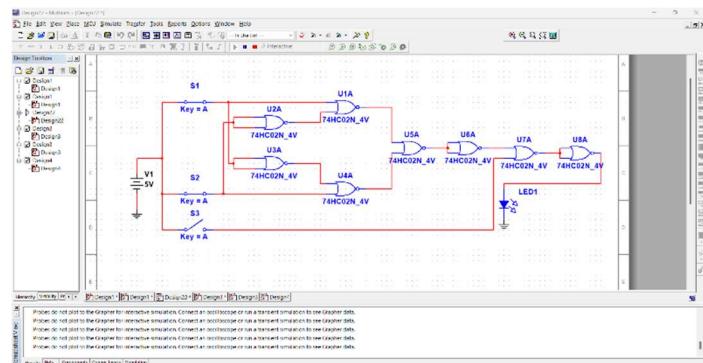




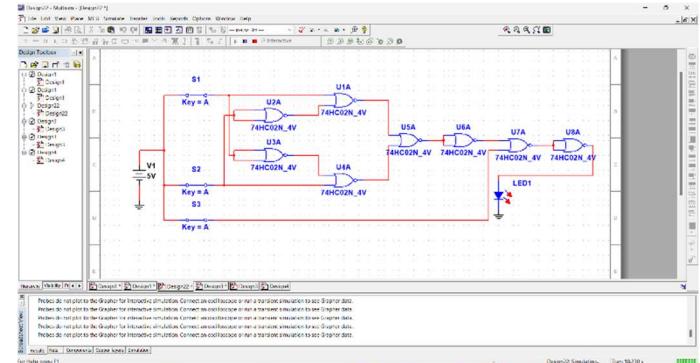
$(A \oplus B) + C$ gate
 $A = 1, B = 0, C = 0, Y = 1$



$(A \oplus B) + C$ gate
 $A = 1, B = 0, C = 1, Y = 1$



$(A \oplus B) + C$ gate
 $A = 1, B = 1, C = 0, Y = 0$



$(A \oplus B) + C$ gate
 $A = 1, B = 1, C = 1, Y = 1$

$$\overline{AB} + DC$$

Truth table:

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

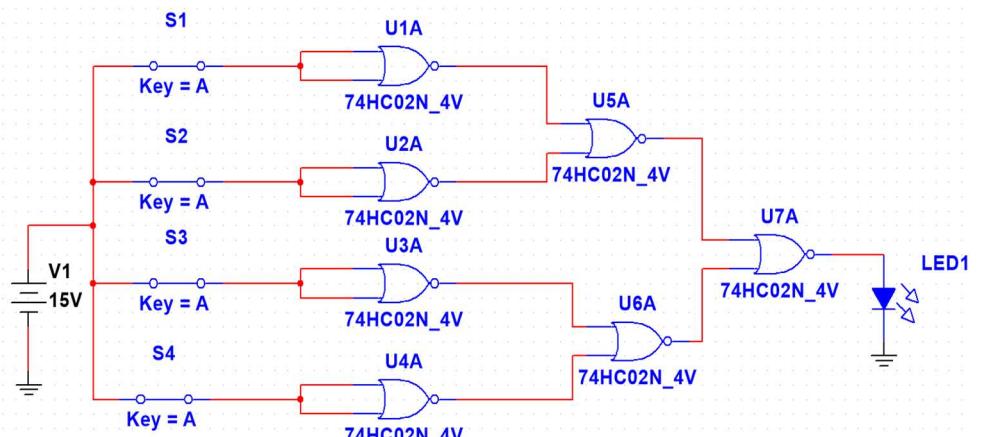
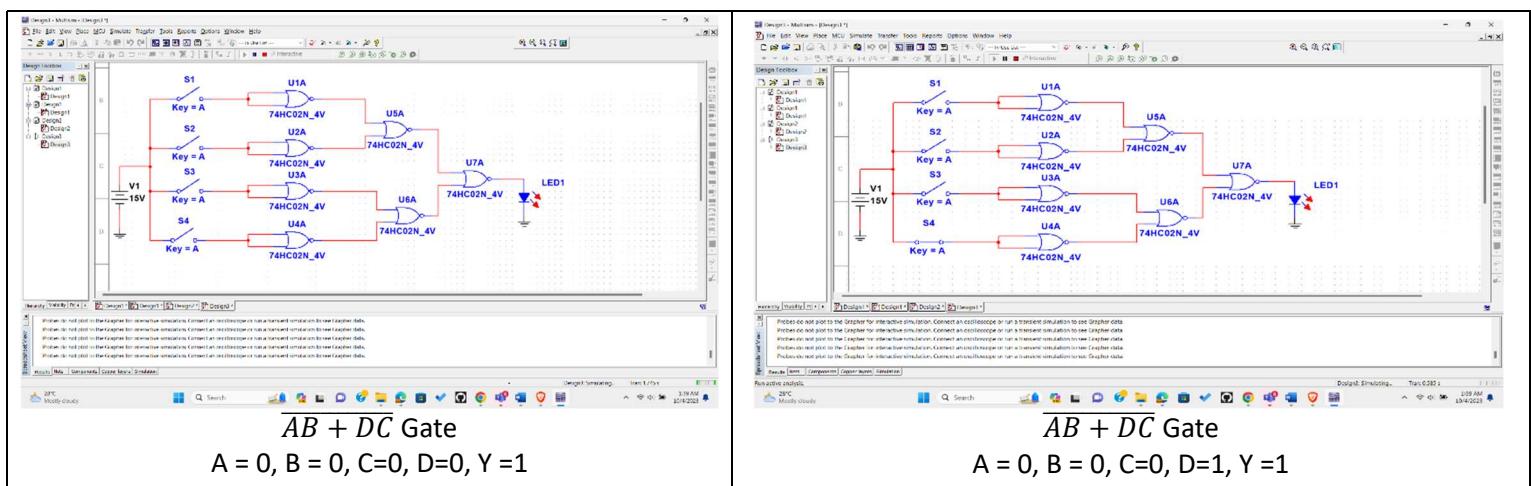
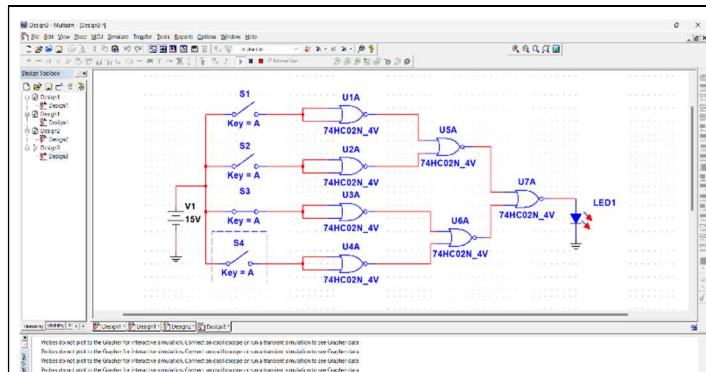


FIG – 1: Symbol of $\overline{AB} + DC$ gate

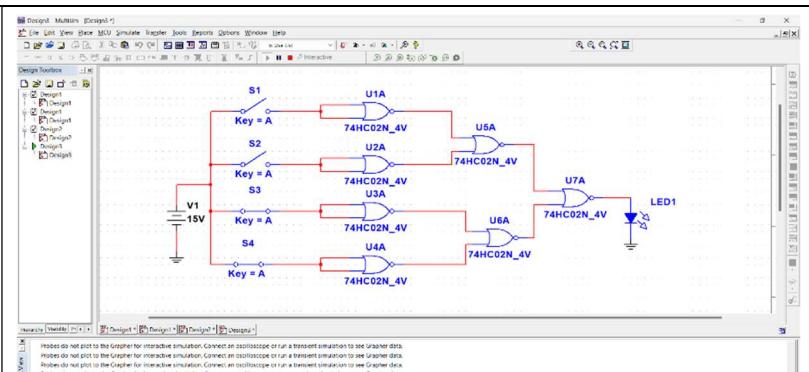
SIMULATION



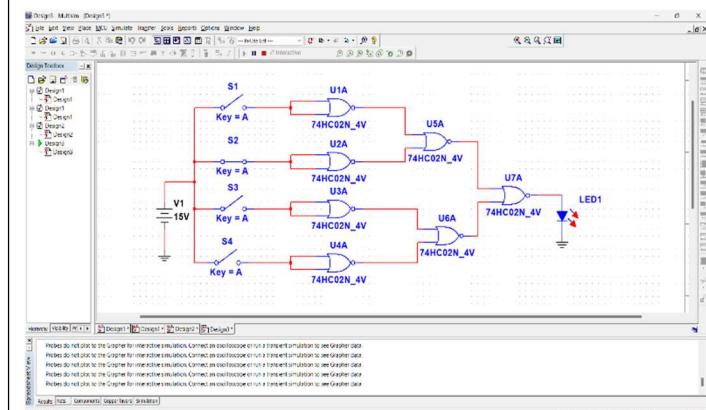
SIMULATION



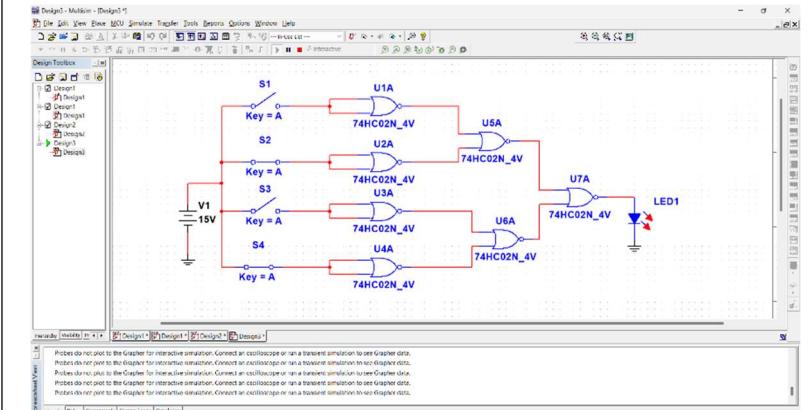
$AB + DC\bar{C}$ Gate
 $A = 0, B = 0, C=1, D=0, Y=1$



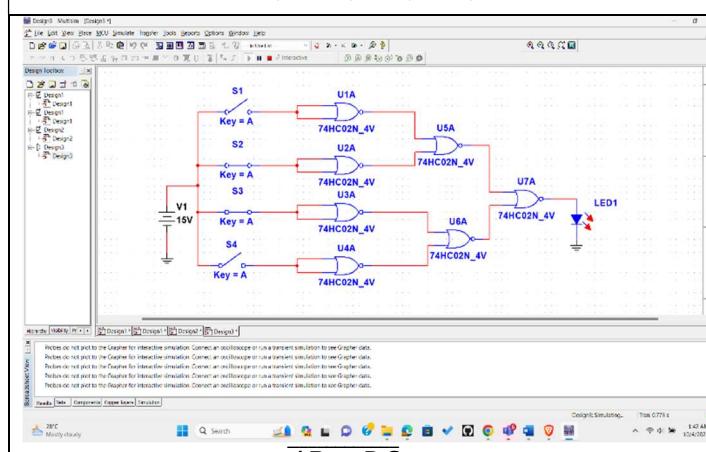
$AB + DC\bar{C}$ Gate
 $A = 0, B = 0, C=1, D=1, Y=0$



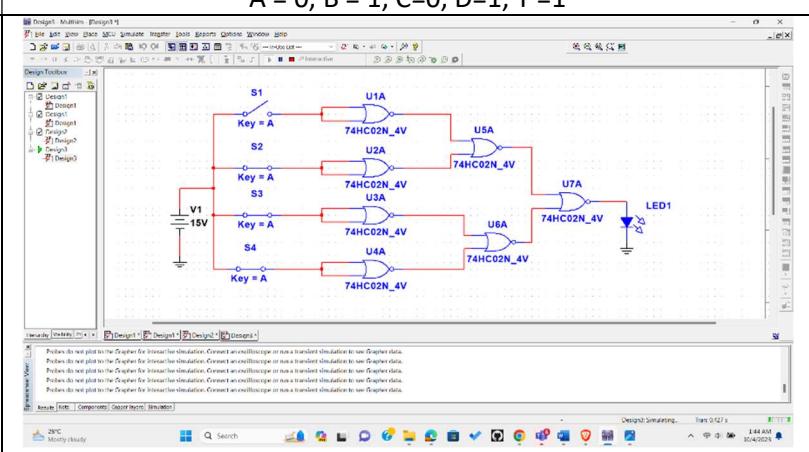
$AB + DC\bar{C}$ Gate
 $A = 0, B = 1, C=0, D=0, Y=1$



$AB + DC\bar{C}$ Gate
 $A = 0, B = 1, C=0, D=1, Y=1$

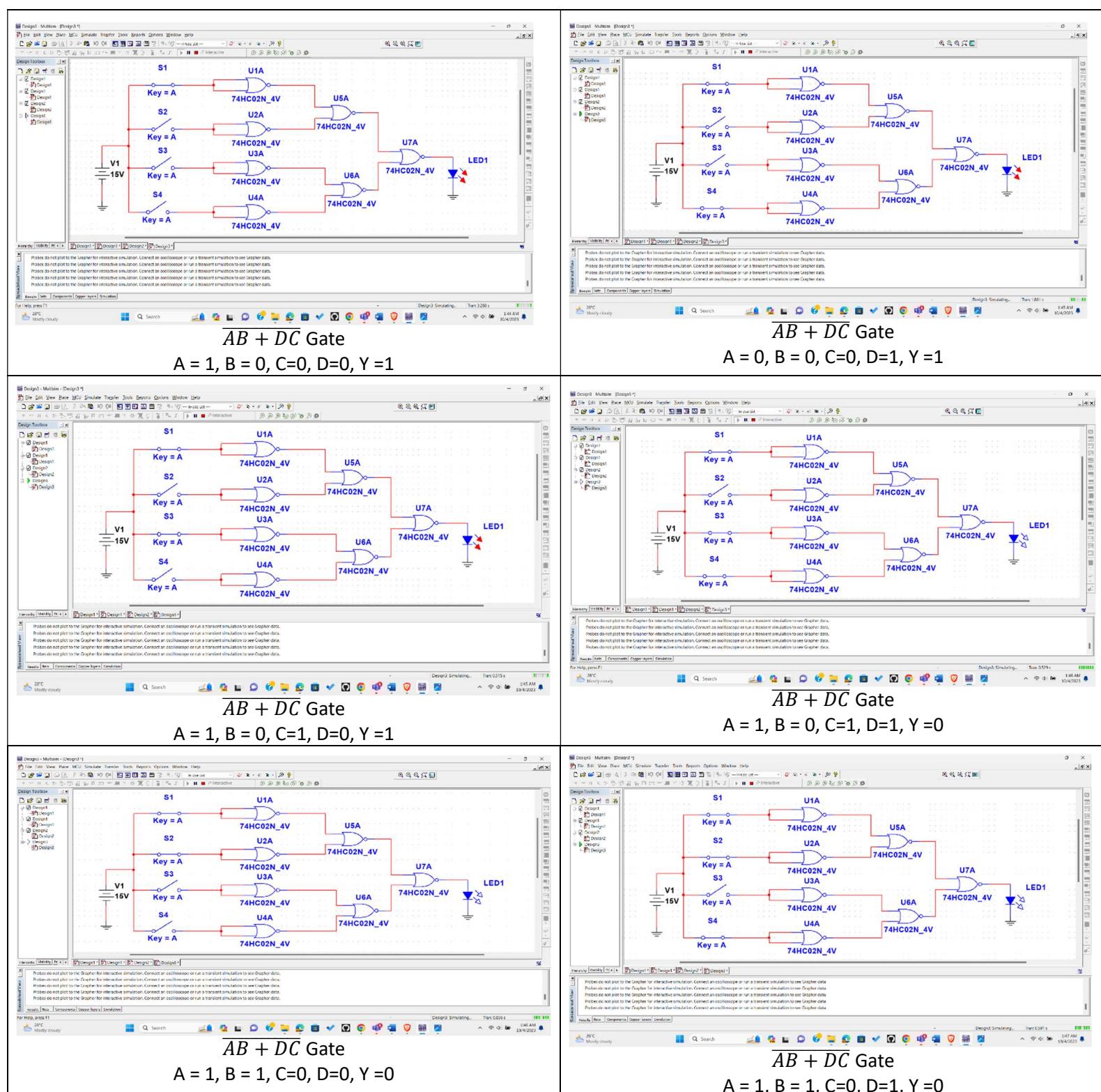


$AB + DC\bar{C}$ Gate
 $A = 0, B = 1, C=1, D=0, Y=1$

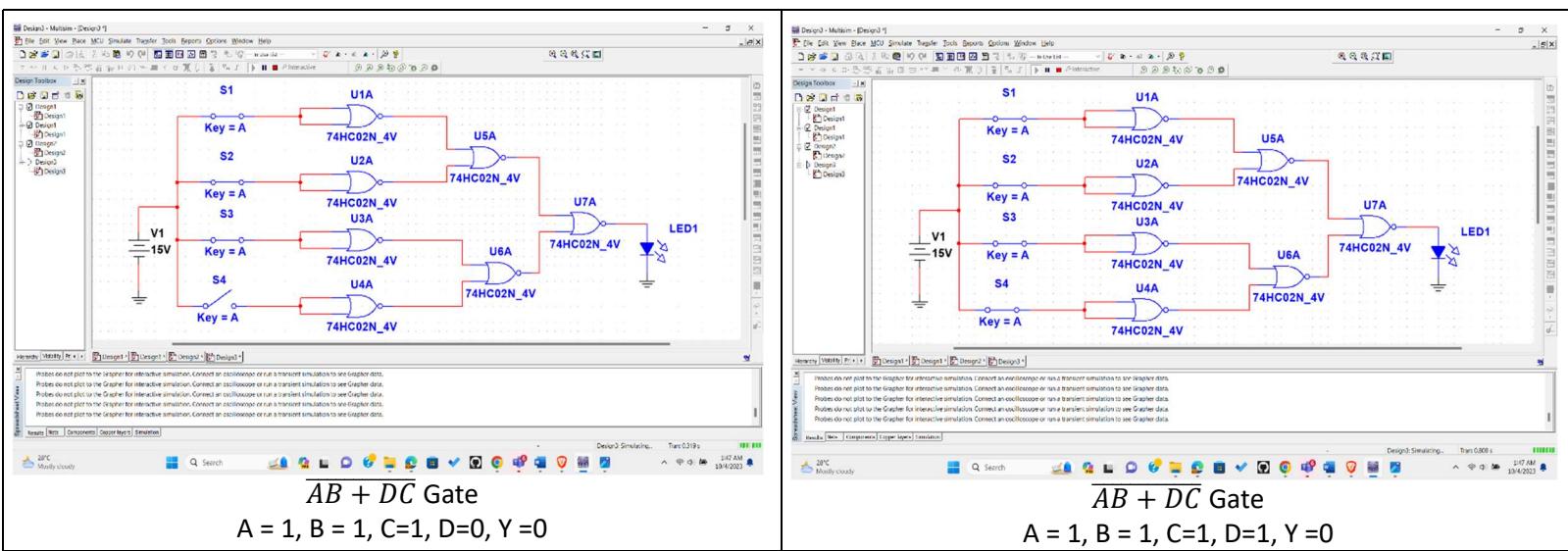


$AB + DC\bar{C}$ Gate
 $A = 0, B = 1, C=1, D=0, Y=1$

SIMULATION



SIMULATION



Questions with answers for report writing:

- 1) What do you mean by universal gate?

Answer : A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. Because using NAND and NOR gate, we can implement any Boolean expression.

NAND gate: The output of a NAND gate will be low, if all the inputs are high, otherwise output will be high.

NOR gate : The output of a NOR gate will be high, if all the inputs are low, otherwise output will be low.

- 2) What are the ICs required in this experiment?

Answer: We needed 6 different type of ICs to perform this experiment. They are,

IC-74HC08N – AND gate

IC-74HC32N – OR gate

IC-74HC04N – NOT gate

IC-74HC00N- NAND gate

IC-74HC02N – NOR gate

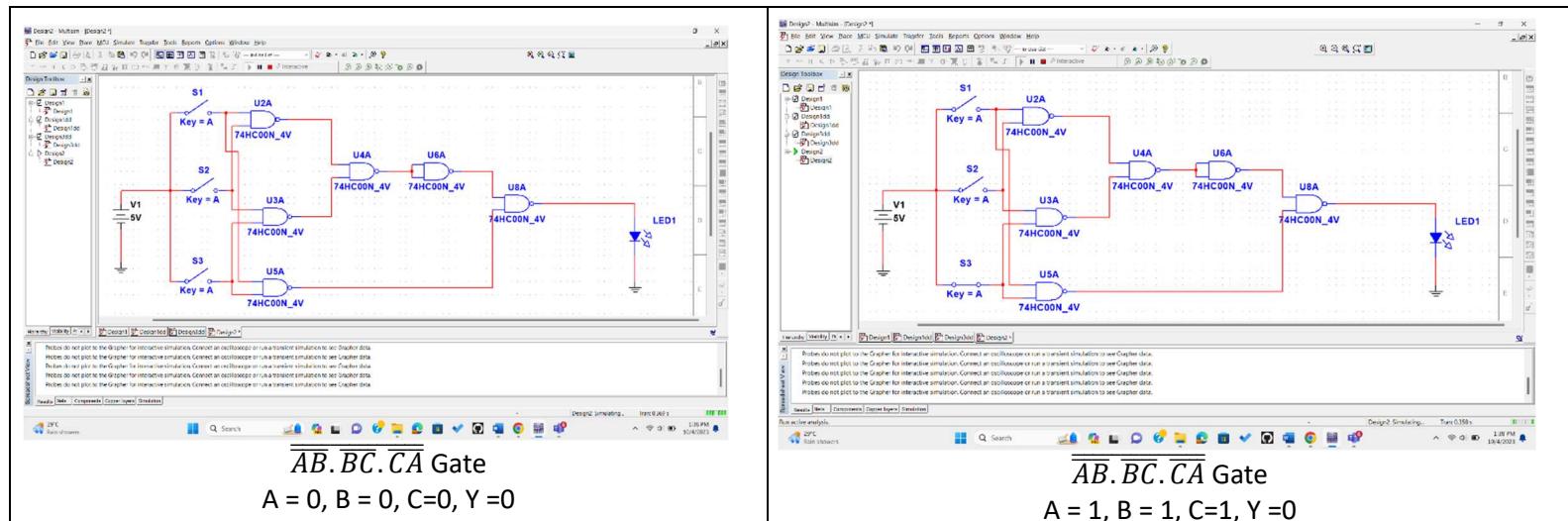
IC – 74HC86N – XOR gate

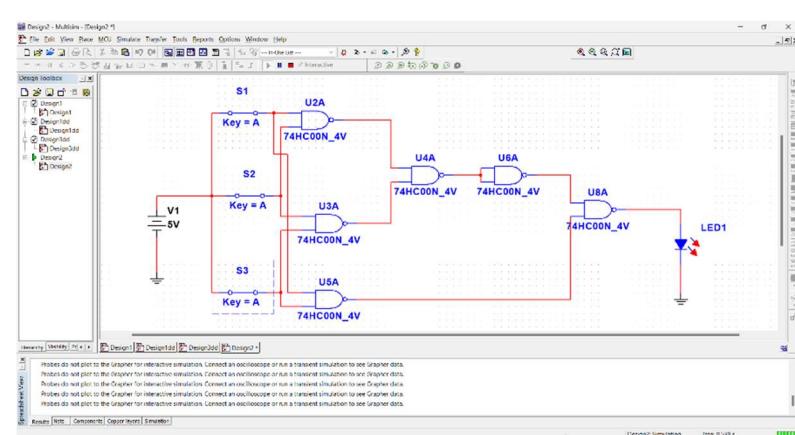
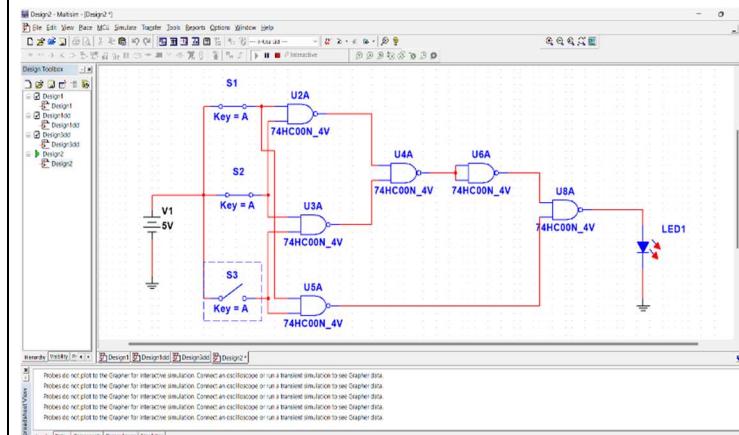
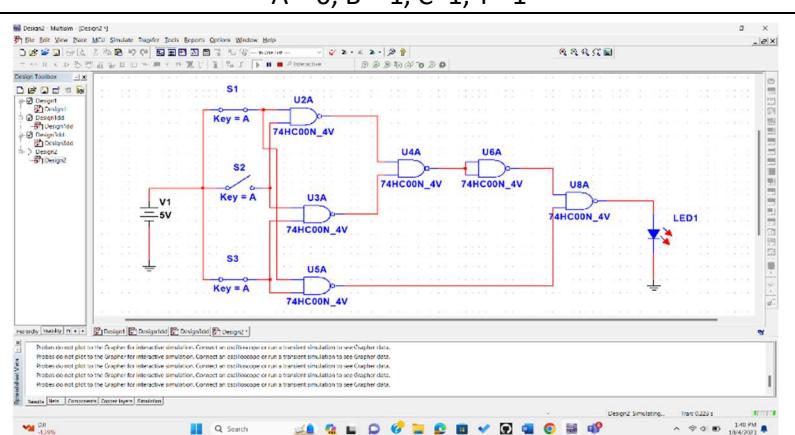
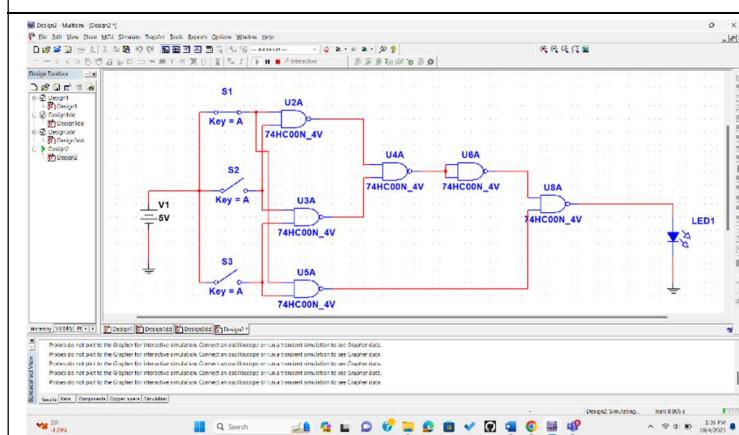
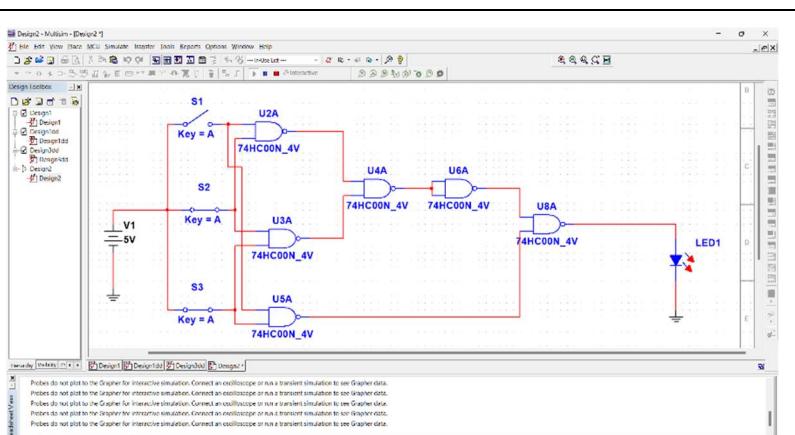
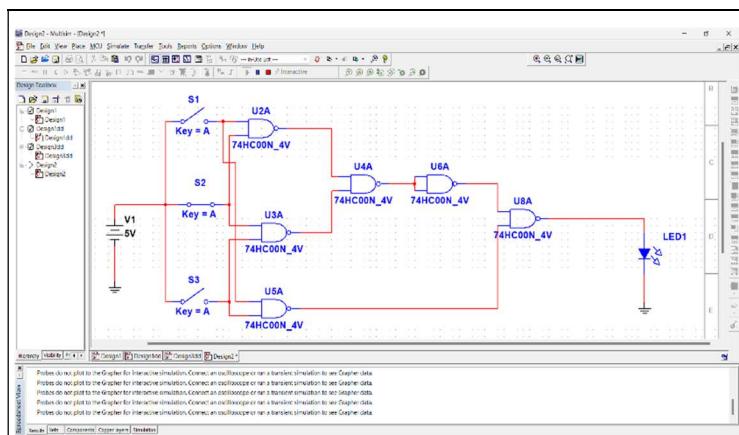
- 3) Construct a circuit of output F, where $F = AB + BC + CA$, by using NAND gates only in the PSIM Software and show the output states for each of the available conditions.

Answer:

$$\begin{aligned}
 F &= AB + BC + CA \\
 &= \overline{\overline{AB} + \overline{BC} + \overline{CA}} \\
 &= \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{CA}}
 \end{aligned}$$

SIMULATION





Discussion and Conclusion:

In this experiment, we implemented all the logic gates. Then implemented some Boolean expression using universal gates as NAND and NOR. At first part all the logic gates like AND, OR, NOT, NAND, NOR, X-OR, X-NOR were implemented and verified the truth table practically, In the second part of the experiment, some Boolean expression were converted into simplified NAND/ NOR logic, then implemented the expression through universal gates. The experiment were checked properly before conducting the experiment

Reference(s):

- 1) www.tutorialspoint.com
- 2) www.electronics-tutorials.ws
- 3) faculty.kfupm.edu.sa
- 4) “Digital Fundamentals” by Thomas L. Floyd