# *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

| Exp. no | Experiment Name | Date of performance | Date of checking | Marks | Signature |
|---|---|---|---|---|---|
| 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)**

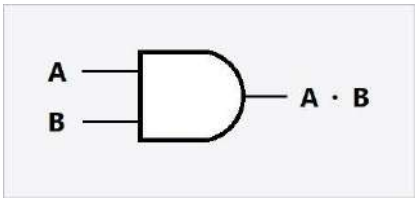
**SOFTWARE REQUIRED:**

**Software stimulator (MULTISIM) – www.multisim.com (free software)**
**Stimulating schematic models of desired circuits**

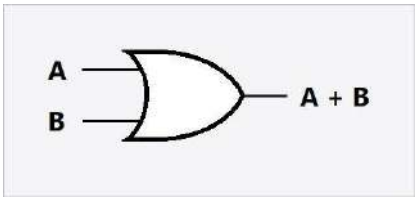
**Components used – Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)**

**CIRCUIT:**

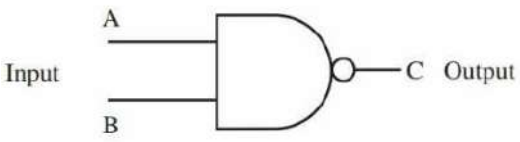
AND GATE



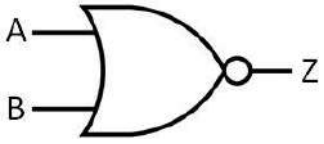

OR GATE



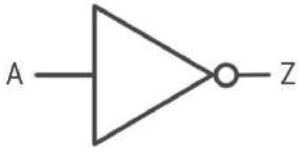

NAND GATE

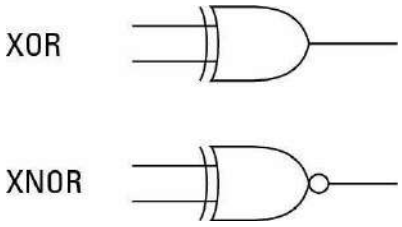



NOR GATE



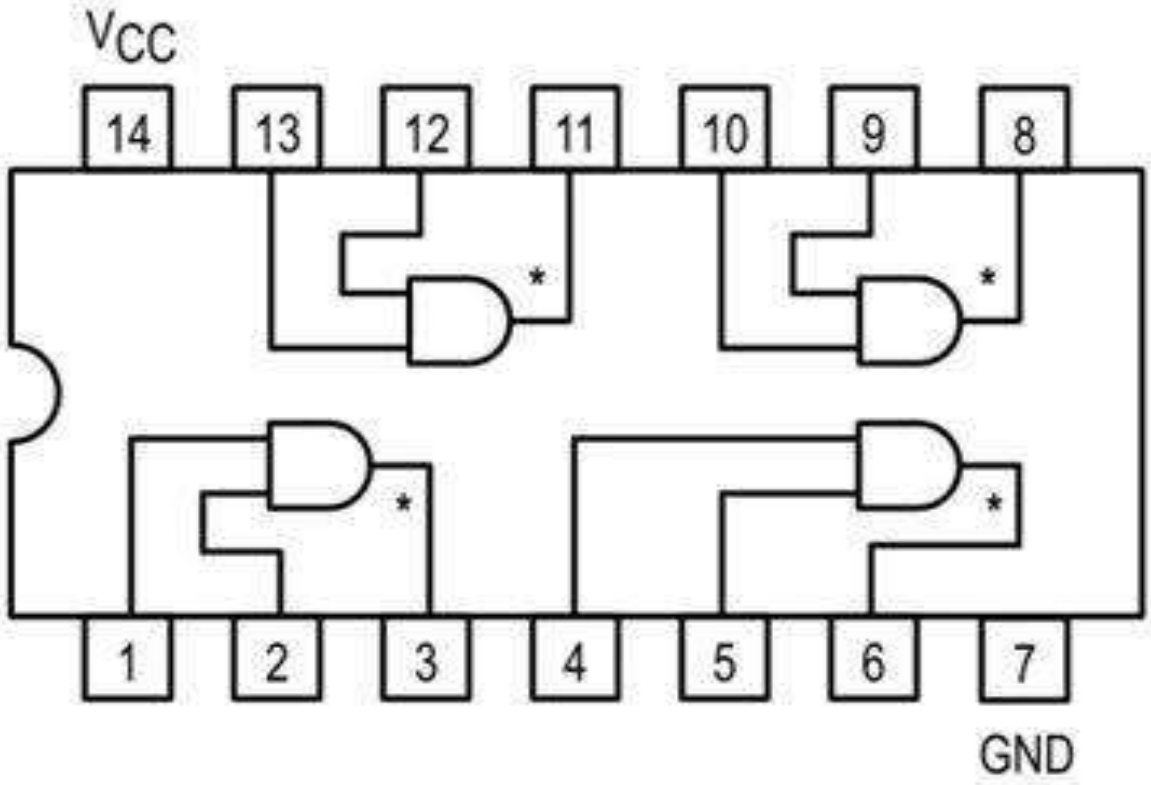

NOT GATE



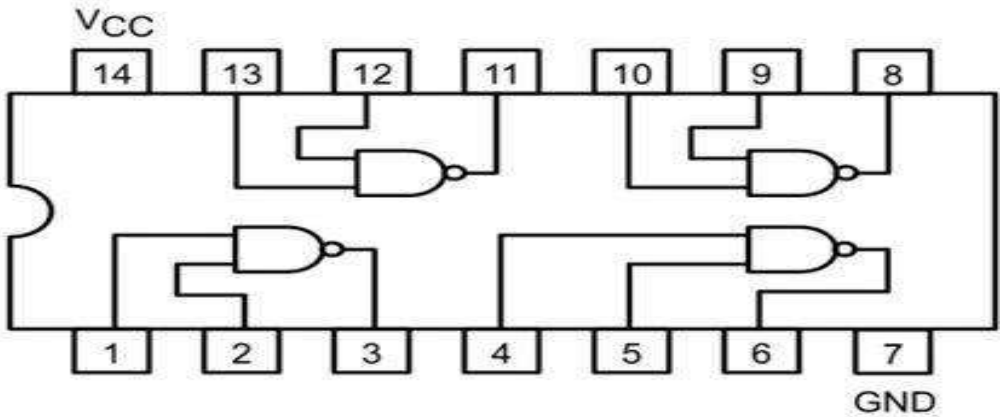



## PIN – DIAGRAM:

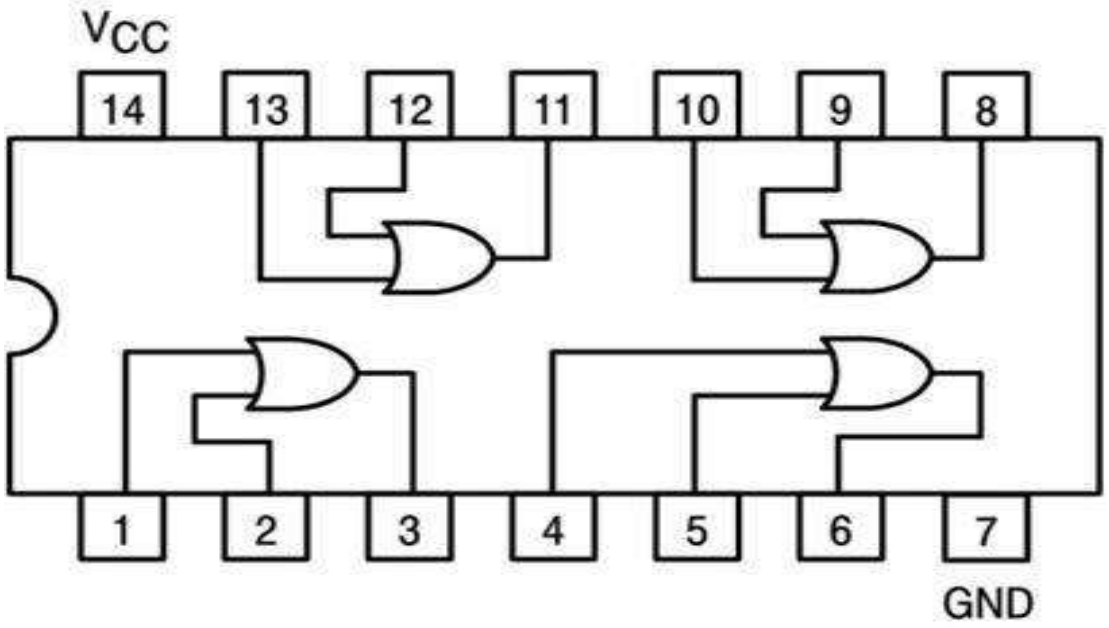
### AND GATE



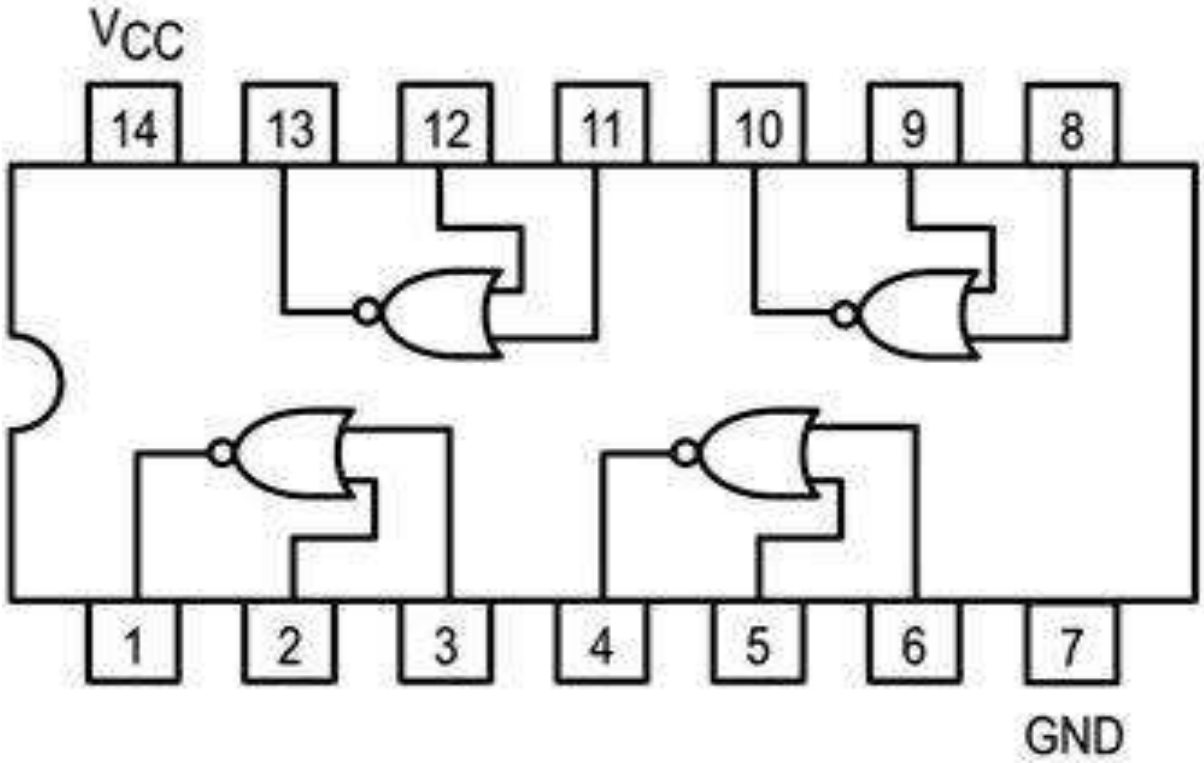

### NAND GATE



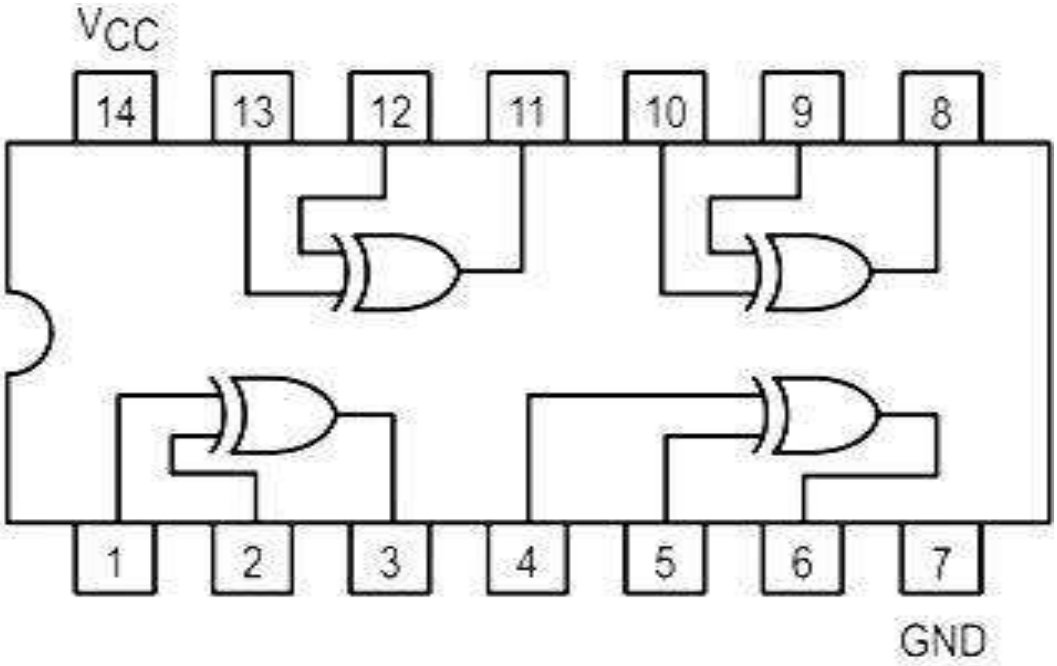
**OR GATE**



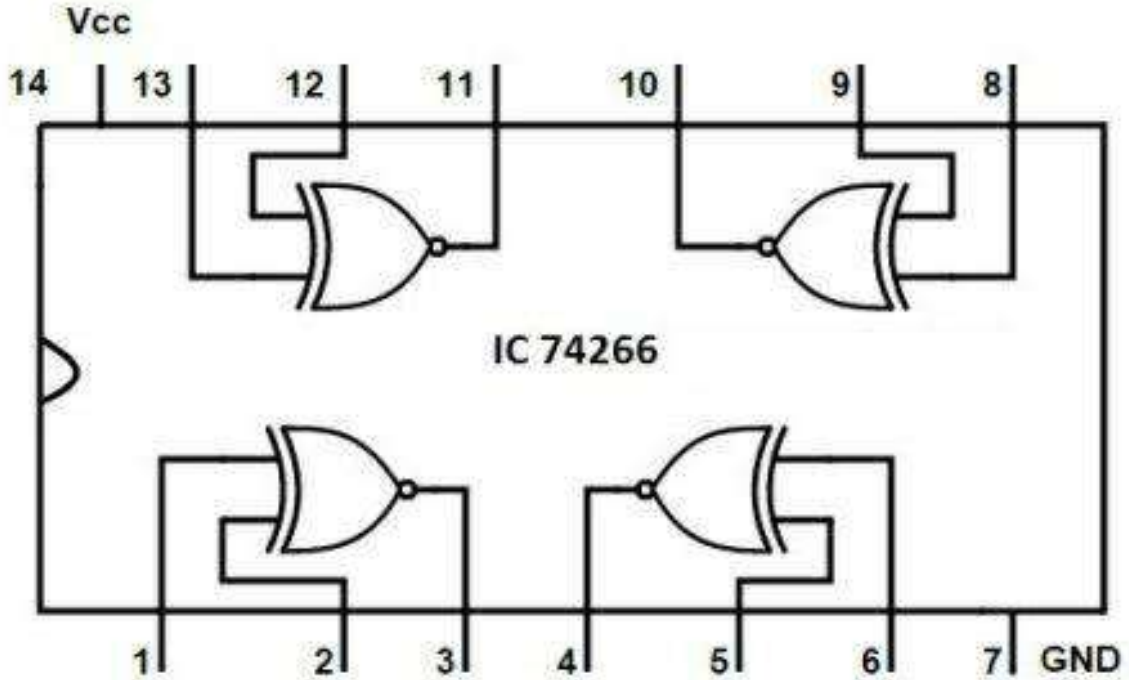

**NOR GATE**



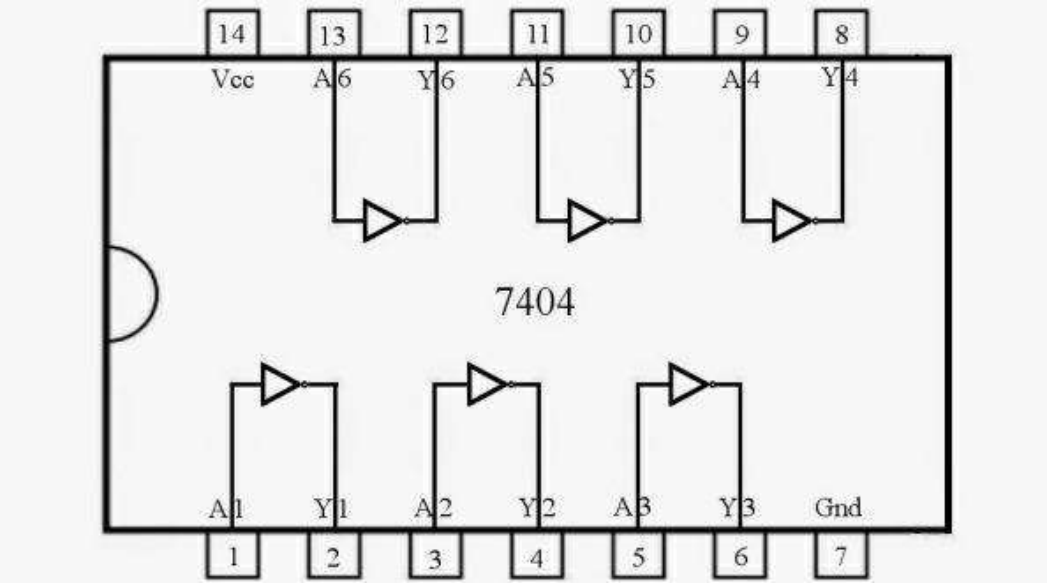
**XOR GATE**




**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.

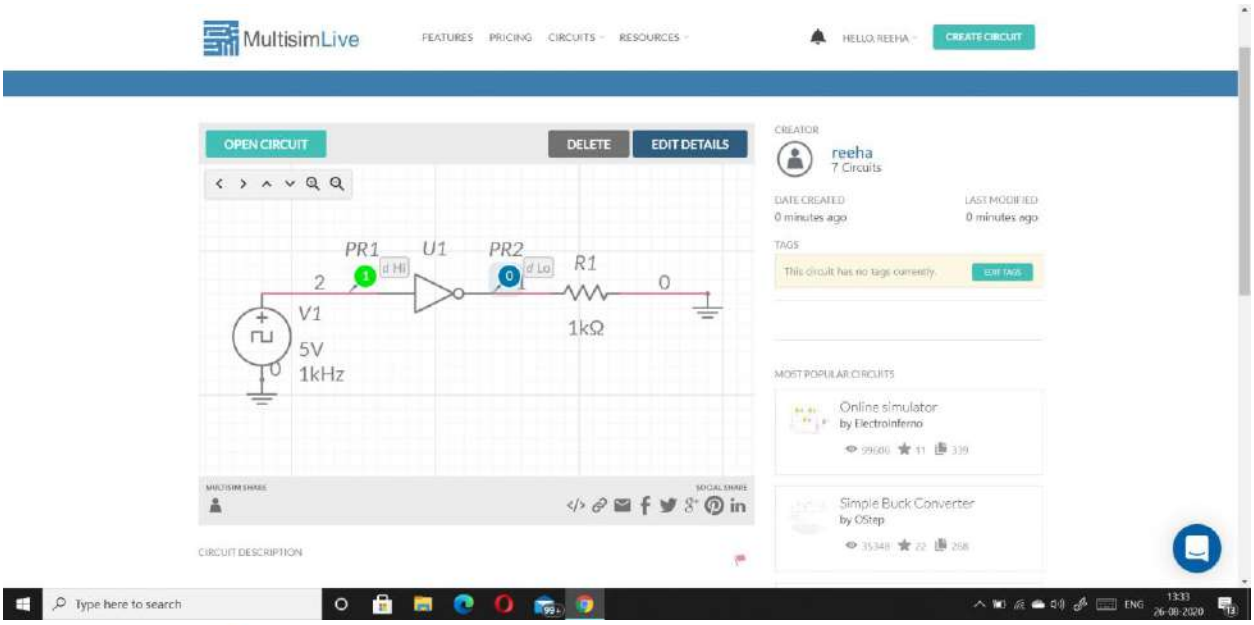
## PROCEDURE (MULTISIM):

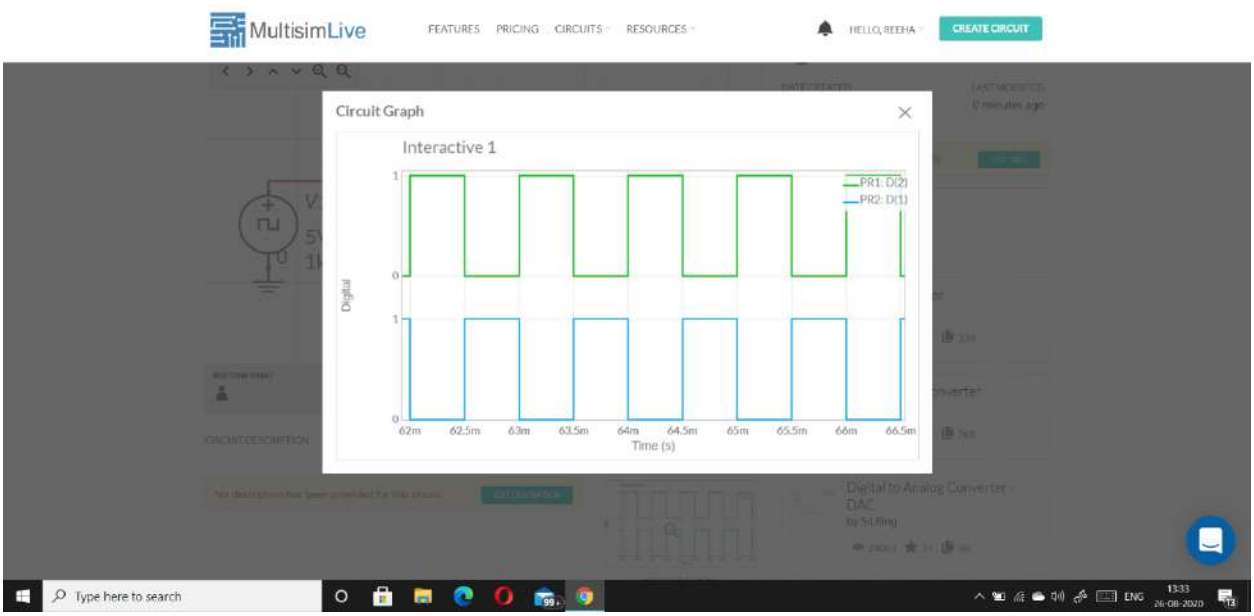
- Make the circuits shown as shown in the figures.
- Select the required components (gates, resistor, voltage sources (Clock Voltage) and ground symbols from the tool bar on the left.
- Ground both the voltage sources (clock Voltages) and then connect them to the input terminal of the gate.
- Connect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

## Precautions:

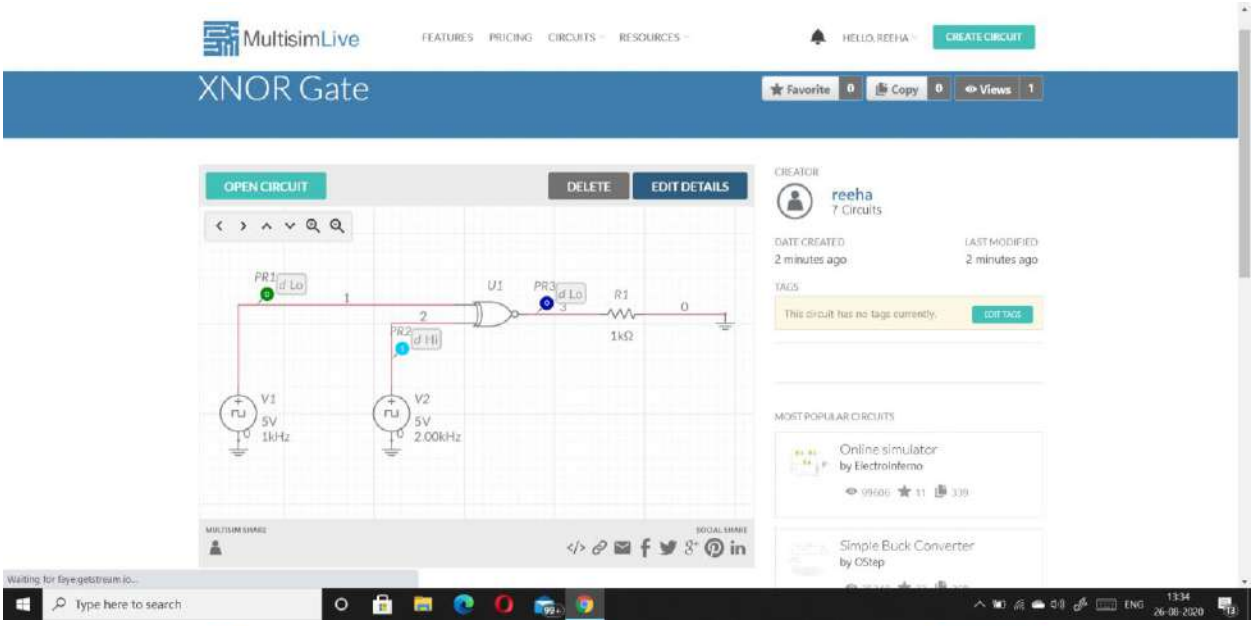
- Power supply should not exceed 5V.
- All the connections should be tight.
- Components should be tested before the practical.

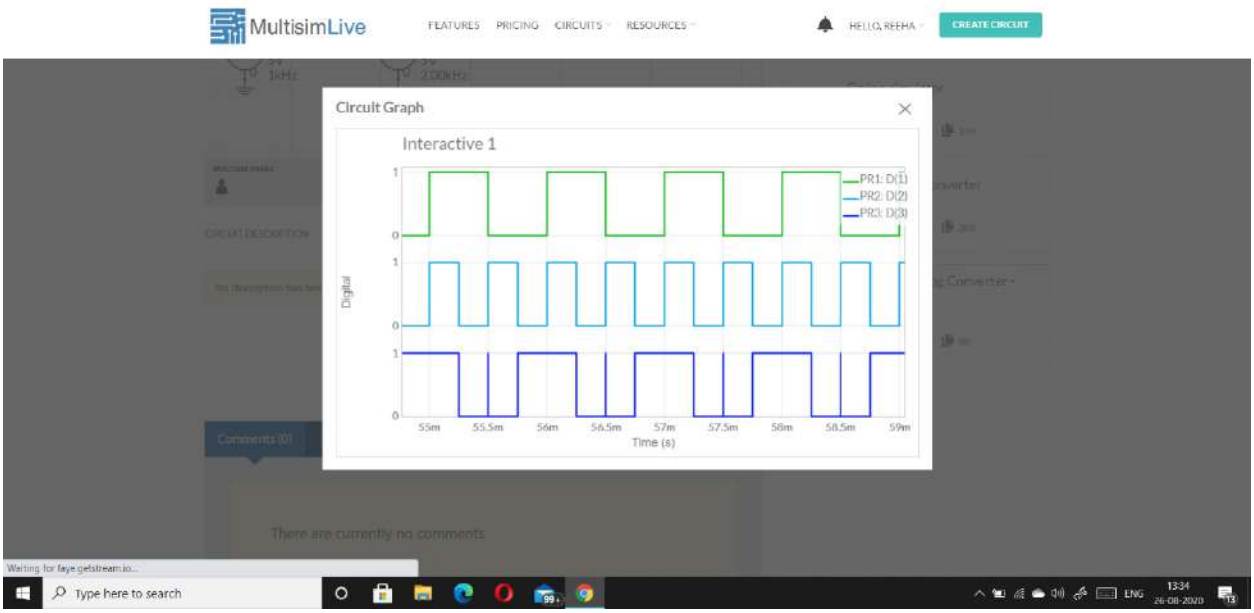
# NOT GATE


MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
OPEN CIRCUIT
DELETE
EDIT DETAILS
PR1
U1
PR2
R1
V1
5V
1kHz
1kΩ
CREATOR
reeha
7 Circuits
DATE CREATED
0 minutes ago
LAST MODIFIED
0 minutes ago
TAGS
MOST POPULAR CIRCUITS
Online simulator
by ElectroInferno
Simple Buck Converter
by OStep
CIRCUIT DESCRIPTION
Type here to search

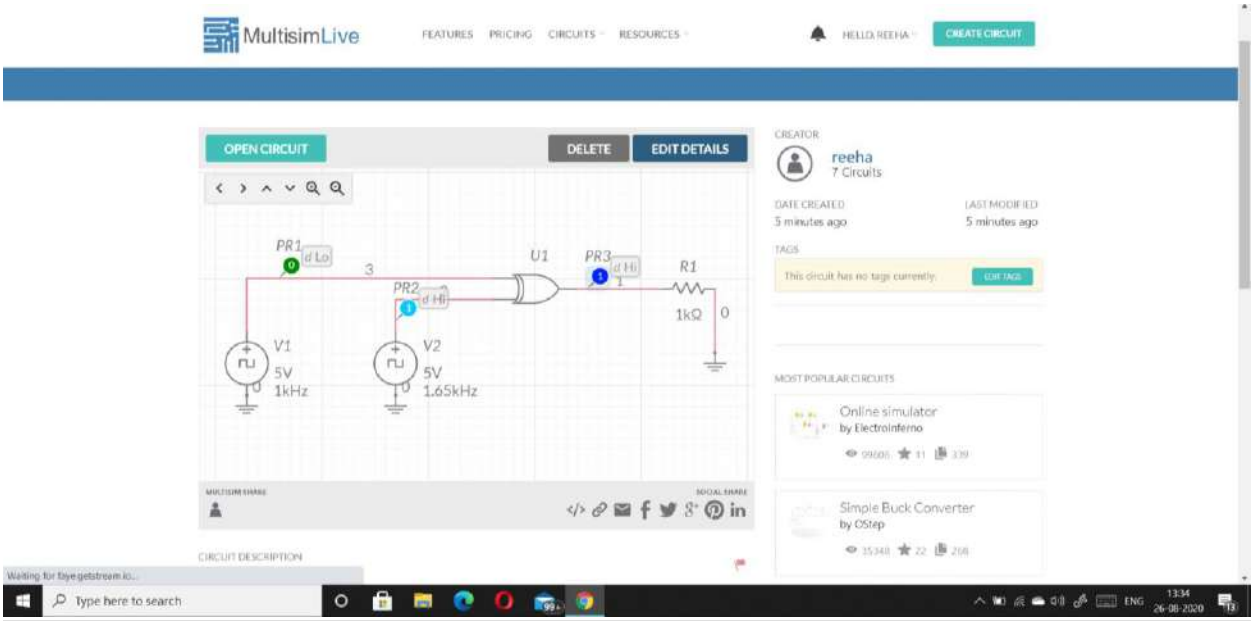

MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(2)
PR2: D(1)
Digital
Time (s)
62m
62.5m
63m
63.5m
64m
64.5m
65m
65.5m
66m
66.5m
Digital to Analog Converter - DAC
Type here to search

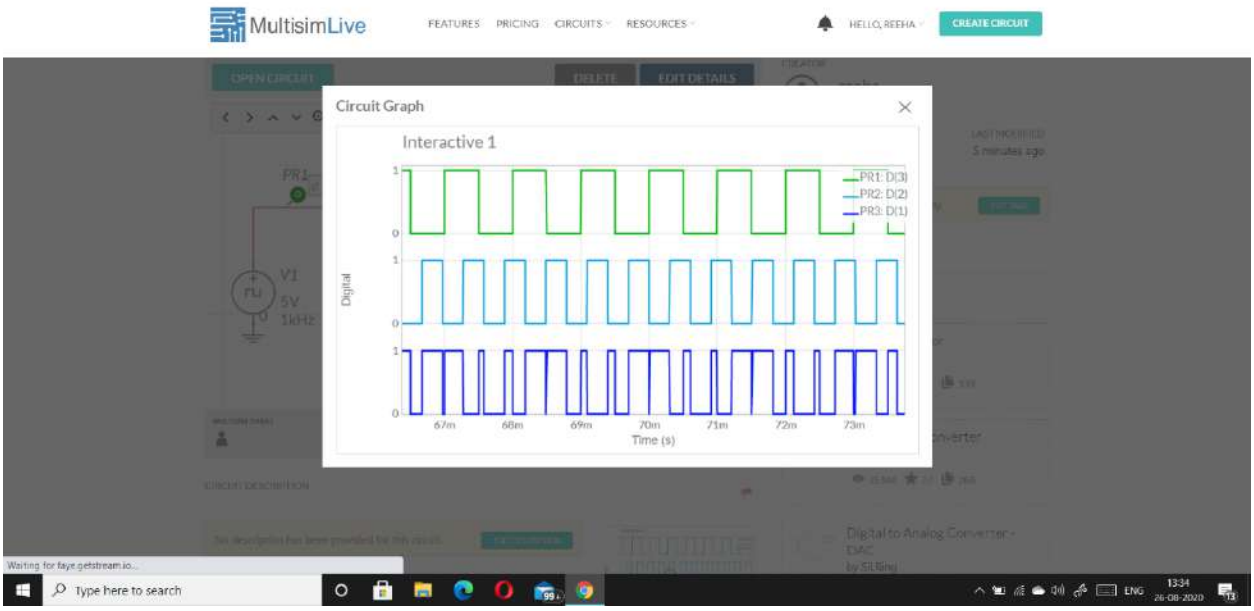
# XNOR GATE


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
PR2
d Hi
U1
PR3
d Lo
R1
1kΩ
V1
5V
1kHz
V2
5V
2.00kHz
CREATOR
reeha
7 Circuits
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 ElectroInferno
Simple Buck Converter
by OStep
Type here to search
ENG
13:34
26-08-2020

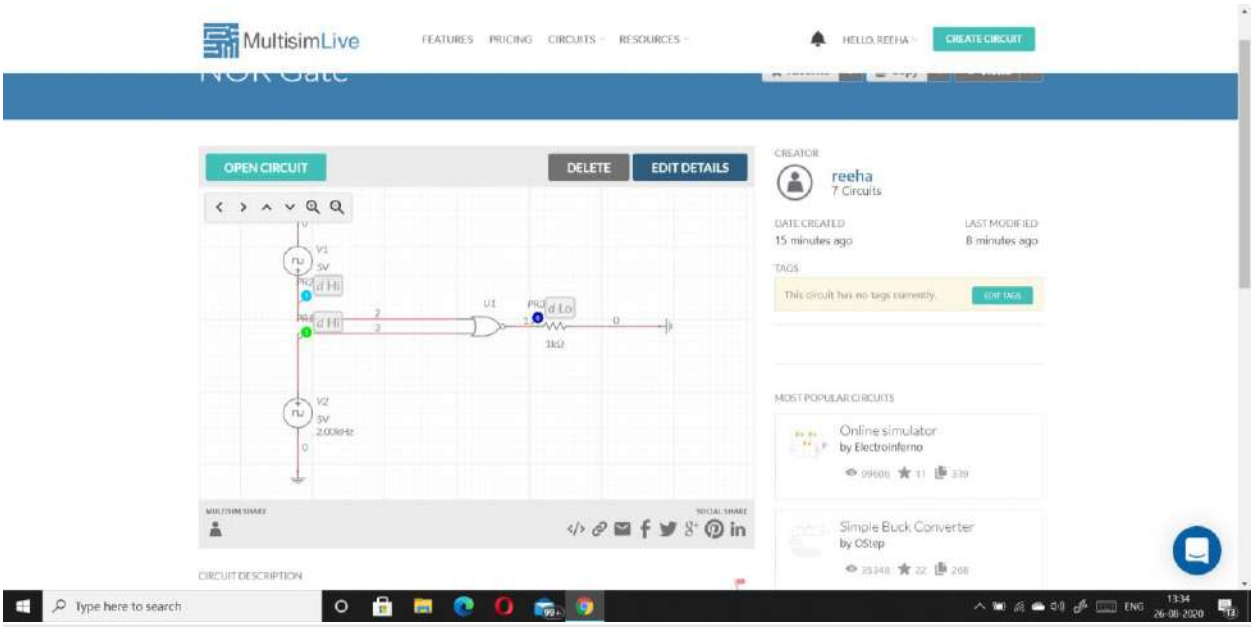

MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(1)
PR2: D(2)
PR3: D(3)
Digital
Time (s)
55m
55.5m
56m
56.5m
57m
57.5m
58m
58.5m
59m
There are currently no comments
Type here to search
ENG
13:34
26-08-2020

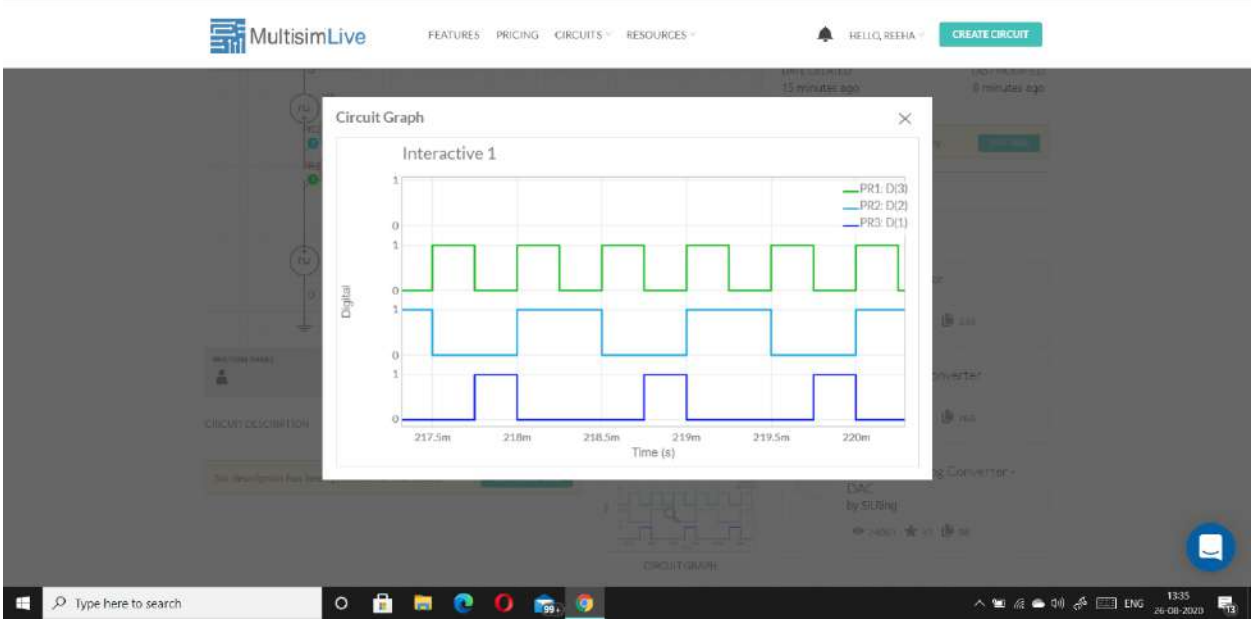
## XOR GATE


MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
OPEN CIRCUIT
DELETE
EDIT DETAILS
PR1
PR2
PR3
U1
R1
1kΩ
V1
5V
1kHz
V2
5V
1.65kHz
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
Simple Buck Converter
by OStep
CIRCUIT DESCRIPTION
Type here to search
ENG
13:34
26-08-2020

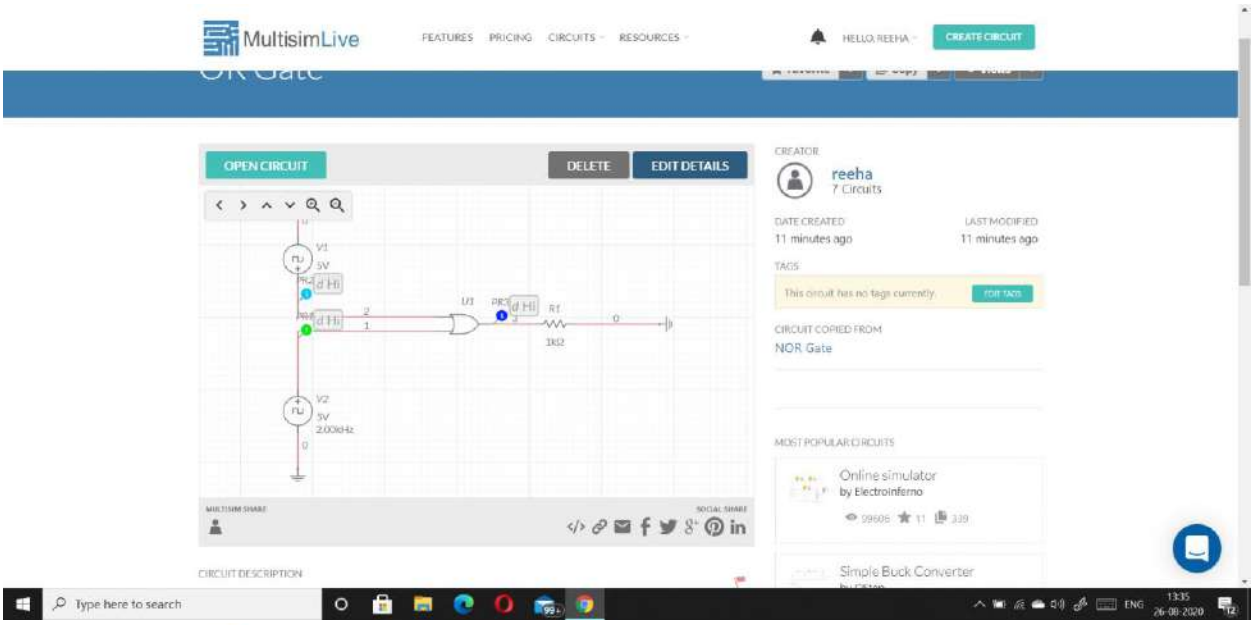

MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(3)
PR2: D(2)
PR3: D(1)
Digital
Time (s)
67m
68m
69m
70m
71m
72m
73m
Type here to search
ENG
13:34
26-08-2020

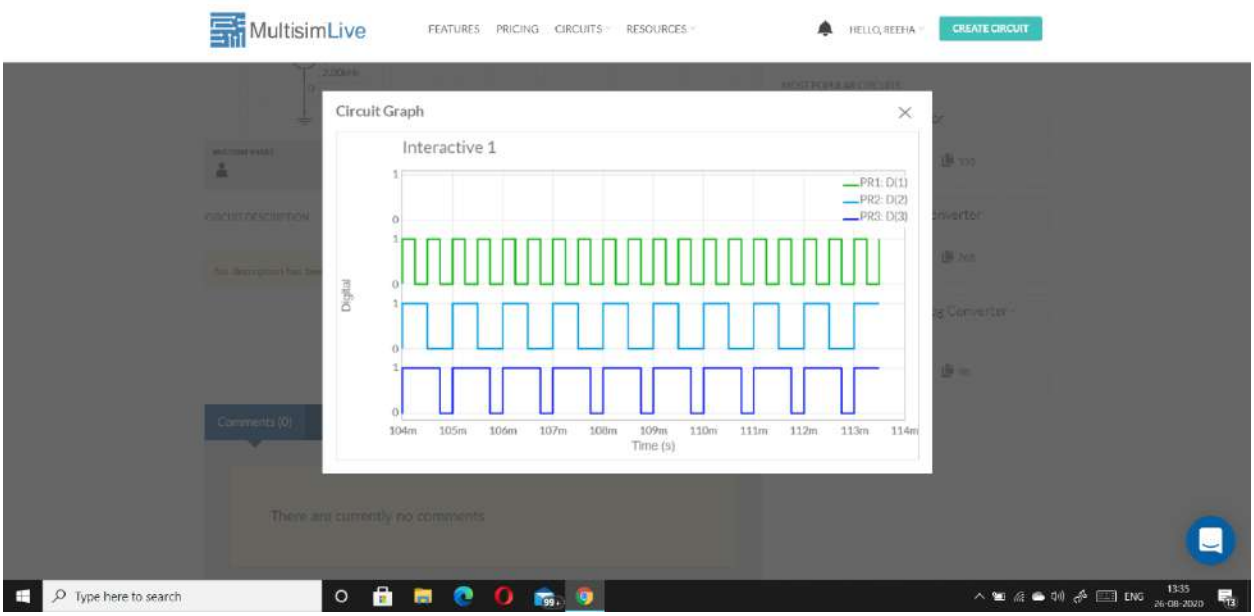
## NOR GATE


MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
OPEN CIRCUIT
DELETE
EDIT DETAILS
CREATOR
reeha
7 Circuits
DATE CREATED
15 minutes ago
LAST MODIFIED
8 minutes ago
TAGS
This circuit has no tags currently.
MOST POPULAR CIRCUITS
Online simulator
by Electroinferno
Simple Buck Converter
by OStep
CIRCUIT DESCRIPTION
Type here to search
ENG
13:34
26-08-2020

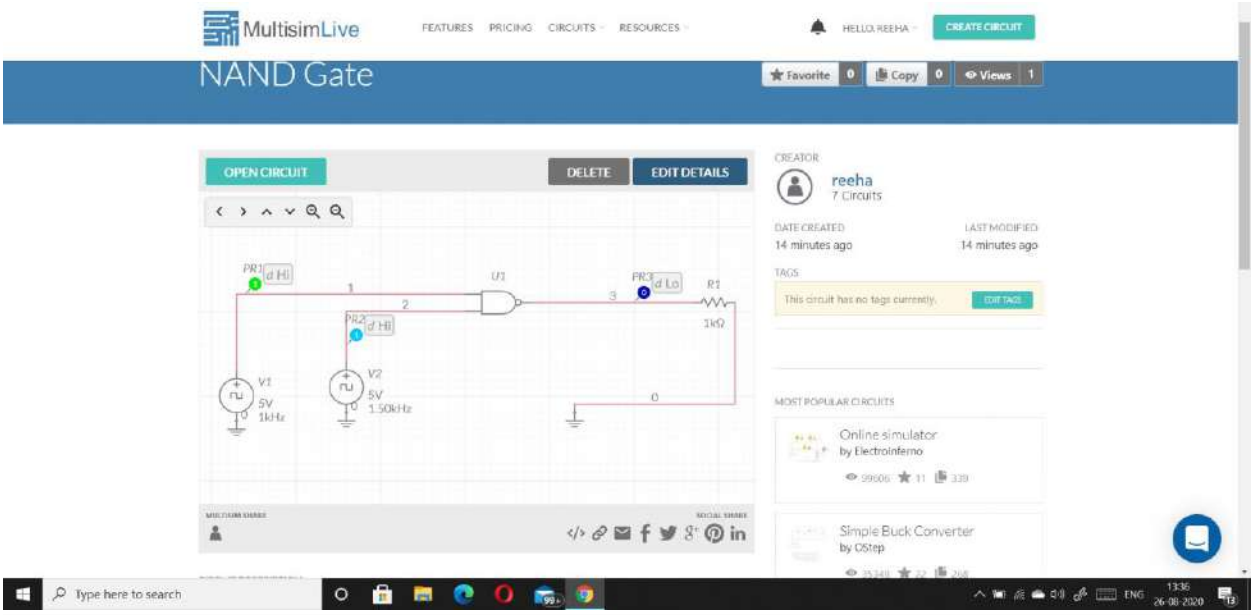

MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(3)
PR2: D(2)
PR3: D(1)
Digital
217.5m
218m
218.5m
219m
219.5m
220m
Time (s)
Type here to search
ENG
13:35
26-08-2020

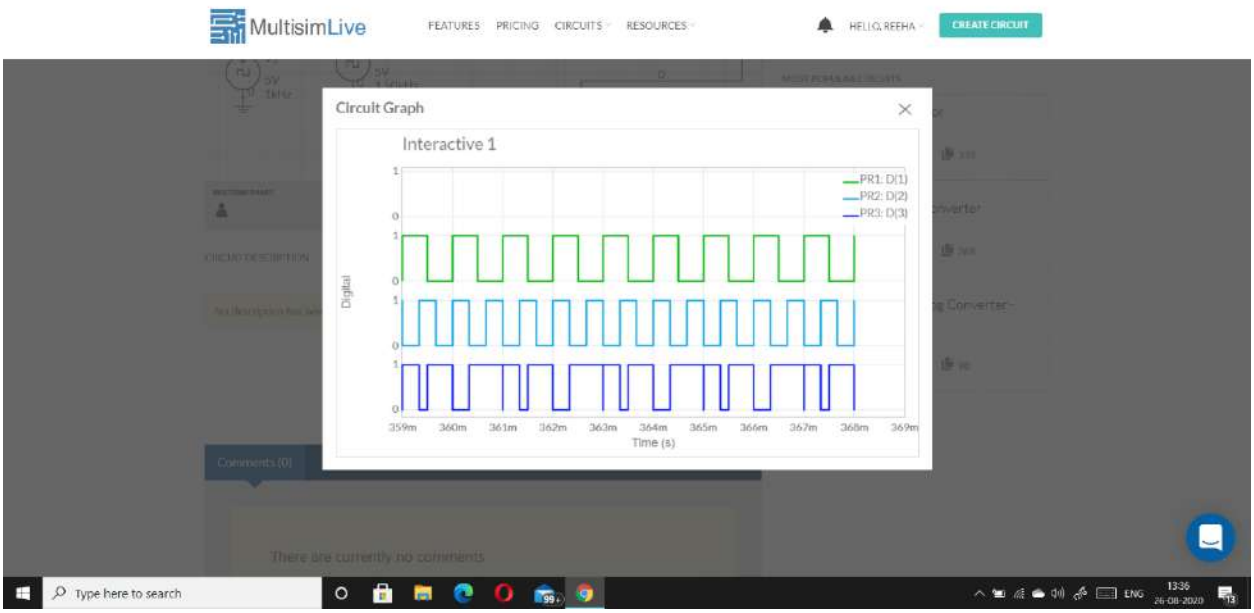
## OR GATE


MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
OR Gate
OPEN CIRCUIT
DELETE
EDIT DETAILS
CREATOR
reeha
7 Circuits
DATE CREATED
11 minutes ago
LAST MODIFIED
11 minutes ago
TAGS
This circuit has no tags currently.
EDIT TAGS
CIRCUIT COPIED FROM
NOR Gate
MOST POPULAR CIRCUITS
Online simulator
by Electroinferno
Simple Buck Converter
CIRCUIT DESCRIPTION
V1
5V
V2
5V
2.000kHz
U1
R1
1kΩ
Type here to search
ENG
13:35
26-08-2020

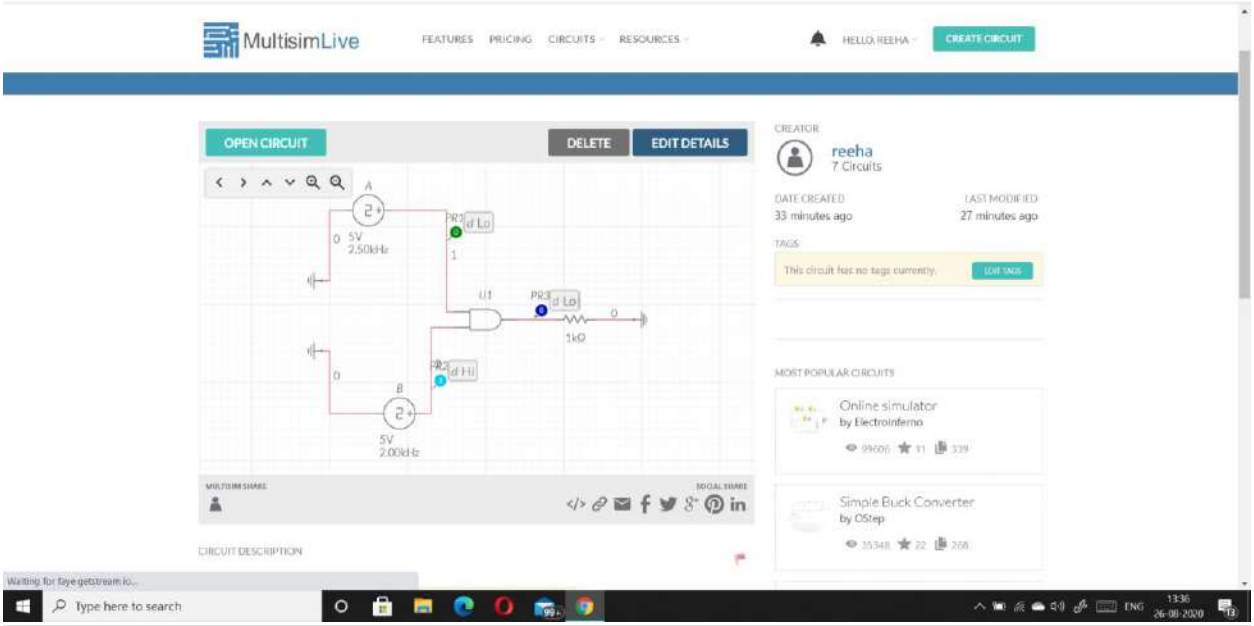

MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(1)
PR2: D(2)
PR3: D(3)
Digital
Time (s)
104m
105m
106m
107m
108m
109m
110m
111m
112m
113m
114m
There are currently no comments
Type here to search
ENG
13:35
26-08-2020

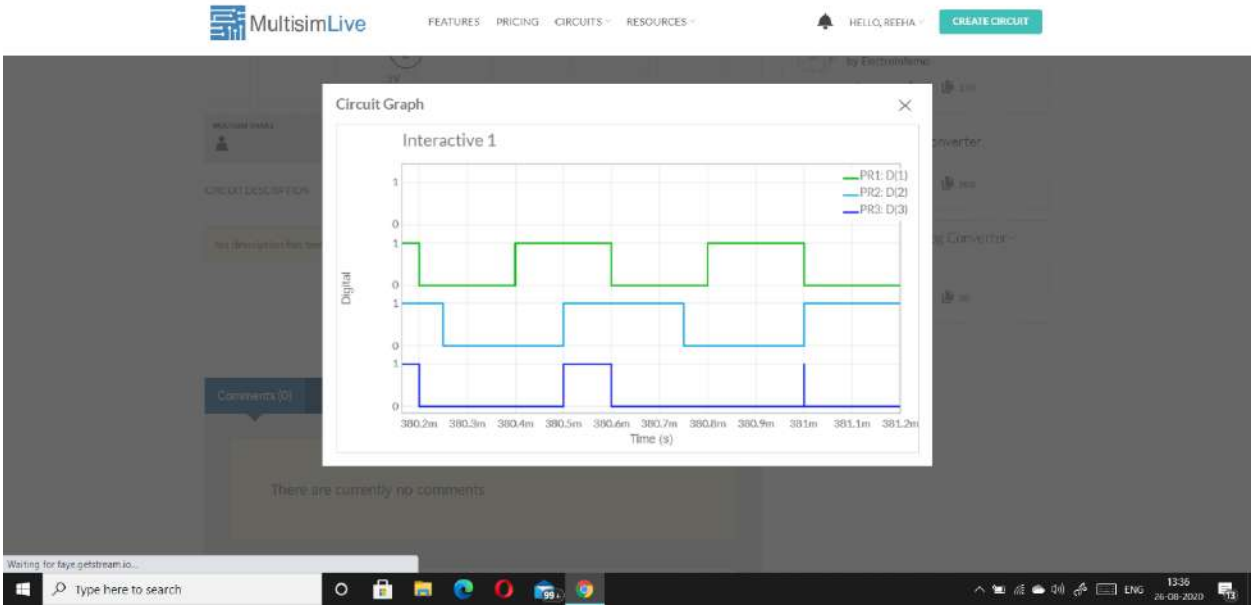
## NAND GATE


MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
NAND Gate
Favorite 0
Copy 0
Views 1
OPEN CIRCUIT
DELETE
EDIT DETAILS
PR1
d Hi
U1
PR3
d Lo
R1
1kΩ
PR2
d Hi
V1
5V
1kHz
V2
5V
1.50kHz
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 ElectroInferno
Simple Buck Converter
by OStep
Type here to search
ENG
13:36
26-08-2020


MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(1)
PR2: D(2)
PR3: D(3)
Digital
Time (s)
359m
360m
361m
362m
363m
364m
365m
366m
367m
368m
369m
There are currently no comments
Type here to search
ENG
13:36
26-08-2020

## 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.

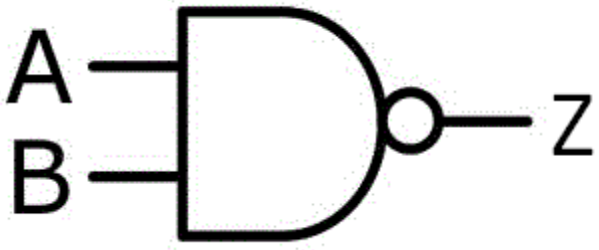
## Hardware and Software Apparatus Required

*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: Sign Up and create a profile.

## Circuit (SYMBOL)

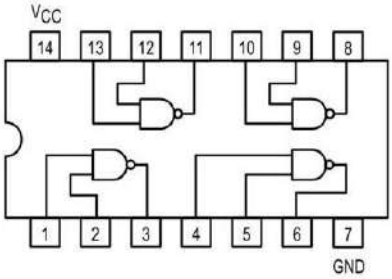



**Digital Logic NAND Gate – Universal Gate**

**Output = $\overline{A.B}$**

## 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

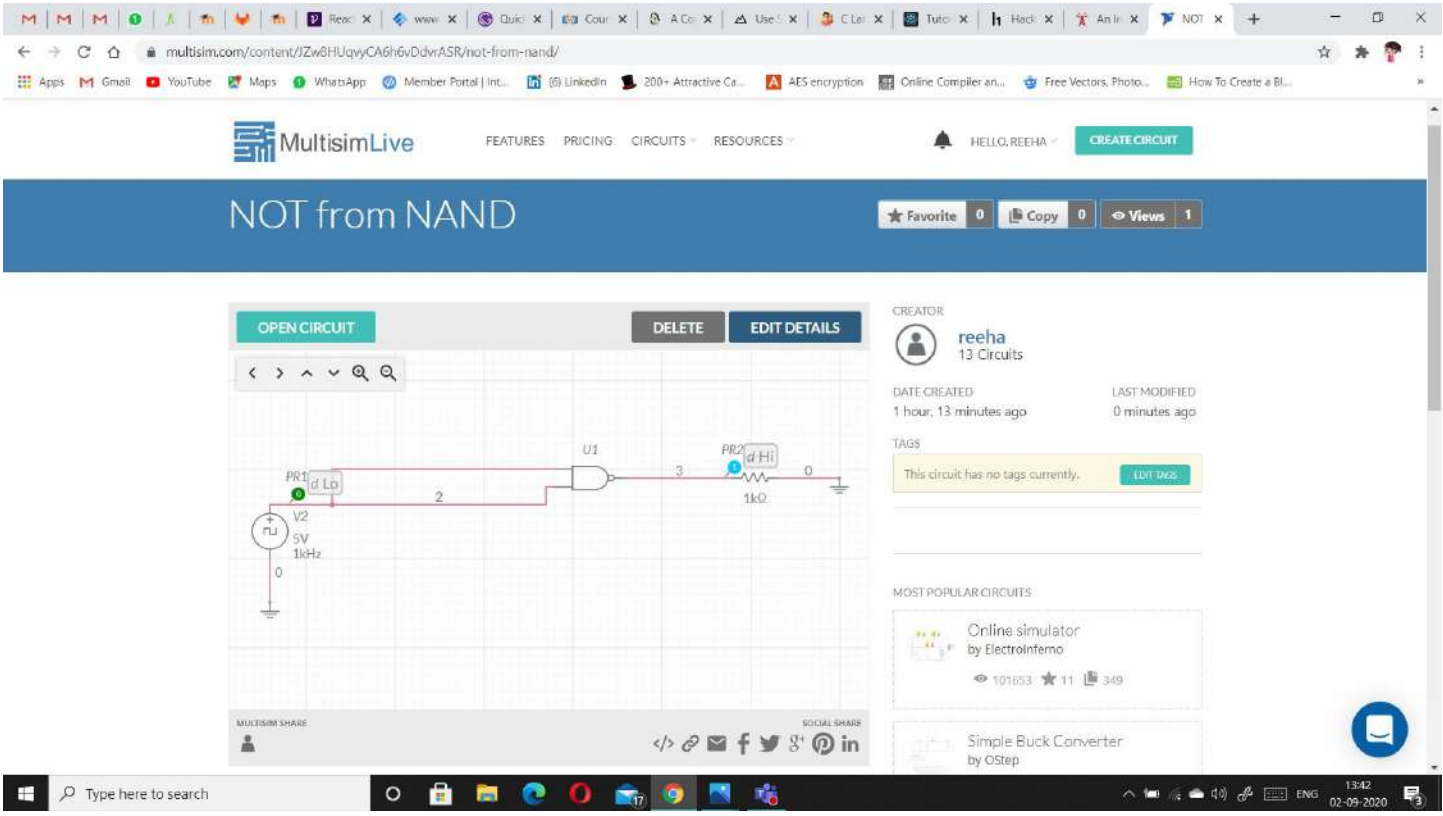
- The Boolean expression for a logic NAND gate is denoted by a single dot or full stop symbol, (.) with a line or Overline, (¯) over the expression to signify the NOT or logical negation of the NAND gate giving us the Boolean expression of: A.B = Q.
- It is a series of AND gate followed by NOT gate. 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

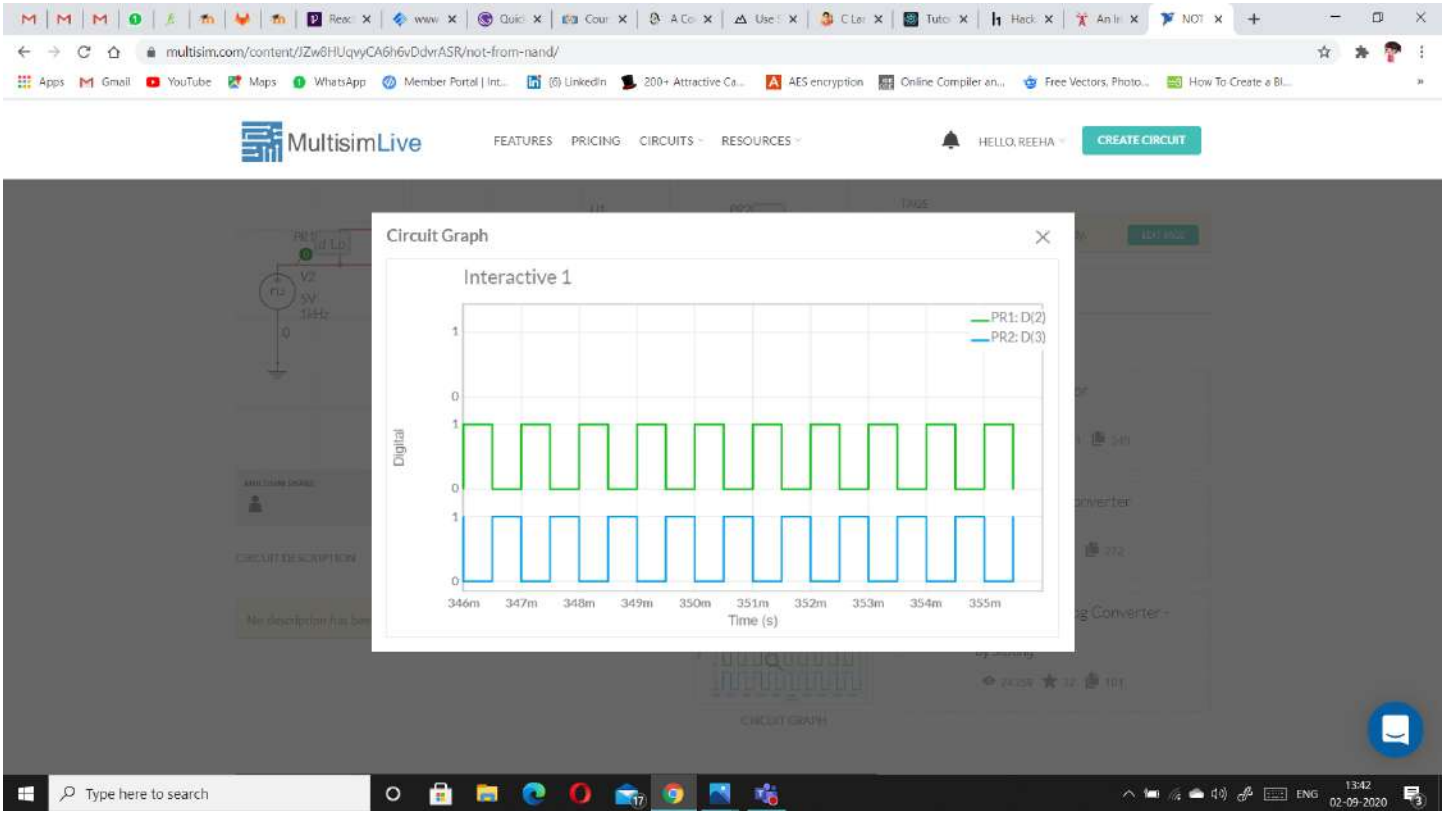
1. LOG IN ON www.multisim.com
2. CREATE THE CIRCUIT
3. SAVE THE CIRCUIT

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)

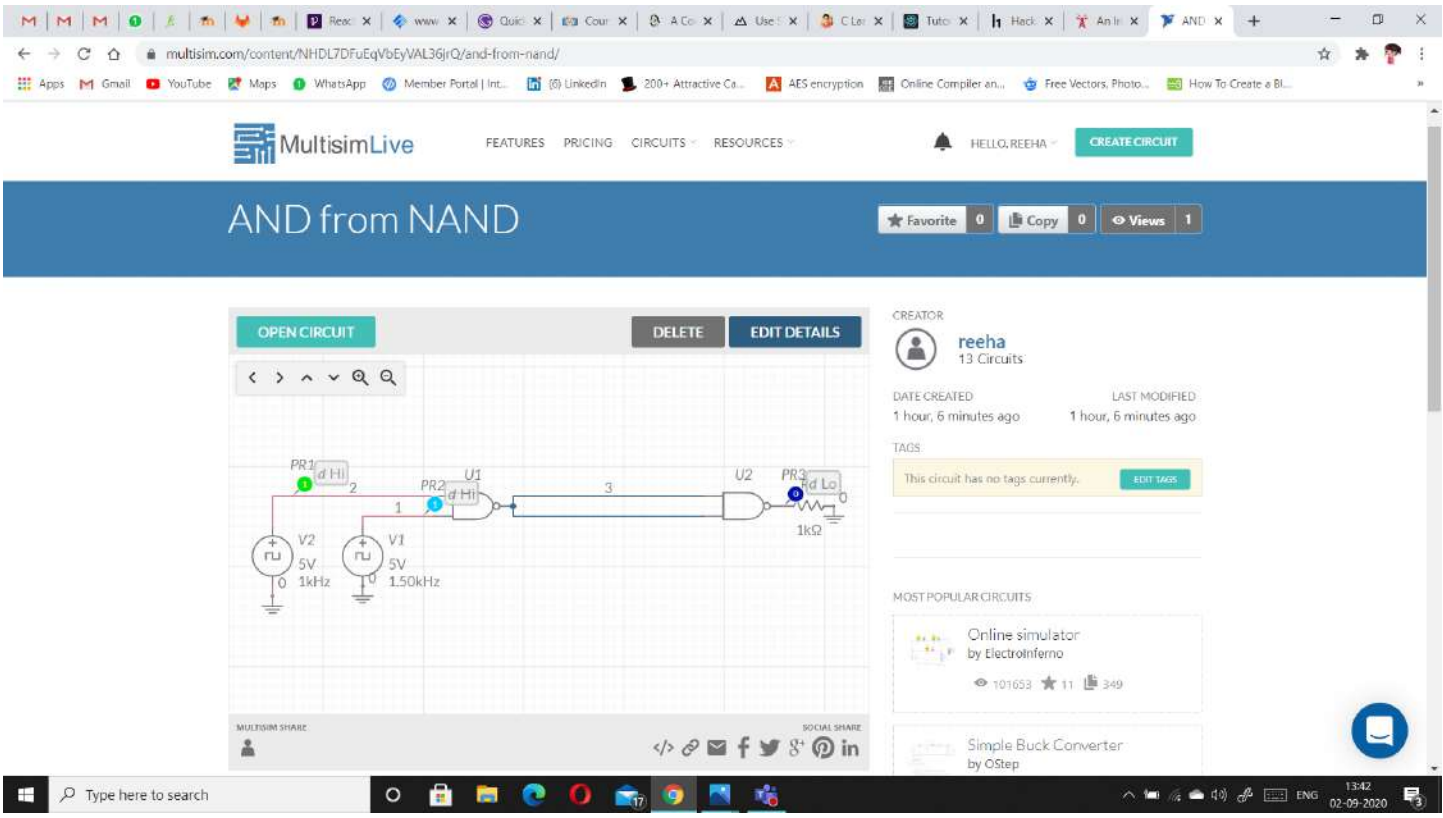
# Screenshot of Circuit: *NOT(INVERTER) USING NAND*



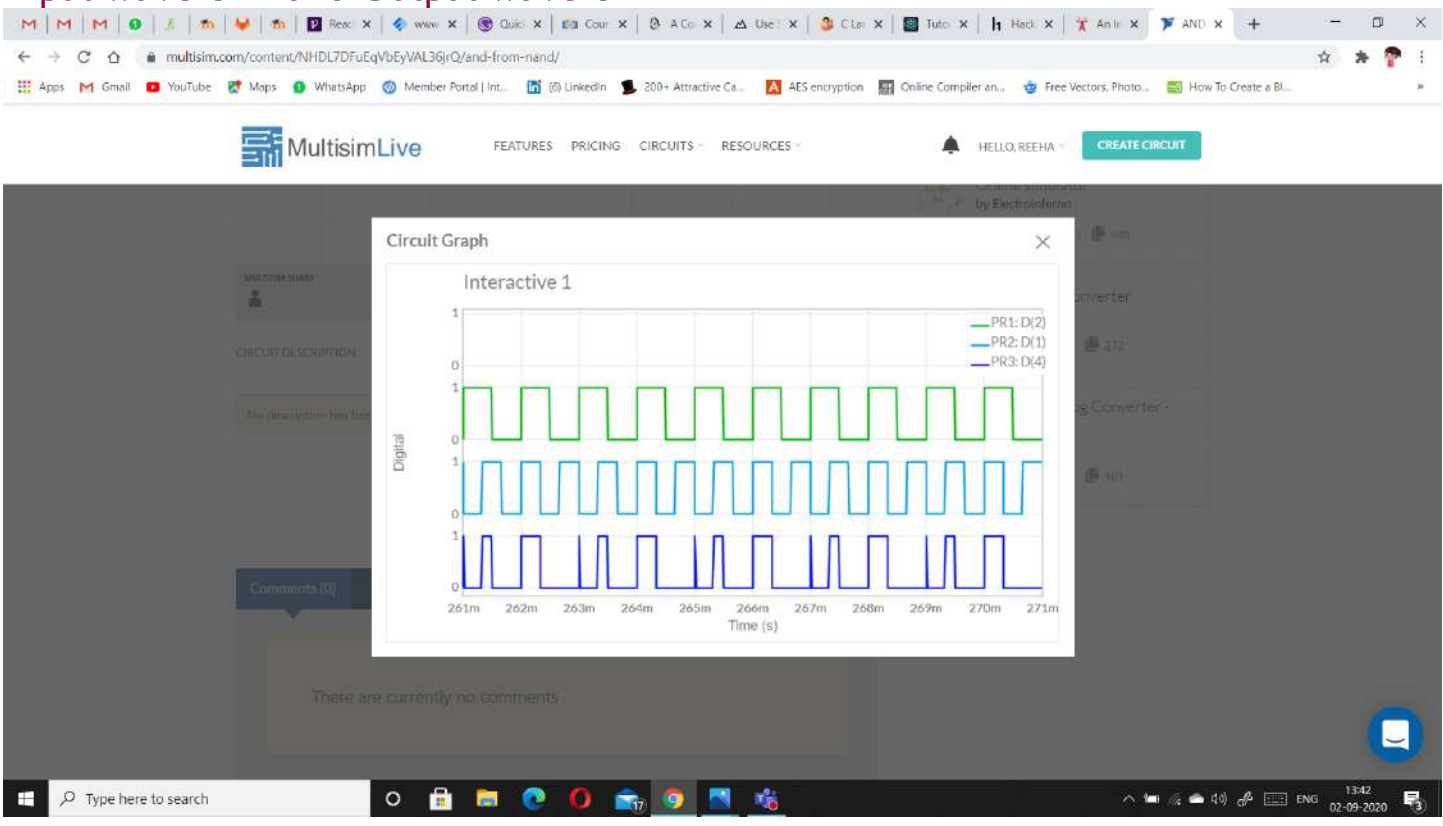
## Input waveform and Output waveform



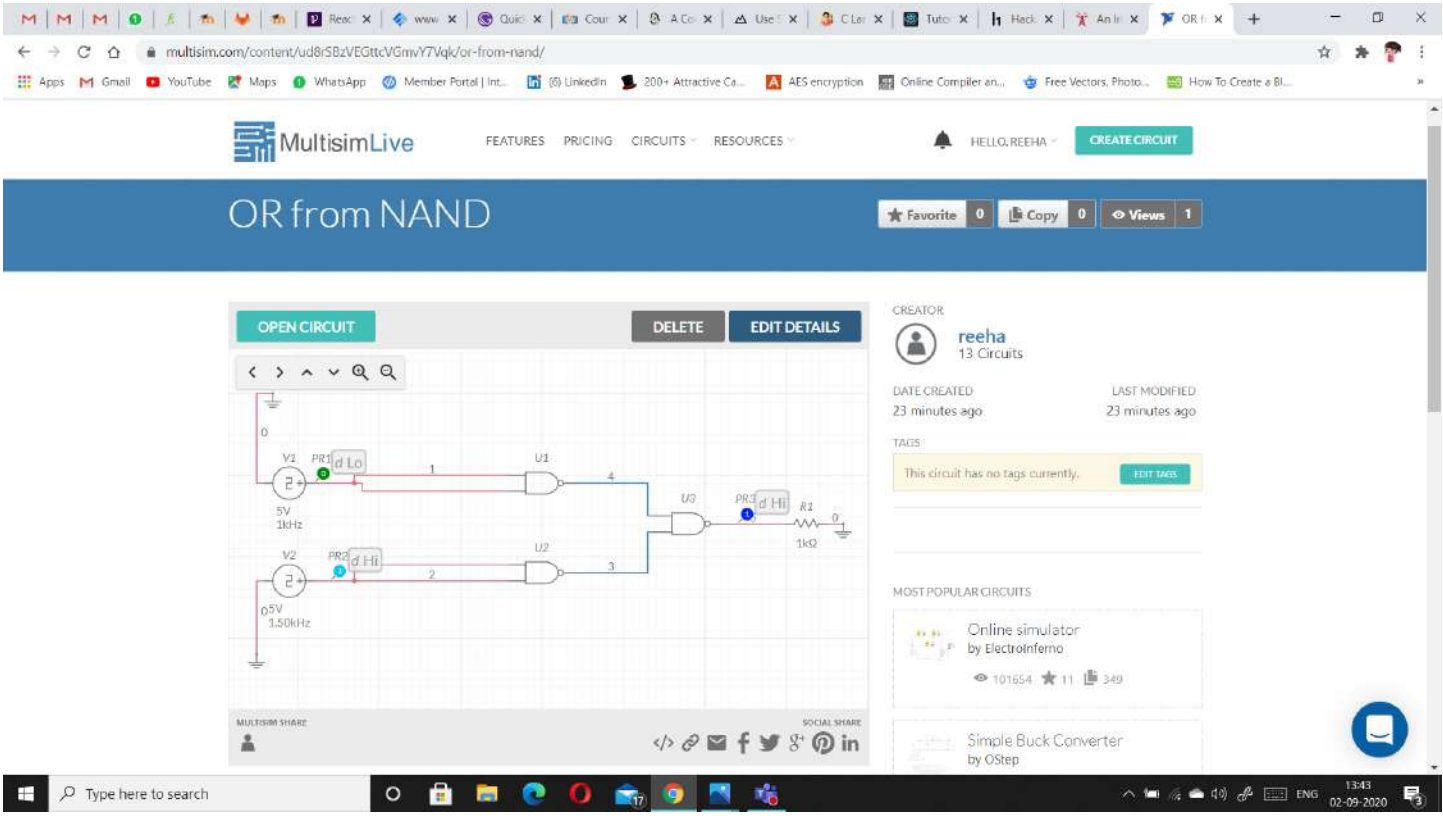

## Screenshot of Circuit: *AND USING NAND*



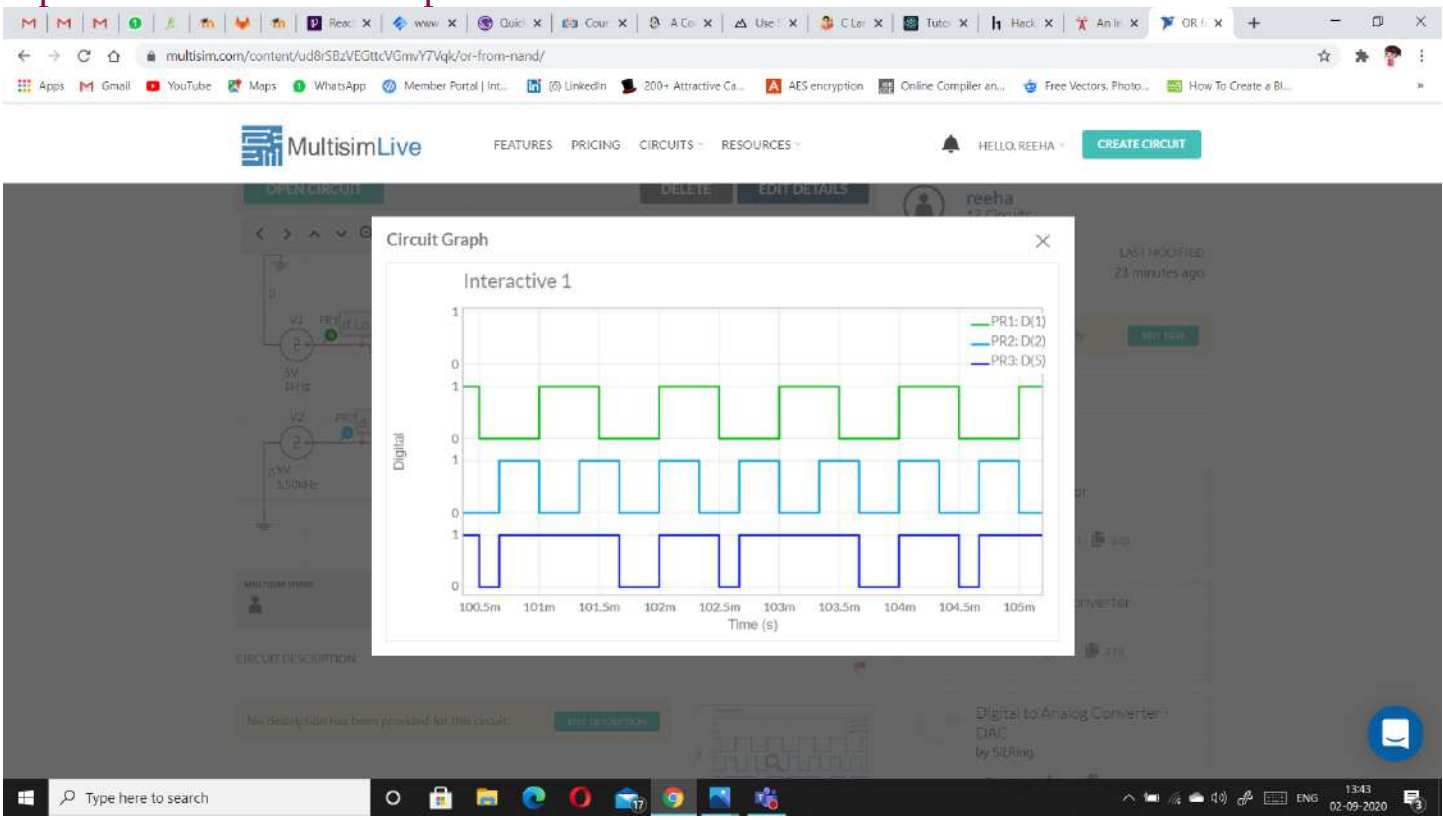

# Input waveform and Output waveform



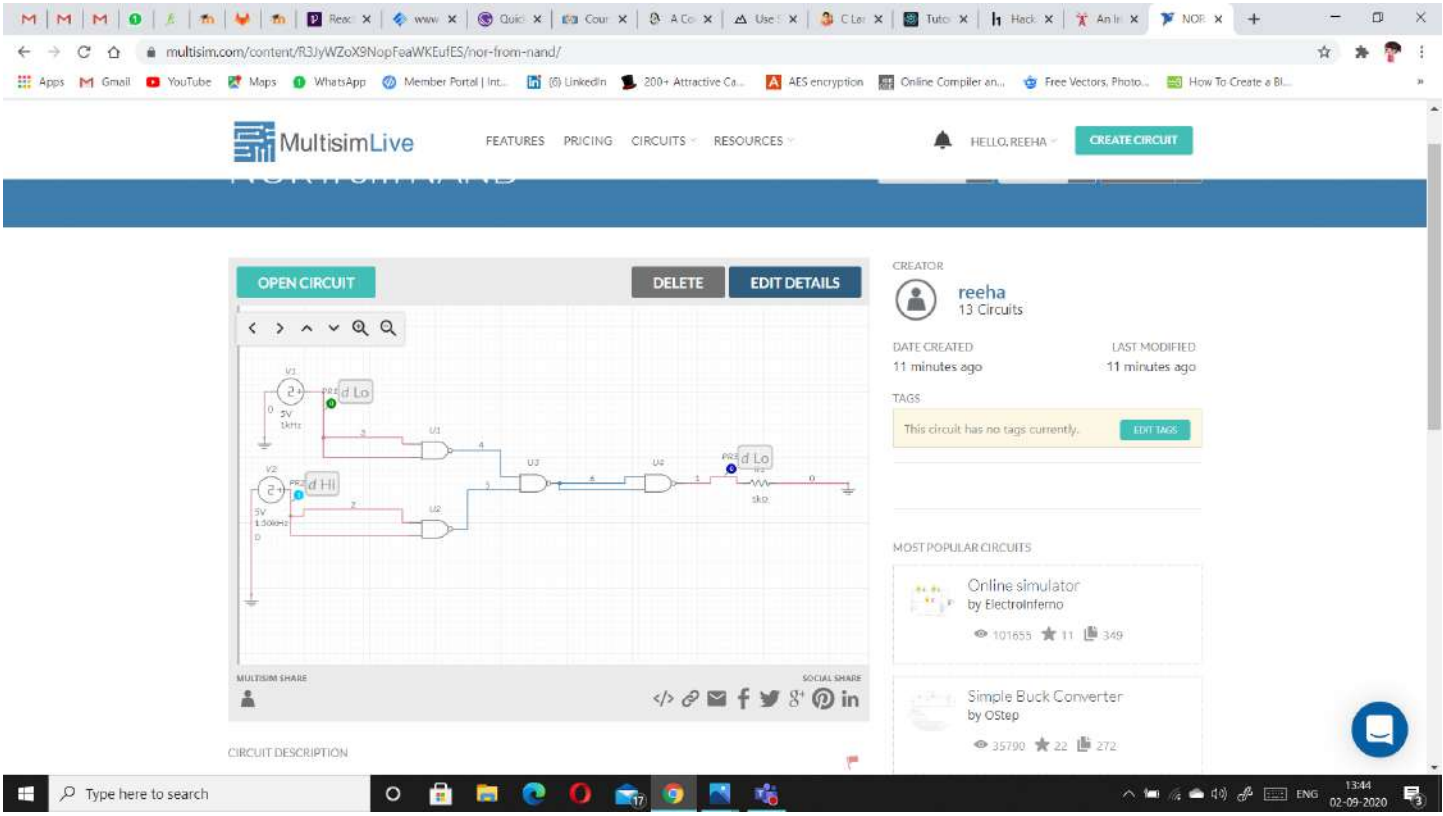
# Screenshot of Circuit: *OR USING NAND*



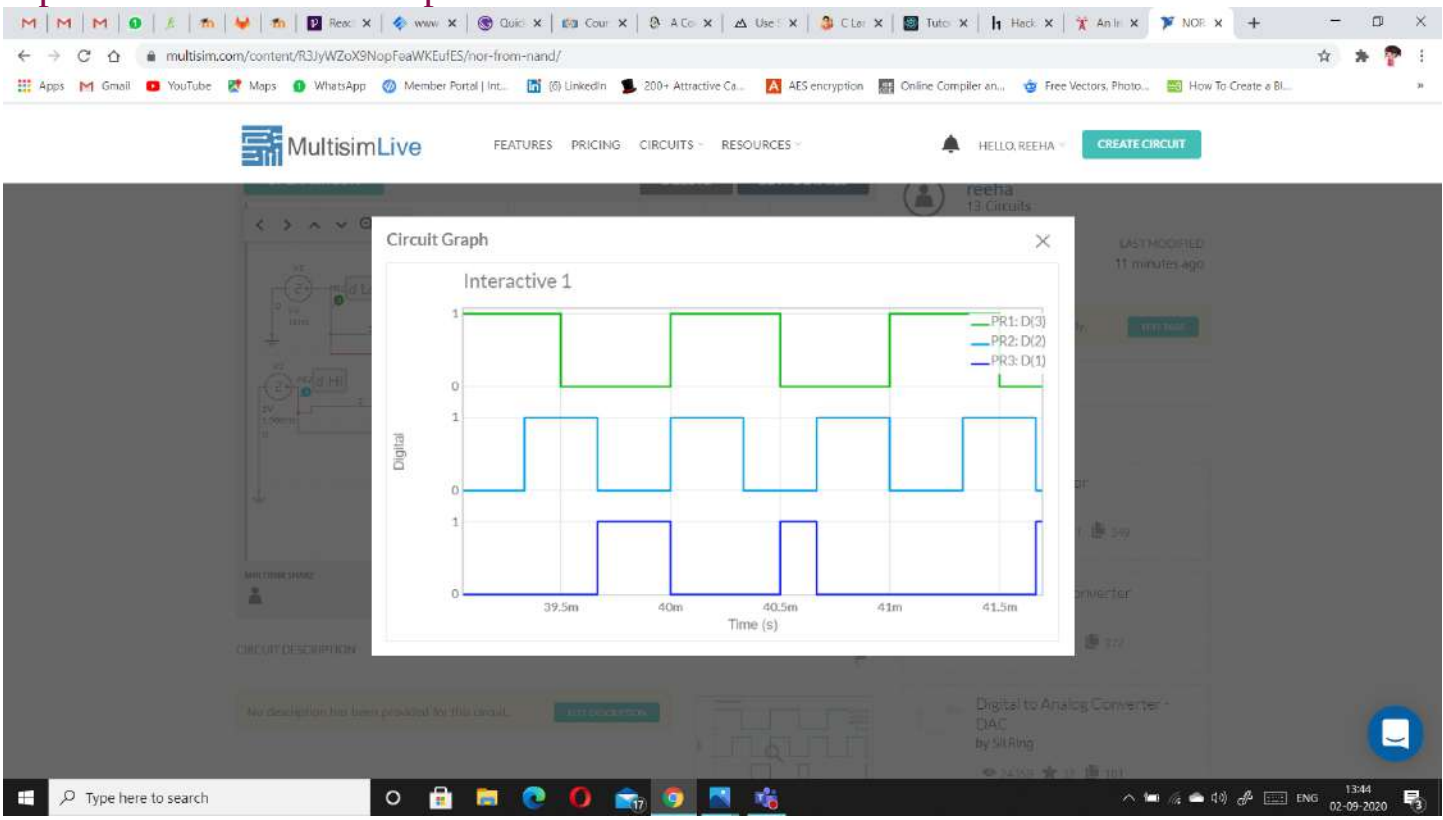
# Input waveform and Output waveform



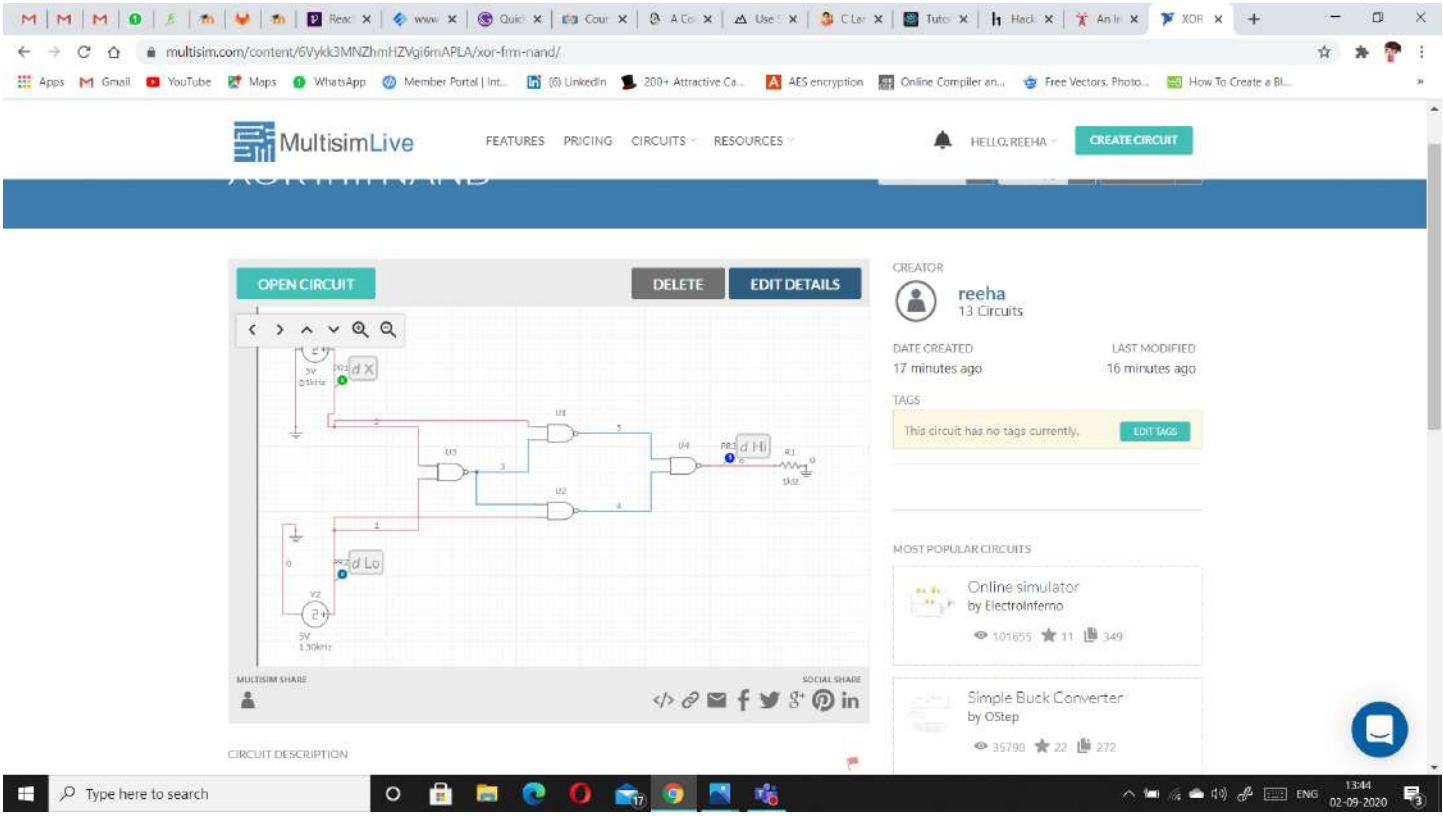
# Screenshot of Circuit: *NOR USING NAND*



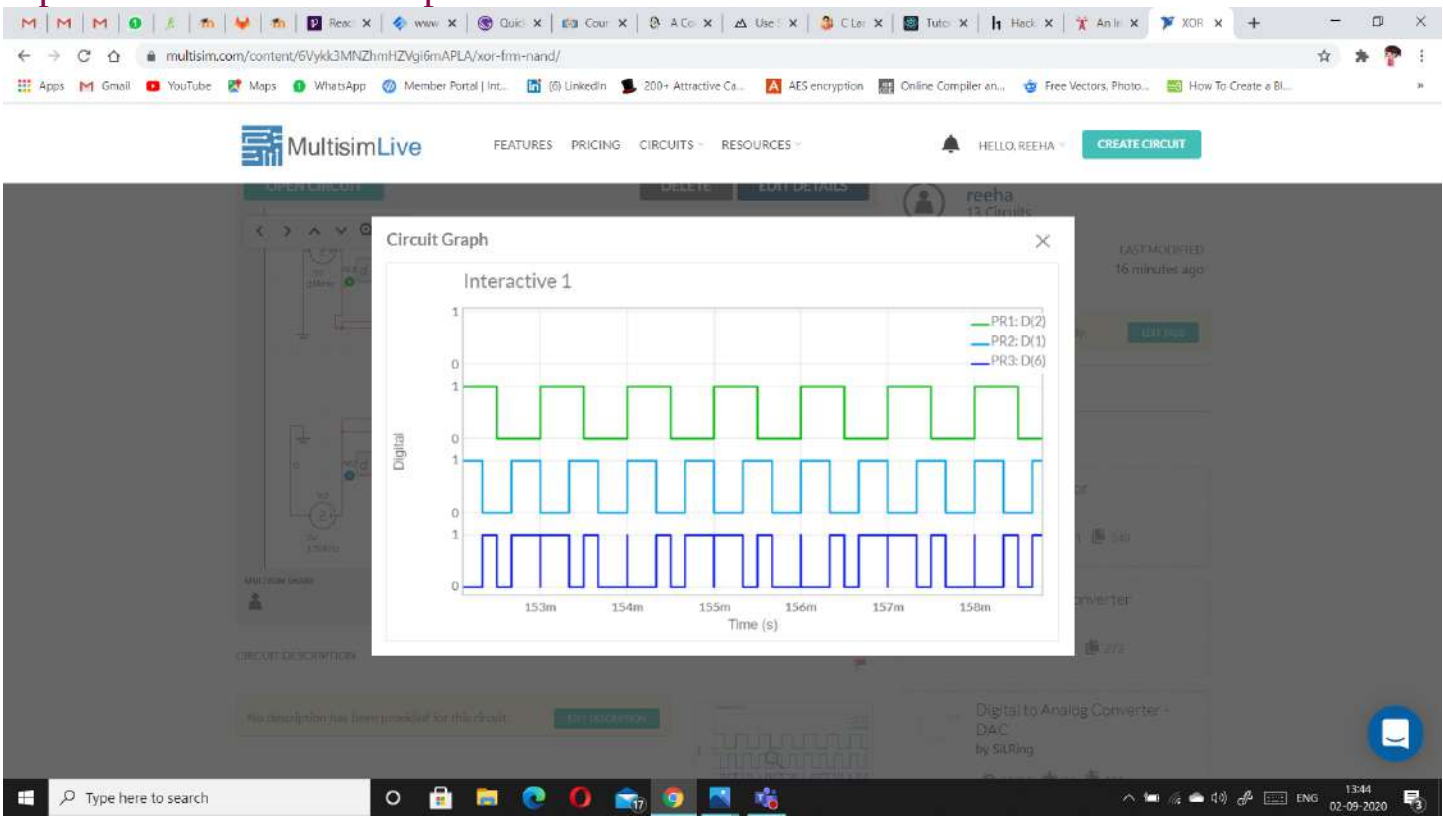
# Input waveform and Output waveform



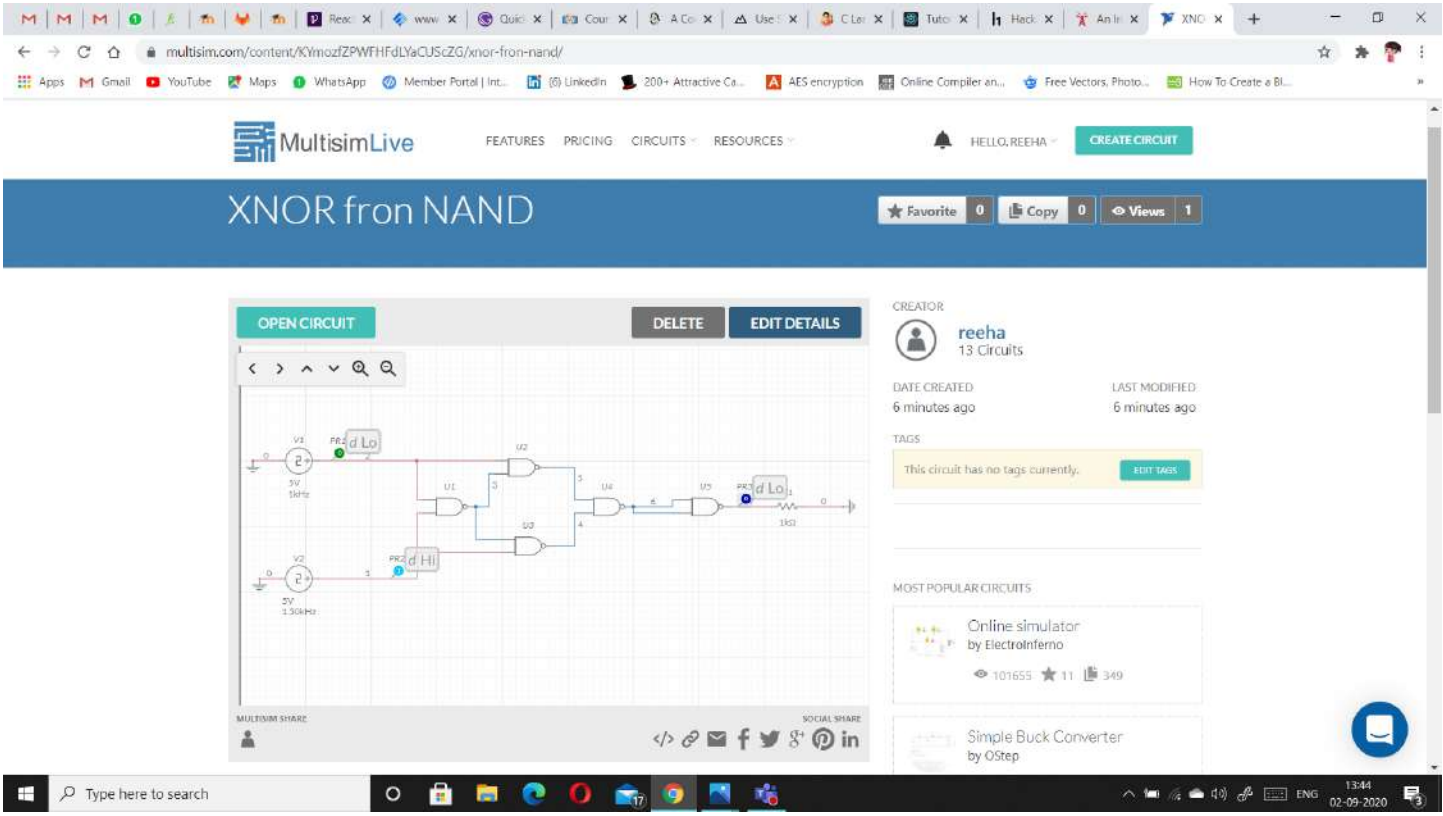
# Screenshot of Circuit: *XOR USING NAND*



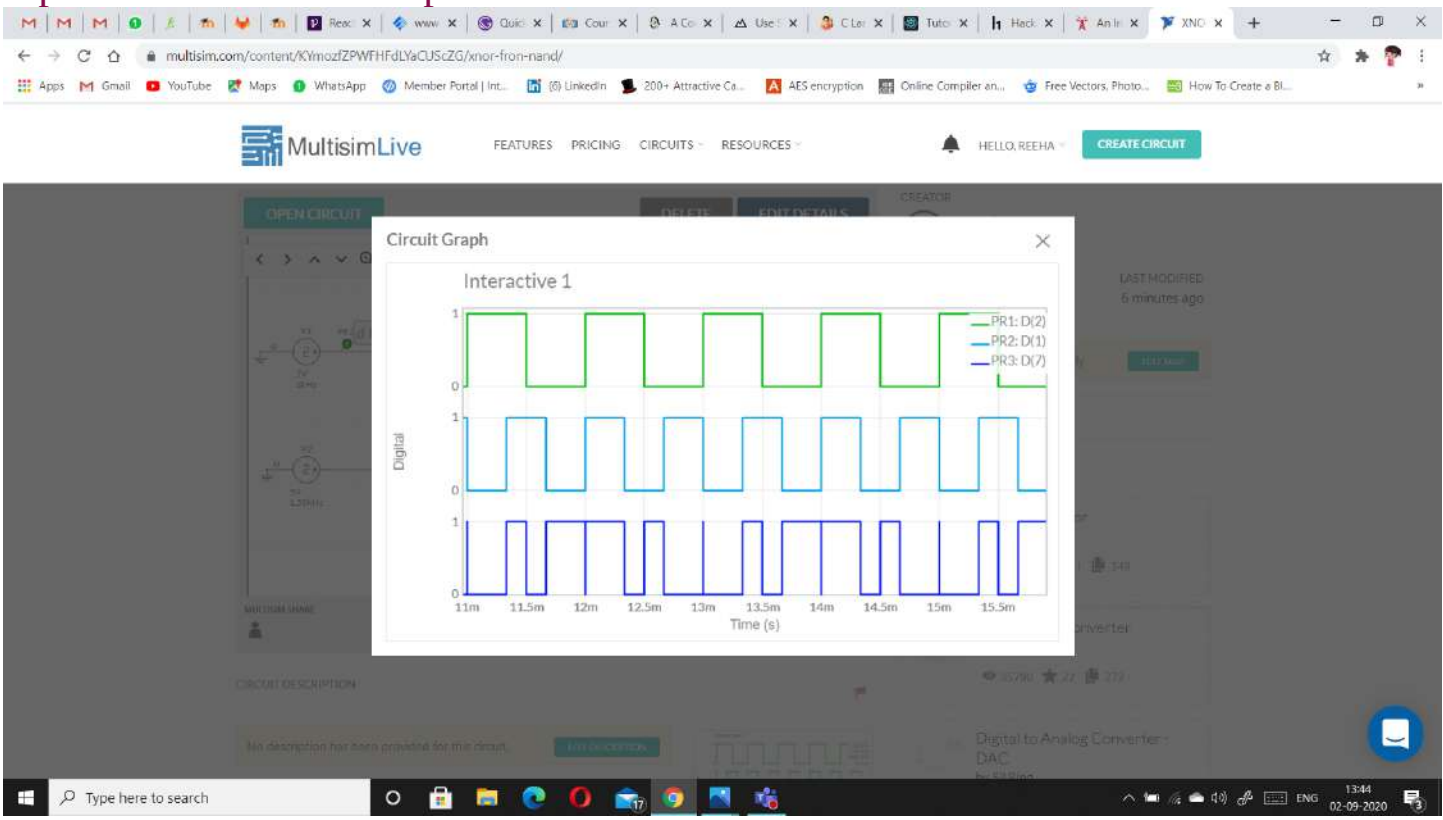
# Input waveform and Output waveform



# Screenshot of Circuit: *XNOR USING NAND*



# 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.

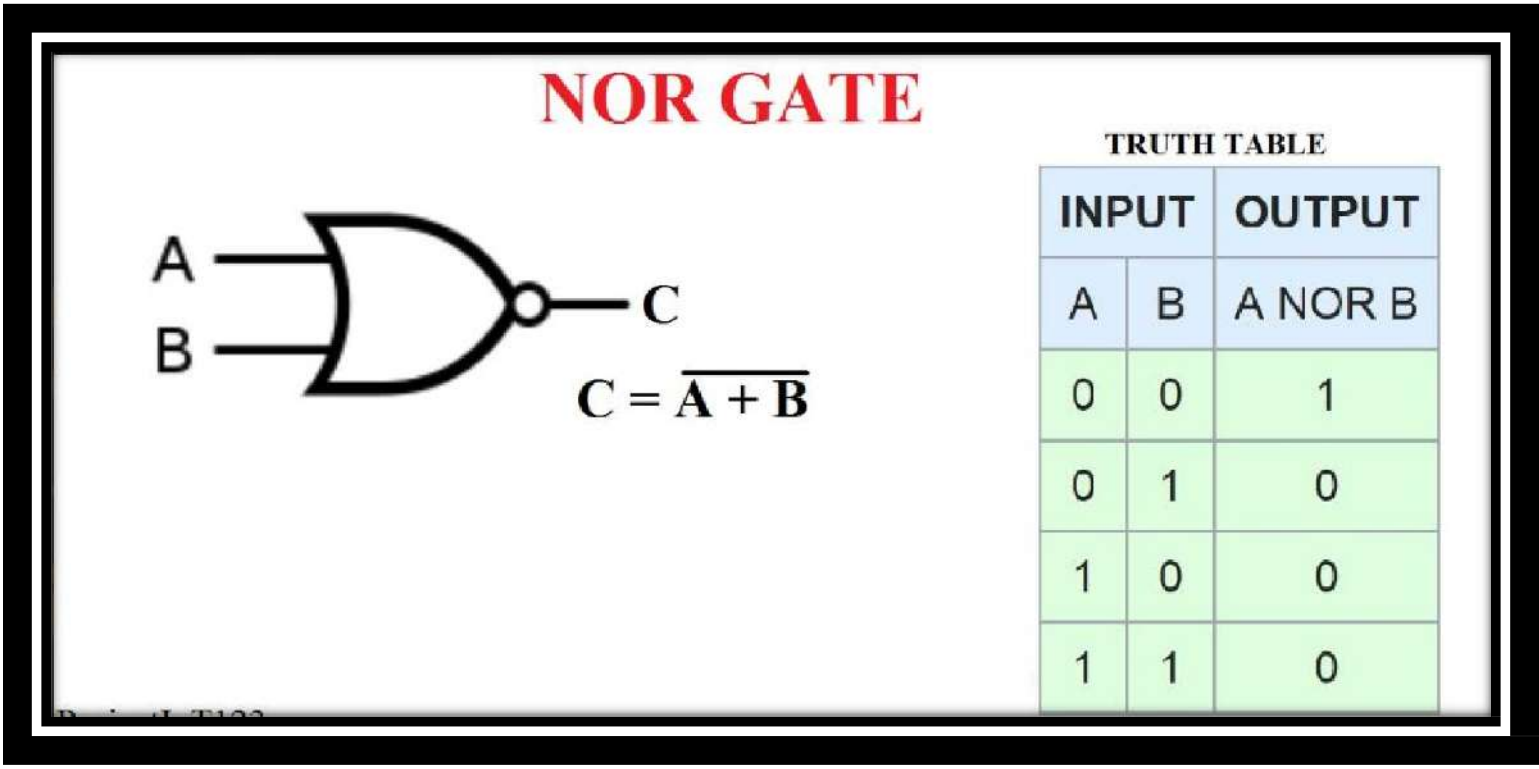
### Software Simulation:

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

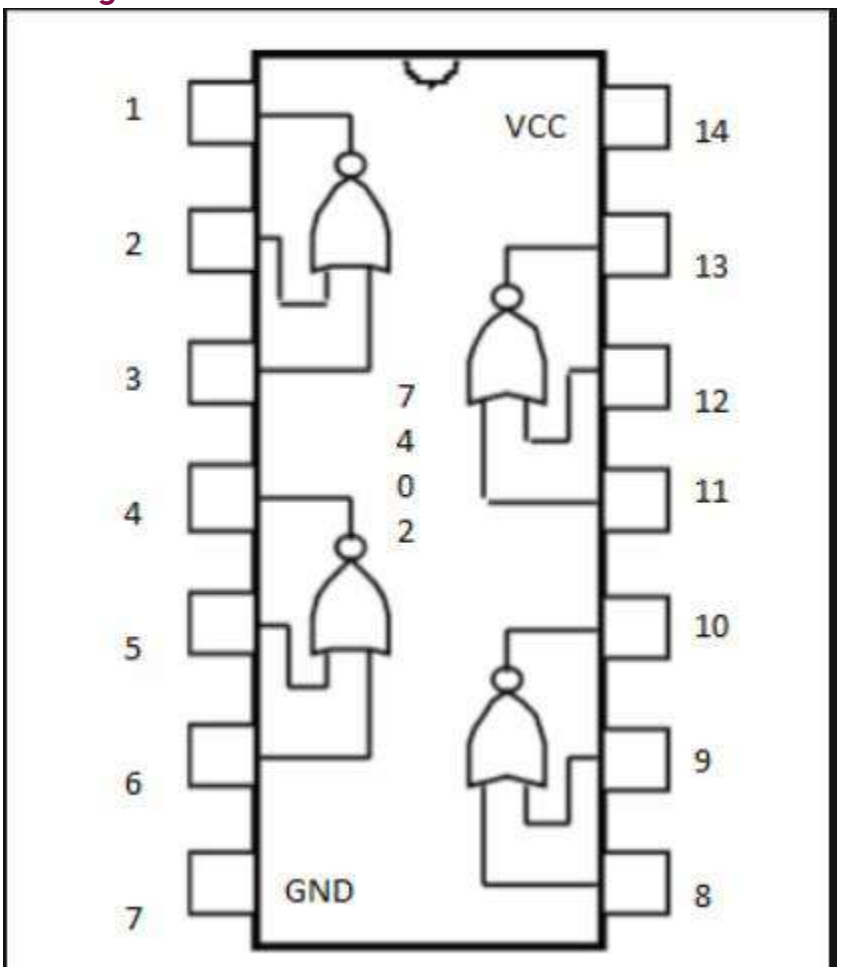
Components used – Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

## Theory:

### Circuit Symbol and Truth table:



***Pin Diagram:***




**NOR gate:**

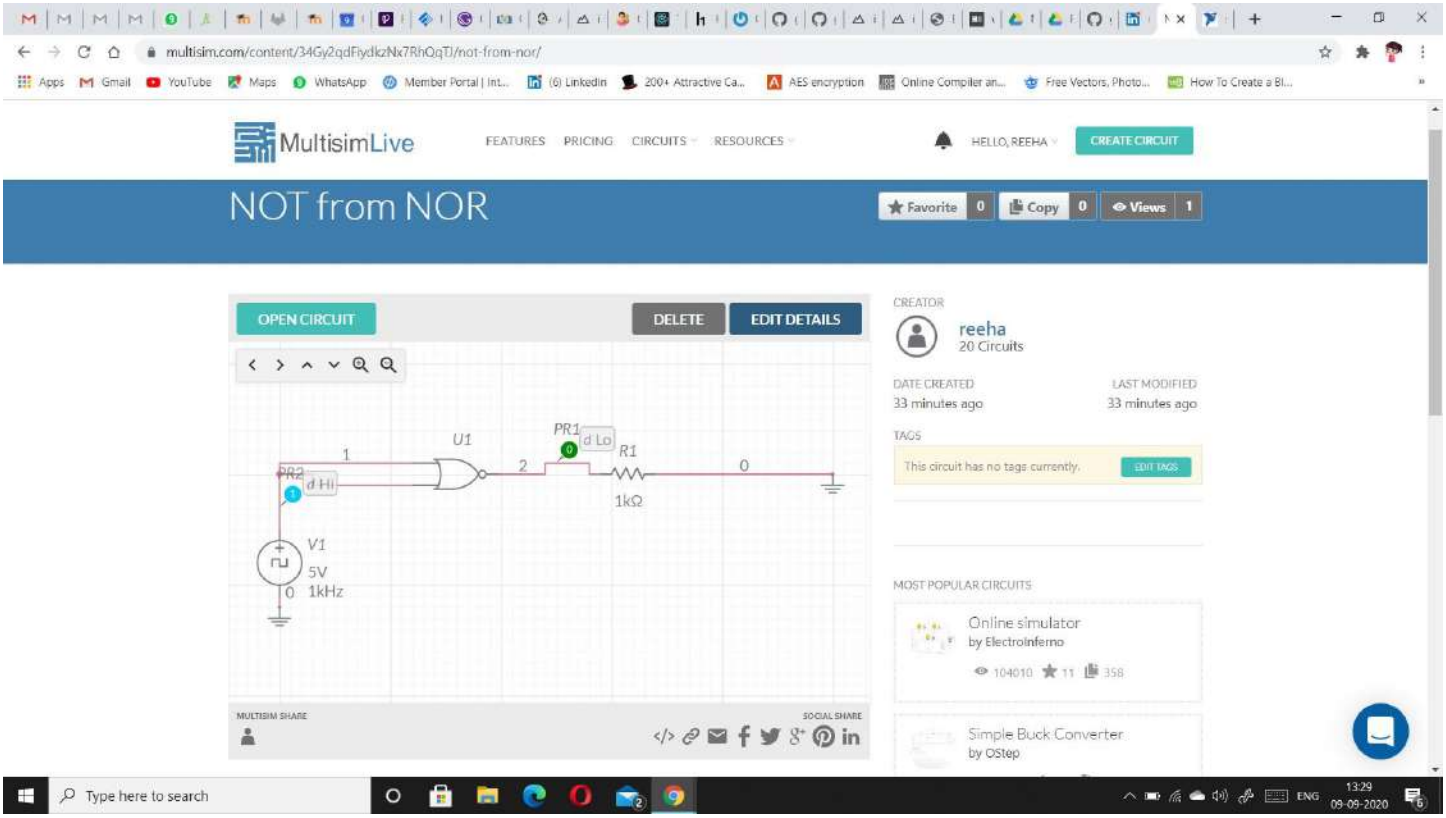
The NOR gate is a digital logic gate that implements logical NOR - it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results.

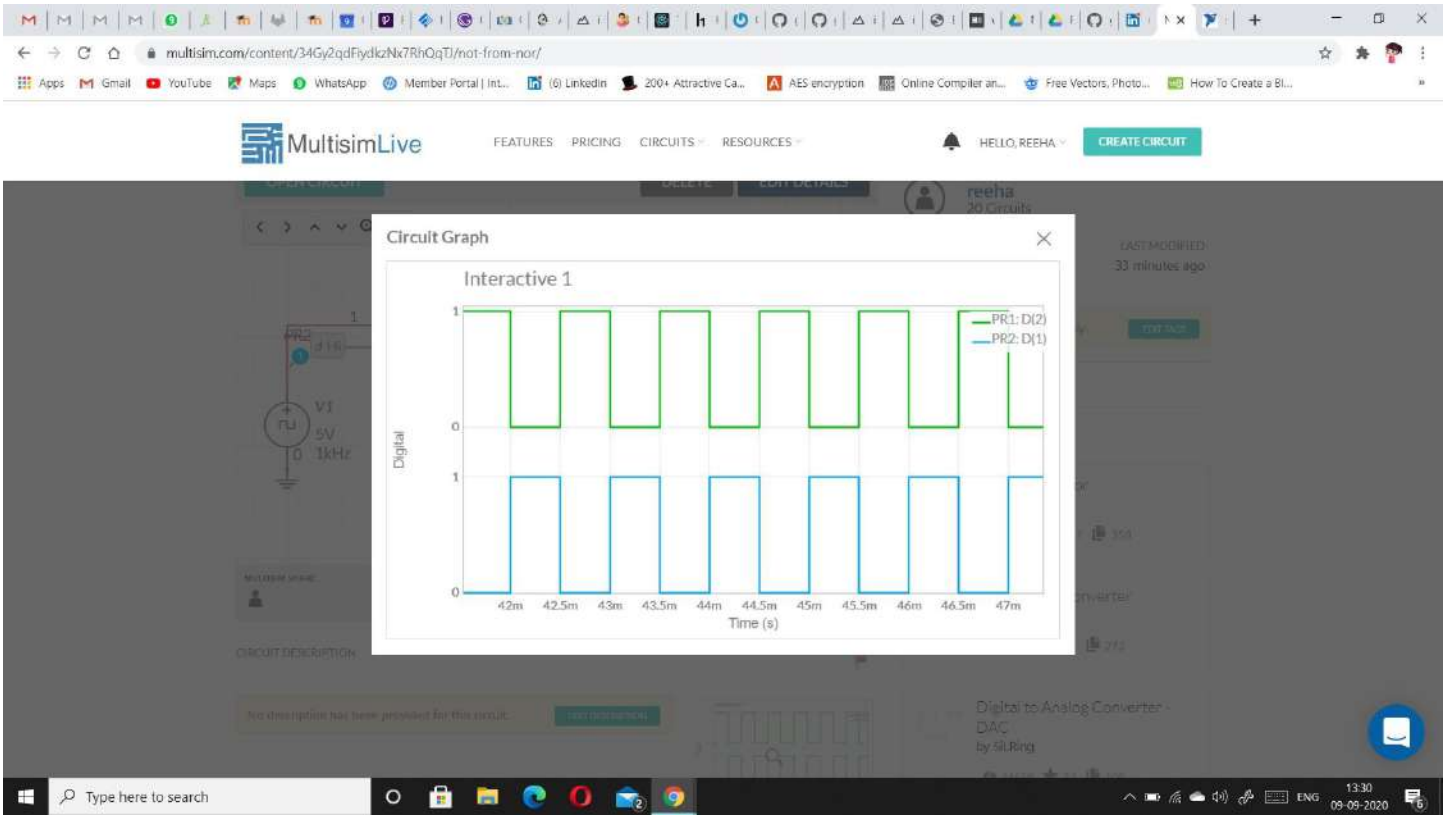
NOR is the result of the negation of the OR operator. It can also in some senses be seen as the inverse of an AND gate. NOR is a functionally complete operation—NOR gates can be combined to generate any other logical function. It shares this property with the NAND gate. By contrast, the OR operator is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

## Procedure followed on MULTISIM:

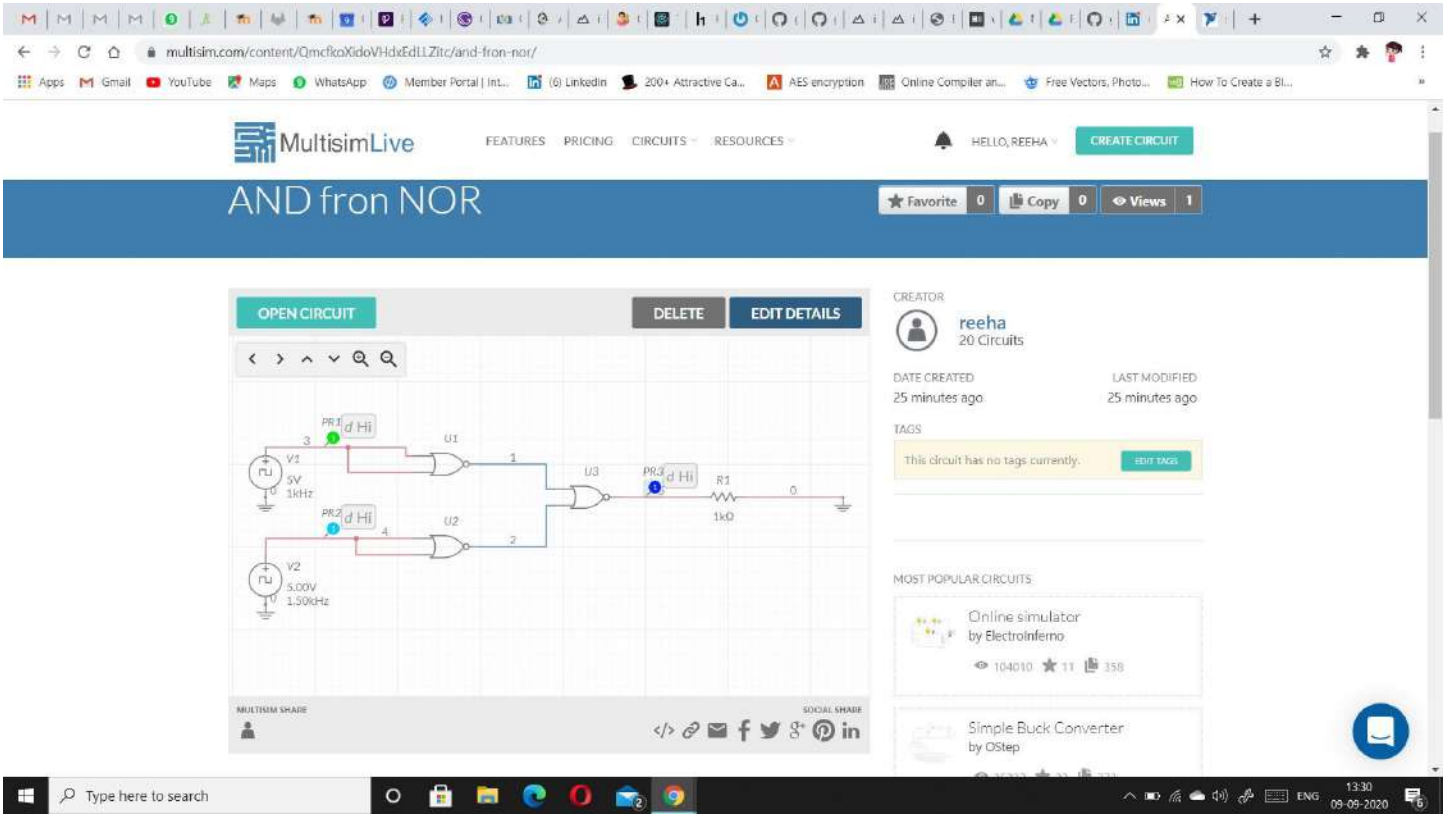
1. LOG IN ON www.multisim.com
2. CREATE THE CIRCUIT
3. SAVE THE CIRCUIT
4. SAVE THE SCREENSHOTS FOR
   i. INPUT & OUTPUT WAVEFORMS (ALONG WITH YOUR ID ON TOP LEFT)
   ii. CIRCUIT (ALONG WITH YOUR ID ON TOP LEFT)

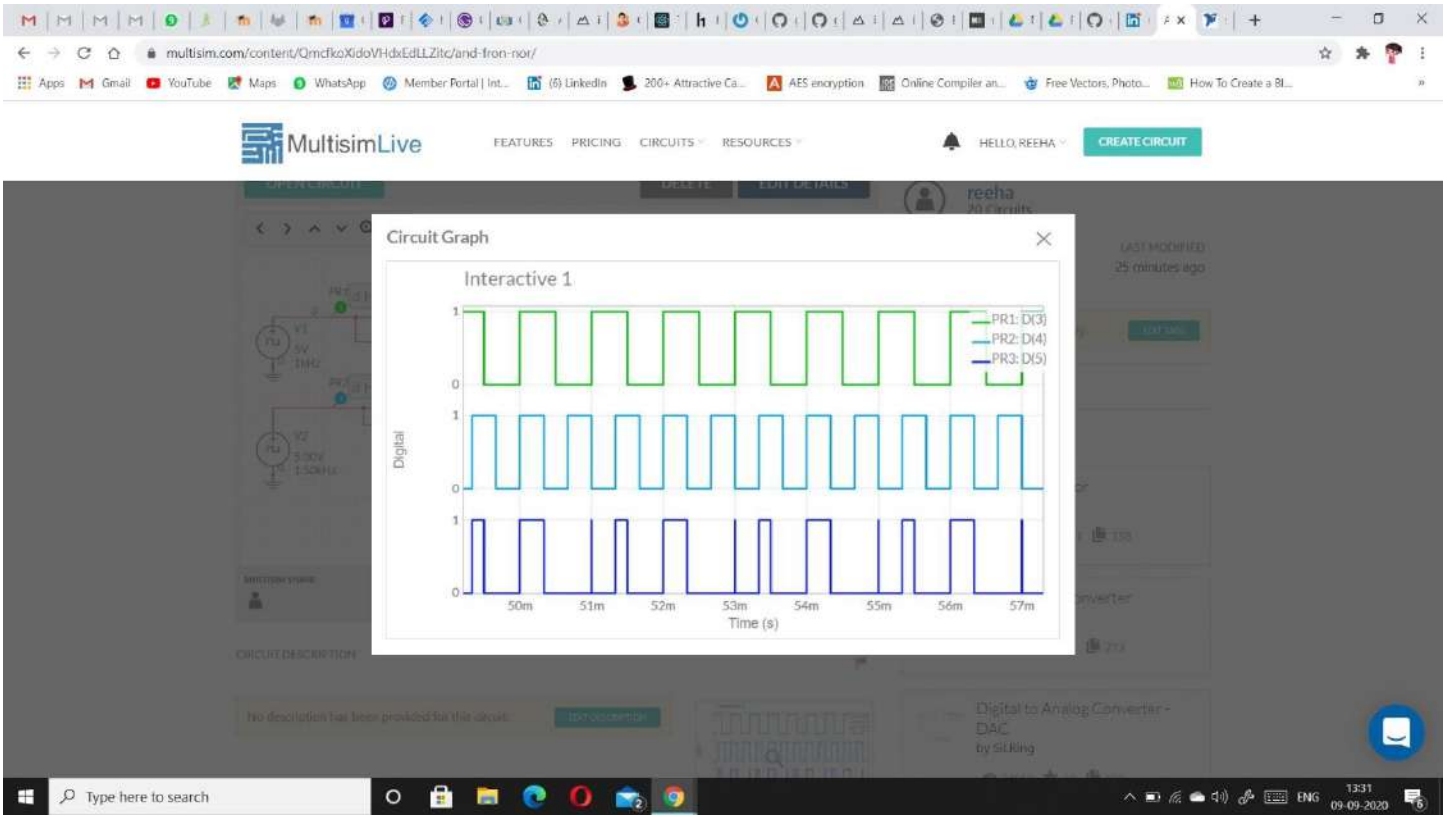
# Screenshot of circuit: NOT(Inverter) using NOR



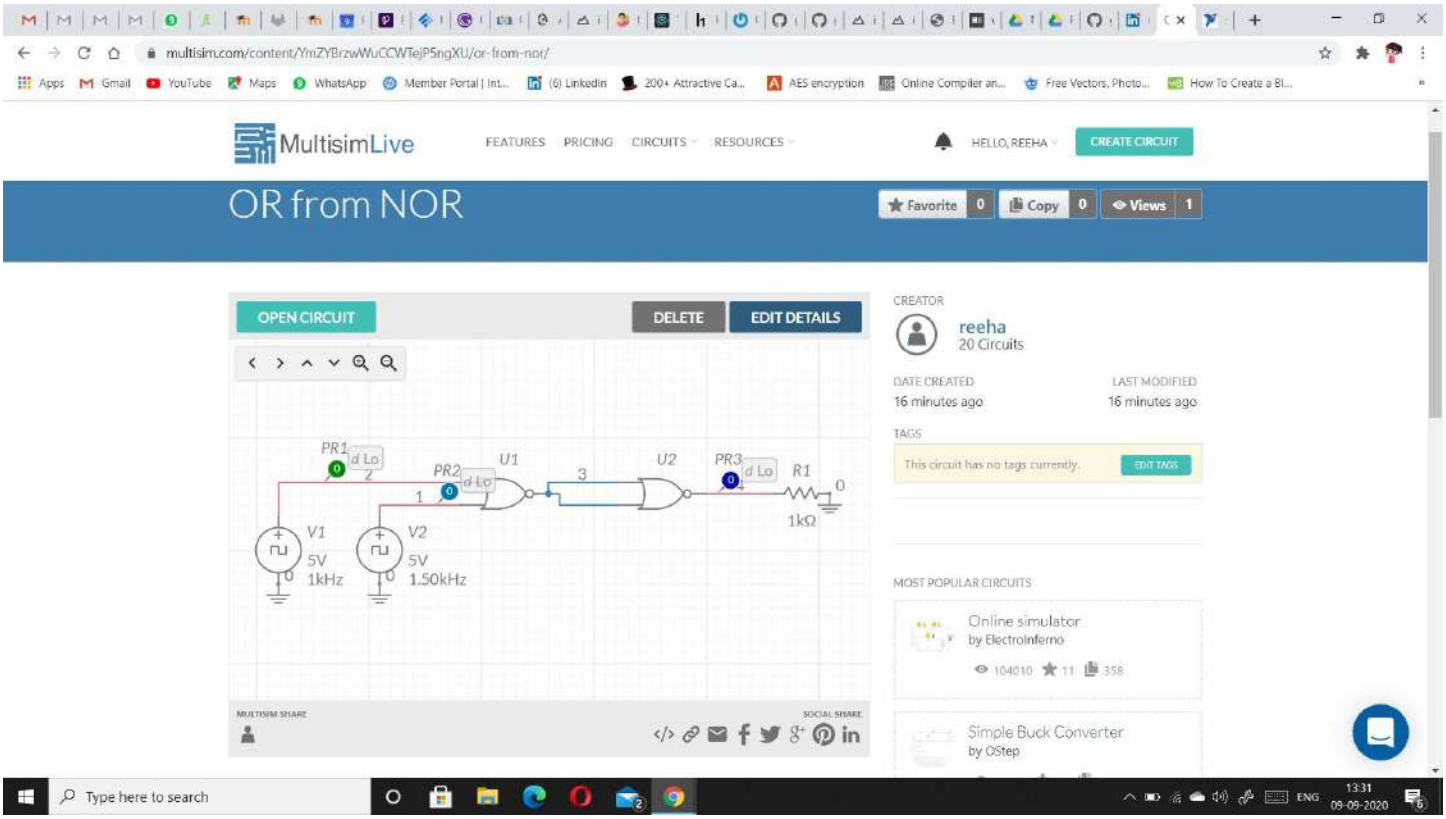



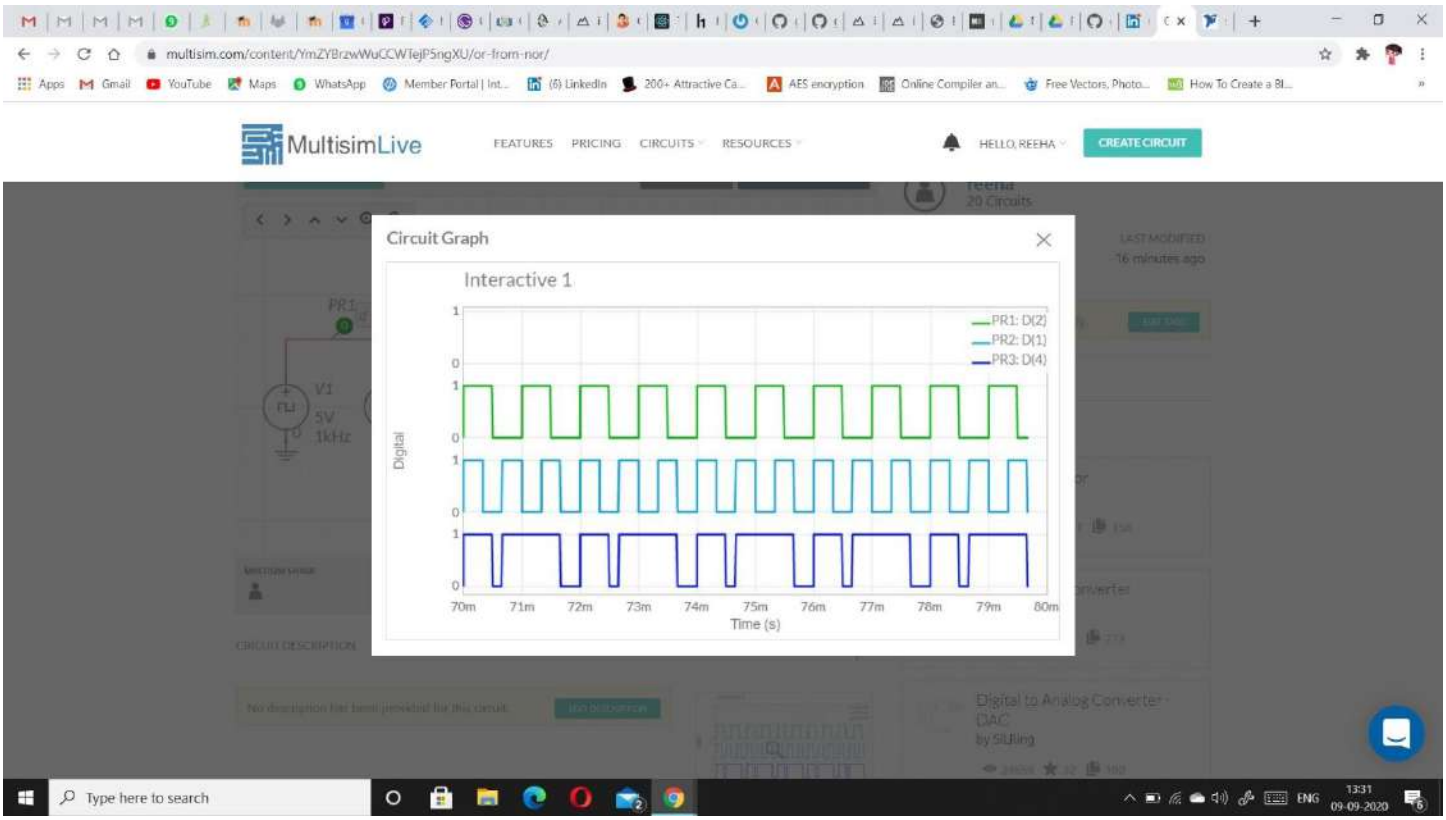
# Screenshot of circuit: AND using NOR



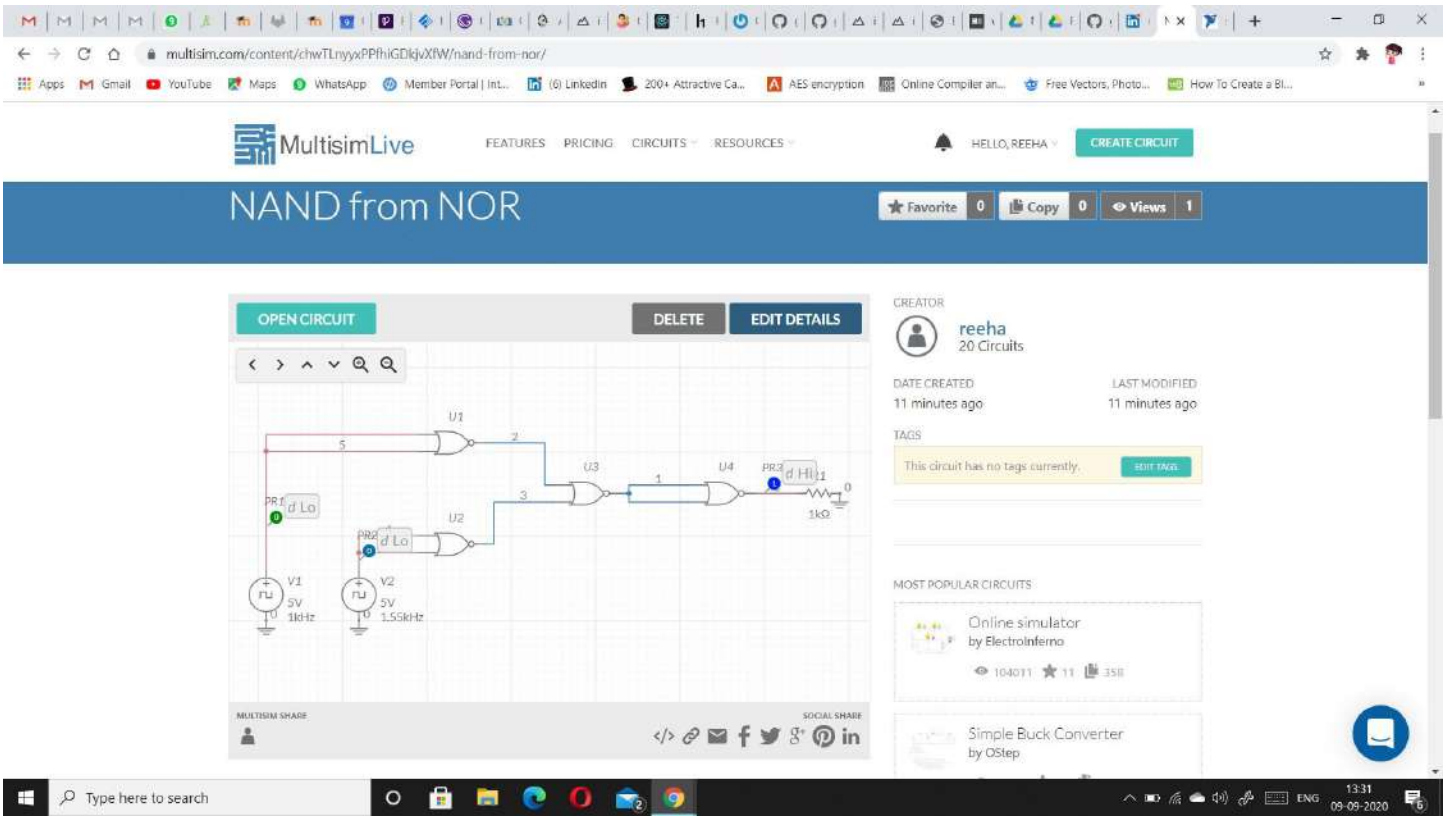



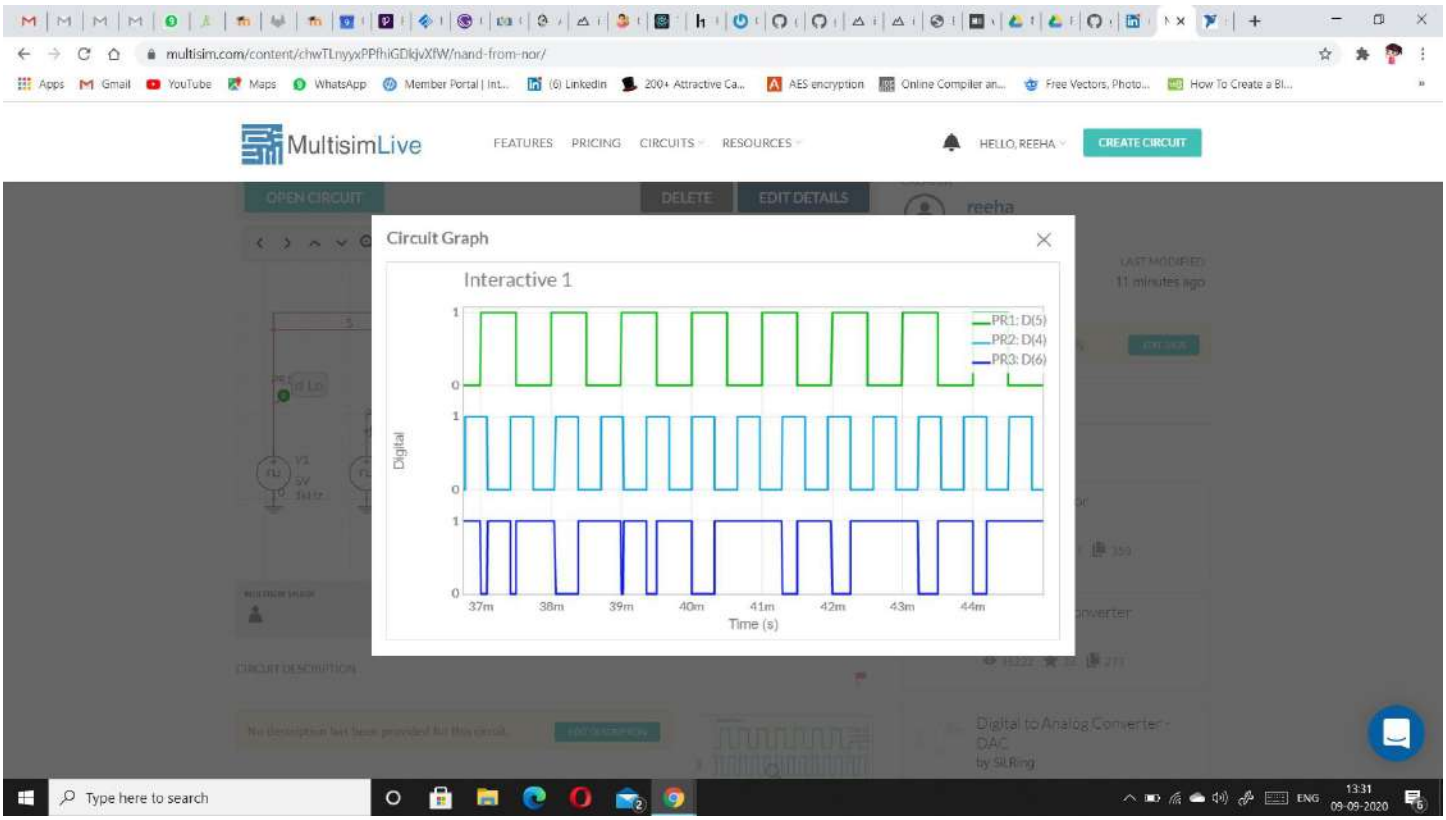
# Screenshot of circuit: OR using NOR



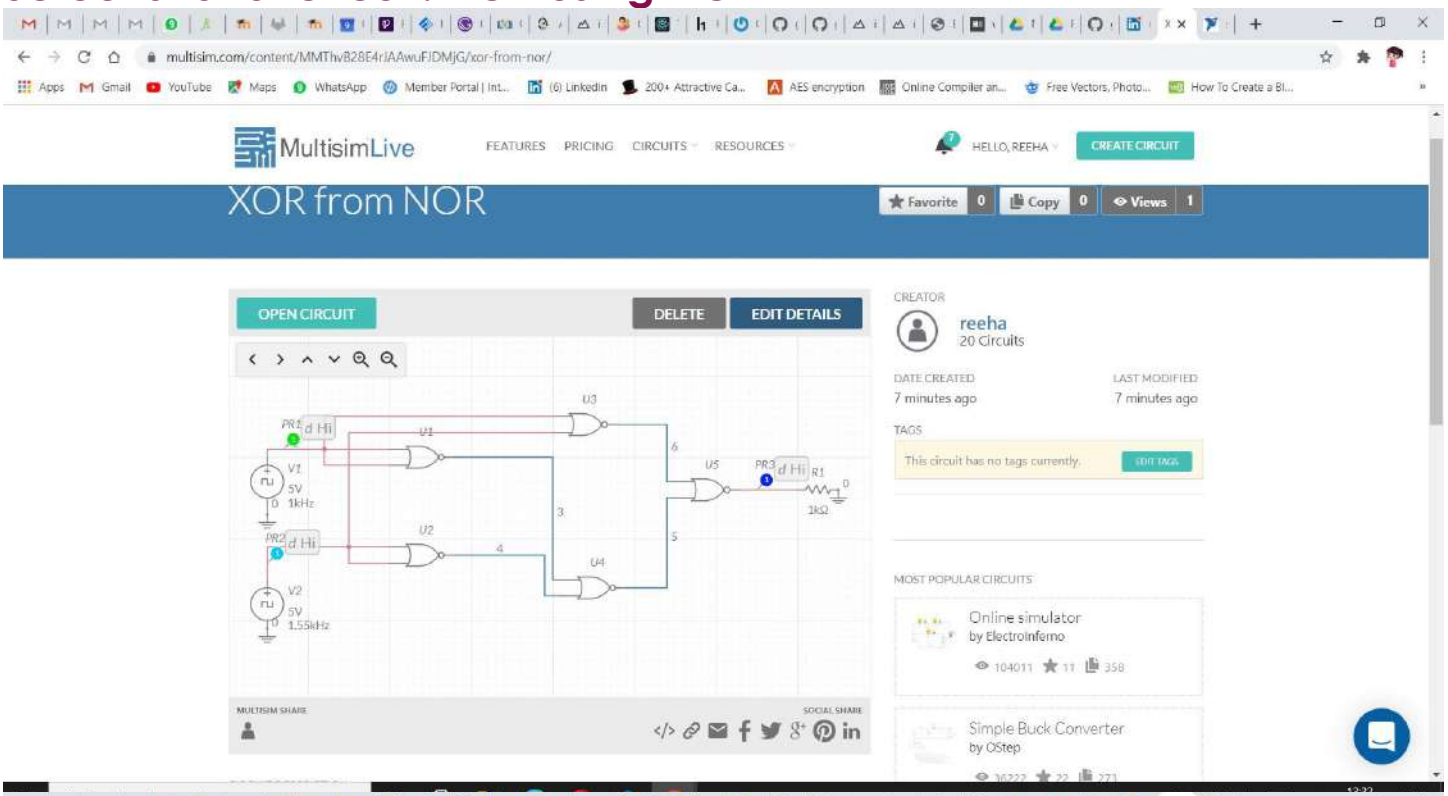



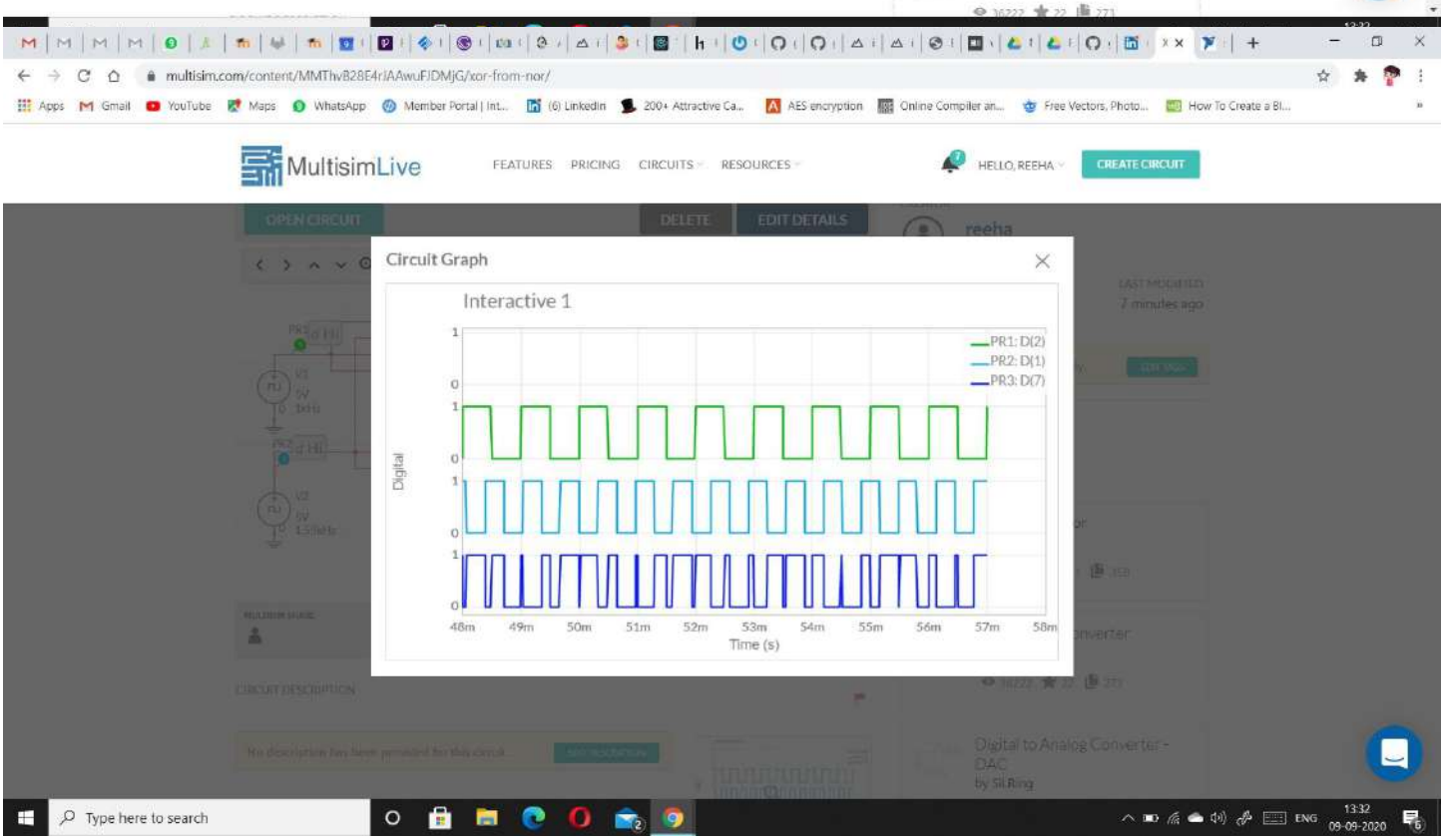
# Screenshot of circuit: NAND using NOR



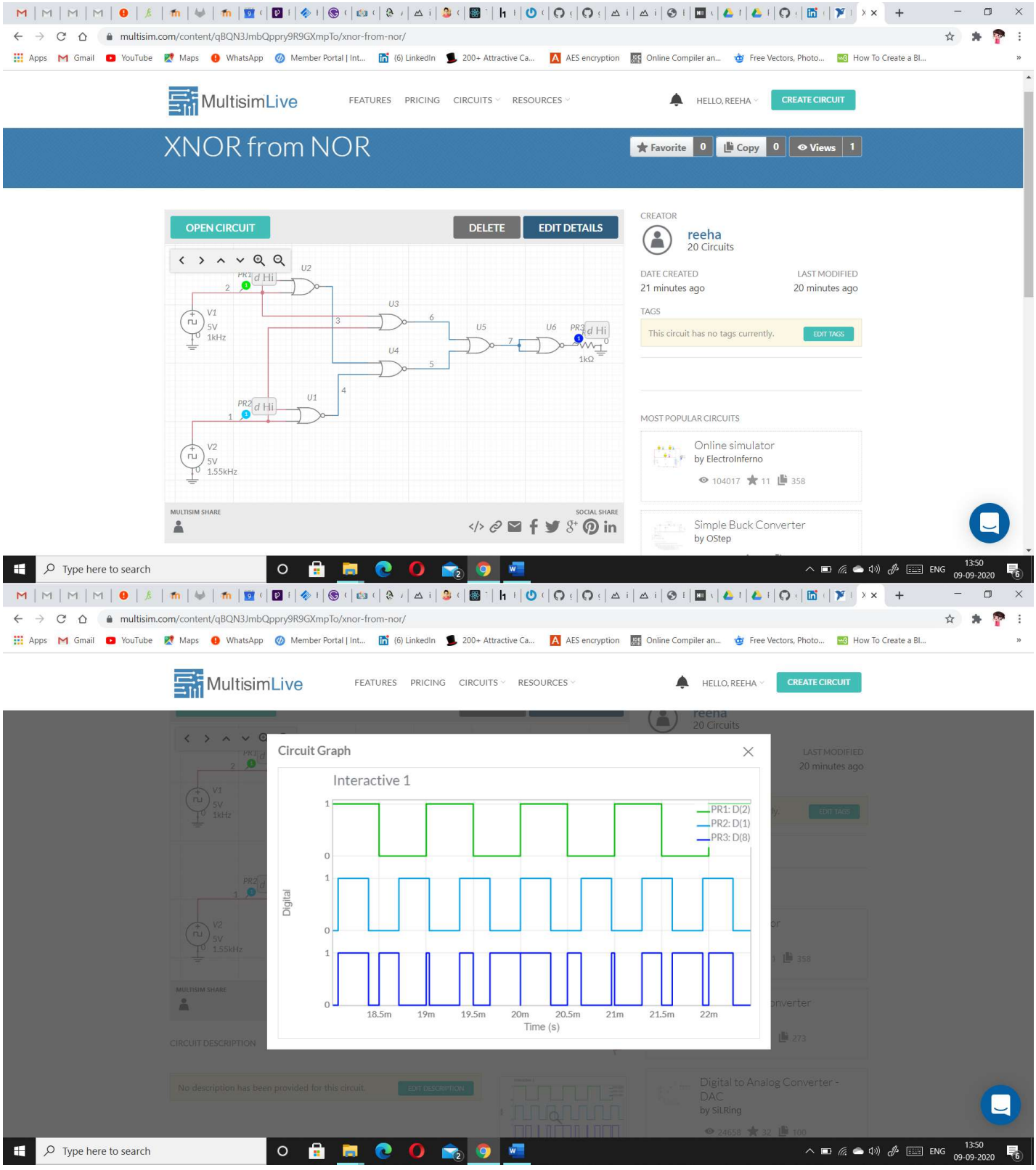



# Screenshot of circuit: XOR using NOR



# 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.

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

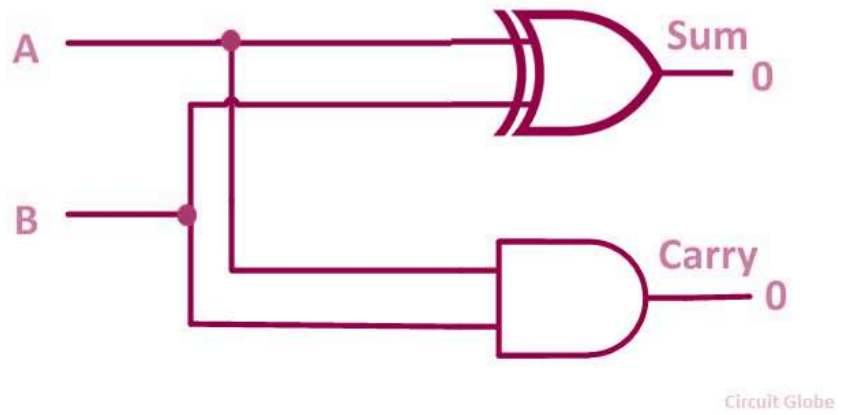
1 + 1 = 10

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 | CIN | COUT | S |
| **0** | 0 | 0 | 0 | 0 |
| **0** | 0 | 1 | 0 | 1 |

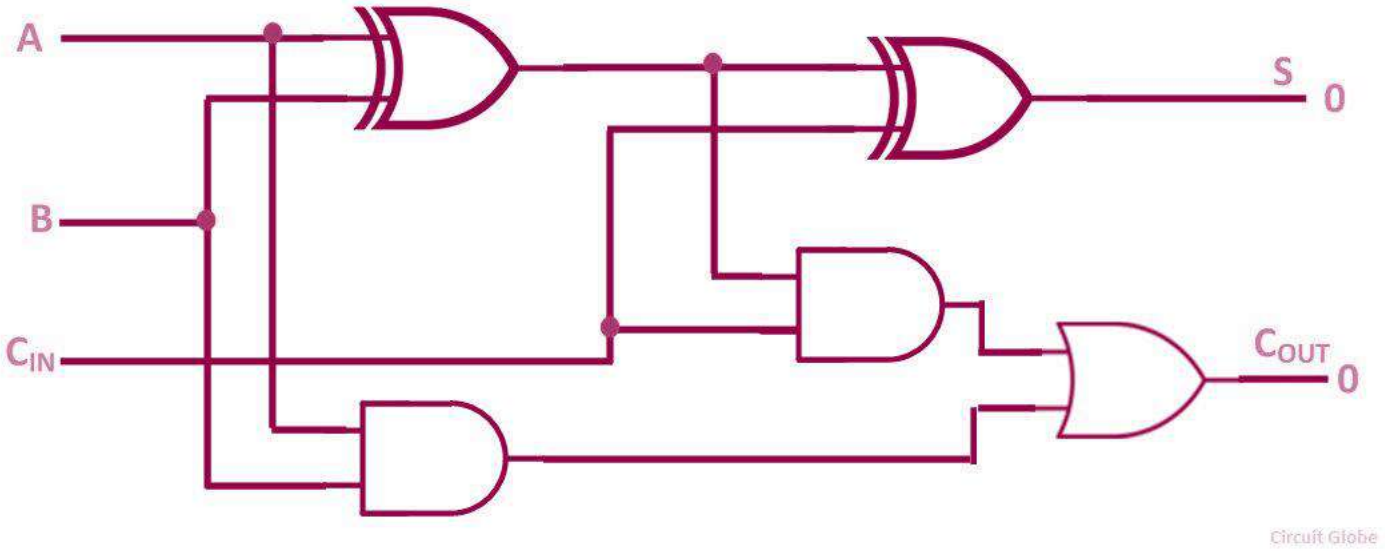
| | | | | |
|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 1 |
| **0** | 1 | 1 | 1 | 0 |
| **1** | 0 | 0 | 0 | 1 |
| **1** | 0 | 1 | 1 | 0 |
| **1** | 1 | 0 | 1 | 0 |
| **1** | 1 | 1 | 1 | 1 |

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.

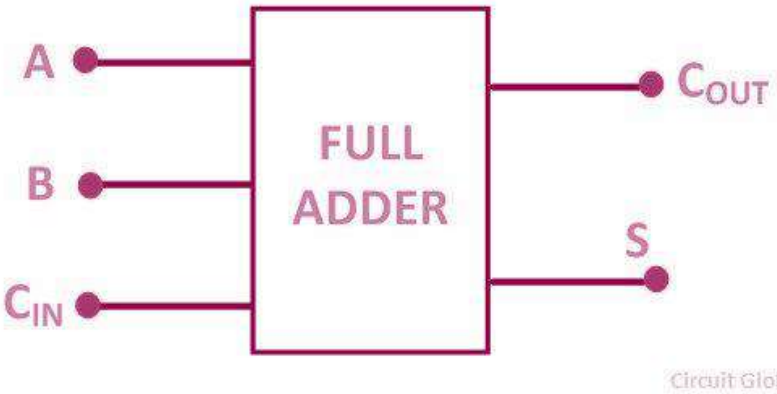
Thus, a full adder circuit can be implemented with the help of two half adder circuits. The first half adder circuit will be used to add A and B to produce a partial sum. The second half adder logic can be used to add $C_{IN}$ to the sum produced by the first half adder circuit. Finally, the output S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

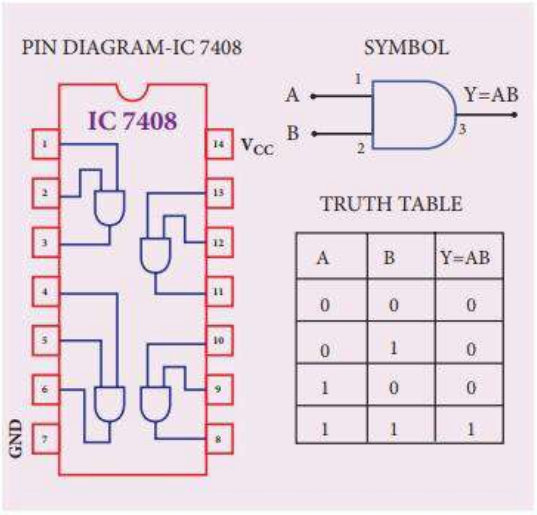



The schematic representation of a single bit Full Adder is shown below.



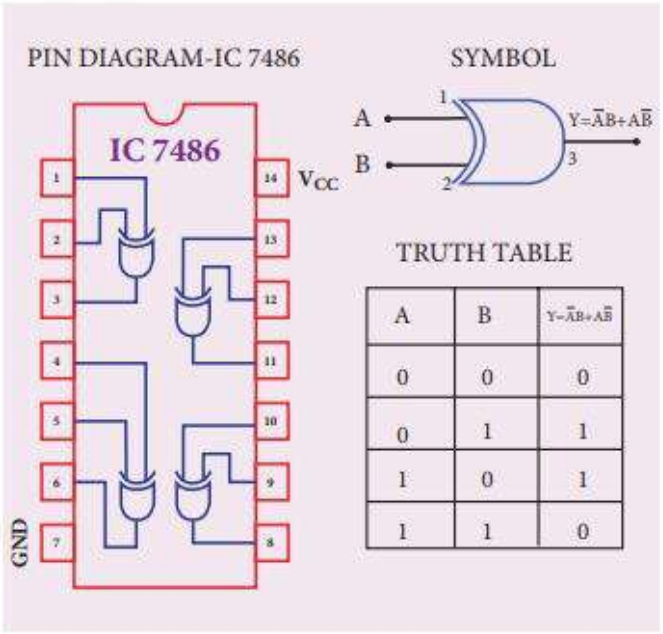
Circuit Symbol, pin diagram and Truth table of gates used:

**AND Gate:**



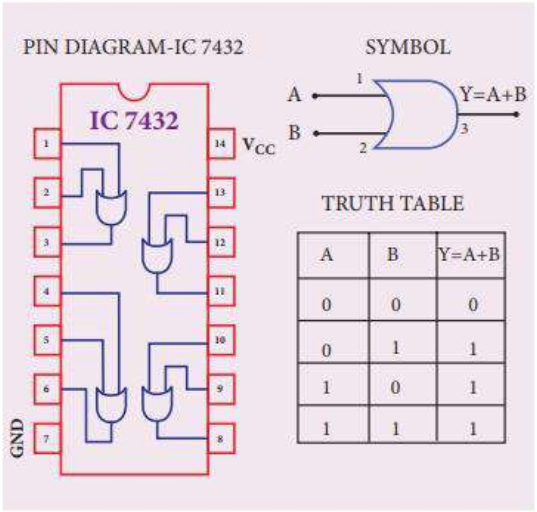

| A | B | Y=AB |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**X-OR Gate :**




| A | B | Y=$\bar{A}$B+A$\bar{B}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**OR Gate:**



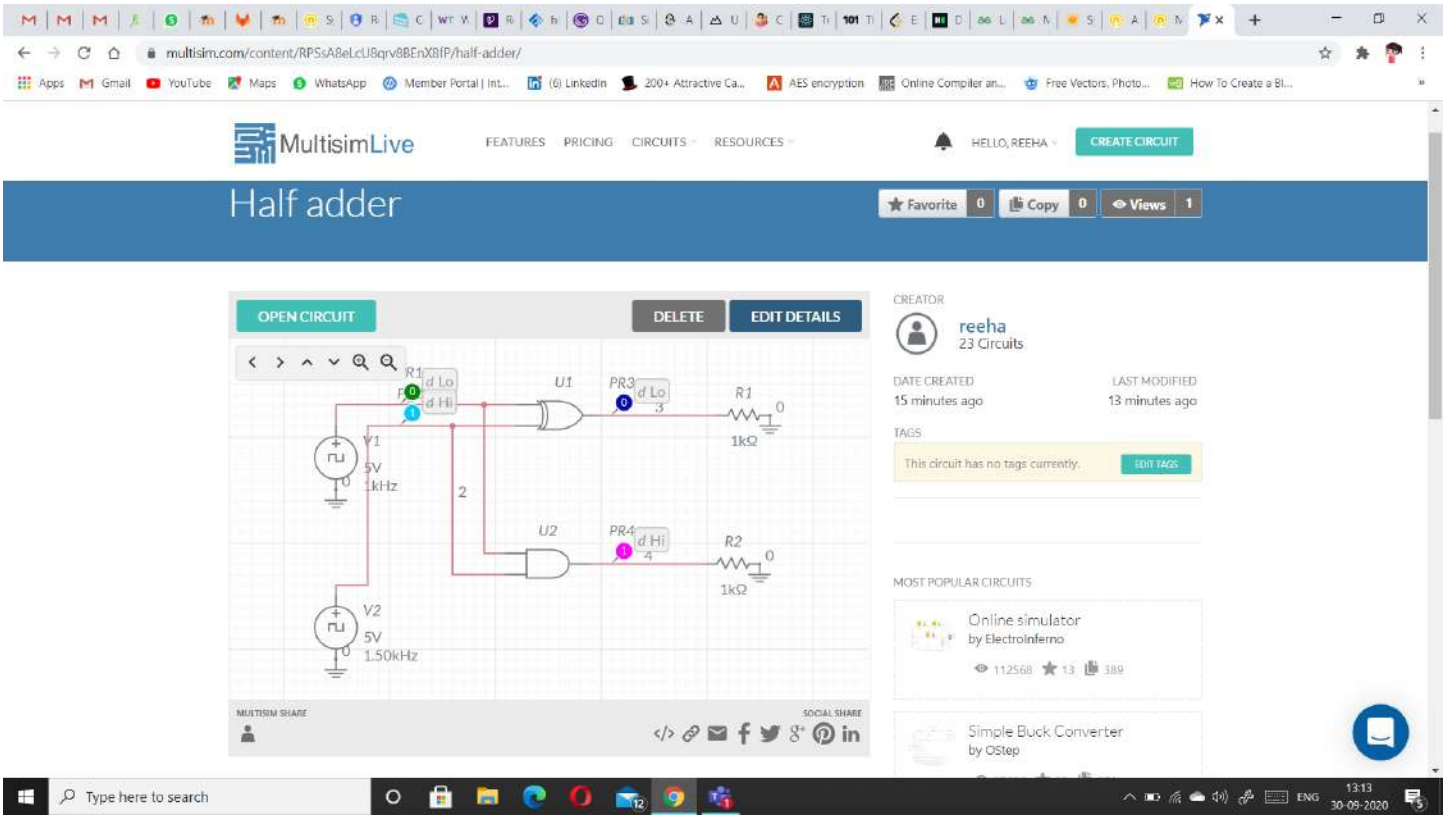

| A | B | Y=A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

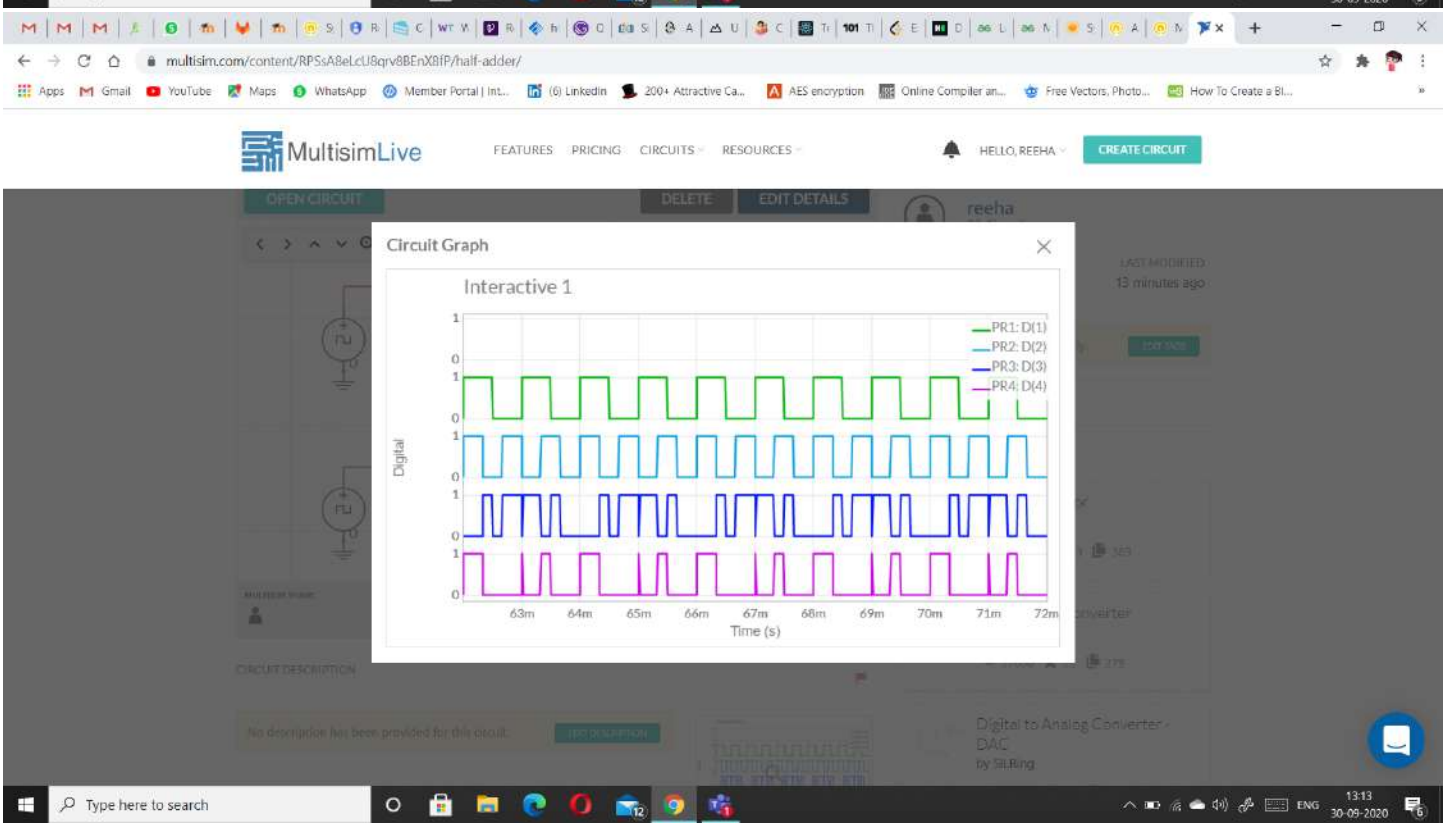
**Procedure followed on MULTISIM:**

1. LOG IN ON 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)

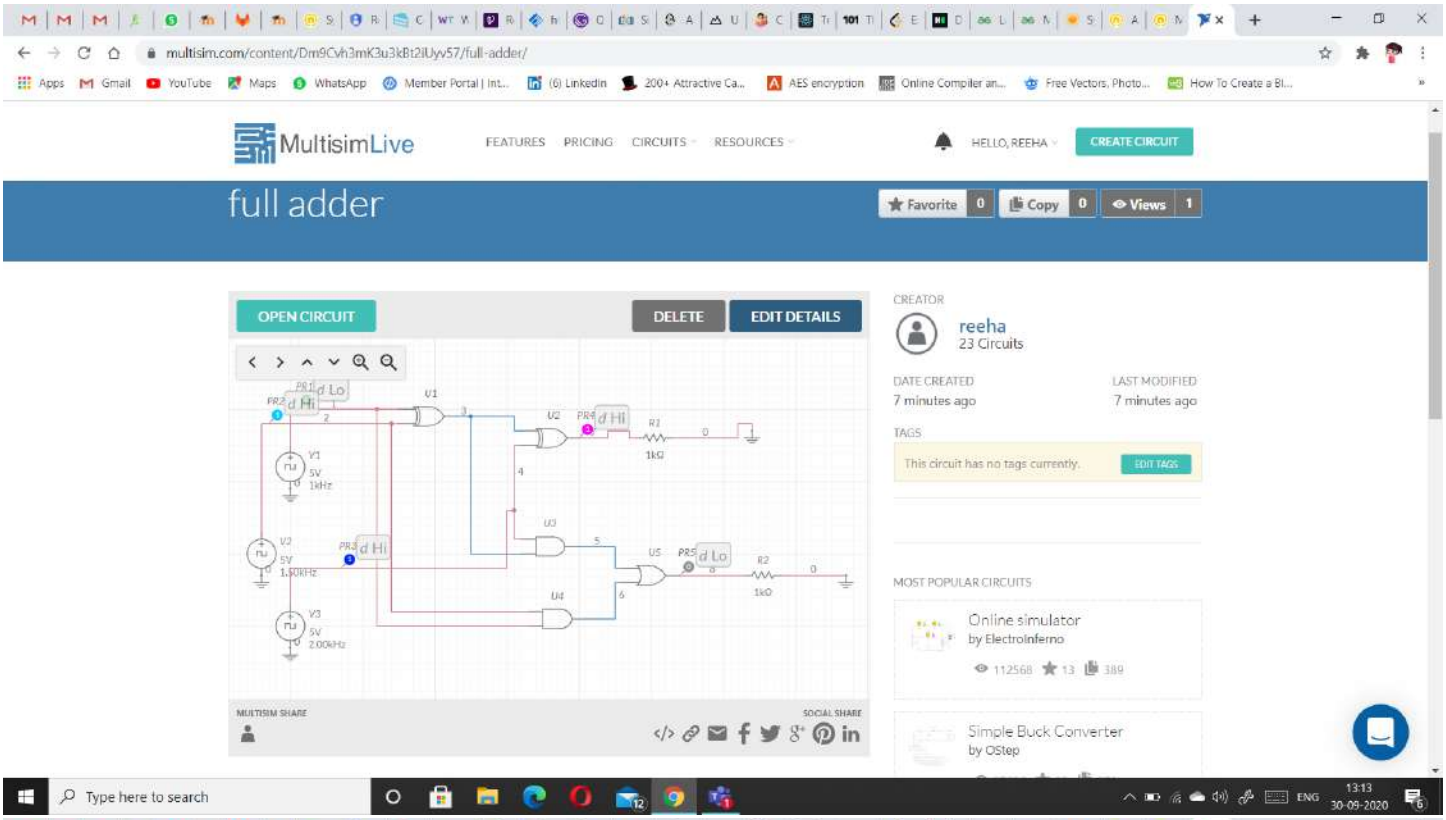
# Circuits and Output waveform

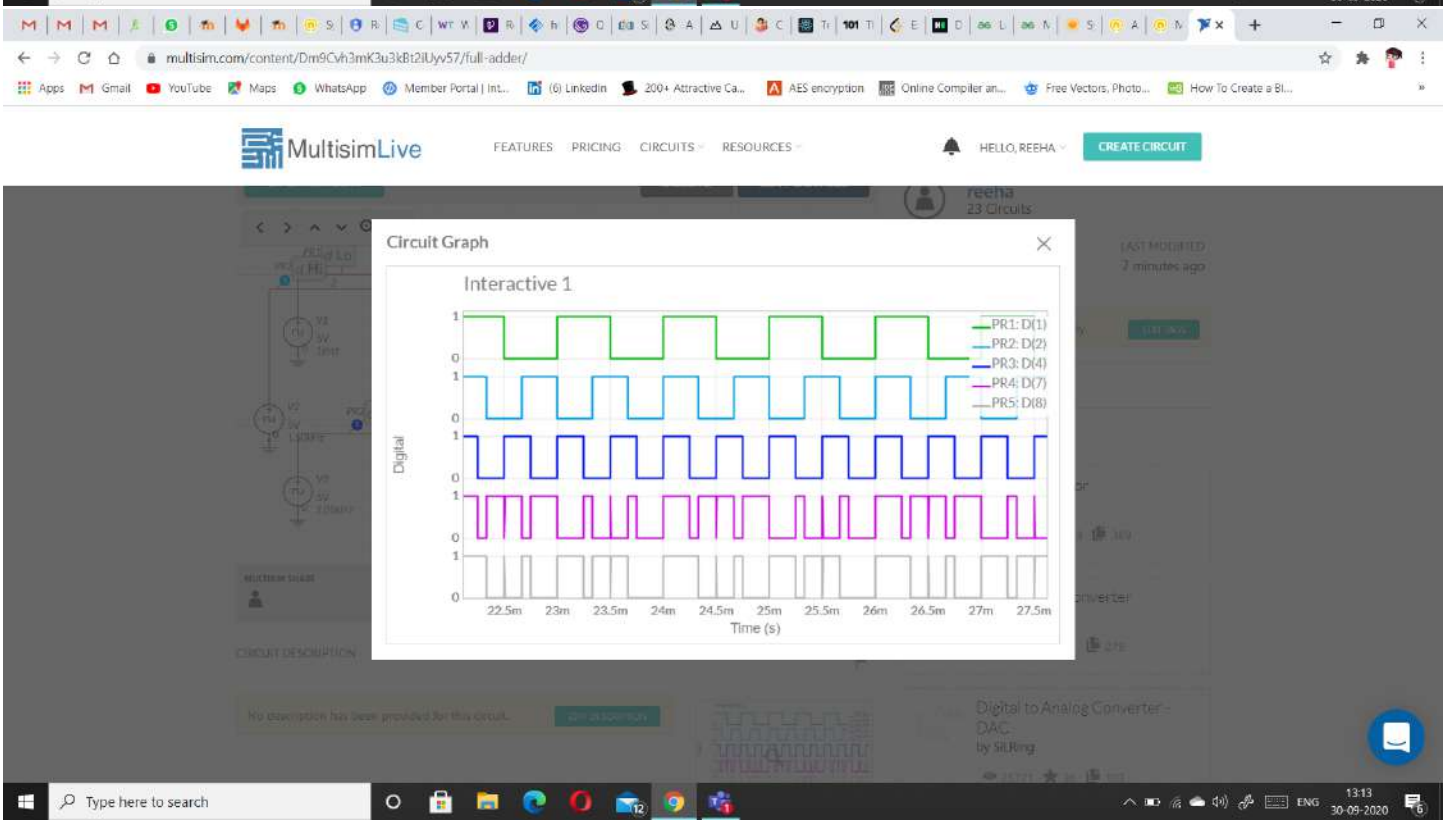
# Half Adder


multisim.com/content/RPSsA8eLcU8qrv8BEnX8fP/half-adder/
MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Half adder
Favorite 0
Copy 0
Views 1
OPEN CIRCUIT
DELETE
EDIT DETAILS
U1
U2
PR3
PR4
R1
R2
1kΩ
Y1
5V
1kHz
V2
5V
1.50kHz
CREATOR
reeha
23 Circuits
DATE CREATED
15 minutes ago
LAST MODIFIED
13 minutes ago
TAGS
This circuit has no tags currently.
EDIT TAGS
MOST POPULAR CIRCUITS
Online simulator
by ElectroInferno
Simple Buck Converter
by OStep
MULTISIM SHARE
SOCIAL SHARE
Type here to search
ENG
13:13
30-09-2020


multisim.com/content/RPSsA8eLcU8qrv8BEnX8fP/half-adder/
MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(1)
PR2: D(2)
PR3: D(3)
PR4: D(4)
Digital
63m
64m
65m
66m
67m
68m
69m
70m
71m
72m
Time (s)
Digital to Analog Converter - DAC
by Sil.Ring
Type here to search
ENG
13:13
30-09-2020

# Full Adder


multisim.com/content/Dm9Cvh3mK3u3kBt2iUyv57/full-adder/
MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
full adder
Favorite 0
Copy 0
Views 1
OPEN CIRCUIT
DELETE
EDIT DETAILS
CREATOR
reeha
23 Circuits
DATE CREATED
7 minutes ago
LAST MODIFIED
7 minutes ago
TAGS
This circuit has no tags currently.
EDIT TAGS
MOST POPULAR CIRCUITS
Online simulator
by ElectroInferno
Simple Buck Converter
by OStep
MULTISIM SHARE
SOCIAL SHARE
Type here to search
ENG
13:13
30-09-2020


multisim.com/content/Dm9Cvh3mK3u3kBt2iUyv57/full-adder/
MultisimLive
FEATURES
PRICING
CIRCUITS
RESOURCES
HELLO, REEHA
CREATE CIRCUIT
Circuit Graph
Interactive 1
PR1: D(1)
PR2: D(2)
PR3: D(4)
PR4: D(7)
PR5: D(8)
Digital
Time (s)
22.5m
23m
23.5m
24m
24.5m
25m
25.5m
26m
26.5m
27m
27.5m
Digital to Analog Converter - DAC
by SILRing
Type here to search
ENG
13:13
30-09-2020

# 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 Explain 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 expresses 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.

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.

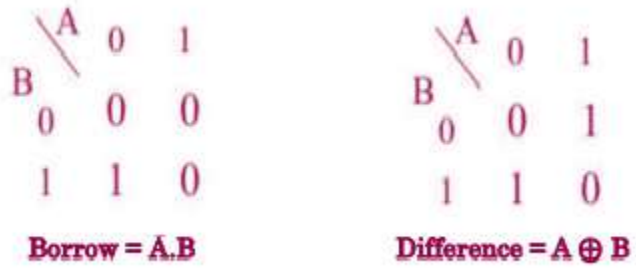
| First Digit | Second Digit | Difference | Borrow |
|---|---|---|---|
| **0** | 0 | 0 | 0 |
| **1** | 0 | 1 | 0 |
| **0** | 1 | 1 | 1 |
| **1** | 1 | 0 | 0 |

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.




**half subtractor k map**

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

Difference **(D) = (**x'y **+** xy**')**

= **x ⊕ y**

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.

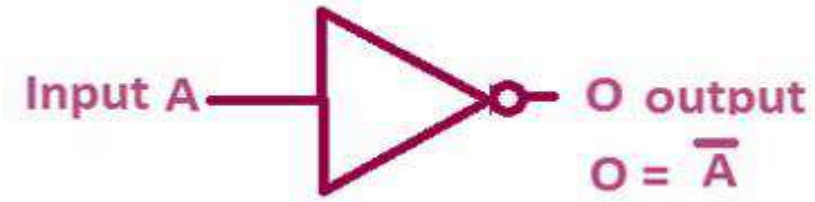

Symbol

| 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.




Symbol

| 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.


Symbol

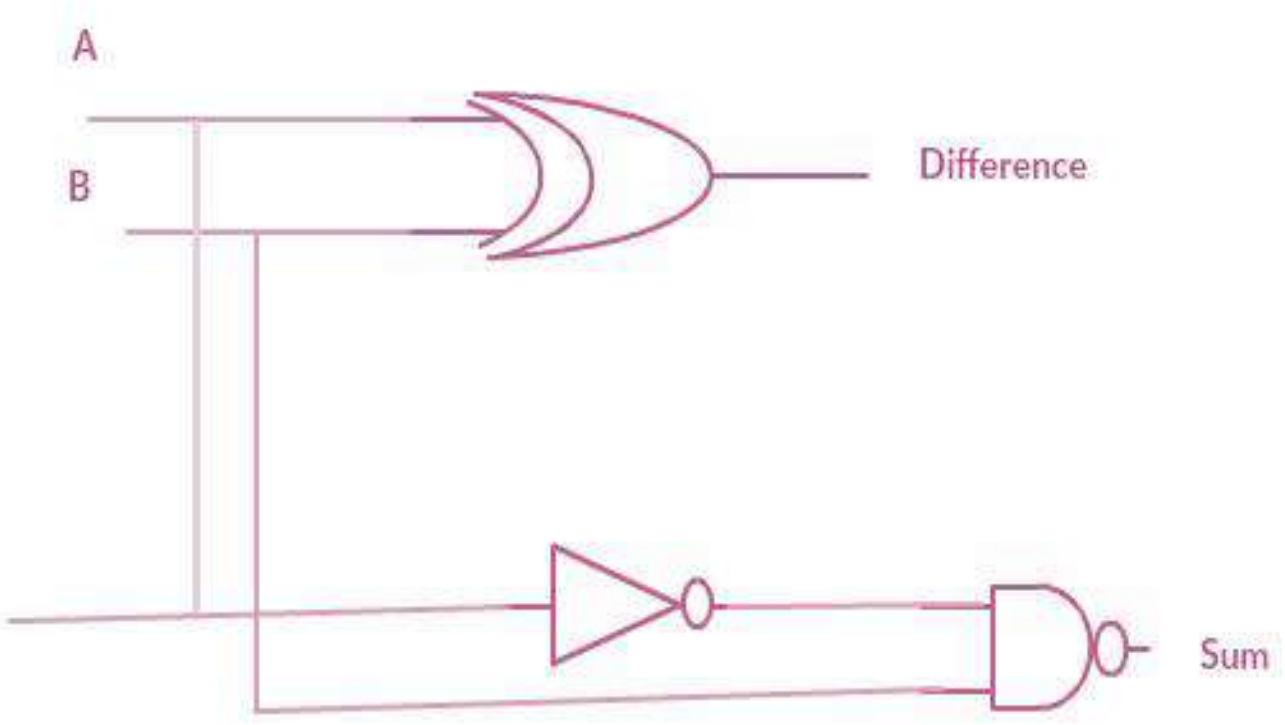
| Inputs | | Output |
|---|---|---|
| A | B | O |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

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

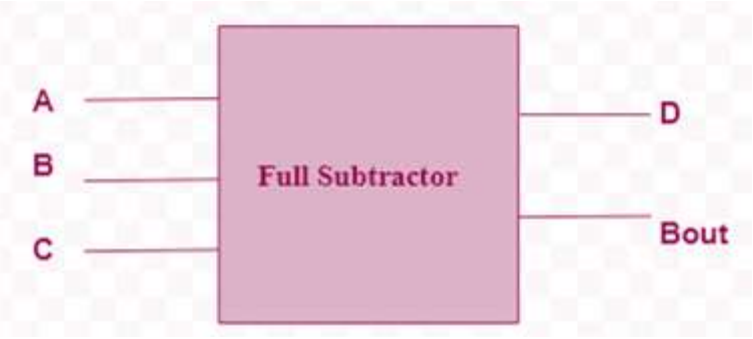
If we monitor cautiously, it is fairly clear that the variety of operation executed by this circuit which is accurately related to the EX-OR gate operation. Therefore, we can simply use the EX-OR gate for making difference. In the same way, the borrow produced by half adder circuit can be simply attained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

**Truth Table**

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

## Full Subtractor

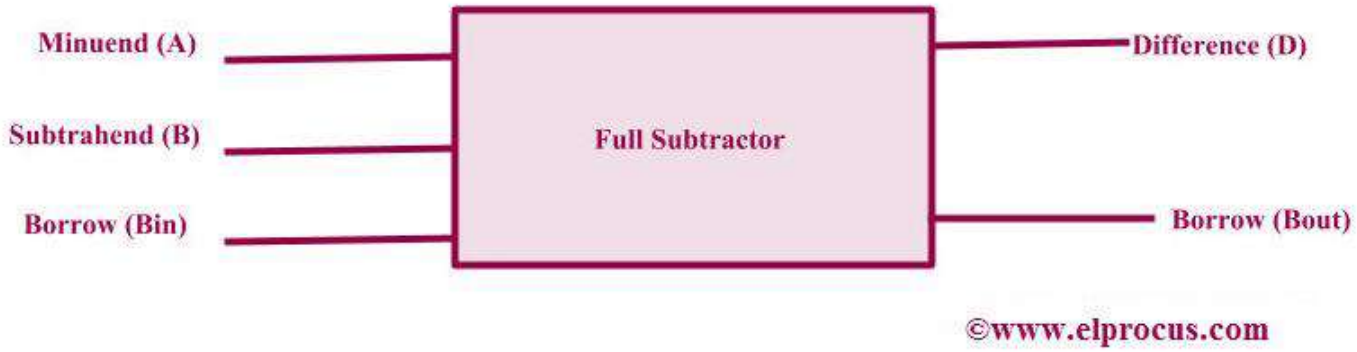
Full subtractor is an electronic device or logic circuit which performs subtraction of two binary digits. It is a combinational logic circuit used in digital electronics. Many combinational circuits are available in integrated circuit technology namely adders, encoders, decoders and multiplexers. In 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/ps. 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.




### Full Subtractor Circuit Diagram with Logic Gates

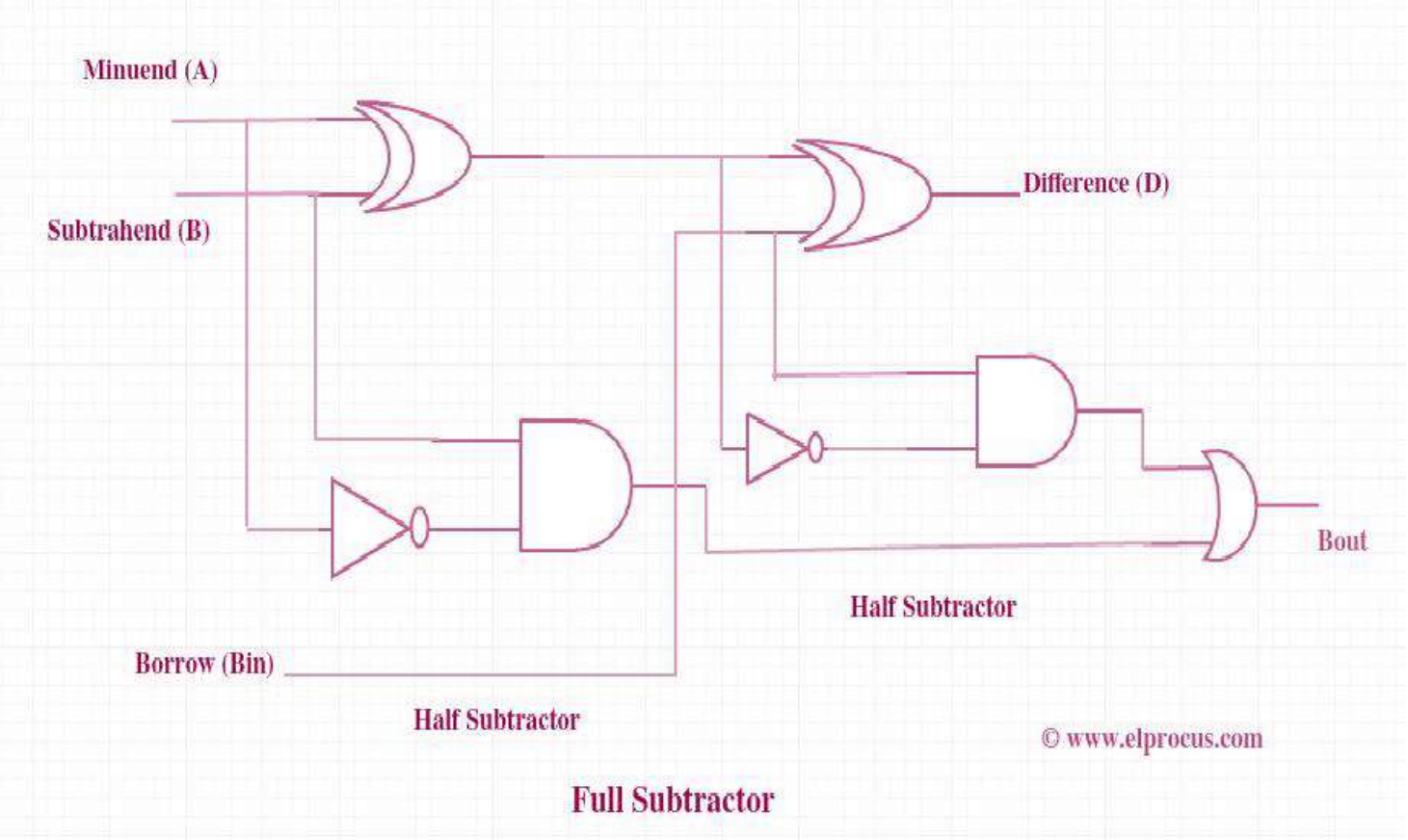
The circuit diagram of full subtractor using basic gates is shown in the following block diagram. This circuit can be done with two half-Subtractor circuits.

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 using Logic Gates

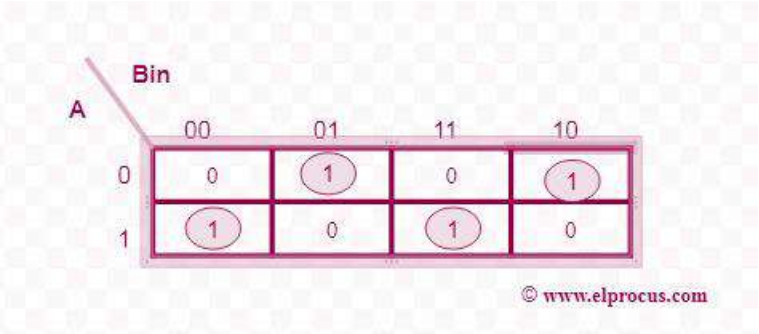
**Full Subtractor Truth Table**

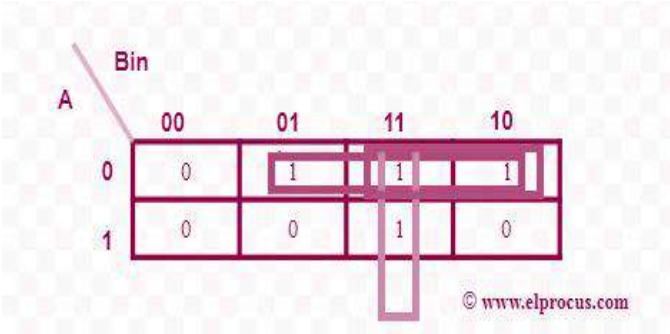
This subtractor circuit executes a subtraction between two bits, which has 3- inputs (A, B and Bin) and two outputs (D and Bout). Here the inputs indicate minuend, subtrahend, & previous borrow, whereas the two outputs are denoted as borrow o/p and difference. The following image shows the truth table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




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

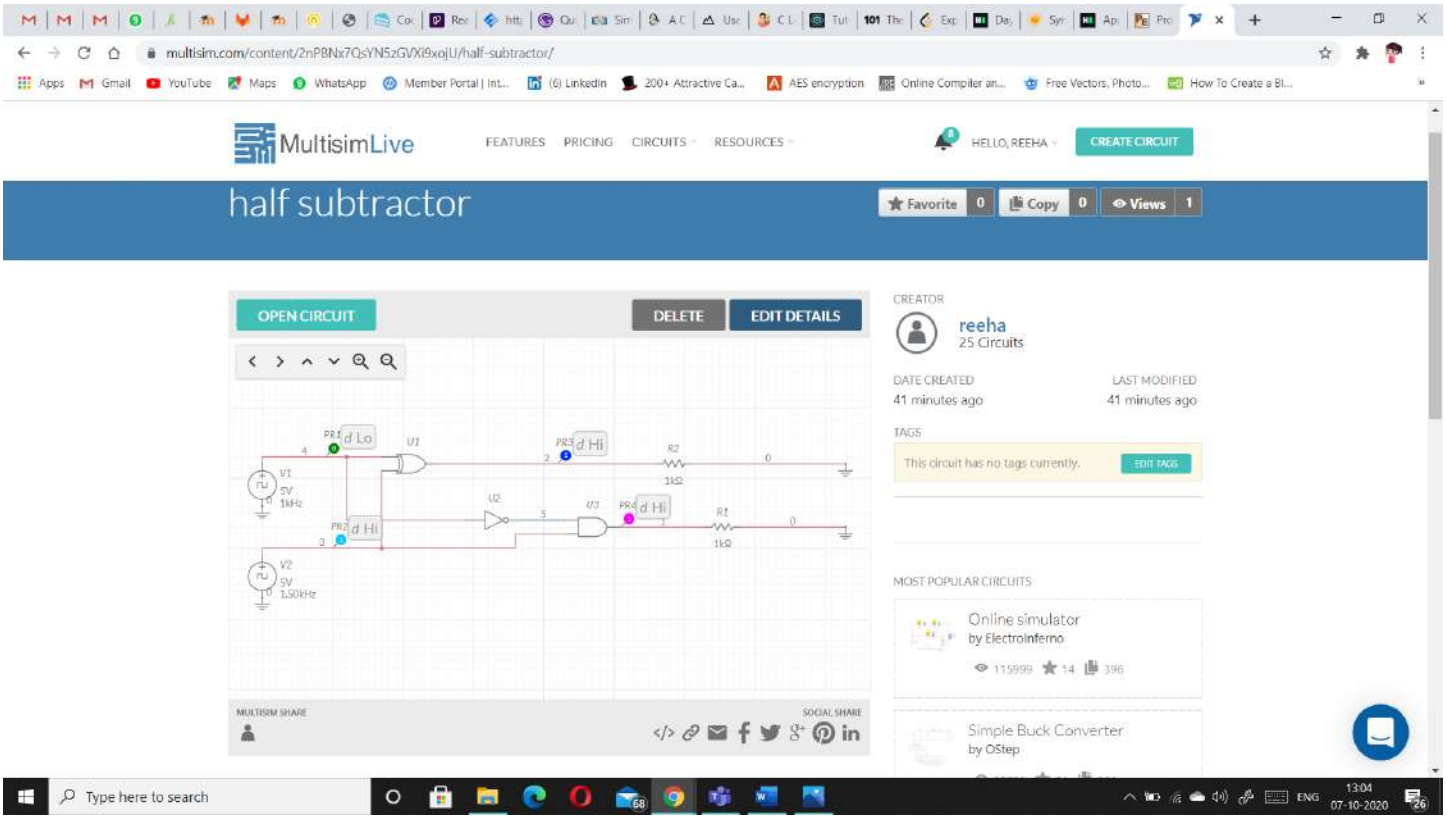
- These are generally employed for ALU (Arithmetic logic unit) in computers to subtract as CPU & GPU for the applications of graphics to decrease the circuit difficulty.
- Subtractors are mostly used for performing arithmetical functions like subtraction, in electronic calculators as well as digital 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.

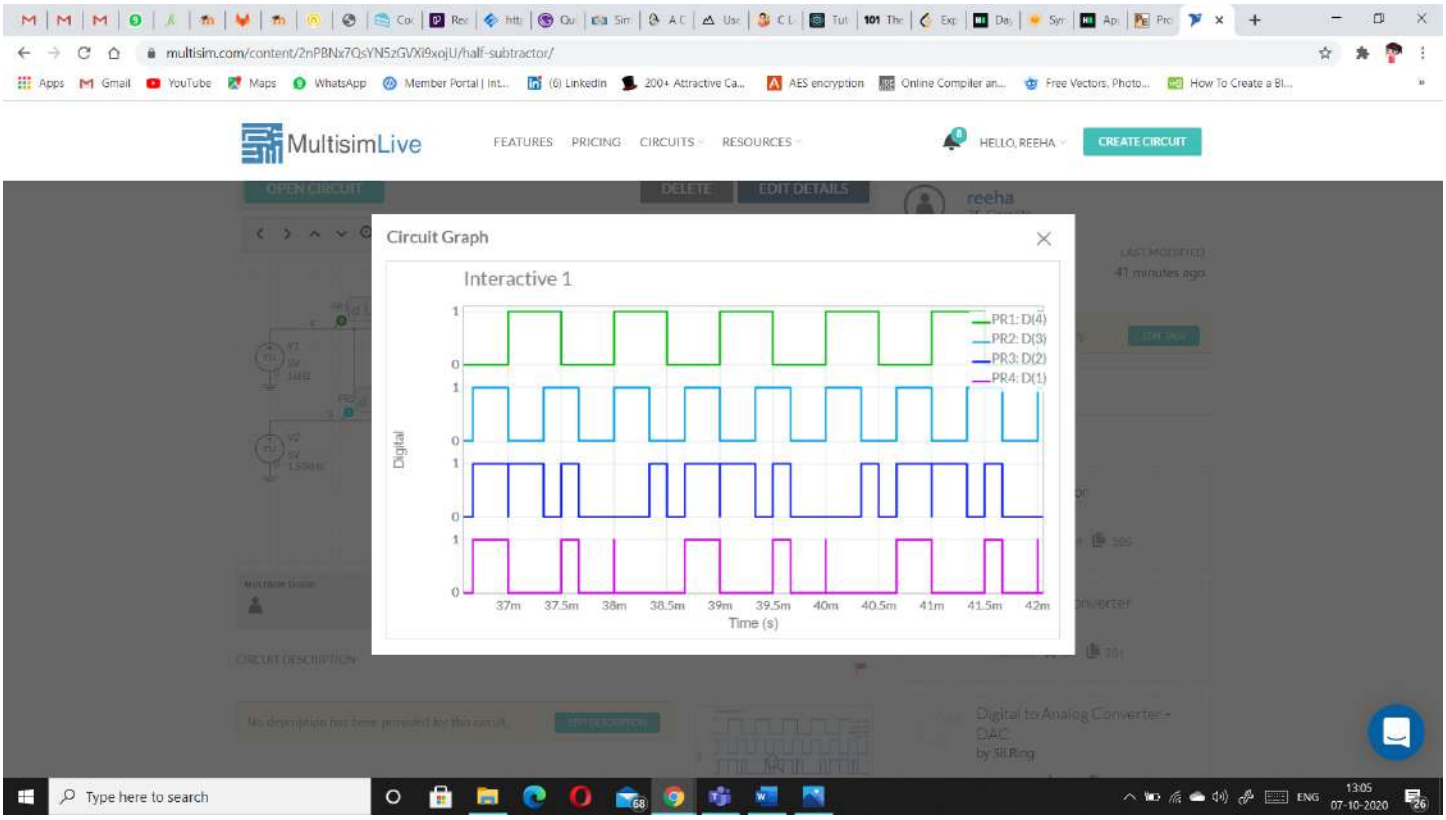
## Procedure followed on MULTISIM:

1. LOG IN ON www.multisim.com
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)

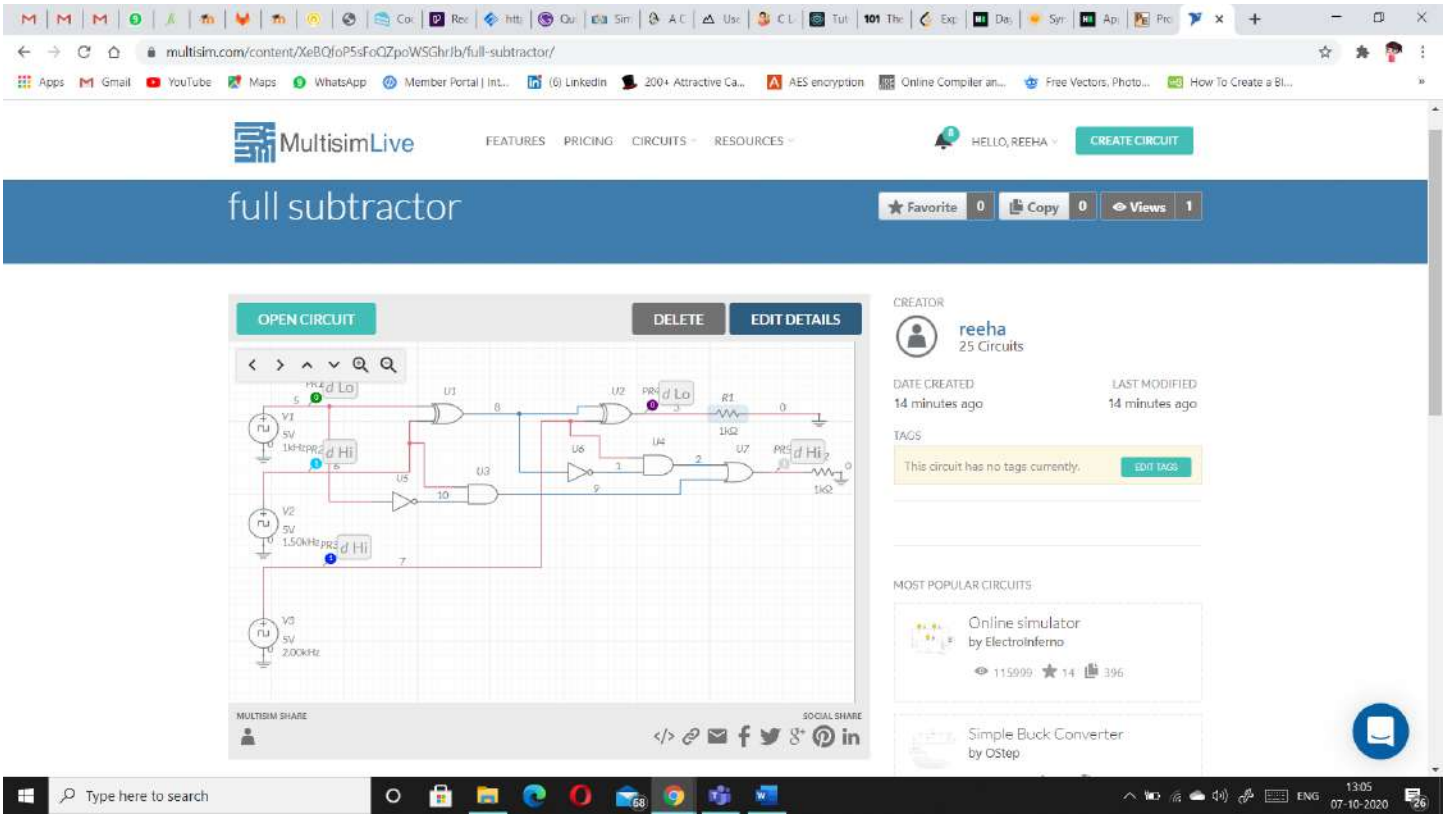
## Circuits and Output waveform

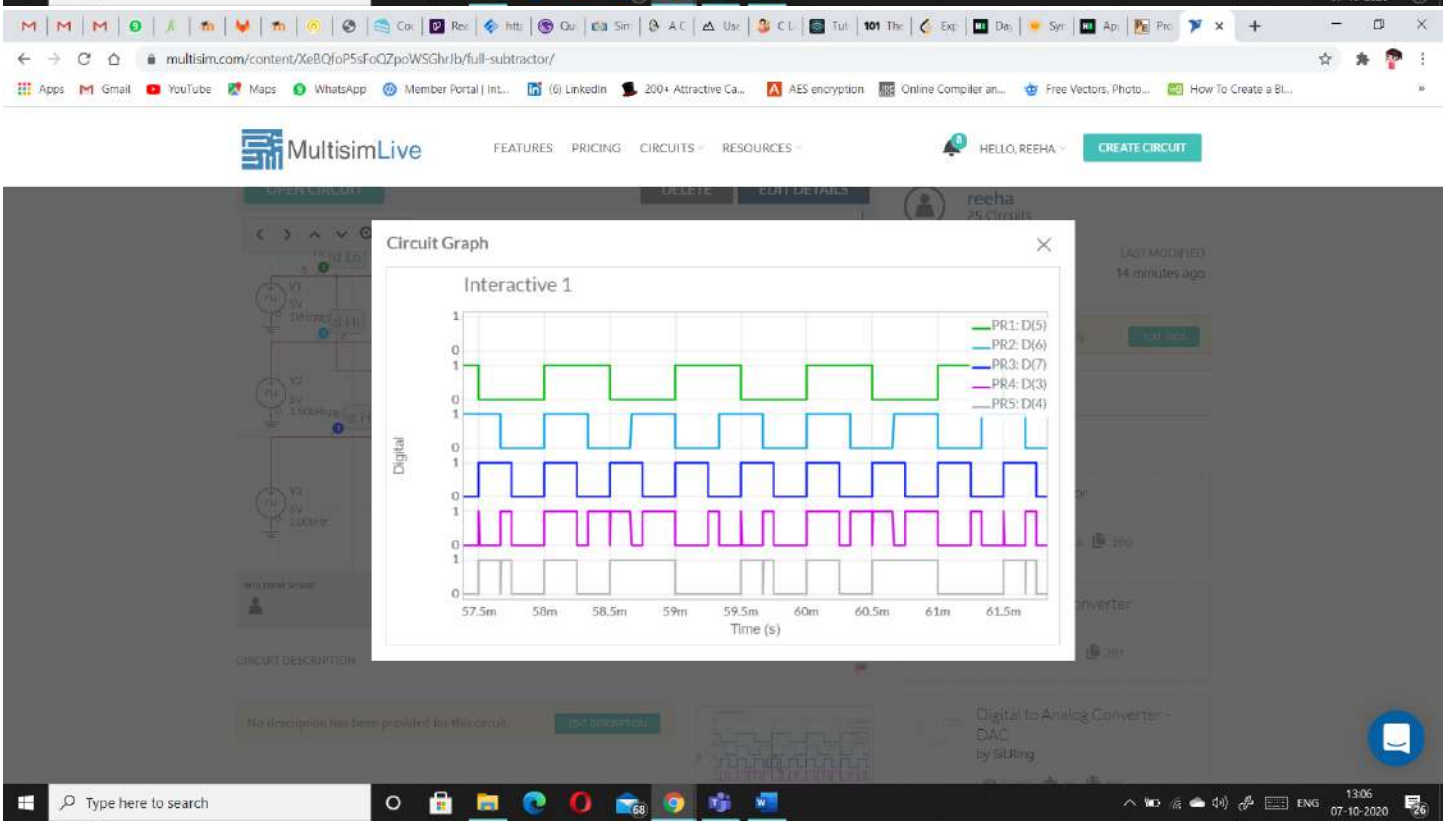
# Half Subtractor



# 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 BOROW 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.

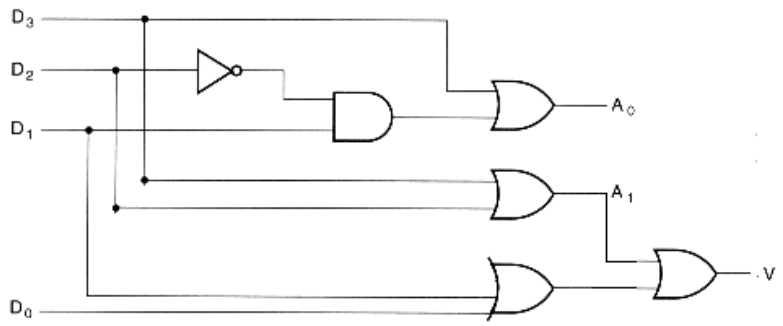
### Software Simulation:

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

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 to 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 inspected 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 |

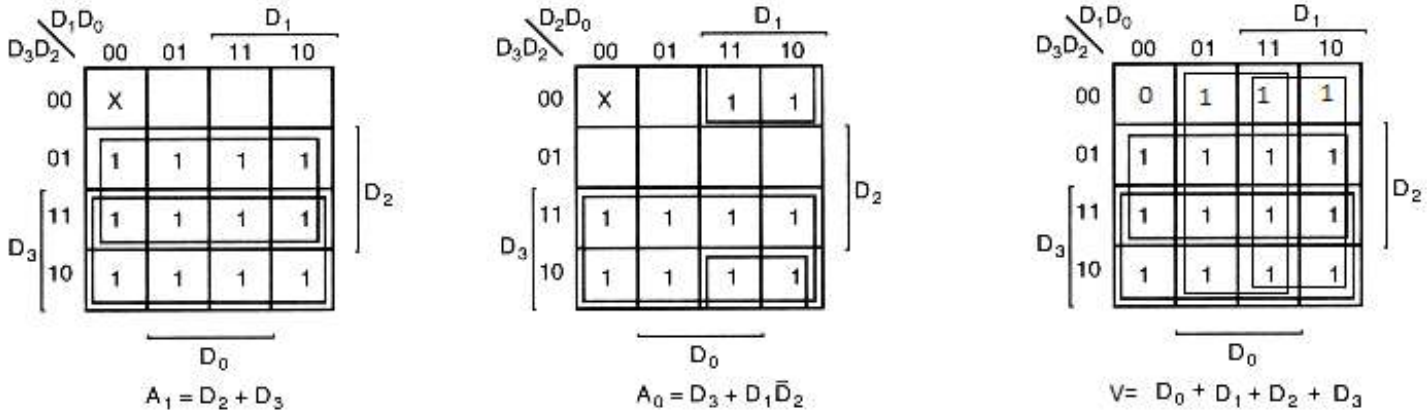



Fig 4.1 Logic diagram along with K-Map minimization

## 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.

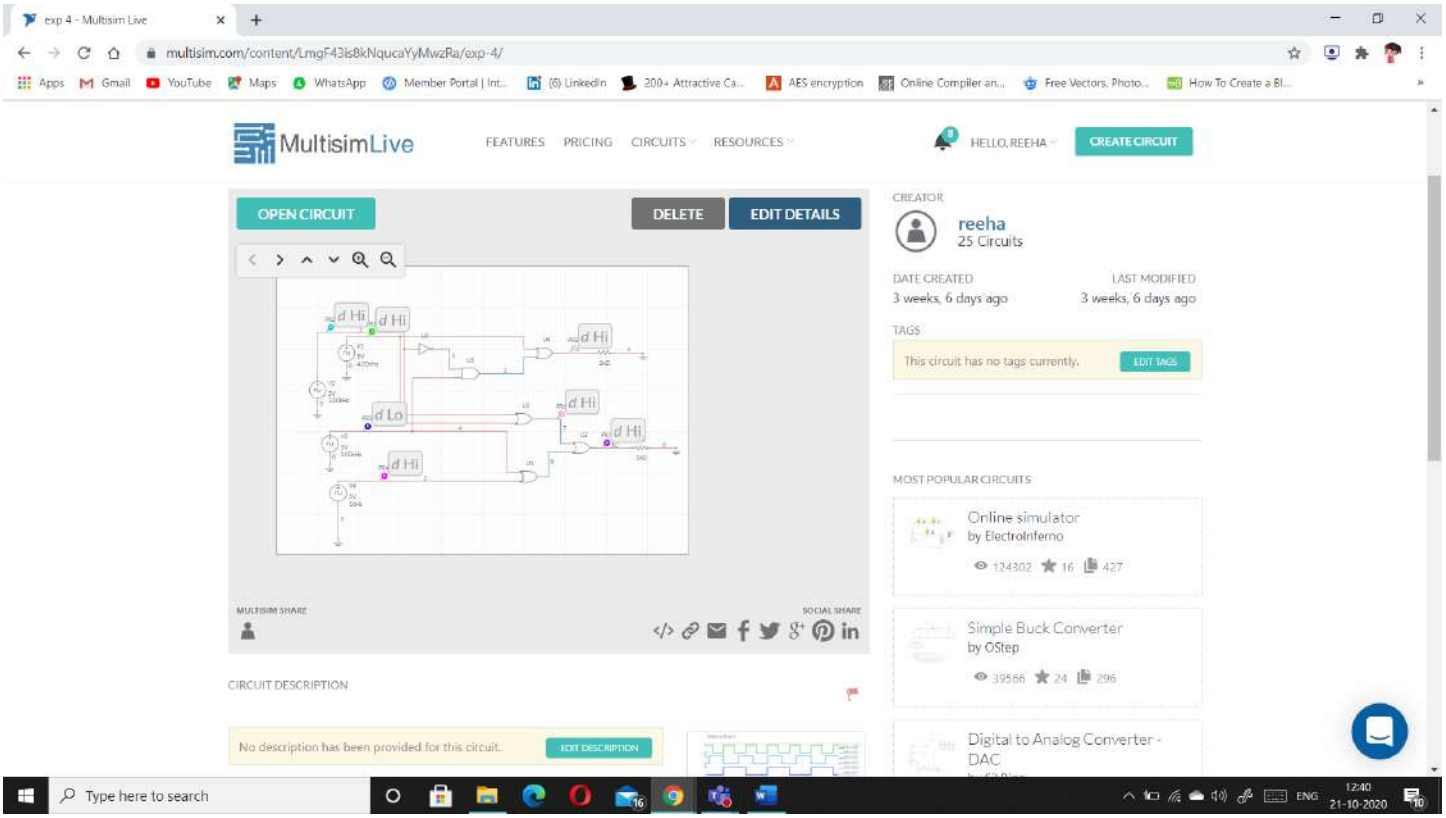
1. All the connection should be tight.
2. Always connect ground first and then the supply.
3. Switch off the power supply after completion of the experiment.

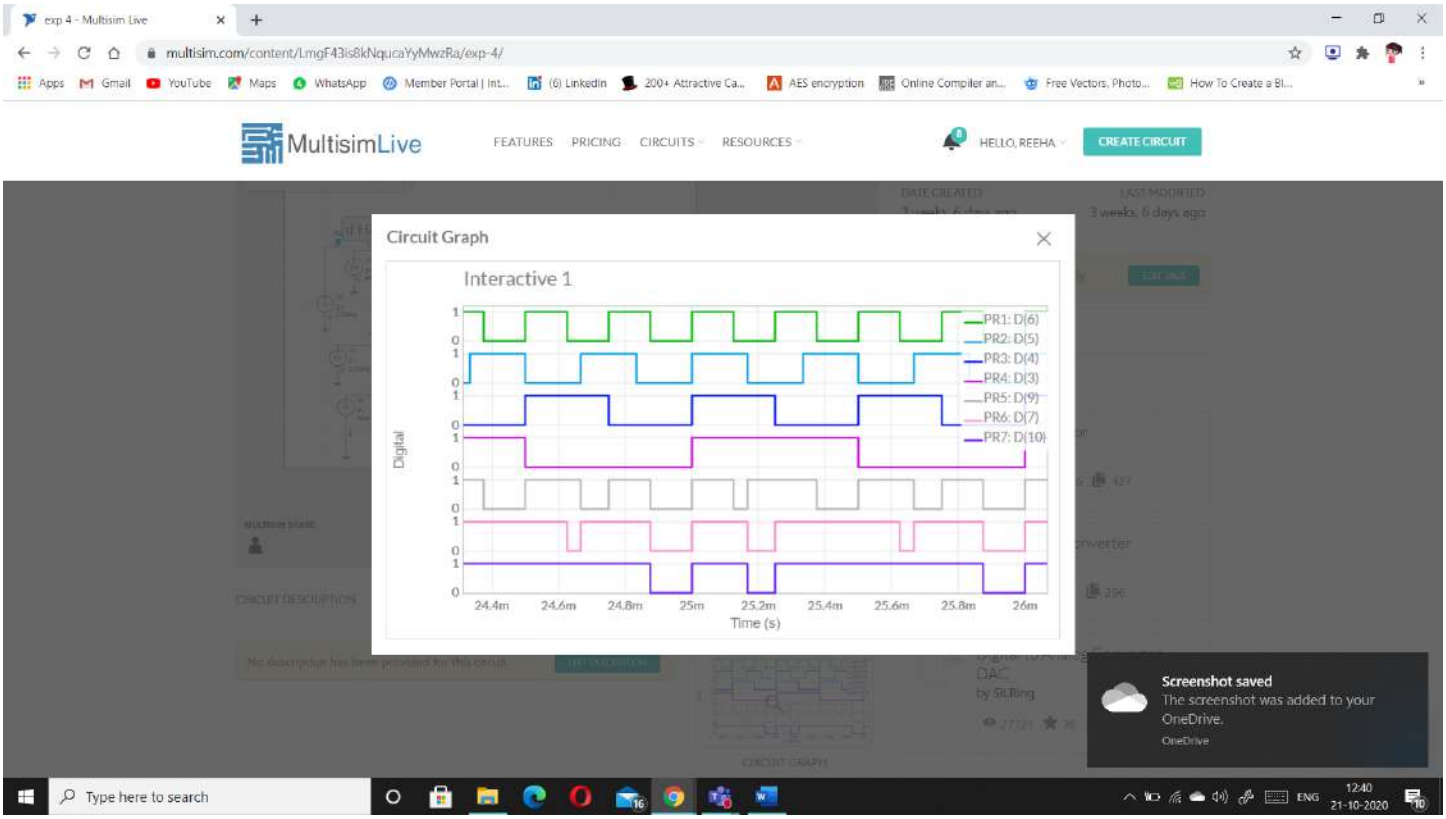
## RESULT:

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

## Circuits and Output waveform

# Priority Encoder



# 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 2n 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.

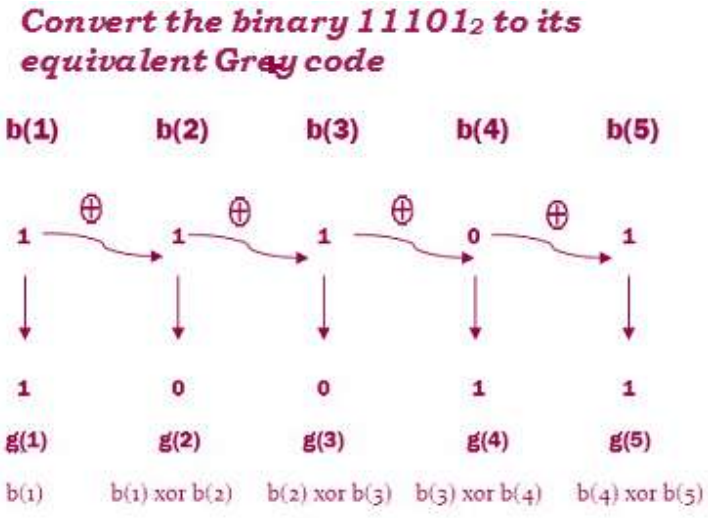
Components used – Source (Clock Voltage), Passive elements (resistor), Digital components (AND, OR, NAND, NOR, XOR, XNOR, Inverter), Probe for Analysis and annotation (Digital), Schematic connectors (Ground)

# 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**

a)



b)

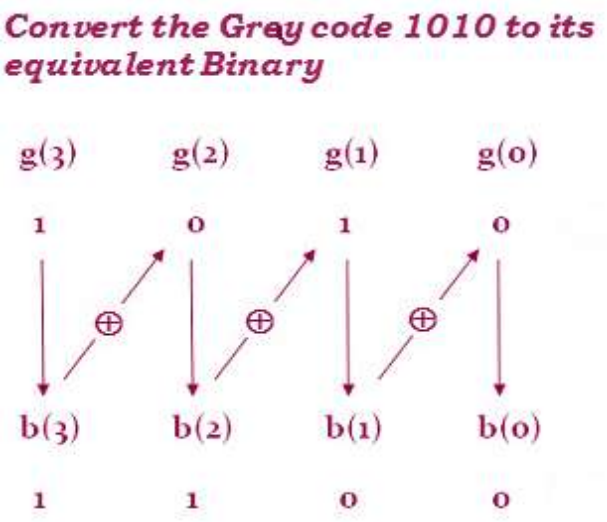
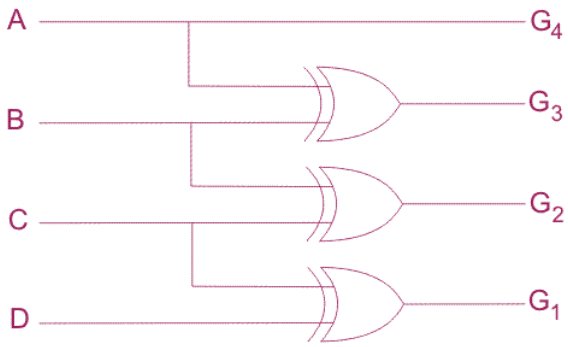



Fig 13.1 a) Binary to gray b) Gray to Binary

## Logic Diagrams

**a)**




**b)**

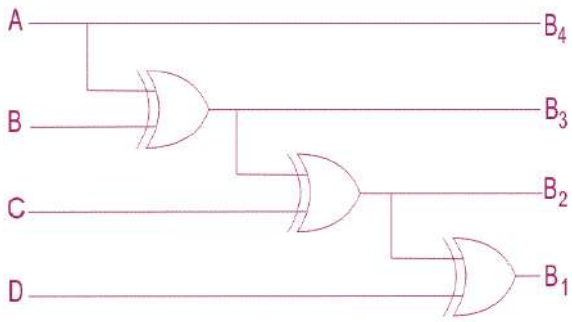



Fig 13.2 Logic Diagram of a) Binary to Gray b) Gray to Binary

## *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:

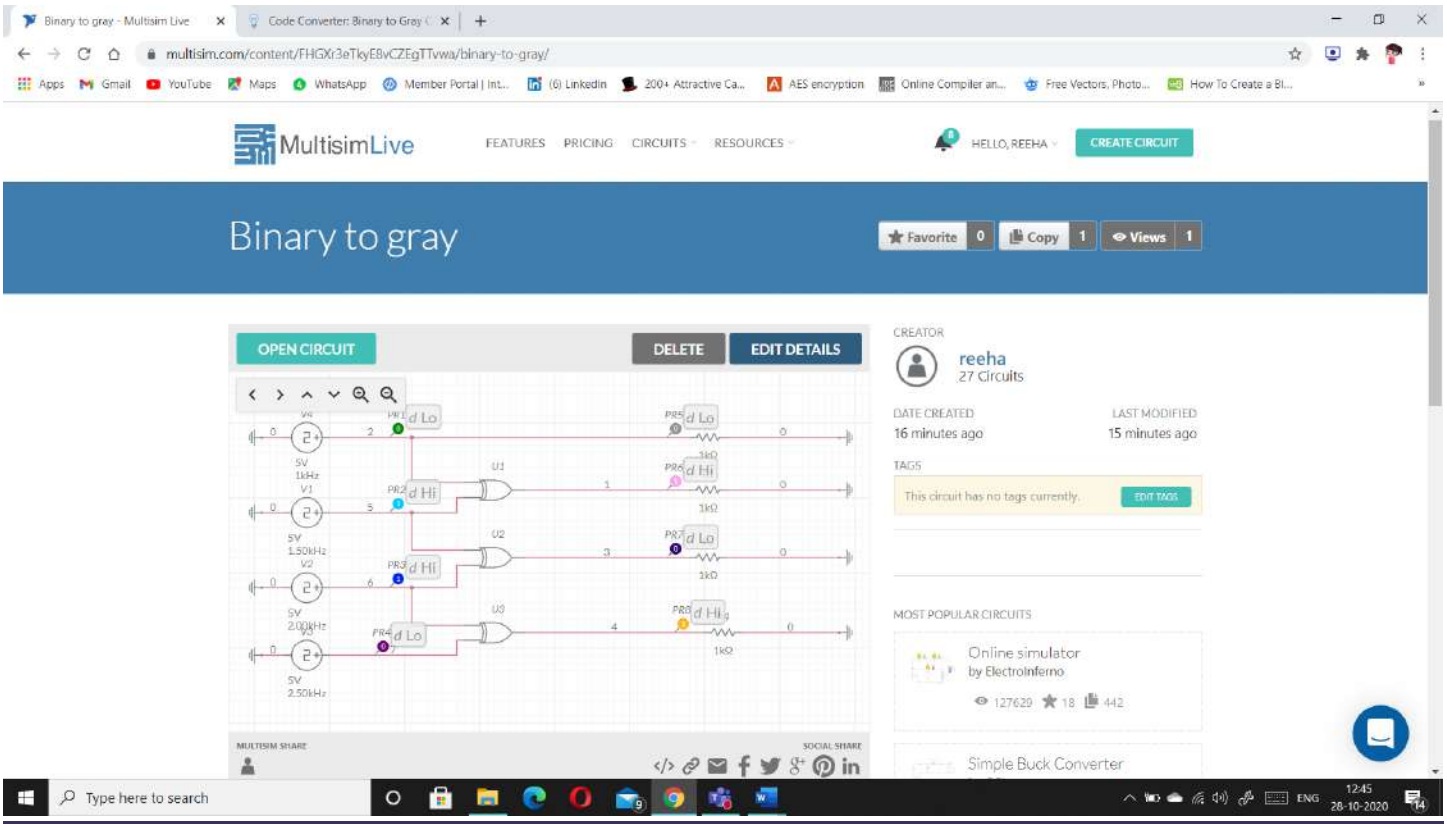
1. All ICs should be checked before starting the experiment.
2. All the connection should be tight.
3. Always connect ground first and then the supply.
4. Switch off the power supply after completion of the experiment.

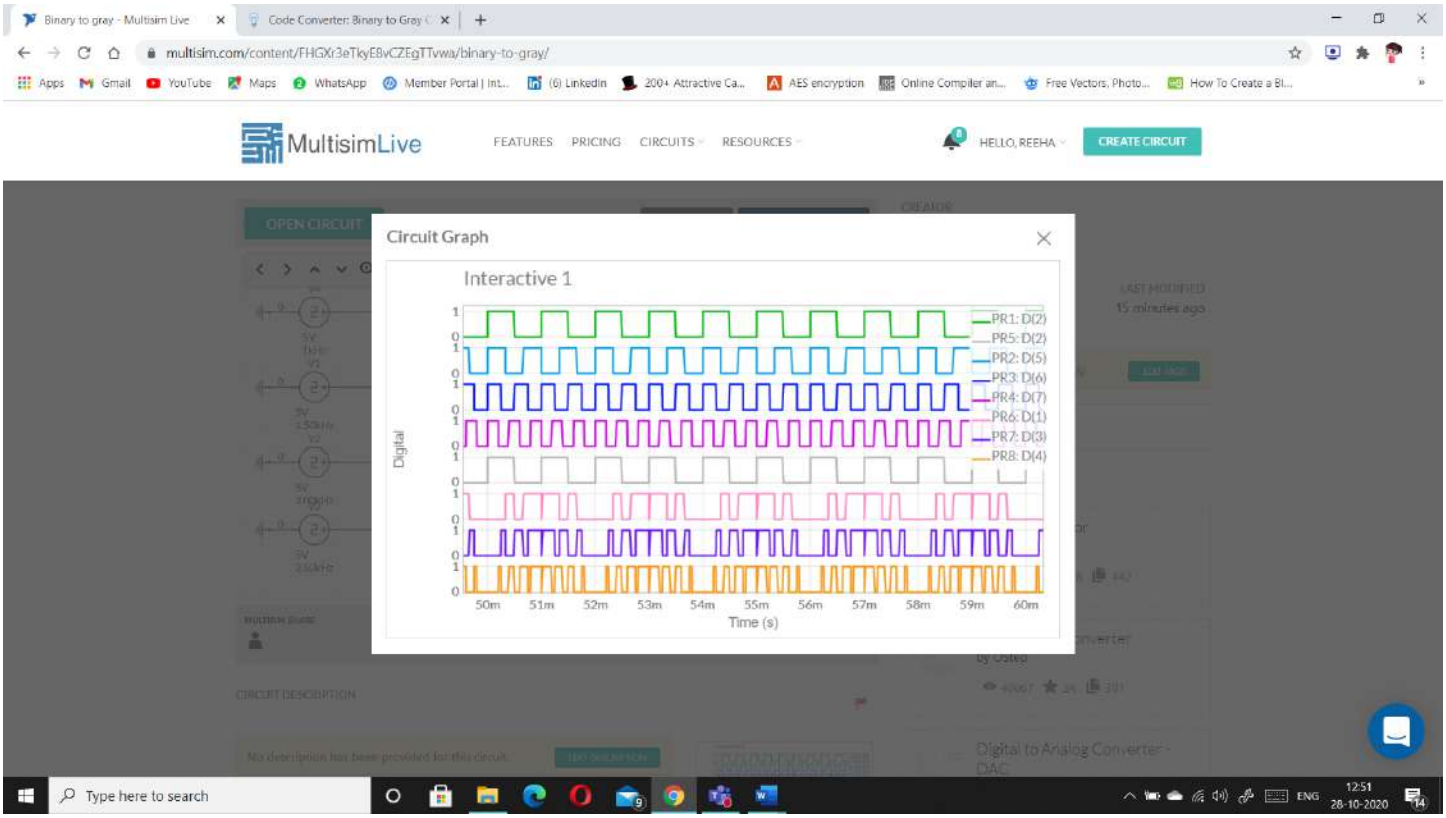
## RESULT:

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

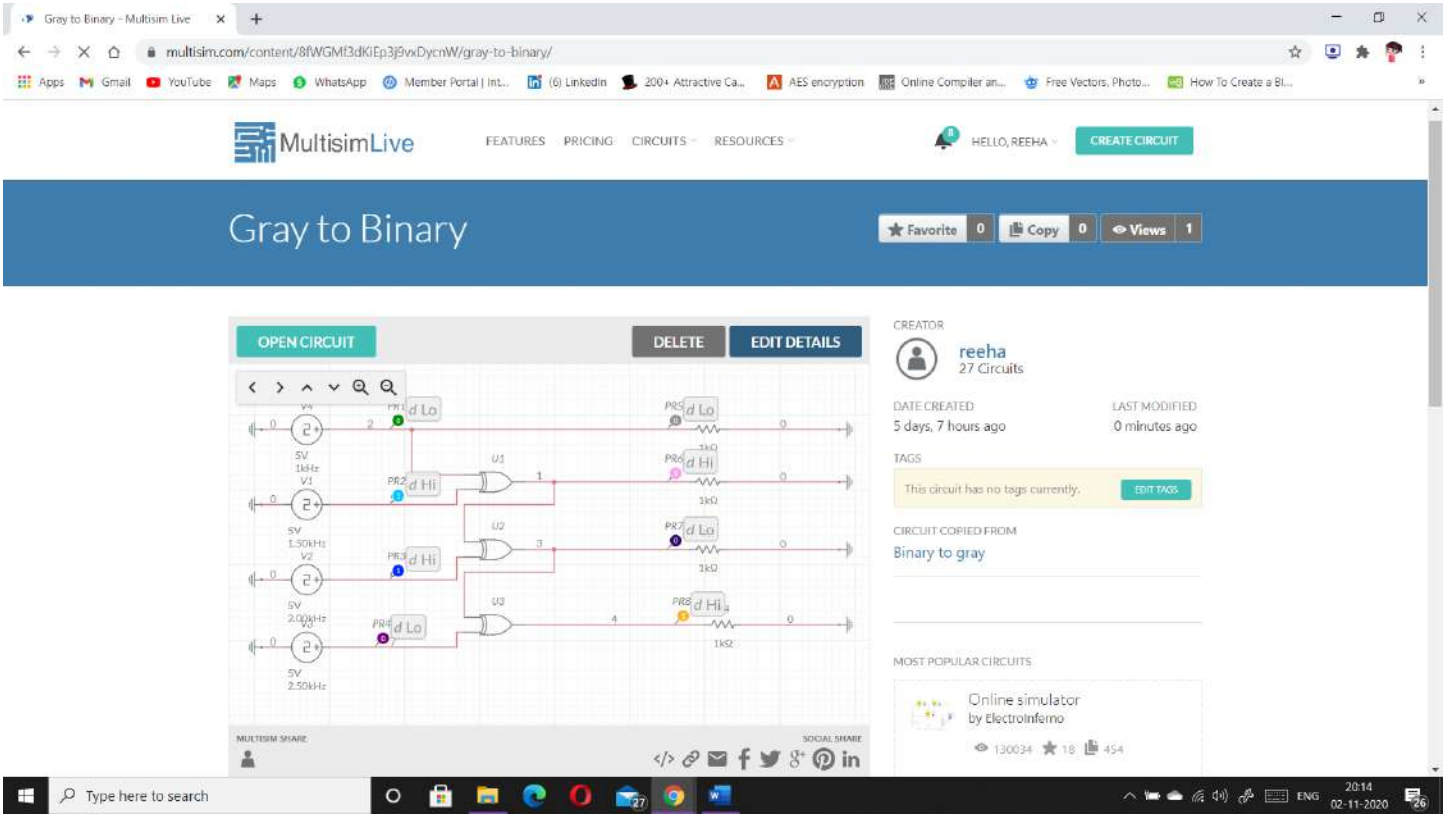
## Circuits and Output waveform

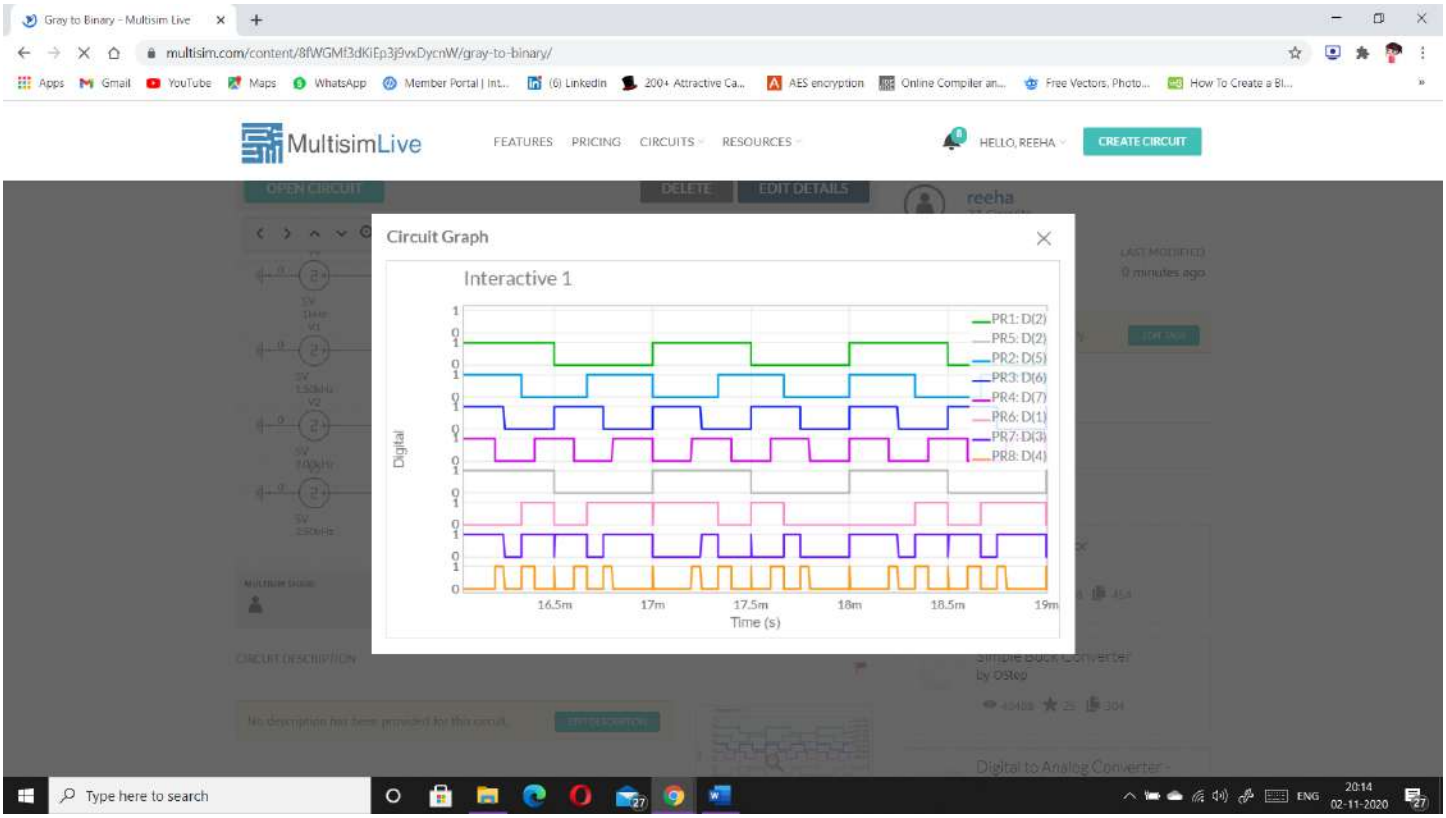
# Binary to Gray






# 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 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.

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 (A1, A0 & B1, B0).

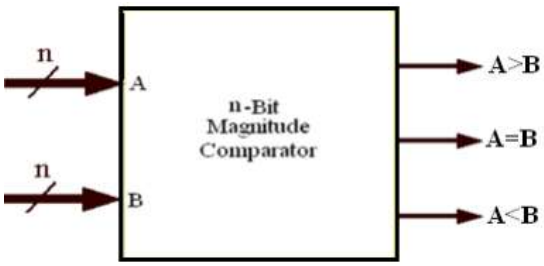


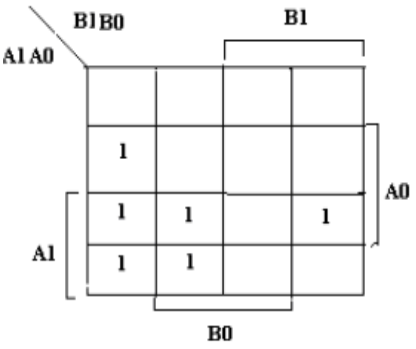

Fig 11.1 Block Diagram Of comparator

**Designing of 2 bit comparator**

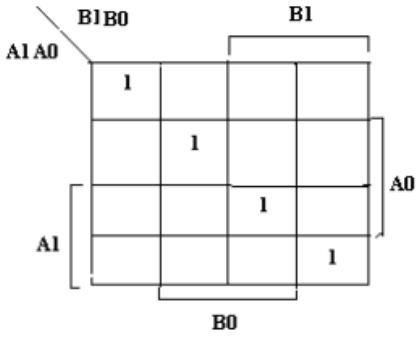
| INPUT | | | | OUTPUT | | |
|---|---|---|---|---|---|---|
| A1 | A0 | B1 | B0 | A>B | A=B | A<B |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |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

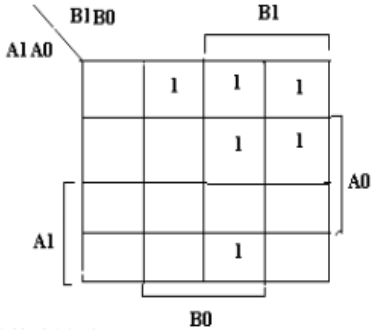
a) **FOR A>B**




**FOR A=B**




**FOR A<B**




A>B: = A1B1'+A0B0'A1'B1'+A0B0'A1B1
= A1B1'+A0B0'(A1'B1'+A1B1)
= A1B1'+A0B0' X1

A=B: = A1'A0'B1'B0'+ A1'A0B1'B0+A1A0'B1B0'+A1A0B1B0
= (A1'B1'+A1B1) (A0'B0'+A0B0)
= X1X0

A<B: = A1'B1+A0'B0A1'B1'+A0'B0A1B1
= A1'B1+A0'B0(A1'B1'+A1B1)
= A1'B1+A0'B0 X1

b)

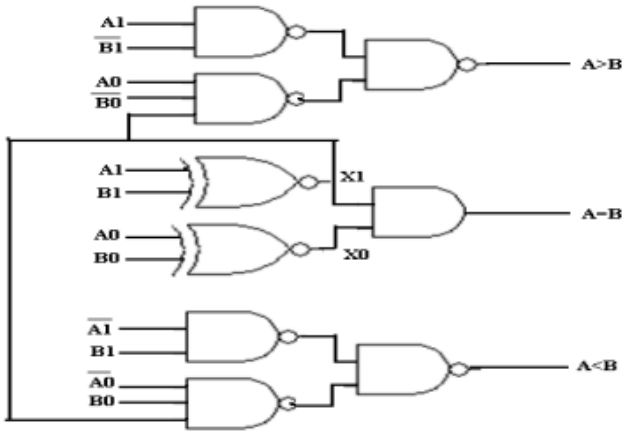



Fig 11.2 a)K Map minimization b) logic Diagram of comparator

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

```
A>B:A1B1' + A0B1'B0' + A1A0B0'
A=B: A1'A0'B1'B0' + A1'A0B1'B0 + A1A0B1B0 + A1A0'B1B0'
   : A1'B1' (A0'B0' + A0B0) + A1B1 (A0B0 + A0'B0')
   : (A0B0 + A0'B0') (A1B1 + A1'B1')
   : (A0 Ex-Nor B0) (A1 Ex-Nor B1)
A<B:A1'B1 + A0'B1B0 + A1'A0'B0
```

## 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.