B.TECH. COMPUTER SCIENCE AND ENGINEERING,
SEMESTER III
GNDU, AMRITSAR

Practical File

Anhat Singh

Roll No: 17032000307

Subject: Digital Circuit Logic Design

Submitted To: Ms. Jaspreet Kaur

Anhat Singh

2021

GURU NANAL DEV UNIVERSITY, AMRITSAR

ECL-119: DIGITAL CIRCUIT LOGIC DESIGN

Contents

Activity 1

AND, OR, NOT gates

1.1 Aim

To verify and interpret the logic and truth table for AND, OR, NOT gates using Resistor Transistor Logic (RTL), Diode Resistance Logic (DRL) and Transitor Logic (TL)

1.2 Apparatus

- Kit for realization of gates
- Connecting Leads

1.3 Circuits

1.4 Theory

Logic gates are the basic building blocks of any digital system. Logic gates are electronic circuits having one or more than one input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as:

1. AND gate
2. OR gate

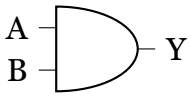
3. NOT gate
4. NAND gate
5. NOT gate
6. Ex-OR gate
7. Ex-NOR gate

1.4.1 AND gate

The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot $(.)$ is used to show the AND operation i.e. $A.B$ or can be written as AB.

$$Y = A.B$$

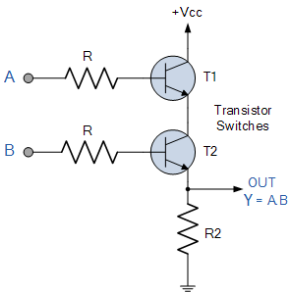
A simple 2-input logic AND gate can be constructed using RTL (Resistor-Transistor-Logic) switches connected together as shown below with the inputs connected directly to the transistor bases. Both transistors must be saturated "ON" for an output at Q.



(a) Symbol

Input		Output
A	B	$Y = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

(b) Truth Table



(c) RTL Design

Figure 1.4.1: *AND gate*

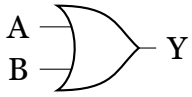
1.4.2 OR gate

The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high. A plus $(+)$ is used to show the OR operation.

$$Y = A + B$$

OR gate can be realized by DRL (Diode-Resistance-Logic) or by TTL (Transistor-Transistor-Logic). Presently, we will learn how to implement the OR gate using DRL (Diode-Resistance-Logic). To realise OR gate, we will use a diode at every input of the OR gate. The anode part of diode is connected with input while the cathode part is joined together and a resistor, connected with the cathode is grounded. In this case, we have taken two inputs which can be seen in the circuit below.

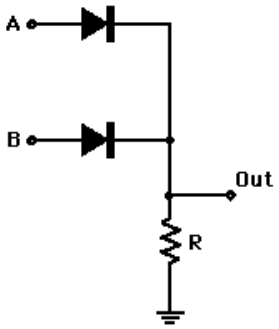
When both the inputs are at logic 0 or low state then the diodes D_1 and D_2 become reverse biased. Since the anode terminal of diode is at lower voltage level than the cathode terminal, so diode will act as open circuit so there is no voltage across resistor and hence output voltage is same as ground. When either of the diodes is at logic 1 or high state then the diode corresponding to that input is forward bias. Since this time anode is at high voltage than cathode therefore current will flow through forward biased diode and this current then appears on resistor causing high voltage at output terminal also. Hence at output we get high or logic 1 or $+5V$. So, if any or both inputs are high, the output will be high or 1.



(a) Symbol

Input		Output
A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(b) Truth Table



(c) DRL Design

Figure 1.4.2: *OR gate*

1.4.3 NOT gate

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as $NOT A$. This is also shown as A' or A with a bar over the top,

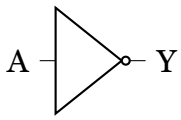
as shown at the outputs.

$$Y = \overline{A}$$

NOT gate can be realized through transistor.The input is connected through resistor R_2 to the transistor's base. When no voltage is present on the input, the transistor turns off. When the transistor is off, no current flows through the collector-emitter path. Thus, current from the supply voltage (V_{cc}) flows through resistor R_1 to the output. In this way, the circuit's output is high when its input is low.

When voltage is present at the input, the transistor turns on, allowing current to flow through the collector-emitter circuit directly to ground. This ground path creates a shortcut that bypasses the output, which causes the output to go low.

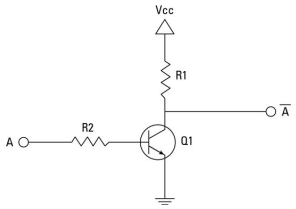
In this way, the output is high when the input is low and low when the input is high.



(a) Symbol

Input	Output
A	$Y = \overline{A}$
0	1
1	0

(b) Truth Table



(c) TL Design

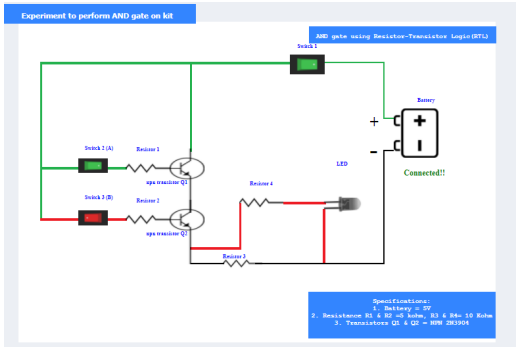
Figure 1.4.3: *NOT gate*

1.5 Procedure

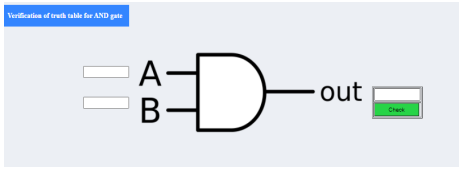
1.5.1 AND gate

Simulator 1

1. Connect the supply(+5V) to the circuit.
2. Press the switches for inputs "A" and "B".



(a) Simulator 1



(b) Simulator 2

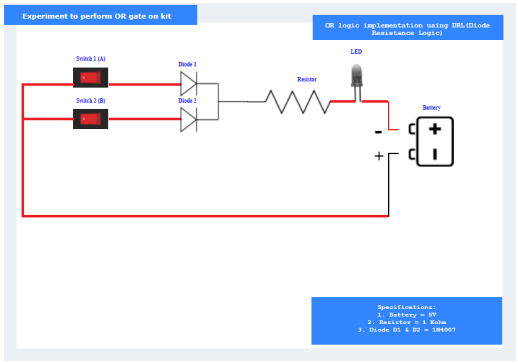
Figure 1.5.1: *Simulator for realizing circuit for AND gate*

3. The bulb does not glow if any one or both the switches (2 and 3) are OFF and glows only if both the switches (2 and 3) are ON.
4. Repeat step-2 and step-3 for all state of inputs.

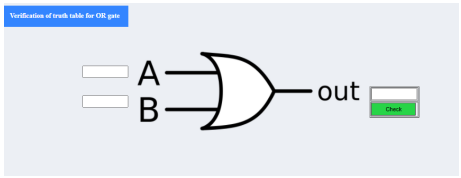
Simulator 2

1. Enter the Boolean input ”A” and ”B”.
2. Enter the Boolean output for your corresponding inputs.
3. Click on ”Check” Button to verify your output.

1.5.2 OR gate



(a) Simulator 1



(b) Simulator 2

Figure 1.5.2: *Simulator for realizing circuit for OR gate*

Simulator 1

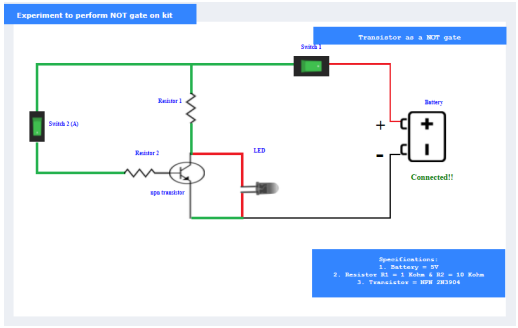
1. Connect the supply(+5V) to the circuit.
2. Press the switches for inputs ”A” and ”B”.

3. The bulb glows if any one or both the switches are ON else it won't glow.
4. Repeat step-2 and step-3 for all state of inputs.

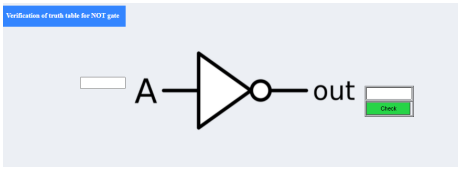
Simulator 2

1. Enter the Boolean input "A" and "B".
2. Enter the Boolean output for your corresponding inputs.
3. Click on "Check" Button to verify your output.

1.5.3 NOT gate



(a) Simulator 1



(b) Simulator 2

Figure 1.5.3: *Simulator for realizing circuit for NOT gate*

Simulator 1

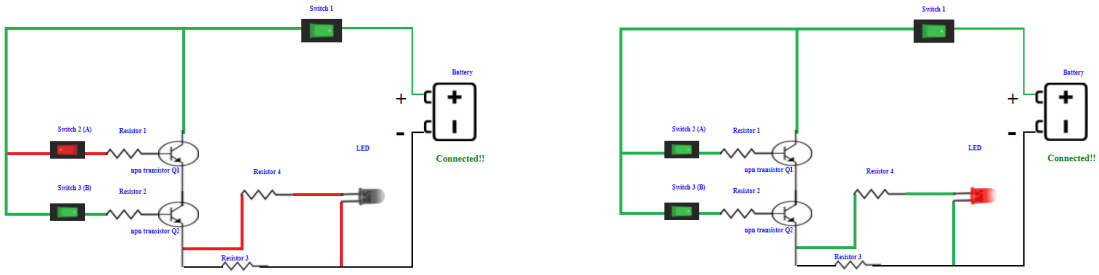
1. Connect the supply(+5V) to the circuit.
2. Press the switch 1 to connect battery to the circuit.
3. Press the switch 2 for input "A".
4. The bulb glows if switch 2 is OFF else it won't glow.

Simulator 2

1. Enter the Boolean input "A".
2. Enter the Boolean output for your corresponding input.
3. Click on "Check" Button to verify your output.

1.6 Observations

1.6.1 AND gate



(a) Either of the Inputs OFF, LED is OFF (b) Both Inputs ON, LED is ON

Figure 1.6.1: *Observations for different Input Values*

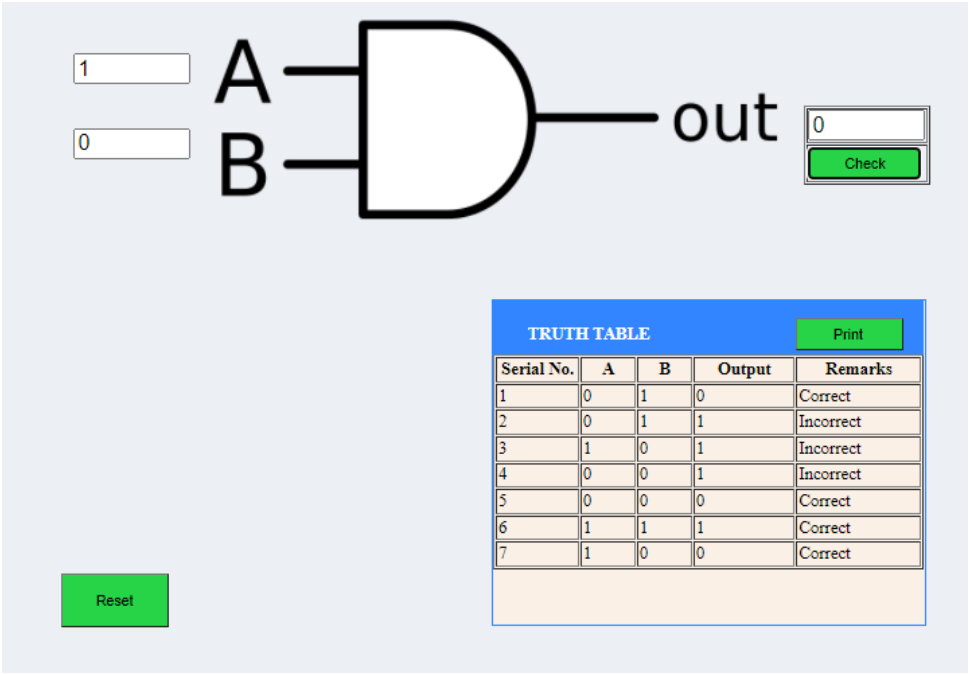
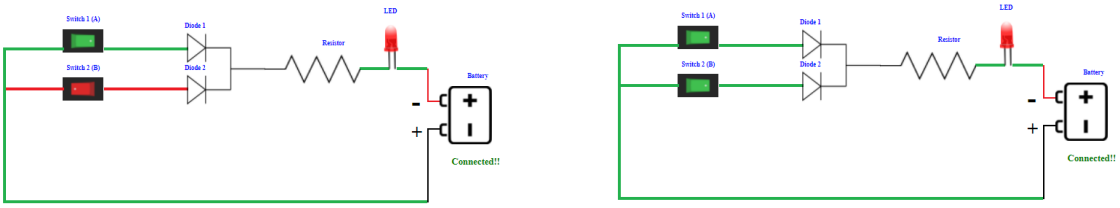


Figure 1.6.2: *Observations for verification of Truth Table of the AND gate*

1.6.2 OR gate



(a) Either of the Inputs ON, LED is ON

(b) Both Inputs OFF, LED is OFF

Figure 1.6.3: *Observations for different Input Values*

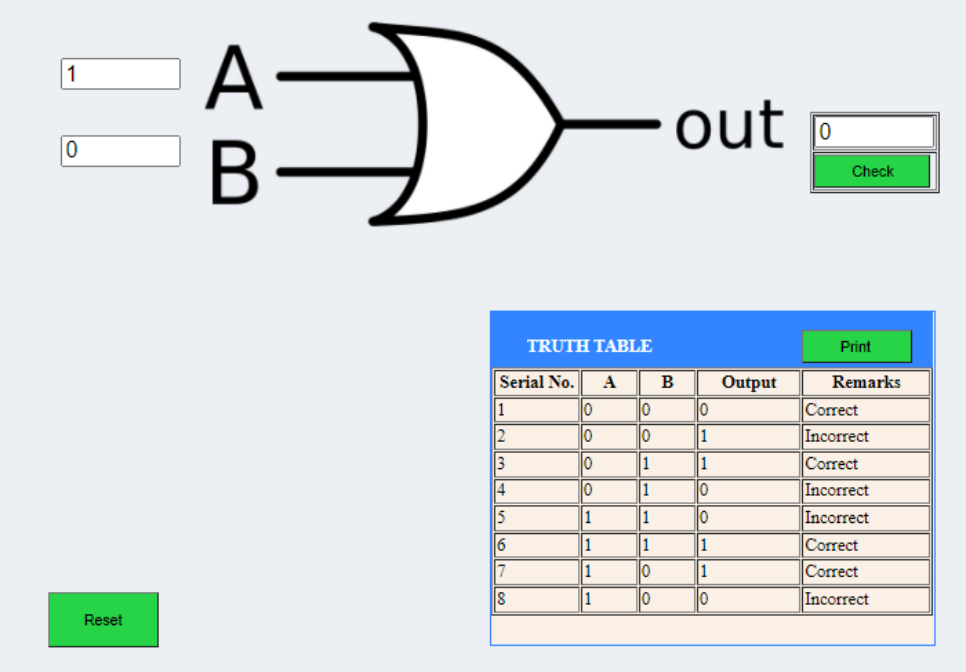
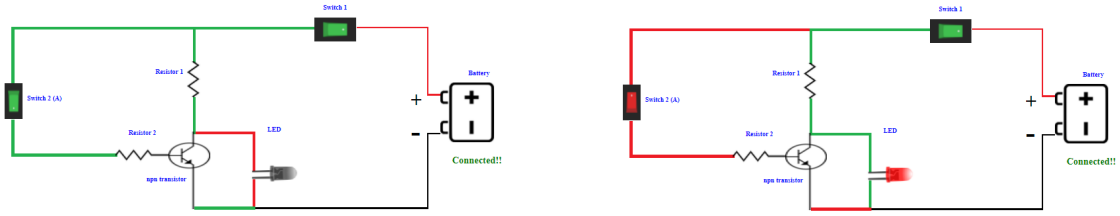


Figure 1.6.4: *Observations for verification of Truth Table of the OR gate*

1.6.3 NOT gate



(a) Input is ON, Output is OFF

(b) Input is OFF, Output is ON

Figure 1.6.5: *Observations for different Input Values*

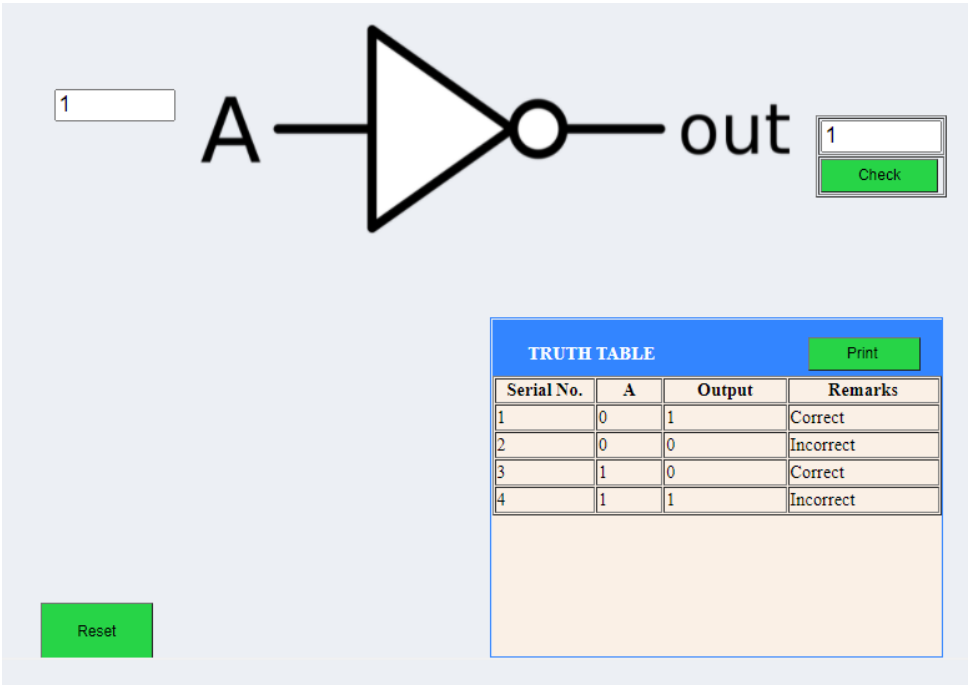


Figure 1.6.6: *Observations for verification of Truth Table of the NOT gate*

1.7 Precautions

1. Make the connections when power supply is OFF.
2. Ensure that the connections are tight.
3. Change the status of inputs only when power supply is OFF.

Activity 2

NAND, NOR gates

2.1 Aim

To verify and interpret the logic and truth table for NAND, NOR gates using Resistor Transistor Logic (RTL)

2.2 Apparatus

- Kit for realization of gates
- Connecting Leads

2.3 Circuits

2.4 Theory

Logic gates are the basic building blocks of any digital system. Logic gates are electronic circuits having one or more than one input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as:

1. AND gate
2. OR gate
3. NOT gate

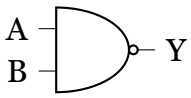
4. NAND gate
5. NOT gate
6. Ex-OR gate
7. Ex-NOR gate

2.4.1 NAND gate

This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

$$Y = \overline{A.B}$$

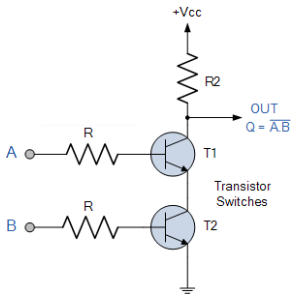
A simple 2-input logic NAND gate can be constructed using RTL (Resistor-transistor-logic) switches connected together as shown in Figure 2.4.1c with the inputs connected directly to the transistor bases. Either transistor must be cut-off or "OFF" for an output at Q.



(a) Symbol

Input		Output
A	B	$Y = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

(b) Truth Table



(c) RTL Design

Figure 2.4.1: *NAND gate*

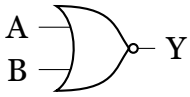
2.4.2 NOR gate

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR

gate with a small circle on the output. The small circle represents inversion.

$$Y = \overline{A + B}$$

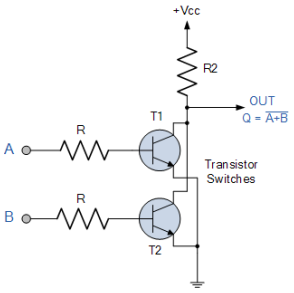
A simple 2-input logic NOR gate can be constructed using RTL (Resistor-transistor-logic) switches connected together as shown in Figure 2.4.2c with the inputs connected directly to the transistor bases. Both transistors must be cut-off or "OFF" for an output at Q.



(a) Symbol

Input		Output
A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

(b) Truth Table

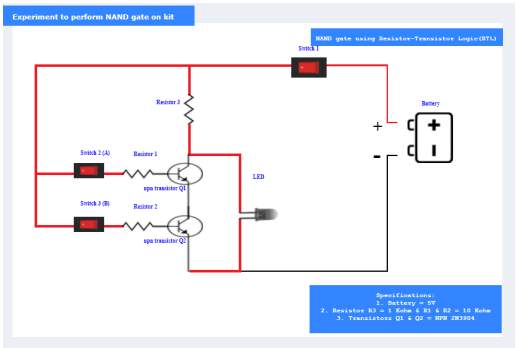


(c) RTL Design

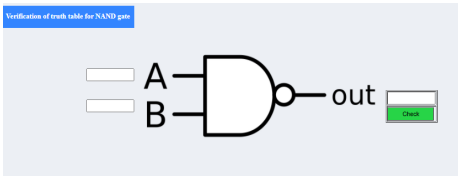
Figure 2.4.2: *NOR gate*

2.5 Procedure

2.5.1 NAND gate



(a) Simulator 1



(b) Simulator 2

Figure 2.5.1: *Simulator for realizing circuit for NAND gate*

Simulator 1

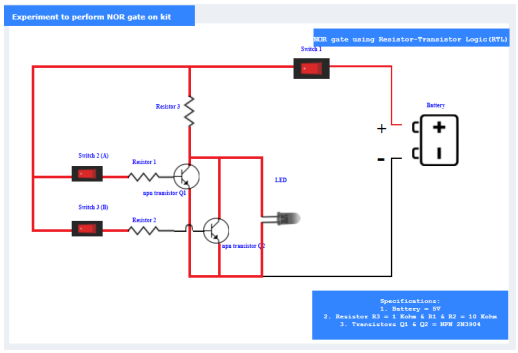
1. Connect the supply(+5V) to the circuit.

2. Press the switches for inputs ”A” and ”B”.
3. The bulb glows if any one or both the switches are OFF else it won’t glow.
4. Repeat step-2 and step-3 for all state of inputs.

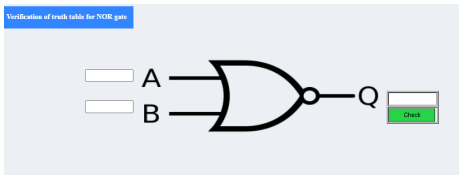
Simulator 2

1. Enter the Boolean input ”A” and ”B”.
2. Enter the Boolean output for your corresponding inputs.
3. Click on ”Check” Button to verify your output.

2.5.2 NOR gate



(a) Simulator 1



(b) Simulator 2

Figure 2.5.2: *Simulator for realizing circuit for NOR gate*

Simulator 1

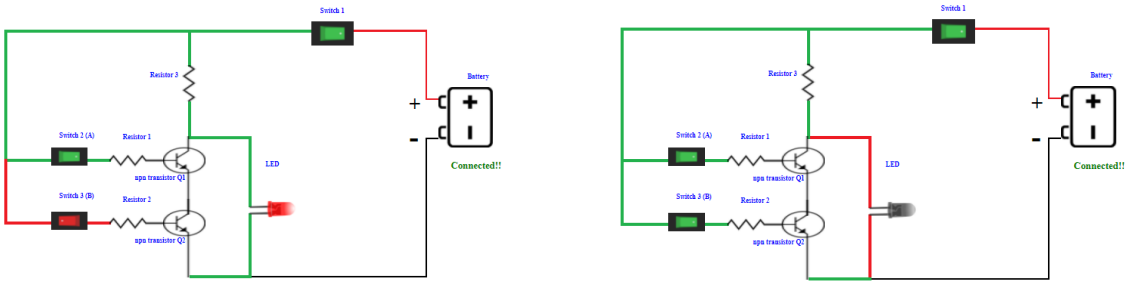
1. Connect the supply(+5V) to the circuit.
2. Press the switches for inputs ”A” and ”B”.
3. The bulb glows if both the switches are OFF else it won’t glow.
4. Repeat step-2 and step-3 for all state of inputs.

Simulator 2

1. Enter the Boolean input ”A” and ”B”.
2. Enter the Boolean output for your corresponding inputs.
3. Click on ”Check” Button to verify your output.

2.6 Observations

2.6.1 NAND gate



(a) Either of the Inputs OFF, LED is ON

(b) Both Inputs ON, LED is OFF

Figure 2.6.1: *Observations for different Input Values*

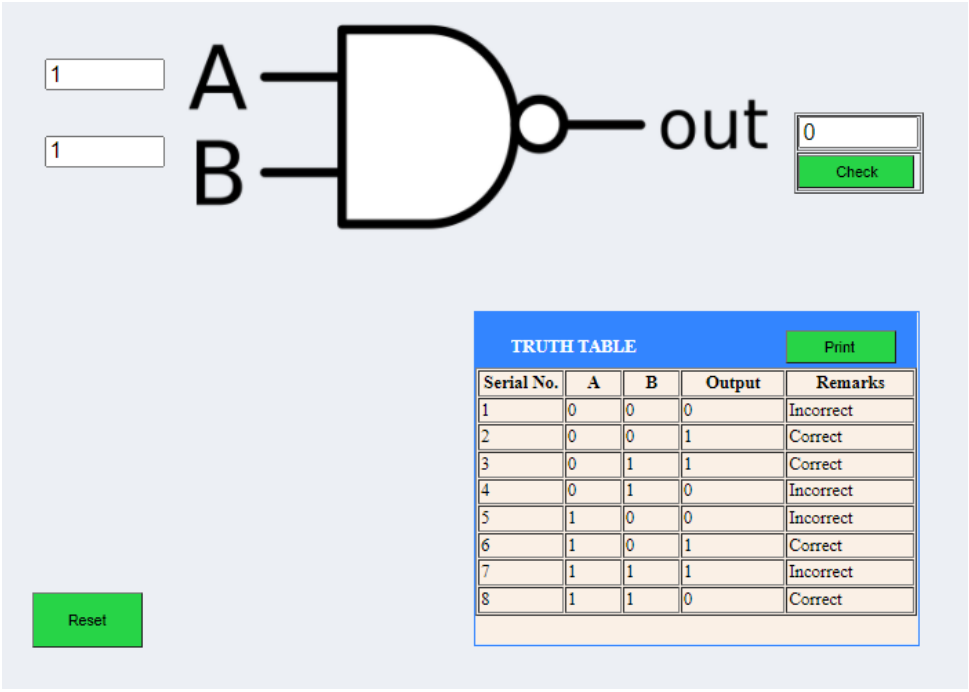
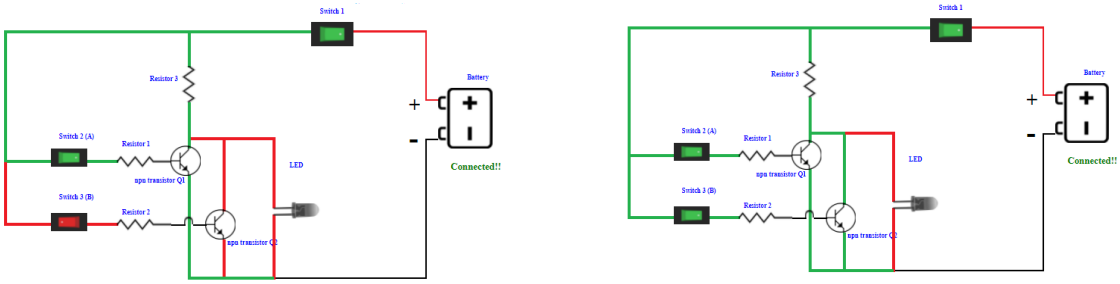


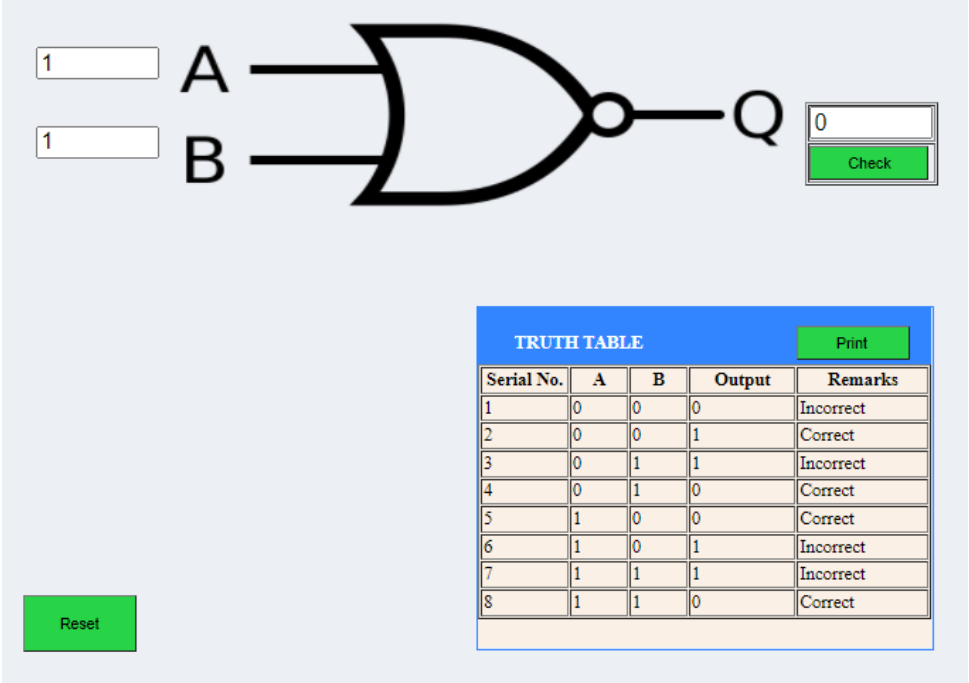
Figure 2.6.2: *Observations for verification of Truth Table of the NAND gate*

2.6.2 NOR gate



(a) Either of the Inputs ON, LED is OFF (b) Both Inputs OFF, LED is ON

Figure 2.6.3: *Observations for different Input Values*



Serial No.	A	B	Output	Remarks
1	0	0	0	Incorrect
2	0	0	1	Correct
3	0	1	1	Incorrect
4	0	1	0	Correct
5	1	0	0	Correct
6	1	0	1	Incorrect
7	1	1	1	Incorrect
8	1	1	0	Correct

Figure 2.6.4: *Observations for verification of Truth Table of the NOR gate*

2.7 Precautions

1. Make the connections when power supply is OFF.
2. Ensure that the connections are tight.
3. Change the status of inputs only when power supply is OFF.

Activity 3

XOR, XNOR gates

3.1 Aim

To verify and interpret the logic and truth table for XOR, XNOR gates using Resistor Transistor Logic (RTL)

3.2 Apparatus

- Kit for realization of gates
- Connecting Leads

3.3 Circuits

3.4 Theory

Logic gates are the basic building blocks of any digital system. Logic gates are electronic circuits having one or more than one input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as:

1. AND gate
2. OR gate
3. NOT gate

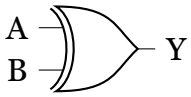
4. NAND gate
5. NOT gate
6. Ex-OR gate
7. Ex-NOR gate

3.4.1 XOR gate

The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both of its two inputs are high. An encircled plus sign ($\oplus$) is used to show the Ex-OR operation.

$$Y = A \oplus B$$

Ex-OR gate is created from AND, NAND and OR gates.The output is high only when both the inputs are different.



(a) Symbol

Input		Output
A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth Table

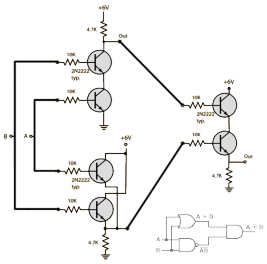

(c) RTL Design

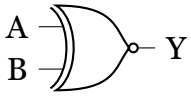
Figure 3.4.1: *XOR gate*

3.4.2 XNOR gate

The 'Exclusive-NOR' gate circuit does the opposite to the EX-OR gate. It will give a low output if either, but not both of its two inputs are high. The symbol is an EX-OR gate with a small circle on the output. The small circle represents inversion.

$$Y = \overline{A \oplus B}$$

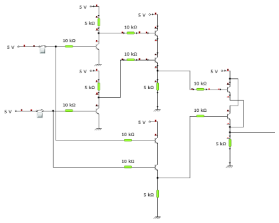
Ex-NOR gate is created from AND, NOT and OR gates.The output is high only when both the inputs are same.



(a) Symbol

Input		Output
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

(b) Truth Table

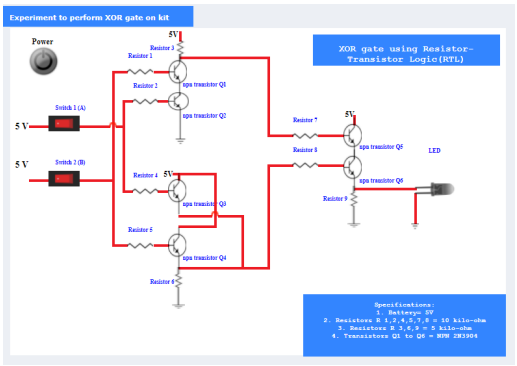


(c) RTL Design

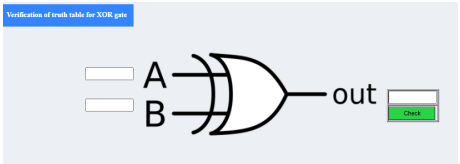
Figure 3.4.2: *XNOR gate*

3.5 Procedure

3.5.1 XOR gate



(a) Simulator 1



(b) Simulator 2

Figure 3.5.1: *Simulator for realizing circuit for XOR gate*

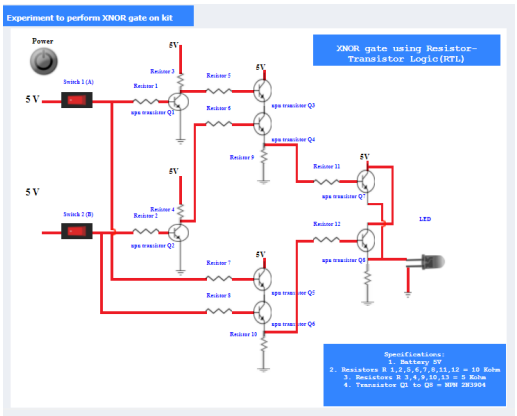
Simulator 1

1. Connect the supply(+5V) to the circuit.
2. Press the switches for inputs ”A” and ”B”.
3. The bulb glows if one of the switches is ON and one of the switches is OFF else it won’t glow.
4. Repeat step-2 and step-3 for all state of inputs.

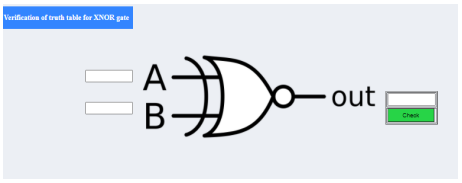
Simulator 2

1. Enter the Boolean input ”A” and ”B”.
2. Enter the Boolean output for your corresponding inputs.
3. Click on ”Check” Button to verify your output.

3.5.2 XNOR gate



(a) Simulator 1



(b) Simulator 2

Figure 3.5.2: *Simulator for realizing circuit for XNOR gate*

Simulator 1

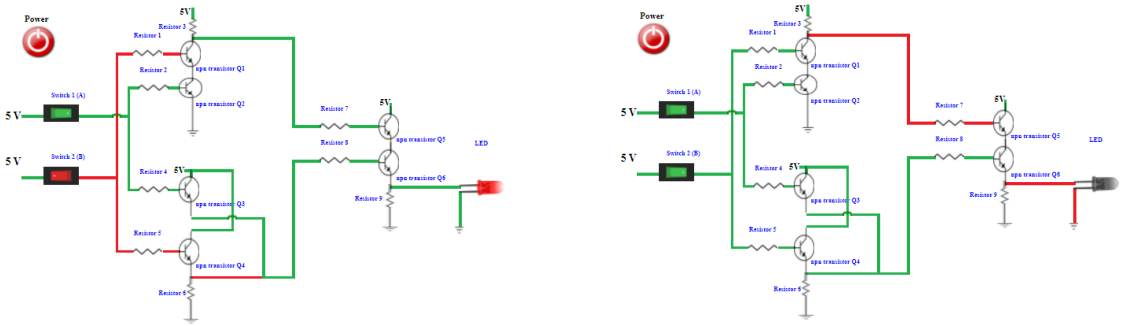
1. Connect the supply(+5V) to the circuit.
2. Press the switches for inputs ”A” and ”B”.
3. The bulb glows if both the switches are ON or if both the switches are OFF else it won’t glow.
4. Repeat step-2 and step-3 for all state of inputs.

Simulator 2

1. Enter the Boolean input ”A” and ”B”.
2. Enter the Boolean output for your corresponding inputs.
3. Click on ”Check” Button to verify your output.

3.6 Observations

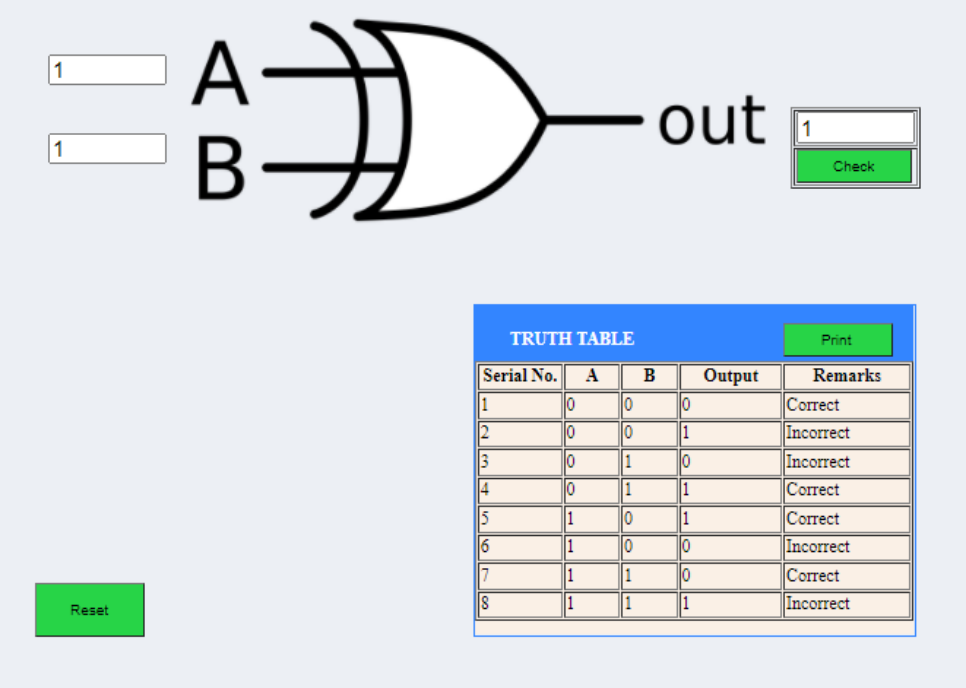
3.6.1 XOR gate



(a) If the Inputs are alternative, LED is ON

(b) If both Inputs are same, LED is OFF

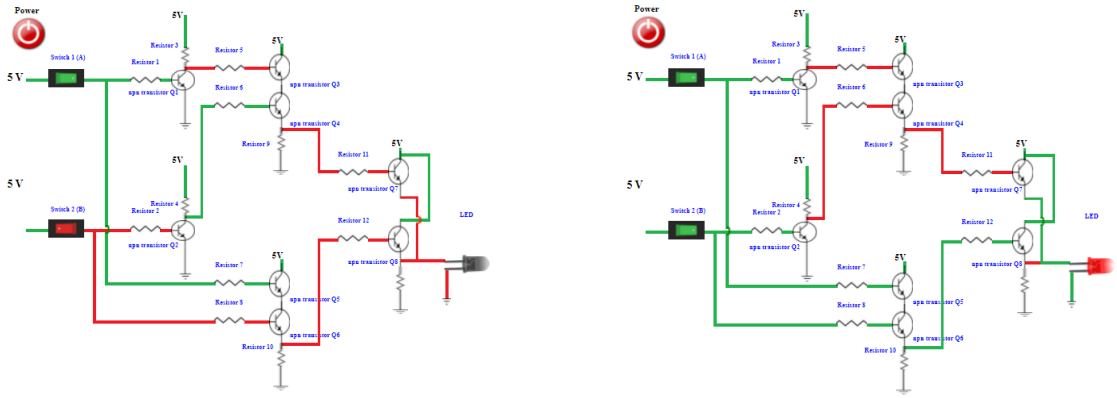
Figure 3.6.1: *Observations for different Input Values*



Serial No.	A	B	Output	Remarks
1	0	0	0	Correct
2	0	0	1	Incorrect
3	0	1	0	Incorrect
4	0	1	1	Correct
5	1	0	1	Correct
6	1	0	0	Incorrect
7	1	1	0	Correct
8	1	1	1	Incorrect

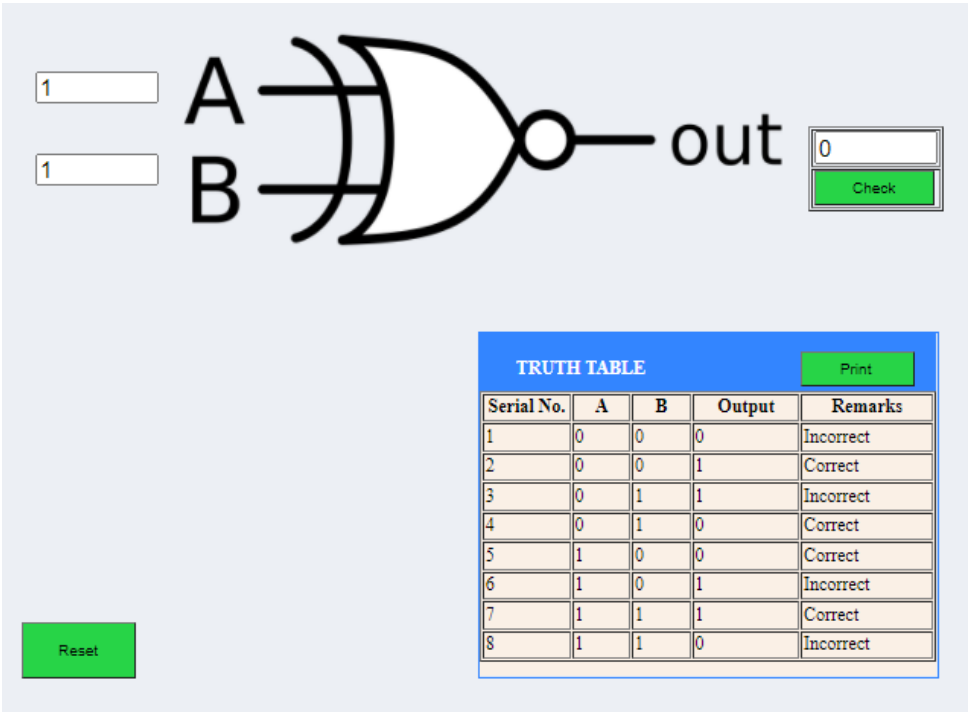
Figure 3.6.2: *Observations for verification of Truth Table of the XOR gate*

3.6.2 XNOR gate



(a) If the Inputs are alternative, LED is OFF (b) If both Inputs are same, LED is ON

Figure 3.6.3: *Observations for different Input Values*



Serial No.	A	B	Output	Remarks
1	0	0	0	Incorrect
2	0	0	1	Correct
3	0	1	1	Incorrect
4	0	1	0	Correct
5	1	0	0	Correct
6	1	0	1	Incorrect
7	1	1	1	Correct
8	1	1	0	Incorrect

Figure 3.6.4: *Observations for verification of Truth Table of the XNOR gate*

3.7 Precautions

1. Make the connections when power supply is OFF.
2. Ensure that the connections are tight.
3. Change the status of inputs only when power supply is OFF.

Activity 4

Deriving other gates from universal gates

4.1 Aim

To implement the logic functions i.e. AND, OR, NOT, Ex-OR, Ex- NOR and a logical expression with the help of NAND and NOR universal gates respectively.

4.2 Apparatus

- Kit for realization of NAND and NOR gates
- Connecting Leads

4.3 Theory

Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output. There are seven logic gates. When all the input combinations of a logic gate are written in a series and their corrresponding outputs written along them, then this input/ output combination is called Truth Table.

AND, OR, NOT, XOR and XNOR gates can be derived from NAND and NOR

gates by converting their respective equations to NAND or NOR form, using he following 3 laws of Boolean Algebra (and their duals according to Duality Principle):

1. Involution Law: $\overline{(\overline{X})} = X$
2. Idempotent Law: $X + X = X$
3. De-Morgan's Law: $\overline{X + Y} = \overline{X}.\overline{Y}$

4.3.1 NAND gate

NAND gate is actually a combination of two logic gates i.e. AND gate followed by NOT gate. So its output is complement of the output of an AND gate.This gate can have minimum two inputs. By using only NAND gates, we can realize all logic functions: AND, OR, NOT, Ex-OR, Ex-NOR, NOR. So this gate is also called as universal gate. The expression for NAND gate is:

$$Y = \overline{A.B}$$

4.3.2 NOR gate

NOR gate is actually a combination of two logic gates: OR gate followed by NOT gate. So its output is complement of the output of an OR gate.This gate can have minimum two inputs, output is always one. By using only NOR gates, we can realize all logic functions: AND, OR, NOT, Ex-OR, Ex-NOR, NAND. So this gate is also called universal gate. The expression for NOR gate is:

$$Y = \overline{A + B}$$

4.4 Circuits

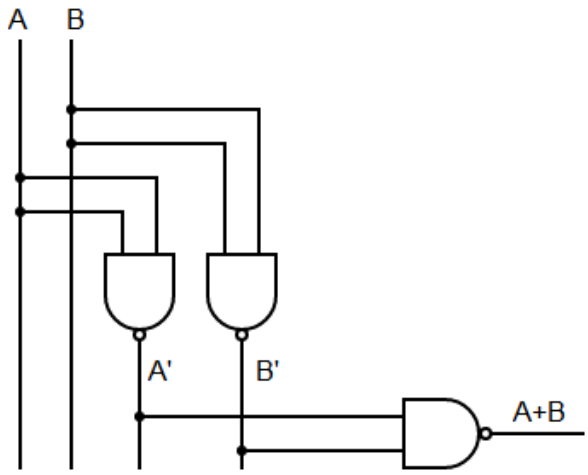
4.4.1 NAND gate as Universal gate

NAND gates as OR gate

From DeMorgan's theorems:

$$(A.B)' = A' + B'$$
$$(A'.B')' = A'' + B''$$
$$= A + B$$

So, give the inverted inputs to a NAND gate, obtain OR operation at output.



(a) OR gate using NAND gates

Input		Output
A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(b) Truth Table for OR

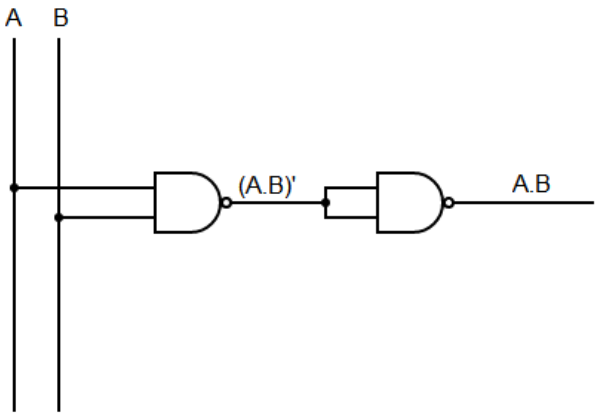
Figure 4.4.1: *OR Gate*

NAND gates as AND gate

From DeMorgan's theorems:

$$Y = ((A.B)')'$$
$$Y = (A.B)$$

A NAND produces complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate.



(a) AND gate using NAND gates

Input		Output
A	B	$Y = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

(b) Truth Table for AND

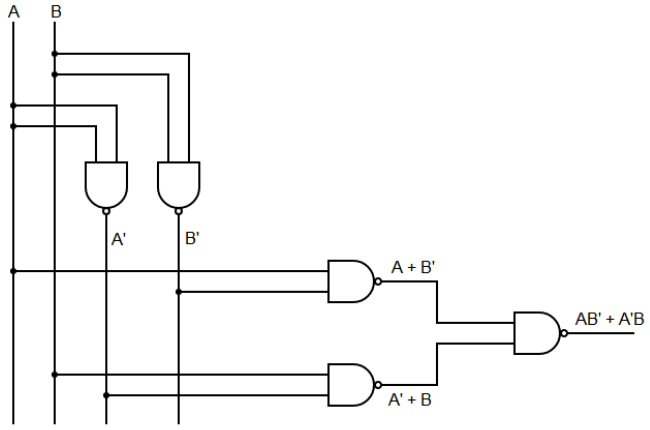
Figure 4.4.2: *AND Gate*

NAND gates as XOR gate

From DeMorgan's theorems:

$$\begin{aligned} Y &= A \oplus B \\ &= \overline{A}B + A\overline{B} \\ &= \overline{(\overline{\overline{A}B + A\overline{B}})} \\ &= \overline{((\overline{\overline{A}B}).(\overline{A\overline{B}}))} \\ &= \overline{(A + \overline{B}).(\overline{A} + B)} \end{aligned}$$

This can be achieved with the logic diagram shown in Figure 4.4.3a.



(a) XOR gate using NAND gates

Input		Output
A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth Table for XOR

Figure 4.4.3: *XOR Gate*

NAND gates as XNOR gate

From DeMorgan's theorems:

$$\begin{aligned} Y &= \overline{A \oplus B} \\ &= \overline{(\overline{A}B + A\overline{B})} \\ &= \overline{\overline{(\overline{A}B + A\overline{B})}} \\ &= \overline{\overline{(\overline{\overline{A}B}).(\overline{A\overline{B}})}} \\ &= \overline{\overline{(A + \overline{B}).(\overline{A} + B)}} \\ &= \overline{(XOR - gate - implemented - using - NAND - Gates)} \end{aligned}$$

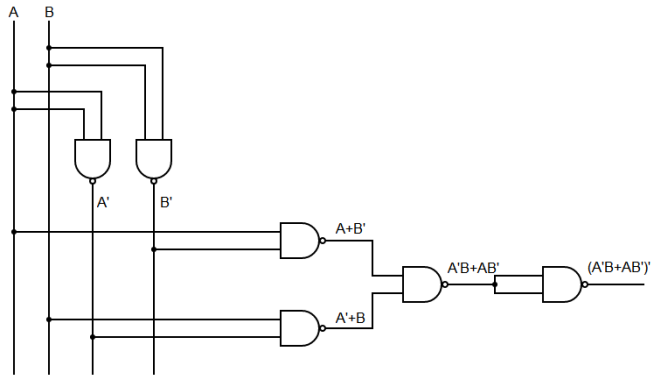
The output of a two input XNOR gate is shown by: $Y = \overline{\overline{A}B + A\overline{B}}$. This can be achieved with the logic diagram shown in Figure 4.4.4a.

4.4.2 NOR gate as Universal gate

NOR gates as OR gate

From DeMorgan's theorems:

$$\begin{aligned} Y &= \overline{\overline{(A + B)}} \\ &= (A + B) \end{aligned}$$



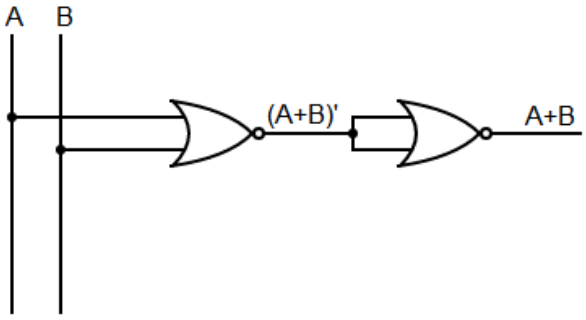
(a) XNOR gate using NAND gates

Input		Output
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

(b) Truth Table for XNOR

Figure 4.4.4: *XNOR Gate*

A NOR produces complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.



(a) OR gate using NOR gates

Input		Output
A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(b) Truth Table for OR

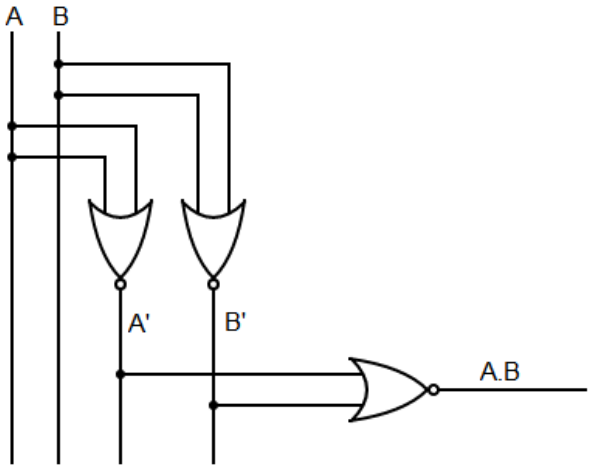
Figure 4.4.5: *OR Gate*

NOR gates as AND gate

From DeMorgan's theorems:

$$\begin{aligned} Y &= A.B \\ &= \overline{(\overline{A.B})} \\ &= \overline{(\overline{A} + \overline{B})} \end{aligned}$$

So, give the inverted inputs to a NOR gate, obtain AND operation at output.



(a) AND gate using NAND gates

Input		Output
A	B	$Y = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

(b) Truth Table for AND

Figure 4.4.6: *AND Gate*

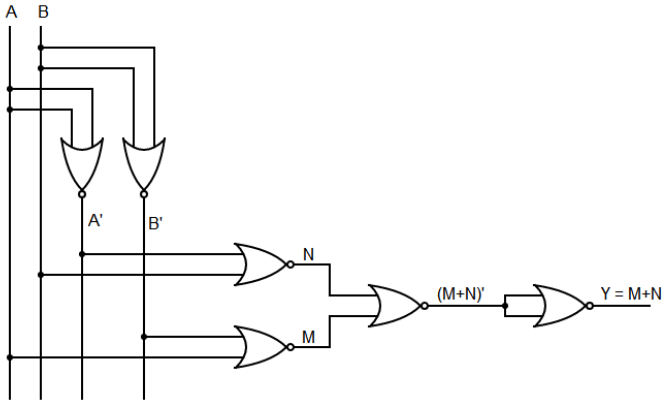
NOR gates as XOR gate

From DeMorgan's theorems:

$$\begin{aligned} Y &= A \oplus B \\ &= \overline{A}B + A\overline{B} \\ &= \overline{(\overline{\overline{A}B + A\overline{B}})} \\ &= \overline{(\overline{(\overline{A}B)}.\overline{(A\overline{B})})} \\ &= \overline{(A + \overline{B}).(\overline{A} + B)} \\ &= \overline{(A + \overline{B})} + \overline{(\overline{A} + B)} \\ &= M + N \\ &= \overline{(\overline{M + N})} \end{aligned}$$

$$M = \overline{(A + \overline{B})}, N = \overline{(\overline{A} + B)}$$

The output of a two input Ex-OR gate is shown by: $Y = \overline{A}B + A\overline{B}$. This can be achieved with the logic diagram shown in Figure 4.4.7a.



(a) XOR gate using NOR gates

Input		Output
A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth Table for XOR

Figure 4.4.7: *XOR Gate*

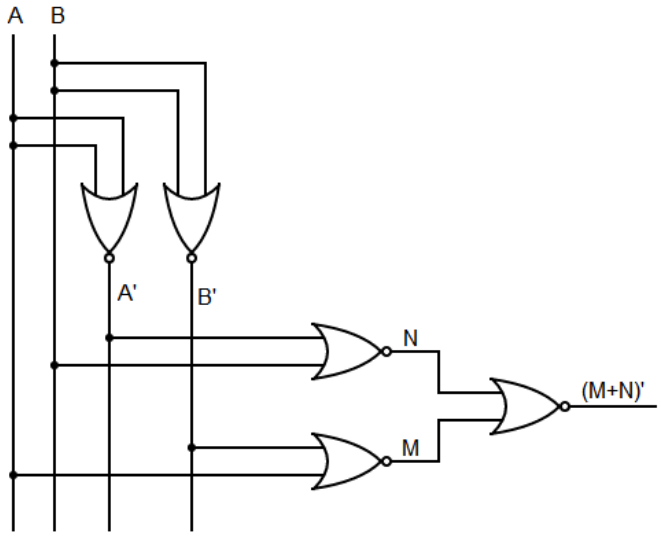
NOR gates as XNOR gate

From DeMorgan's theorems:

$$\begin{aligned} Y &= \overline{A \oplus B} \\ &= \overline{\overline{A}B + A\overline{B}} \\ &= \overline{\overline{(\overline{A}B + A\overline{B})}} \\ &= \overline{\overline{(\overline{\overline{A}B}).(\overline{A\overline{B}})}} \\ &= \overline{\overline{(A + \overline{B}).(\overline{A} + B)}} \\ &= \overline{\overline{(A + \overline{B})} + \overline{(\overline{A} + B)}} \\ &= \overline{M + N} \end{aligned}$$

$$M = \overline{(A + \overline{B})}, N = \overline{(\overline{A} + B)}$$

The output of a two input Ex-NOR gate is shown by: $Y = \overline{\overline{A}B + A\overline{B}}$. This can be achieved with the logic diagram shown in Figure 4.4.8a.



(a) XNOR gate using NOR gates

Input		Output
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

(b) Truth Table for XNOR

Figure 4.4.8: *XOR Gate*

4.5 Procedure

1. Check the components for their working (NAND or NOR kit).
2. Insert the wire-leads into the appropriate places on the kit as per the cicuits shown.
3. For output, use the LED on the provided kit.
4. Provide the input data in the input switches and observe the output on the LED.
5. Verify the truth tables of various gates.

4.6 Precautions

1. The leads must be connected properly.
2. Wires must be connected during power supply being off only.
3. Change the input switches only when supply is off.

4.7 Result

NOT, AND, OR, XOR, XNOR gates can be realized using NAND and NOR gates.

Activity 5

SR Flip Flop

5.1 Aim

To verify Truth Table of SR flip-flop and analyse its circuit with the help of LED's output.

5.2 Apparatus

- Kit for realization of NAND, NOR gates.
- Connecting Leads

5.3 Theory

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems.

There are 4 types of flip-flops:

1. R-S flip flop
2. D flip flop
3. J-K flip flop
4. T flip flop

The basic NAND gate RS flip flop circuit is used to store the data and thus provides feedback from both of its outputs again back to its inputs. The SR flip flop actually has three inputs: SET, RESET and clock pulse.

5.4 Circuits

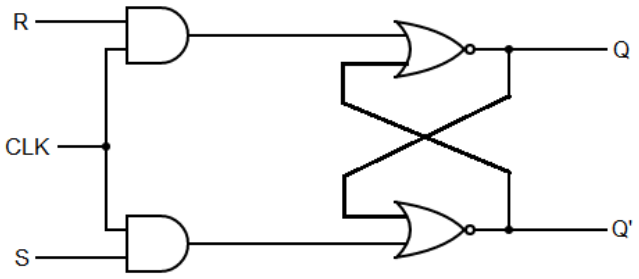


Figure 5.4.1: *SR Flip Flop Circuit Diagram*

5.5 Procedure

1. Connect the 1^{st} lead to the S terminal of SR flip flop.
2. Connect the 2^{nd} lead to the R terminal of SR flip flop.
3. Connect clock to CLK terminal of SR flip flop
4. Connect the Q terminal to one of the output LEDs.
5. Connect the $\overline{Q}$ terminal to other of the output LEDs.
6. Turn the power supply on.
7. Set the input and click on CLK PULSE button.
8. Observe the output.
9. Repeat the above three steps for different inputs and note them down.

5.6 Observation

We observe the following values for the Output of SR flip flop:

INPUTS			OUTPUT	STATE
CLK	S	R	Q	
X	0	0	No change	Previous
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	-	Forbidden

Figure 5.6.1: *SR Flip Flop Truth Table*

5.7 Precautions

1. The leads must be connected properly.
2. Wires must be connected during power supply being off only.
3. Change the input switches only when supply is off.

5.8 Result

The Truth Table of SR Flip Flop is verified.

Activity 6

JK Flip Flop

6.1 Aim

To verify Truth Table of JK flip-flop and analyse its circuit with the help of LED's output.

6.2 Apparatus

- Kit for realization of NAND, NOR gates.
- Connecting Leads

6.3 Theory

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems.

There are 4 types of flip-flops:

1. R-S flip flop
2. D flip flop
3. J-K flip flop
4. T flip flop

The basic NAND gate RS flip flop circuit is used to store the data and thus provides feedback from both of its outputs again back to its inputs. The SR flip flop actually has three inputs: SET, RESET and clock pulse. In a RS flip-flop the input $R = S = 1$ leads to an indeterminate output. The RS flip-flop circuit may be re-joined if both inputs are 1 than also the outputs are complement of each other. This type of flip-flop is known as J-K flip flop

6.4 Circuits

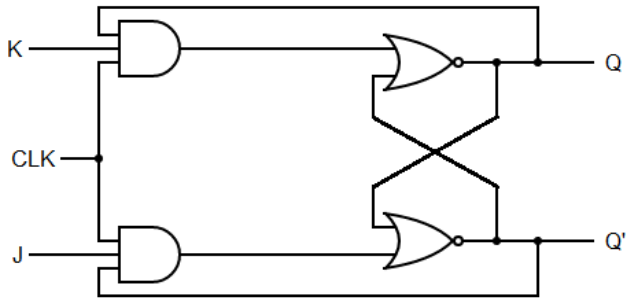


Figure 6.4.1: *JK Flip Flop Circuit Diagram*

6.5 Procedure

1. Connect the 1^{st} lead to the K terminal of JK flip flop.
2. Connect the 2^{nd} lead to the J terminal of JK flip flop.
3. Connect clock to CLK terminal of JK flip flop
4. Connect the Q terminal to one of the output LEDs.
5. Connect the $\overline{Q}$ terminal to other of the output LEDs.
6. Turn the power supply on.
7. Set the input and click on CLK PULSE button.
8. Observe the output.
9. Repeat the above three steps for different inputs and note them down.

6.6 Observation

We observe the following values for the Output of JK flip flop:

Trigger	Inputs		Output				Inference
			Present State		Next State		
CLK	J	K	Q	$\overline{Q}$	Q	$\overline{Q}$	
X	X	X	-		-		Latched
↑	0	0	0	1	0	1	No change
↑			1	0	1	0	
↑	0	1	0	1	0	1	Reset
↑			1	0	0	1	
↑	1	0	0	1	1	0	Set
↑			1	0	1	0	
↑	1	1	0	1	1	0	Toggles
↑			1	0	0	1	

Figure 6.6.1: *JK Flip Flop Truth Table*

6.7 Precautions

1. The leads must be connected properly.
2. Wires must be connected during power supply being off only.
3. Change the input switches only when supply is off.

6.8 Result

The Truth Table of JK Flip Flop is verified.

Bibliography

[1] IIT Roorkee Virtual Labs
(https://de-iitr.vlabs.ac.in/)

[2] Desmos
(https://www.desmos.com/calculator)

[3] M. Morris Mano, Michael D Ciletti, "Digital Design", Pearson, 4th ed. 2008
(https://www.pdfdrive.com/digital-logic-and-computer-design-by-m-morris
html)

[4] Concepts of Physics 2 - HC Verma
(https:
//eg4.nic.in/jnv/DFILES/EBOOKS/IR/conceptsofphysics2.pdf)

[5] Basic Electrical Engineering - SK Sahdev
(https:
//book-drive.com/basic-electrical-engineering-pearson/)

Anhat Singh