

# *Switching Theory*

*And*

# *Logic Design*

**ETEC 205**

Faculty name: Anubha Goel

Student name: Syeda Reeha Quasar

Roll No.: 14114802719

Semester: 3C7 (3rd semester)



Maharaja Agrasen Institute of Technology, PSP area, Sector – 22,  
Rohini, New Delhi – 110085

(Affiliated to Guru Gobind Singh Indraprastha University, New Delhi)

# Index

<b>Exp. no</b>	<b>Experiment Name</b>	<b>Date of performance</b>	<b>Date of checking</b>	<b>Marks</b>	<b>Signature</b>
1.	Realize all gates by verifying their truth tables.	31 – 08 - 2020	31 – 08 - 2020		
2.	To verify the truth tables of all logical gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) using NAND gate only.	02 – 09 - 2020	02 – 09 - 2020		
3.	To verify the truth tables of all logical gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) using NOR gate only.	09 – 09 - 2020	09 – 09 - 2020		
4.	To realize the circuit for Half Adder and Full Adder using logic gates.	30 – 09 - 2020	30 – 09 - 2020		
5.	To realize the circuit for Half Subtractor and Full Subtractor using logic gates.	07 – 10 - 2020	07 – 10 - 2020		
6	To realize priority encoder using basic Gates.	21 – 10 - 2020	21 – 10 - 2020		
7.	To realize binary to gray and gray to binary code converter.	28 – 10 - 2020	28 – 10 - 2020		
8.	To realize 2 bit Magnitude Comparator.	11 – 11 - 2020	11 – 11 - 2020		
9.	To realize 4-Bit Binary to BCD Convertor.	17 – 11 - 2020	17 – 11 - 2020		
10.	To realize Multiplexer and Demultiplexer using only NAND gates.	25 – 11 - 2020	25 – 11 - 2020		
11.	To realize J-K flip flop using logic gates or by using kit.	02 – 12 - 2020	02 – 12 - 2020		

# EXPERIMENT - 1

Switching Theory and Logic Design (STLD)

## Aim

Realize all gates by verifying their truth tables.

Syeda Reeha Quasar  
14114802719  
3C7

# EXPERIMENT – 1

**AIM:** Realize all gates by verifying their truth tables.

**HARDWARE REQUIRED:** Power supply/ Voltage supply, Bread Board, Resistors, LEDs, Connecting Wires, Integrated Chips ICs (7404, 7408, 7432, 7486, 7400, 7402, 74266)

**SOFTWARE REQUIRED:**

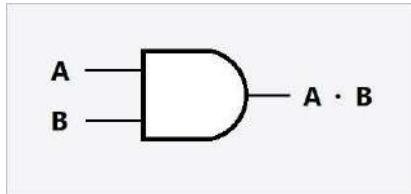
Software simulator (MULTISIM) - [www.multisim.com](http://www.multisim.com) (free software)

Stimulating schematic models of desired circuits

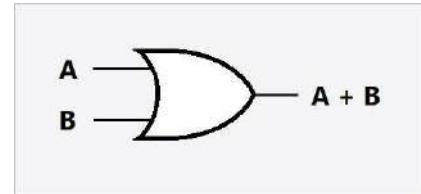
Components used - Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

**CIRCUIT:**

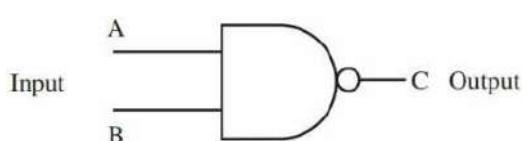
AND GATE



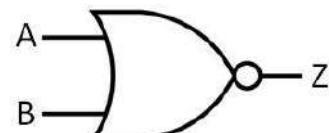
OR GATE



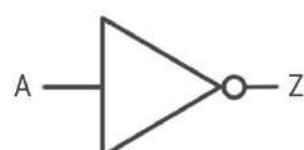
NAND GATE

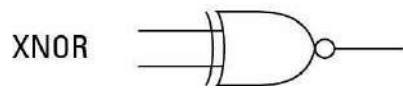
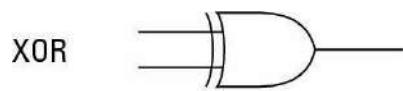


NOR GATE



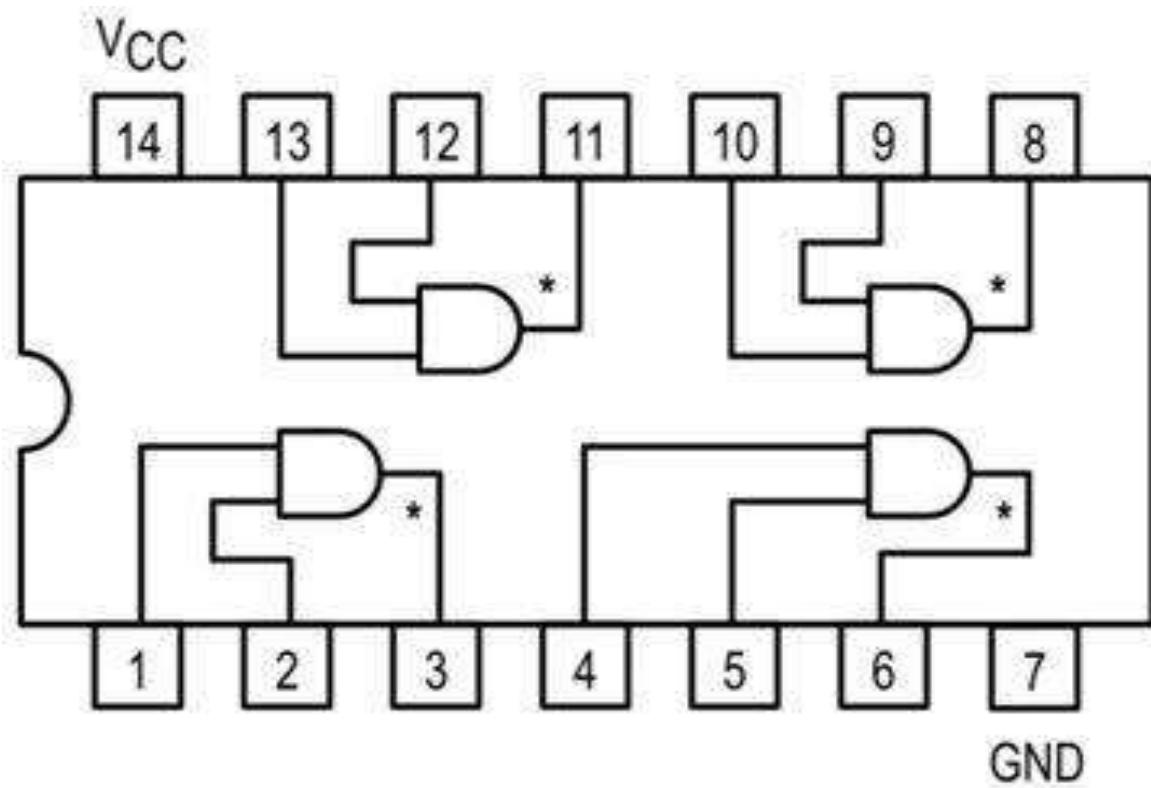
NOT GATE



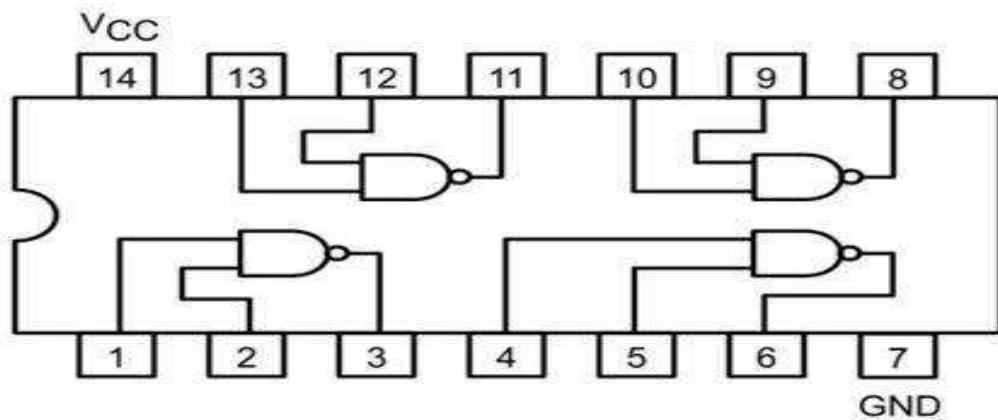


### PIN - DIAGRAM:

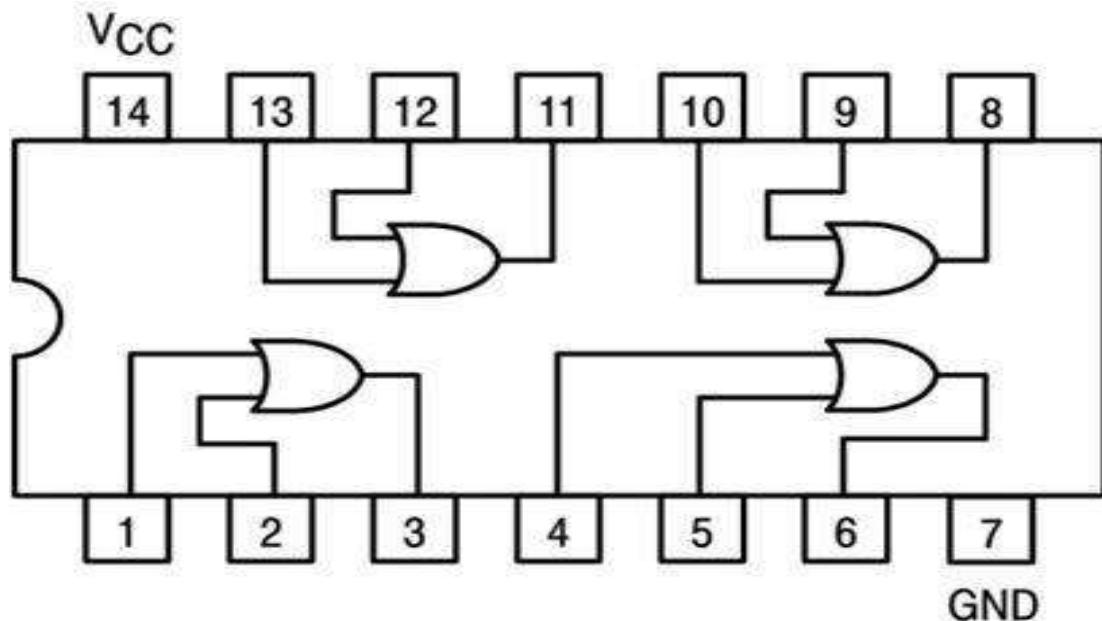
#### AND GATE



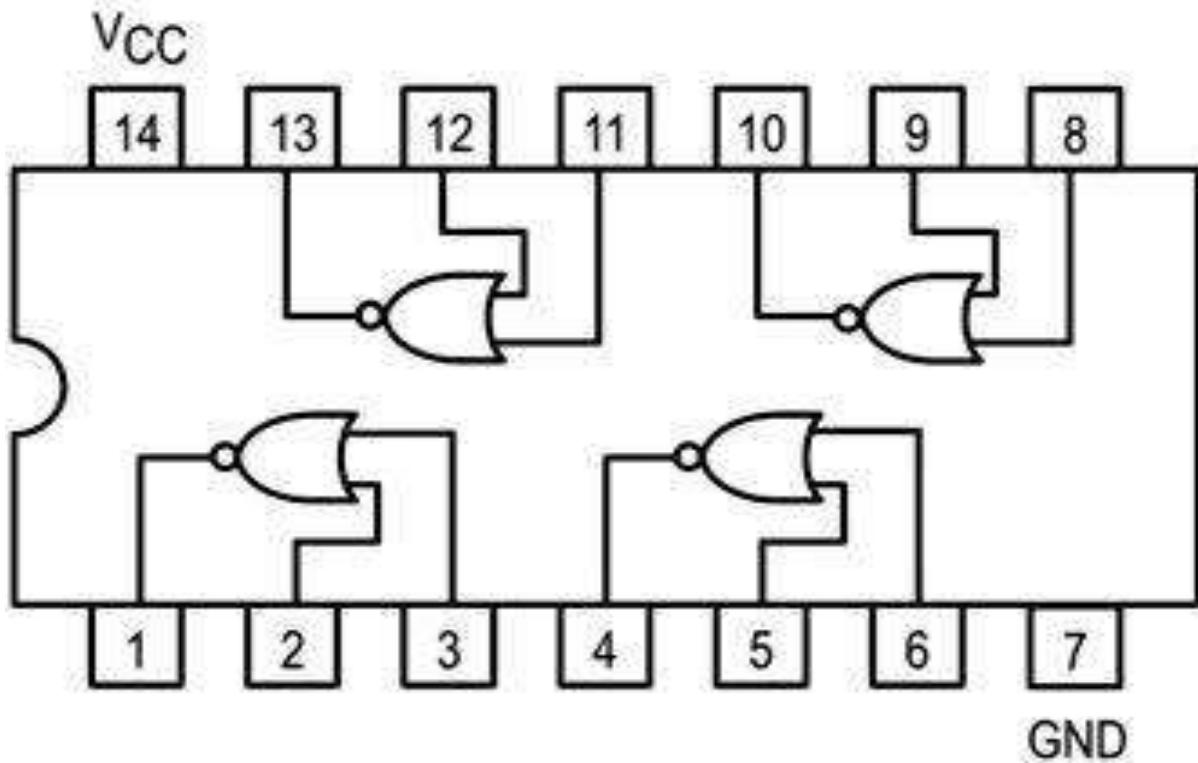
#### NAND GATE



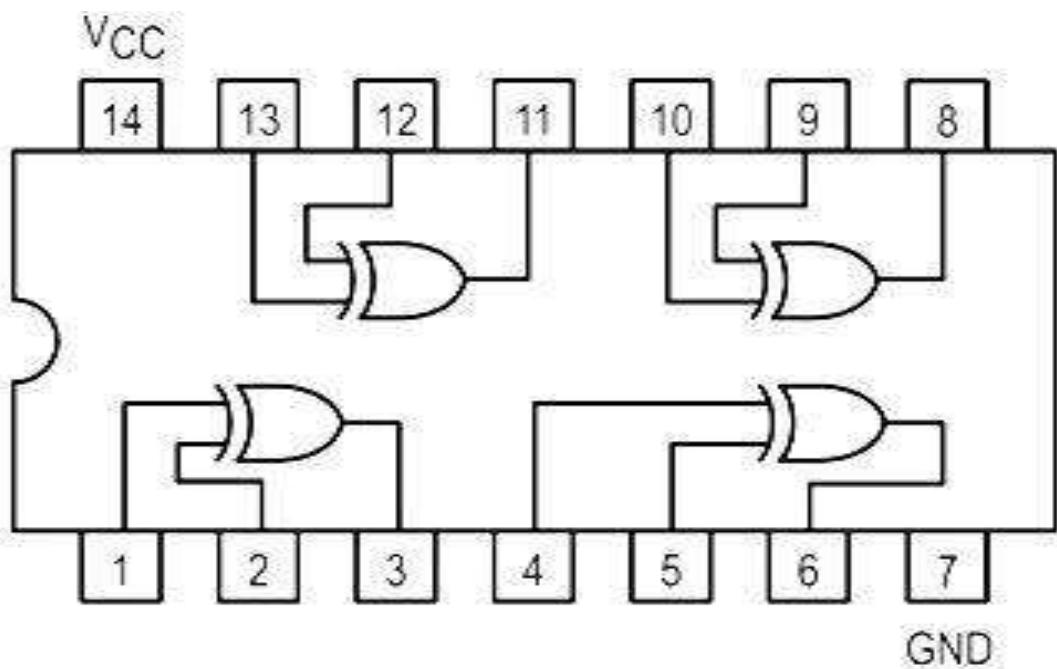
### OR GATE



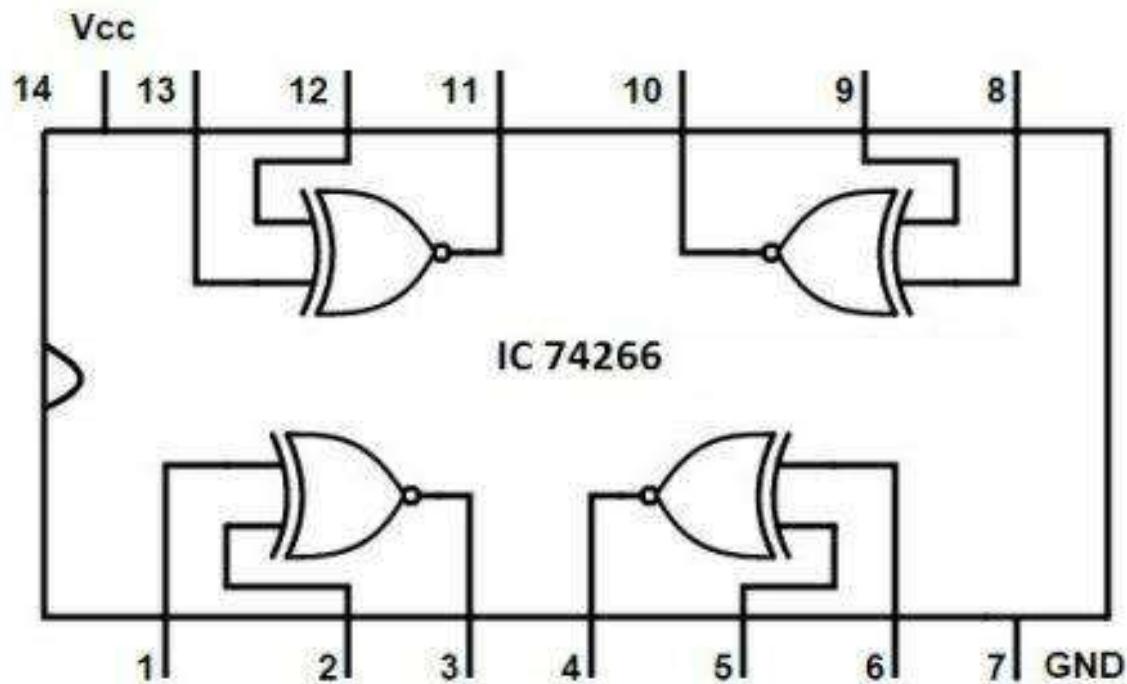
### NOR GATE



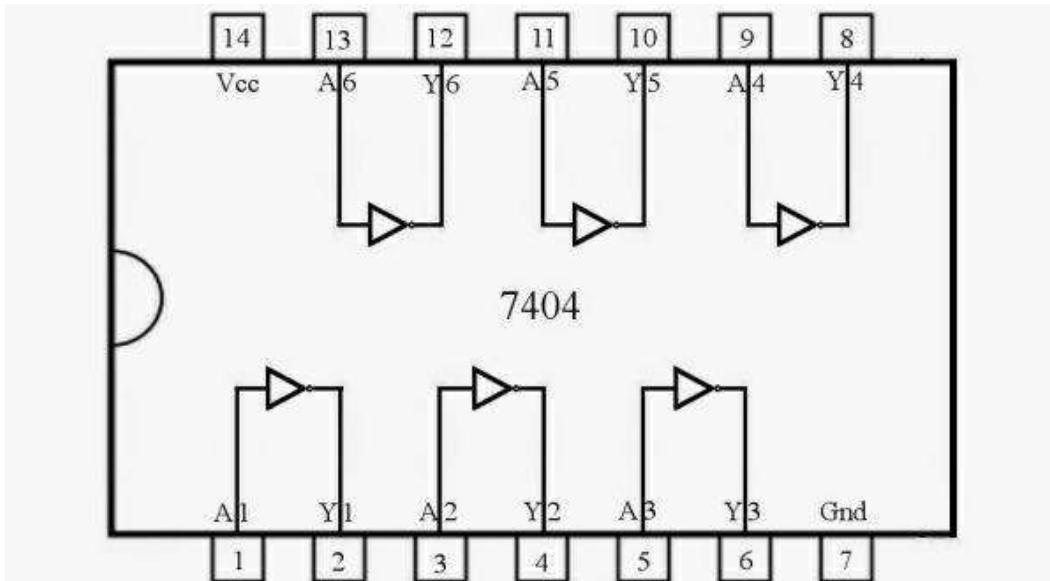
## XOR GATE



## XNOR GATE



## NOT GATE



## TRUTH TABLES

AND Truth Table		
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

NAND Truth Table		
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NOR Truth Table		
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

NOT Truth Table	
A	Q
0	1
1	0

XOR Truth Table		
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

XNOR Truth Table		
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Or Truth Table		
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

**THEORY:** Logic gates are electronic circuits which perform logical operations on one or more inputs to produce a signal output. There are 7 logic gates. These include the AND, NAND, OR, NOR, XOR, XNOR and NOT.

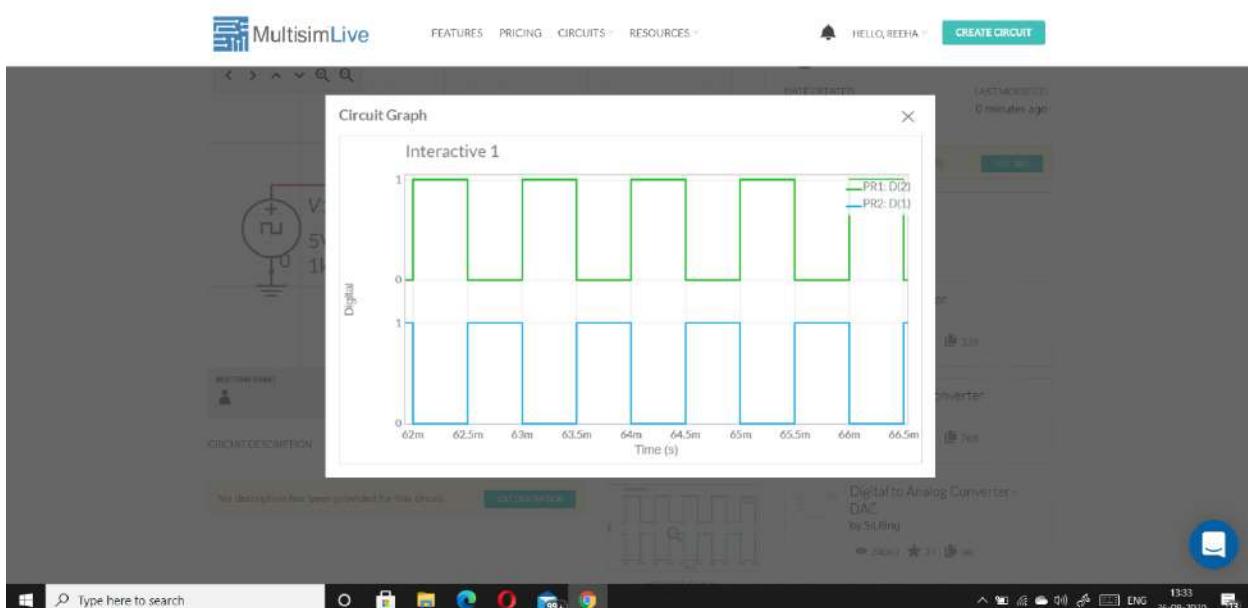
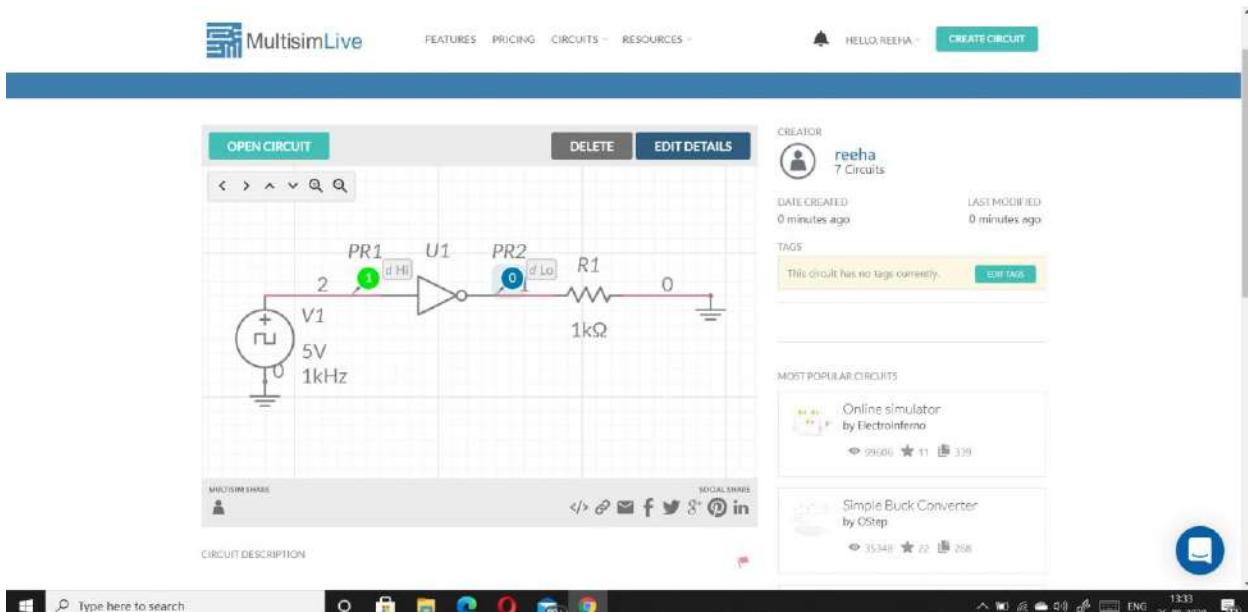
#### PROCEDURE (MULTISIM):

- Make the circuits shown as shown in the figures.
- Select the required components (gates, resistor, voltage sources (Clock Voltage) and ground symbols from the tool bar on the left.
- Ground both the voltage sources (clock Voltages) and then connect them to the input terminal of the gate.
- Connect the output terminal to 1 k ohm resistor and ground it.

#### Precautions:

- Power supply should not exceed 5V.
- All the connections should be tight.
- Components should be tested before the practical.

## NOT GATE



## XNOR GATE

MultisimLive

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA CREATE CIRCUIT

### XNOR Gate

Favorite 0 Copy 0 Views 1

OPEN CIRCUIT DELETE EDIT DETAILS

PR1(d Lo) 1 U1 PR3(d Lo) 3 R1 1kΩ 0

V1 5V 1kHz V2 5V 2.00kHz

MULTISIM SHARE SOCIAL SHARE

DATE CREATED 2 minutes ago LAST MODIFIED 2 minutes ago

TAGS This circuit has no tags currently. EDIT TAGS

MOST POPULAR CIRCUITS

Online simulator by Electroinfomo 99666 ★ 11 339

Simple Buck Converter by CStep

Waiting for faye.gelstream.io...

Type here to search

MultisimLive

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA CREATE CIRCUIT

### Circuit Graph

Interactive 1

Digital

Time (s)

PR1: D(1)  
PR2: D(2)  
PR3: D(3)

There are currently no comments

Waiting for faye.gelstream.io...

Type here to search

## XOR GATE

MultisimLive

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA! CREATE CIRCUIT

OPEN CIRCUIT DELETE EDIT DETAILS

PR1 d Lo 3 U1 PR3 d Hi 1 R1 1kΩ 0

V1 5V 1kHz V2 5V 1.65kHz

MULTISIM SHARE CIRCUIT DESCRIPTION

SOCIAL SHARE

Creator: reeha 7 Circuits Date Created: 5 minutes ago Last Modified: 5 minutes ago Tags: This circuit has no tags currently. Edit tags

MOST POPULAR CIRCUITS

Online simulator by ElectroInferno 99608 11 339

Simple Buck Converter by OStep 35348 22 268

Type here to search

MultisimLive

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA! CREATE CIRCUIT

OPEN CIRCUIT DELETE EDIT DETAILS

Circuit Graph

Interactive 1

Digital

Time (s)

Legend: PR1 D(3) (Green), PR2 D(2) (Blue), PR3 D(1) (Red)

Waiting for faye.getstream.io...

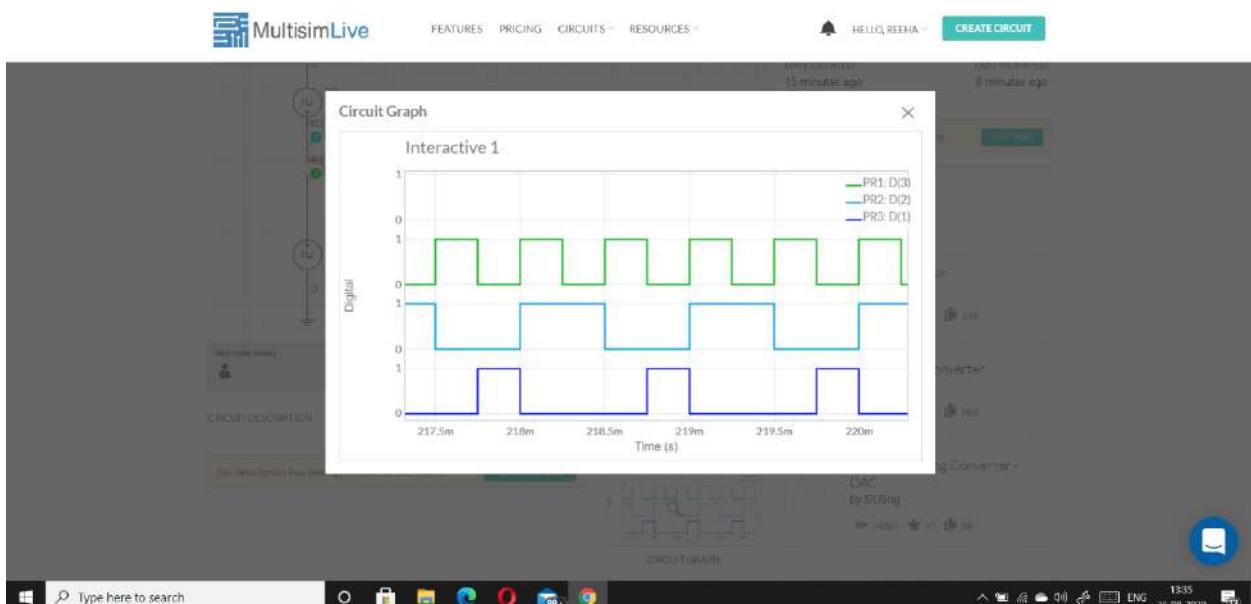
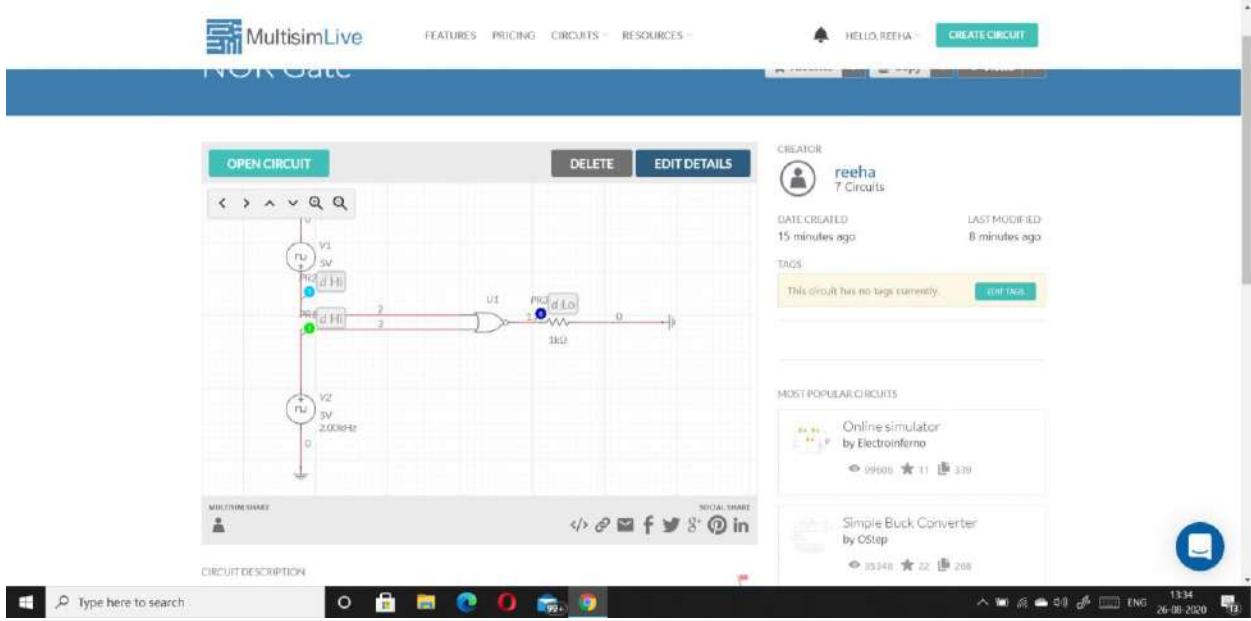
Type here to search

This description has been provided for this circuit.

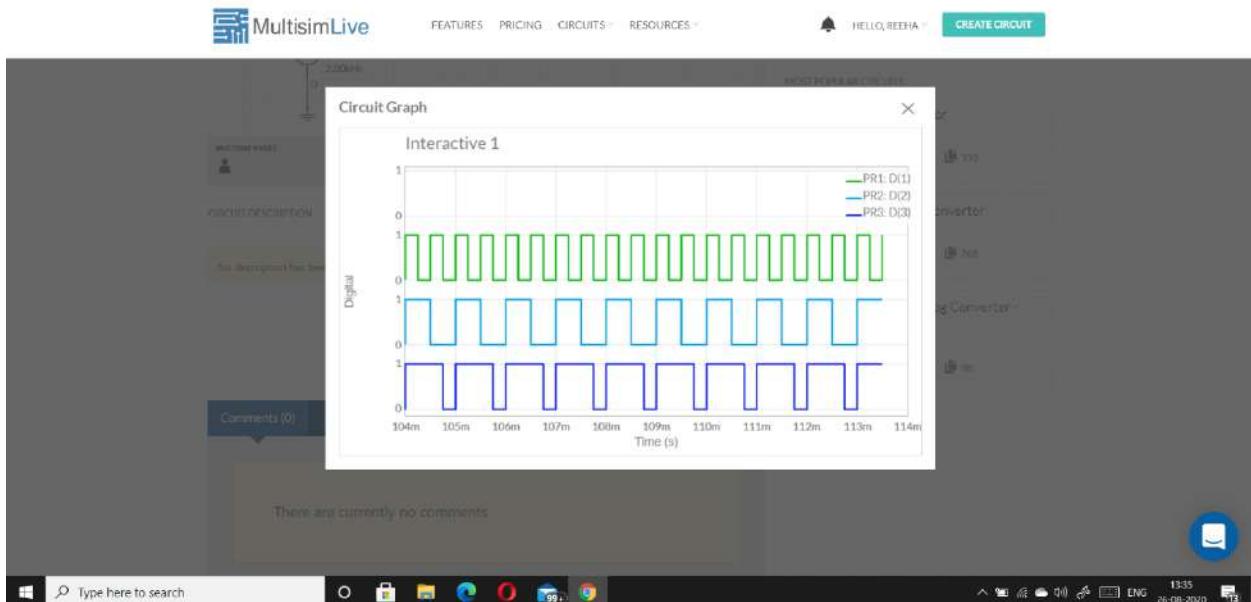
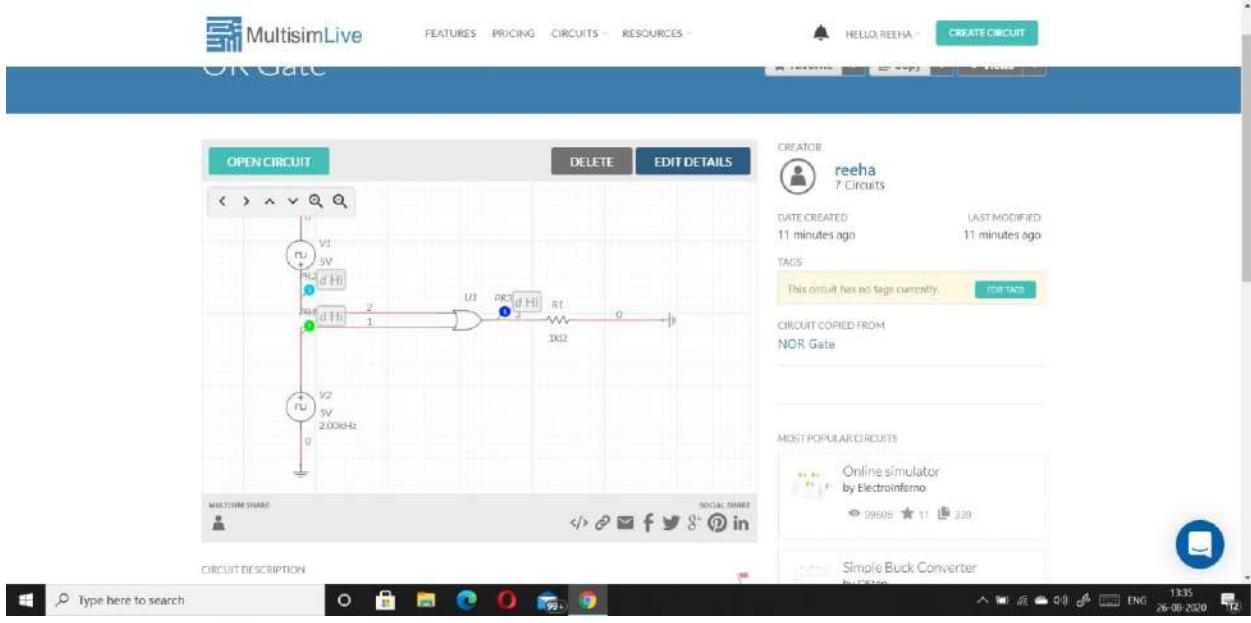
Last Modified: 5 minutes ago

Digital to Analog Converter - DAC by SLSim 333 26 1334 26-08-2020

## NOR GATE



## OR GATE



## NAND GATE

MultisimLive

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA CREATE CIRCUIT

NAND Gate

OPEN CIRCUIT DELETE EDIT DETAILS

PR1 d Hi V1 5V 1kHz

PR2 d Hi V2 5V 1.5kHz

U1

PR3 d Lo R1 1kΩ

0

Creator: reeha 7 Circuits

Date Created: 14 minutes ago Last Modified: 14 minutes ago

Tags: This circuit has no tags currently. EDIT TAGS

MOST POPULAR CIRCUITS

Online simulator by Electrofimo

Simple Buck Converter by CStep

Type here to search

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA CREATE CIRCUIT

Circuit Graph

Interactive 1

Digital

PR1: D(1)

PR2: D(2)

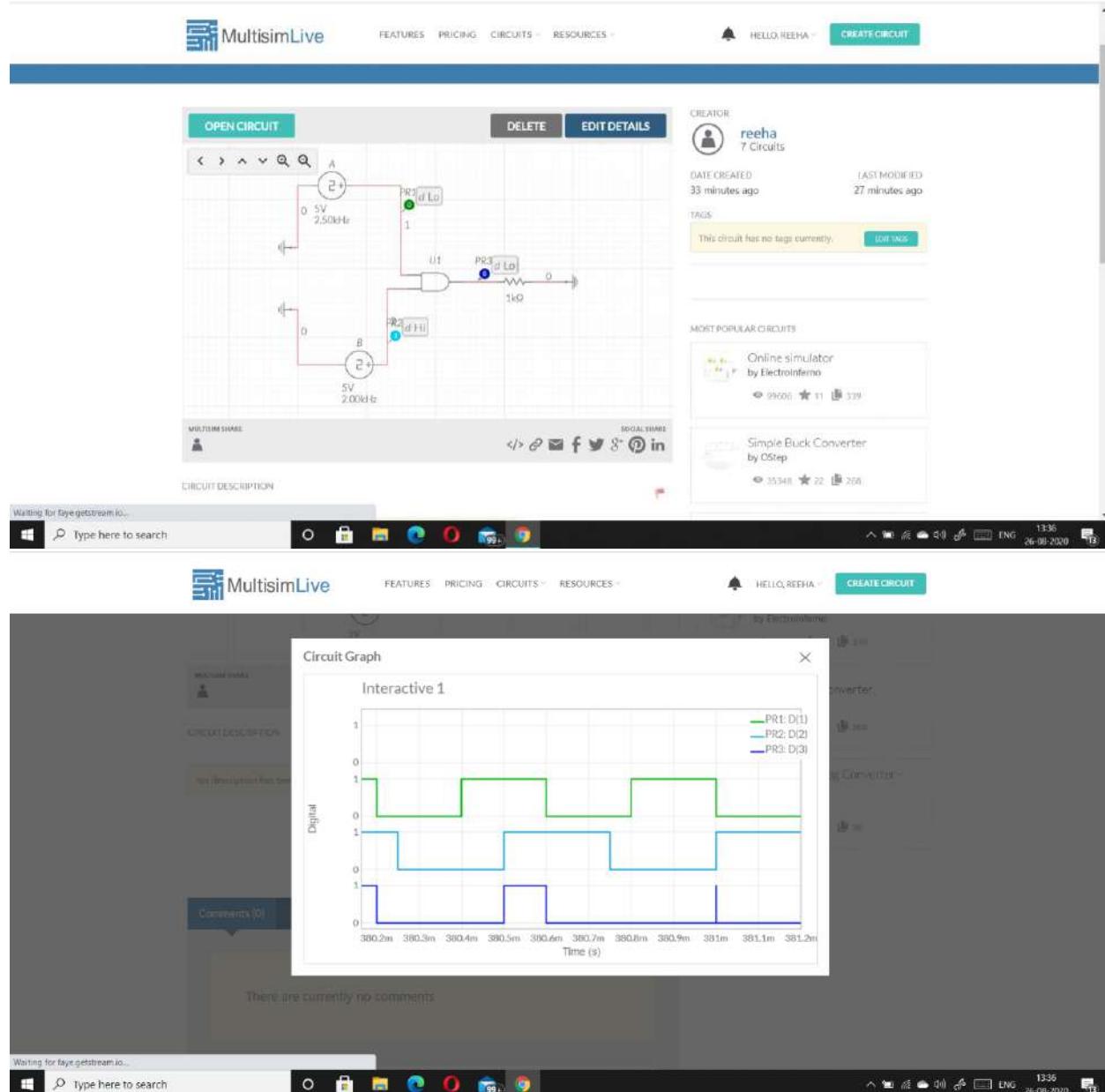
PR3: D(3)

Time (s)

Comments (0)

There are currently no comments.

## AND GATE



## Result:

All Gates has been realized.

## **VIVA VOCE**

### **Q1. Explain what is Boolean Algebra?**

Boolean algebra is a mathematic system of logic in which truth functions are expresses as symbols and then these symbols are manipulated to arrive at conclusion.

### **Q2. Explain what are the basic logic elements?**

Basic logic elements are NOT gate, AND gate, OR gate and the flip-flop.

### **Q3. Explain what is a truth table?**

Truth table is a table that gives outputs for all possible combinations of inputs to a logic circuit.

### **Q4. Define positive logic and negative logic.**

If the higher of the two voltages represents a 1 and the lower voltage represents a 0, then the logic is called a positive logic. On the other hand, if the lower voltage represents a 1 and the higher voltage a 0, we have a negative logic.

### **Q5. Explain what is pulse logic system?**

A logic system in which a bit is recognized by the presence or absence of a pulse is called a pulse or dynamic logic system.

# EXPERIMENT 2

Switching Theory and Logic Design (STLD)

## Aim

To verify the truth tables of all logical gates(AND, OR, NOT, NAND, NOR, XOR, XNOR) **using NAND gate** only.

Syeda Reeha Quasar

14114802719

3C7

# EXPERIMENT 2

## Aim:

To verify the truth tables of all logical gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) using NAND gate.

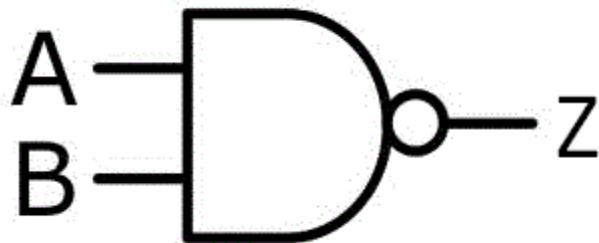
## Hardware and Software Apparatus Required

Hardware: Power supply, Bread Board, Connecting Wires, respective IC, LED, Wire Cutter.

- ❖ Circuit is designed on bread board using Integrated Chips (ICs), Voltage supply and LEDs.
- ❖ The set-up of apparatus and working of the circuit were demonstrated via recorded videos.

Software Simulation: The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com): Sign Up and create a profile.

## Circuit (SYMBOL)

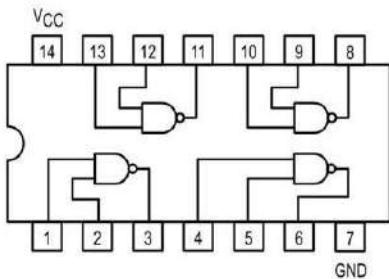


**Digital Logic NAND Gate – Universal Gate**

$$\text{Output} = \overline{A \cdot B}$$

## IC pin diagram

NAND GATE (IC 7400)



## Truth Table

NAND GATE

NAND Truth Table		
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

## Relevant Theory

NAND Gate

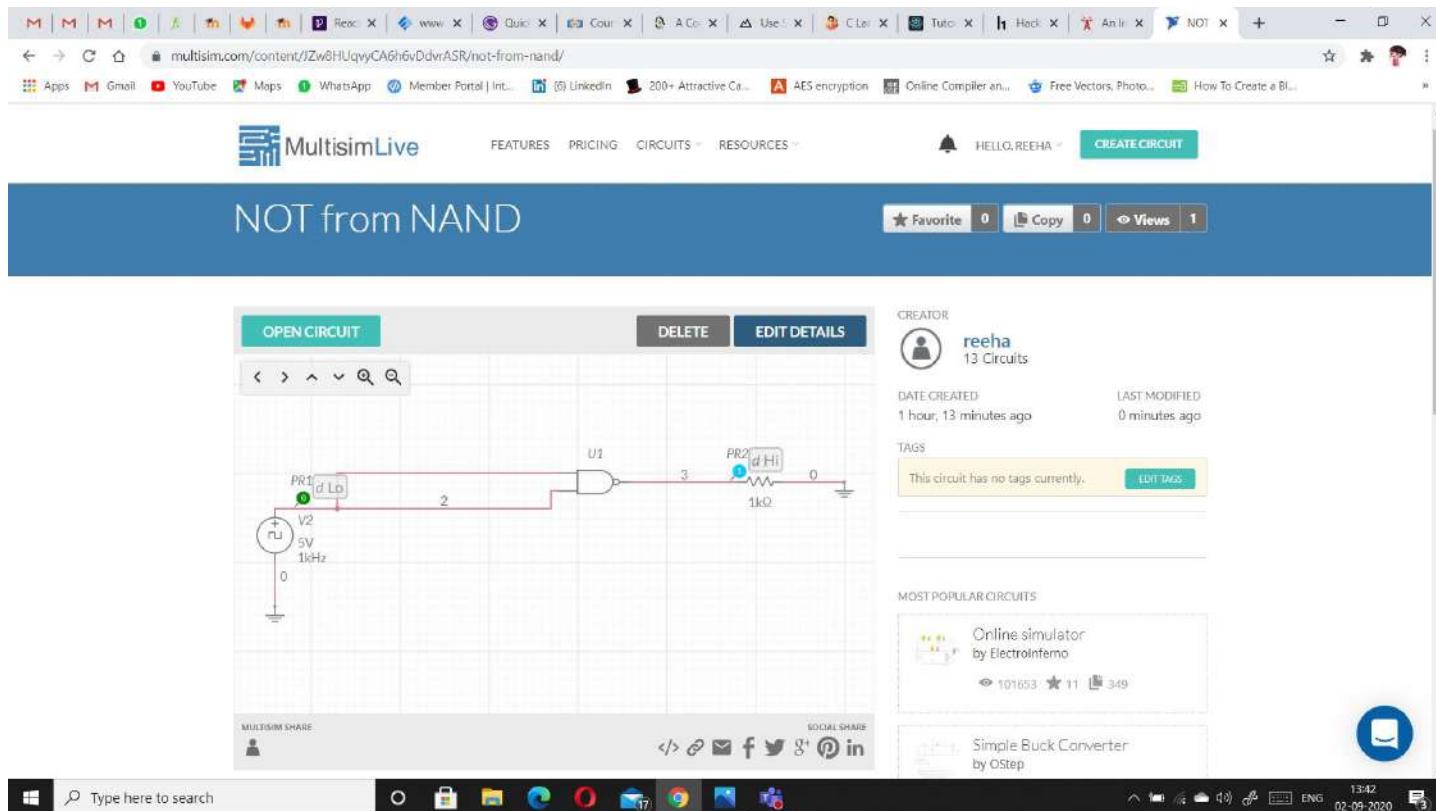
- The Boolean expression for a logic NAND gate is denoted by a single dot or full stop symbol, (.) with a line or Overline, ( $\overline{\cdot}$ ) over the expression to signify the NOT or logical negation of the NAND gate giving us the Boolean expression of:  $A \cdot B = Q$ .
- It is a series of AND gate followed by NOT gate. If the output is 0 then any or all inputs are 1. Otherwise, output is 1.
- **It is a universal gate:** Universal Logic gates can be used to produce any other logic or Boolean function with the NAND and NOR gates being minimal. Individual logic gates can be connected together to form a variety of different switching functions and combinational logic circuits.

## Procedure followed on MULTISIM

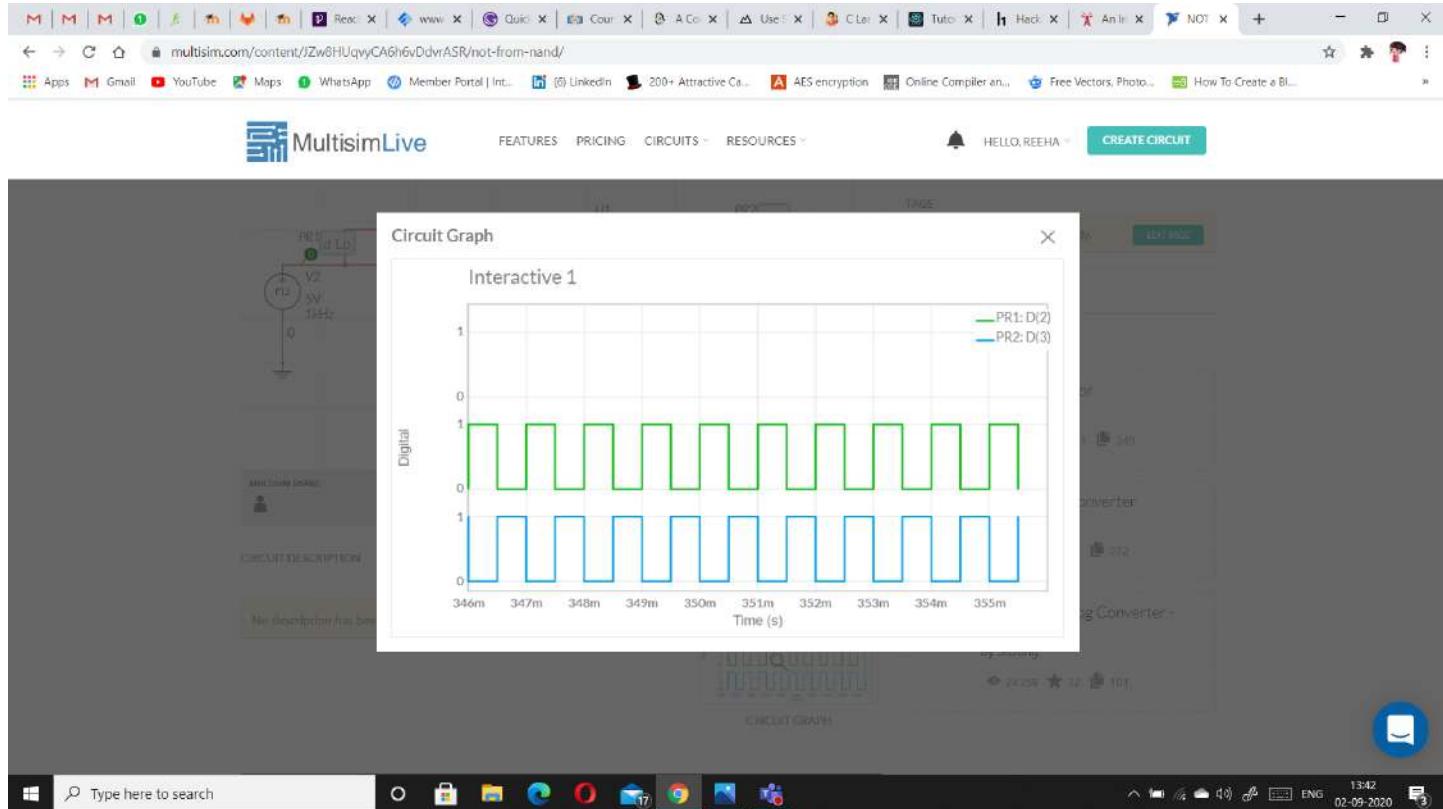
1. LOG IN ON [www.multisim.com](http://www.multisim.com)
2. CREATE THE CIRCUIT
3. SAVE THE CIRCUIT

- 4. SAVE THE SCREENSHOTS FOR**
  - i. INPUT & OUTPUT WAVEFORMS (ALONG WITH YOUR ID ON TOP LEFT)**
  - ii. CIRCUIT (ALONG WITH YOUR ID ON TOP LEFT)**

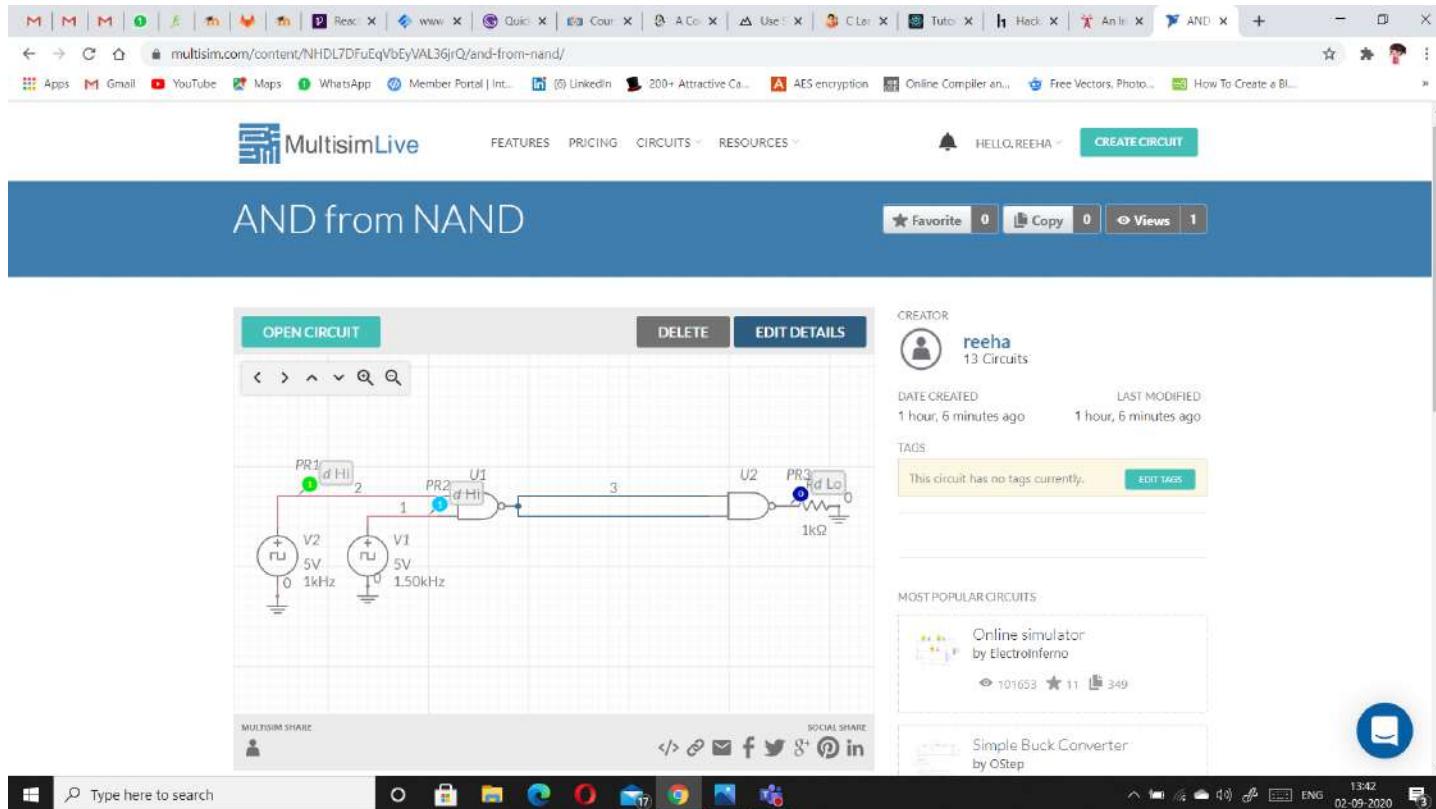
## Screenshot of Circuit: NOT(INVERTER) USING NAND



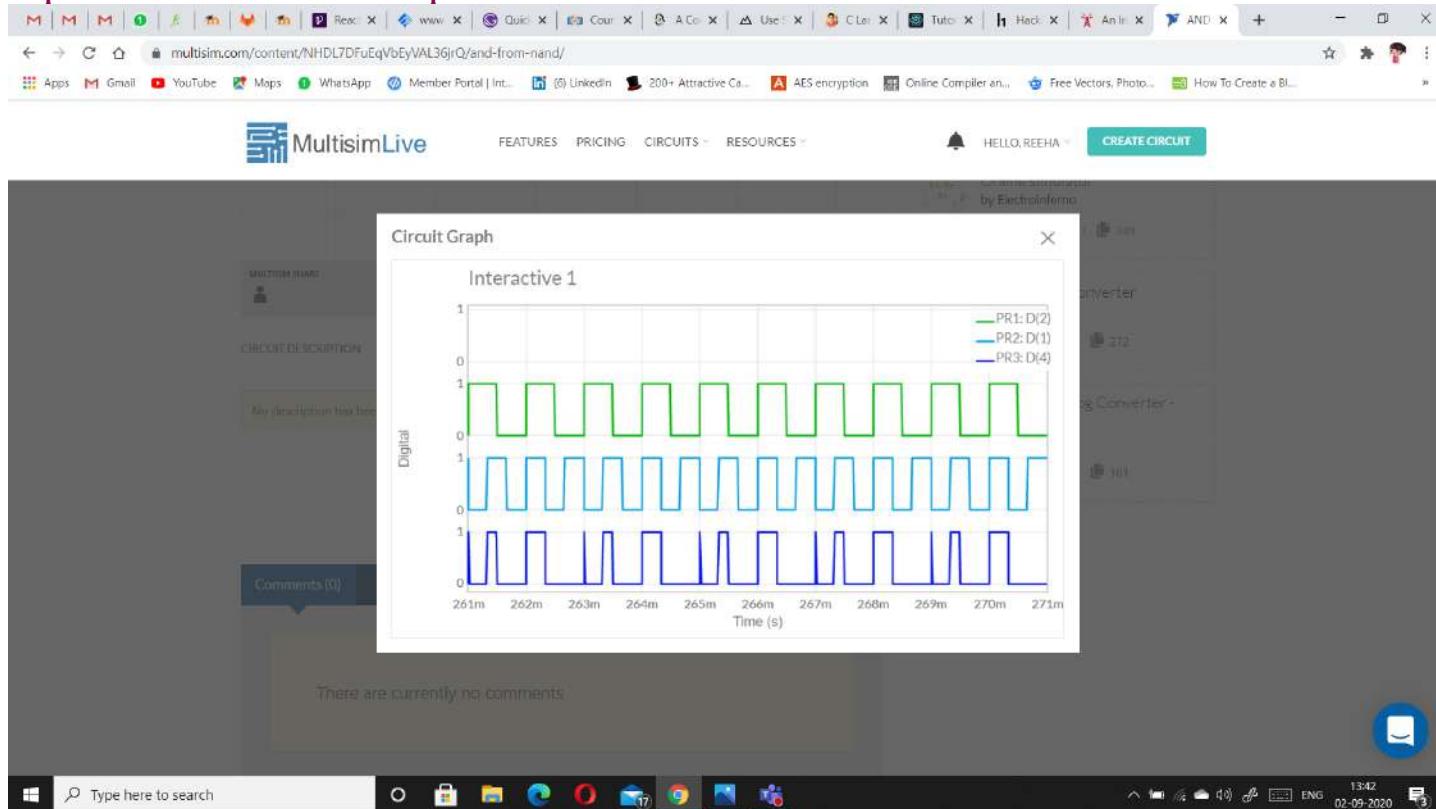
## Input waveform and Output waveform



Screenshot of Circuit: *AND USING NAND*



## Input waveform and Output waveform



## Screenshot of Circuit: *OR USING NAND*

MultisimLive

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA CREATE CIRCUIT

OR from NAND

Favorite 0 Copy 0 Views 1

OPEN CIRCUIT DELETE EDIT DETAILS

V1 5V 1kHz PR1 d Lo U1 4

V2 6.5V 1.50kHz PR2 d Hi U2 3

U3 R1 1kΩ

DATE CREATED 23 minutes ago LAST MODIFIED 23 minutes ago

TAGS This circuit has no tags currently. EDIT TAGS

MOST POPULAR CIRCUITS

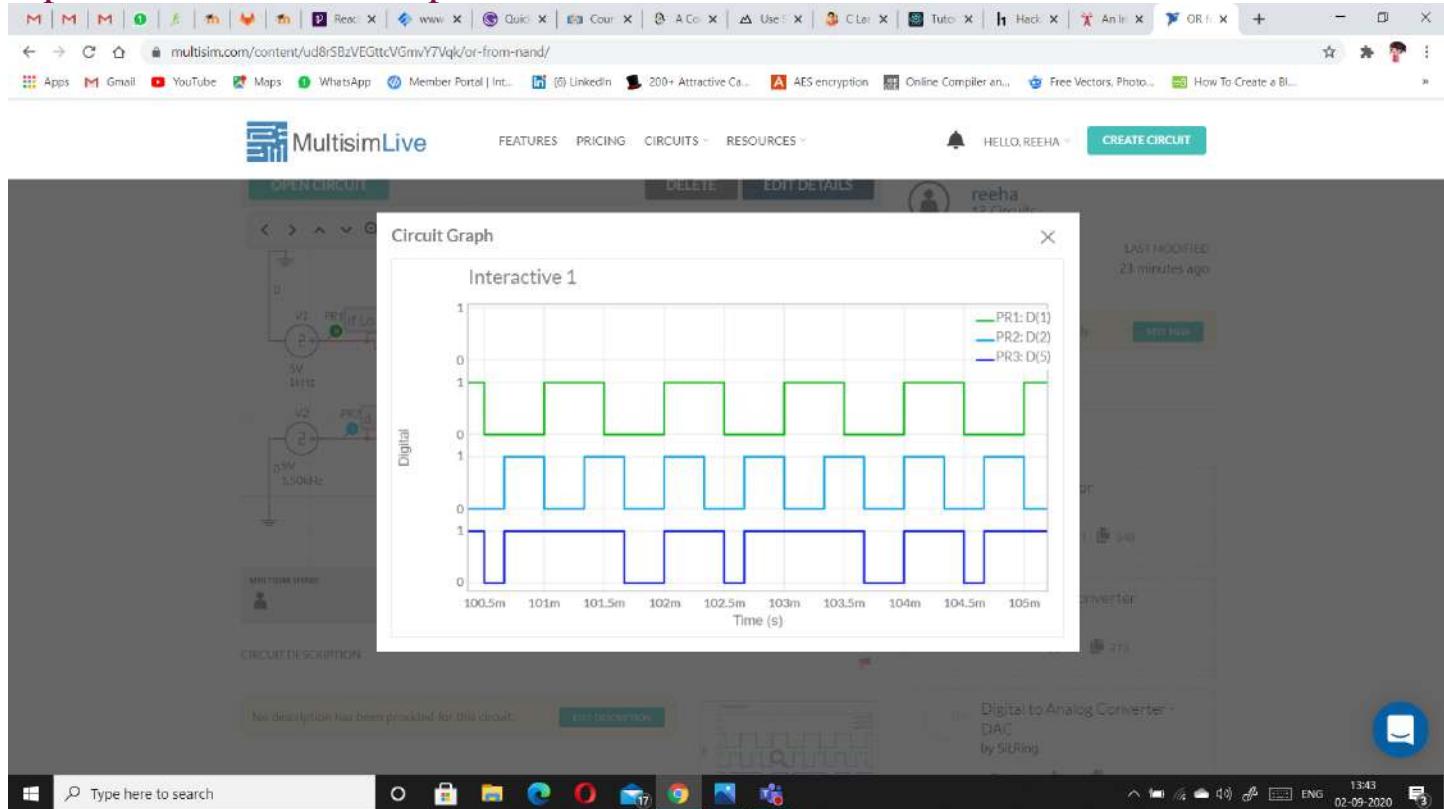
Online simulator by ElectroInferno 101654 11 349

Simple Buck Converter by OStep

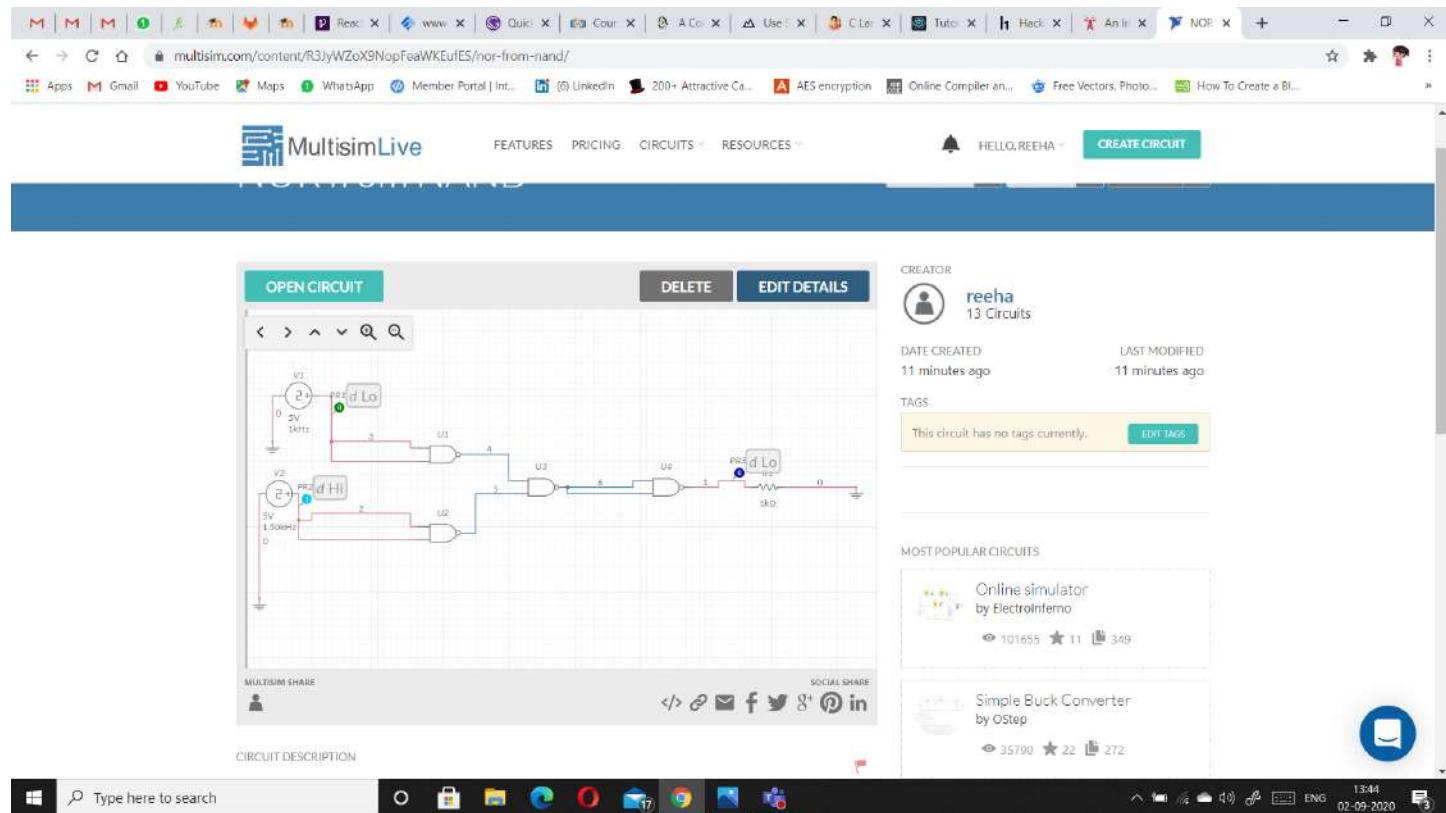
Type here to search

The screenshot shows a logic circuit titled "OR from NAND" on MultisimLive. The circuit consists of three NAND gates (U1, U2, U3) configured as inverters. Input V1 (5V, 1kHz) is connected to the first input of U1 and the second input of U2. Input V2 (6.5V, 1.50kHz) is connected to the second input of U1 and the first input of U2. The outputs of U1 and U2 are connected to the inputs of U3. The output of U3 is connected to ground through a 1kΩ resistor (R1). The circuit is powered by a 5V DC source. The creator is "reeha" with 13 circuits. The date created and last modified are both 23 minutes ago. There are no tags. The most popular circuit is "Online simulator" by ElectroInferno with 101654 views, 11 ratings, and 349 comments.

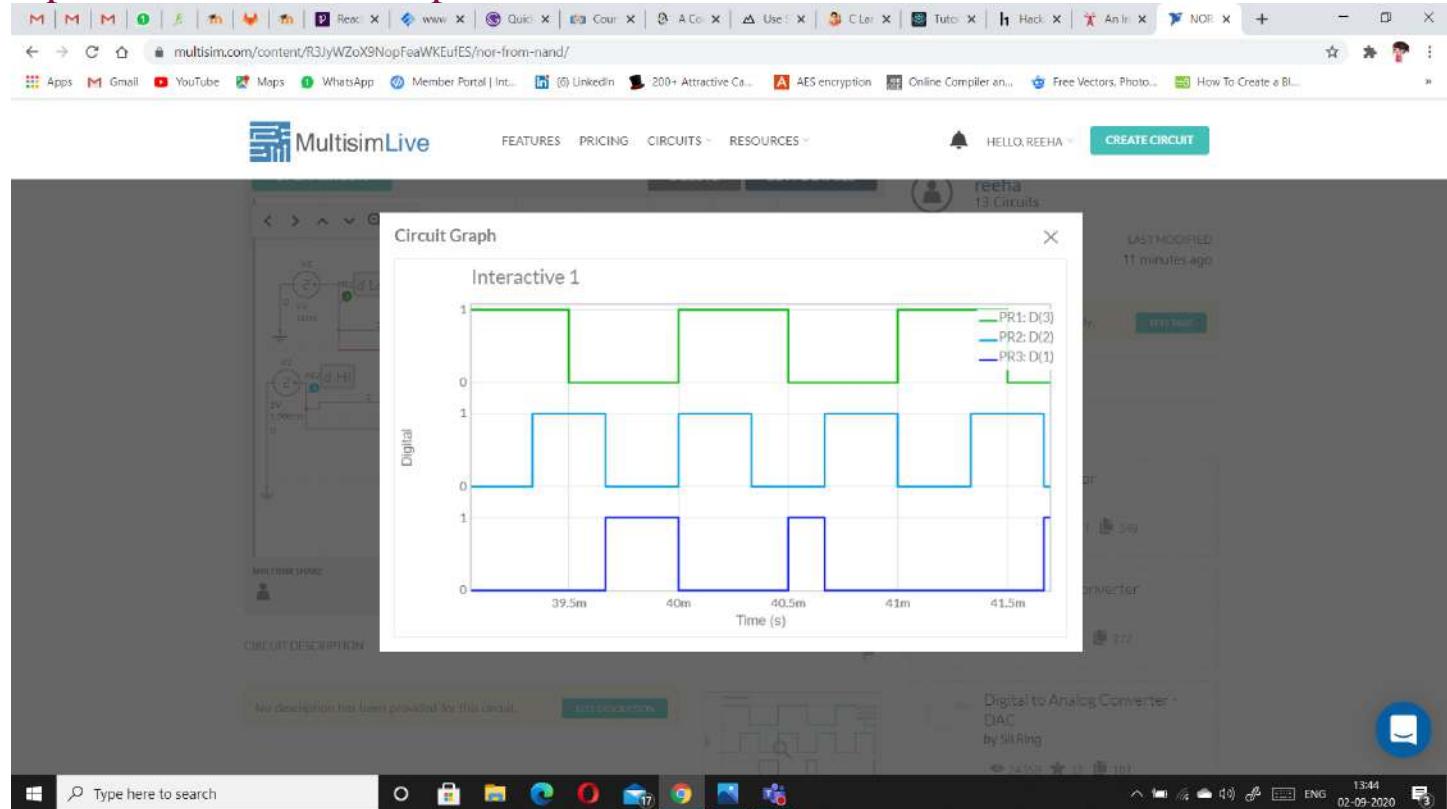
## Input waveform and Output waveform



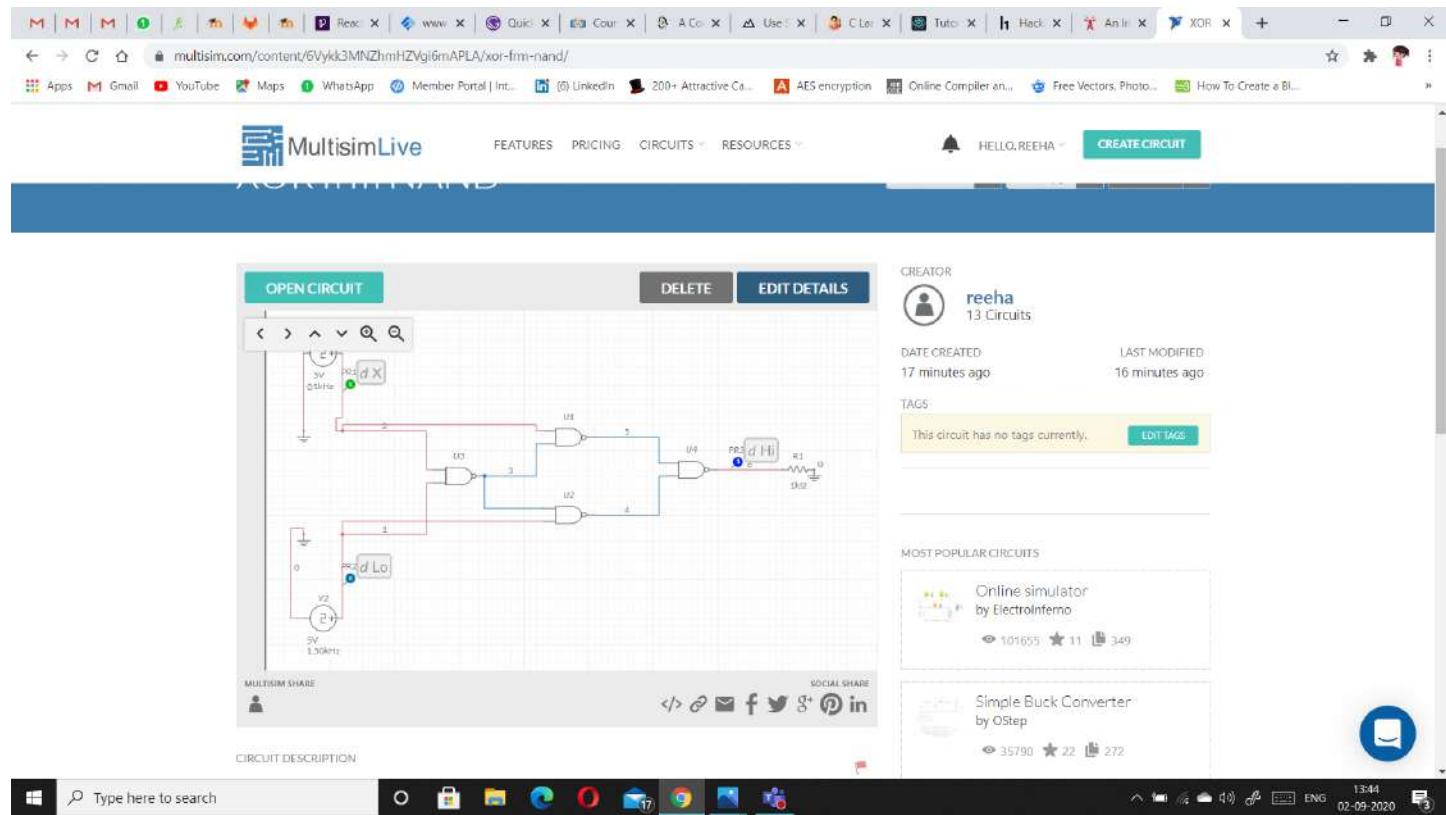
## Screenshot of Circuit: NOR USING NAND



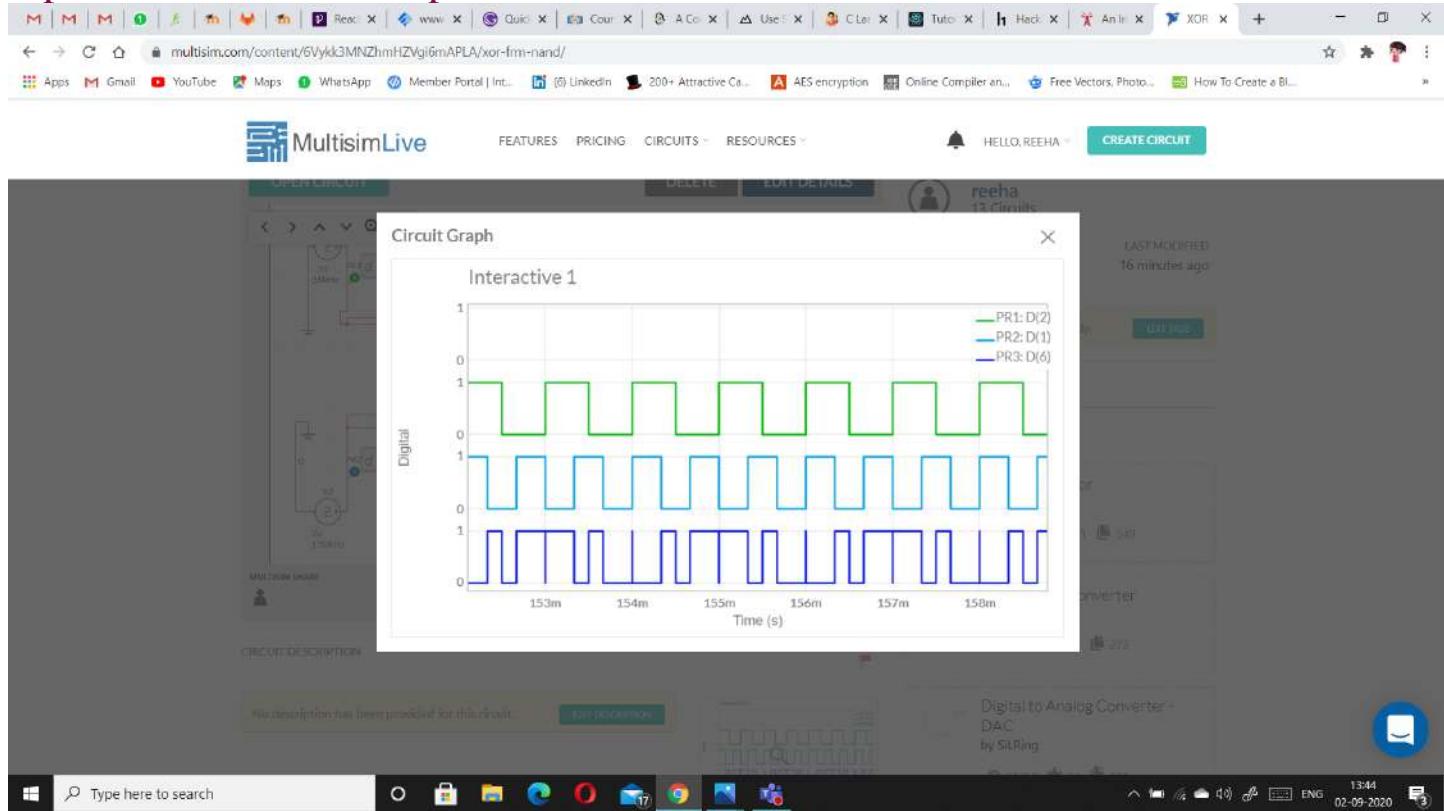
## Input waveform and Output waveform



## Screenshot of Circuit: *XOR USING NAND*



## Input waveform and Output waveform



## Screenshot of Circuit: XNOR USING NAND

MultisimLive

FEATURES PRICING CIRCUITS RESOURCES

HELLO, REEHA CREATE CIRCUIT

XNOR from NAND

Favorite 0 Copy 0 Views 1

OPEN CIRCUIT DELETE EDIT DETAILS

REDACTED Lo

REDACTED Hi

U1 U2 U3 U4 U5

V1 3V 5MHz V2 3V 1.5MHz

MULTIMIM SHARE

SOCIAL SHARE

Creator: reeha (13 Circuits)

DATE CREATED: 6 minutes ago LAST MODIFIED: 6 minutes ago

TAGS: This circuit has no tags currently. EDIT TAGS

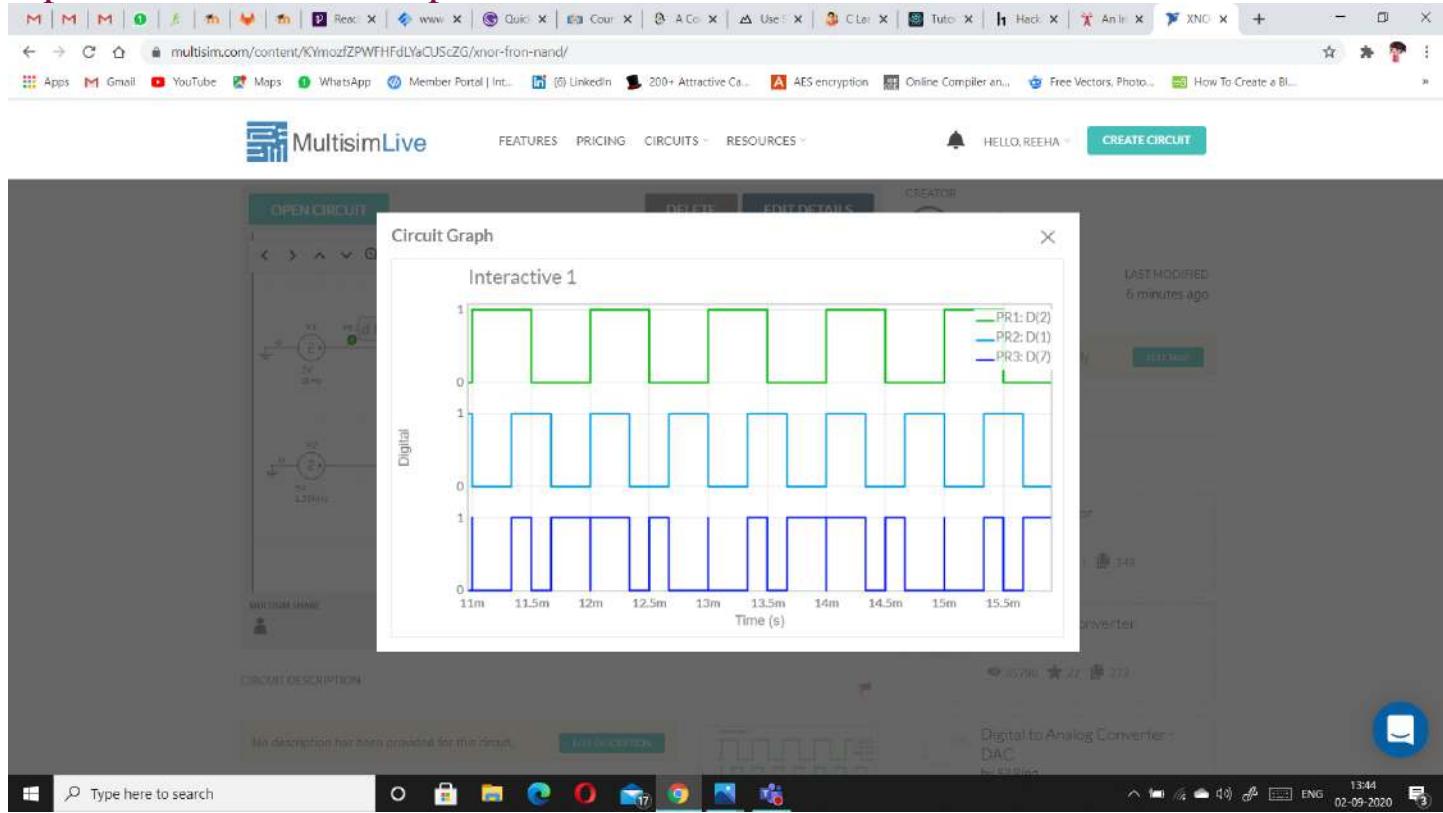
MOST POPULAR CIRCUITS

Online simulator by ElectroInferno 101655 11 349

Simple Buck Converter by OStep

```
graph LR; V1((V1)) --> U1; V2((V2)) --> U1; U1((U1)) --> U2((U2)); U1 --> U3((U3)); U2 --> U4((U4)); U3 --> U4; U4((U4)) --> U5((U5)); U5 --> Output(( ));
```

## Input waveform and Output waveform



## Precautions (MULTISIM):

1. Frequency of clock voltage source should be different for both inputs.
2. Place the probes carefully only at the input and output sources.
3. Use digital analyzer probe.
4. Set the type to transient.
5. Ground both the voltage sources(clock) and the resistor.

## **VIVA VOCE**

### **Q1. Explain what is an inverter?**

An inverter is a logic gate whose output is the inverse or complement of its input.

### **Q2. Explain what are the universal logic gates?**

Universal gate is a gate that can perform all the basic logical operations such as NAND and NOR gates.

### **Q3. Explain what is the specialty of NAND and NOR gates?**

The specialty of NAND and NOR gates is that they are universal gates and can perform all the basic logical operations.

### **Q4. Explain why NAND-NAND realization is preferred over AND-OR realization?**

NAND-NAND realization needs only one type of gate(NAND), that minimizes IC package counter.

### **Q5. Explain why is a two-input NAND gate called universal gate?**

NAND gate is called universal gate because any digital system can be implemented with the NAND gate. Sequential and combinational circuits can be constructed with these gates because element circuits like flip-flop can be constructed from two NAND gates connected back-to-back. NAND gates are common in hardware because they are easily available in the ICs form. A NAND gate is in fact a NOT-AND gate. It can be obtained by connecting a NOT gate in the output of an AND gate.

# EXPERIMENT - 3

Switching Theory and Logic Design (STLD)

## Aim

To verify the truth tables of all logical gates(AND, OR, NOT, NAND, NOR, XOR, XNOR) using NOR gate only.

Syeda Reeha Quasar

14114802719

3C7

## EXPERIMENT - 3

### AIM:

To verify the truth tables of all logical gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) using NOR gate only.

### Hardware and Software Apparatus Required

#### Hardware:

- ❖ Power supply, Bread Board, Connecting Wires, respective IC, LED, Wire Cutter.
- ❖ Circuit is designed on bread board using Integrated Chips (ICs), Voltage supply and LEDs.
- ❖ The set-up of apparatus and working of the circuit were demonstrated via recorded videos.

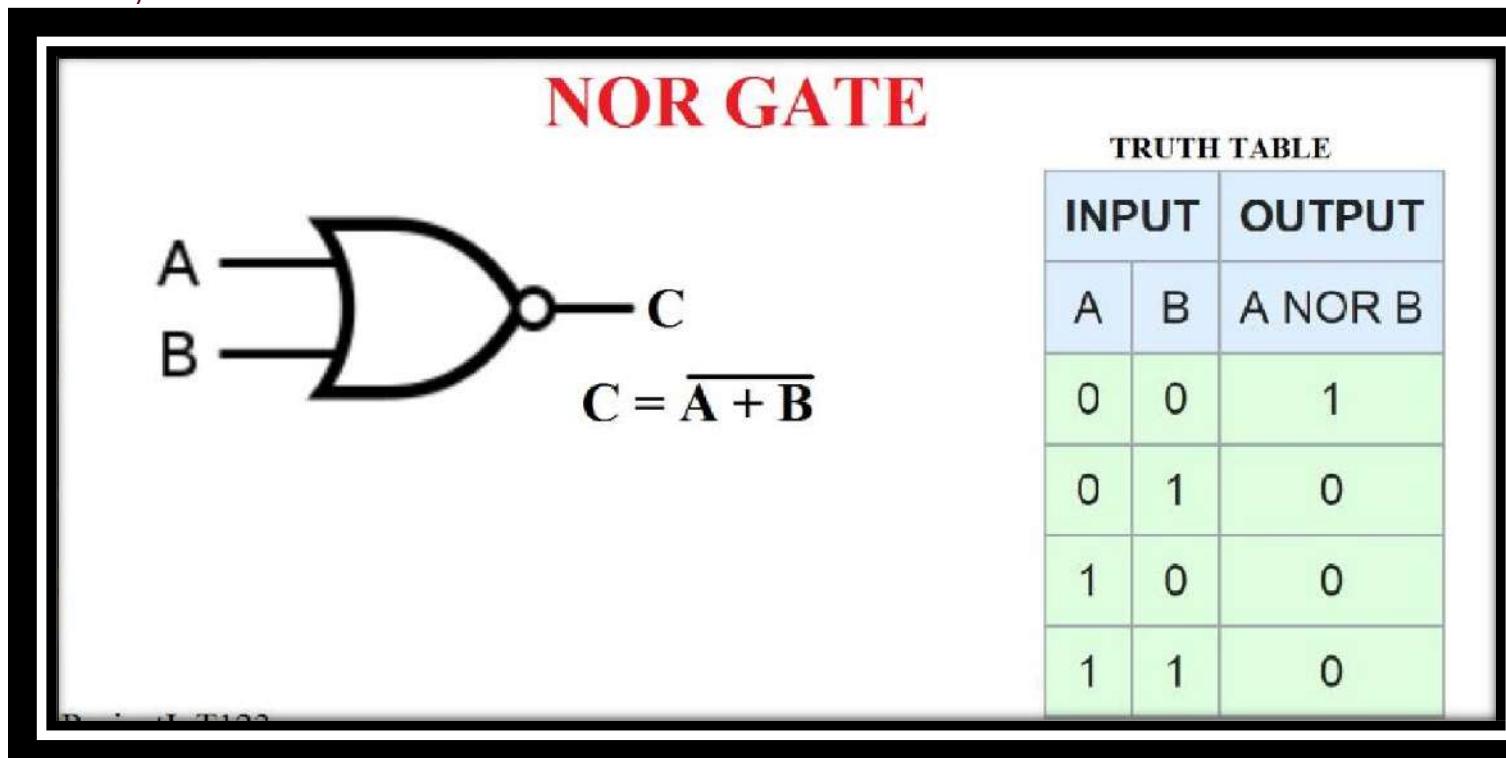
#### Software Simulation:

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

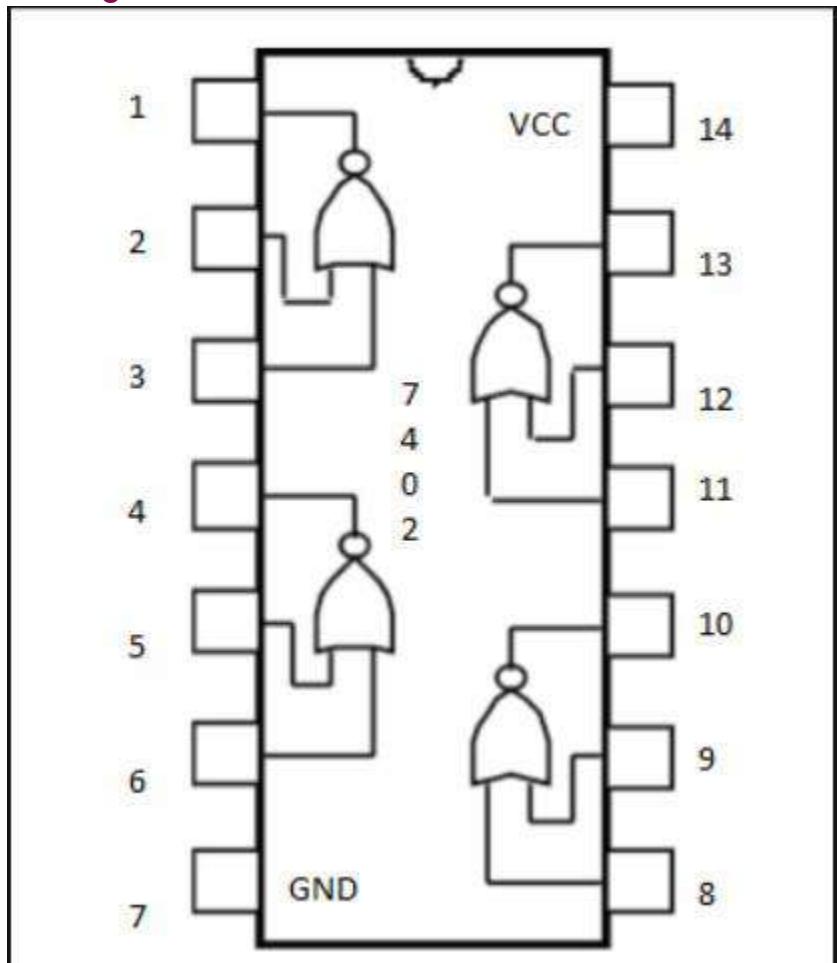
**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

### Theory:

Circuit Symbol and Truth table:



### Pin Diagram:



### NOR gate:

The NOR gate is a digital logic gate that implements logical NOR - it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results.

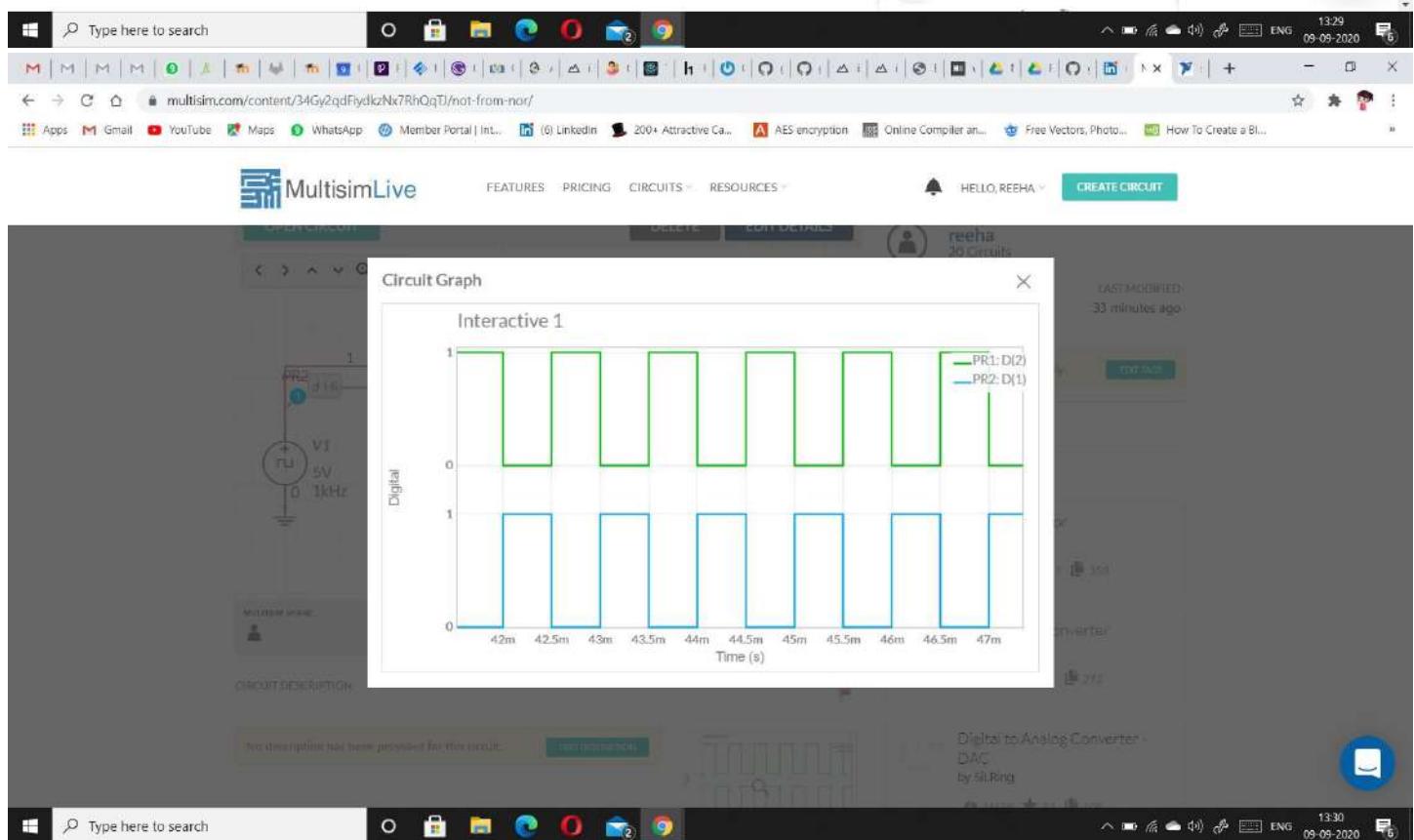
NOR is the result of the negation of the OR operator. It can also in some senses be seen as the inverse of an AND gate. NOR is a functionally complete operation—NOR gates can be combined to generate any other logical function. It shares this property with the NAND gate. By contrast, the OR operator is monotonic as it can only change LOW to HIGH but not vice versa.

### Procedure followed on MULTISIM:

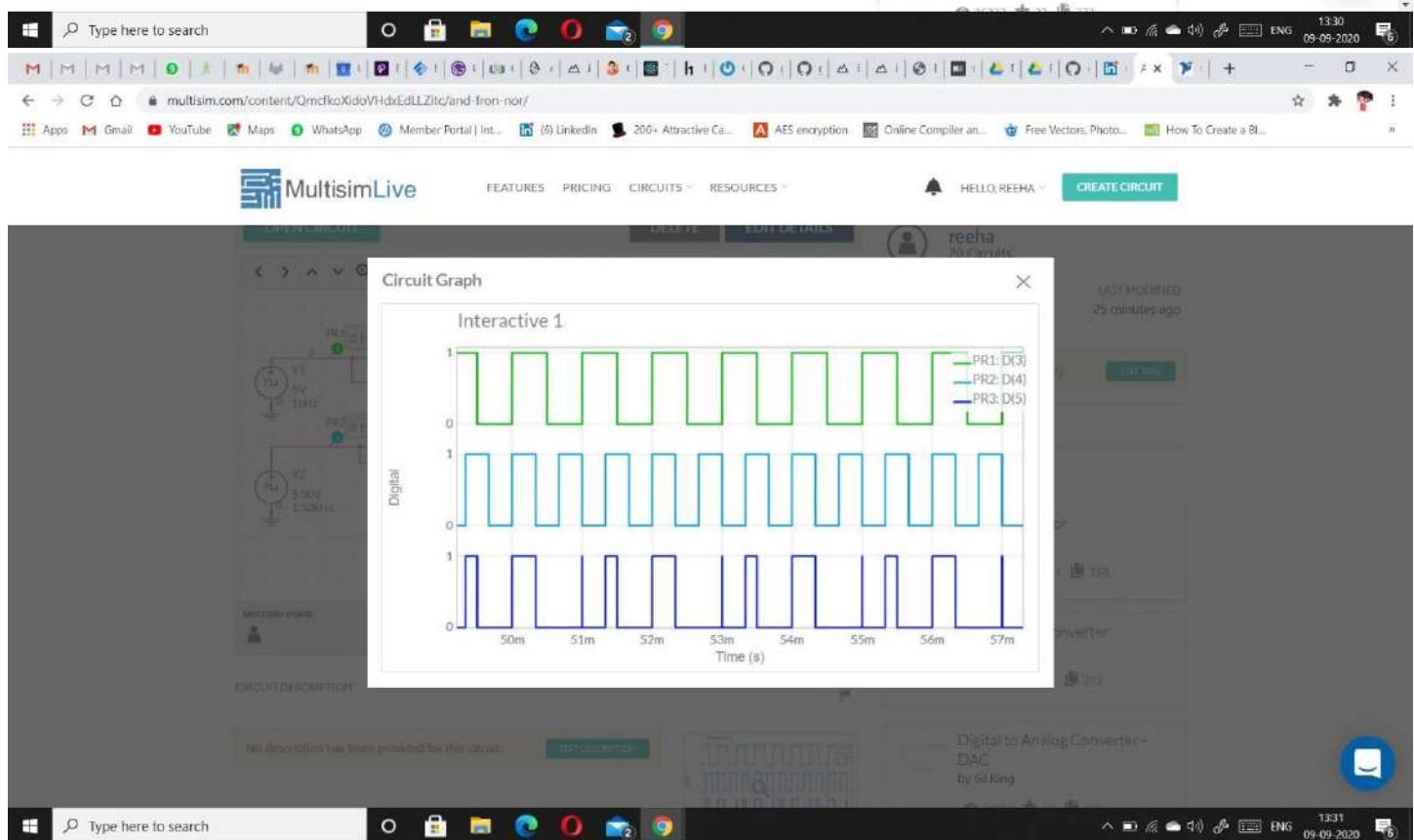
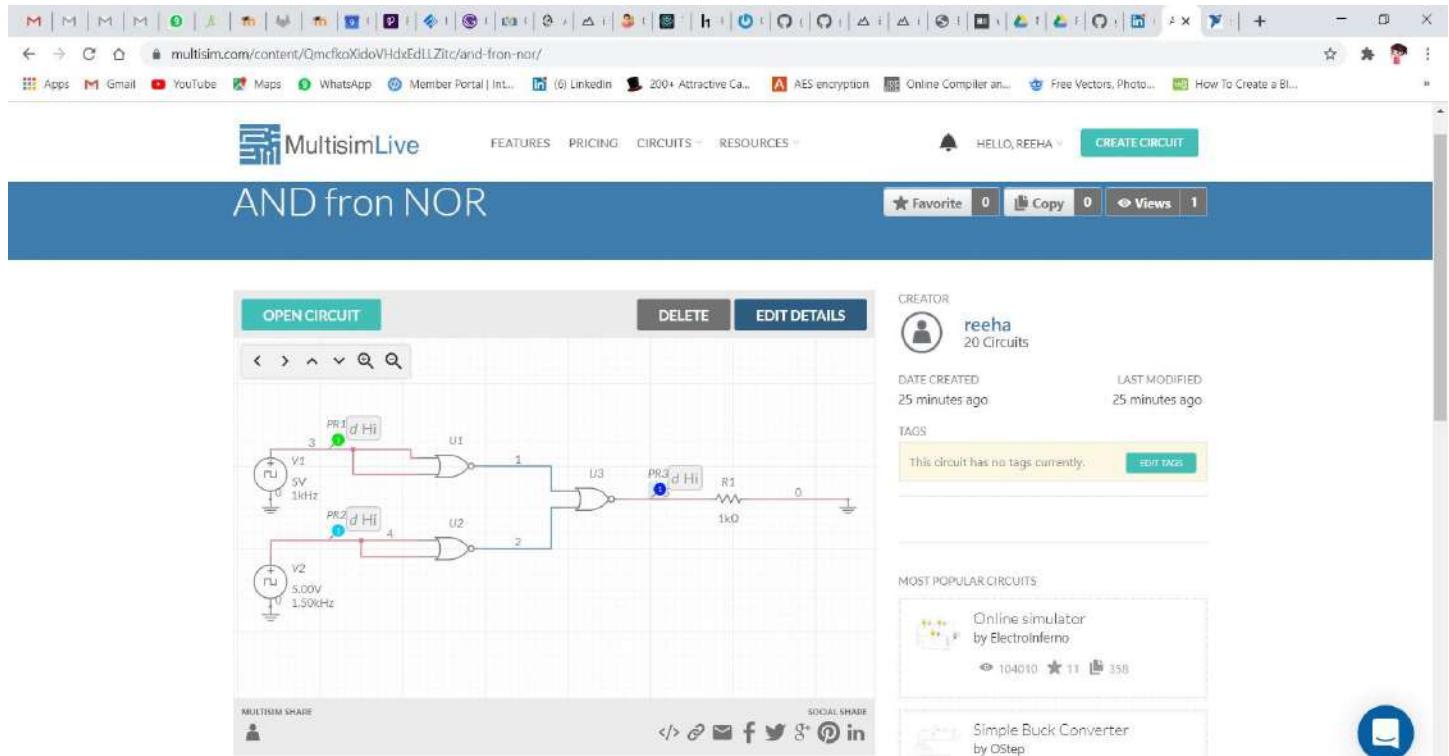
1. LOG IN ON [www.multisim.com](http://www.multisim.com)
2. CREATE THE CIRCUIT
3. SAVE THE CIRCUIT
4. SAVE THE SCREENSHOTS FOR
  - i. INPUT & OUTPUT WAVEFORMS (ALONG WITH YOUR ID ON TOP LEFT)
  - ii. CIRCUIT (ALONG WITH YOUR ID ON TOP LEFT)

# Screenshot of circuit: NOT(Inverter) using NOR

The screenshot shows a digital circuit on MultisimLive. The circuit consists of a NOR gate U1 with its inputs connected to ground through pull-down resistors PR1 and PR2. The output of the NOR gate is connected to a 1kΩ resistor R1, which is connected to ground. A 5V AC voltage source V1 is connected between the input of the NOR gate and ground. The circuit is titled "NOT from NOR". On the right side, there is a "CREATOR" section for "reeha" who has 20 Circuits. The circuit was last modified 33 minutes ago. Below the circuit diagram, there are social sharing options and a "Circuit Graph" button.

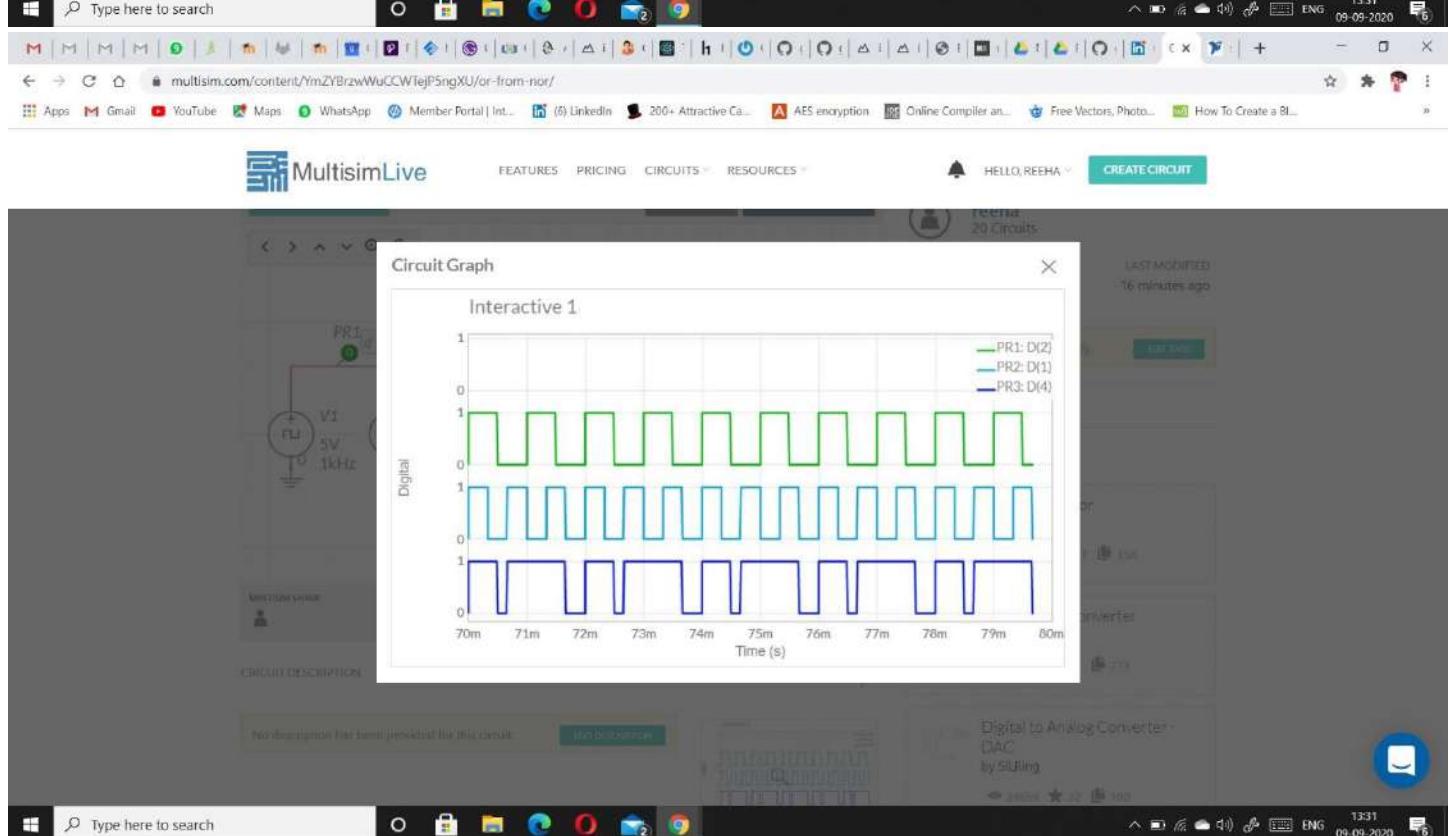


# Screenshot of circuit: AND using NOR

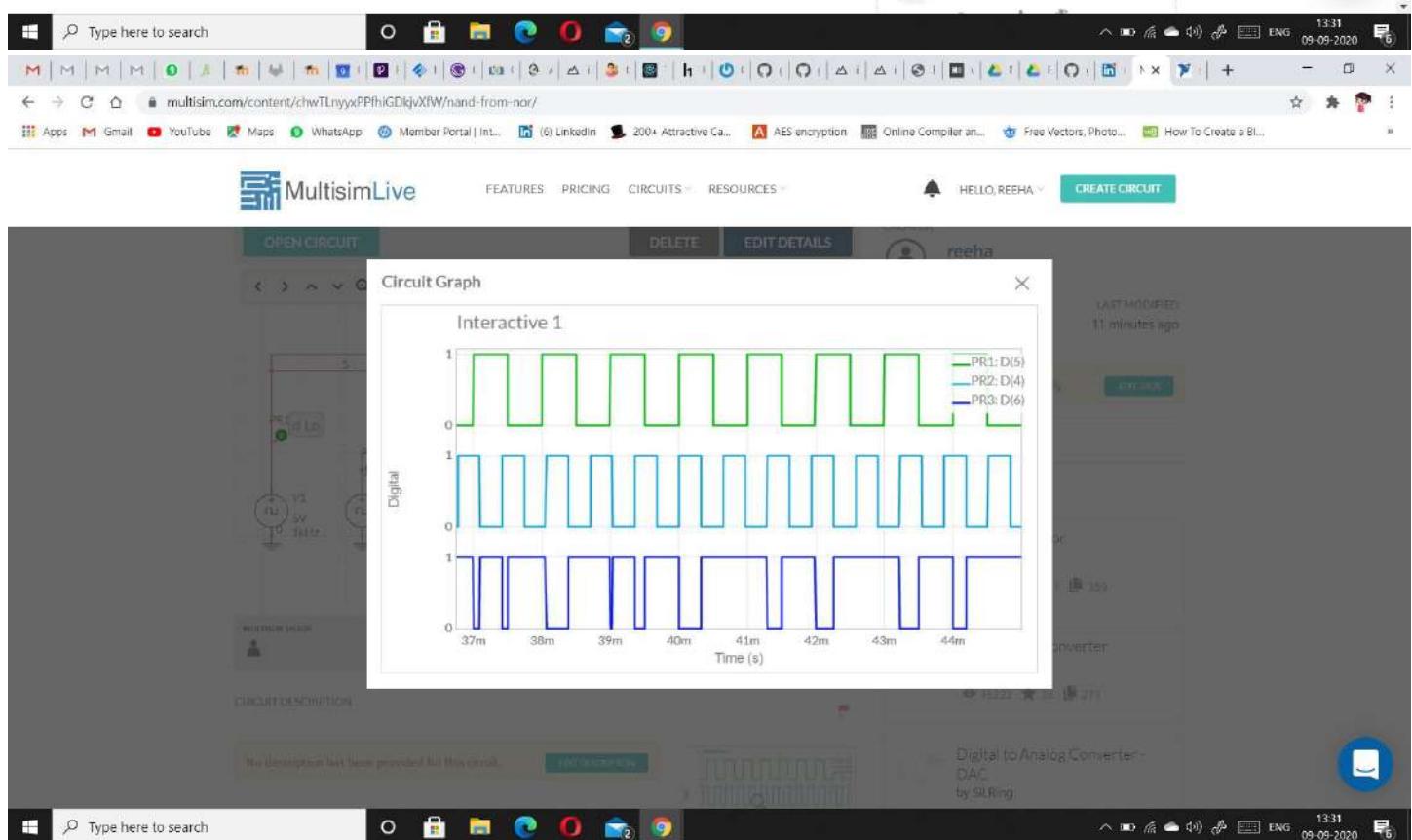
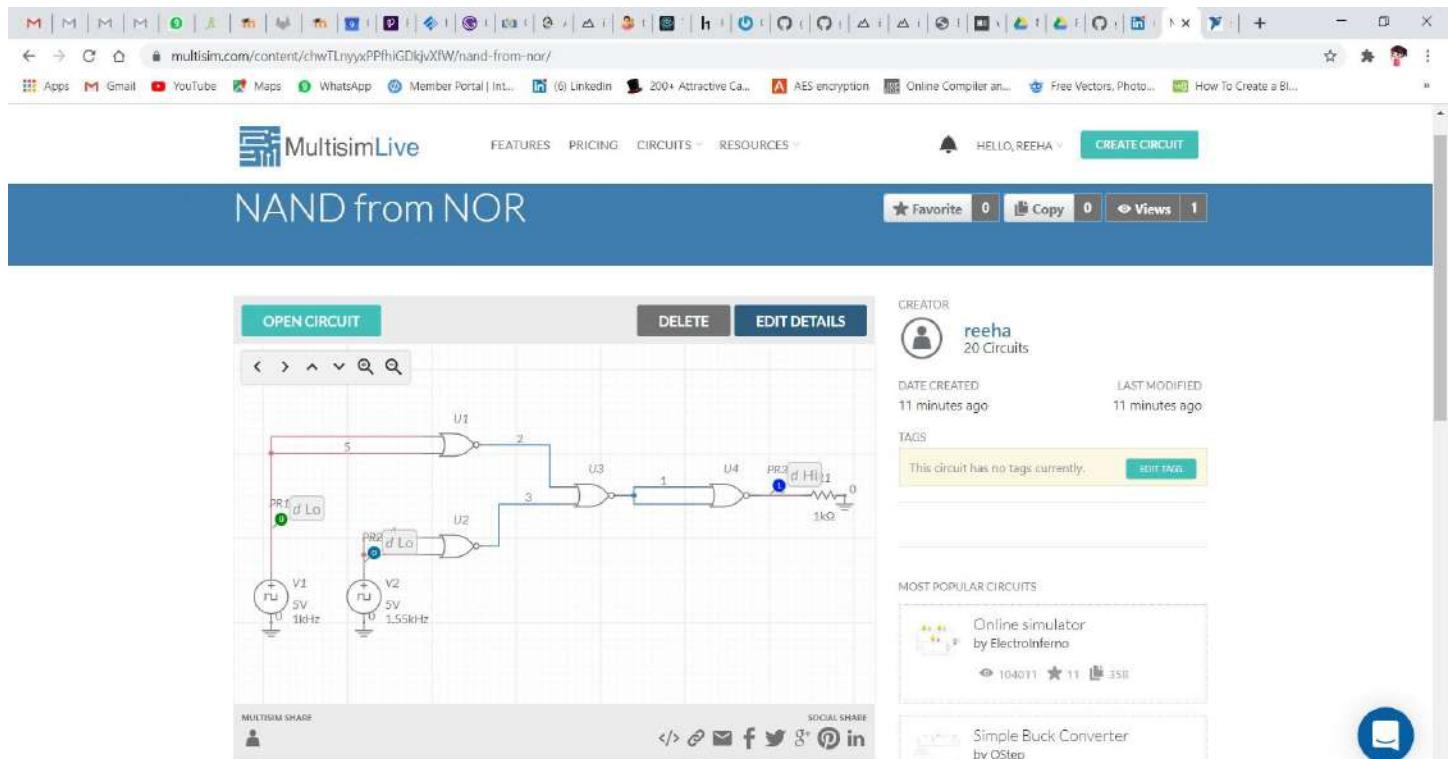


## Screenshot of circuit: OR using NOR

The screenshot shows a digital logic circuit on MultisimLive. The circuit consists of two inputs, V1 and V2, connected to the inputs of a NOR gate U1. The output of U1 is connected to one input of a second NOR gate U2. The other input of U2 is connected to ground through a resistor R1 (1kΩ). The output of U2 is labeled PR3(d). The inputs V1 and V2 are AC voltage sources with 5V amplitude and 1kHz frequency. The output PR3(d) is also labeled with a 1kHz frequency.



## Screenshot of circuit: NAND using NOR

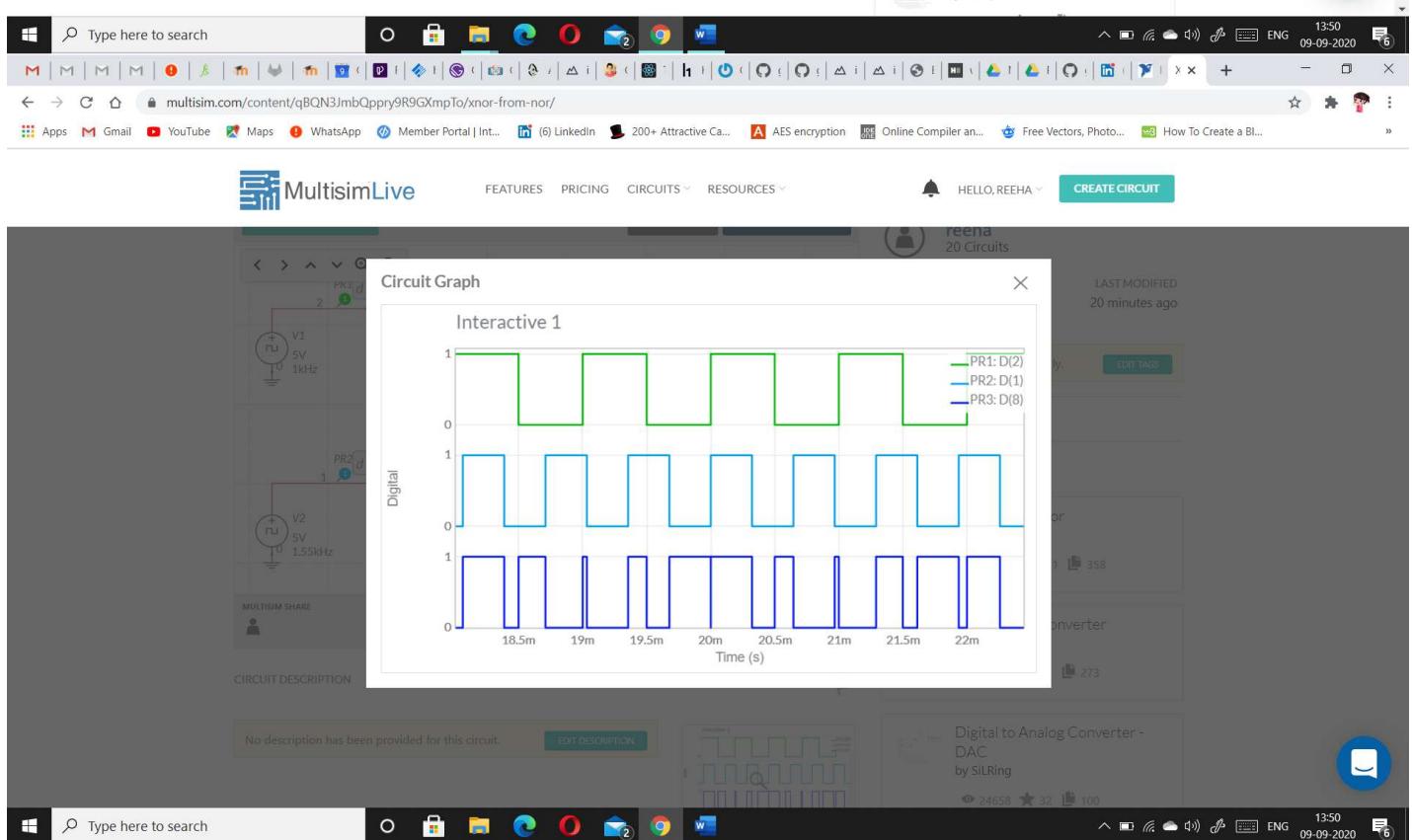
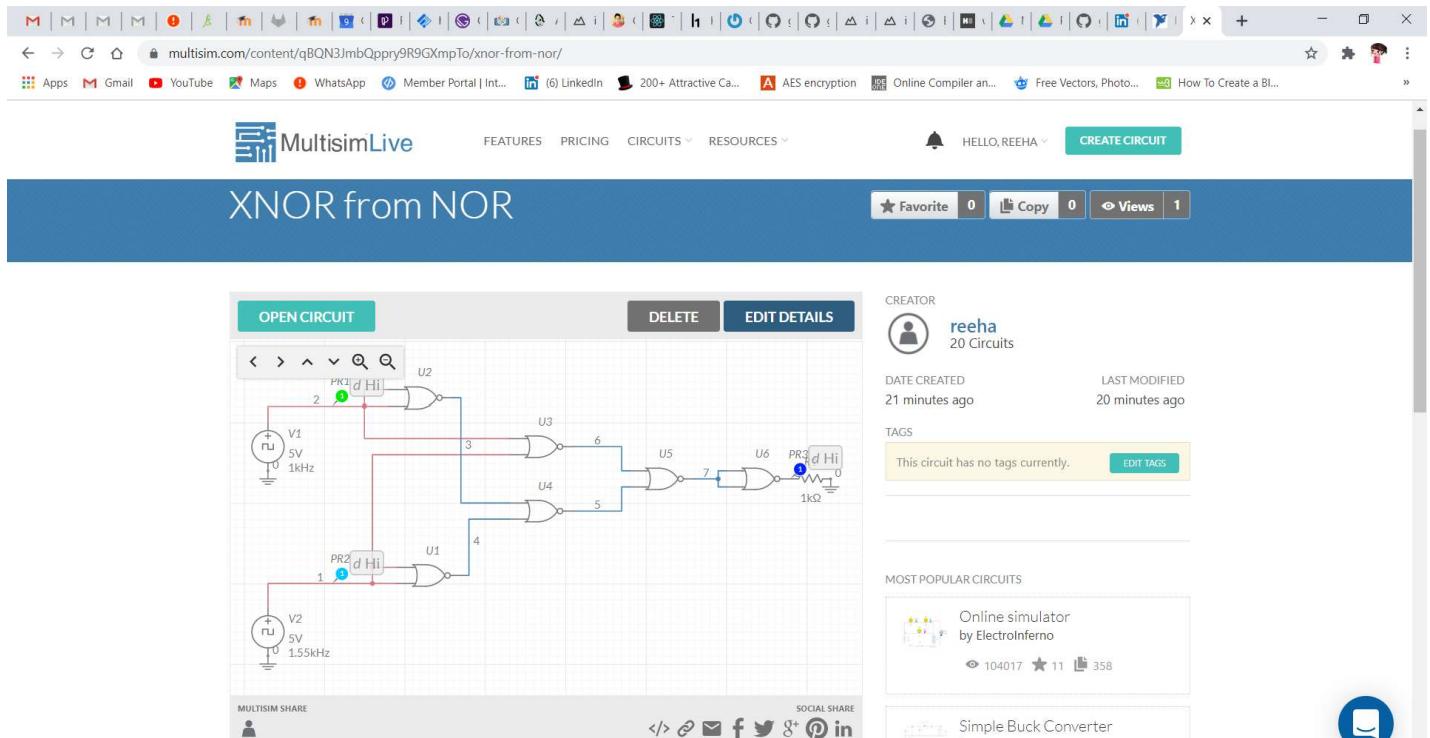


## Screenshot of circuit: XOR using NOR

The screenshot shows a digital circuit on MultisimLive. The circuit consists of four NOR gates (U1, U2, U3, U4) configured to implement an XOR function. The inputs are V1 (0V, 5V, 1kHz) and V2 (0V, 5V, 1.55kHz). The output of U1 is connected to one input of U3. The output of U2 is connected to one input of U4. The outputs of U3 and U4 are connected to the inputs of U5. The output of U5 is connected to ground through a 1kΩ resistor (R1). The circuit is powered by a 5V DC source. The creator of the circuit is 'reeha' with 20 circuits.

The screenshot shows the circuit graph for the same XOR circuit. The graph displays three digital waveforms: PR1: D(2) (green), PR2: D(1) (cyan), and PR3: D(7) (blue). The waveforms are periodic square waves. The graph includes a legend and a time axis from 48m to 58m. The circuit description indicates no description has been provided for this circuit.

## Screenshot of circuit: XNOR using NOR



## **Result:**

All gates has been verified

## **Viva Questions**

1. A logic circuit that provides a HIGH output for both inputs HIGH or both inputs LOW is a(n):

- A. Ex-NOR gate
- B. OR gate
- C. Ex-OR gate
- D. NAND gate

ANS.

Option A

2. How many two-input AND and OR gates are required to realize  $Y = CD + EF + G$ ?

- a) 2, 2
- b) 2, 3
- c) 3, 3
- d) 3, 2

Ans.

Answer: a

Explanation:  $Y = CD + EF + G$

The number of two input AND gate = 2

The number of two input OR gate = 2.

3. Which of following are known as universal gates?

- a) NAND & NOR
- b) AND & OR
- c) XOR & OR
- d) EX-NOR & XOR

Ans.

Answer: a

Explanation: The NAND & NOR gates are known as universal gates because any digital circuit can be realized completely by using either of these two gates, and also they can generate the 3 basic gates AND, OR and NOT.

4. A single transistor can be used to build which of the following digital logic gates?

- A. AND gates
- B. OR gates
- C. NOT gates
- D. NAND gates

Ans.

C) Or Gates

# **EXPERIMENT - 4**

Switching Theory and Logic Design (STLD)

## **Aim**

To realize the circuit for Half Adder and Full Adder using logic gates.

Syeda Reeha Quasar

14114802719

3C7

# **EXPERIMENT - 4**

## **AIM:**

To realize the circuit for Half Adder and Full Adder using logic gates.

## **Hardware and Software Apparatus Required**

### **Hardware:**

- ❖ Power supply, Bread Board, Connecting Wires, respective IC, LED, Wire Cutter.
- ❖ Circuit is designed on bread board using Integrated Chips (ICs), Voltage supply and LEDs.
- ❖ The set-up of apparatus and working of the circuit were demonstrated via recorded videos.

### **Software Simulation:**

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

## **Theory:**

An Adder is a device that can add two binary digits. It is a type of digital circuit that performs the operation of additions of two number. It is mainly designed for the addition of binary number, but they can be used in various other applications like binary code decimal, address decoding, table index calculation, etc. There are two types of Adder. One is Half Adder, and another one is known as Full Adder.

### **Half Adder**

There are two inputs and two outputs in a Half Adder. Inputs are named as A and B, and the outputs are named as Sum (S) and Carry (C). The Sum is X-OR of the input A and B. Carry is AND of the input A and B. With the help of half adder, one can design a circuit that is capable of performing simple addition with the help of logic gates. Let us first take a look at the addition of single bits.

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

These are the least possible single bit combinations. But the result for  $1 + 1 = 10$ . This problem can be solved with the help of an EX – OR gate. The sum results can be re-written as a 2-bit output. Thus the above combination can be written as

$$0 + 0 = 00$$

$$0 + 1 = 01$$

$$1 + 0 = 01$$

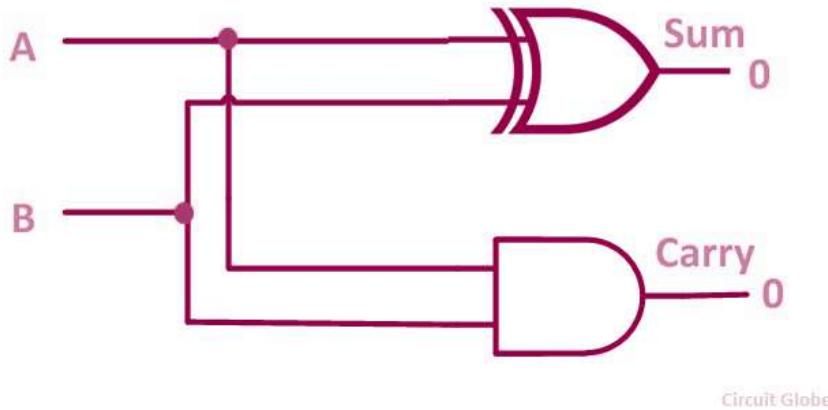
$$1 + 1 = 10$$

Here the output "1" of "10" becomes the carry-out. **SUM** is the normal output and the **CARRY** is the carry-out.

The **truth table** of the half adder is shown below.

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

The **Half Adder Circuit** is shown below.



The main disadvantage of this circuit is that it can only add two inputs and if there is any carry it is neglected. Thus, the process is incomplete. To overcome this difficulty Full Adder is designed. While performing complex addition, there may be cases when you have to add two 8 bit bytes together. This can be done with the help of Full Adder.

## Full Adder

The full adder is a little more difficult to implement than a half adder. The main difference between a half adder and a full adder is that the full adder has three inputs and two outputs. The two inputs are A and B, and the third input is a carry input  $C_{IN}$ . The output carry is designated as  $C_{OUT}$ , and the normal output is designated as S.

The **truth table** of the Full Adder Circuit is shown below.

Inputs			Outputs	
A	B	$C_{IN}$	$C_{OUT}$	S
0	0	0	0	0
0	0	1	0	1

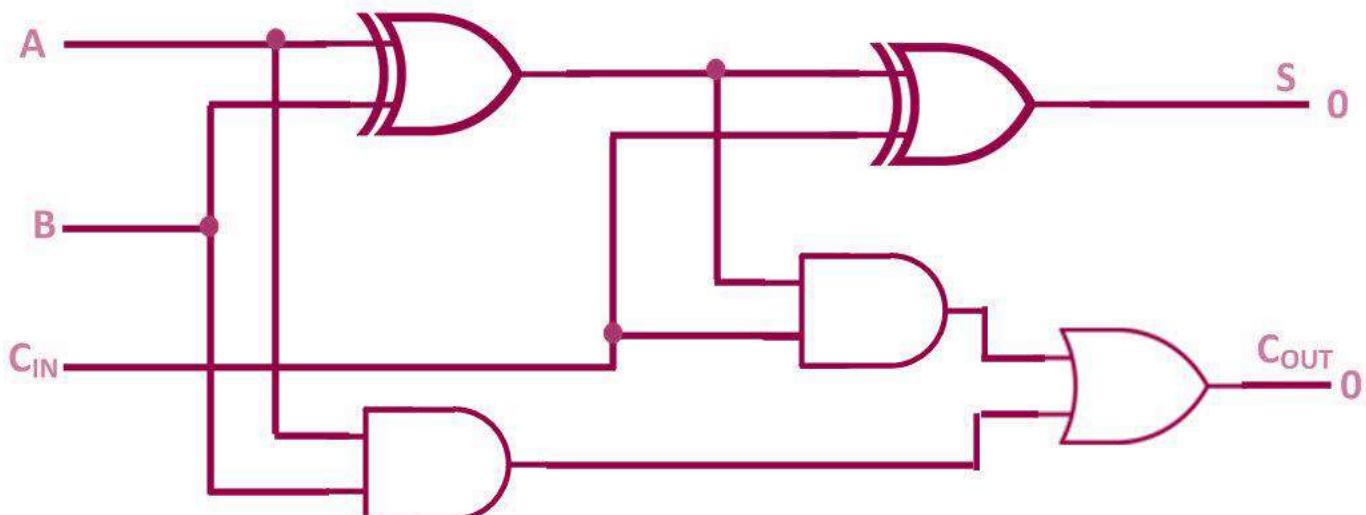
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

The output S is an EX – OR between the input A and the half adder SUM output B. The  $C_{OUT}$  will be true only if any of the two inputs out of the three are HIGH or at logic 1.

Thus, a full adder circuit can be implemented with the help of two half adder circuits. The first half adder circuit will be used to add A and B to produce a partial sum. The second half adder logic can be used to add  $C_{IN}$  to the sum produced by the first half adder circuit. Finally, the output S is obtained.

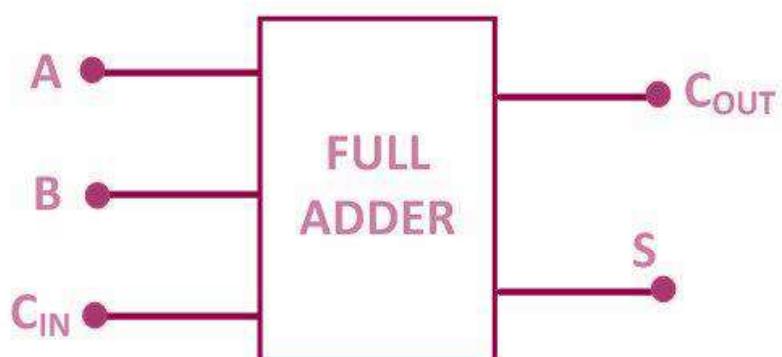
If any of the half adder logic produces a carry, there will be an output carry. Thus,  $C_{OUT}$  will be an OR function of the half adder CARRY outputs.

The **Full adder circuit diagram** is shown below.



Circuit Globe

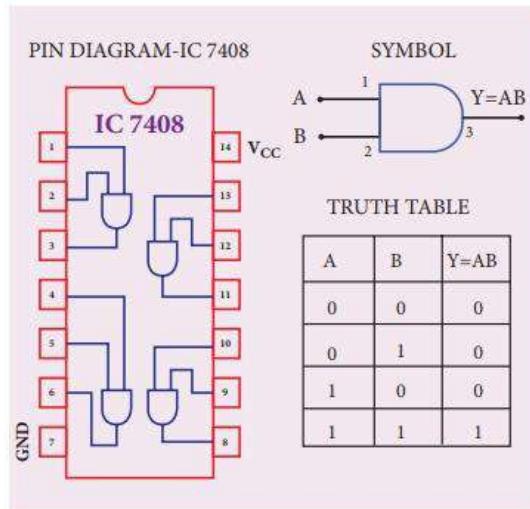
The schematic representation of a single bit Full Adder is shown below.



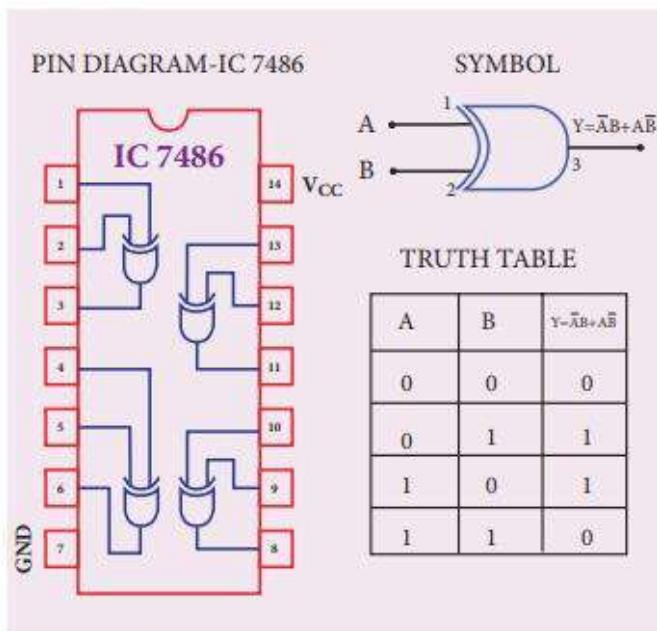
Circuit Globe

Circuit Symbol, pin diagram and Truth table of gates used:

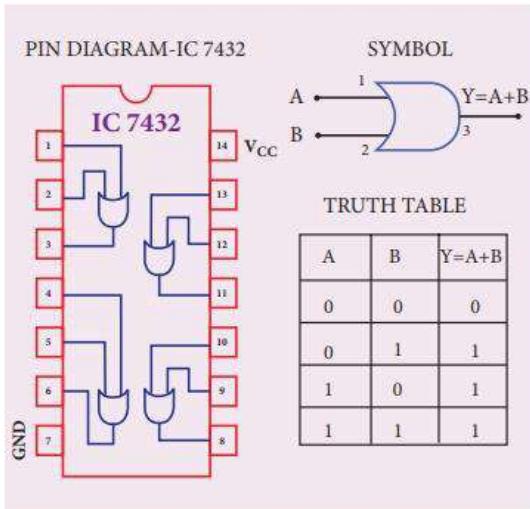
#### AND Gate:



#### X-OR Gate :



#### OR Gate:

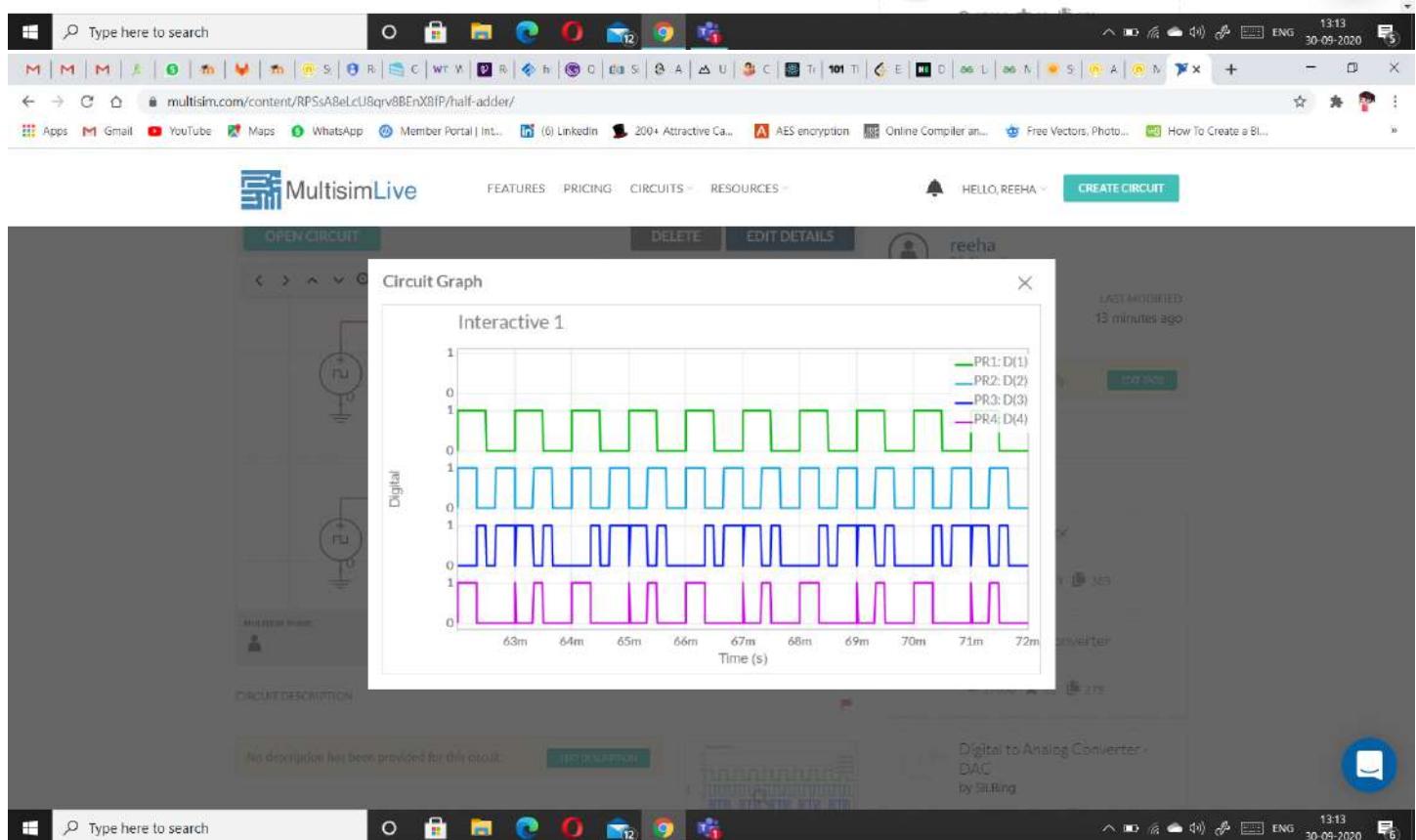
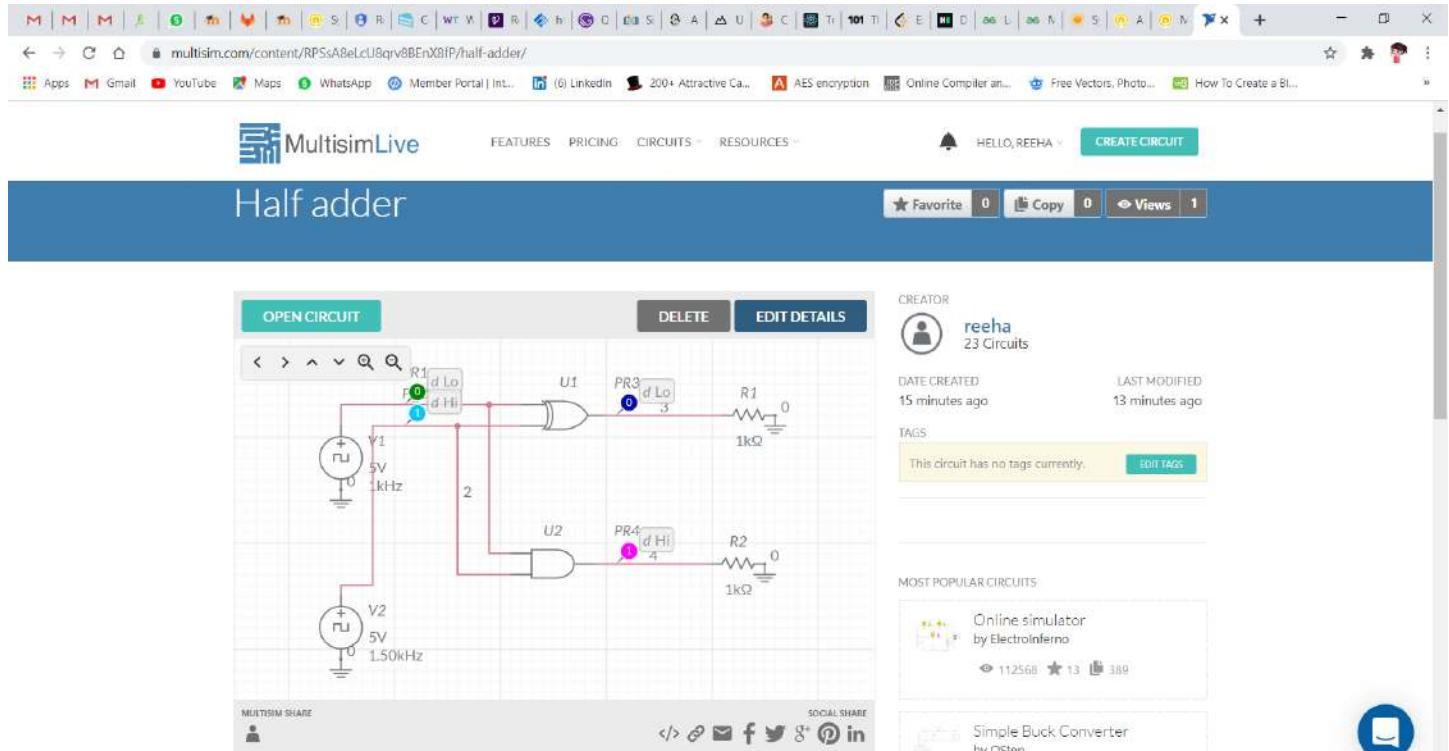


#### Procedure followed on MULTISIM:

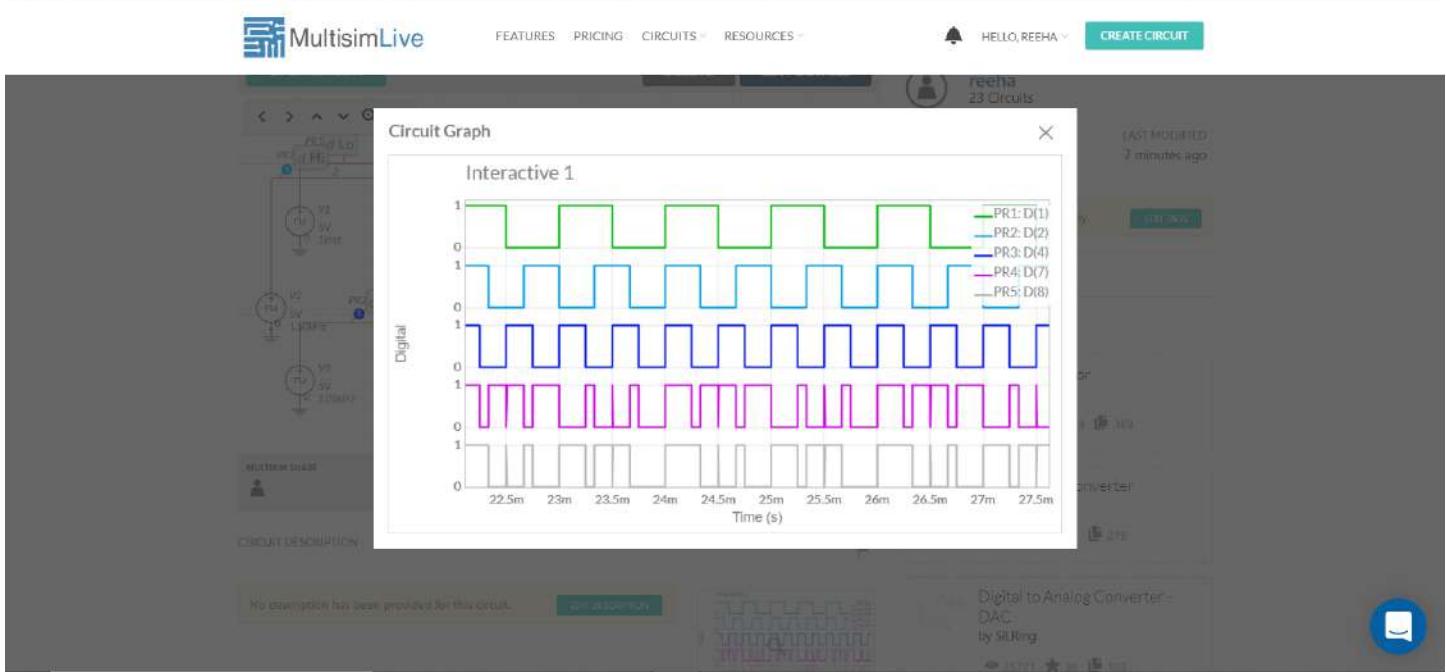
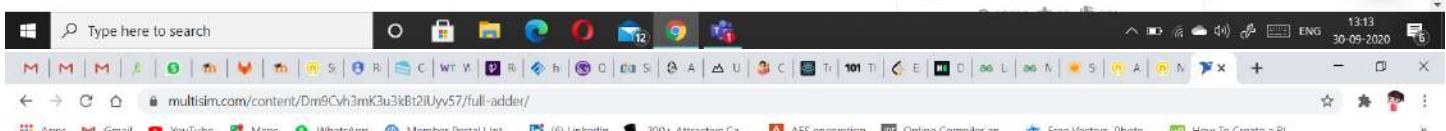
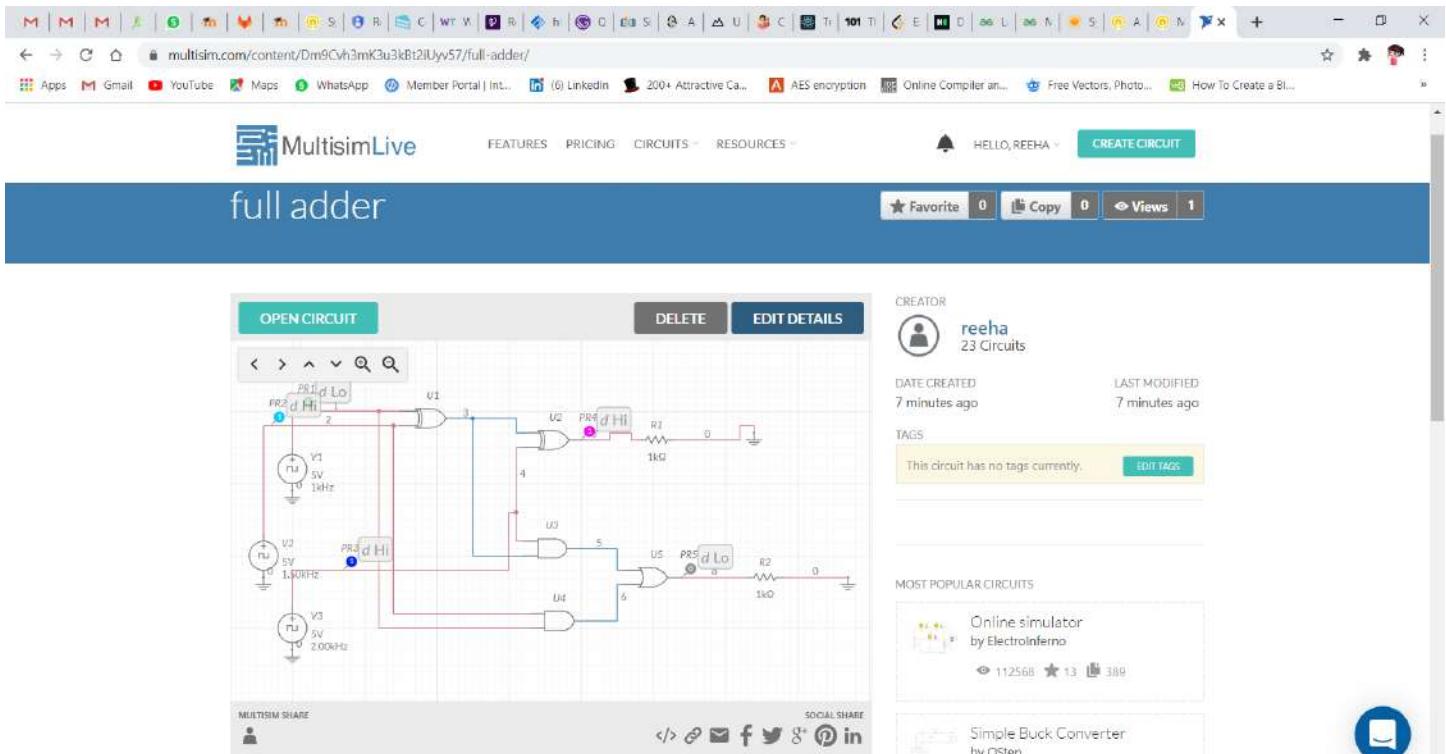
1. LOG IN ON [www.multisim.com](http://www.multisim.com)
2. CREATE THE CIRCUIT
3. SAVE THE CIRCUIT
4. SAVE THE SCREENSHOTS FOR
  - i. INPUT & OUTPUT WAVEFORMS (ALONG WITH YOUR ID ON TOP LEFT)
  - ii. CIRCUIT (ALONG WITH YOUR ID ON TOP LEFT)

# Circuits and Output waveform

# Half Adder



# Full Adder



## Viva Questions

### **Q1. Explain what is a half-adder?**

A logic circuit, that can add two 1-bit numbers and produce outputs for sum and carry, is called a half-adder.

### **Q2. Explain what is a full-adder?**

A binary adder, which can add two 1-bit binary numbers along with a carry bit and produces outputs for sum and carry is called a full-adder.

### **Q3. Explain what is an excitation table?**

Excitation table gives an information about what should be the flip-flop inputs if the outputs are specified before and after the clock pulses.

### **Q4. Explain what is a state table?**

State table consists of complete information about present state, next state, and outputs of a sequential circuit.

### **Q5. Explain what is Boolean Algebra?**

Boolean algebra is a mathematic system of logic in which truth functions are expressed as symbols and then these symbols are manipulated to arrive at conclusion.

# EXPERIMENT - 5

Switching Theory and Logic Design (STLD)

## Aim

To realize the circuit for Half Subtractor and Full Subtractor using logic gates.

Syeda Reeha Quasar

14114802719

3C7

# **EXPERIMENT - 5**

## **AIM:**

To realize the circuit for Half Subtractor and Full Subtractor using logic gates.

## **Hardware and Software Apparatus Required**

### **Hardware:**

- ❖ Power supply, Bread Board, Connecting Wires, respective IC, LED, Wire Cutter.
- ❖ Circuit is designed on bread board using Integrated Chips (ICs), Voltage supply and LEDs.
- ❖ The set-up of apparatus and working of the circuit were demonstrated via recorded videos.

### **Software Simulation:**

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

# **Theory:**

Subtractor is an electronic logic circuit for calculating the difference between two binary numbers, the minuend and the number to be subtracted, the subtrahend (see table). A full subtractor performs this calculation with three inputs: minuend bit, subtrahend bit, and borrow bit.

## **Half Subtractor**

Half subtractor is the most essential combinational logic circuit which is used in digital electronics. Basically, this is an electronic device or in other terms, we can say it as a logic circuit. Half subtractor is used to perform two binary digits subtraction. In the previous article, we have already discussed the concepts of half adder and a full adder circuit which uses the binary numbers for the calculation. Similarly, the subtractor circuit uses binary numbers (0,1) for the subtraction. The circuit of the half subtractor can be built with two logic gates namely NAND and EX-OR gates. This circuit gives two elements such as the difference as well as the borrow.

As in binary subtraction, the major digit is 1, we can generate borrow while the subtrahend 1 is superior to minuend 0 and due to this, borrow will need. The following example gives the binary subtraction of two binary bits.

First Digit	Second Digit	Difference	Borrow
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	0

In the above subtraction, the two digits can be represented with A and B. These two digits can be subtracted and gives the resultant bits as difference and borrow.

When we observe the first two and fourth rows, the difference between these rows, then the difference and borrow are similar because the subtrahend is lesser than the minuend. Similarly, when we observe the third row, the minuend value is subtracted from the subtrahend. So the difference and borrow bits are 1 because the subtrahend digit is superior to the minuend digit.

Half subtractor is an essential tool for any kind of [digital circuit](#) to know the possible combinations of inputs and outputs. For instance, if the subtractor has two inputs then the resultant outputs will be four. The o/p of the half subtractor is mentioned in the below table that will signify the difference bit as well as borrow bit. The half subtractor truth table explanation can be done by using the logic gates like EX-OR logic gate and AND gate operation followed by NOT gate.

Solving the truth table using **K-Map** is shown below.

A	0	1
B	0	0
	1	1

$$\text{Borrow} = \bar{A} \cdot B$$

A	0	1
B	0	0
	1	1

$$\text{Difference} = A \oplus B$$

## half subtractor k map

The Boolean expression of the half subtractor using truth table and K-map can be derived as

$$\text{Difference (D)} = (x'y + xy')$$

$$= x \oplus y$$

$$\text{Borrow (B)} = x'y$$

### Half-Subtractor Block Diagram

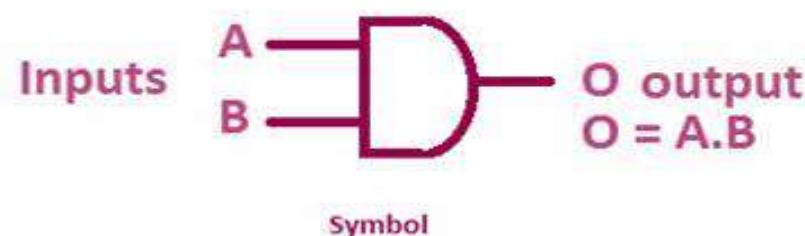
The block diagram of the half subtractor is shown above. It requires two inputs as well as gives two outputs. Here inputs are represented with A&B, and outputs are Difference and Borrow.

The above circuit can be designed with EX-OR & NAND gates. Here, the NAND gate can be build by using AND and NOT gates. So we require three logic gates for making half a subtractor circuit namely the EX-OR gate, NOT gate, and NAND gate.

A combination of AND and NOT gate produce a different combined gate named NAND Gate. The Ex-OR gate output will be the Difference bit and the NAND Gate output will be the Borrow bit for the same inputs A&B.

### AND-Gate

The AND-gate is one type of digital logic gate with multiple inputs and a single output and based on the inputs combinations it will perform the logical conjunction. When all the inputs of this gate are high, then the output will be high otherwise the output will be low. The logic diagram of AND gate with truth table is shown below.

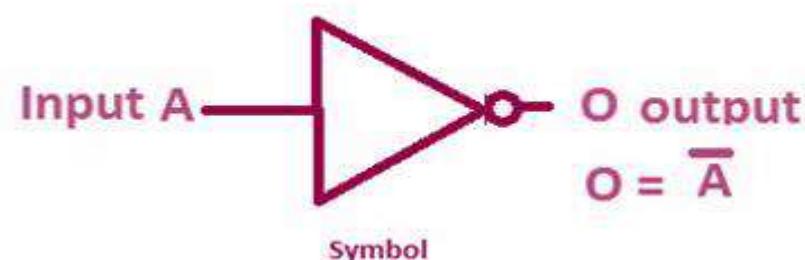


Inputs		Output
A	B	O
0	0	0
0	1	0
1	0	0
1	1	1

**Truth table**

### NOT Gate

The NOT-gate is one type of digital logic gate with a single input and based on the input the output will be reversed. For instance, when the input of the NOT gate is high then the output will be low. The logic diagram of NOT-gate with truth table is shown below. By using this type of logic gate, we can execute NAND and NOR gates.

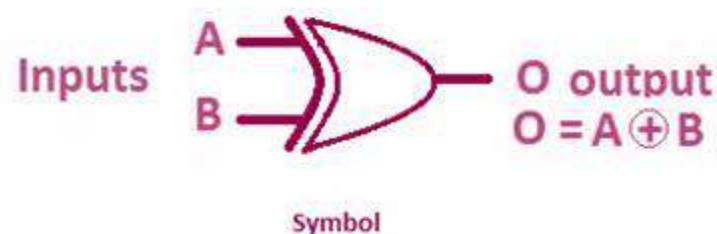


Inputs		Output
A	O	
0	1	
1	0	

**Truth table**

### Ex-OR Gate

The Exclusive-OR or EX-OR gate is one type of digital logic gate with 2-inputs & single output. The working of this logic gate depends on OR gate. If any one of the inputs of this gate is high, then the output of the EX-OR gate will be high. The symbol and truth table of the EX-OR are shown below.



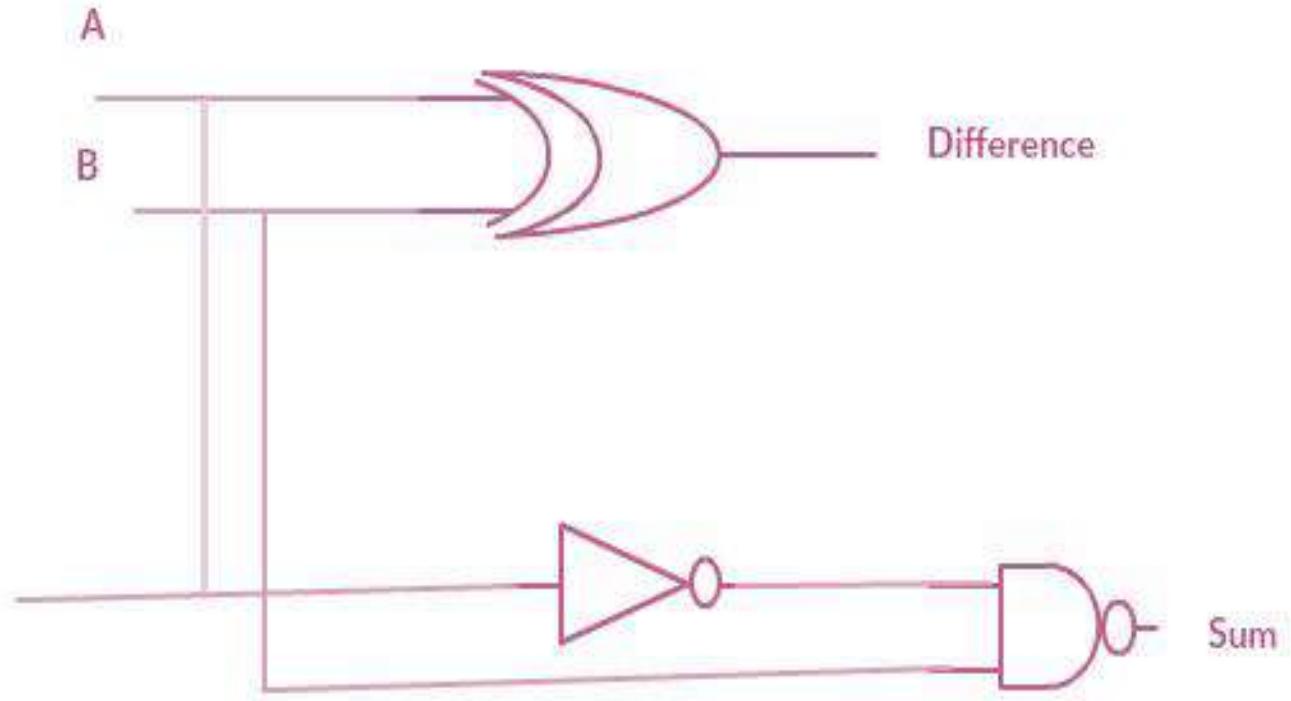
Inputs		Output
A	B	O
0	0	0
0	1	1
1	0	1
1	1	0

**Truth table**

**EXOR Gate and its Truth Table**

**Half Subtractor Circuit using Nand Gate**

The designing of half subtractor can be done by using logic gates like NAND gate & Ex-OR gate. In order to design this half subtractor circuit, we have to know the two concepts namely difference and borrow.



Half Subtractor Circuit using Nand Gate

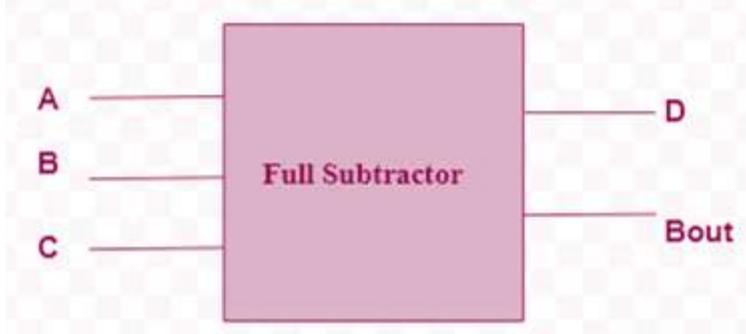
If we monitor cautiously, it is fairly clear that the variety of operation executed by this circuit which is accurately related to the EX-OR gate operation. Therefore, we can simply use the EX-OR gate for making difference. In the same way, the borrow produced by half adder circuit can be simply attained by using the blend of logic gates like AND- gate and NOT-gate.

#### Truth Table

First Bit	Second Bit	Difference	Borrow
		(EX-OR Out)	(NAND Out)
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	0

## Full Subtractor

Full subtractor is an electronic device or logic circuit which performs subtraction of two binary digits. It is a combinational logic circuit used in digital electronics. Many combinational circuits are available in integrated circuit technology namely adders, encoders, decoders and multiplexers. In this article, we are going to discuss full subtractor construction using half subtractor and also the terms like truth table.



A full subtractor is formed by two half subtractors, which involves three inputs such as minuend, subtrahend and borrow, borrow bit among the inputs is obtained from subtraction of two binary digits and is subtracted from next higher order pair of bits, outputs as difference and borrow.

### Full Subtractor Block Diagram

The foremost disadvantage of the half subtractor is, we cannot make a Borrow bit in this subtractor. Whereas in full subtractor design, actually we can make a Borrow bit in the circuit & can subtract with remaining two i/p's. Here A is minuend, B is subtrahend & Bin is borrow in. The outputs are Difference (Diff) & Bout (Borrow out). The complete subtractor circuit can obtain by using two half subtractors with an extra OR gate.



©www.elprocus.com

### Full Subtractor Circuit Diagram with Logic Gates

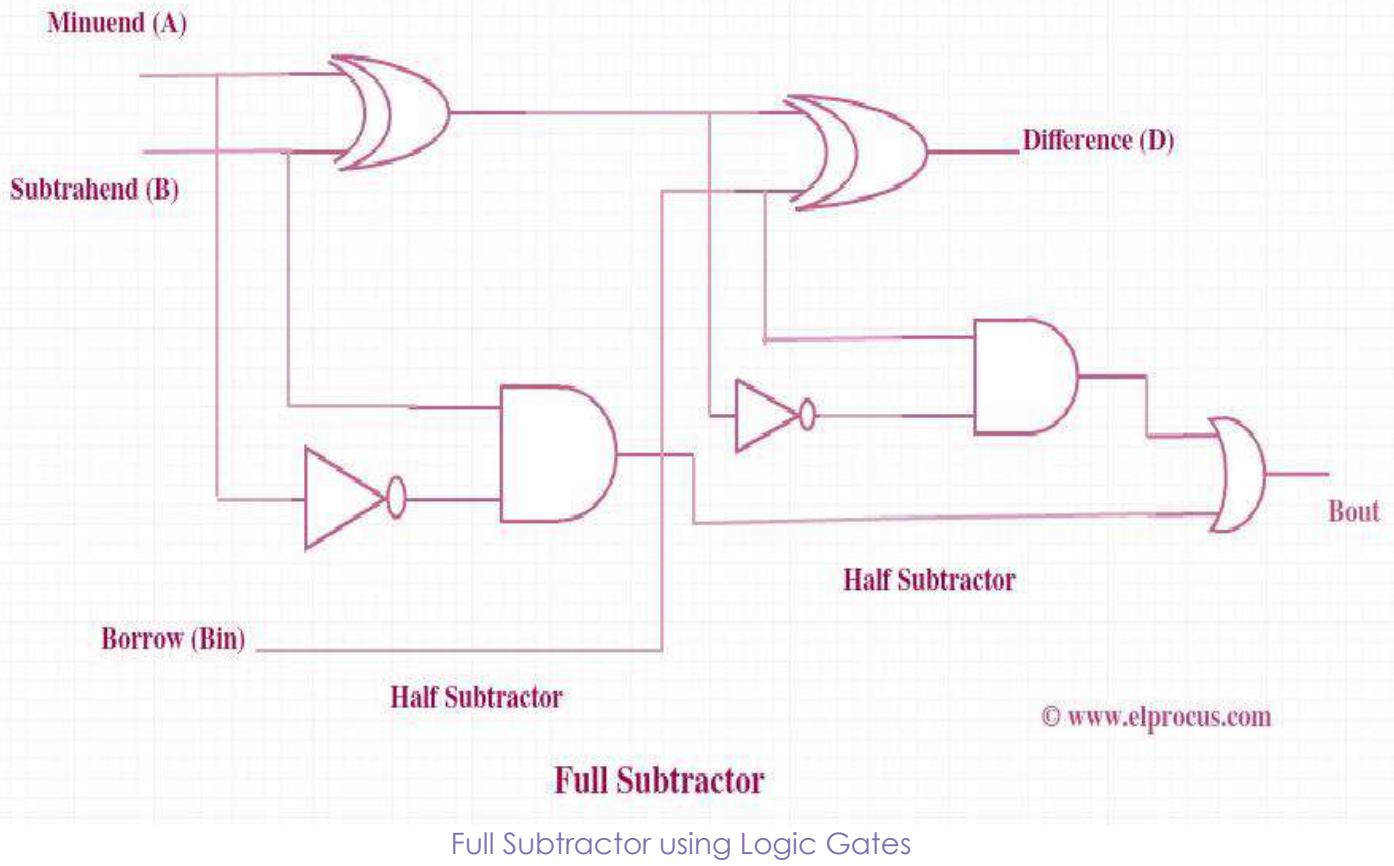
The circuit diagram of full subtractor using basic gates is shown in the following block diagram. This circuit can be done with two half-Subtractor circuits.

In the initial half-Subtractor circuit, the binary inputs are A and B. As we have discussed in the previous half-Subtractor article, it will generate two outputs namely difference (Diff) & Borrow.

The difference o/p of the left subtractor is given to the Left half-Subtractor circuit's. Diff output is further provided to the input of the right half Subtractor circuit. We offered the Borrow in bit across the other i/p of next half subtractor circuit. Once more it will give Diff out as well as Borrow out the bit. The final output of this subtractor is Diff output.

On the other hand, the Borrow out of both the half Subtractor circuits is connected to OR logic gate. Later than giving out OR logic for two output bits of the subtractor, we acquire the final Borrow out of the subtractor. The last Borrow out to signify the MSB (a most significant bit).

If we observe the internal circuit of the full Subtractor, we can see two Half SubTRACTORS with NAND gate and XOR gate with an extra OR gate.



### Full Subtractor Truth Table

This subtractor circuit executes a subtraction between two bits, which has 3- inputs (A, B and Bin) and two outputs (D and Bout). Here the inputs indicate minuend, subtrahend, & previous borrow, whereas the two outputs are denoted as borrow o/p and difference. The following image shows the truth table of full-subtractor.

Inputs			Outputs	
Minuend (A)	Subtrahend (B)	Borrow (Bin)	Difference (D)	Borrow (Bout)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

## Full Subtractor K-Map

The simplification of the K-map for the above difference and borrow is shown below.

		Bin				
		00	01	11	10	
A		0	0	1	0	1
1		1	0	1	0	

© www.elprocus.com

		Bin				
		00	01	11	10	
A		0	0	1	1	1
1		0	0	1	0	

© www.elprocus.com

The full subtractor equations for the difference as well as Bin are mentioned below.

The full subtractor expression for Difference is,

$$D = A'B'Bin + AB'Bin' + A'BBin' + ABBin$$

The full-subtractor expression for Borrow is,

$$Bout = A'Bin + A'B + BBin$$

## Applications of Full Subtractor

Some of the applications of full-subtractor include the following

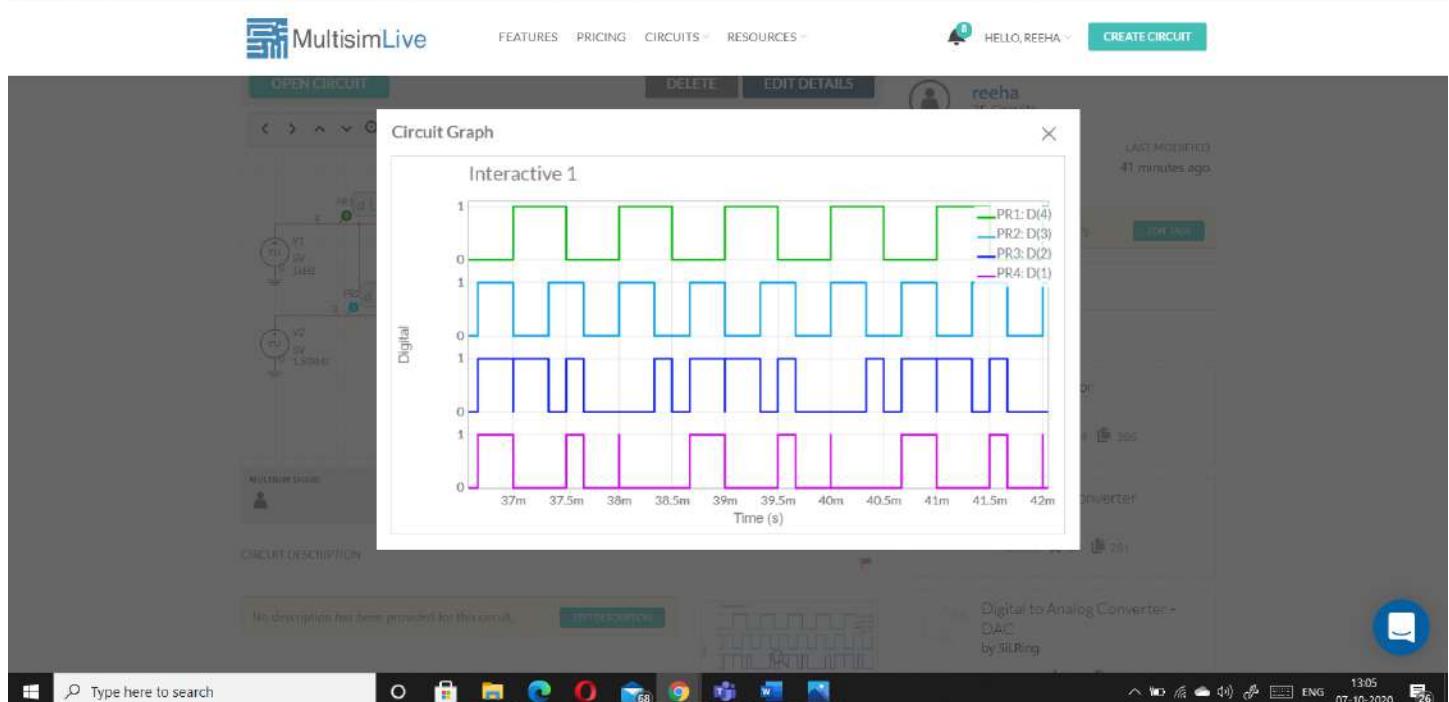
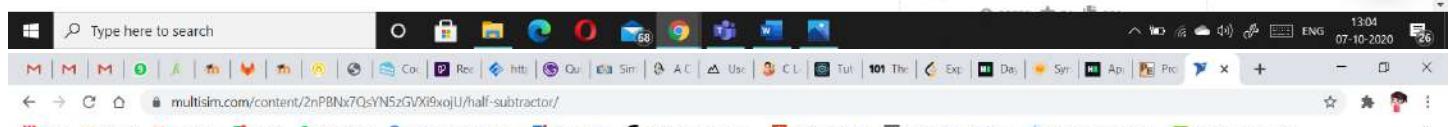
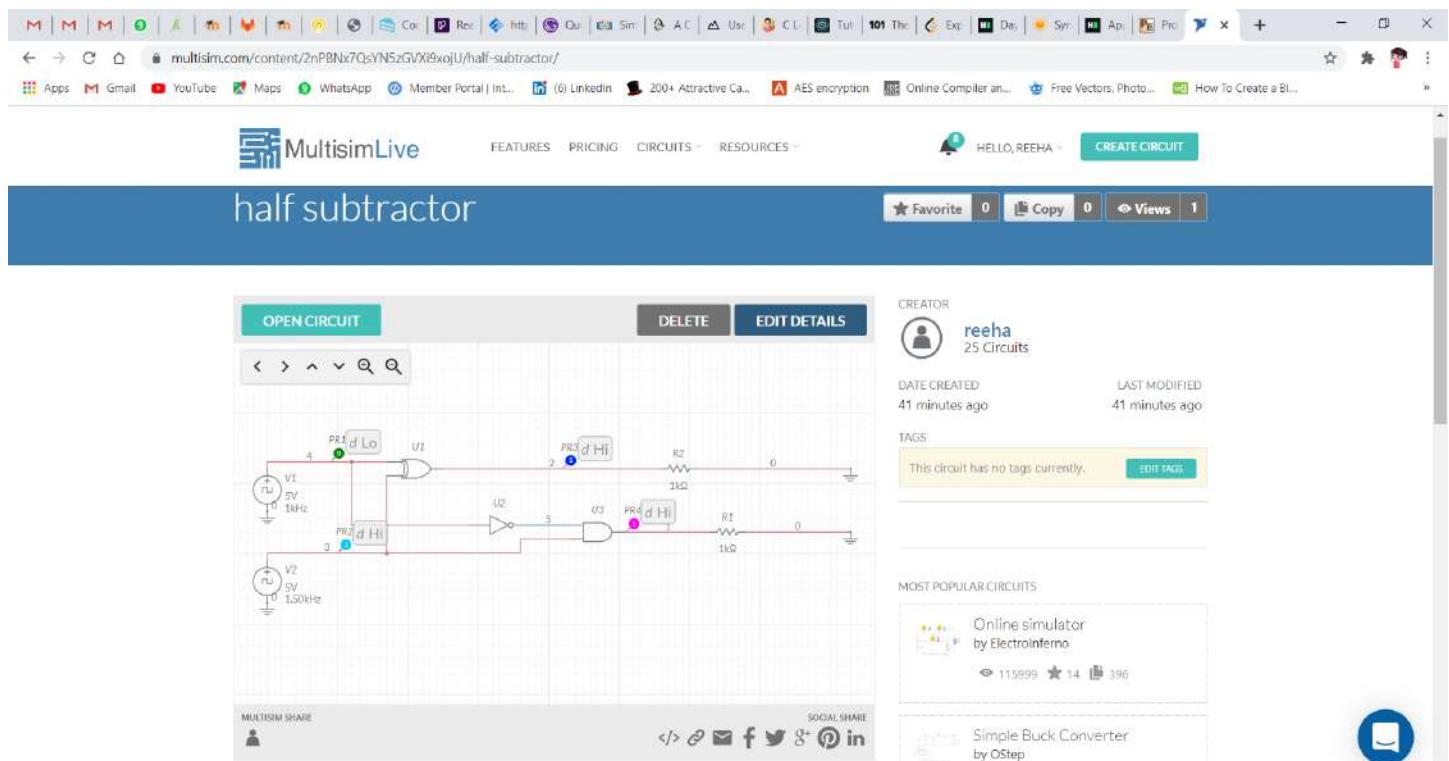
- These are generally employed for ALU (Arithmetic logic unit) in computers to subtract as CPU & GPU for the applications of graphics to decrease the circuit difficulty.
- Subtractors are mostly used for performing arithmetical functions like subtraction, in electronic calculators as well as digital devices.
- These are also applicable for different microcontrollers for arithmetic subtraction, timers, and program counter (PC)
- Subtractors are used in processors to compute tables, address, etc.
- It is also useful for DSP and networking based systems.

## Procedure followed on MULTISIM:

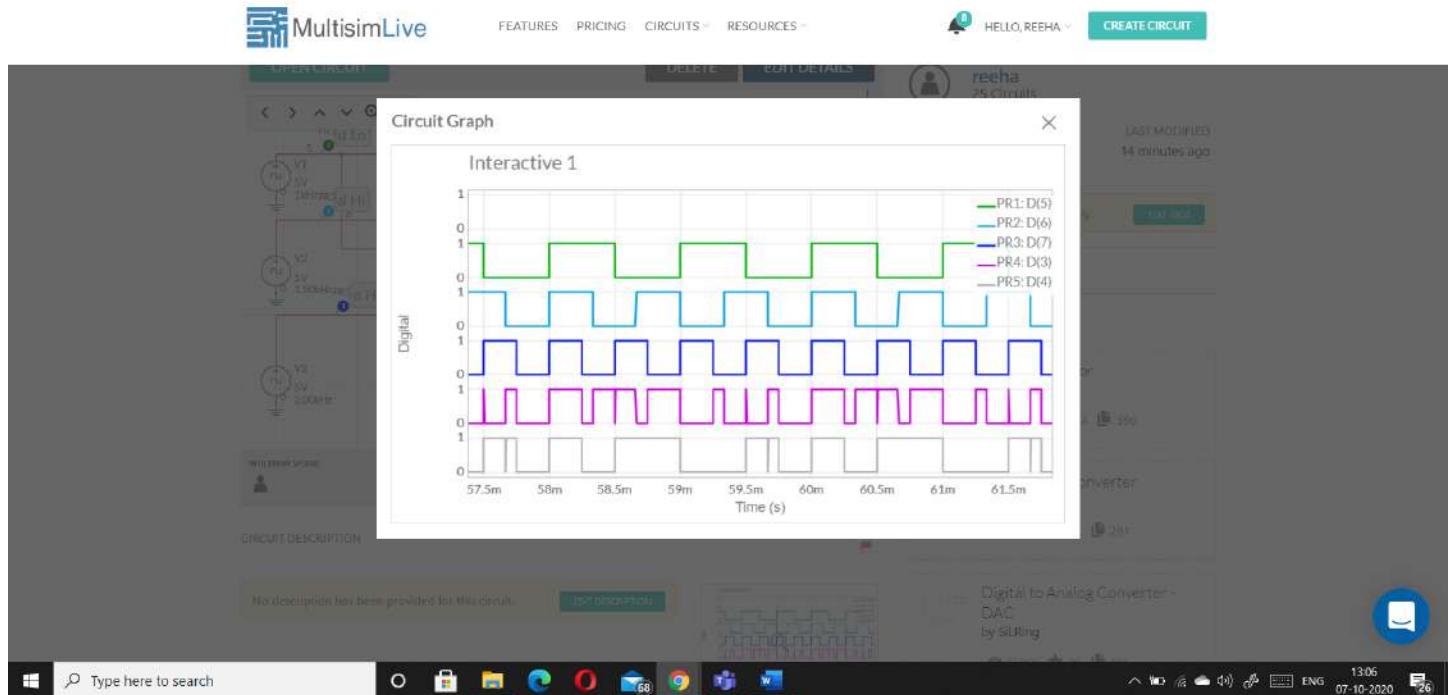
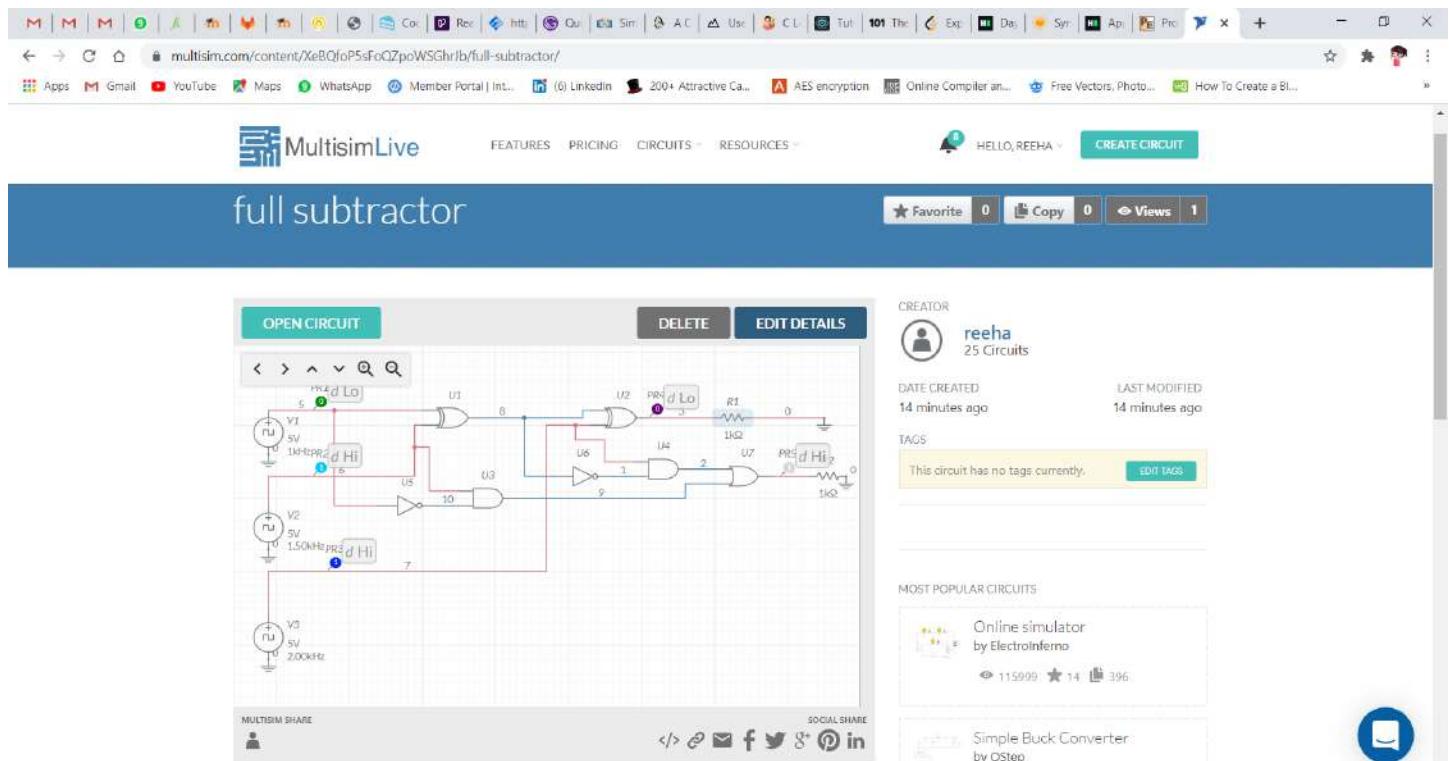
1. LOG IN ON [www.multisim.com](http://www.multisim.com)
2. CREATE THE CIRCUIT
3. SAVE THE CIRCUIT
4. SAVE THE SCREENSHOTS FOR
  - i. INPUT & OUTPUT WAVEFORMS (ALONG WITH YOUR ID ON TOP LEFT)
  - ii. CIRCUIT (ALONG WITH YOUR ID ON TOP LEFT)

## Circuits and Output waveform

# Half Subtractor



# Full Subtractor



## **Precautions (MULTISIM):**

1. Frequency of clock voltage source should be different for both inputs.
2. Place the probes carefully only at the input and output sources.
3. Use digital analyzer probe.
4. Set the type to transient.
5. Ground both the voltage sources(clock) and the resistor.

## **VIVA VOCE**

1. Half subtractor is used to perform subtraction of \_\_\_\_\_
  - a) 2 bits
  - b) 3 bits
  - c) 4 bits
  - d) 5 bits

Ans.

Answer: a

Explanation: Half subtractor is a combinational circuit which is used to perform subtraction of two bits, namely minuend and subtrahend and produces two outputs, borrow and difference.

2. For subtracting 1 from 0, we use to take a \_\_\_\_\_ from neighbouring bits.
  - a) Carry
  - b) Borrow
  - c) Input
  - d) Output

Ans.

Answer: b

Explanation: For subtracting 1 from 0, we use to take a borrow from neighbouring bits because carry is taken into consideration during addition process.

3. How many outputs are required for the implementation of a subtractor?
  - a) 1
  - b) 2
  - c) 3
  - d) 4

Ans.

Answer: b

Explanation: There are two outputs required for the implementation of a subtractor. One for the difference and another for borrow.

**4. Let the input of a subtractor is A and B then what the output will be if A = B?**

- a) 0
- b) 1
- c) A
- d) B

Ans.

Answer: a

Explanation: The output for A = B will be 0. If A = B, it means that A = B = 0 or A = B = 1. In both of the situation subtractor gives 0 as the output.

**5. Let A and B is the input of a subtractor then the output will be \_\_\_\_\_**

- a) A XOR B
- b) A AND B
- c) A OR B
- d) A EXNOR B

Ans.

Answer: a

Explanation: The subtractor has two outputs BORROW and DIFFERENCE. Since, the difference output of a subtractor is given by  $AB' + BA'$  and this is the output of a XOR gate. So, the final difference output is  $AB' + BA'$ .

# EXPERIMENT - 6

Switching Theory and Logic Design (STLD)

## Aim

To realize priority encoder using basic Gates.

Syeda Reeha Quasar

14114802719

3C7

# EXPERIMENT - 6

## AIM:

To realize priority encoder using basic Gates.

## Hardware and Software Apparatus Required

### Hardware:

Breadboard, IC 7408 (AND), IC 7404(NOT), IC 7432(OR), LEDs, 5V power supply, connecting wires.

- ❖ Power supply, Bread Board, Connecting Wires, respective IC, LED, Wire Cutter.
- ❖ Circuit is designed on bread board using Integrated Chips (ICs), Voltage supply and LEDs.
- ❖ The set-up of apparatus and working of the circuit were demonstrated via recorded videos.

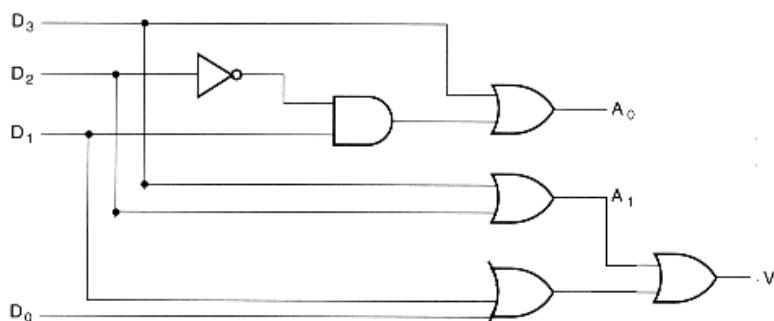
### Software Simulation:

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

## Theory:

The priority encoder is an encoder circuit that includes the priority function. The operation of priority encoder is such that if two or more inputs are equal to 1 at the same time the input having the highest priority will take precedence. The 4-input priority encoder has 4 inputs and 3 outputs. 'V' is a valid-bit indicator that is set to 1 when one or more inputs are equal to 1. If all inputs are zero, there is no valid input and V is equal to 0. The other two outputs are not specified when V equals 0 and are specified as don't care conditions.



D3	D2	D1	D0	A1	A0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

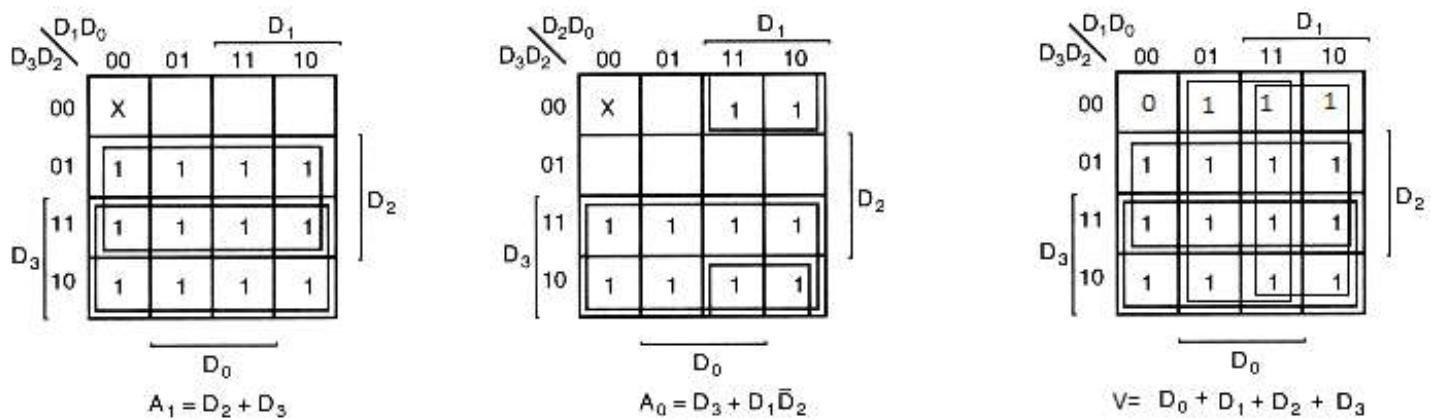


Fig 4.1 Logic diagram along with K-Map minimization

### Procedure followed on MULTISIM:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the logic diagram.
- Verify the results and observe the outputs.

### PRECAUTIONS:

All ICs should be checked before starting the experiment.

1. All the connection should be tight.
2. Always connect ground first and then the supply.
3. Switch off the power supply after completion of the experiment.

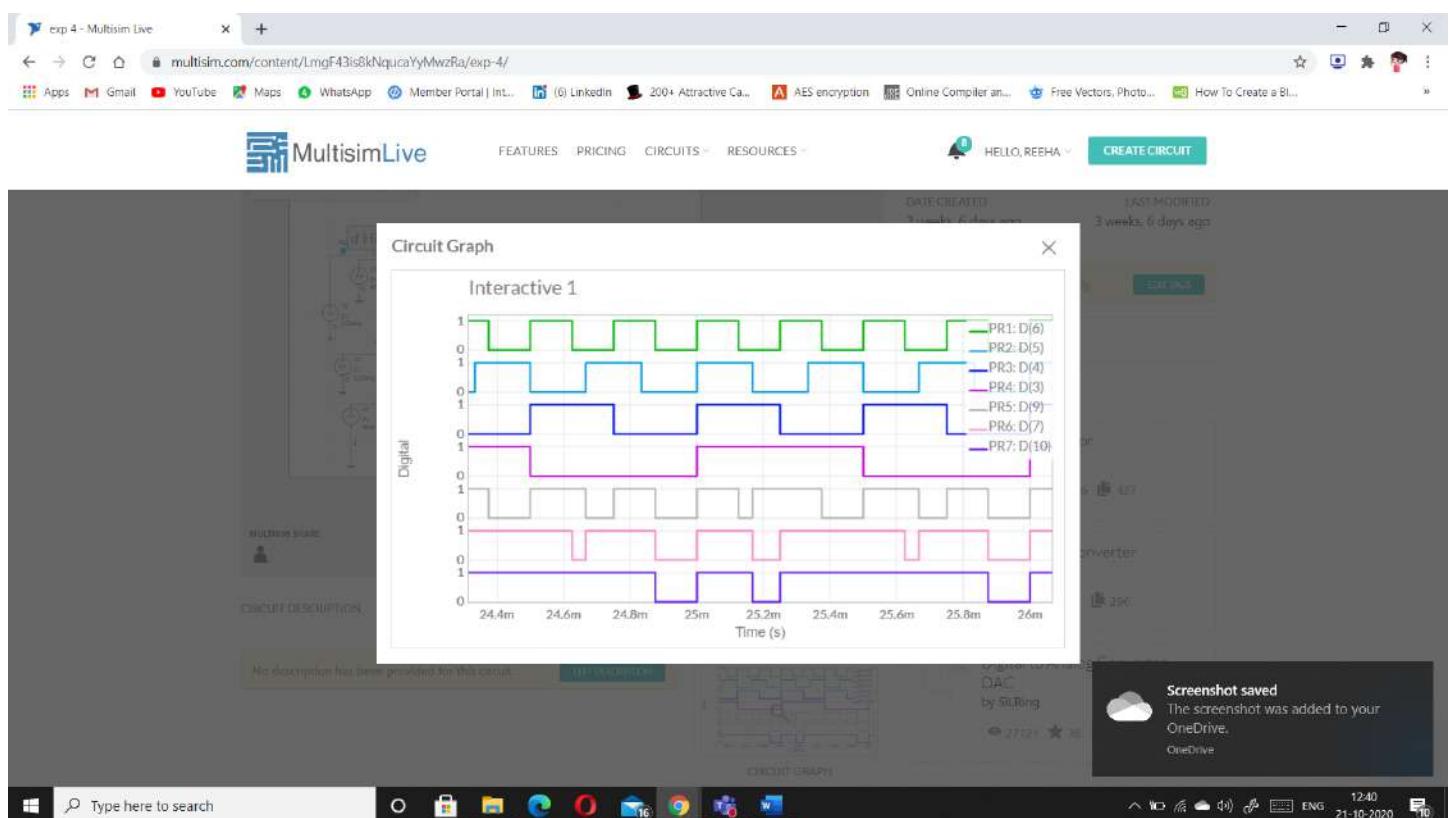
### RESULT:

Priority Encoder has been studied and its truth table has been verified.

## Circuits and Output waveform

# Priority Encoder

The screenshot shows a priority encoder circuit on MultisimLive. The circuit consists of four inputs (d<sub>4</sub>, d<sub>5</sub>, d<sub>6</sub>, d<sub>7</sub>) connected to a 4-to-1 multiplexer. The multiplexer's outputs are connected to a 3-to-2 decoder, which generates three enable signals (d<sub>Lo</sub>, d<sub>Hi</sub>, d<sub>Hi</sub>) for three 2-to-1 multiplexers. The outputs of these multiplexers are summed at a final stage. The circuit is powered by a 5V DC source. On the right, the creator's profile shows they have 25 circuits, created 3 weeks ago, and last modified 3 weeks ago. Below the circuit, there is a 'Circuit Graph' section showing digital waveforms for the inputs and outputs.



## **VIVA-VOCE QUESTIONS:**

### **1. What is an encoder?**

Ans.

An encoder in digital electronics is a one-hot to binary converter. That is, if there are  $2^n$  input lines, and at most only one of them will ever be high, the binary code of this 'hot' line is produced on the n-bit output lines. A binary encoder is the dual of a binary decoder.

### **2. State the difference between simple encoder and priority encoder.**

Ans.

An ordinary encoder has a number of input lines but only one of them is activated at a given time. A priority encoder can have more than one input activated at the same time.

Binary Encoder converts one of  $2^n$  inputs into an n-bit output. It has fewer output bits than the input code. The Priority Encoder is another type of combinational circuit similar to a binary encoder, except that it generates an output code based on the highest prioritized input.

### **3. Specify the applications of Encoder and decoder.**

Ans.

- Speed synchronization of multiple motors in industries
- War field flying robot with a night vision flying camera
- Robotic vehicle with the metal detector
- RF based home automation system
- Automatic health monitoring systems

### **4. How an encoder is different from a multiplexer?**

Ans.

A multiplexer or MUX is a combination circuit that contains more than one input line, one output line and more than one selection line. Whereas, an encoder is also considered a type of multiplexer but without a single output line. It is a combinational logic function that has  $2^n$  (or fewer) input lines and n output lines.

# EXPERIMENT - 7

Switching Theory and Logic Design (STLD)

## Aim

To realize binary to gray and gray to binary code converter.

Syeda Reeha Quasar

14114802719

3C7

# EXPERIMENT - 7

## AIM:

To realize binary to gray and gray to binary code converter.

## Hardware and Software Apparatus Required

### Hardware:

Breadboard, IC 7486 (XOR), LEDs, 5V power supply, connecting wires.

### Software Simulation:

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

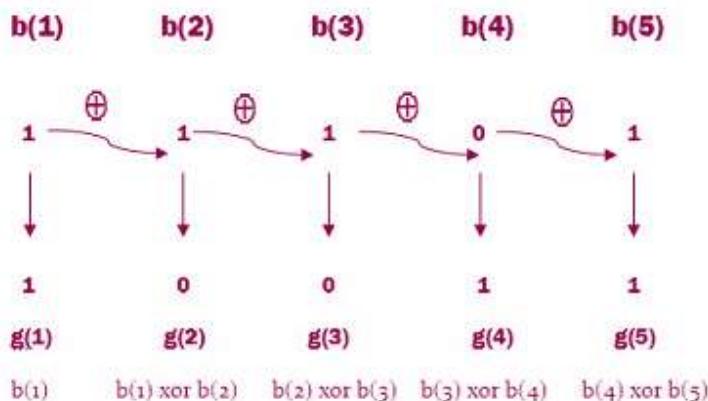
## Theory:

The gray code is a non weighted code. The successive gray code differs in one bit position only that means it is a unit distance code. It is also referred as cyclic code. It is not suitable for arithmetic operations. It is the most popular of the unit distance codes. It is also a reflective code. An n-bit Gray code can be obtained by reflecting an n-1 bit code about an axis after  $2^{n-1}$  rows, and putting the MSB of 0 above the axis and the MSB of 1 below the axis. The below solved examples may useful to understand how to perform binary to gray and gray to binary code conversion. This conversion method strongly follows the EX-OR gate operation between binary bits.

### Method of Conversion

**Convert the binary  $11101_2$  to its equivalent Gray code**

a)



b)

**Convert the Gray code 1010 to its equivalent Binary**

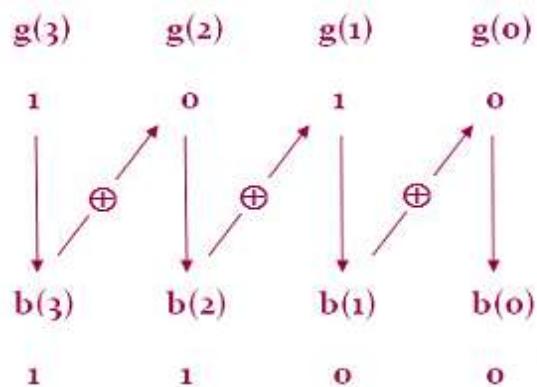
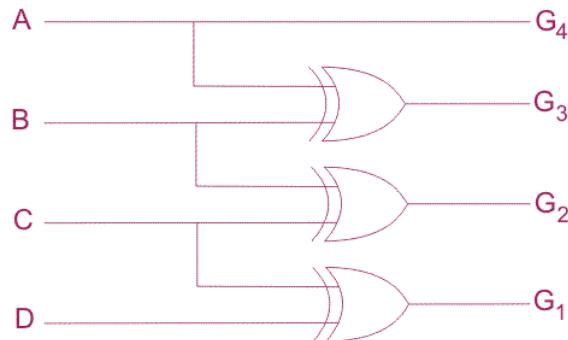


Fig 13.1 a) Binary to gray b) Gray to Binary

### Logic Diagrams

a)



b)

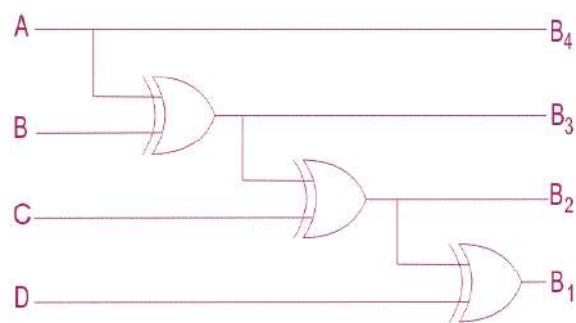


Fig 13.2 Logic Diagram of a) Binary to Gray b) Gray to Binary

## *Binary to Gray Code Converter Table*

Decimal Number	Binary Code	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

## Gray to Binary Code Converter Table

Decimal Number	Gray Code	Binary Code
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

### Procedure followed on MULTISIM:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the results and observe the outputs.

## PRECAUTIONS:

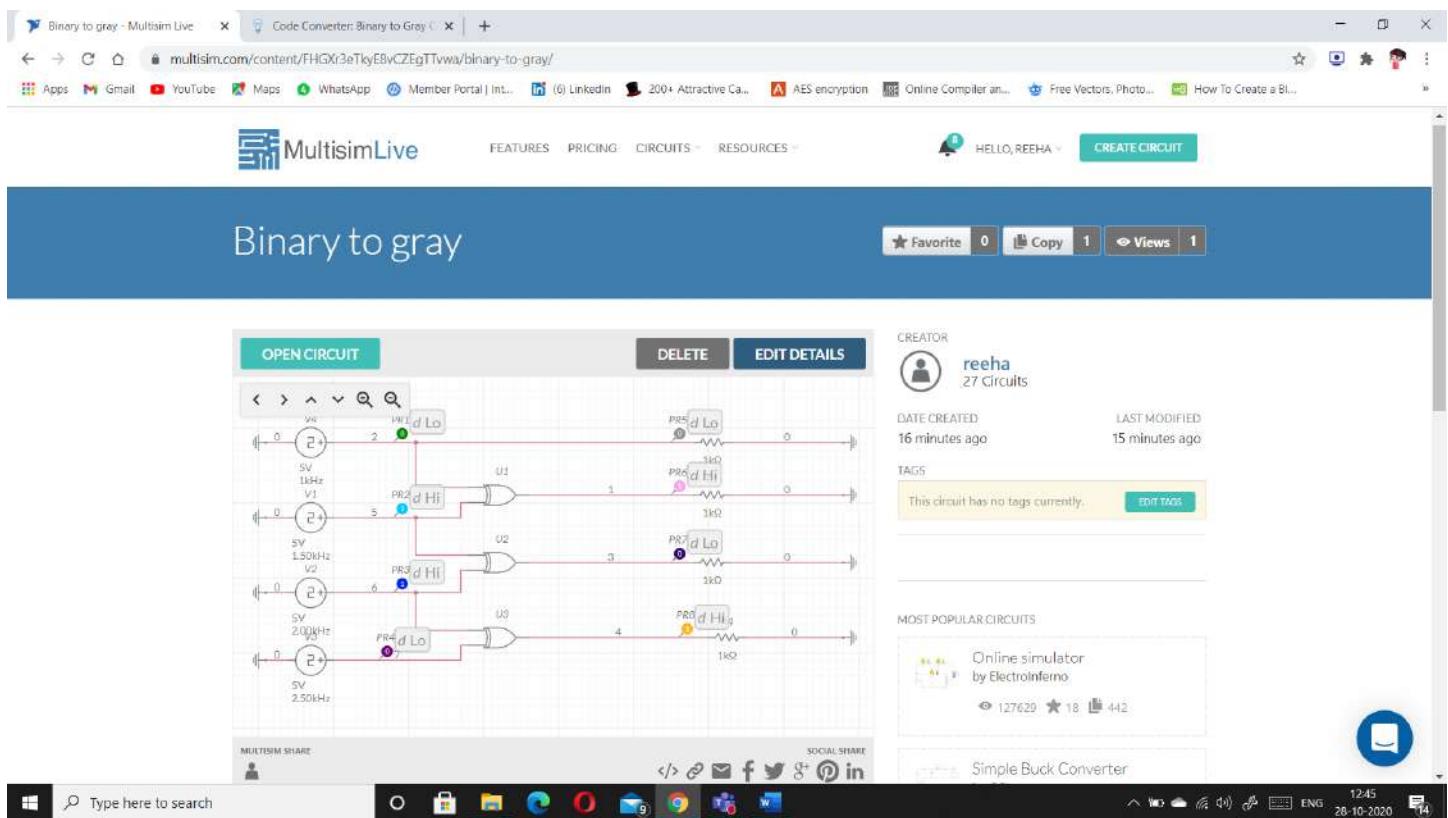
1. All ICs should be checked before starting the experiment.
2. All the connection should be tight.
3. Always connect ground first and then the supply.
4. Switch off the power supply after completion of the experiment.

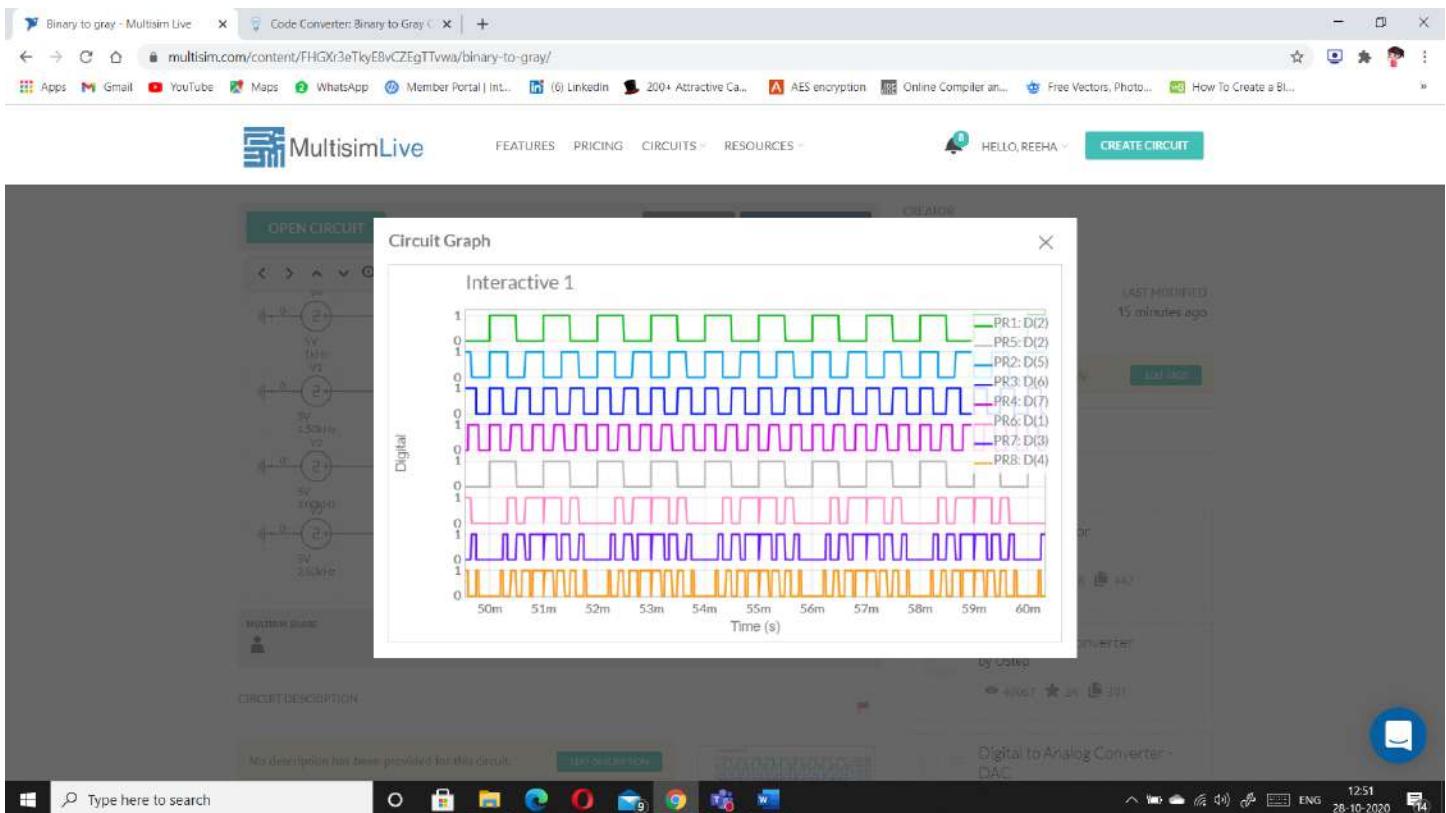
## RESULT:

Gray to binary and binary to Gray code converter has been studied and its truth table is verified.

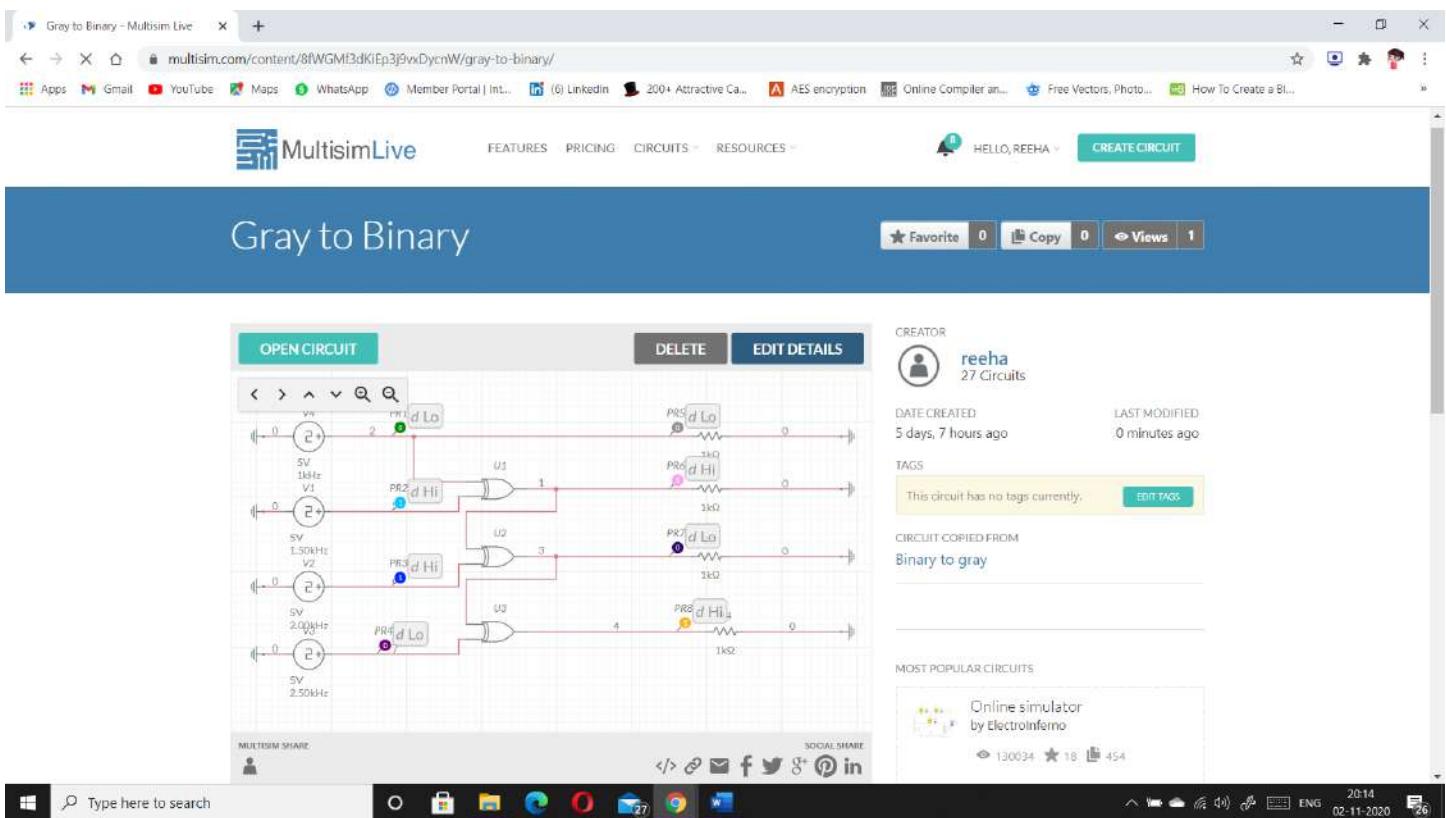
## Circuits and Output waveform

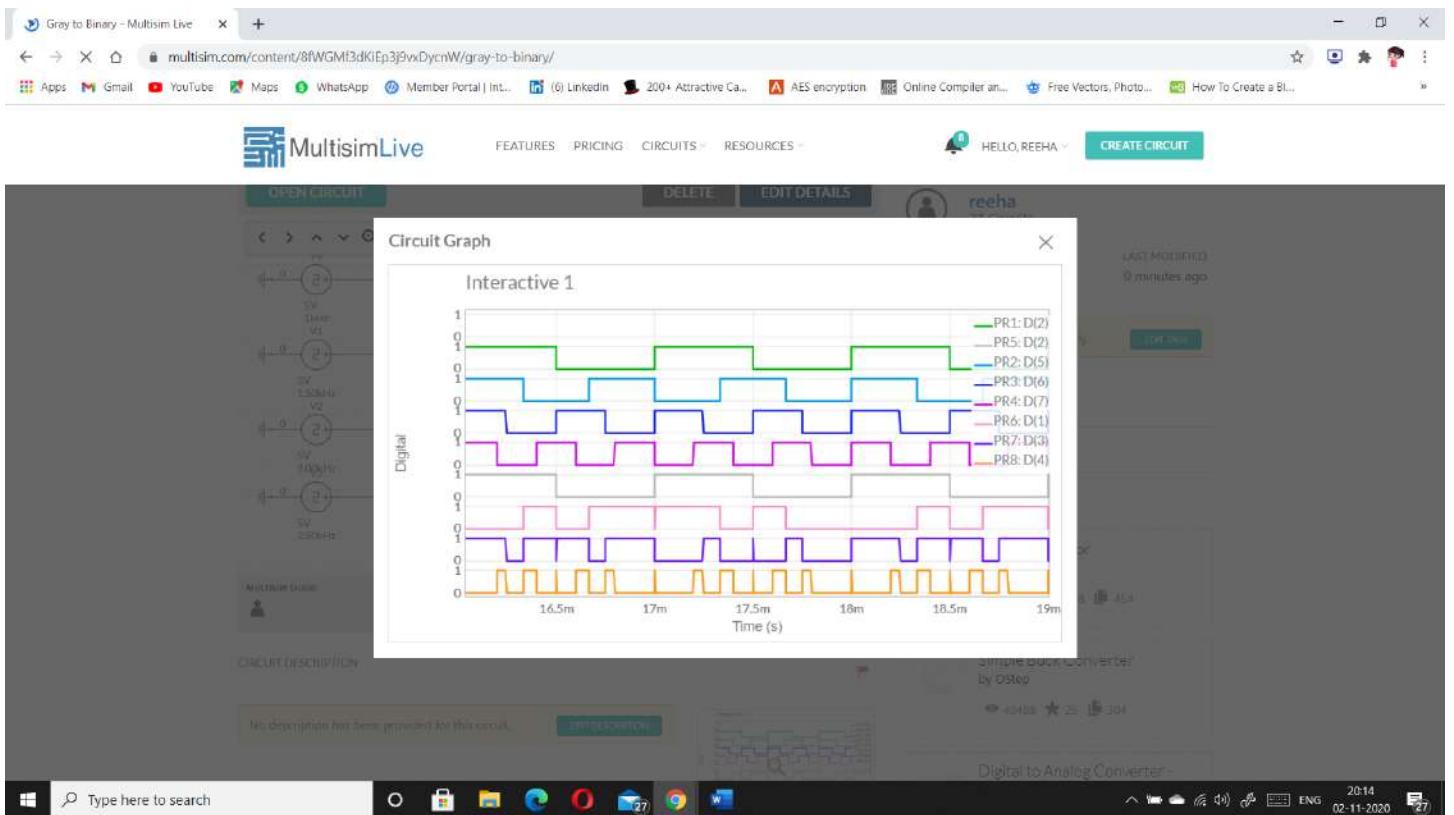
### Binary to Gray





## Gray to Binary





## VIVA-VOCE QUESTIONS:

### 1. Why Gray code are called Unit Distance Code?

Ans.

A unit distance code derives its name from the fact that there is only one bit change between two consecutive numbers. The excess 3 gray code is such a code, the values for zero and nine differ in only 1 bit, and so do all values for successive numbers.

### 2. What are applications of Gray Codes?

Ans.

Its used in digital electronics. For example an adder. Practical applications can be thought of any condition which requires not more than one bit to switch at a time. Even in terms of devices when we think at a level which is at a threshold of device physics and digital vlsi; if a cmos gate switches which is a part of series of other gates; its fine.

### 3. What are Reflected codes?

Ans.

The reflected binary code or Gray code is an ordering of the binary numeral system such that two successive values differ in only one bit (binary digit).

#### 4. Why Gray Coding is used in K- mapping instead of binary?

Ans.

Because Gray code only changes one bit at a time as you move between adjacent states, so it makes the groupings of terms possible, because they're next to each other. If you used binary code, the regions would be disjointed and the grouping of terms not obvious.

# EXPERIMENT - 8

Switching Theory and Logic Design (STLD)

## Aim

To realize 2 bit Magnitude Comparator.

Syeda Reeha Quasar

14114802719

3C7

# EXPERIMENT - 8

## AIM:

To realize 2 bit Magnitude Comparator.

## Hardware and Software Apparatus Required

### Hardware:

Breadboard, IC 7400 (NAND), IC 7410 (3 input NAND), IC 7404(NOT), LEDs, 5V power supply, connecting wires.

### Software Simulation:

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

## Theory:

Magnitude Comparator is a logical circuit, which compares two signals A and B and generates three logical outputs, whether  $A > B$ ,  $A = B$ , or  $A < B$ . The outcome of comparison is specified by three binary variables that indicate whether  $A > B$ ,  $A = B$ , or  $A < B$ . 2-Bit Magnitude Comparator Compares two numbers each having two bits ( $A_1, A_0$  &  $B_1, B_0$ ).

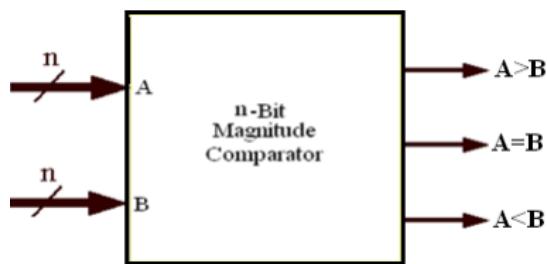


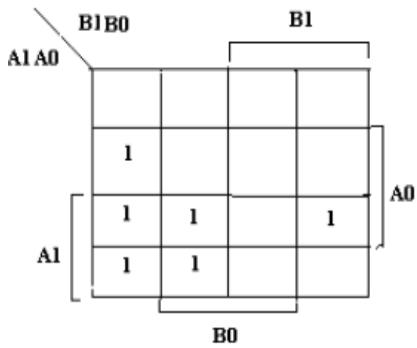
Fig 11.1 Block Diagram Of comparator

## Designing of 2 bit comparator

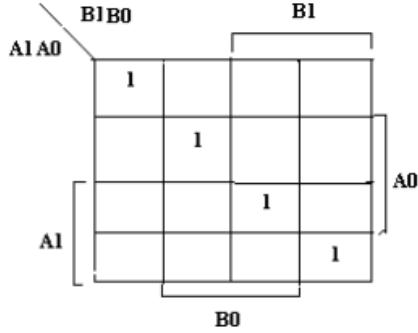
INPUT				OUTPUT		
A1	A0	B1	B0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

K-Map is used to minimize Boolean function obtained from truth table as shown below

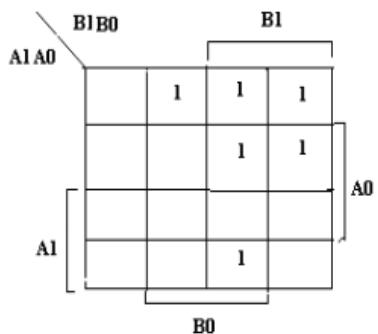
### a) FOR A>B



### FOR A=B



### FOR A<B



$$\begin{aligned}
 A>B &= A_1 B_1' + A_0 B_0' A_1' B_1' + A_0 B_0' A_1 B_1 \\
 &= A_1 B_1' + A_0 B_0' (A_1' B_1' + A_1 B_1) \\
 &= A_1 B_1' + A_0 B_0' X_1
 \end{aligned}$$

$$\begin{aligned}
 A=B &= A_1' A_0' B_1' B_0' + A_1' A_0 B_1' B_0 + A_1 A_0' B_1 B_0' + A_1 A_0 B_1 B_0 \\
 &= (A_1' B_1' + A_1 B_1) (A_0' B_0' + A_0 B_0) \\
 &= X_1 X_0
 \end{aligned}$$

$$\begin{aligned}
 A<B &= A_1' B_1 + A_0' B_0 A_1' B_1' + A_0' B_0 A_1 B_1 \\
 &= A_1' B_1 + A_0' B_0 (A_1' B_1' + A_1 B_1) \\
 &= A_1' B_1 + A_0' B_0 X_1
 \end{aligned}$$

b)

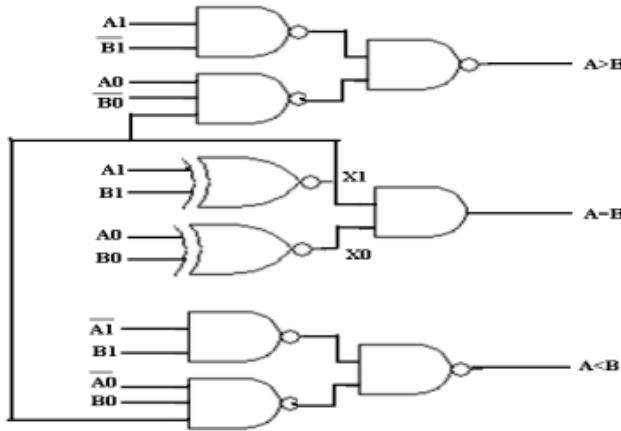


Fig 11.2 a)K Map minimization b) logic Diagram of comparator

From the above K-maps logical expressions for each output can be expressed as follows:

$$A>B: A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$$

$$A=B: A_1'A_0'B_1'B_0' + A_1'A_0B_1'B_0 + A_1A_0B_1B_0 + A_1A_0'B_1B_0'$$

$$: A_1'B_1' (A_0'B_0' + A_0B_0) + A_1B_1 (A_0B_0 + A_0'B_0')$$

$$: (A_0B_0 + A_0'B_0') (A_1B_1 + A_1'B_1')$$

$$: (A_0 \text{ Ex-Nor } B_0) (A_1 \text{ Ex-Nor } B_1)$$

$$A<B: A_1'B_1 + A_0'B_1B_0 + A_1'A_0'B_0$$

### Procedure:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the results and observe the outputs.

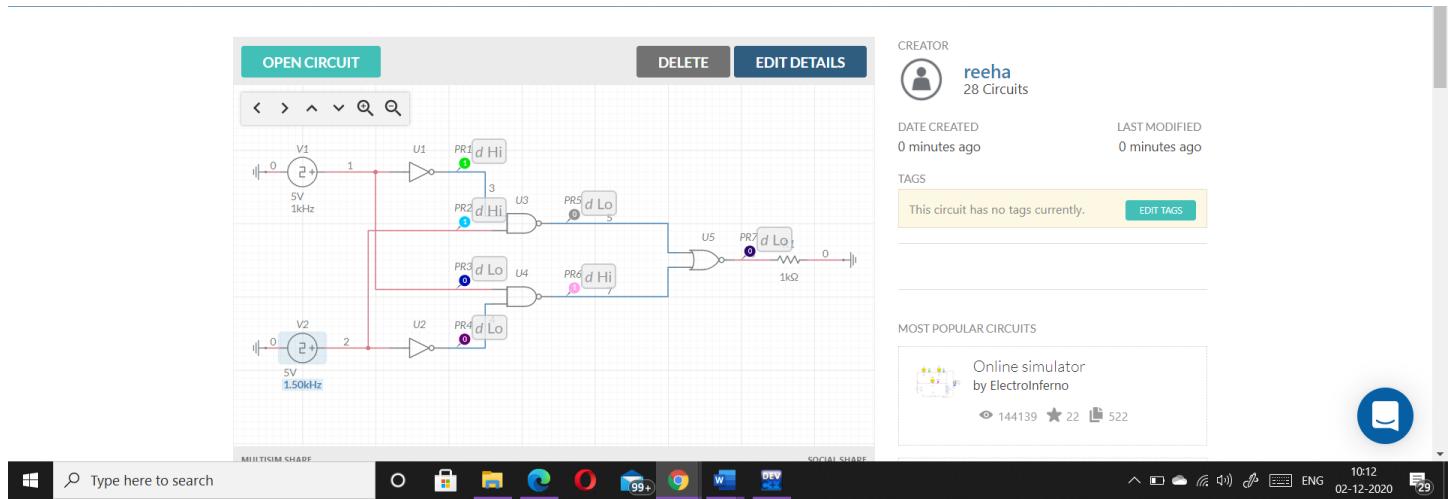
### PRECAUTIONS:

1. All ICs should be checked before starting the experiment.
2. All the connection should be tight.
3. Always connect ground first and then the supply.
4. Switch off the power supply after completion of the experiment.

### RESULT:

2 bit magnitude comparator has been studied and its truth table is verified.

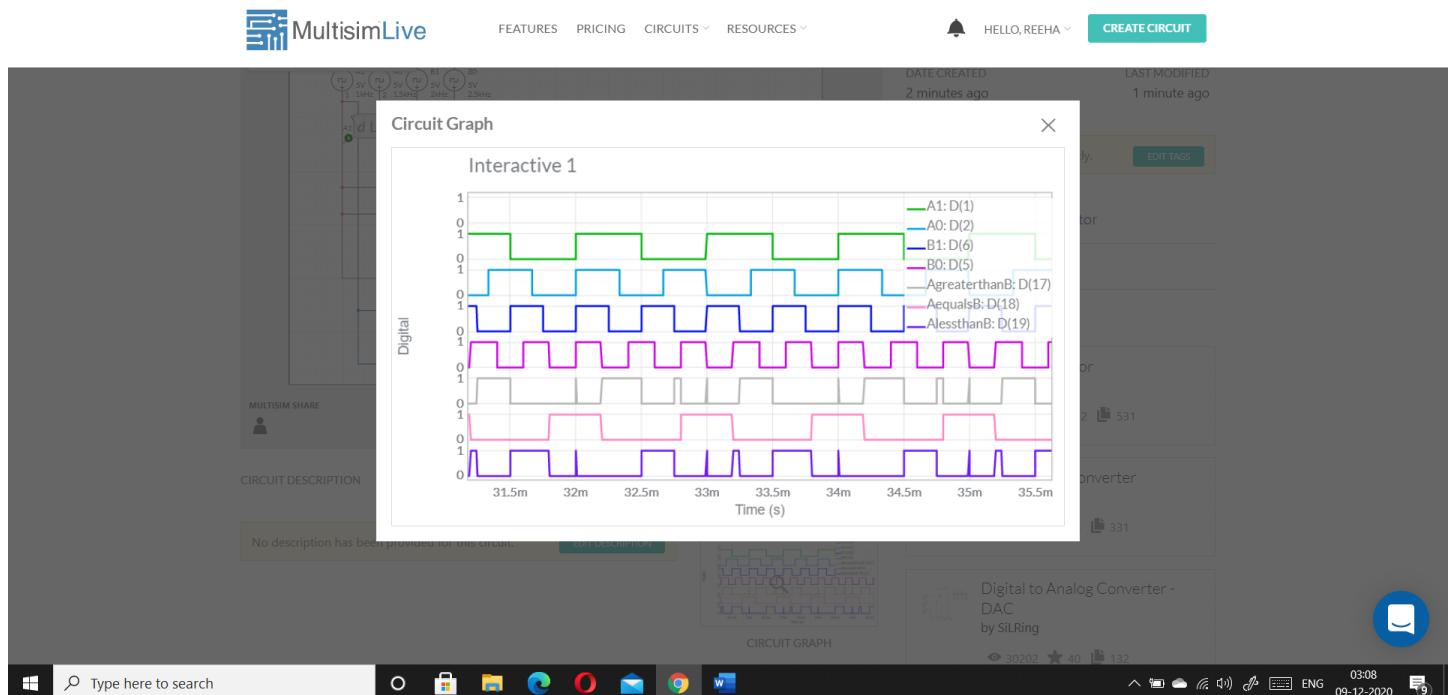
# Circuits and Output waveform for 1 bit magnitude comparator.



# Circuits and Output waveform for 2 - bit magnitude comparator:

The screenshot shows a 2-bit magnitude comparator circuit on MultisimLive. The circuit uses two 4-bit magnitude comparators (U5 and U8) to compare two 2-bit binary numbers (A and B). The outputs of these comparators are then used by a logic network (U13, U14, U15) to produce six output lines: d Lo, d Hi, A greater than B, A equals B, and A less than B. The circuit is powered by 5V and 2.5V supplies. The creator of the circuit is 'reeha' with 34 circuits.

This screenshot is identical to the one above, showing the same 2-bit magnitude comparator circuit on MultisimLive. The circuit diagram, component labels, and creator information ('reeha') are all the same.



## VIVA-VOCE QUESTIONS:

**Q1.** Write down expressions for 4-bit magnitude comparator.

Ans.

- If  $A_3 = 1$  and  $B_3 = 0$ , then A is greater than B ( $A > B$ ). Or
- If  $A_3$  and  $B_3$  are equal, and if  $A_2 = 1$  and  $B_2 = 0$ , then  $A > B$ . Or
- If  $A_3$  and  $B_3$  are equal &  $A_2$  and  $B_2$  are equal, and if  $A_1 = 1$ , and  $B_1 = 0$ , then  $A > B$ . Or
- If  $A_3$  and  $B_3$  are equal,  $A_2$  and  $B_2$  are equal and  $A_1$  and  $B_1$  are equal, and if  $A_0 = 1$  and  $B_0 = 0$ , then  $A > B$ .

From the above statements, the output  $A > B$  logic expression can be written as

$$G = A_3 \overline{B_3} + (A_3 \text{ Ex-NOR } B_3) A_2 \overline{B_2} + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) A_1 \overline{B_1} + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) (A_1 \text{ Ex-NOR } B_1) A_0 \overline{B_0}$$

Similarly the logic expression for the L or  $A < B$  output can be expressed as

$$L = \overline{A_3} B_3 + (A_3 \text{ Ex-NOR } B_3) \overline{A_2} B_2 + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) \overline{A_1} B_1 + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) (A_1 \text{ Ex-NOR } B_1) \overline{A_0} B_0$$

The equal output is produced when all the individual bits of one number are exactly coincides with corresponding bits of another number. Then the logical expression for  $A=B$  output can be written as

$$E = (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) (A_1 \text{ Ex-NOR } B_1) (A_0 \text{ Ex-NOR } B_0)$$

**Q2.** What are applications of magnitude comparator?

Ans.

Digital comparator and magnitude comparator is used in different applications where data comparison is mostly required in many of the activities, and these hold many benefits too.

- Now, look into few of the applications of comparators
- Used for authorization purposes (such as password management) and biometric applications.
- These are implemented in process controllers and also in servo motor controls.
- Implemented for the data comparison of variables like temperature, the pressure is compared with that of reference values.
- Used to address decoding circuitry in computers.

Thus, this is all about digital comparator and magnitude comparator. So, the augmented performance of comparators allowed these devices to gain more prominence in the electronics industry and let them be implemented in many applications.

### Q3. What are Magnitude comparators?

Ans.

A magnitude digital Comparator is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and other for B and have three output terminals, one for  $A > B$  condition, one for  $A = B$  condition and one for  $A < B$  condition.



# EXPERIMENT - 9

Switching Theory and Logic Design (STLD)

## Aim

To realize 4-Bit Binary to BCD Convertor.

Syeda Reeha Quasar

14114802719

3C7

# EXPERIMENT - 9

## AIM:

To realize 4-Bit Binary to BCD Convertor.

## Hardware and Software Apparatus Required

### Hardware:

Breadboard, IC 7400 (NAND), IC 7410 (3 input NAND), IC 7404(NOT), LEDs, 5V power supply, connecting wires.

### Software Simulation:

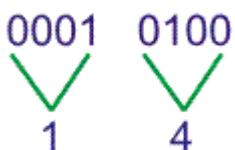
The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

## Theory:

BCD is binary coded decimal number, where each digit of a decimal number is respected by its equivalent binary number. That means, LSB of a decimal number is represented by its equivalent binary number and similarly other higher significant bits of decimal number are also represented by their equivalent binary numbers.

For example, BCD Code of 14 is-



Let us design a 4 bit **binary to BCD code converter**. As the 4 bit can

represent 0 to 15, we can draw the conversion table as follows,

Binary Code A B C D	Decimal Number	BCD Code B <sub>5</sub>   B <sub>4</sub> B <sub>3</sub> B <sub>2</sub> B <sub>1</sub>
0 0 0 0	0	0 0 0 0 0
0 0 0 1	1	0 0 0 0 1
0 0 1 0	2	0 0 0 1 0
0 0 1 1	3	0 0 0 1 1
0 1 0 0	4	0 0 1 0 0
0 1 0 1	5	0 0 1 0 1
0 1 1 0	6	0 0 1 1 0
0 1 1 1	7	0 0 1 1 1
1 0 0 0	8	0 1 0 0 0
1 0 0 1	9	0 1 0 0 1
1 0 1 0	10	1 0 0 0 0
1 0 1 1	11	1 0 0 0 1
1 1 0 0	12	1 0 0 1 0
1 1 0 1	13	1 0 0 1 1
1 1 1 0	14	1 0 1 0 0
1 1 1 1	15	1 0 1 0 1

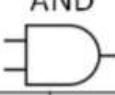
Here, B<sub>5</sub> bit represents MSB of decimal number and B<sub>4</sub>, B<sub>3</sub>, B<sub>2</sub>, B<sub>1</sub> represents 4 bit binary equivalent of LSB of decimal number.

From, above conversion table, we can write SOP form for different bits of BCD code.

$$B_5 = \sum m(10, 11, 12, 13, 14, 15), \quad B_4 = \sum m(8, 9), \quad B_3 = \sum m(4, 5, 6, 7, 14, 15), \\ B_2 = \sum m(2, 3, 6, 7, 12, 13), \quad B_1 = \sum m(1, 3, 5, 7, 9, 11, 13, 15)$$

### TRUTH TABLE :

AND



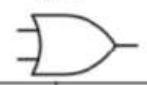
INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

NOT



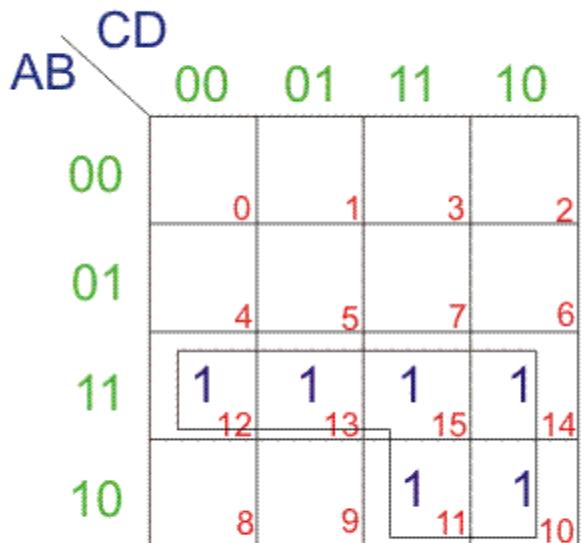
INPUT	OUTPUT
A	
0	1
1	0

OR



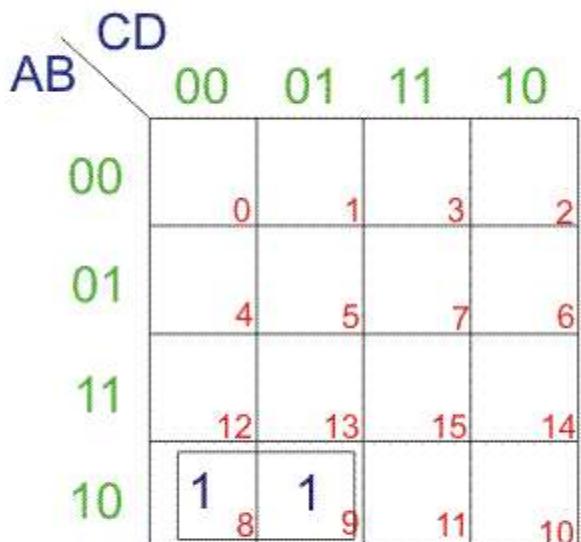
INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

K - map for  $B_5$



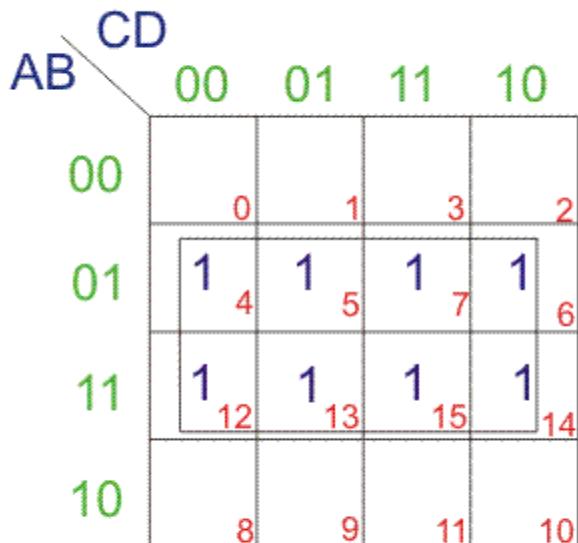
$$B_5 = AB + BC$$

K - map for  $B_4$



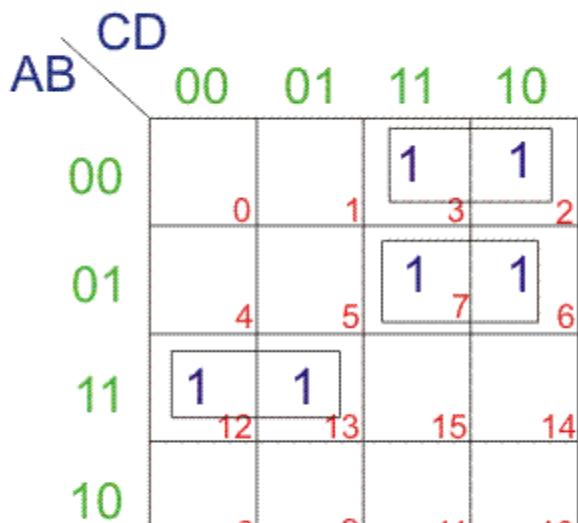
$$B_4 = \bar{A}\bar{B}\bar{C}$$

K - map for  $B_3$



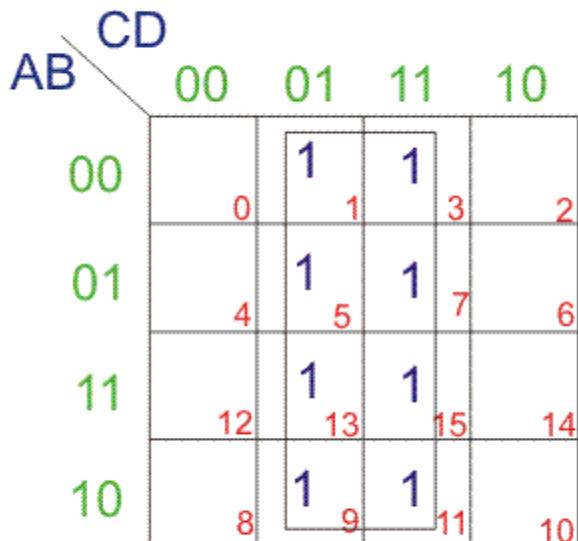
$$B_3 = \overline{A}B + CB$$

K - map for  $B_2$



$$B_2 = ABC + \overline{A}C$$

## K - map for $B_1$



$$B_1 = D$$



### **Procedure:**

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the results and observe the outputs.

### Multisim:

1. Click on the 'Create Circuit' option on the top right corner of the profile in NI
2. Multisim webpage.

3. The schematic representation opens in a new tab.
4. Place three 'Ground' Schematic connector on the screen.
5. Place the logic gates from the digital section on the board, as per the required circuit diagram for 4-bit Binary to BCD Converter
6. Now, add clock voltages to the input of the logic gate and connect them with the help of Ground present in 'Schematic Connectors'.
7. Change the frequency of clock voltages e.g V1(say=5kHz) and V2(say=3kHz)
8. etc.
9. Connect a resistor to the output of the logic gate and then, Ground it with the help of Ground Schematic Connector.
10. Connect the components with connecting wires.
11. Add digital probes to both input and output connections.
12. Set the display to 'Transient' from Interactive and press the 'Start Simulation' button.
13. Note the graph

### **PRECAUTIONS:**

1. All ICs should be checked before starting the experiment.
2. All the connection should be tight.
3. Always connect ground first and then the supply.
4. Switch off the power supply after completion of the experiment.

### **RESULT:**

4-Bit Binary to BCD Convertor has been studied and its truth table is verified.

# Circuits and Output waveform

The screenshot shows a MultisimLive project titled "4-Bit Binary to BCD Converter". The circuit diagram consists of several logic gates (NOT, AND, OR) connected to convert four binary inputs (d1, d2, d3, d4) into two BCD outputs (bcd1, bcd2). The creator of the circuit is "reeha" with 29 circuits. The date created was 10 minutes ago, and it was last modified 0 minutes ago. There are no tags assigned to this circuit.

The screenshot shows the MultisimLive interface with the same 4-bit binary to BCD converter project. A waveform graph titled "Interactive 1" is displayed, showing the digital signals over time. The graph displays ten waveforms labeled PR1 through PR10, each representing a different bit or part of the BCD output. The x-axis represents time from 5m to 8.5m. The y-axis shows the signal levels (0 or 1). The creator information and most popular circuits section are also visible on the right side of the screen.

## VIVA-VOCE QUESTIONS:

Q1.  $(10110)_2 = (\quad ? \quad )_{BCD}$

Ans.

$$(10110)_2 = (\underline{\hspace{2cm}})_{BCD}$$

Q2. Convert decimal to BCD  $(22)_{10} = (\underline{\hspace{2cm}})_{BCD}$

2	2	$\therefore (22)_{10} = (\underline{00100010})_{BCD}$	
<u>0010</u>	<u>0010</u>	$\therefore (10110)_2 = (\underline{00100010})_{BCD}$	
$\therefore (10110)_2 = (\underline{00100010})_{BCD}$			

Q3. Write the algorithm to convert binary to BCD.

Ans.

**Algorithm:**

1. If any column (100's, 10's, 1's, etc.) is 5 or greater, add 3 to that column.
2. Shift all #'s to the left 1 position.
3. If 8 shifts have been performed, it's done! Evaluate each column for the BCD values.
4. Go to step 1.

100's	10's	1's	Binary	Operation
			1010 0010	
		1	010 0010	<< #1
		10	10 0010	<< #2
		101	0 0010	<< #3
		1000		add 3
	1	0000	0010	<< #4
	10	0000	010	<< #5
	100	0000	10	<< #6
	1000	0001	0	<< #7
	1011			add 3
1	0110	0010		<< #8

162

#### **Q4. Why do we convert Binary to BCD?**

Ans.

Conversion of a binary number into separate binary numbers representing digits of the decimal number.

(this example is for 8-bits, other sizes follow the same pattern)

```
for(i=0; i<8; i++) {
    //check all columns for >= 5
    for each column {
        if (column >= 5)
            column += 3;

    //shift all binary digits left 1
    Hundreds <<= 1;
    Hundreds[0] = Tens[3];
    Tens <<= 1;
    Tens[0] = Ones[3];
    Ones <<= 1;
    Ones[0] = Binary[7];
    Binary <<= 1;
}
```

# EXPERIMENT - 10

Switching Theory and Logic Design (STLD)

## Aim

To realize Multiplexer and Demultiplexer using only NAND gates.

Syeda Reeha Quasar

14114802719

3C7

# **EXPERIMENT - 10**

## **AIM:**

To realize Multiplexer and Demultiplexer using only NAND gates.

## **Hardware and Software Apparatus Required**

### **Hardware:**

Breadboard, IC 7400 (NAND), LEDs, 5V power supply, connecting wires.

### **Software Simulation:**

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

## **Theory:**

Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output. By using control signals (select lines) we can select any input to the output. Multiplexer is also called as data selector because the output bit depends on the input data bit that is selected. The general multiplexer circuit has  $2^n$  input signals,  $n$  control/select signals and 1 output signal.

De-multiplexers perform the opposite function of multiplexers. They transfer a small number of information units (usually one unit) over a larger number of channels under the control of selection signals. The general de-multiplexer circuit has 1 input signal,  $n$  control/select signals and  $2^n$  output signals. De-multiplexer circuit can also be realized using a decoder circuit with enable.

## REALIZATION OF 2:1 MUX

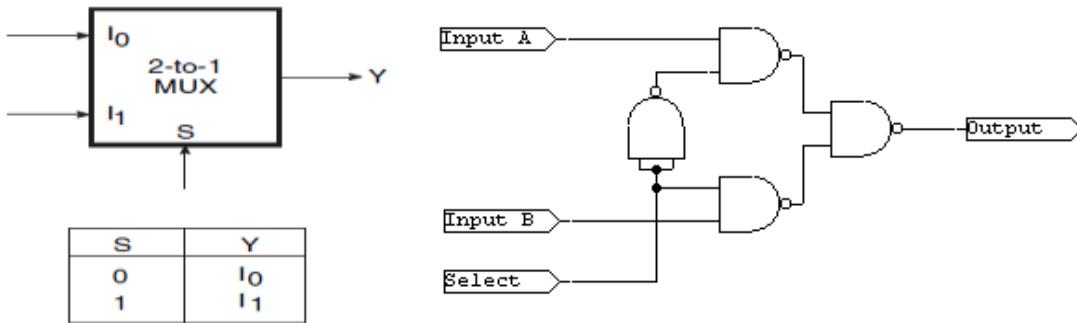


Fig 6.1 Truth Table and circuit implementation of 2:1 mux

## REALIZATION OF 1:2 DEMUX

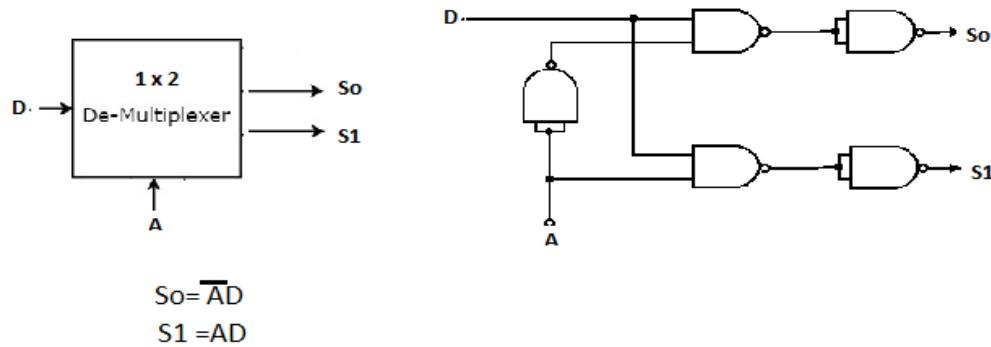


Fig 6.2 Logic Diagram and Circuit of 1:2 DMUX

### Procedure:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the results and observe the outputs.

### Multisim:

1. Click on the 'Create Circuit' option on the top right corner of the profile in NI Multisim webpage.
3. The schematic representation opens in a new tab.
4. Place three 'Ground' Schematic connector on the screen.
5. Place the logic gates from the digital section on the board, as per the required circuit diagram for MUX and DEMUX circuits.
6. Now, add clock voltages to the input of the logic gate and connect them with the help of Ground present in 'Schematic Connectors'.

7. Change the frequency of clock voltages e.g V1(say=5kHz) and V2(say=3kHz) etc.
8. Connect a resistor to the output of the logic gate and then, Ground it with the help of Ground Schematic Connector.
9. Connect the components with connecting wires.
10. Add digital probes to both input and output connections.
11. Set the display to 'Transient' from Interactive and press the 'Start Simulation' button.
12. Note the graph

## PRECAUTIONS:

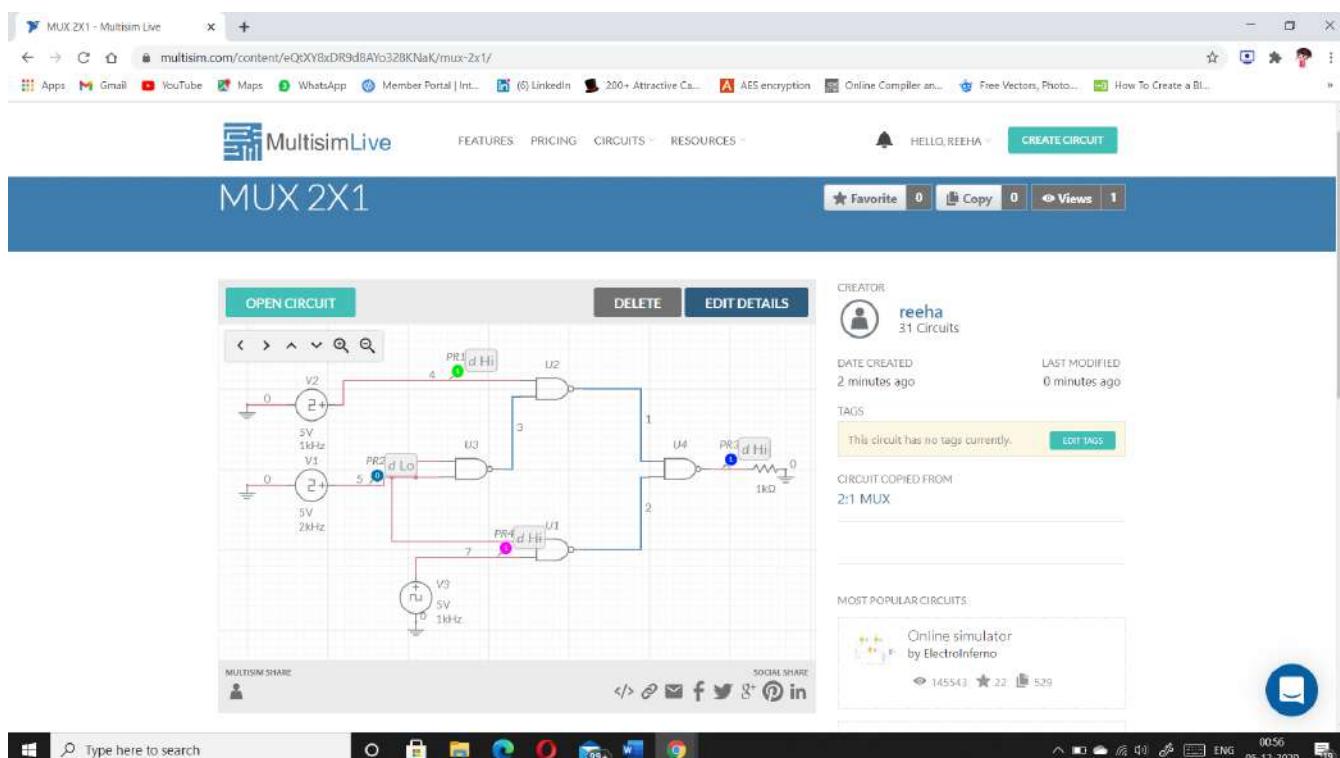
1. All ICs should be checked before starting the experiment.
2. All the connection should be tight.
3. Always connect ground first and then the supply.
4. Switch off the power supply after completion of the experiment.

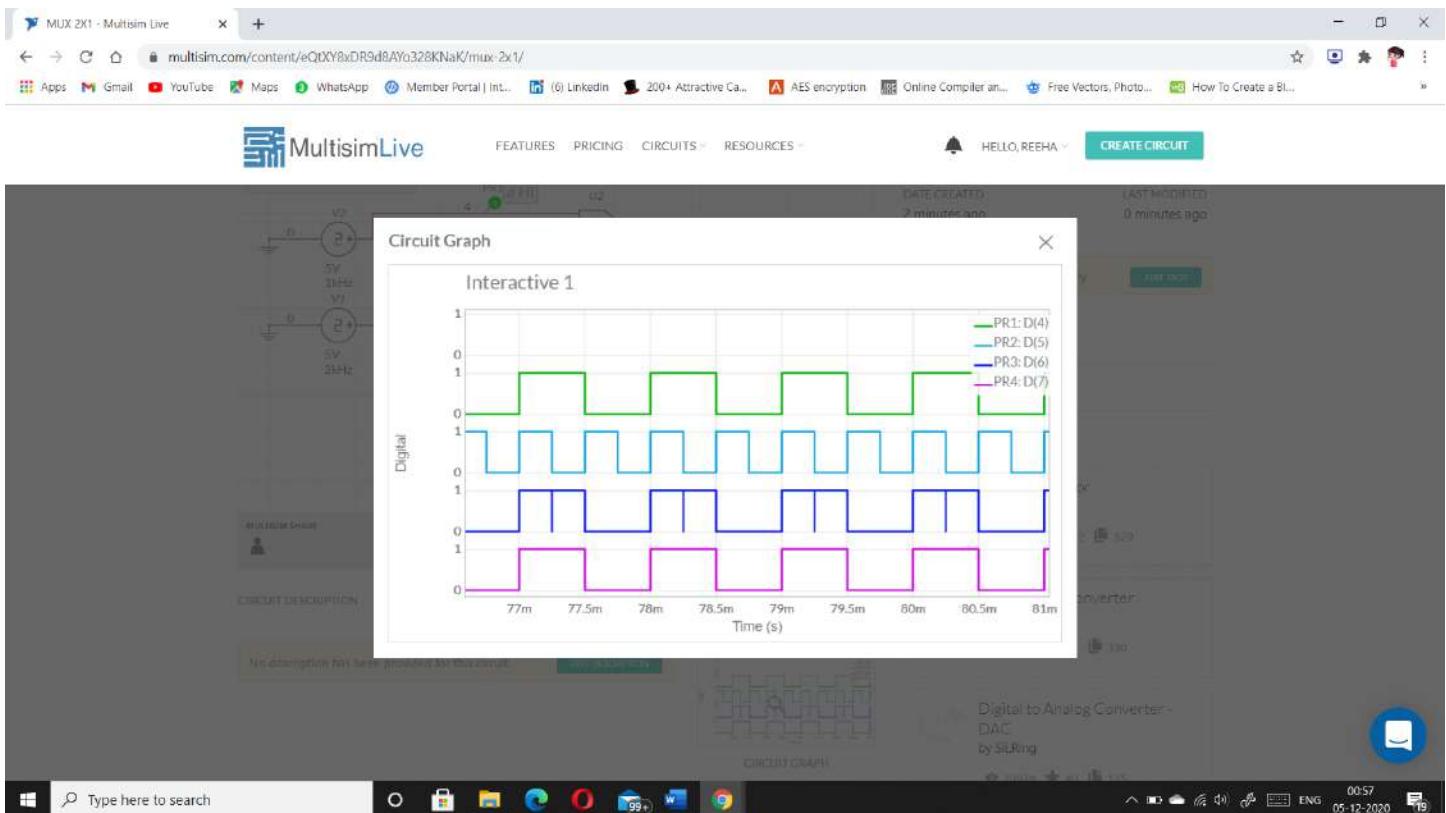
## RESULT:

Multiplexer and Demultiplexer have been studied and their truth table has been verified.

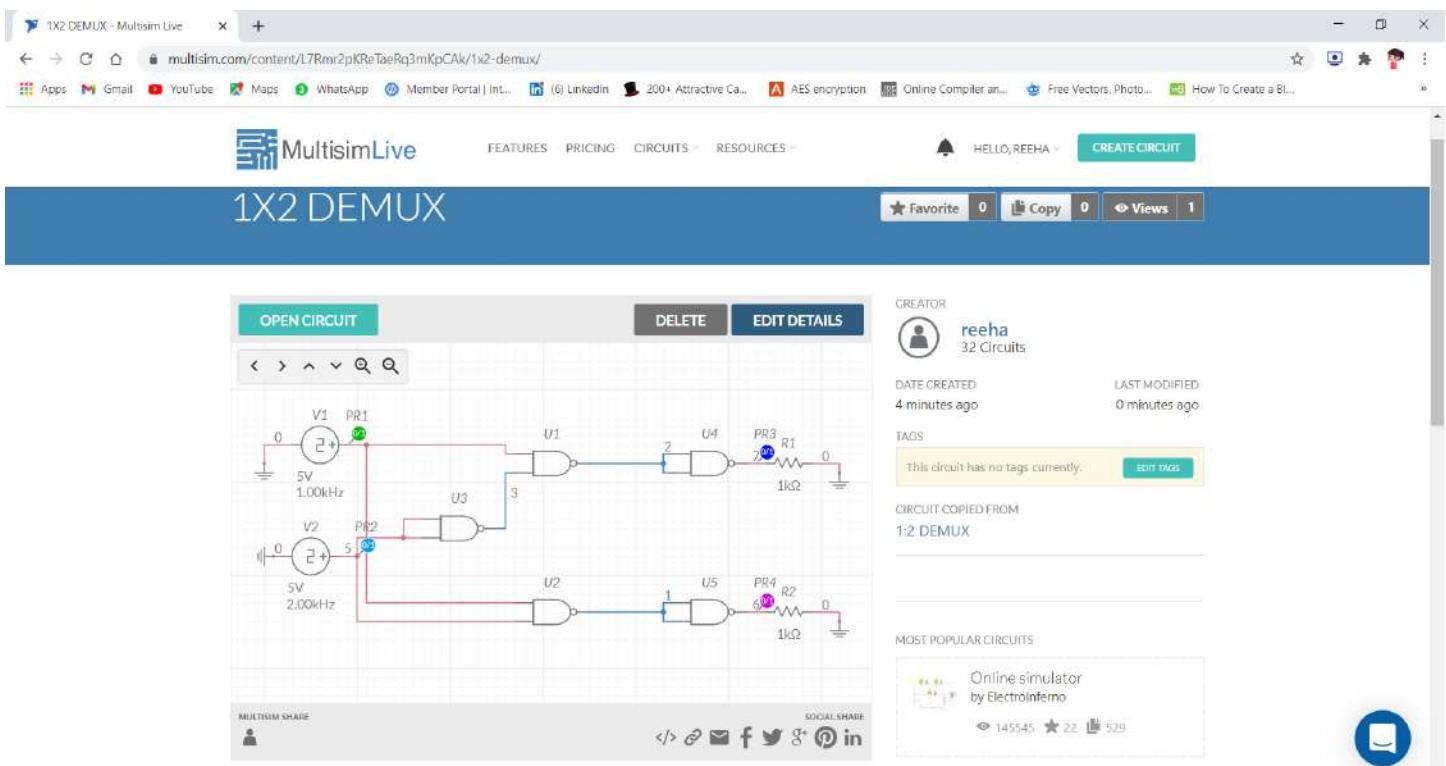
## Circuits and Output waveform

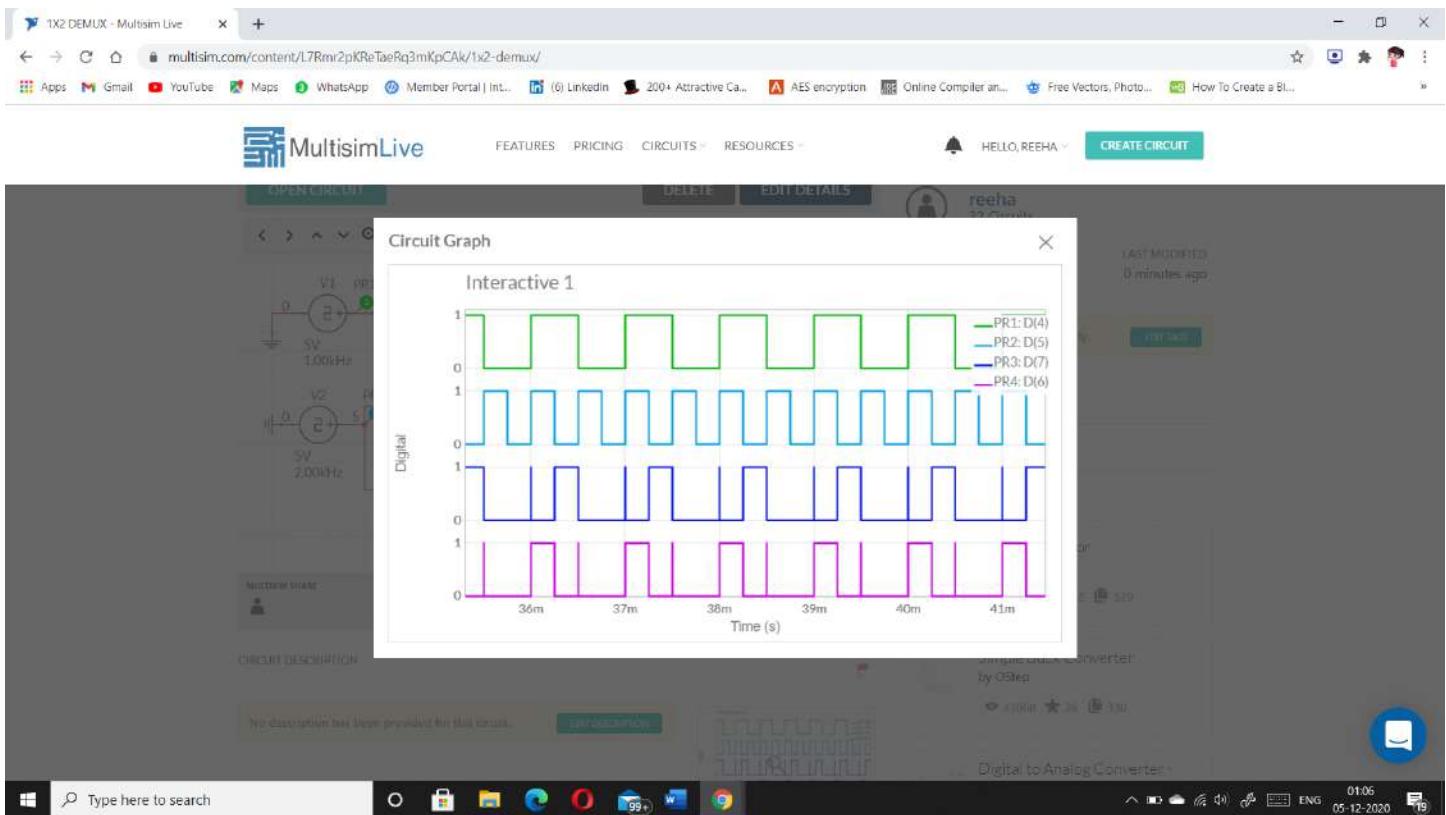
### 2 X 1 MUX





## 1 X 2 DEMUX





## VIVA-VOCE QUESTIONS:

### Q1. What is multiplexer?

Ans.

Multiplexer means many into one. A multiplexer is a circuit used to select and route any one of the several input signals to a signal output. An simple example of an non electronic circuit of a multiplexer is a single pole multiposition switch.

### Q2. What is difference between decoder and demultiplexer?

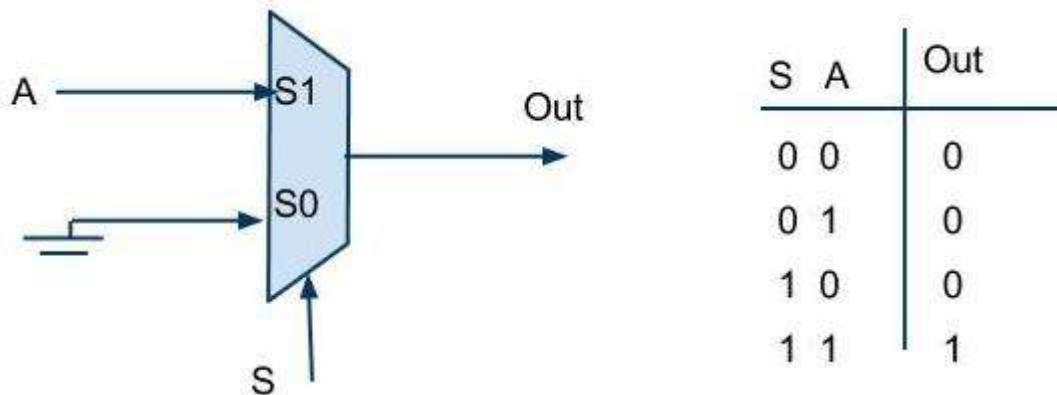
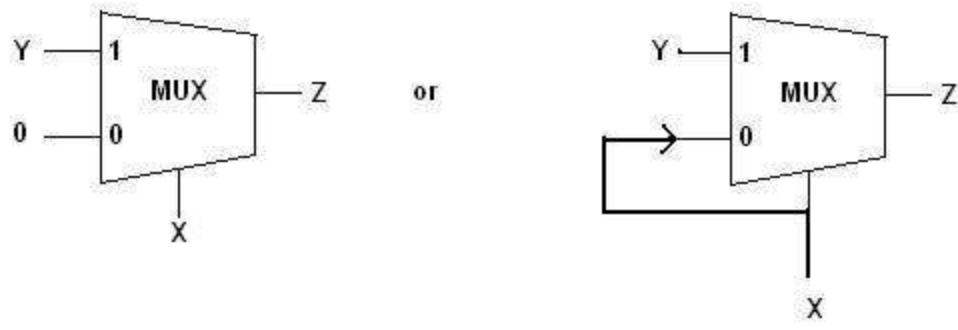
Ans.

S.NO.	COMPARISON	DECODER	DEMULTIPLEXER
1.	Basic	These are Logic circuit which decodes an encrypted input stream from one to another format.	It is a Combination circuit which routes a single input signal to one of several output signals.
2.	Input/Output	$n$ number of input lines and $2n$ number of output lines.	$n$ number of select lines and $2n$ number of output lines.
3.	Inverse of	Encoder.	Multiplexer.

4.	Application	In Detection of bits, data encoding.	In Distribution of the data, switching.
5.	Use	It is used for changing the format of the instruction in the machine specific language.	It is used as a routing device to route the data coming from one signal into multiple signals.
6.	Select Lines	Not contains.	Contains.
7.	Implementation	Majorly implemented in the networking application.	Employed in data-intensive applications where data need to be changed into another form.

### Q3. Implement a AND gate with multiplexers.

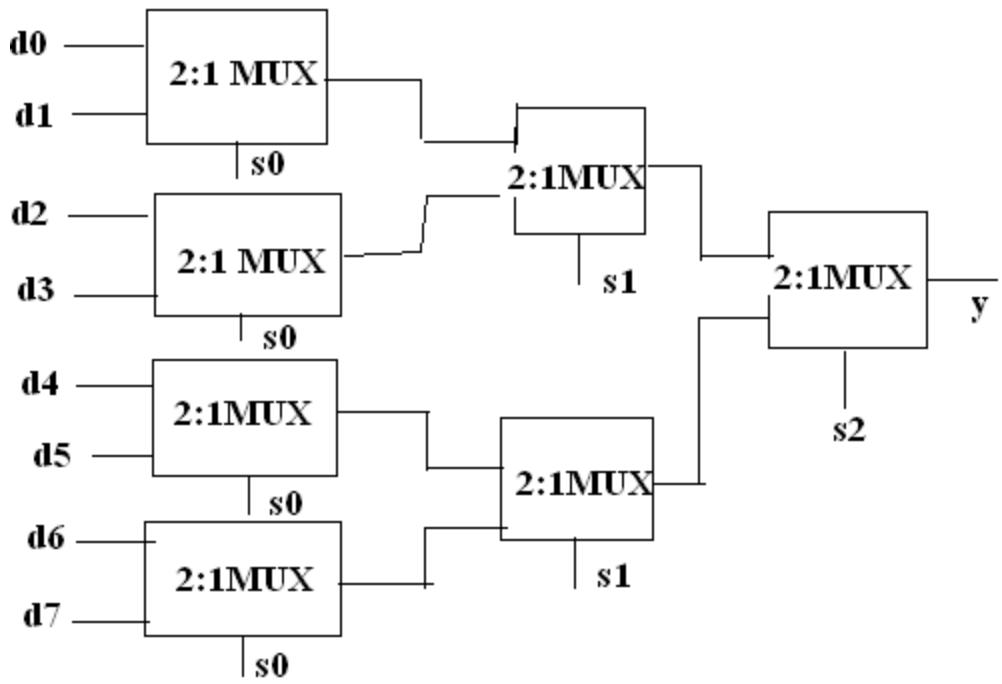
Ans.



### Q4. How many 2x1 mux required to make 8x1 mux?

Ans.

Seven 2X1 MUX are required.



# EXPERIMENT - 11

Switching Theory and Logic Design (STLD)

## Aim

To realize J-K flip flop using logic gates or by using kit.

Syeda Reeha Quasar

14114802719

3C7

# EXPERIMENT - 11

## AIM:

To realize J-K flip flop using logic gates or by using kit.

## Hardware and Software Apparatus Required

### Hardware:

Breadboard, IC 7400 (NAND), IC 7410 (3 input NAND), LEDs, resistor(1k), capacitor(.1uF), diode, 5V power supply, connecting wires or Omega LTB-826, patch cords.

### Software Simulation:

The schematic models of the desired circuits will be stimulated on MULTISIM (Free Software), easily accessible at [www.multisim.com](http://www.multisim.com).

**Components used –** Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

## Theory:

The **JK Flip Flop** is the most widely used flip flop. The sequential operation of the JK Flip Flop is same as for the RS flip-flop with the same **SET** and **RESET** input. The difference is that the JK Flip Flop does not have the invalid input states of the RS Latch (when S and R are both 1). The JK Flip Flop name has been kept on the inventor name of the circuit known as **Jack Kilby**. S and R inputs of the RS bistable have been replaced by the two inputs called the J and K input respectively. Here  $J = S$  and  $K = R$ . The two input AND gates of the RS flip-flop is replaced by the two 3 inputs NAND gates as shown in figure 7.1(a) ,with the third input of each gate connected to the outputs at Q and  $\bar{Q}$ . This cross coupling of the RS Flip-Flop is used to produce toggle action. Clock will be provided by R-C circuit followed by diode as shown in figure 7.1(b).

### Logic Diagram of J-K Flip Flop-

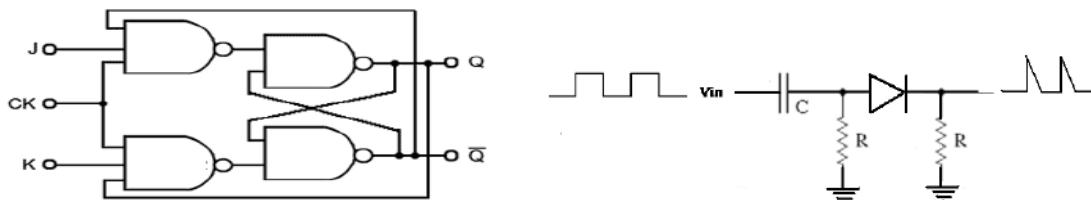


Fig 7.1 a) J-K Flip Flop b) clock provider circuit

Working of flip flop is depicted in truth table given below

### Truth Table For J-K Flip Flop:

J	K	CLK	Q	$\bar{Q}$	Comment
0	0	↑	Q	$\bar{Q}$	Latch
1	0	↑	1	0	SET
0	1	↑	0	1	RESET
1	1	↑	$\bar{Q}$	Q	TOGGLE
X	X	0	Q	$\bar{Q}$	NO Change !

### Pin Configuration of IC 7410

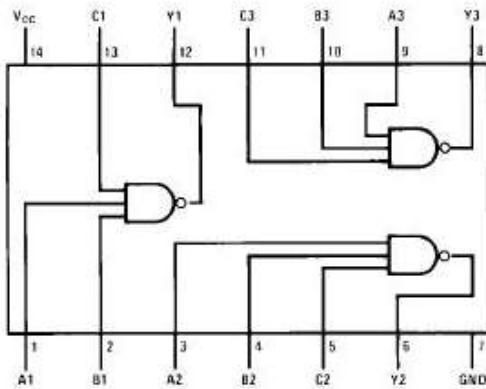


Fig 7.2 Pin configuration of IC used

### **Procedure:**

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the results and observe the outputs.
5. While working on kit
6. Made connections as shown in circuit diagram on kit using patch cords.

### Multisim:

1. Click on the 'Create Circuit' option on the top right corner of the profile in NI
2. Multisim webpage.
3. The schematic representation opens in a new tab.
4. Place three 'Ground' Schematic connector on the screen.
5. Place the logic gates from the digital section on the board, as per the required circuit diagram for circuits given.
6. Now, add clock voltages to the input of the logic gate and connect them with the help of Ground present in 'Schematic Connectors'.
7. Change the frequency of clock voltages e.g V1(say=5kHz) and V2(say=3kHz) etc.
8. Connect a resistor to the output of the logic gate and then, Ground it with the help of Ground Schematic Connector.
9. Connect the components with connecting wires.
10. Add digital probes to both input and output connections.
11. Set the display to 'Transient' from Interactive and press the 'Start Simulation' button.
13. Note the graph

### PRECAUTIONS:

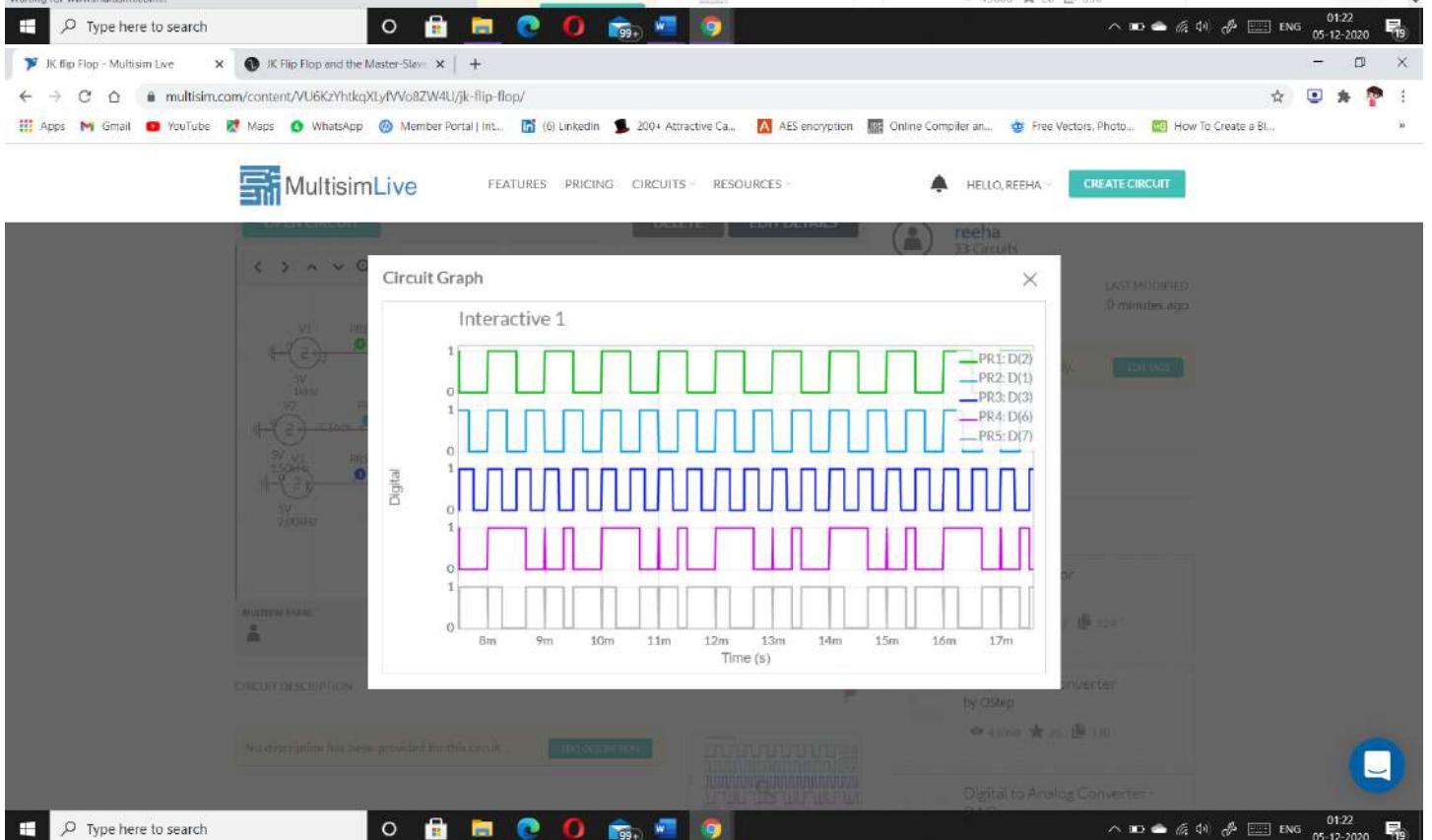
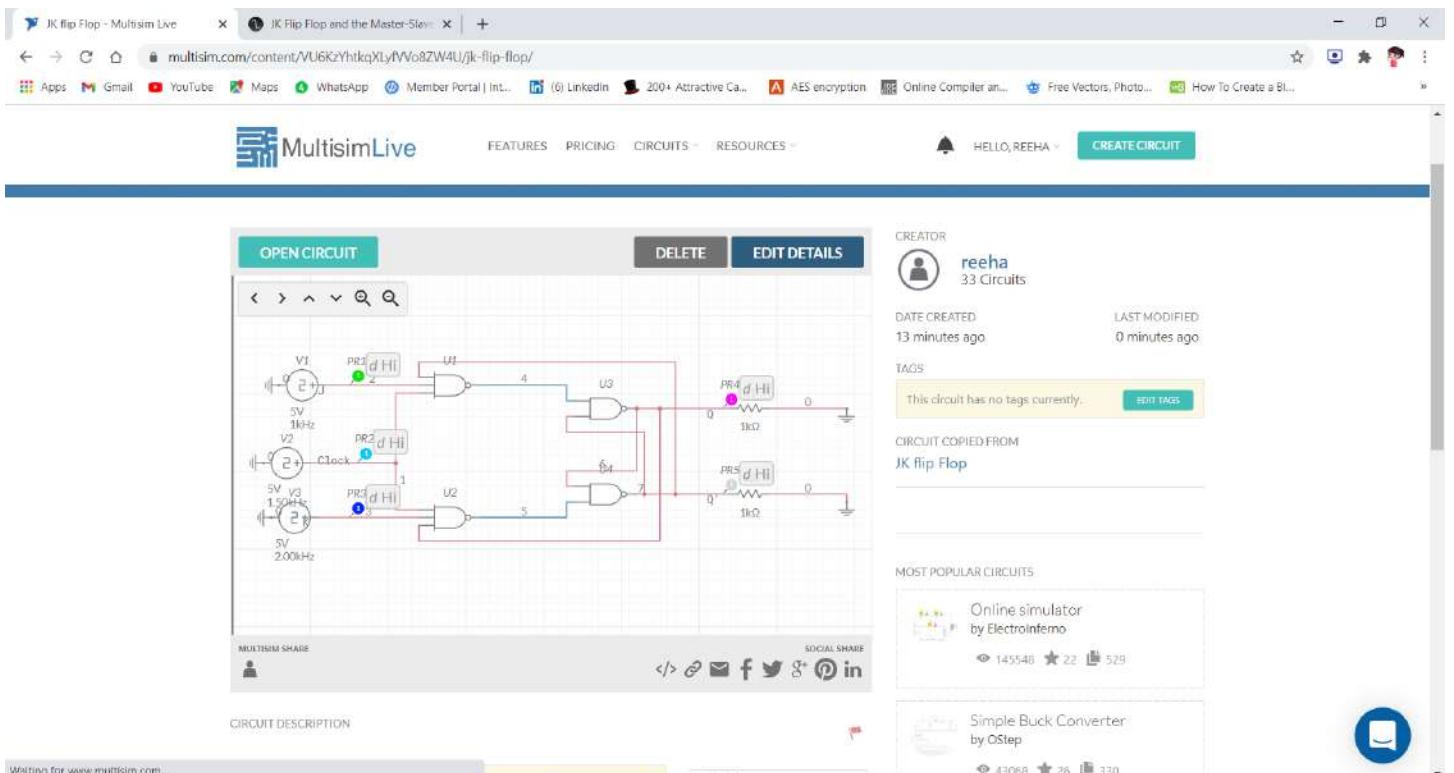
1. All ICs should be checked before starting the experiment.
2. All the connection should be tight.
3. Always connect ground first and then the supply.
4. Switch off the power supply after completion of the experiment.

### RESULT:

The J-K flip flop has been studied and its truth table has been verified.

## Circuits and Output waveform

### JK Flip Flop



## VIVA-VOCE QUESTIONS:

### Q1. Difference between sequential and combinational circuits.

Ans.

**Combinational circuits** are defined as the time independent circuits which do not depend upon previous inputs to generate any output are termed as combinational circuits. **Sequential circuits** are those which are dependent on clock cycles and depends on present as well as past inputs to generate any output.

#### Combinational Circuit –

1. In this output depends only upon present input.
2. Speed is fast.
3. It is designed easy.
4. There is no feedback between input and output.
5. This is time independent.
6. Elementary building blocks: Logic gates
7. Used for arithmetic as well as boolean operations.
8. Combinational circuits don't have capability to store any state.
9. As combinational circuits don't have clock, they don't require triggering.
10. These circuits do not have any memory element.
11. It is easy to use and handle.

**Examples –** Encoder, Decoder, Multiplexer, Demultiplexer

#### Block Diagram –



Figure: Combinational Circuits

#### Sequential Circuit –

1. In this output depends upon present as well as past input.
2. Speed is slow.
3. It is designed tough as compared to combinational circuits.

4. There exists a feedback path between input and output.
5. This is time dependent.
6. Elementary building blocks: Flip-flops
7. Mainly used for storing data.
8. Sequential circuits have capability to store any state or to retain earlier state.
9. As sequential circuits are clock dependent they need triggering.
10. These circuits have memory element.
11. It is not easy to use and handle.

**Examples –** Flip-flops, Counters

**Block Diagram –**

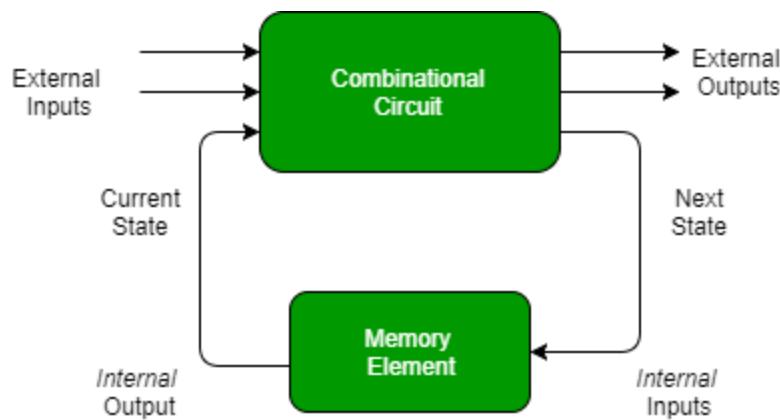


Figure: Sequential Circuit

## Q2. List four Basic Flip-flop applications.

Ans.

Application of the flip flop circuit mainly involves in bounce elimination switch, data storage, data transfer, latch, registers, counters, frequency division, memory, etc.

Some of the most common applications of flip – flops are

- Counters
- Registers
- Frequency Divider circuits
- Data transfer

## Q3. What advantage does a J-K Flip-flop have over an S-R?

Ans.

The J-K flip-flop is much faster. The J-K flip-flop does not have propagation delay problems. The J-K flip-flop has a toggle state.

The RS flip-flop has a “forbidden input state” with S and R both 1. In this state both Q and Not Q outputs will be 1 - which is bad enough - and the state of the flip-flop after R and S return to more sensible inputs depends on the order in which that happens.

The big difference of the JK flip-flop is, that the case  $J = K = 1$  is allowed and toggles the output state. In everything else it works like a RS flip-flop.

So the JK flip-flop is a slightly enhanced version of the RS flip-flop. But better yet are gated flip-flops - they need slightly more gates but you have a lot better control over when the output changes.

#### **Q4. What is meant by Race around condition and how we can overcome this problem?**

Ans.

If the clock is High for a time interval less than the propagation delay of the flip flop then racing around condition can be eliminated. This is done by using the edge-triggered flip flop rather than using the level-triggered flip-flop.

When the input to the JK flip-flop is  $j=1$  and  $k=1$ , the race around condition occurs, i.e it occurs when the time period of the clock pulse is greater than the propagation delay of the flip flop. so the output changes or toggles in a single clock period.