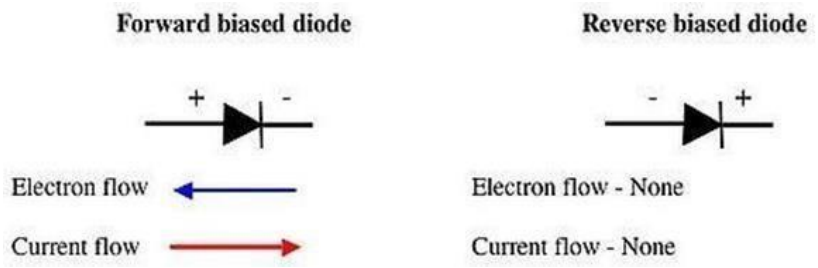
Part I: Construction of Diode Logic Gates

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 in another direction. The diode's two terminals are called the anode and cathode. In the diode symbol, the arrow points from the anode (the flat part of the triangle) toward the cathode (the point of the triangle).



The device operates by allowing current to flow from the anode to the 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 (DL) OR gate consists of nothing more than diodes (one for each input signal) and a resistor. Here, the 10 kΩ 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.

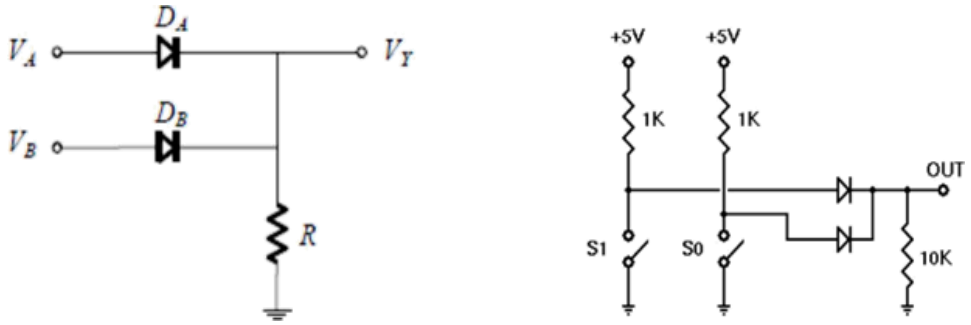


Fig. 1 DL OR Gate

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

Table 1: (a) Voltage-based truth table, (b) Logic-based or binary truth table of DL OR gate

V_A (volt)	V_B (volt)	V_Y (volt)
0	0	0
0	5	5
5	0	5
5	5	5

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	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 10KΩ 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.

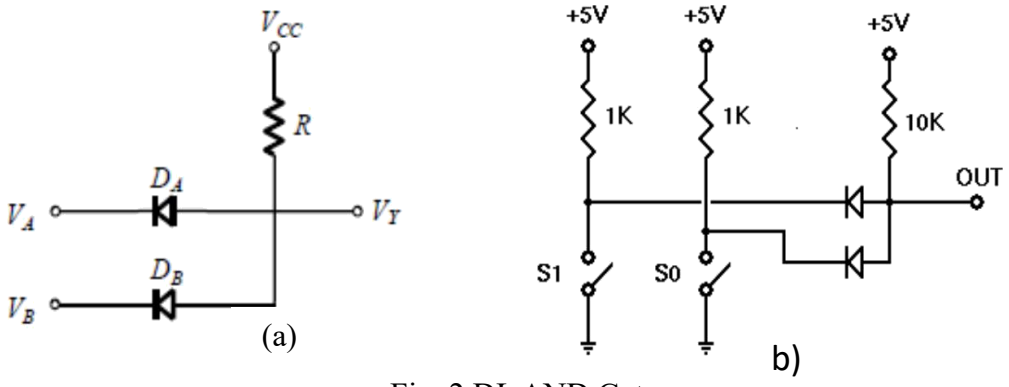


Fig. 2 DL AND Gate

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).

Table 2: (a) Voltage-based truth table, (b) Logic-based or binary truth table of DL AND gate

V_A (volt)	V_B (volt)	V_Y (volt)
0	0	0
0	5	0
5	0	0
5	5	5

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

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.

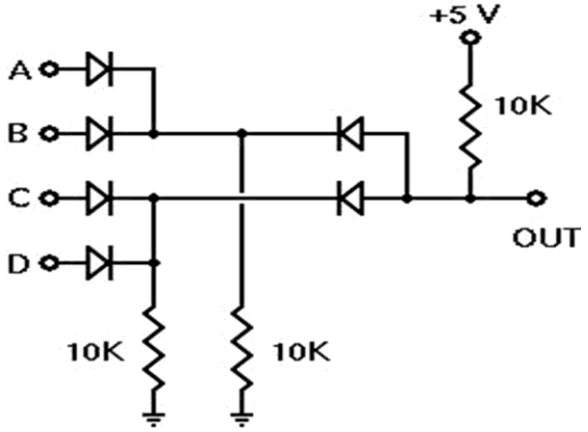


Fig.3 DL-AND-OR Gate

Table 3: truth table of DL AND-OR gate

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

Apparatus:

SL	Apparatus	Picture	Quantity	Remarks
1.	Digital trainer board.		1	Good condition
2.	Resistor 1 k ohm (Color band: brown-black-red). 10 k ohm (Color band: brown-black-orange).	 	4 4	Good condition
3.	1N4002 diodes		6	Good condition
4.	Power Supply		1	Good condition
5.	Connecting wires		multiple	Good condition

Experimental Procedure:

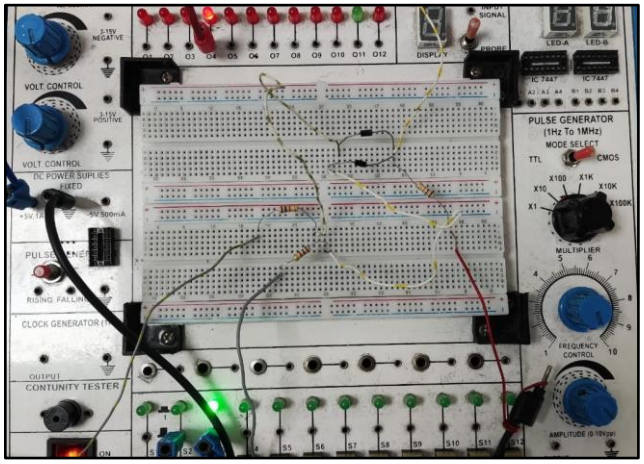
1. The DL-OR gate was constructed on the breadboard as shown in Fig. 1. A Truth Table similar to the provided one was drawn, and the experimental results were filled in.

2. The DL-AND gate was constructed on the breadboard as shown in Fig. 2. A Truth Table similar to the provided one was drawn, and the experimental results were filled in.

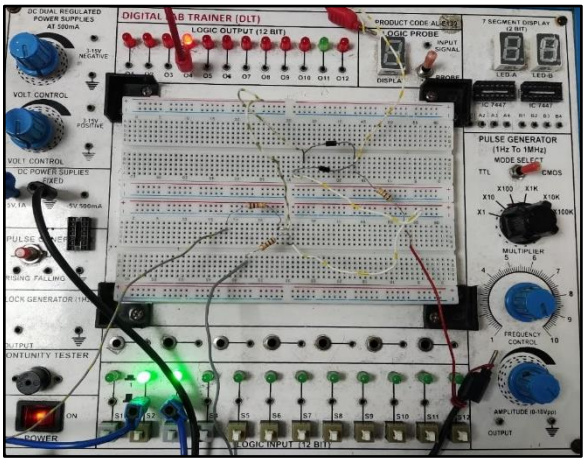
3. The DL-AND-OR gate was constructed on the breadboard as shown in Fig. 3. Before beginning the experiment, the expected results for all the different input combinations were calculated and put in a Truth Table similar to the one provided. Then, a second Truth Table was drawn, and it was filled with the experimental output values.

Simulation and Measurement:

Experimental Setup:

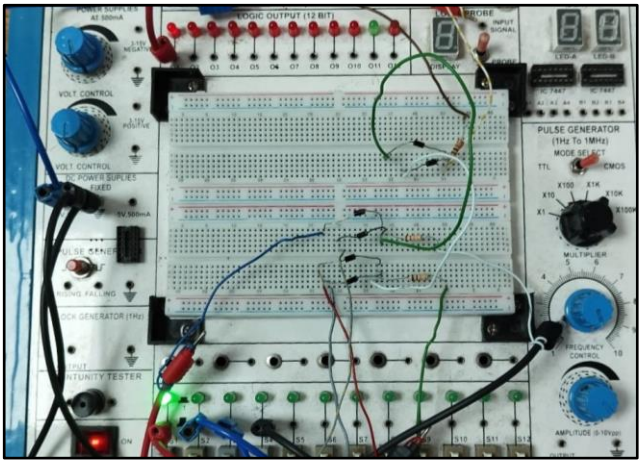


a) A=0, B=1 and output Y=1

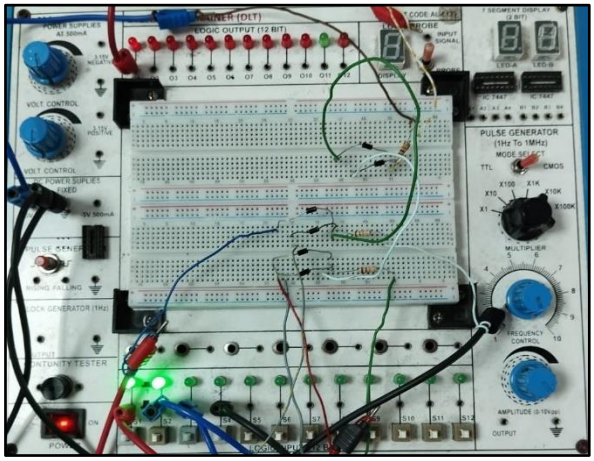


b) A=1, B=1 and output Y=1

Fig. 4: Implementing DL OR GATE



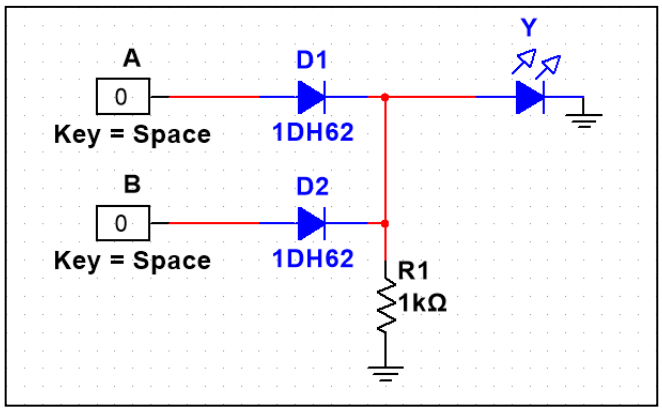
a) A=0, B=1 and output Y=1



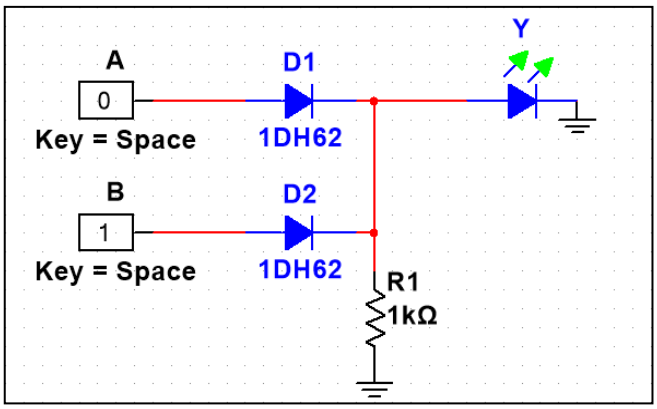
b) A=1, B=1 and output Y=1

Fig. 5: Implementing DL AND-OR GATE

Simulations:

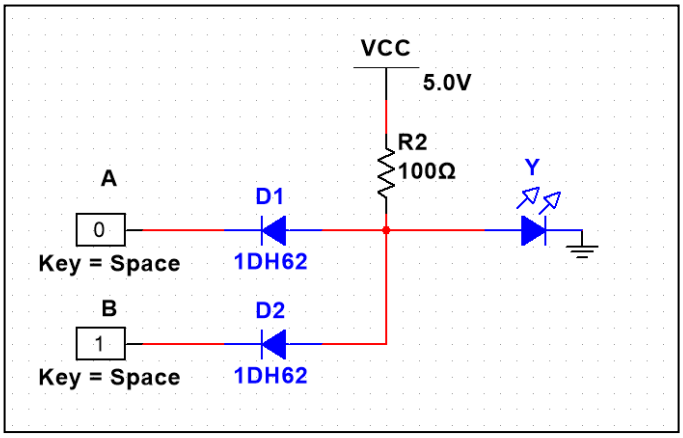


a) A=0, B=0 and output Y=0

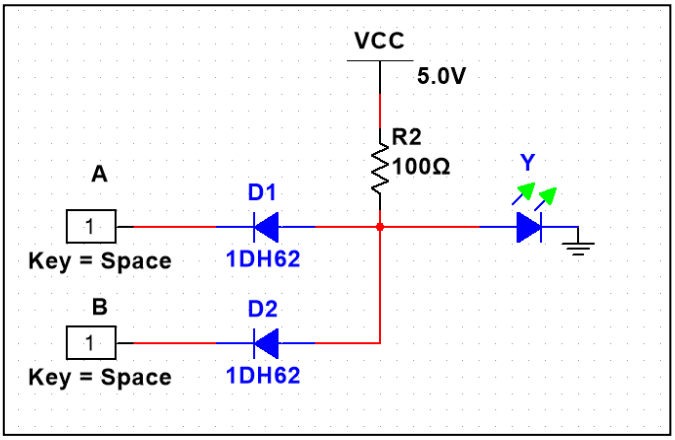


b) A=0, B=1 and output Y=1

Fig. 6: Implementing DL OR GATE

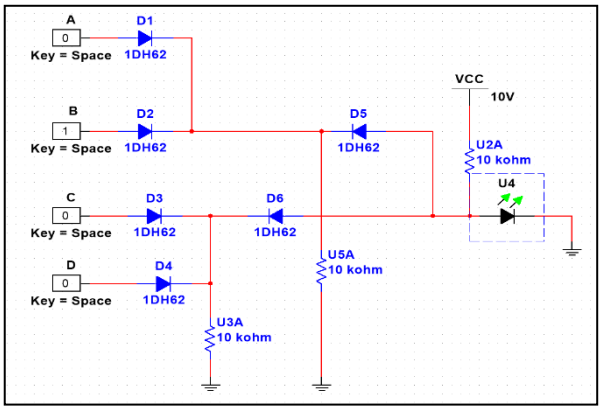


a) A=0, B=0 and output Y=0

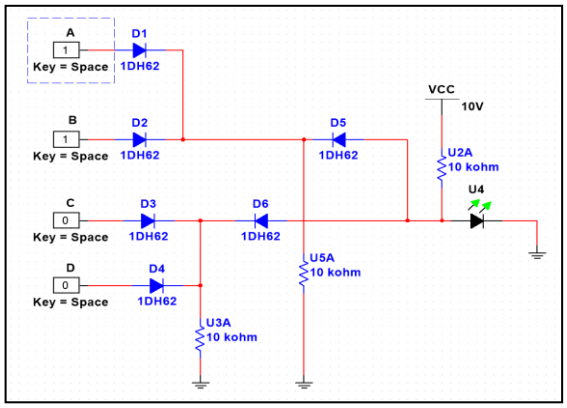


b) A=0, B=1 and output Y=1

Fig. 7: Implementing DL AND GATE



a) A=0, B=1 and output Y=1



b) A=1, B=1 and output Y=1

Fig. 8: Implementing DL AND-OR GATE

Results and Discussion:

The experiment on "Construction of Diode Logic Gates" proceeded smoothly, with no notable difficulties encountered and devoid of errors. Our understanding of digital logic facilitated effortless derivation of logical expressions, implemented seamlessly in both hardware and software using Ni Multisim version-14.3. Meticulous planning ensured accurate representations of logic functions, while effective collaboration ensured a seamless workflow. The precise functionality of our constructed diode logic gates was confirmed through meticulous verification of the truth table. This experience enhanced both our theoretical understanding and practical skills in digital electronics.

Questions with answers for report writing:

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.

Ans: Each of the above setups are various combinations of signals according to the truth table. The data represents when the output will be high, and for which kind of inputs the data will be high or low. And our results match the expected ideal outputs.

2. Why are diode logic gates not suitable for cascading operations?

Ans: Diode logic gates are not suitable for cascading operations because of voltage drop issues. They have a high voltage drop across them, leading to signal degradation and the inability to reliably pass signals from one stage to the next.

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. 9: 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.

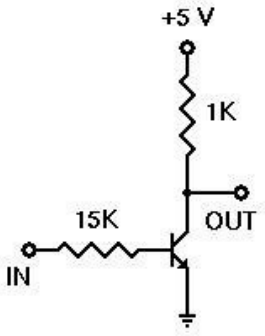


Fig.10: RTL Inverter

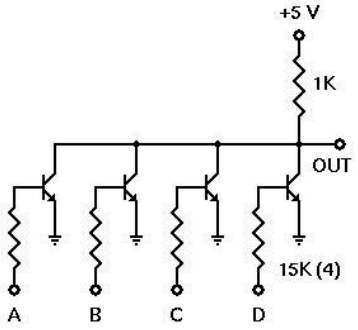


Fig .11: 4-input RTL Inverter

Table 4: truth table of two input RTL Inverter

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

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.

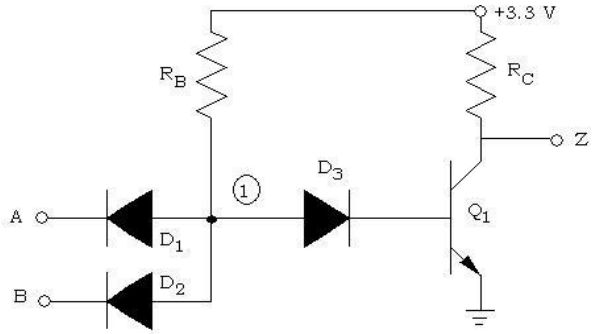


Fig. 12: 2-input DTL NAND Gate

Table 5: truth table of two input DTL NAND gate

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

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 Rc 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.

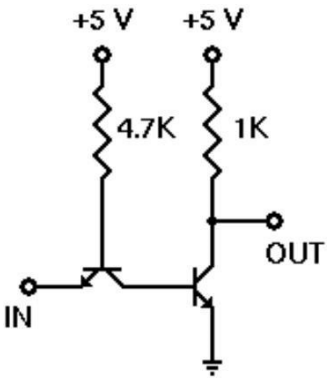


Fig. 13: TTL Inverter

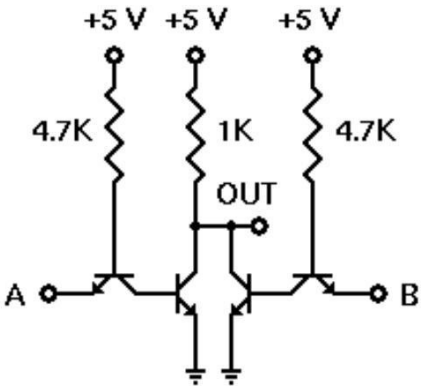
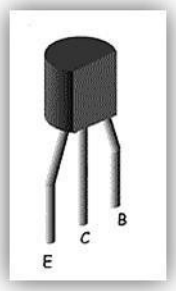


Fig. 14: 2-input TTL NOR gate

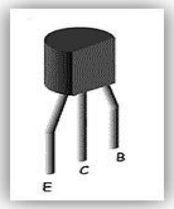
Table 6: truth table of two input TTL NOR gate

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

BJT pin configuration:



Apparatus:

SL	Apparatus	Picture	Quantity	Remarks
1.	Digital trainer board.		1	Good condition
2.	Resistor 1 k ohm (Color band: brown-black-red). 15 k ohm (Color band: brown-green-orange).	 	4 4	Good condition
3.	2N4124 NPN silicon transistor		6	Good condition
4.	Power Supply		1	Good condition
5.	Connecting wires		multiple	Good condition

Experimental Procedure:

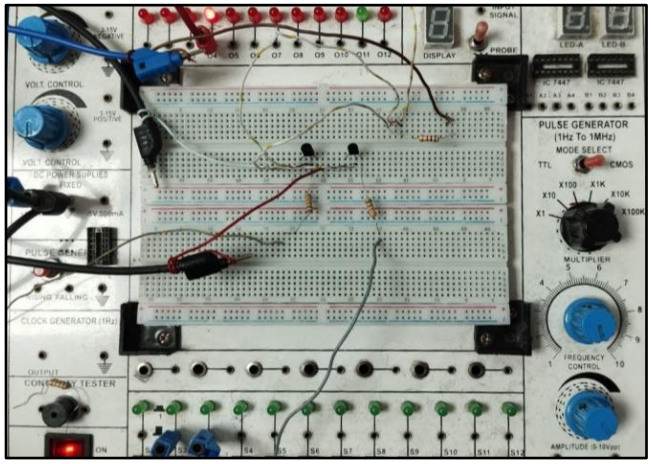
(1) The circuit for the RTL inverter was set up as shown in Fig. 10.

(2) For each input combination, the output was found and placed in a Truth Table. The Truth Table had two sets of outputs - one ideal and one experimental.

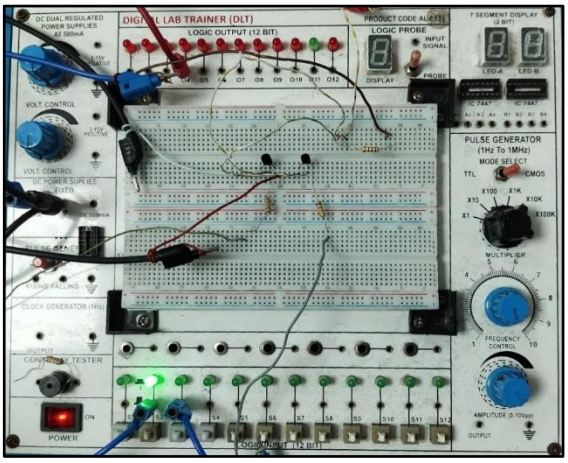
(3) Steps 1 and 2 were repeated for each circuit set-up from Fig. 11 and Fig. 13.

Simulation and Measurement:

Experimental Setup:

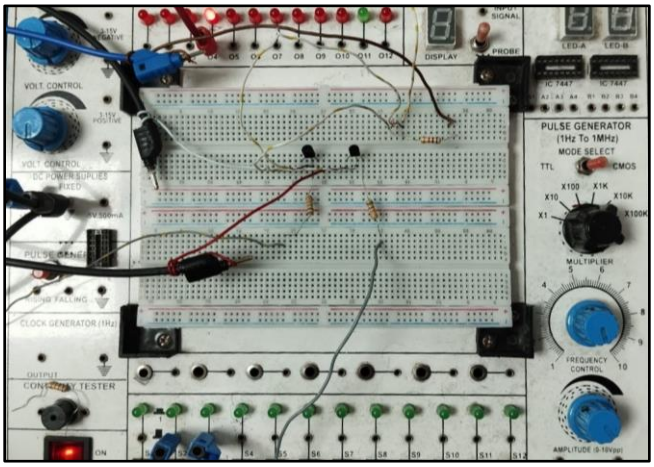


a) For A=0, B=0 output Y=1.

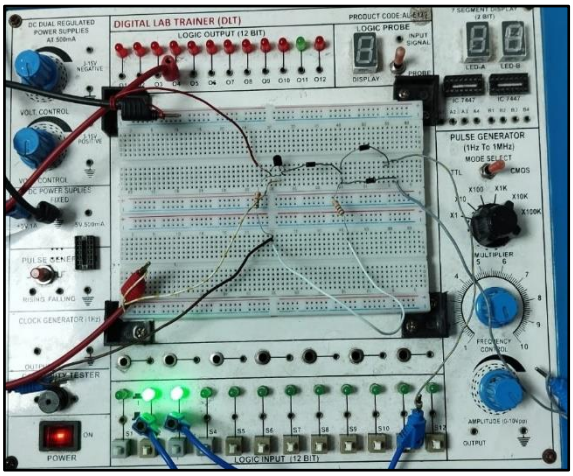


b) For A=1, B=0 output Y=0.

Fig. 15: Implementing two input RTL Inverter

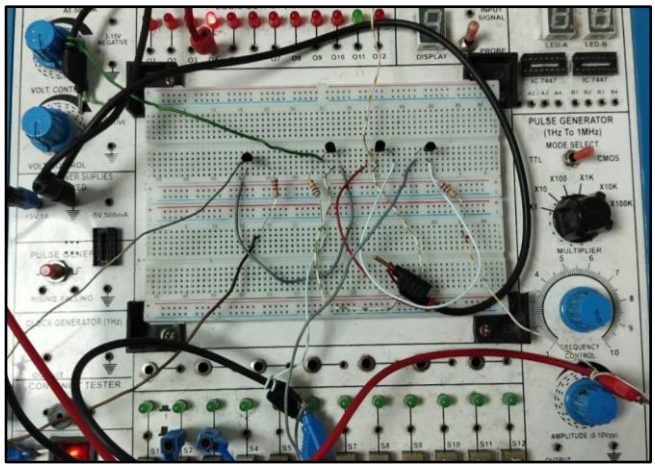


a) For A=0, B=0 output Y=1.

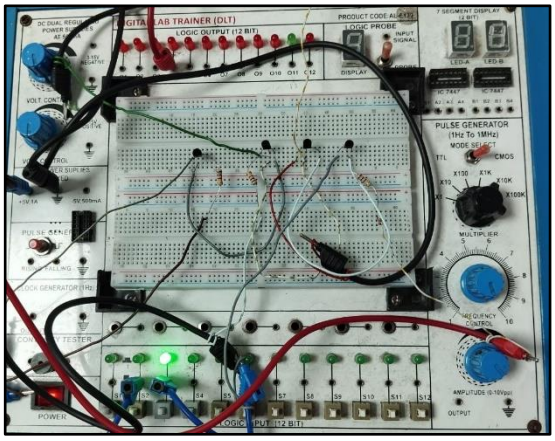


b) For A=1, B=1 output Y=0.

Fig. 16: Implementing two input DDL NAND Gate



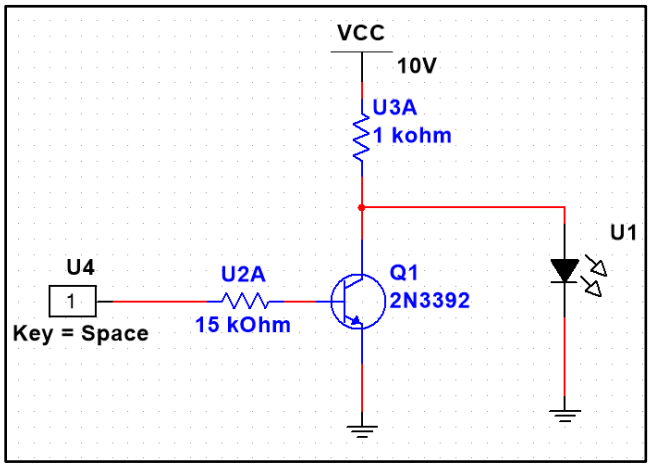
a) For A=0, B=0 output Y=1.



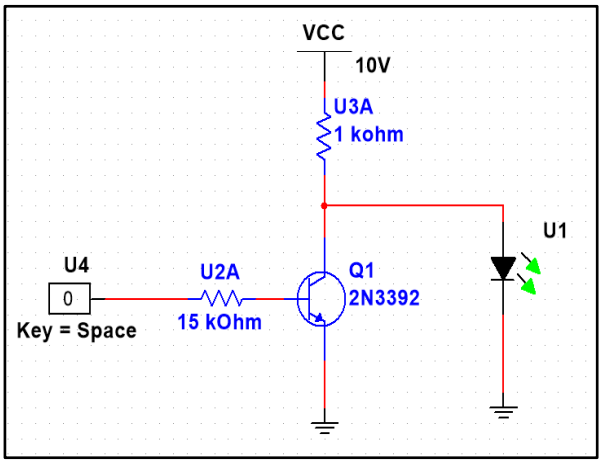
b) For A=0, B=1 output Y=0.

Fig. 17: Implementing two input TTL NOR Gate

Simulations:

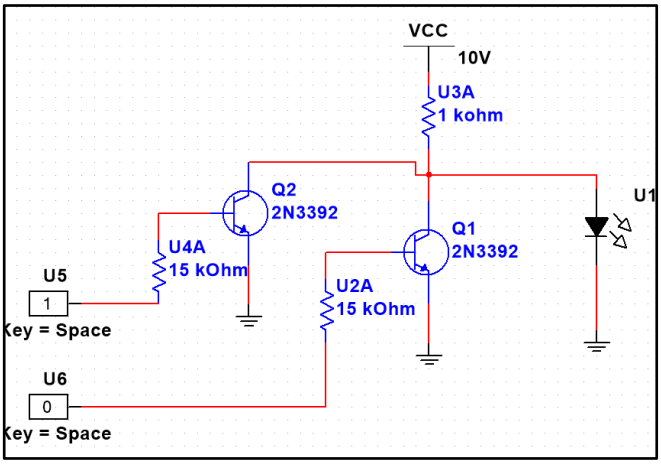


a) For A=1, Output Y=0

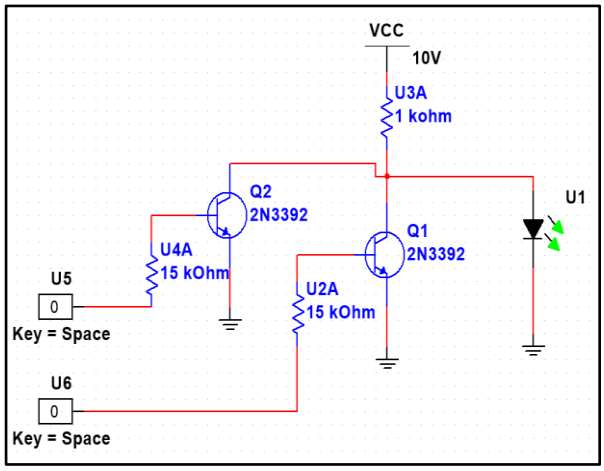


b) For A=0, Output Y=1

Fig. 18: Implementing RTL Inverter

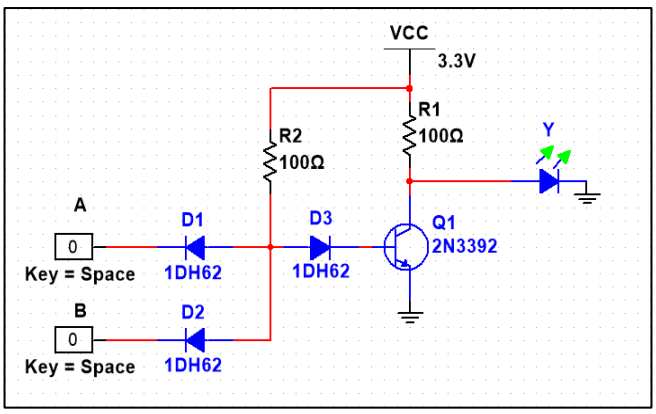


a) For A=1, B=0 output Y=0

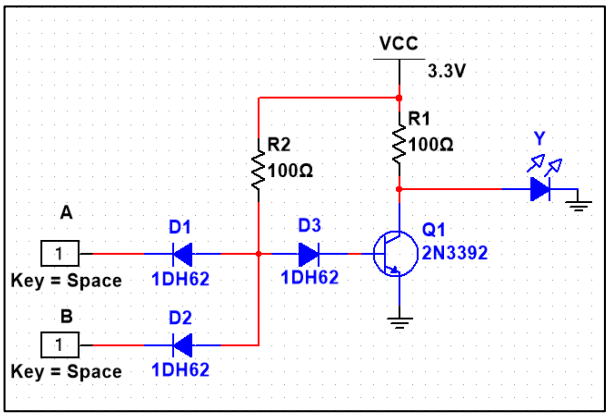


b) For A=0, B=0 output Y=1

Fig. 19: Implementing two input RTL Inverter

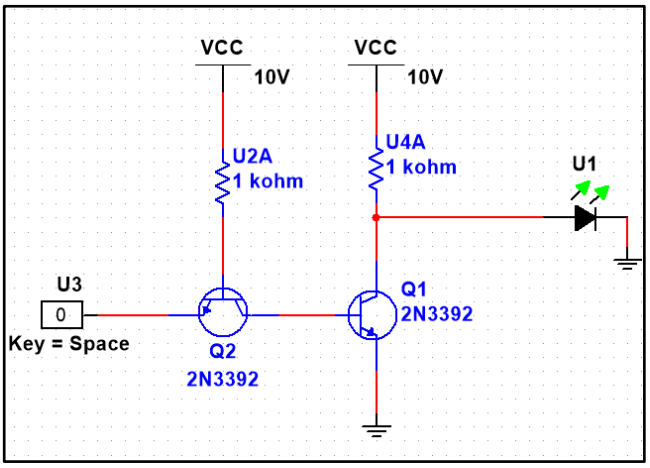


a) For A=0, B=0 output Y=1.

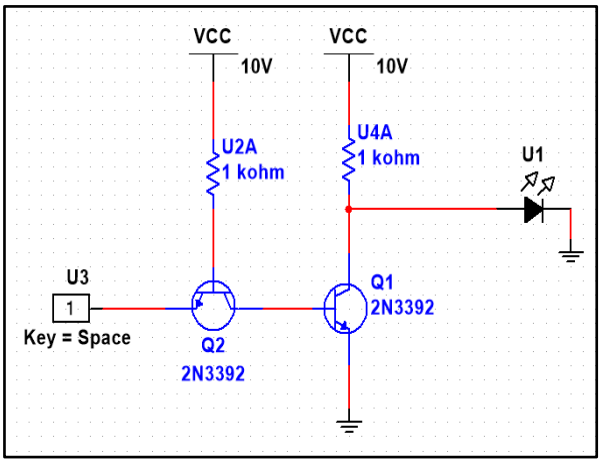


b) For A=1, B=1 output Y=0.

Fig. 20: Implementing two input DDL NAND Gate

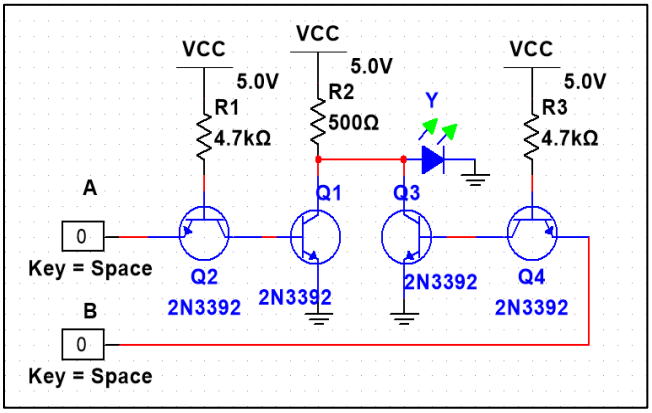


a) For A=0, output Y=1

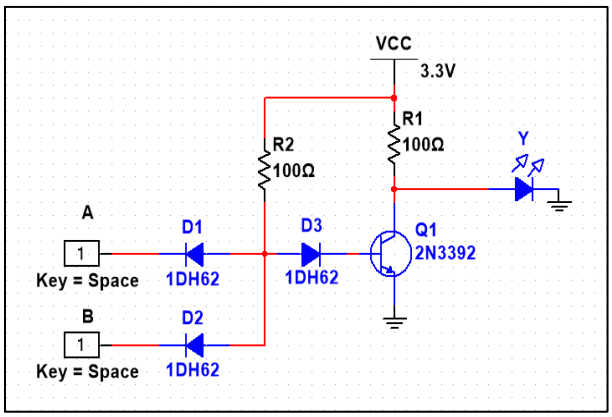


b) For A=1, output Y=0

Fig. 21: Implementing TTL Inverter



a) For A=0, B=0 output Y=1.



b) For A=0, B=1 output Y=0.

Fig. 22: Implementing two input TTL NOR Gate

Results and Discussions:

In the experiment "Construction of Diode Logic Gates," we seamlessly derived logical expressions and translated them into hardware on a breadboard and software simulations using Ni Multisim version-14.3. We ensured error-free constructions and meticulously verified the truth table, confirming the precise functionality of our diode logic gates. To get output in simulation we had to use 10V input instead of 5V .This experiment deepened our understanding of logic gate operations across different circuit configurations, enhancing both theoretical knowledge and practical skills in digital logic design.

Questions with answers for report writing:

1. For, each of the above set-ups, describe in words what the data means. Did your results match theexpected ideal outputs? If not, explain why?

Ans: The data collected from these setups involves input-output pairs for various logic gates. This data helps us understand how different types of gates (RTL, DTL, TTL) respond to different input combinations, revealing their behavior and characteristics in digital logic circuits And our results match the expected ideal outputs.

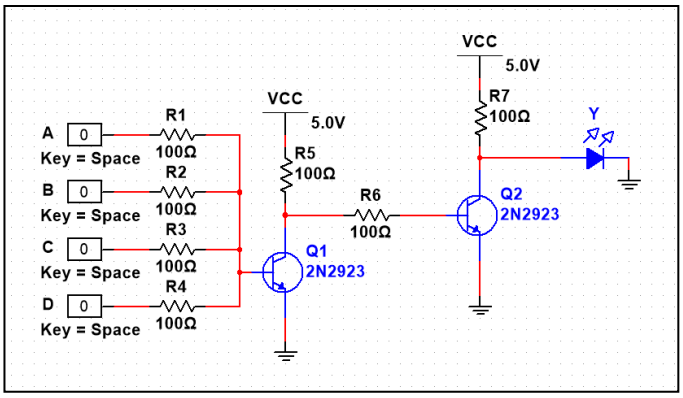
2. Design RTL and TTL 4-input OR gates.

Ans:

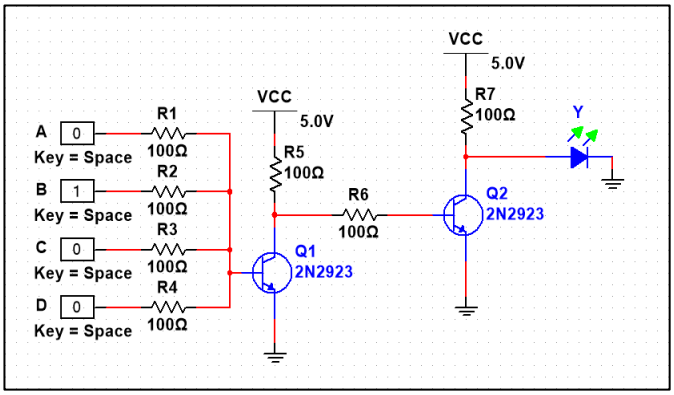
Table 7: truth table of four input RTL and TTL OR gate

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

4-input RTL OR Gate:



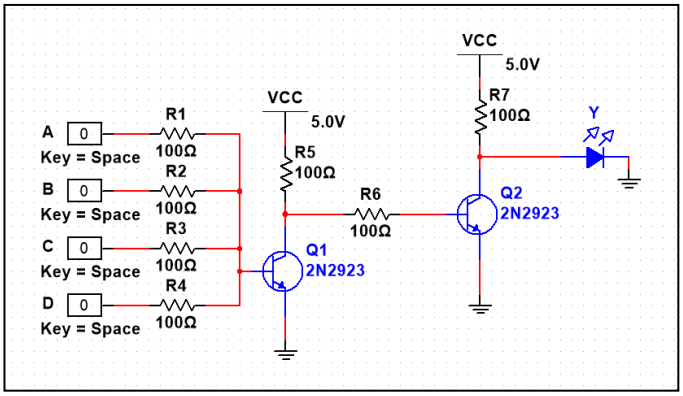
a) For A=0, B=0, C=0, D=0, Output Y=0



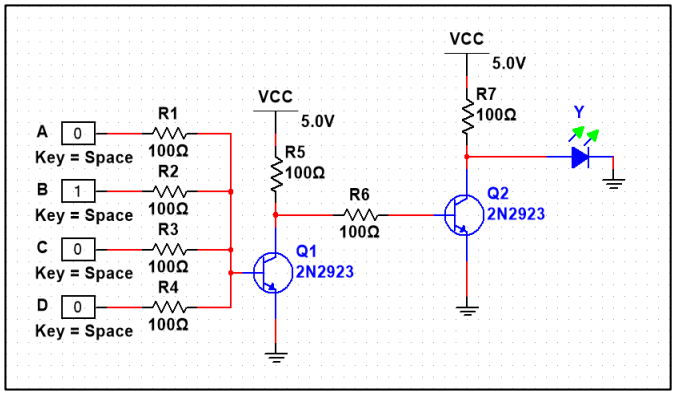
b) For A=0, B=1, C=0, D=0, Output Y=1

Fig. 23: Implementing Four input RTL OR Gate

4-input TTL OR Gate:



a) For A=0, B=0, C=0, D=0, Output Y=0

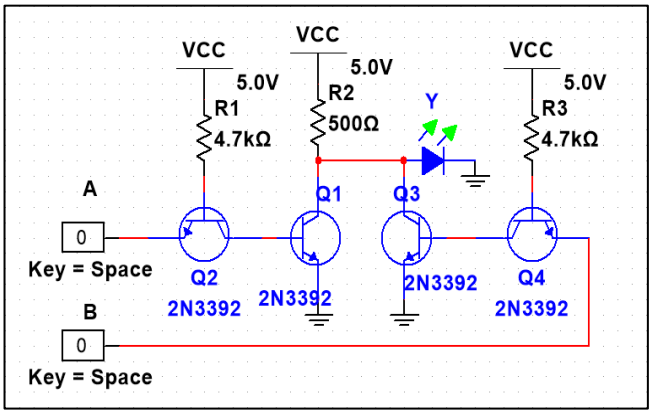


b) For A=0, B=1, C=0, D=0, Output Y=1

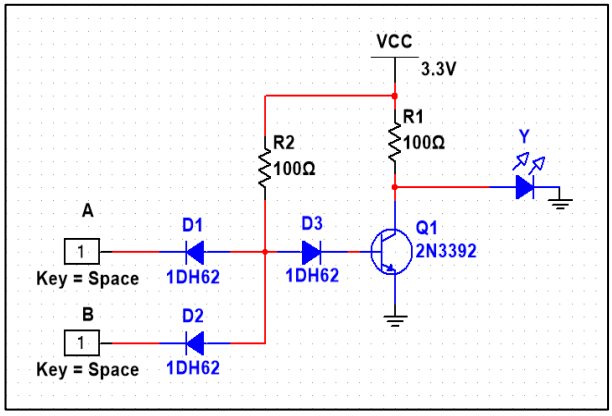
Fig. 24: Implementing Four input RTL OR Gate

3. Design 2-input TTL NAND and NOR gates.

2-input TTL NOR Gate



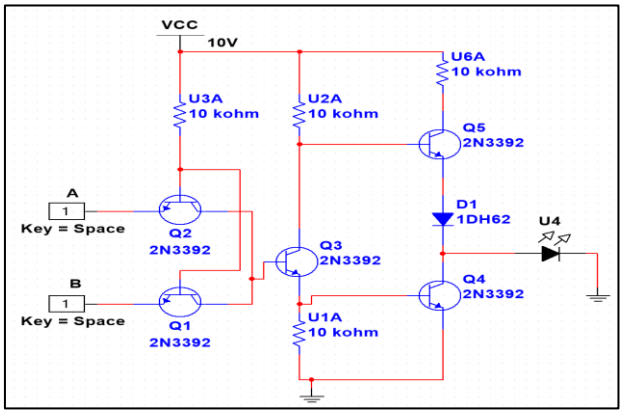
a) For A=0, B=0 output Y=1.



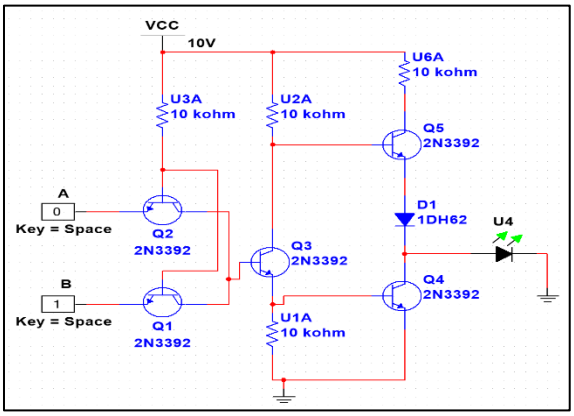
b) For A=0, B=1 output Y=0.

Fig. 25: Implementing two input TTL NOR Gate

2-input TTL NAND Gate



a) For A=1, B=1 output Y=0.



b) For A=0, B=1 output Y=1.

Fig. 26: Implementing two input TTL NAND Gate

References:

[1] www.tutorialspoint.com
[2] www.electronics-tutorials.ws
[3] faculty.kfupm.edu.sa
[4] “Digital Fundamentals” by Thomas L. Floyd
[5] American International University–Bangladesh (AIUB) Digital Logic And Circuits Lab Manual.