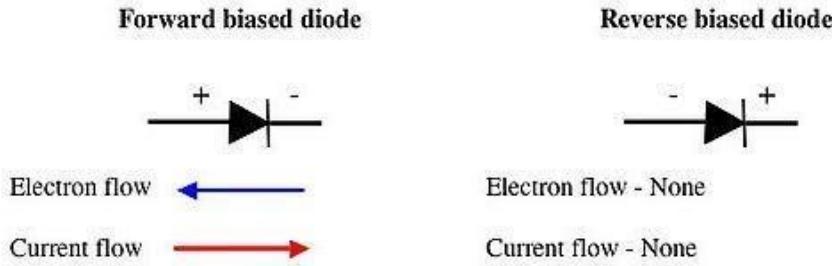


Title: Construction of different Logic Gates using various types of semiconductor devices.

Part I: Construction of Diode Logic Gates

Introduction:

A diode is a two-terminal electrical device that allows current to flow in one direction but not the other. It is like a pipe with an internal valve that allows water to flow freely in one direction but shuts down if the water tries to flow backward. The diode's two terminals are called the anode and cathode. In the diode symbol, the arrow points from the anode (flat part of triangle) toward the cathode (point of the triangle).



The device operates by allowing current to flow from anode to cathode, basically in the direction of the triangle. Recall that current is defined to flow from the more positive voltage toward the more negative voltage (electrons flow in the opposite direction). If the diode's anode is at a higher voltage than the cathode, the diode is said to be forward biased, its resistance is very low, and current flows. If the anode is at a lower voltage than the cathode, the diode is reverse-biased, its resistance is very high, and no current flows. The diode is not a perfect conductor, so there is a small voltage drop, approximately 0.7 V, across it.

In this group of experiments we will implement some logic functions using the DL circuits and discover the potential benefits and problems of using the DL logic.

Theory and Methodology:

Diode Logic OR Gate:

A Diode Logic OR gate consists of nothing more than diodes (one for each input signal) and a resistor. Here, the $10K\Omega$ resistor (R) is added to provide a ground reference for the output signal. If there are no input signals connected to the diodes, the output will be ground, or logic 0. Thus, an open input is equivalent to a logic 0 input, and will have no effect on the operation of the rest of the circuit.

It is possible to add any number of input diodes to this circuit, each with its separate input signal. However, two inputs are quite sufficient to demonstrate the operation of the circuit.

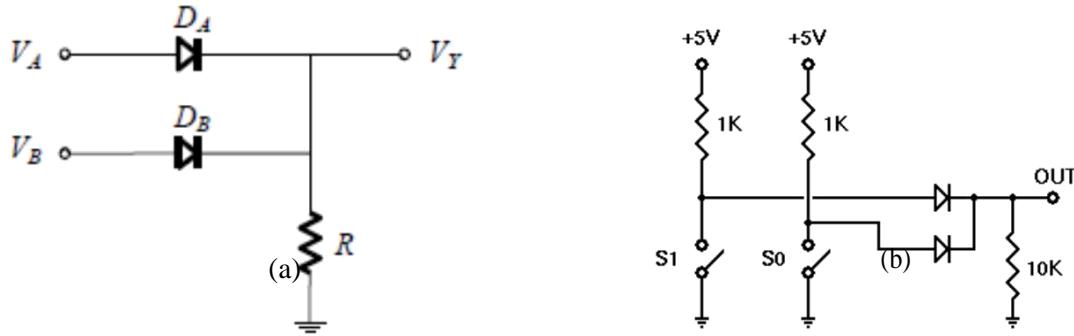


Fig.1 DL-OR Gate

Assuming the diodes are ideal, the voltage truth table as given in Table 1(a) is obtained. The corresponding logic truth table is given in Table 1(b):

V_A (volt)	V_B (volt)	V_Y (volt)		A	B	Y
0	0	0		0	0	0
0	5	5		0	1	0
5	0	5		1	0	0
5	5	5		1	1	1

Table 1

Diode Logic AND Gate:

A Diode Logic AND gate consists of diodes (one for each input signal) and a resistor. As with the DL OR gate, the $10K\Omega$ resistor (R) provides a reference connection. Unlike the OR gate, however, this is a reference to +5 volts, rather than to ground. If there are no input signals connected to the diodes, the output will be +5 volts, or logic 1. Thus, an open input will not affect the rest of the circuit, which will continue to operate normally.

As with DL-OR gates, it is possible to add any number of input diodes to this circuit, each with its separate input signal. However, two inputs are quite sufficient to demonstrate the operation of the circuit.

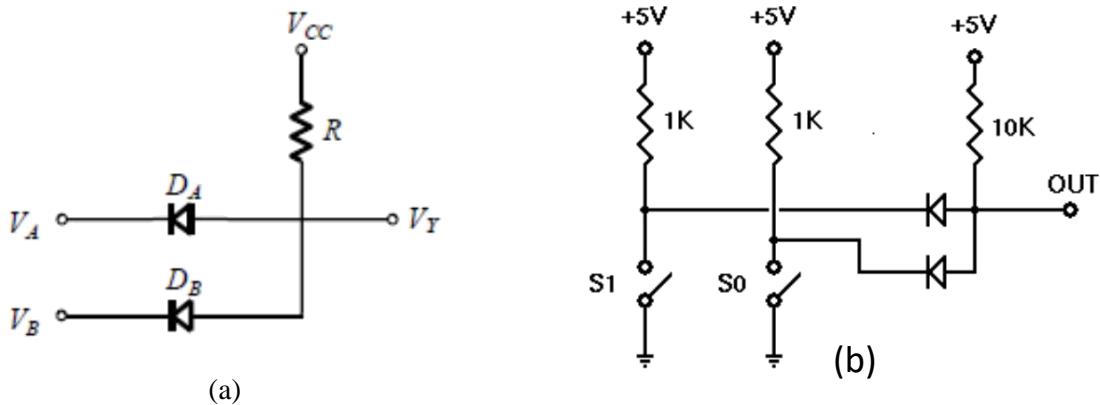


Fig.2 DL-AND Gate

Assuming the diodes are ideal, the voltage truth table of the above AND gate is as given in Table 2(a). The corresponding logic truth table is in Table 2(b).

V_A (volt)	V_B (volt)	V_Y (volt)	A	B	Y
0	0	0	0	0	0
0	5	0	0	1	0
5	0	0	1	0	0
5	5	5	1	1	1

Table 2

Two-Input DL AND -OR Gate:

After looking at both the Diode Logic (DL) OR gate and AND gate and evaluating whether their operations were within acceptable parameters, the AND and OR gates will be cascaded. The OR gate will be used to combine the outputs of two AND gates and how well this combination works will be observed.

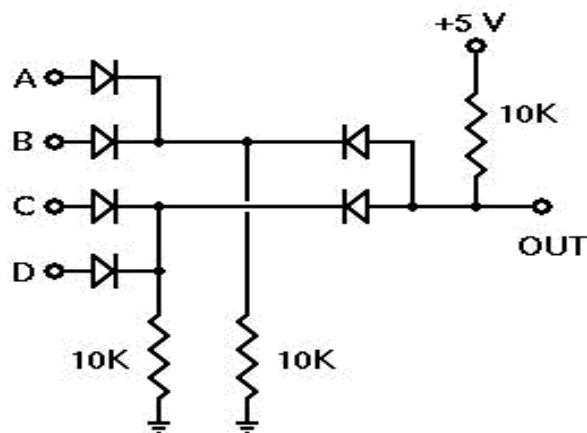


Fig.3 DL-AND-OR Gate

Diode polarity:

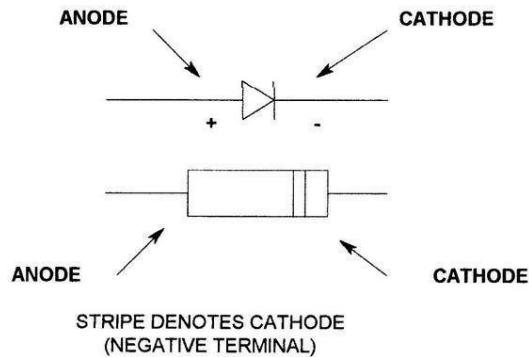


Fig.4 Diode polarity

Pre-Lab Homework:

1. Explain how a p-n junction or diode works? When does it conduct?
2. What is a wired logic?
3. Explain the operation of depletion region for different biasing conditions.

Students must install PSpice/LTSpice/ Psim software and MUST present the simulation results using transistors to the instructor before the start of the experiment.

Apparatus:

- (1) 10K ohm resistor (brown-black-orange).
- (2) 1N914/1N4002 diodes or equivalent.
- (3) Connecting wires.
- (4) Trainer Board

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 (within V_{DD}) to turn on the transistors and/or chip, otherwise it may get damaged.

Experimental Procedure:

1. Construct the DL-OR gate on your breadboard as shown in Fig. 1. Then draw a Truth Table similar to the one provided and fill in your experimental results.
2. Construct the DL-AND gate on your breadboard as shown in Fig. 2. Then draw a Truth Table similar to the one provided and fill in your experimental results.
3. Construct the DL-AND-OR gate on your breadboard as shown in Fig. 3. Before beginning the experiment, calculate your expected results for all the different input combinations and put them

in a Truth Table similar to the one provided. Then draw a second Truth Table and fill it with your experimental output values.

Results and Discussion:

Students will summarize the experiment and discuss it as a whole. Interpret the data/findings and determine the extent to which the experiment was successful in complying with the goal that was initially set. Discuss any mistake you might have made while conducting the investigation and describe ways the study could have been improved.

Report:

1. For, each of the above set-ups, describe in words what the data means. Did your results match the expected ideal outputs? If not, explain why?
2. Why are diode logic gates not suitable for cascading operation?

Part 2: Construction of Bipolar Transistor Logic Gate Introduction:

A bipolar transistor is a three-terminal semiconductor device. Under the control of one of the terminals, called the base, current can flow selectively from the collector terminal to the emitter terminal.



Fig 5: Bipolar junction transistor circuit symbols

In this experiment we examine how to build logic gates from bipolar transistors using the RTL, DTL and TTL design.

Theory and Methodology:

Resistor-Transistor Logic (RTL):

Resistor-Transistor Logic (RTL) is a large step beyond Diode Logic (DL). Basically, RTL replaces the diode switch with a transistor switch. If a +5v signal (logic 1) is applied to the base of the transistor (through an appropriate resistor to limit base-emitter forward voltage and current), the transistor turns fully on and grounds the output signal. If the input is grounded (logic 0), the transistor is off and the output signal is allowed to rise to +5 volts. In this way, the transistor not only inverts the logic sense of the signal, but it also ensures that the output voltage will always be a valid logic level under all circumstances. Because of this, RTL circuits can be cascaded indefinitely, where DL circuits cannot be cascaded reliably at all.

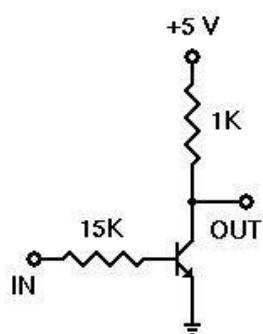


Fig.6: RTL Inverter

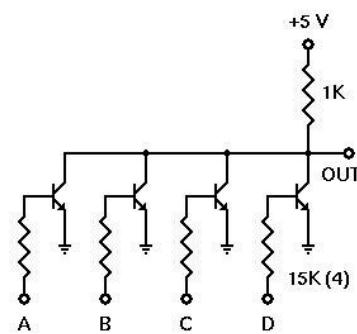


Fig .7: 4-input RTL Inverter

Diode-Transistor Logic:

Diode-Transistor Logic (DTL) is a class of digital circuits built from bipolar junction transistors (BJT), diodes and resistors; it is the direct ancestor of transistor-transistor logic (TTL).

DTL offers better noise margins and greater fan-outs than RTL, but suffers from low speed (especially in comparison to TTL).

RTL allows the construction of NOR gates easily, but NAND gates are relatively more difficult to get from RTL. DTL, however, allows the construction of simple NAND gates from a single transistor, with the help of several diodes and resistors.

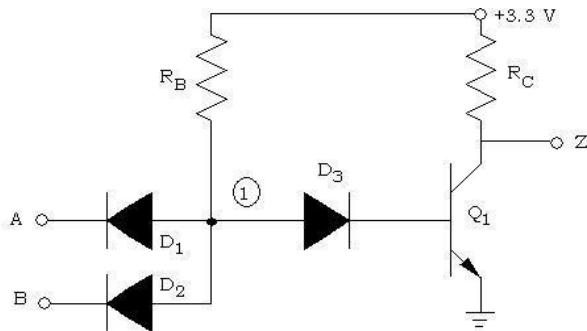


Fig 8: 2-input DTL NAND Gate

Transistor-Transistor Logic:

We can think of a bipolar transistor as two diodes placed very close together, with the point between the diodes being the transistor base. Thus, we can use transistors in place of diodes to obtain logic gates that can be implemented with transistors and resistors only; this is called transistor-transistor logic (TTL).

One problem that DTL doesn't solve is its low speed, especially when the transistor is being turned off. Turning off a saturated transistor in a DTL gate requires it to first pass through the active region before going into cut-off. Cut-off, however, will not be reached until the stored charge in its base has been removed. The dissipation of the base charge takes time if there is no available path from the base to ground. This is why some DTL circuits have a base resistor that's tied to ground, but even this requires some trade-offs. Another problem with turning off the DTL output transistor is the fact that the effective capacitance of the output needs to charge up through R_C before the output voltage rises to the final logic '1' level, which also consumes a relatively large amount of time. TTL, however, solves the speed problem of DTL elegantly.

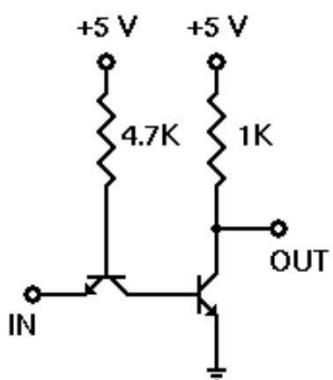


Fig 9: TTL Inverter

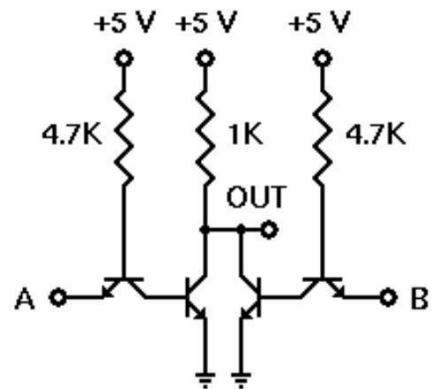
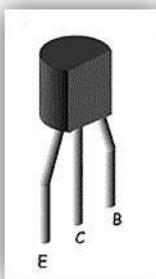


Fig 10: 2-input TTL NOR gate

BJT pin configuration:



Apparatus:

1. 2N4124 NPN silicon transistor (or equivalent).
2. Resistors $15\text{K}\Omega$, $1\text{K}\Omega$, $4.7\text{ K}\Omega$
3. Connecting wires.
4. Trainer Board

Precautions:

Instructor checked all our connections after we are done setting up the circuit and made sure that we apply only enough voltage (within V_{DD}) to turn on the transistors and/or chip, otherwise it may get damaged.

Experimental Procedure:

- (1) Set up the circuit for RTL inverter as shown in Fig.7.
- (2) For each input combination, find the output and place them in a Truth Table. The Truth Table should have two sets of outputs- one ideal and one experimental.
- (3) Repeat steps 1 and 2 for each circuit set-up from Fig.8 to Fig. 10.

Simulation & Measurements:

A	B	C	D	A+B	C+D	(A+B).(C+D)
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	0	0
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

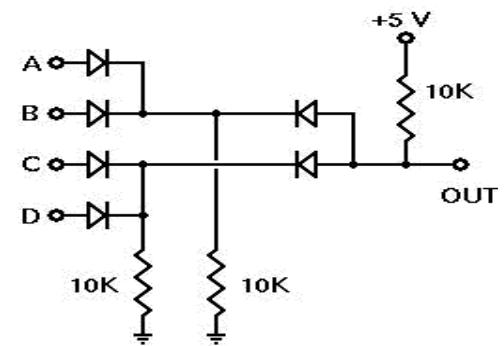
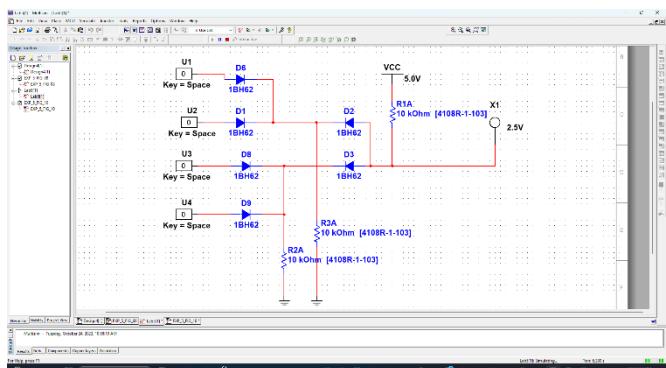
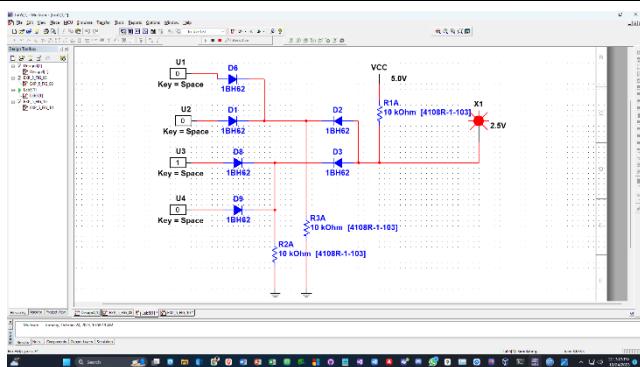
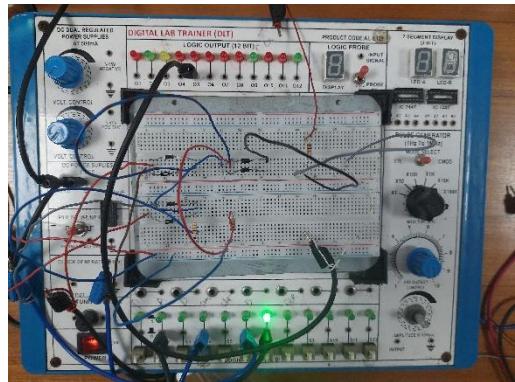


Figure-3: DL-AND-OR Gate

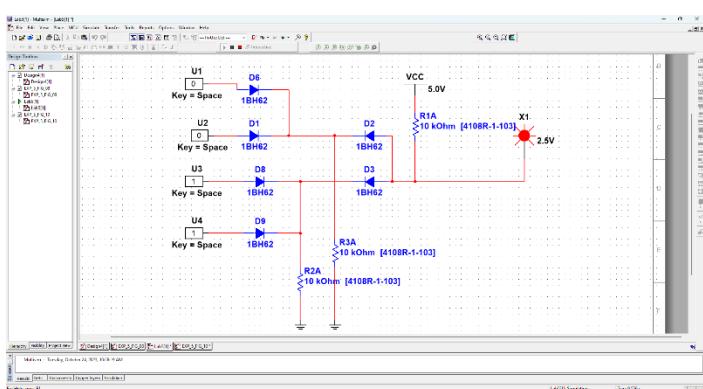
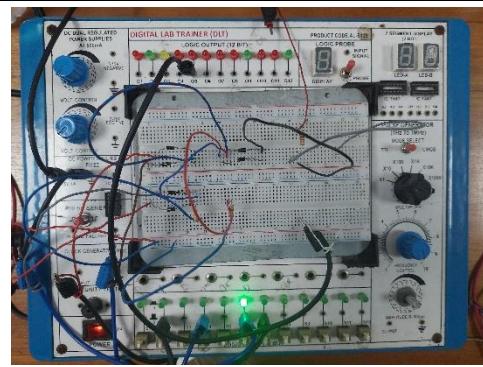
SIMULATION	HARDWARE
<p>A=0, B=0, C=0, D=0 $(A+B)(C+D)=0$</p>	



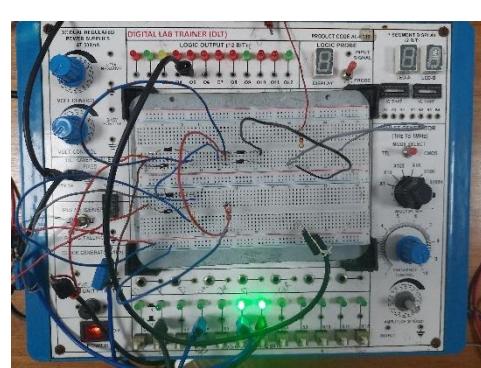
$$A=0, B=0, C=0, D=1 \\ (A+B)(C+D)=0$$

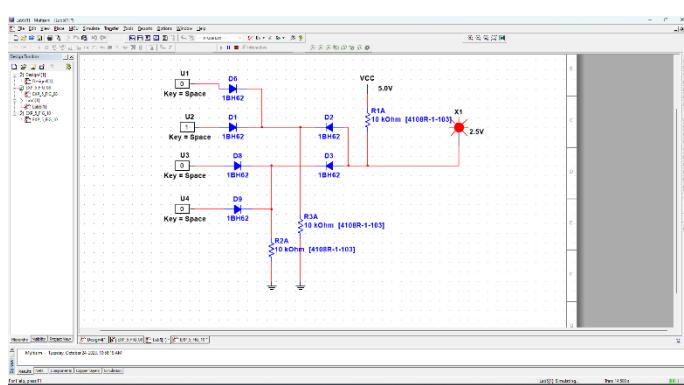


$$A=0, B=0, C=1, D=0 \\ (A+B)(C+D)=0$$

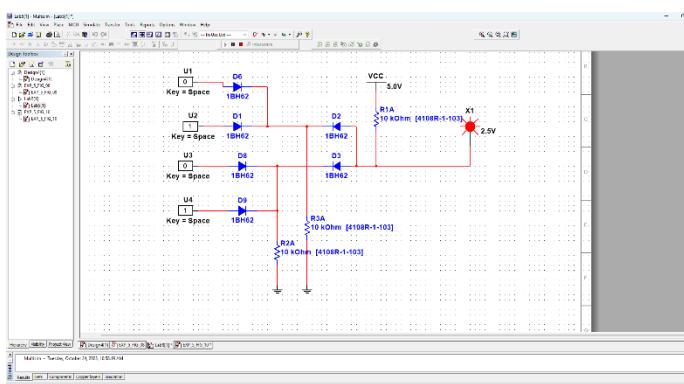
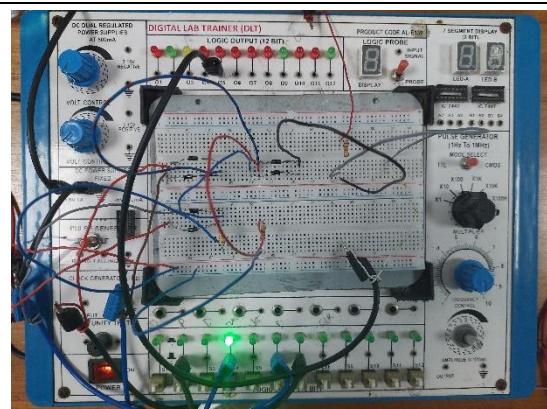


$$A=0, B=0, C=1, D=1 \\ (A+B)(C+D)=0$$

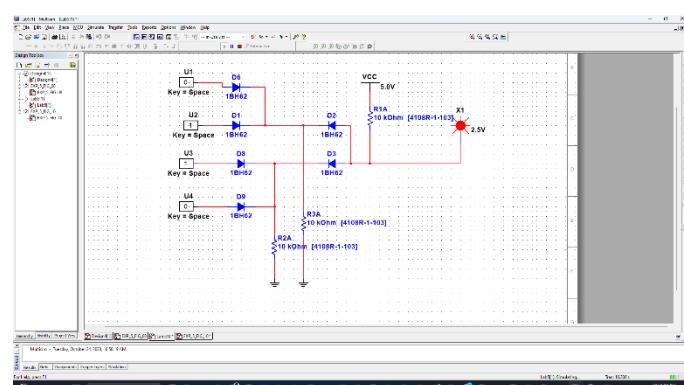
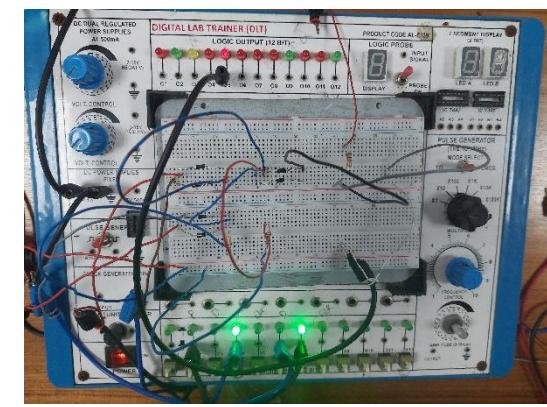




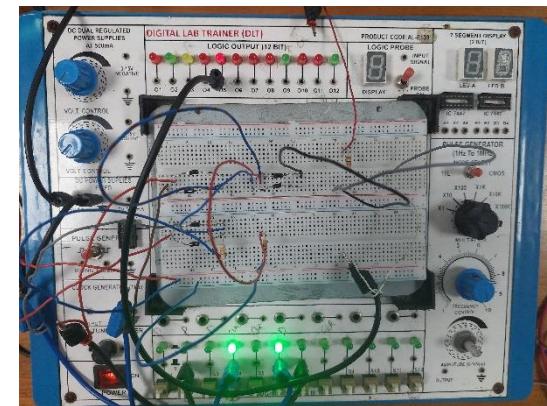
$$A=0, B=1, C=0, D=0 \\ (A+B)(C+D)=0$$

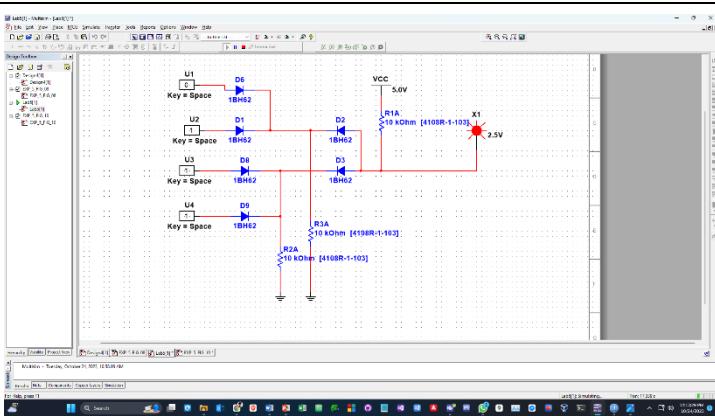


$$A=0, B=1, C=0, D=1 \\ (A+B)(C+D)=1$$



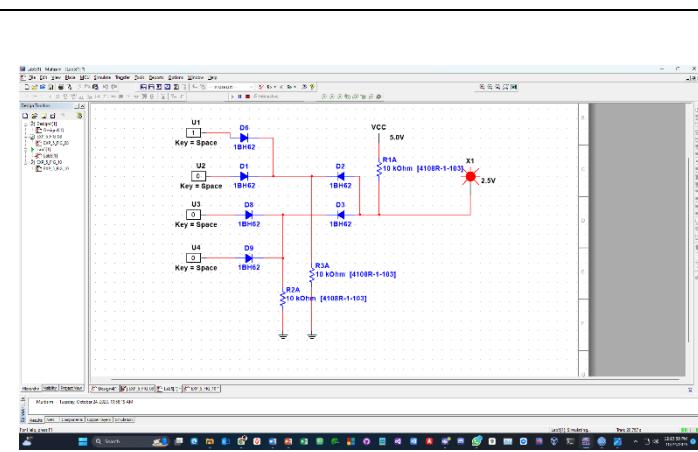
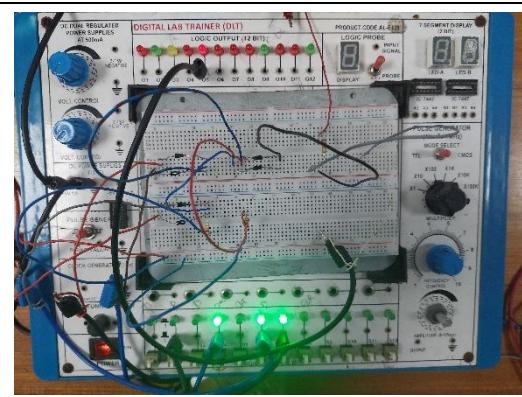
$$A=0, B=1, C=1, D=0 \\ (A+B)(C+D)=1$$





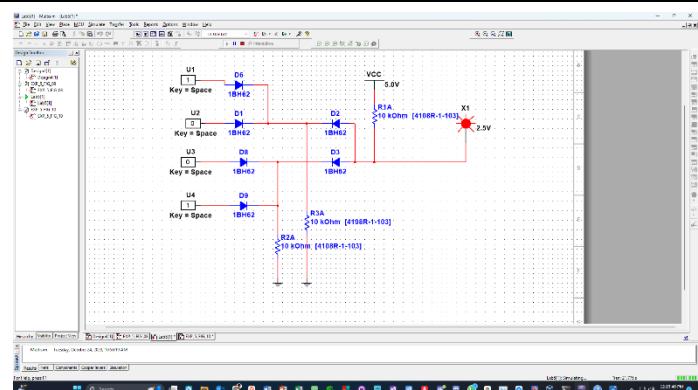
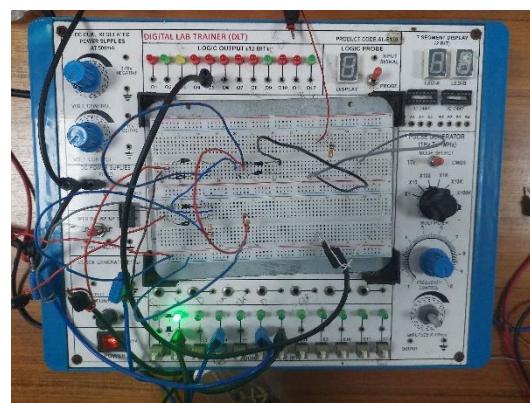
$$A=0, B=1, C=1, D=1$$

$$(A+B)(C+D)=1$$



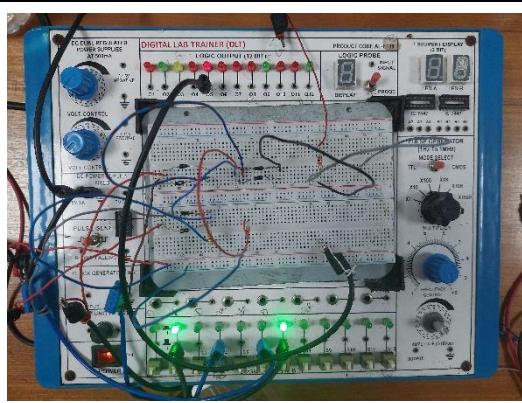
$$A=1, B=0, C=0, D=0$$

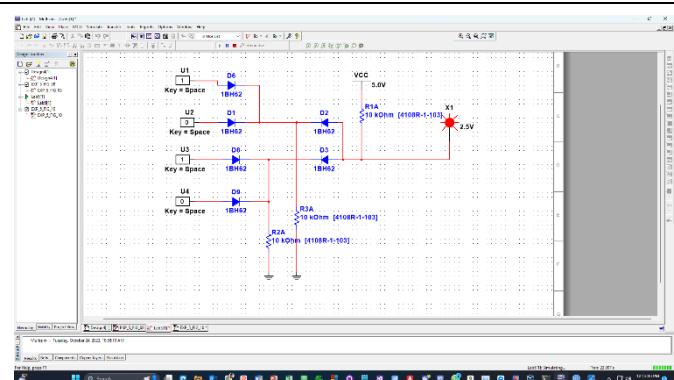
$$(A+B)(C+D)=0$$



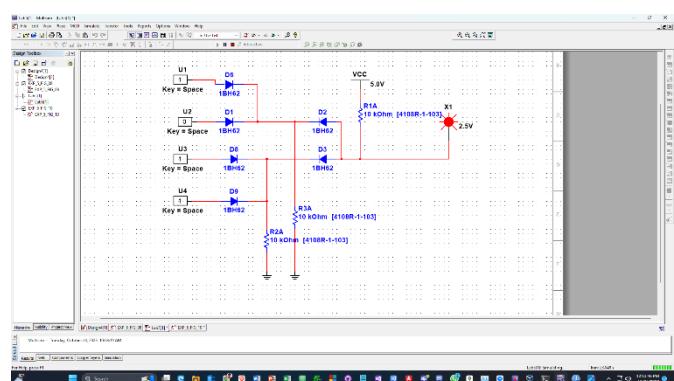
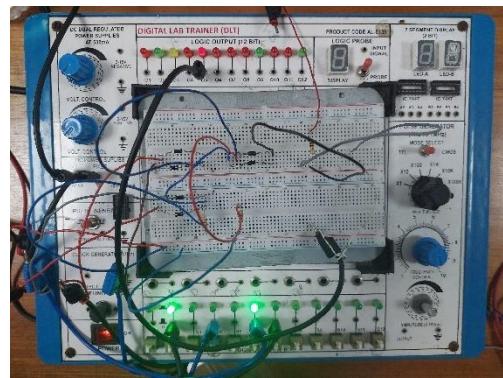
$$A=1, B=0, C=0, D=1$$

$$(A+B)(C+D)=1$$

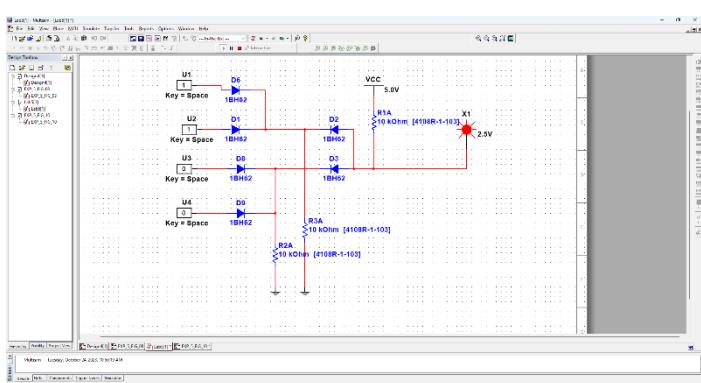
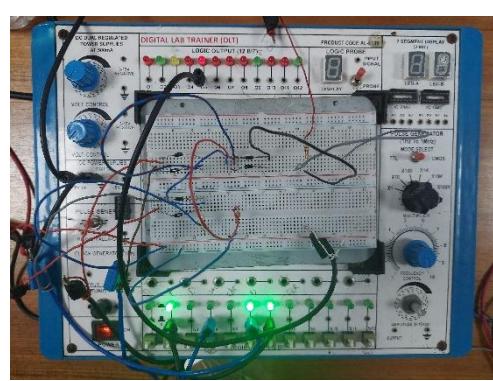




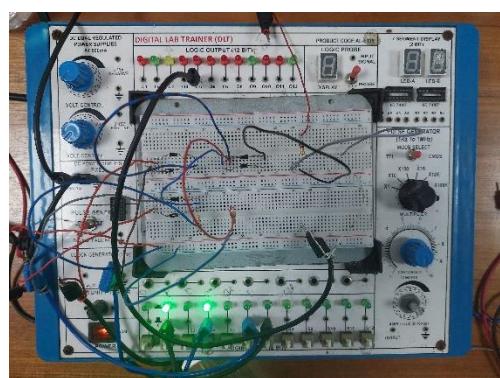
A=1, B=0, C=1, D=0
 $(A+B)(C+D)=1$

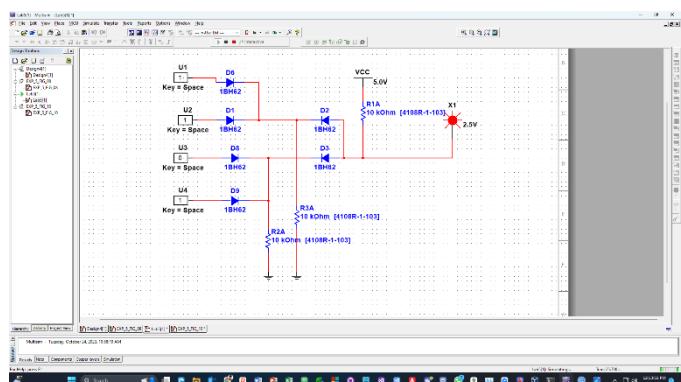


A=1, B=0, C=1, D=1
 $(A+B)(C+D)=1$

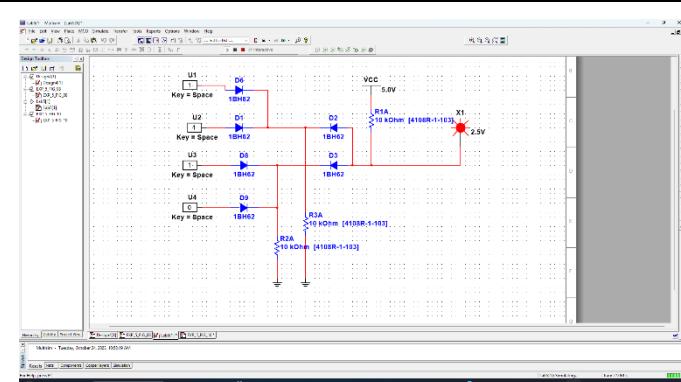
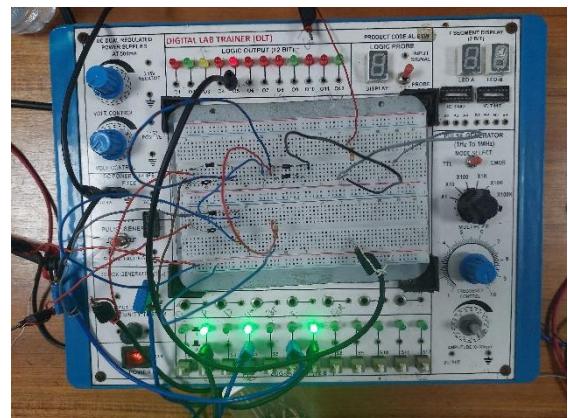


A=1, B=1, C=0, D=0
 $(A+B)(C+D)=0$

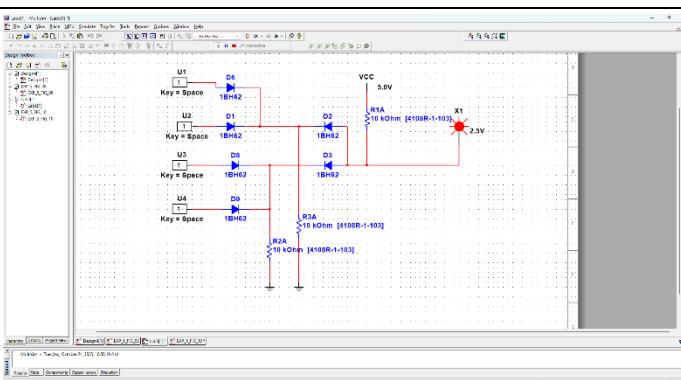
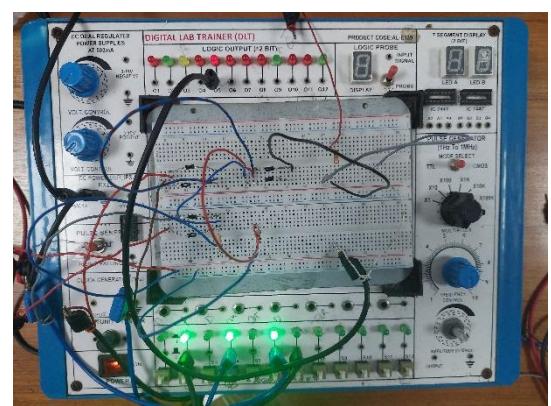




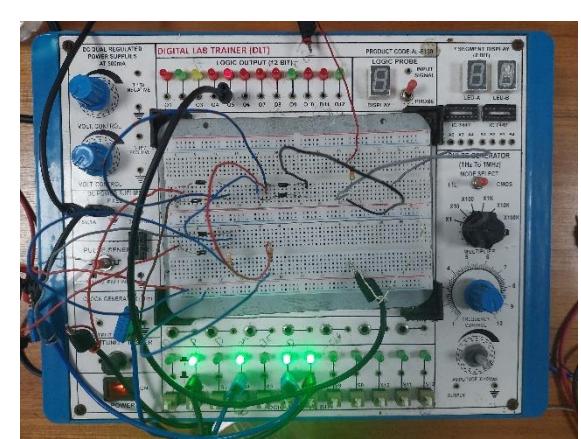
$A=1, B=1, C=0, D=1$
 $(A+B)(C+D)=1$



$A=1, B=1, C=1, D=0$
 $(A+B)(C+D)=1$



$A=1, B=1, C=1, D=1$
 $(A+B)(C+D)=1$



Truth Table:

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

Logic Diagram:

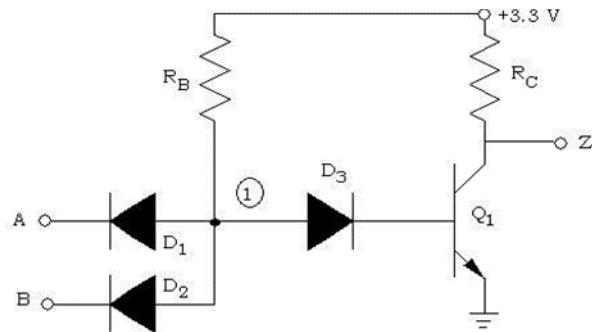
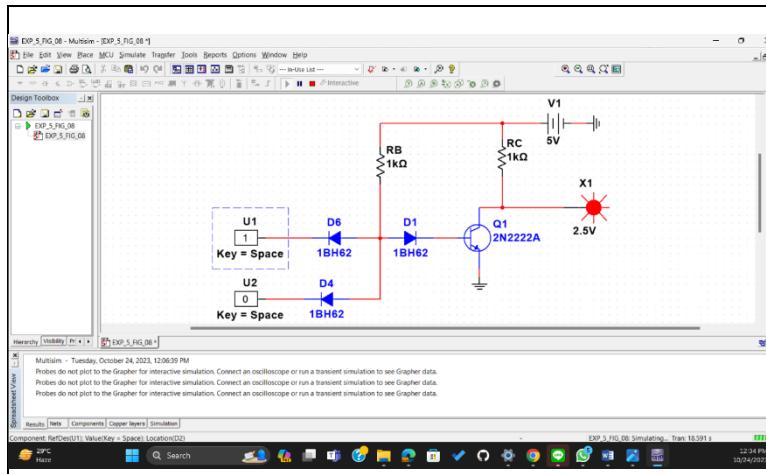
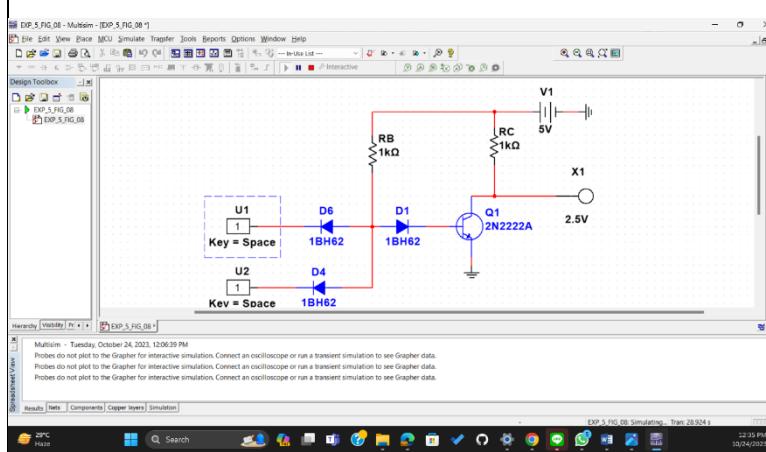
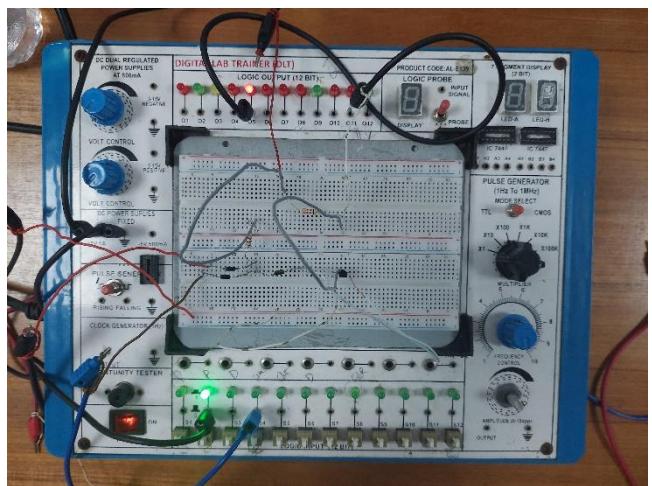


Figure-8 : 2 input DTL NAND Gate

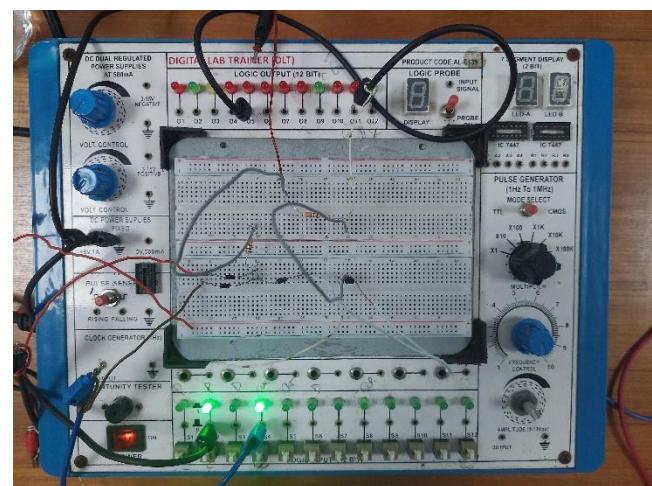
SIMULATION	HARDWARE
<p>A=0, B=0, Z=1</p>	
<p>A=0, B=1, Z=1</p>	



A=1, B=0, Z=1



A=1, B=1, Z=0

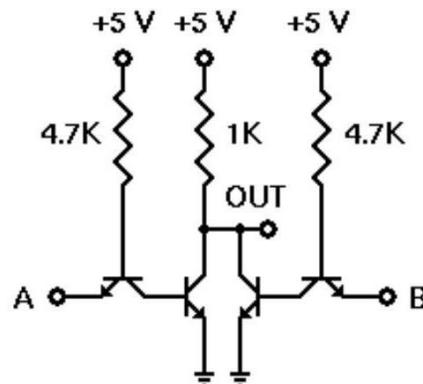


For figure-10: 2 input TTL NOR gate

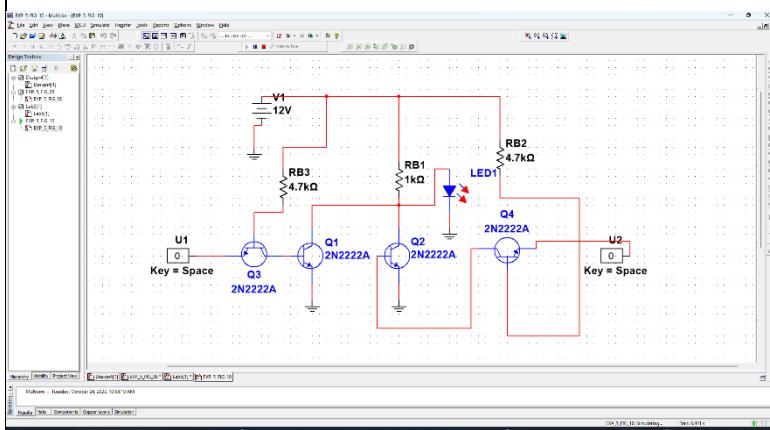
Truth Table:

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

Logic Diagram:

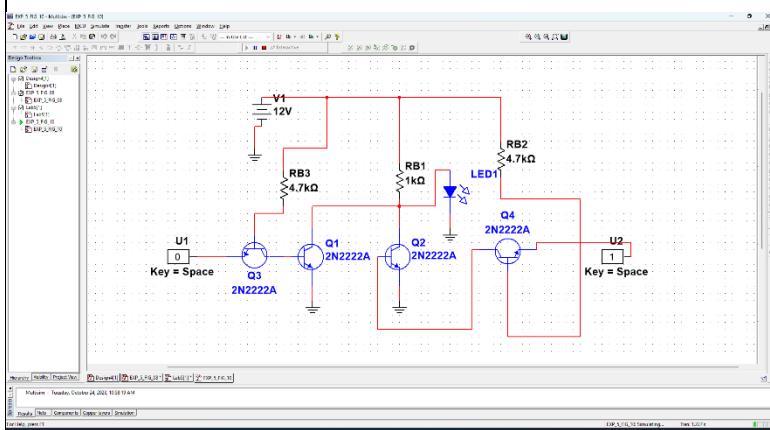
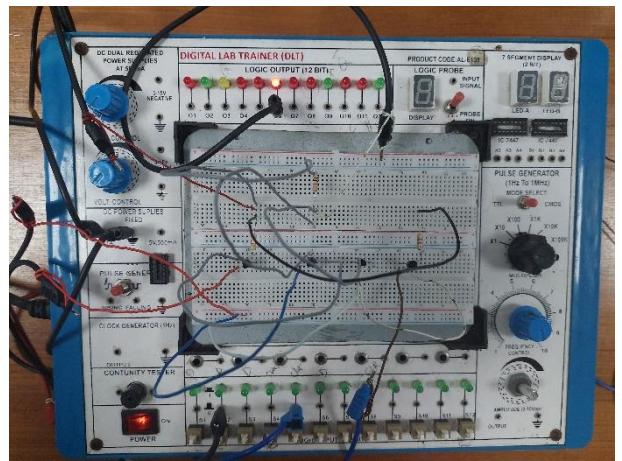


SIMULATION

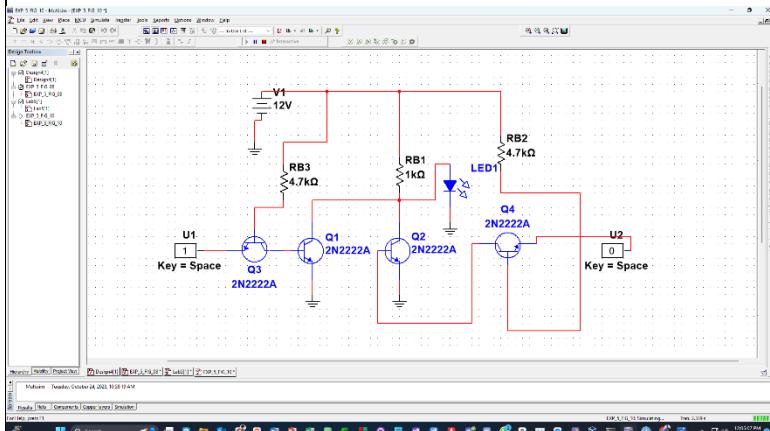
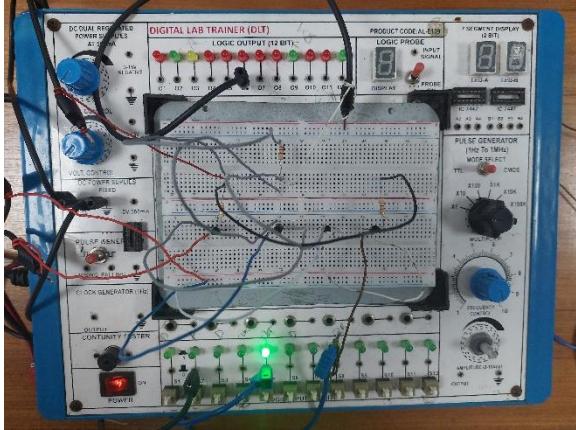


A=0, B=0, Z=0

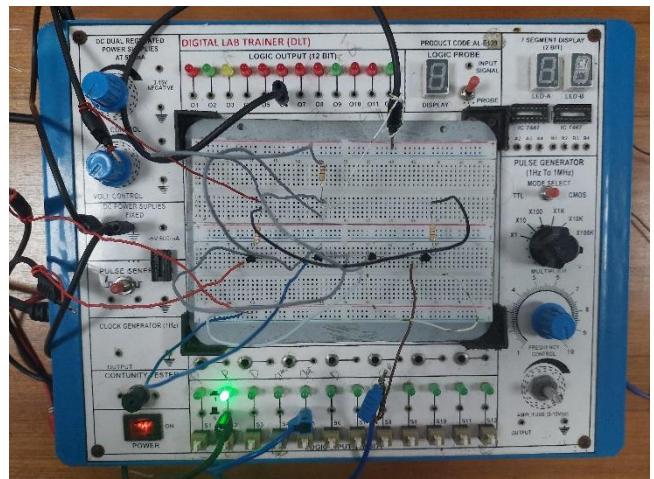
HARDWARE

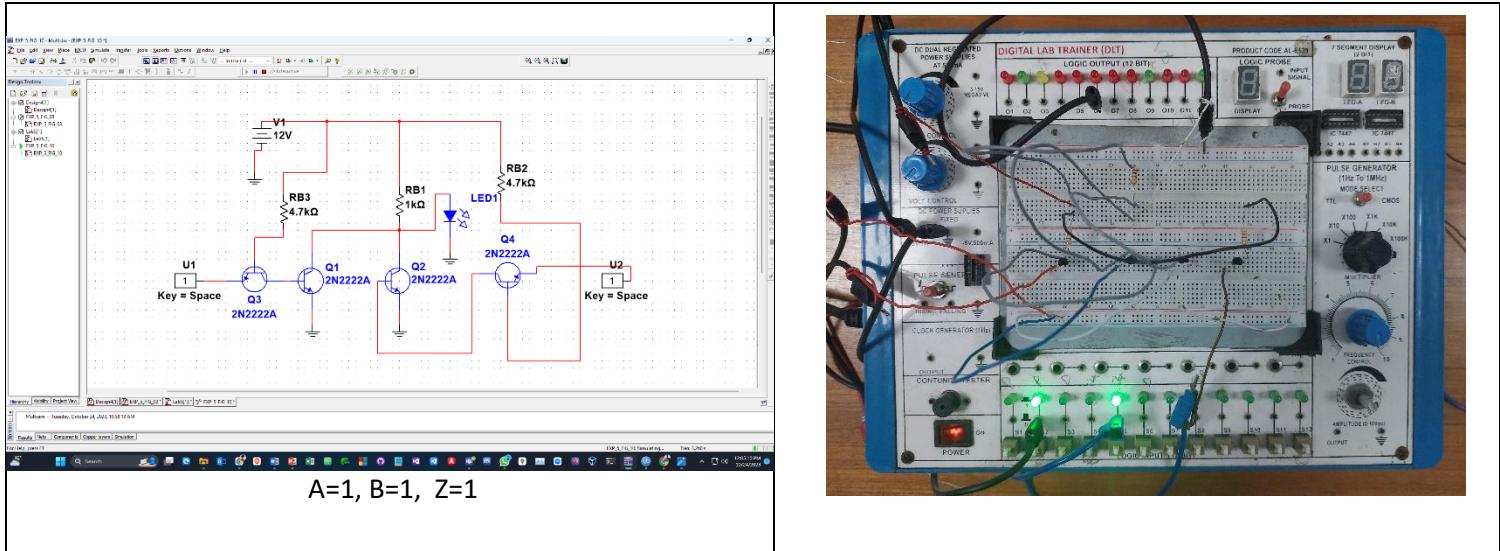


A=0, B=1, Z=0



A=1, B=0, Z=0





Report:

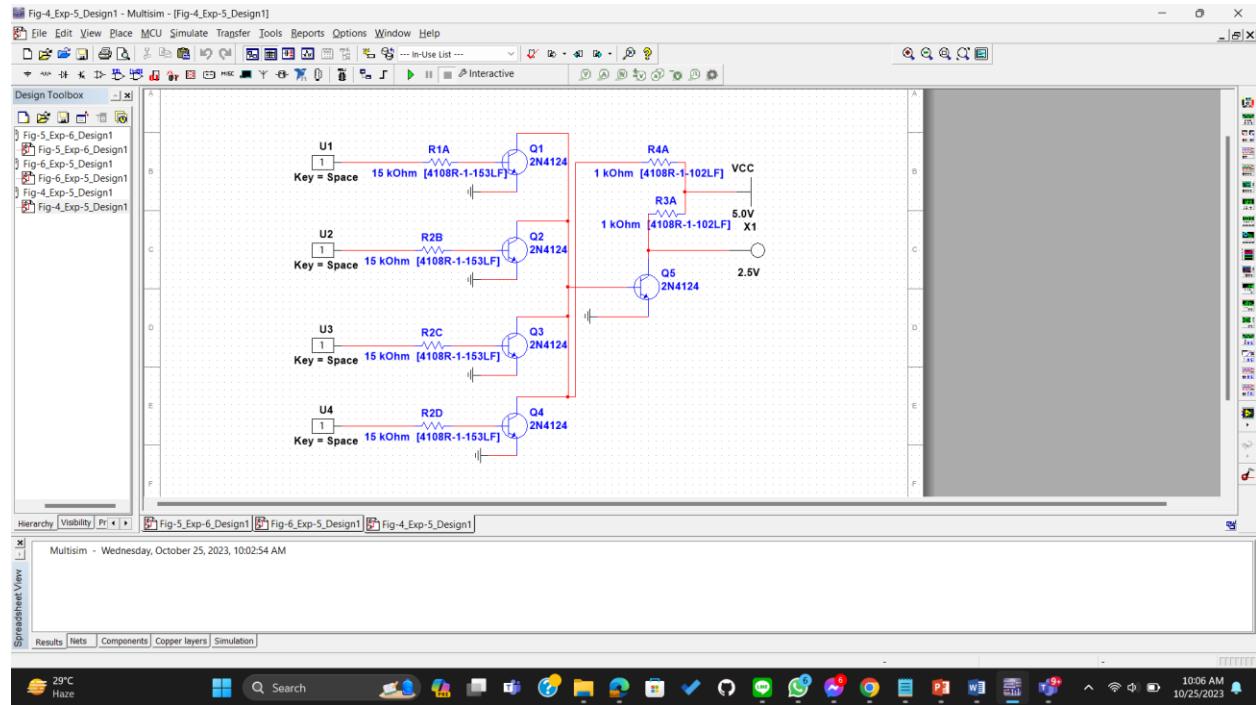
4. For each of the setups, the input 0 means low voltage and 1 means high voltage. For the different set ups we set different input and acquired the desired results. For every specific figure, we got the required output according the truth table as all the outputs matched with the truth table.

For an example, this is a truth table for 2 input NOR gate:

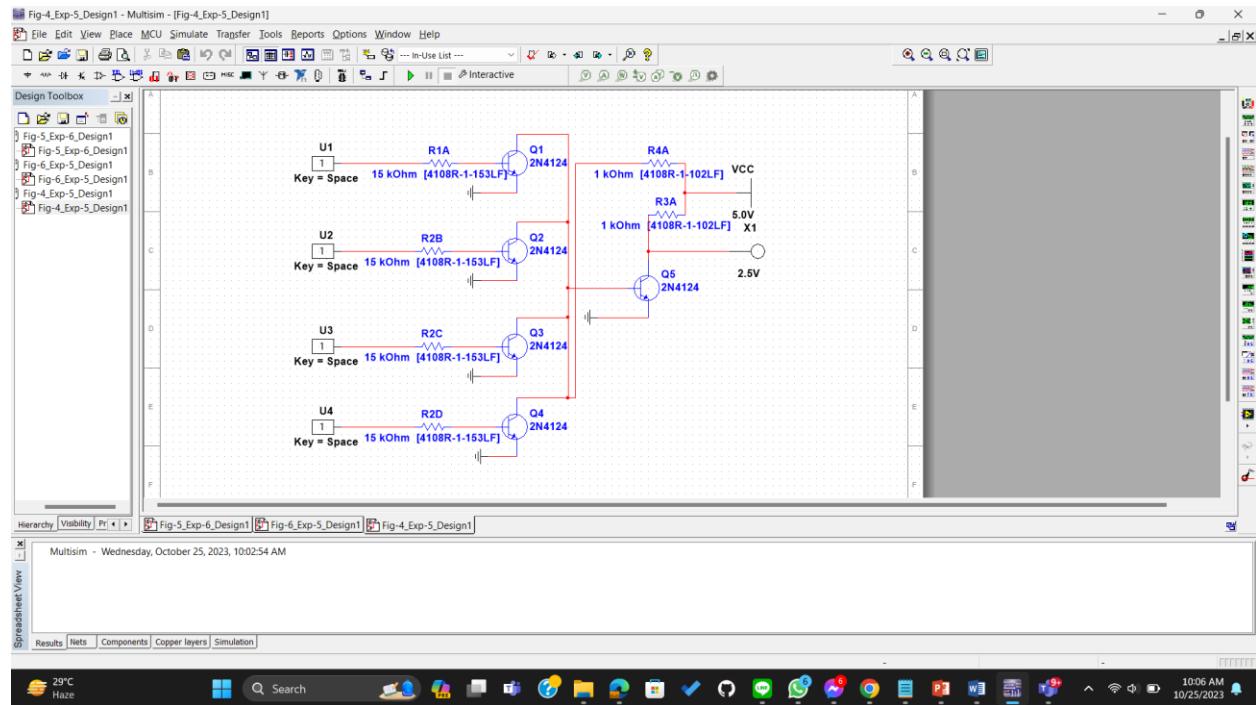
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

From the truth table, we can see the output will be low one and only when the both output are high as, we got the same result in both simulation & hardware.

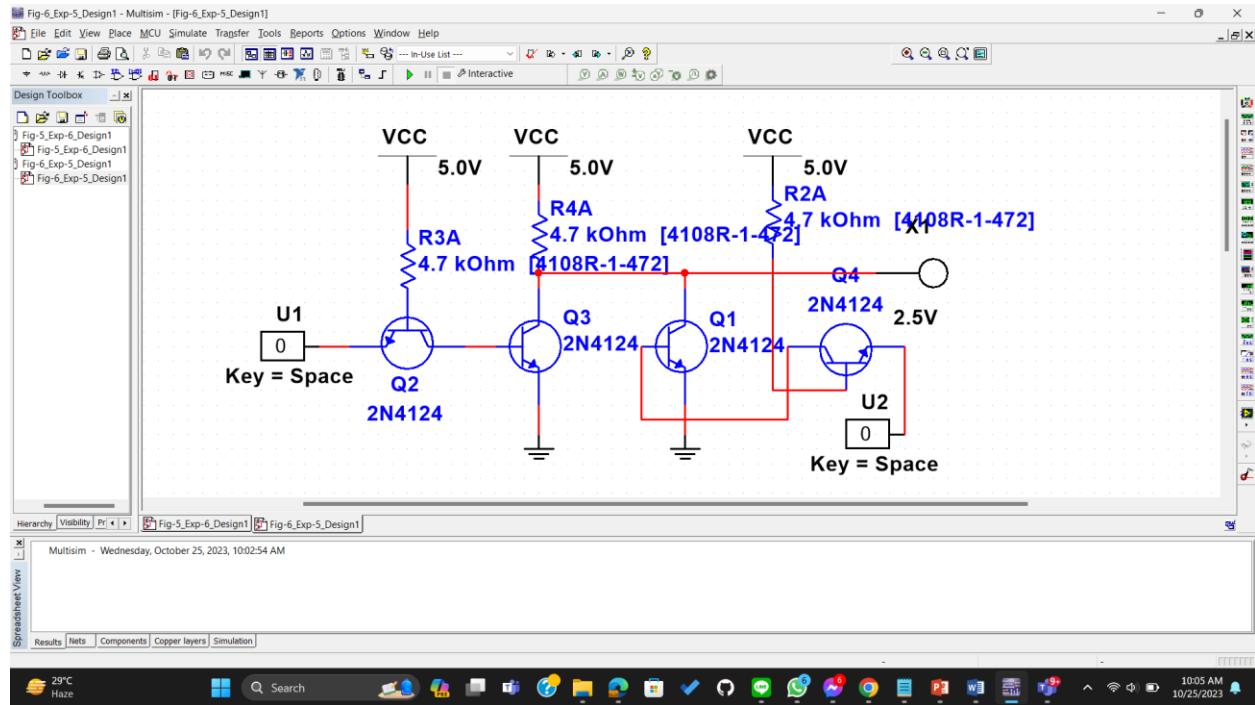
5. Figure: RTL 4-input OR gates:



6. Figure: 2-input TTL NAND gates



. Figure-2: 2-input TTL OR Gates



Result & Discussion:

In our experiments, we learned how to create different logic gates using electronic components. We started with diode logic circuits and successfully made OR and AND gates, and even combined them for more complex functions. In the second part of our experiment, we explored how to build logic gates using bipolar transistors, following three different designs: RTL, DTL, and TTL. We managed to make an RTL inverter, a DTL NAND gate, and a TTL NOR gate, and we could adapt them to have as many inputs as we needed. Throughout these experiments, we used DL (Diode Logic) and Bipolar Transistor logic circuits to perform various logic functions. We discovered that RTL makes it easy to create NOR gates, but NAND gates are a bit tricky. DTL, on the other hand, allows us to build simple NAND gates using just one transistor, along with some diodes and resistors. One thing we noticed is that DTL is slow because it takes time for the base charge to dissipate when there's no direct path to ground. It also requires some time to charge up the output's effective capacitance through an R_C (resistor-capacitor) network. TTL, however, solves the speed issue that DTL has, making it work more efficiently.

References:

1. Thomas L. Floyd, Digital Fundamentals, 9th Edition, 2006, Prentice Hal

