

LABORATORY JOURNAL

*Submitted in partial fulfillment of the requirement
For the Subject*

“DIGITAL ELECTRONICS AND LOGIC DESIGN” (EC 207)

: Prepared & Submitted By :

**Ms. Krina S. Patel
(Admission No. U19CS008)**

**B. TECH. II (CSE) 3rd Semester
(Academic Year : 2020-21)
ONLINE MODE**



(Aug to Dec - 2020)

ELECTRONICS ENGINEERING DEPARTMENT
Sardar Vallabhbhai National Institute of Technology
Surat-395 007, Gujarat, INDIA.

Certificate

This is to Certify That

Mr./Ms. Krina Patel of B.Tech IInd Computer Admission No. U19CS008 has Satisfactorily Completed His/Her course work in **Digital Electronics and Logic Design Laboratory during the 3rd Semester Session of Academic Year 2020-2021 and Submitted on 03-December' 2020.**

**Dr. Shilpi Gupta
Subject Co-ordinator**

Director



**Sardar Vallabhbhai National Institute of Technology,
Surat**

Sr.	LIST OF EXPERIMENTS	PAGE NO.
1.	Introduction to Multisim	5
2.	Study of Basic and Universal Gates	26
3.	Half Adder and Half Subtractor	53
4.	Full Adder and Full Subtractor	64
5.	V-I Characteristics of PN – Junction Diode	76
6.	Diode Clipper Circuits (Series – Configuration)	83
7.	Diode Clipper Circuits (Shunt – Configuration)	106
8.	Diode Clamper Circuits	130
9.	Full Wave Rectifier	144
10.	Common Emitter Characteristics & Common Emitter Amplifier	152
11.	Registers and Counters	176

12.	Multiplexers and Code Converters	218
13.	High Pass and Low Pass Filters	251

- Submitted By
Krina S. Patel

Admission Number : U19CS008

B.Tech. II (CSE) 3rd Semester



Expt. No:

1

Date:

14-08-2020**Introduction to Multisim**

AIM: To study the Multisim software interface and the tools thereby get acquainted with implementing and simulating circuits using Multisim Live Simulator.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Live (Online Interface)

WORKING ON MULTISIM LIVE SIMULATOR:

NI Multisim (formerly MultiSIM) is an electronic schematic capture and simulation program which is part of a suite of circuit design programs, along with NI Ultiboard. Multisim is one of the few circuit design programs to employ the original Berkeley SPICE based software simulation. Multisim was originally created by a company named Electronics Workbench, which is now a division of National Instruments. Multisim includes microcontroller simulation (formerly known as MultiMCU), as well as integrated import and export features to the printed circuit board layout software in the suite, NI Ultiboard. Multisim is widely used in academia and industry for circuits education, electronic schematic design and SPICE simulation.

Multisim Live is a free online circuit simulator that includes SPICE software, which lets you create, learn and share electronics circuits online.

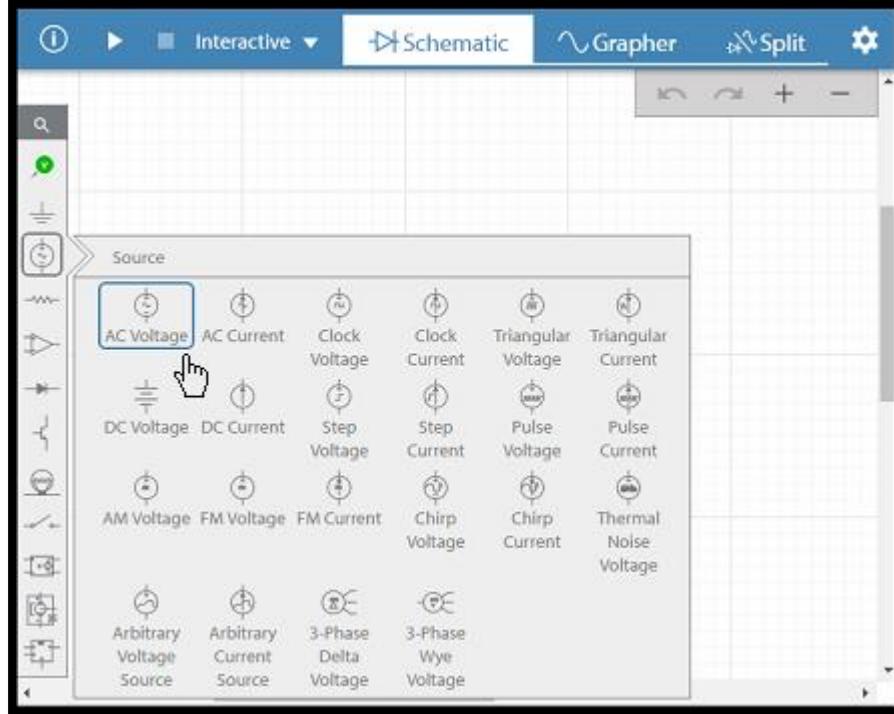
Creating Circuits on Multisim:

The screenshot shows the MultisimLive homepage. At the top right, there is a user profile placeholder 'HELLO, SUOHANSHU12291'. Below it, a prominent green button labeled 'CREATE CIRCUIT' is circled in red. The main banner features the text 'Discover Electronics with Online SPICE Simulation' and a 'SEE HOW IT WORKS' button. In the background, a person is shown working on a laptop and a tablet, illustrating the software's use in a real-world setting.

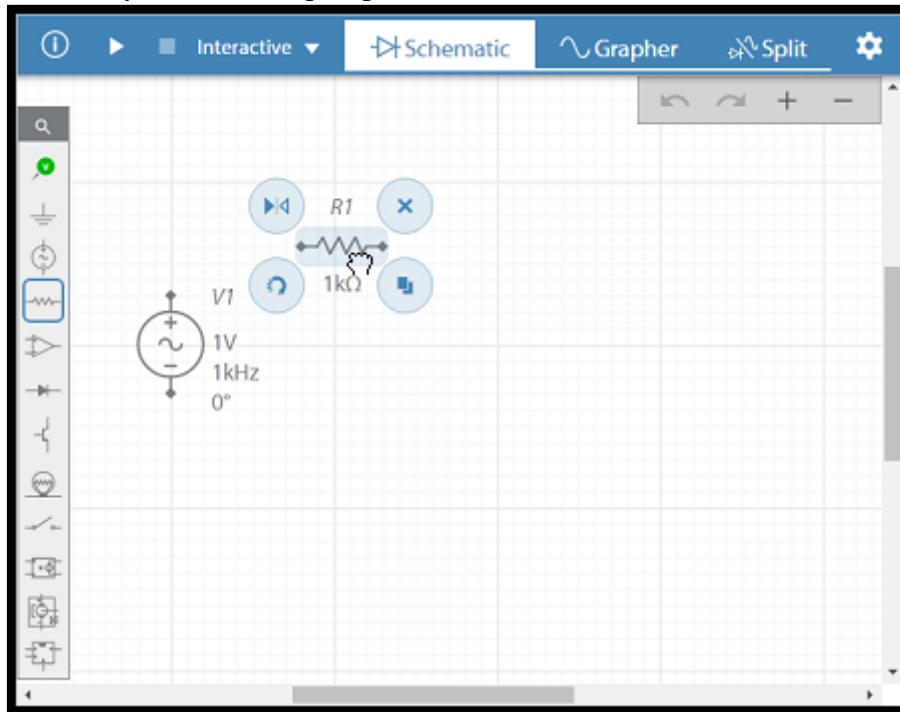
- ▶ UI appears as shown.
- ▶ Click on Create Circuit

**Placing Voltage Source:**

Tap the Source subpalette and tap AC Voltage and tap on the workspace or Type V if you are using a device with a keyboard, and tap to place the source.

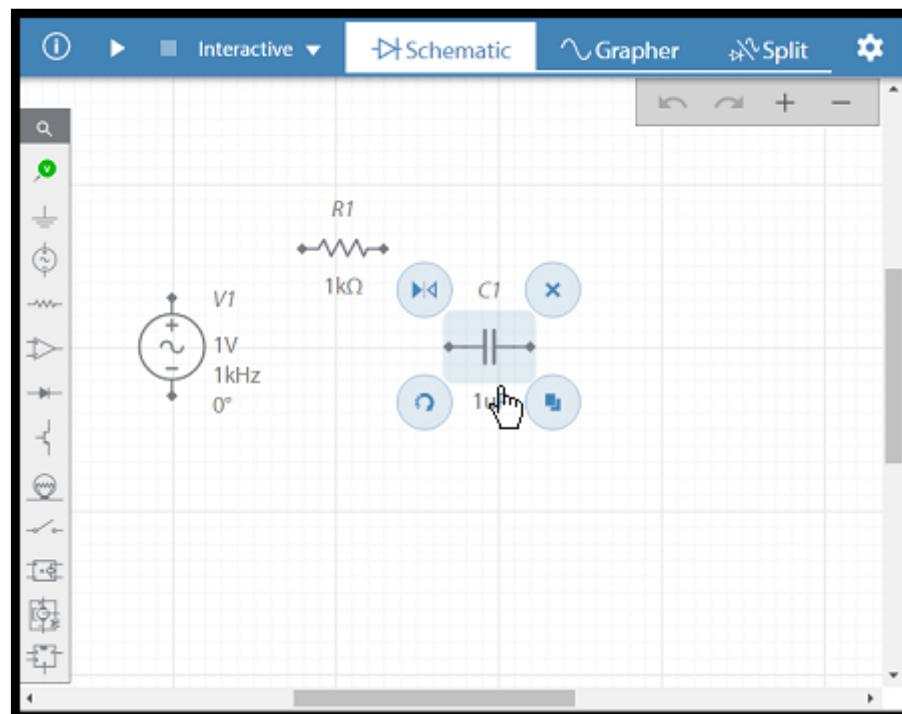
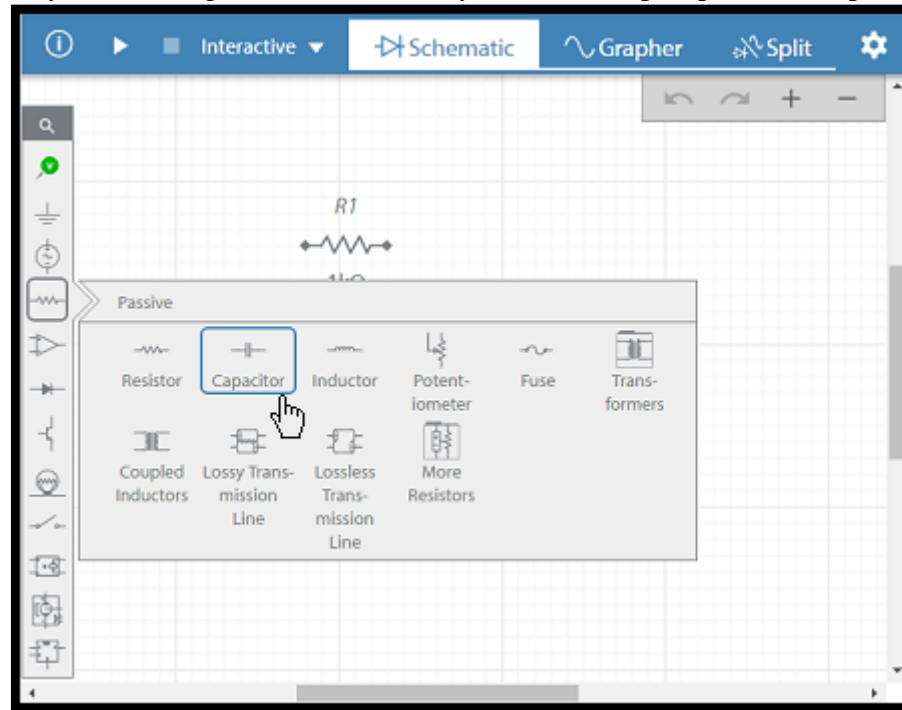
**Placing Resistor:**

Place a resistor by dragging from the Passive subpalette or Type R if you are using a device with a keyboard, and tap to place the resistor.



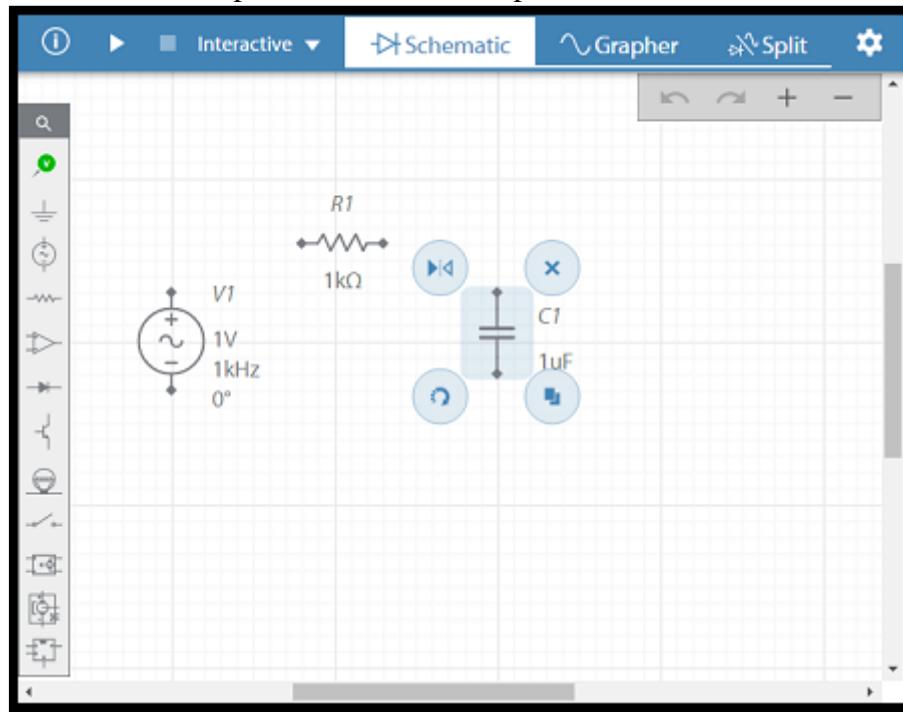
**Placing Capacitor:**

Type C if you are using a device with a keyboard, and tap to place the capacitor.

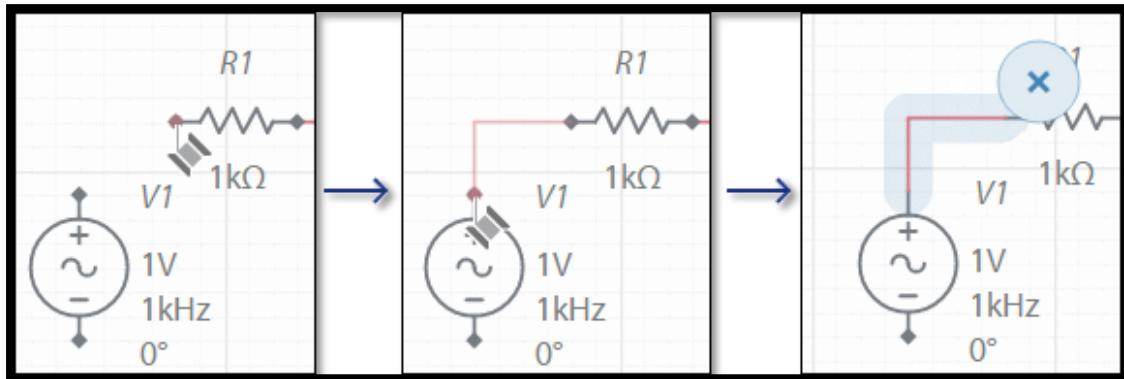


**Rotating Components:**

Tap  to rotate the capacitor and other components.

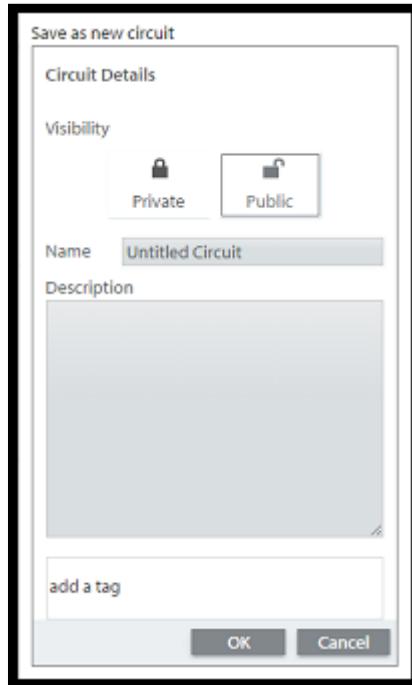
**Wiring the components:**

Tap a component's wiring point (black diamond) and tap another wiring point. The connection is automatically made, and the new wire is selected.



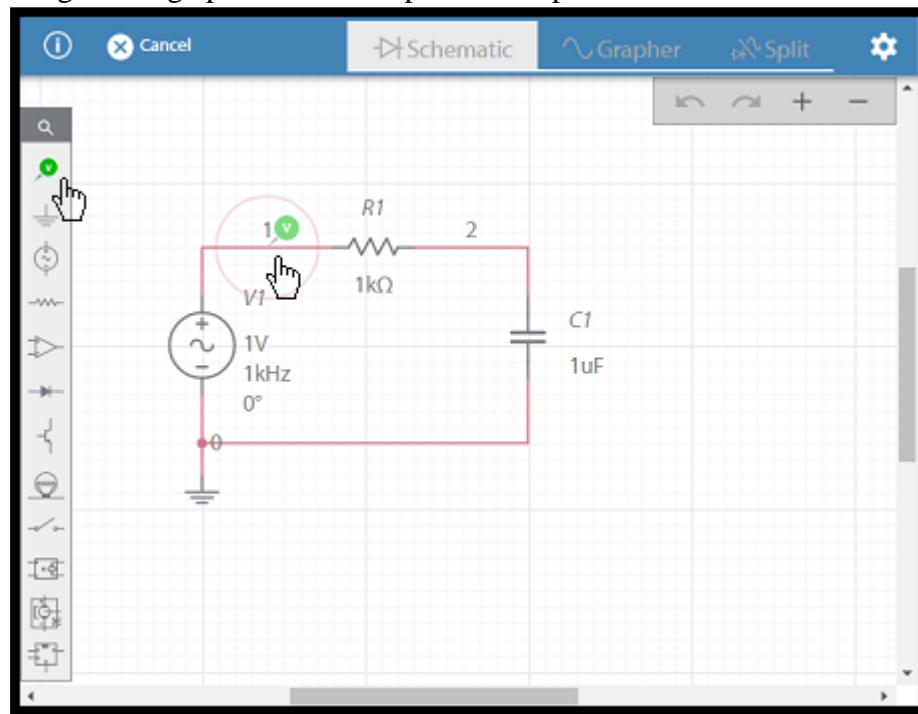
**Saving the design:**

Tap in the title bar and select Save as.

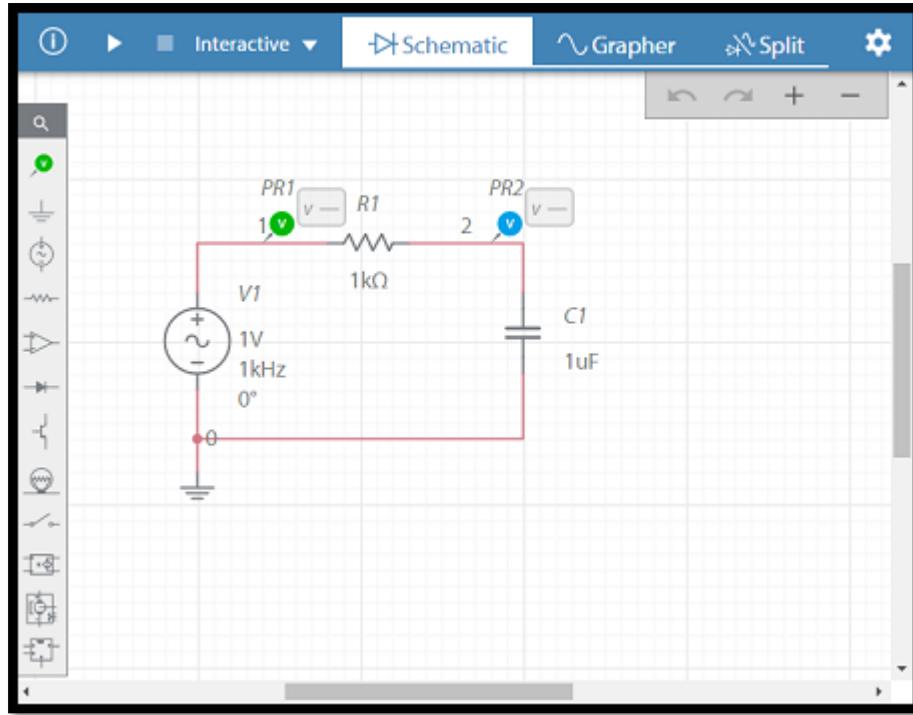
**Simulating a Design:**

To run a simulation, you must place at least one probe.

1. Drag a voltage probe from the palette and place as shown below.

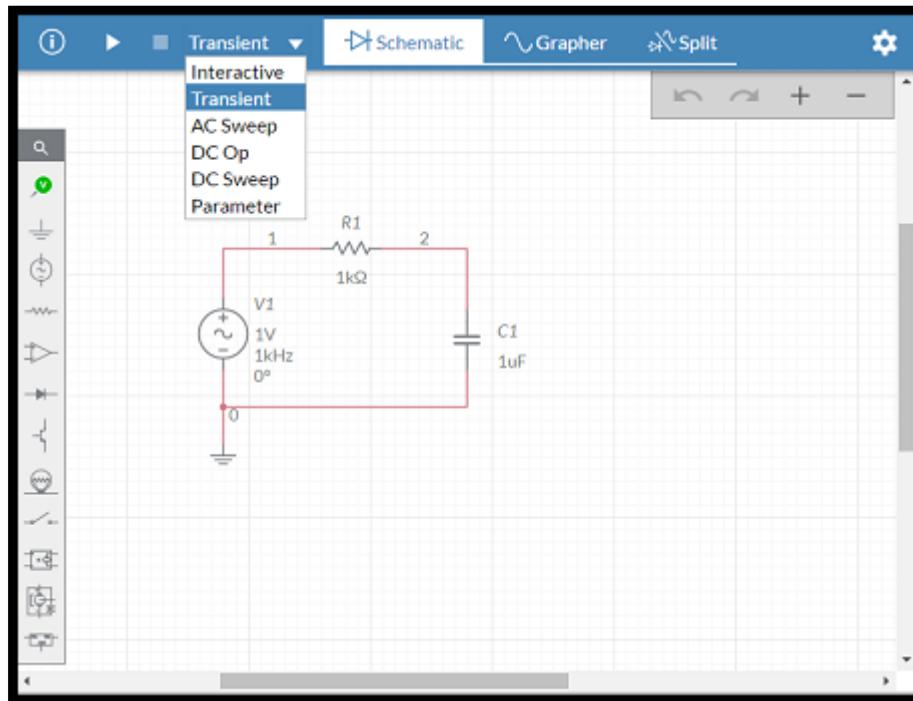


2. Place a second probe.



Select and run simulation

1. Select Transient from the toolbar.



2. Tap in the toolbar. For transient simulation, the view switches to the [grapher](#).
3. Tap in the toolbar to open the configuration pane. You can also double-tap on the grapher.
4. Use the Plots and Axes sections to manipulate the grapher as desired.



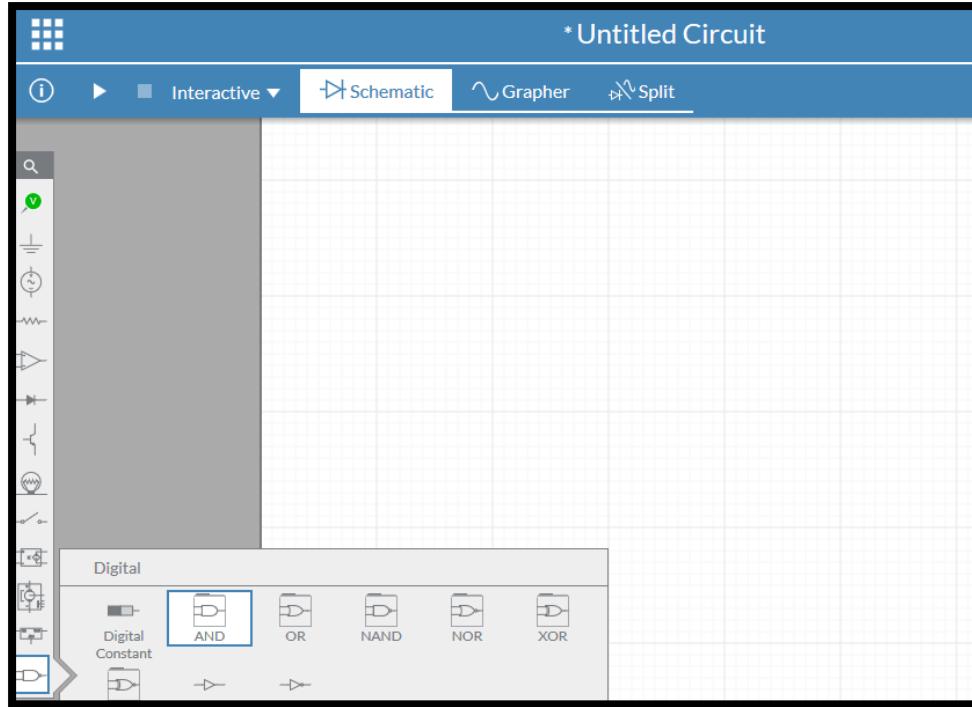
5. Switch the simulation type to AC Sweep and tap



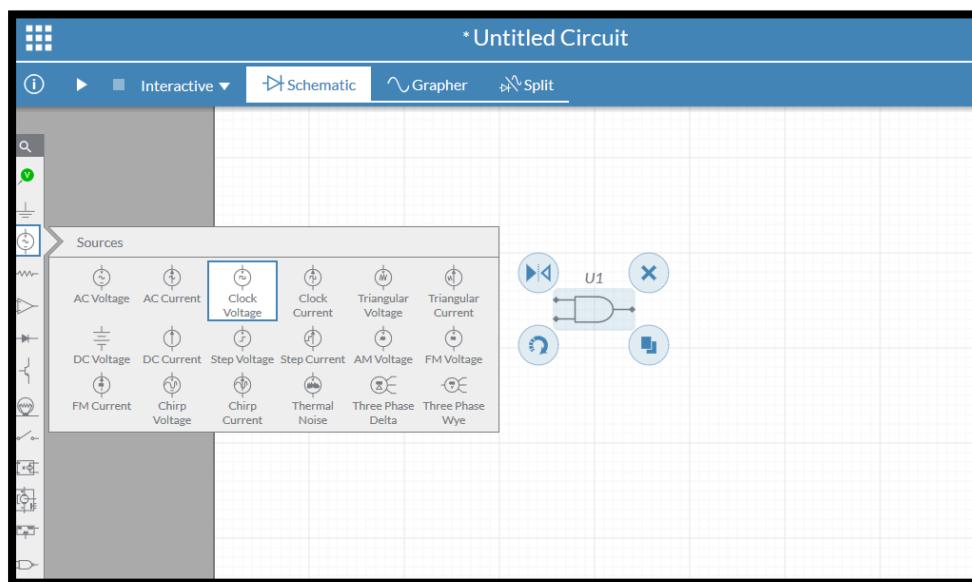


Simulating a simple Digital Circuit:

Step 1: Selecting AND Gate

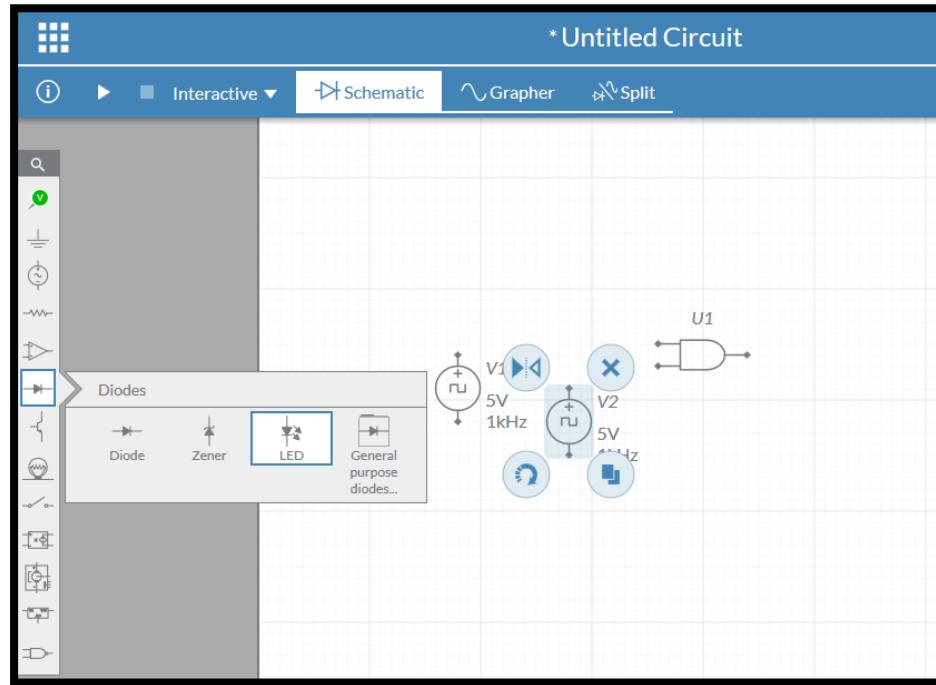


Step 2: Adding Source (Clock Voltages)

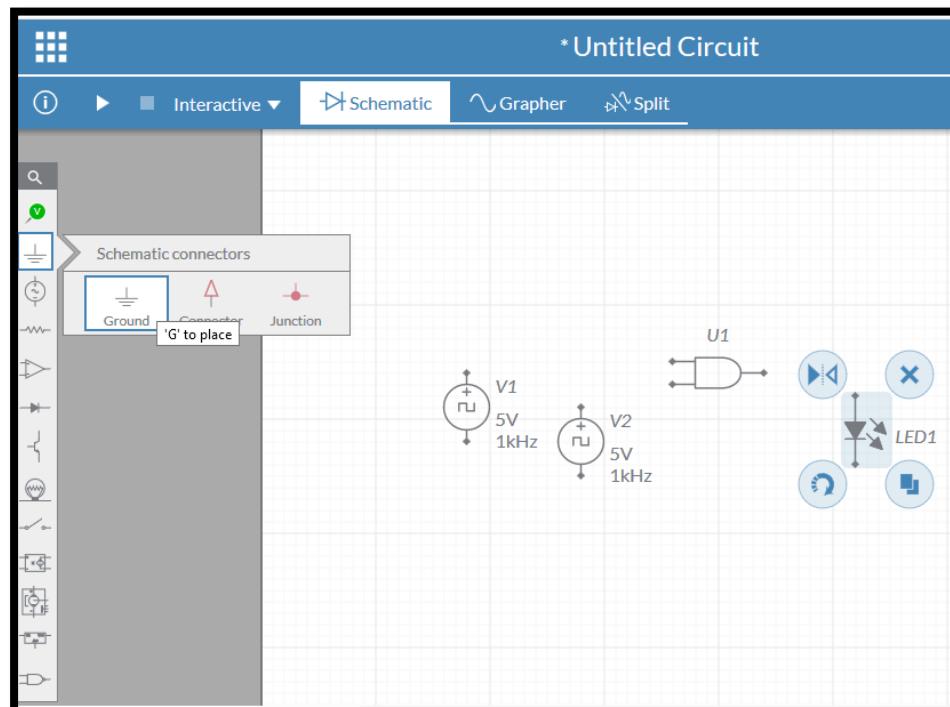




Step 3: Adding Load (LED)

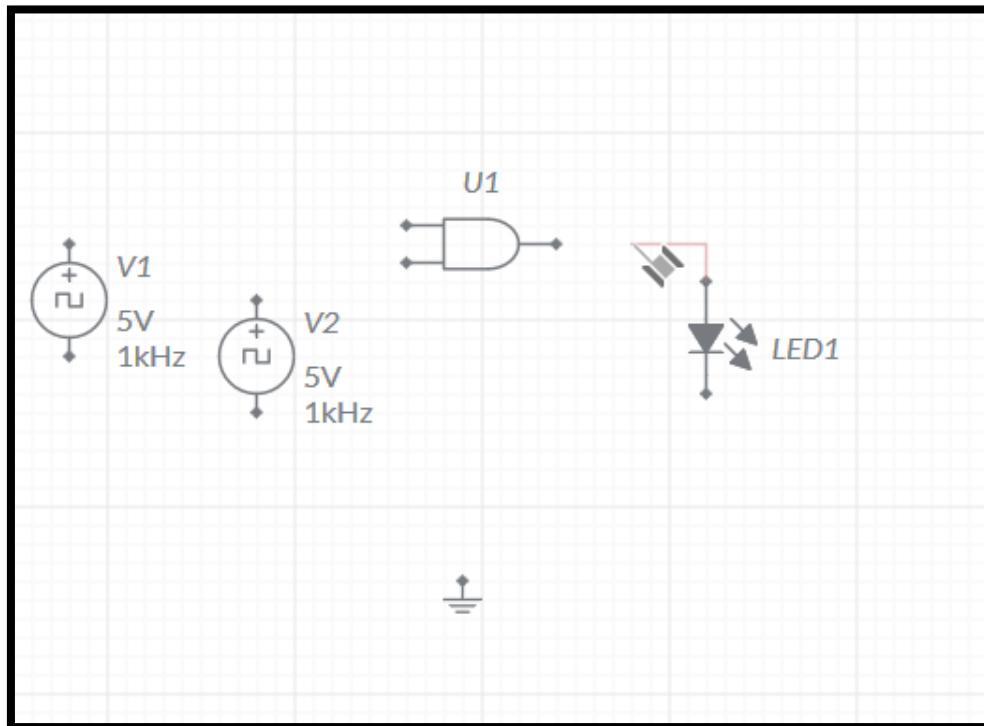


Step 4: Grounding the Components:

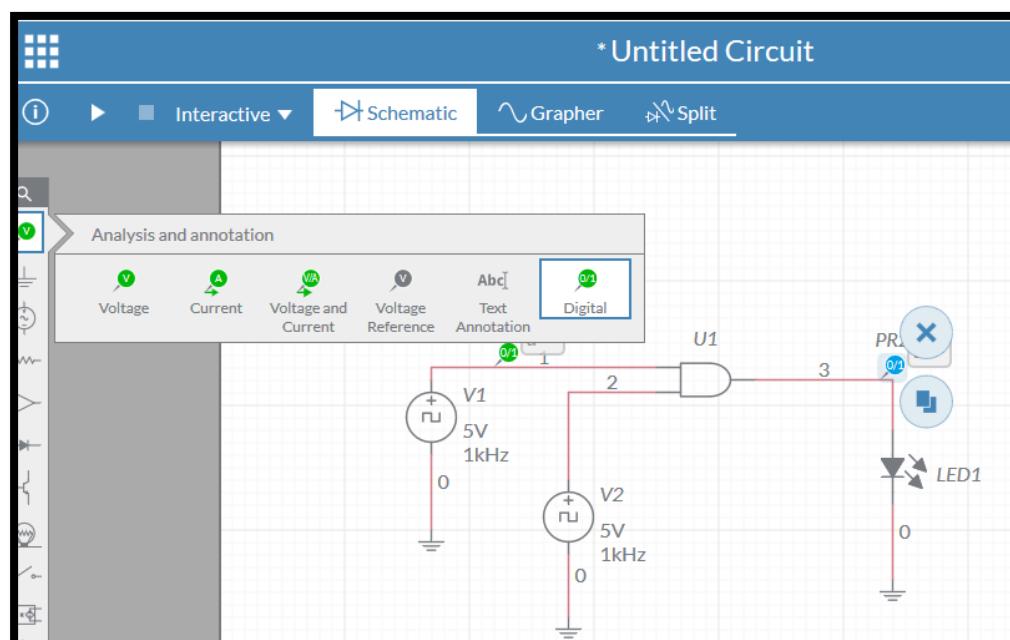




Step 5: Connecting Components



Step 6: Adding Probes



Step 7: Finally we Simulate the Design

**Step 8: Exporting Schematic/Grapher Images/Screenshots****CONCLUSIONS:**

WE LEARNED AND IMPLEMENTED MULTISIM SOFTWARE INTERFACE AND THE TOOLS THEREBY GET ACQUAINED WITH IMPLEMENTING AND SIMULATING CIRCUITS USING MULTISIM LIVE SIMULATOR. ABLE TO SIMULATE THE ASSIGNMENT CIRCUIT AND VERIFY OHM'S LAW..



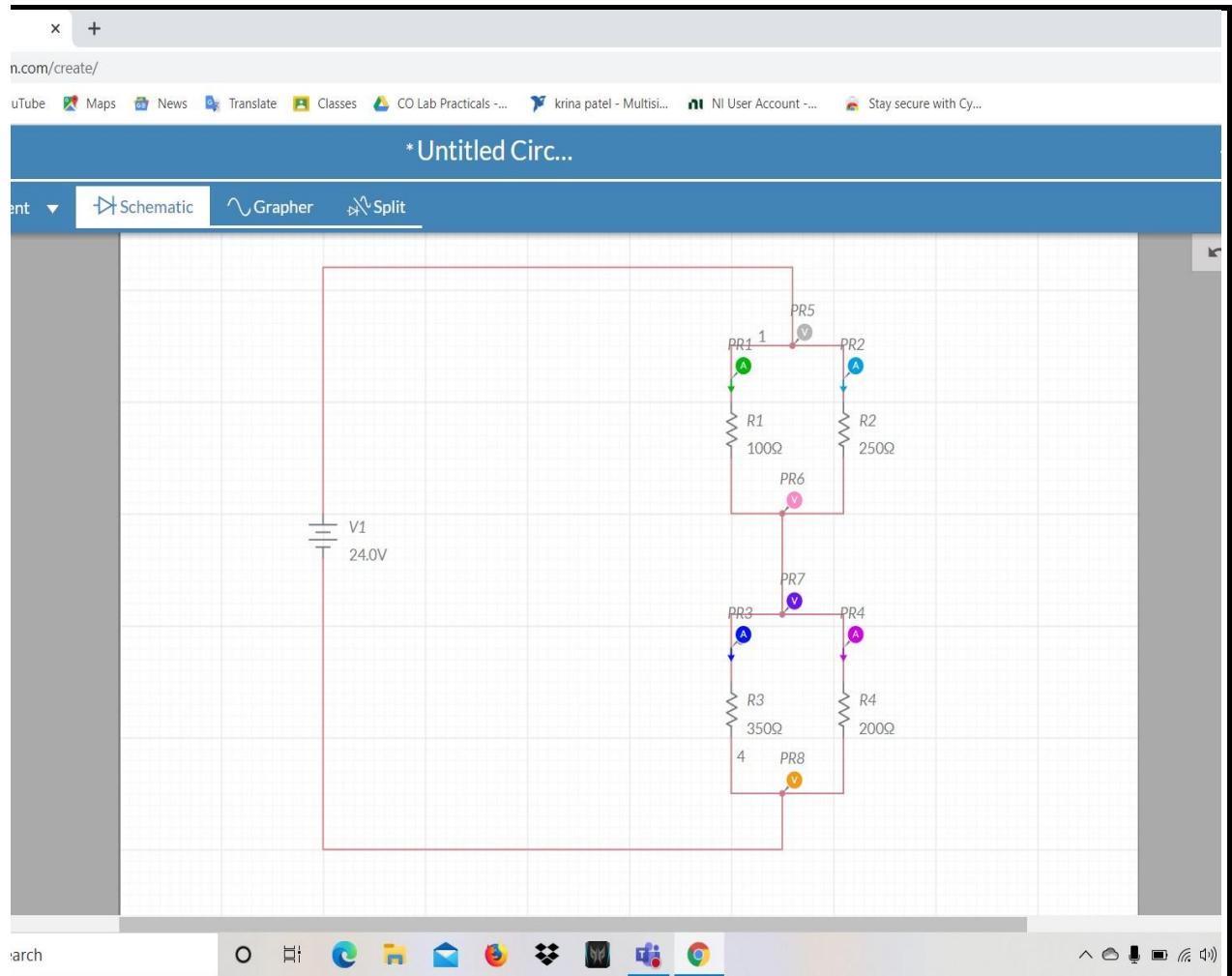
DLD LAB : ASSIGNMENT 1 AND 2

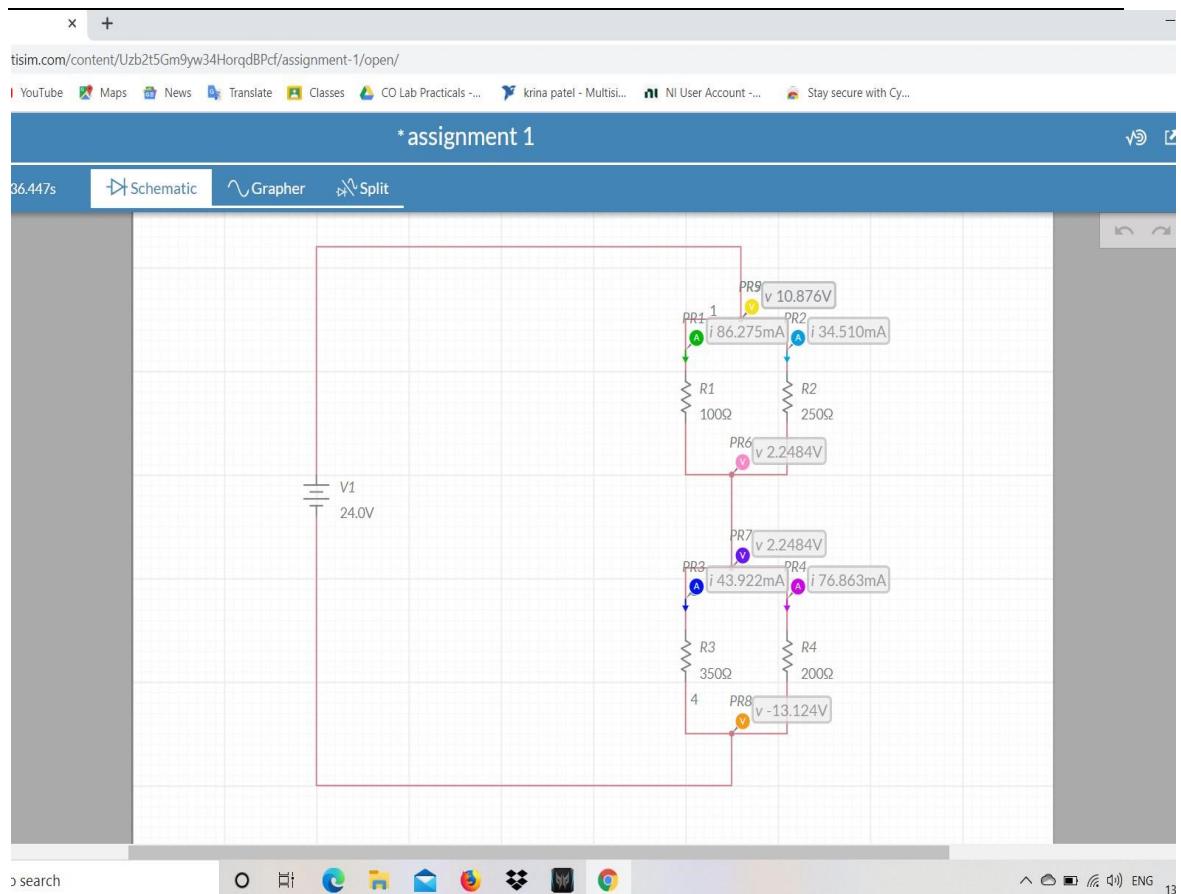
Admission Number: U19CS008

Name : Krina Patel

Assignment 1

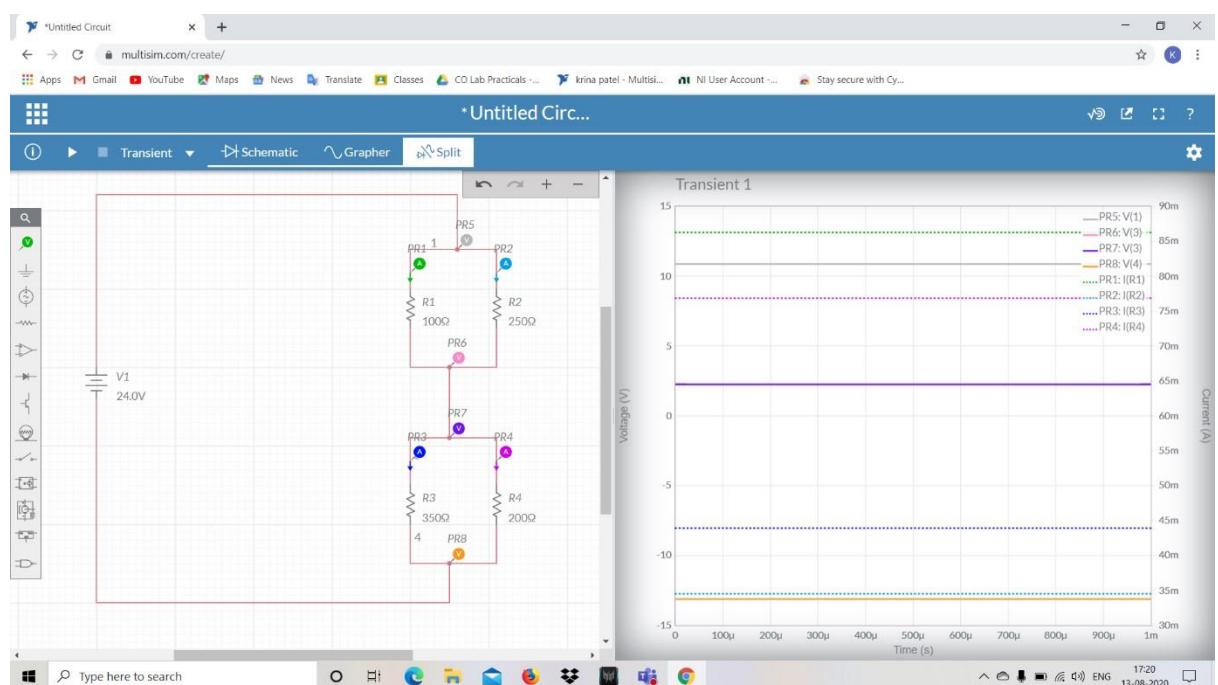
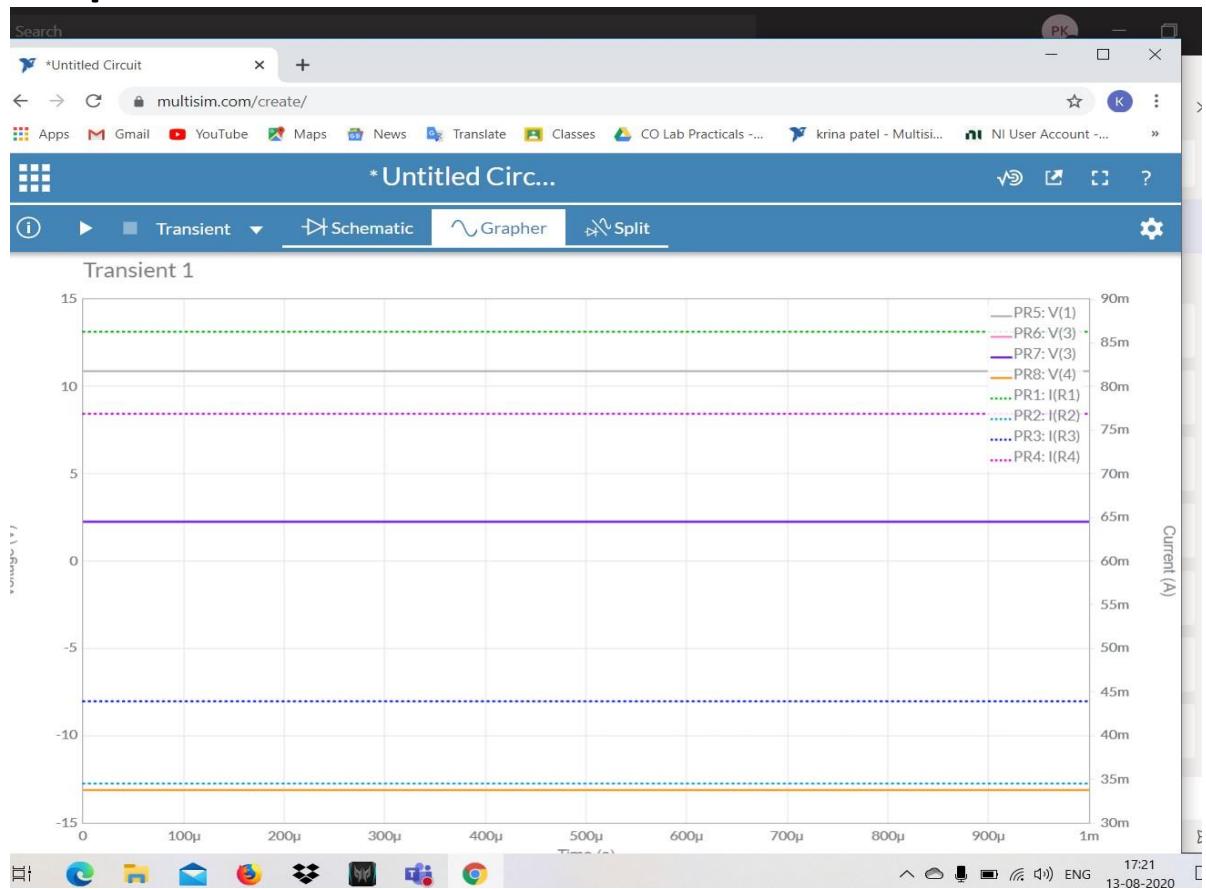
Circuit Diagram:







Graph:



**Calculation:**

Assignment-1

Good Luck	Page No.
Date	

⇒ Theoretical values

$V_{net} = 24 \text{ V}$.

$$R_{net} = \frac{1}{\left(\frac{1}{100} + \frac{1}{250}\right)} + \frac{1}{\left(\frac{1}{350} + \frac{1}{200}\right)}$$

$$= 71.43 + 127.27$$

$$= 198.7 \Omega$$

$$\therefore I_{net} = \frac{V_{net}}{R_{net}} = \frac{24}{198.7} = 0.121 \text{ A}$$

for resistor R_1 :

$$I_1 = \frac{250}{350} \times I_{net} = 0.0864 \text{ A}$$

$$V_1 = I_1 \times R_1 = 0.0864 \times 100 = 8.64 \text{ V}$$

for resistor R_2 :

$$I_2 = \frac{100}{350} \times I_{net} = 0.0346 \text{ A}$$

$$V_2 = 8.64 \text{ V}$$

for resistor R_3 :

$$I_3 = \frac{200}{500} \times I_{net} = 0.044 \text{ A}$$

$$\textcircled{1} V_3 = I_3 \times R_3 = 15.4 \text{ V}$$

for Resistor R_4 :



Handwritten notes on lined paper:

$$I_{\text{net}} = \frac{350}{550} \times I_{\text{net}} = 0.077 \text{ A}$$

$$V_4 = 15.4 \text{ V}$$

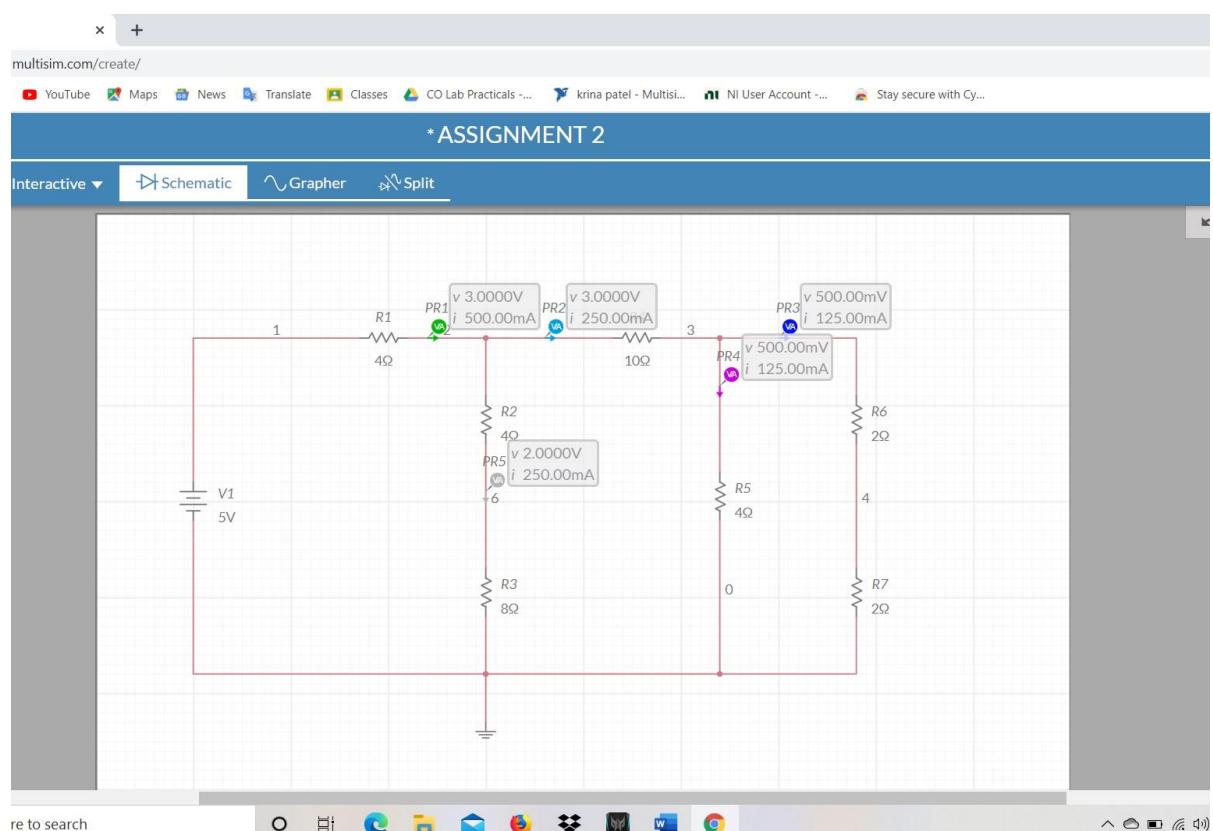
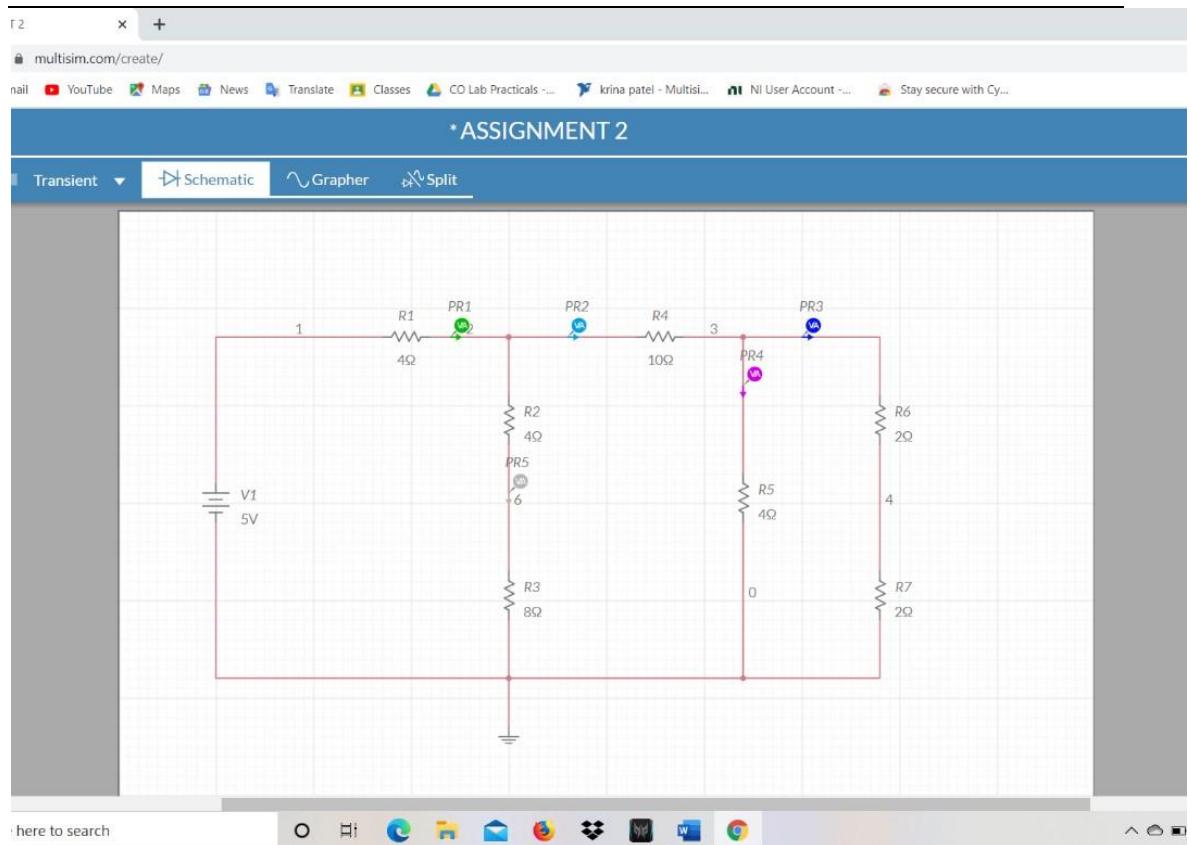
Result:**For Voltage:**

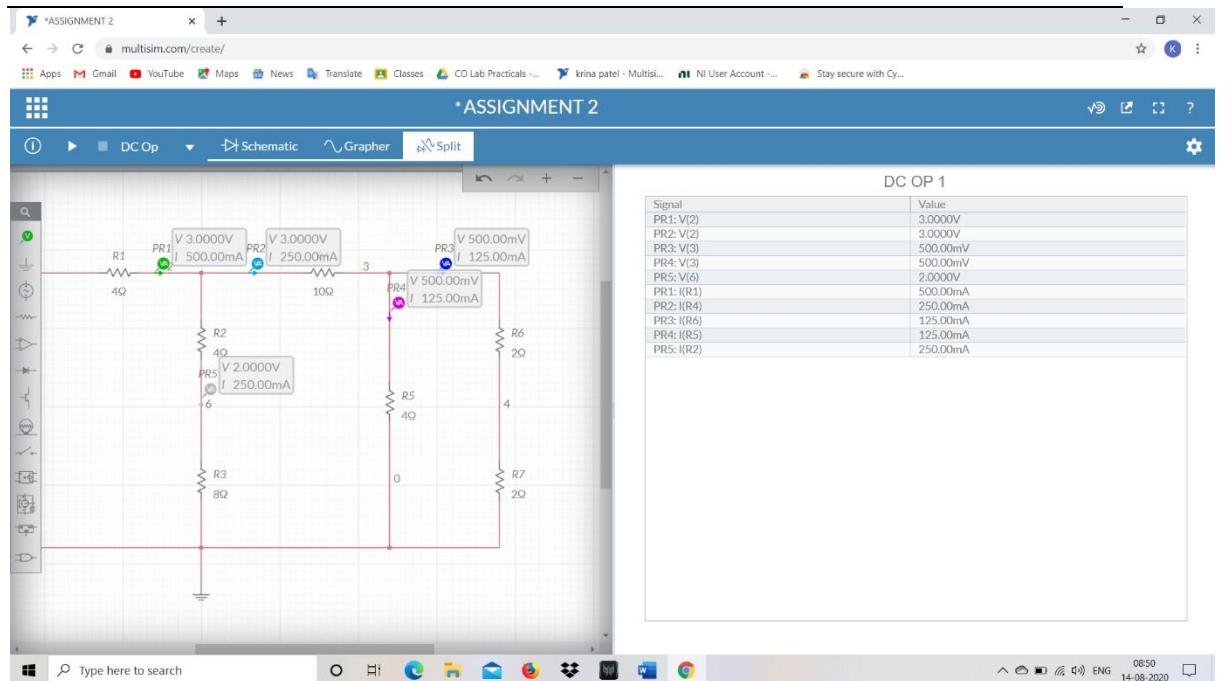
	MULTISIM	THEORETICAL
R1	8.6275 V	8.64 V
R2	8.6275 V	8.64 V
R3	15.373 V	15.4 V
R4	15.373 V	15.4 V

For Current:

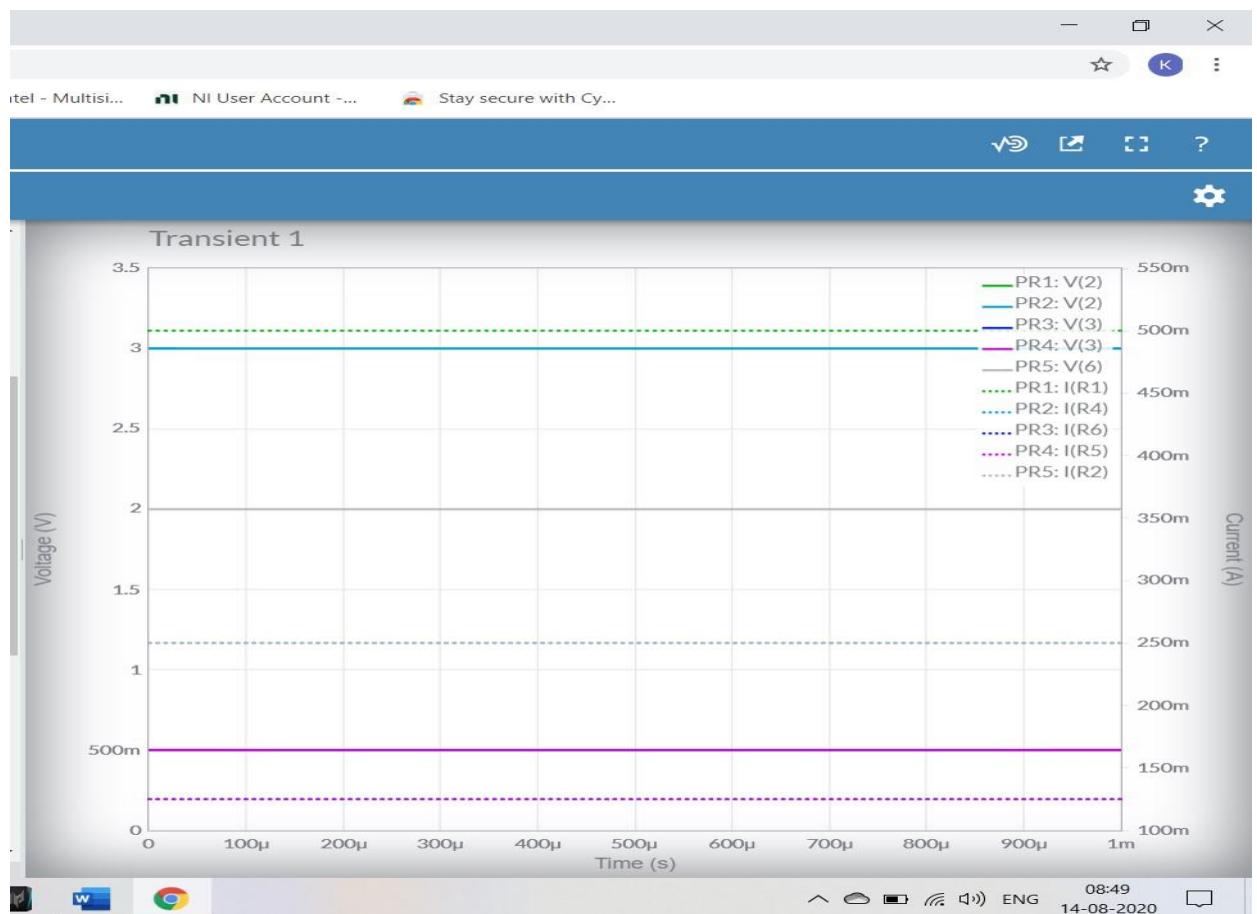
CURRENT	MULTISIM	THEORETICAL
R1	0.08628 A	0.0864 A
R2	0.0345 A	0.0356 A
R3	0.044 A	0.0445 A
R4	0.077 A	0.0768 A

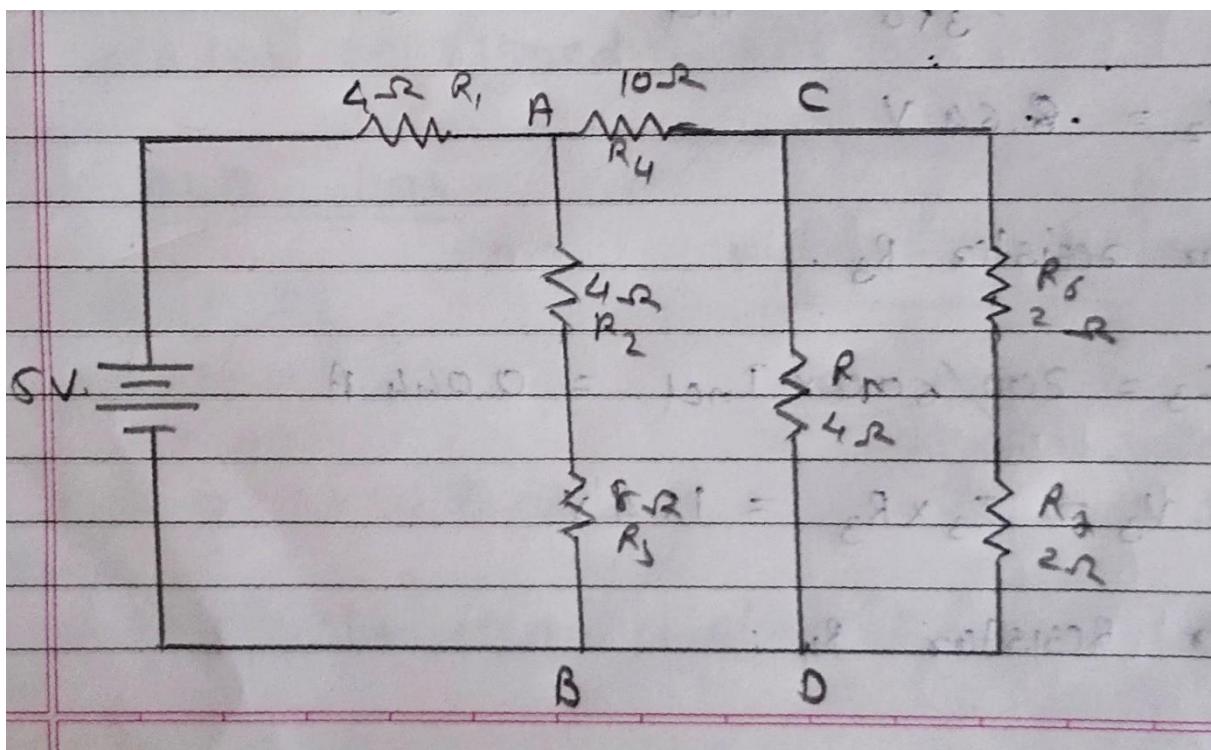
ASSIGNMENT 2**CIRCUIT DIADRAM:**





Graph:



**Calculation:**



Assignment - 2

Good Luck	Page No.
Date	

→ R_6 series with R_7

$$R = 4 \Omega$$

$$R_{CD} = (4 \parallel 4) = 2 \Omega$$

→ R_4 and R_{CD} are in series $\Rightarrow R = 12 \Omega$.

$$R_{AB} = (4 + 8) \parallel R = 6 \Omega$$

$$R_{eq} = 4 + 6 = 10 \Omega$$

~~$R_{eq} = 4 \Omega$~~

$$I_{eq} = \frac{V}{10} = 0.5 A = 500 \text{ mA} \quad \textcircled{1}$$

using eq. \textcircled{1}

$$I_{R_1} = 500 \text{ mA}$$

$$I_{R_4} = I_{R_2} = I_{R_3} = \frac{12}{24} \times 500 = 250 \text{ mA}$$

$$I_{R_F} = \frac{4}{8} \times 250 = 125 \text{ mA}$$

$$I_{R_7} = I_{R_8} = \frac{4}{8} \times 250 = 125 \text{ mA}$$

So,

$$V_{R_1} = I_{R_1} R_1 = 2 \text{ V}$$

$$V_{R_2} = I_{R_2} R_2 = 250 \times 4 = 1 \text{ V}$$

$$V_{R_3} = I_{R_3} R_3 = 2 \text{ V}$$

$$V_{R_4} = I_{R_4} R_4 = 2.5 \text{ V}$$

$$V_{R_F} = I_{R_F} R_F = 125 \times 4 = 0.5 \text{ V}$$

$$V_{R_6} = I_{R_6} R_6 = 0.25 \text{ V}$$

$$V_{R_7} = I_{R_7} R_7 = 0.25 \text{ V}$$

**Result:****For Voltage:**

	Multisim (V)	Theoretical (V)
R1	2	2
R2	1	1
R3	2	2
R4	2.5	2.5
R5	0.5	0.5
R6	0.25	0.25
R7	0.25	0.25

For Current:

	Multisim (mA)	Theoretical (mA)
R1	500	500
R2	250	250
R3	250	250
R4	250	150
R5	125	125
R6	125	125
R7	125	125



Expt. No:

2

Date:

21-08-2020**Study of Basic and Universal Gates**

AIM: To verify the truth table of basic gates. Also implement all the basic gates using Universal (NAND & NOR) gates.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator
2. Basic Gates (AND, OR and NOT)
3. Universal Gates (NAND and NOR)

THEORY:

A **Digital Logic Gate** is an electronic circuit which makes logical decisions based on the combination of digital signals present on its inputs. Digital logic gates can have more than one input, for example, inputs A, B, C, D etc., but generally only have one digital output, (Q). Individual logic gates can be connected or cascaded together to form a logic gate function with any desired number of inputs, or to form combinational and sequential type circuits, or to produce different logic gate functions from standard gates. Standard commercially available digital logic gates are available in two basic families or forms, **TTL** which stands for *Transistor-Transistor Logic* such as the 7400 series, and **CMOS** which stands for *Complementary Metal-Oxide-Silicon* which is the 4000 series of chips. This notation of TTL or CMOS refers to the logic technology used to manufacture the integrated circuit, (IC) or a “chip” as it is more commonly called.

The **Digital Logic Gate** is the basic building block from which all digital electronic circuits and microprocessor based systems are constructed. Basic digital logic gates perform logical operations of AND, OR and NOT on binary numbers. In digital logic design only two voltage levels or states are allowed and these states are generally referred to as Logic “1” and Logic “0”, or HIGH and LOW, or TRUE and FALSE. These two states are represented in Boolean Algebra and standard truth tables by the binary digits of “1” and “0” respectively. A good example of a digital state is a simple light switch. The switch can be either “ON” or “OFF”, one state or the other, but not both at the same time.

Most *digital logic gates* and digital logic systems use “Positive logic”, in which a logic level “0” or “LOW” is represented by a zero voltage, 0v or ground and a logic level “1” or “HIGH” is represented by a higher voltage such as +5 volts, with the switching from one voltage level to the other, from either a logic level “0” to a “1” or a “1” to a “0” being made as quickly as possible to prevent any faulty operation of the logic circuit. There also exists a complementary “Negative Logic” system in which the values and the rules of logic “0” and a logic “1” are reversed.



AND GATE: The Logic AND Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when all of its inputs are HIGH. The output state of a digital logic AND gate only returns “LOW” again when **ANY** of its inputs are at a logic level “0”. In other words for a logic AND gate, any LOW input will give a LOW output.

Symbol	Truth Table		
	A	B	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1

OR GATE:

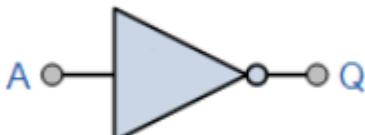
The Logic OR Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when one or more of its inputs are HIGH. The output, Q of a “Logic OR Gate” only returns “LOW” again when **ALL** of its inputs are at a logic level “0”. In other words for a logic OR gate, any “HIGH” input will give a “HIGH”, logic level “1” output.

Symbol	Truth Table		
	A	B	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1

NOT GATE:

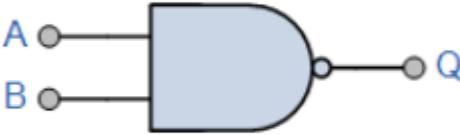


The Logic NOT Gate is the most basic of all the logical gates and is often referred to as an Inverting Buffer or simply an Inverter. Inverting NOT gates are single input devices which have an output level that is normally at logic level “1” and goes “LOW” to a logic level “0” when its single input is at logic level “1”, in other words it “inverts” (complements) its input signal. The output from a NOT gate only returns “HIGH” again when its input is at logic level “0”.

Symbol	Truth Table	
	A	Q
	0	1
	1	0

NAND GATE:

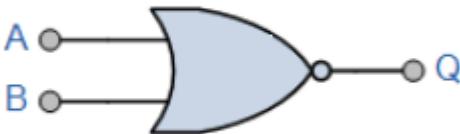
The Logic NAND Gate is a combination of a digital logic AND gate and a NOT gate connected together in series. The NAND (Not – AND) gate has an output that is normally at logic level “1” and only goes “LOW” to logic level “0” when **ALL** of its inputs are at logic level “1”.

Symbol	Truth Table		
	A	B	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0

NOR GATE:

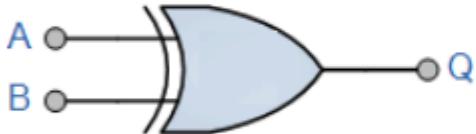
The Logic NOR Gate is a combination of the digital logic OR gate and an inverter or NOT gate connected together in series. The inclusive NOR (Not-OR) gate has an output that is normally at logic level “1” and only goes “LOW” to logic level “0” when **ANY** of its inputs are at logic level “1”.



Symbol	Truth Table		
	A	B	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	0

EX-OR GATE:

The output of an Exclusive-OR gate **ONLY** goes “HIGH” when its two input terminals are at “**DIFFERENT**” logic levels with respect to each other.

Symbol	Truth Table		
	A	B	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	0

EX-NOR GATE:

The Exclusive-NOR Gate function is a digital logic gate that is the reverse or complementary form of the Exclusive-OR function. This gate gives output “1” when its inputs are “*logically equal*” or “*equivalent*” to each other, which is why an **Exclusive-NOR** gate is sometimes called an **Equivalence Gate**.

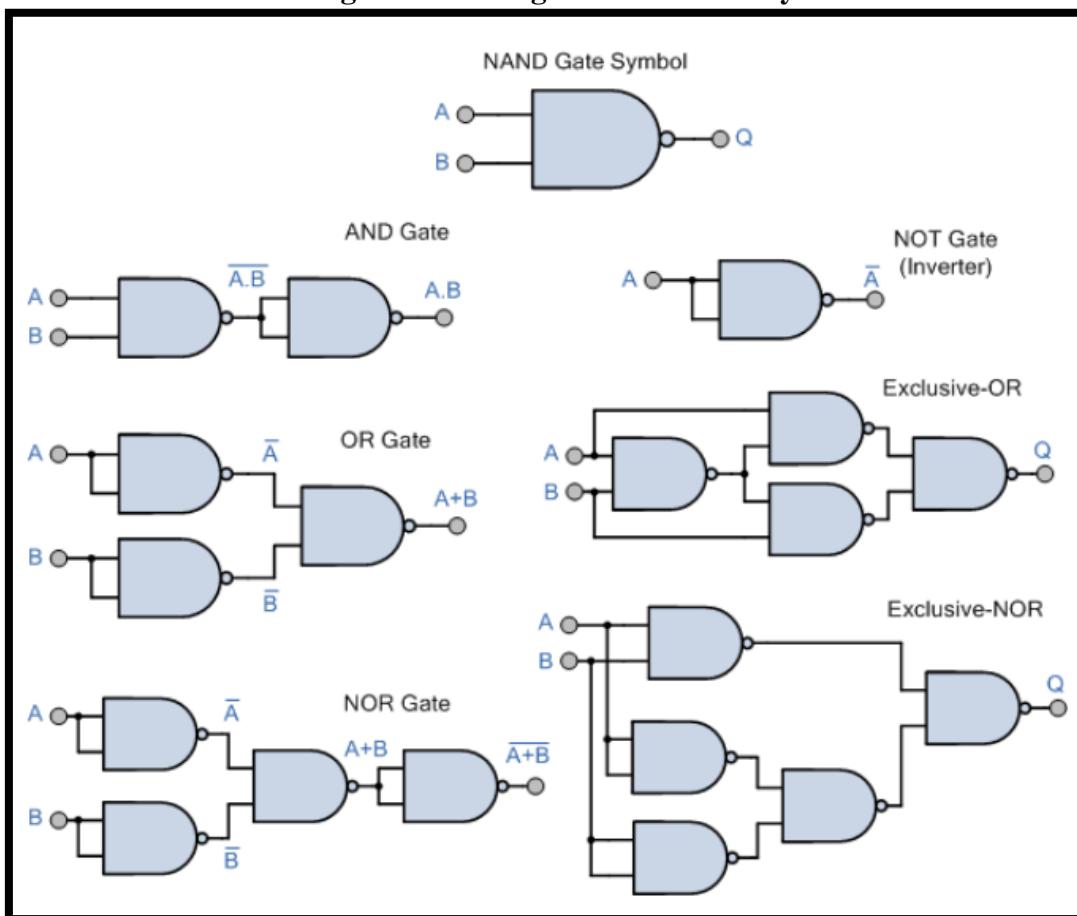


Symbol	Truth Table		
	A	B	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1

2-input Ex-NOR Gate

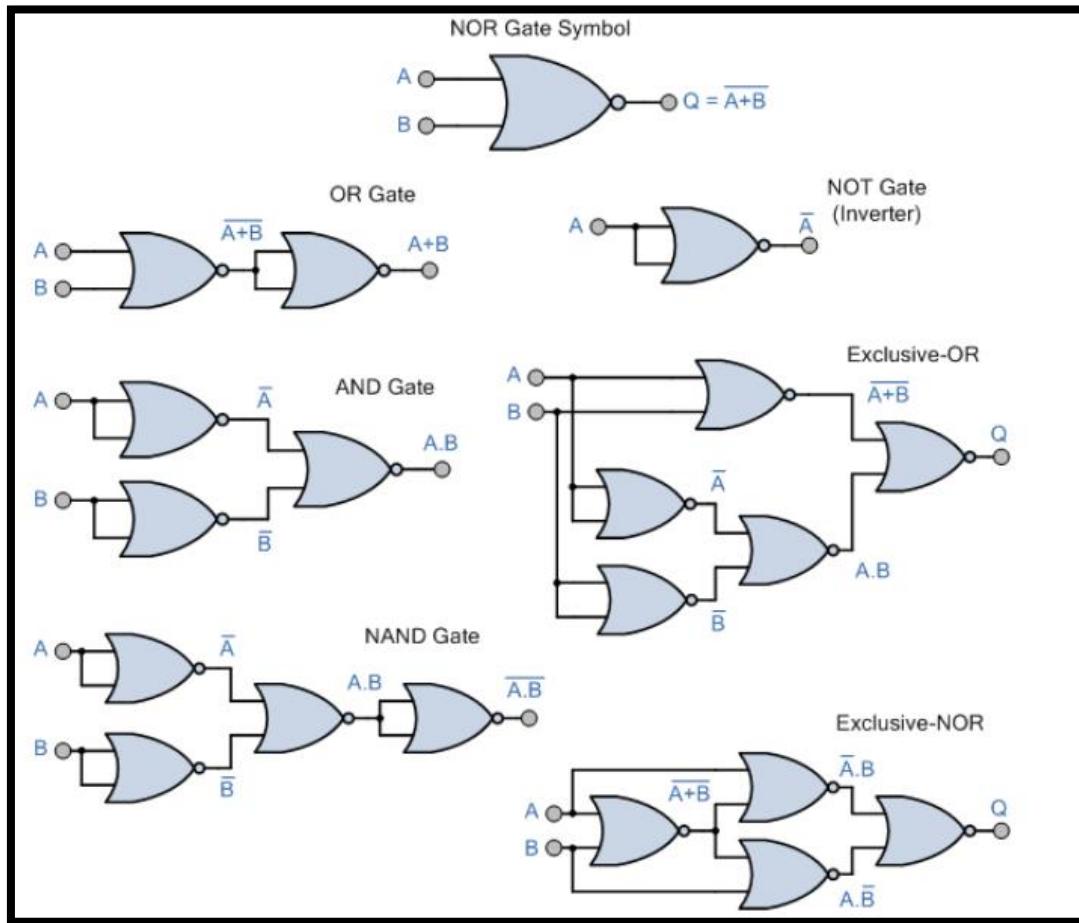
UNIVERSAL GATES:

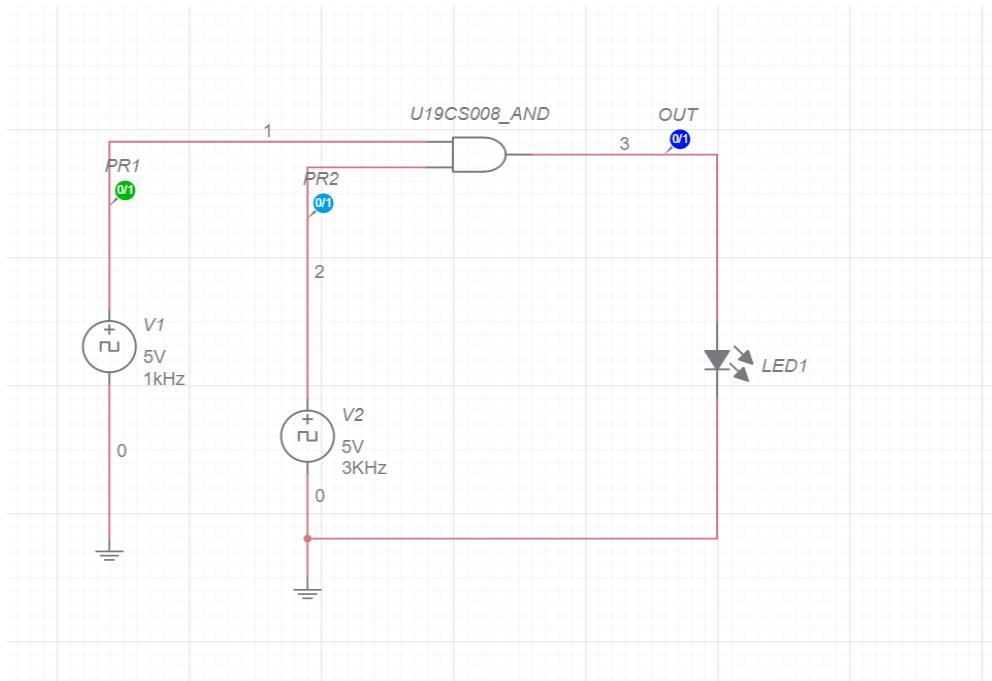
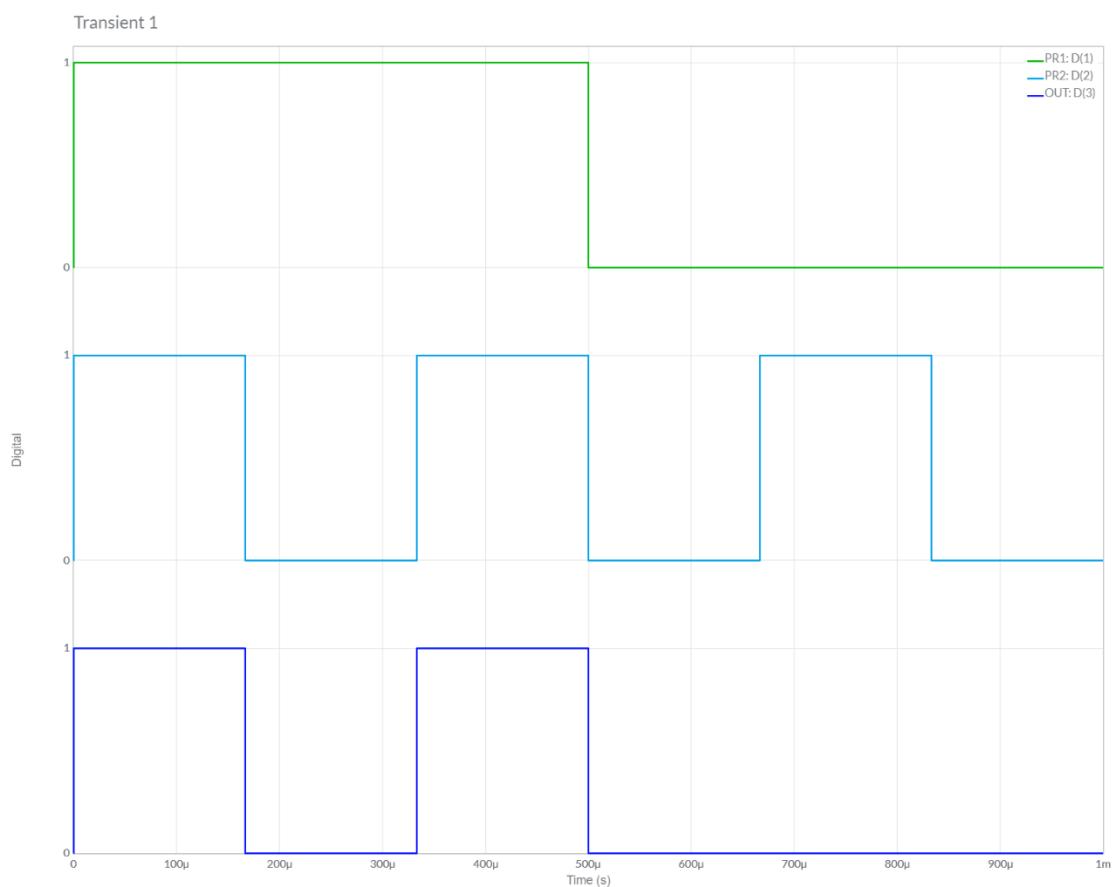
Universal Logic gates can be used to produce any other logic or Boolean function with the NAND and NOR gates. Thus ALL other logic gate functions can be created using only NAND/NOR gates making them universal logic gates.

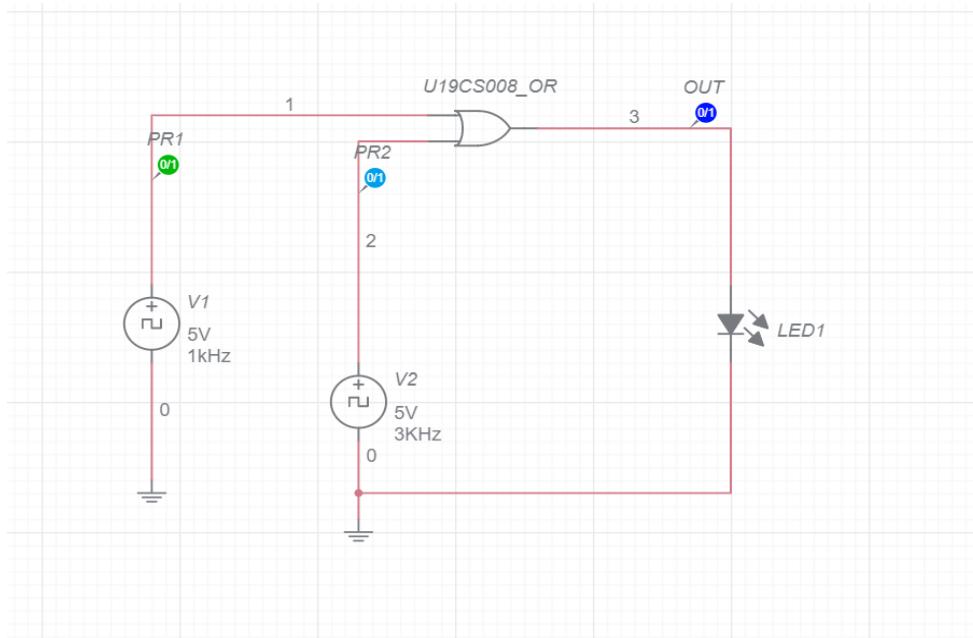
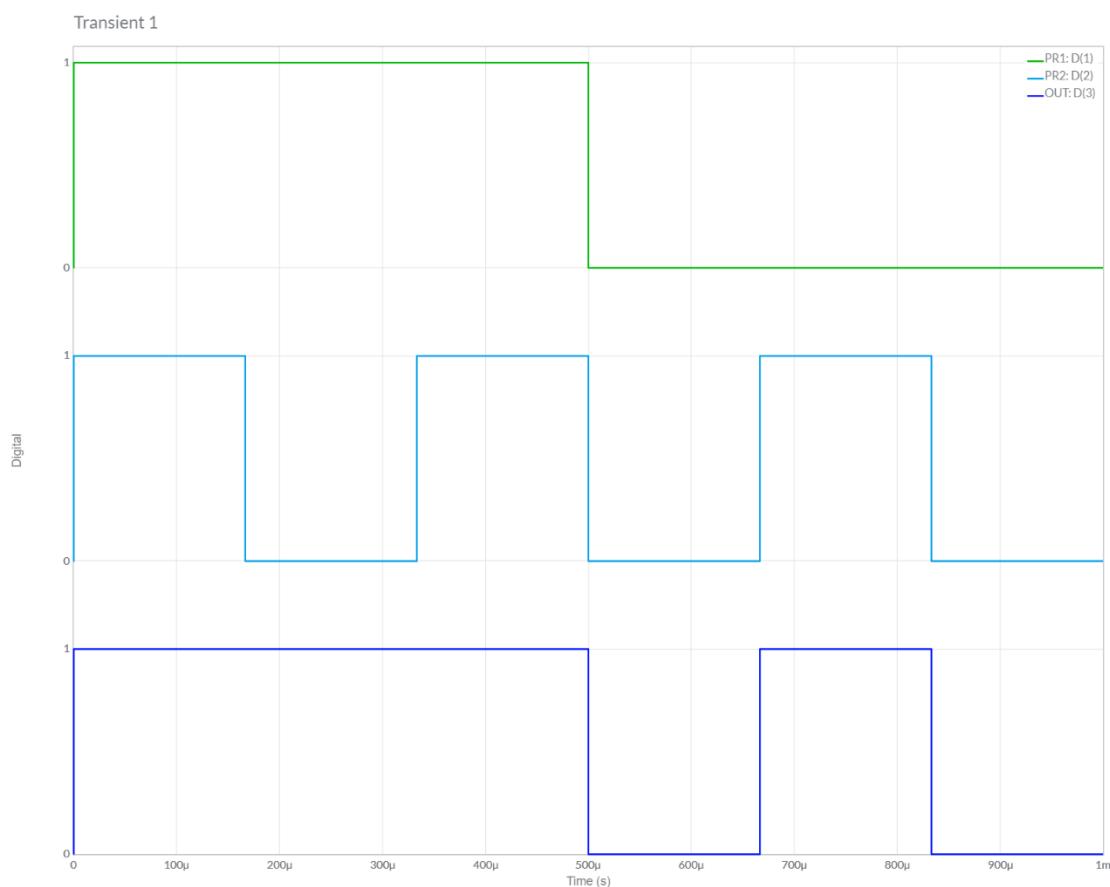
Logic Gates using NAND Gate Only:



Logic Gates using NOR Gate Only:



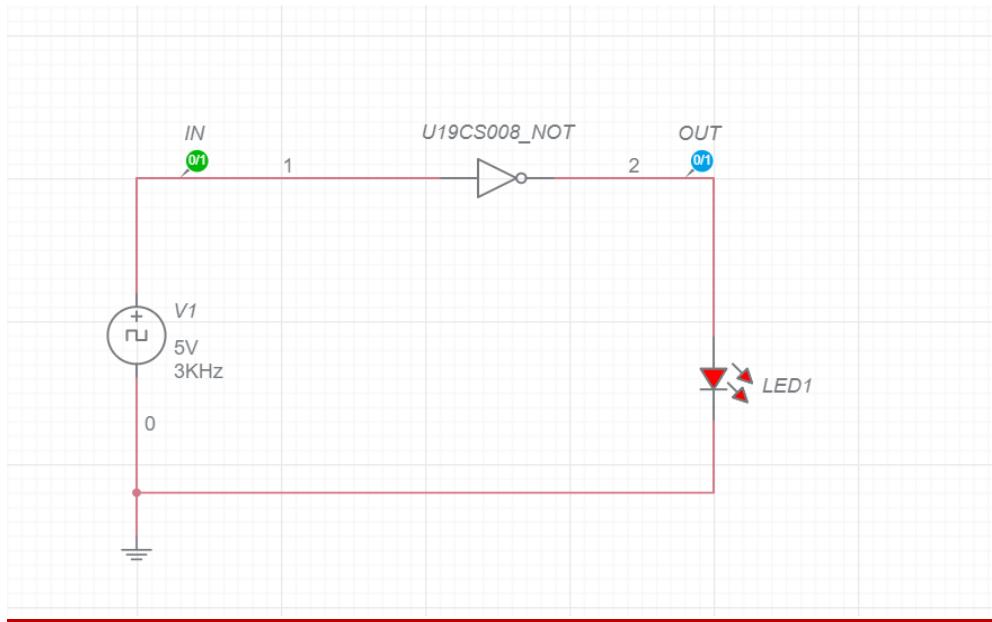
**AND GATE****CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)****OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):**

**OR GATE****CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)****OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):**

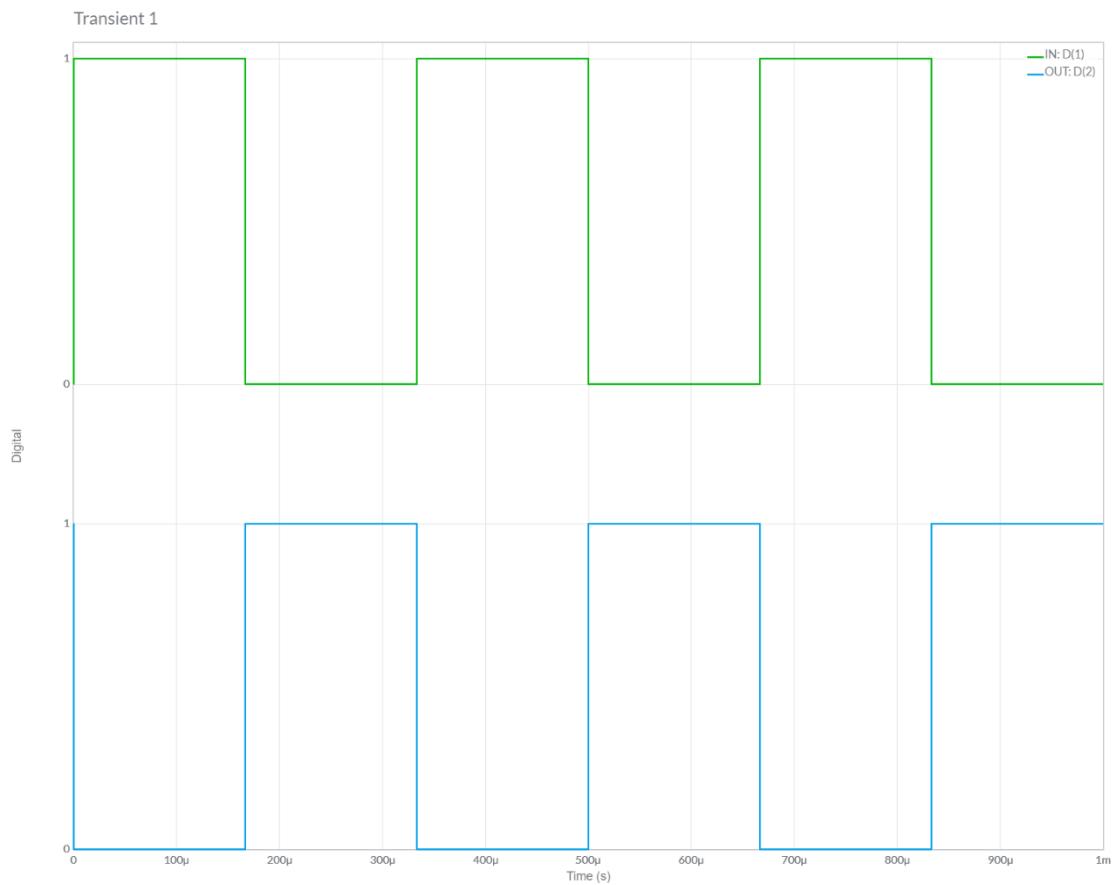


NOT GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

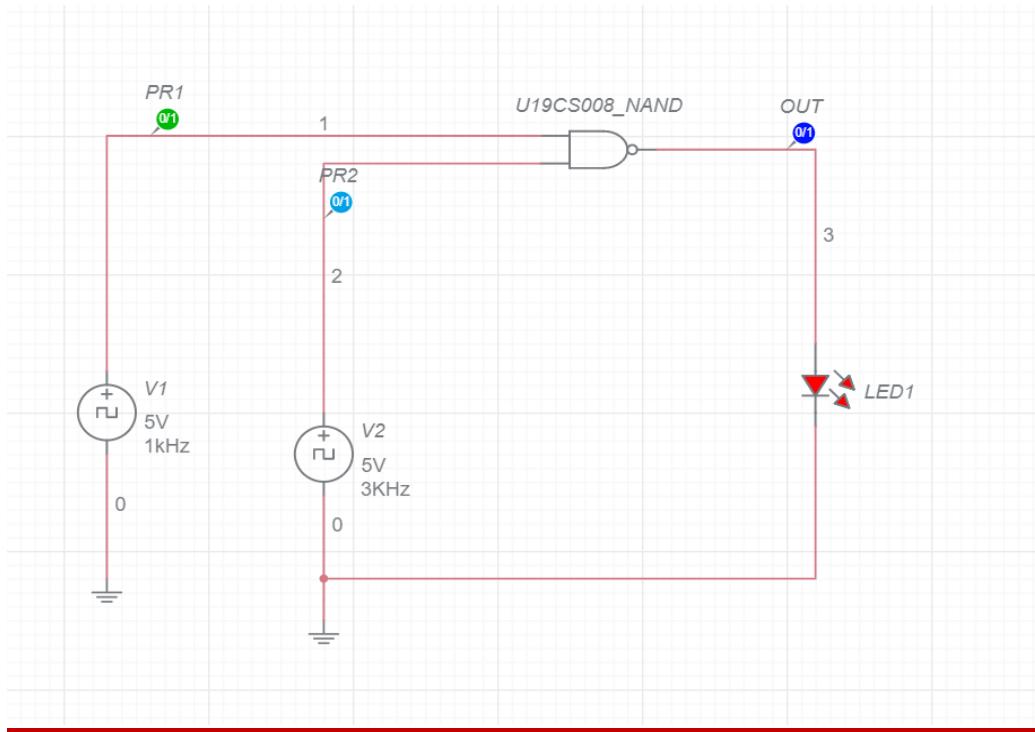


OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

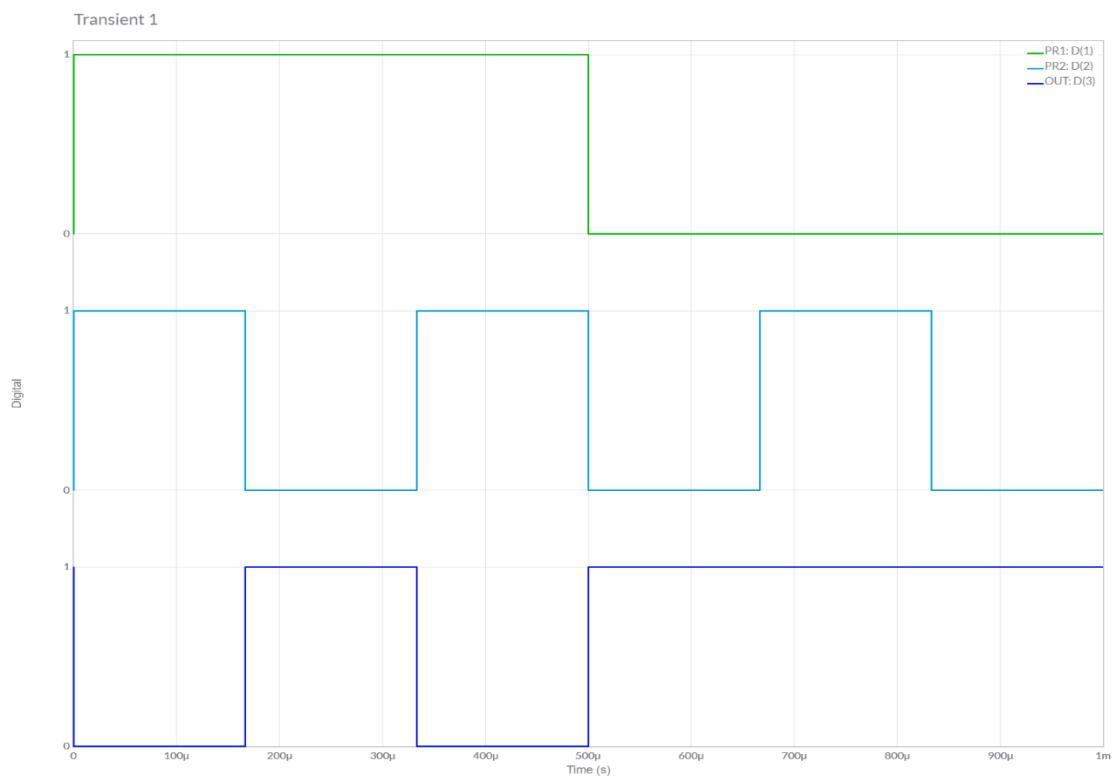


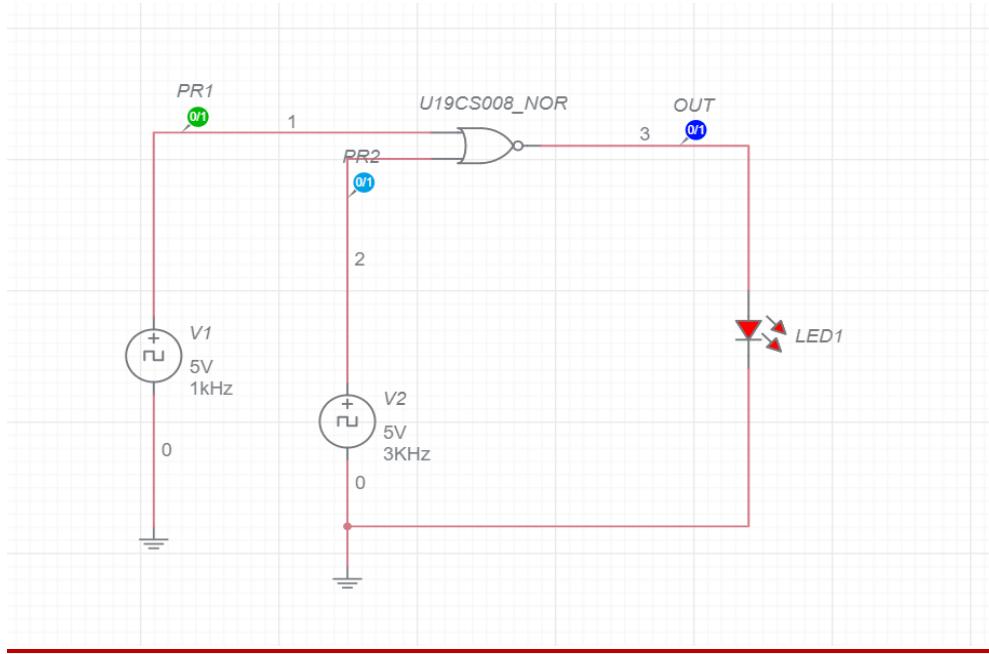
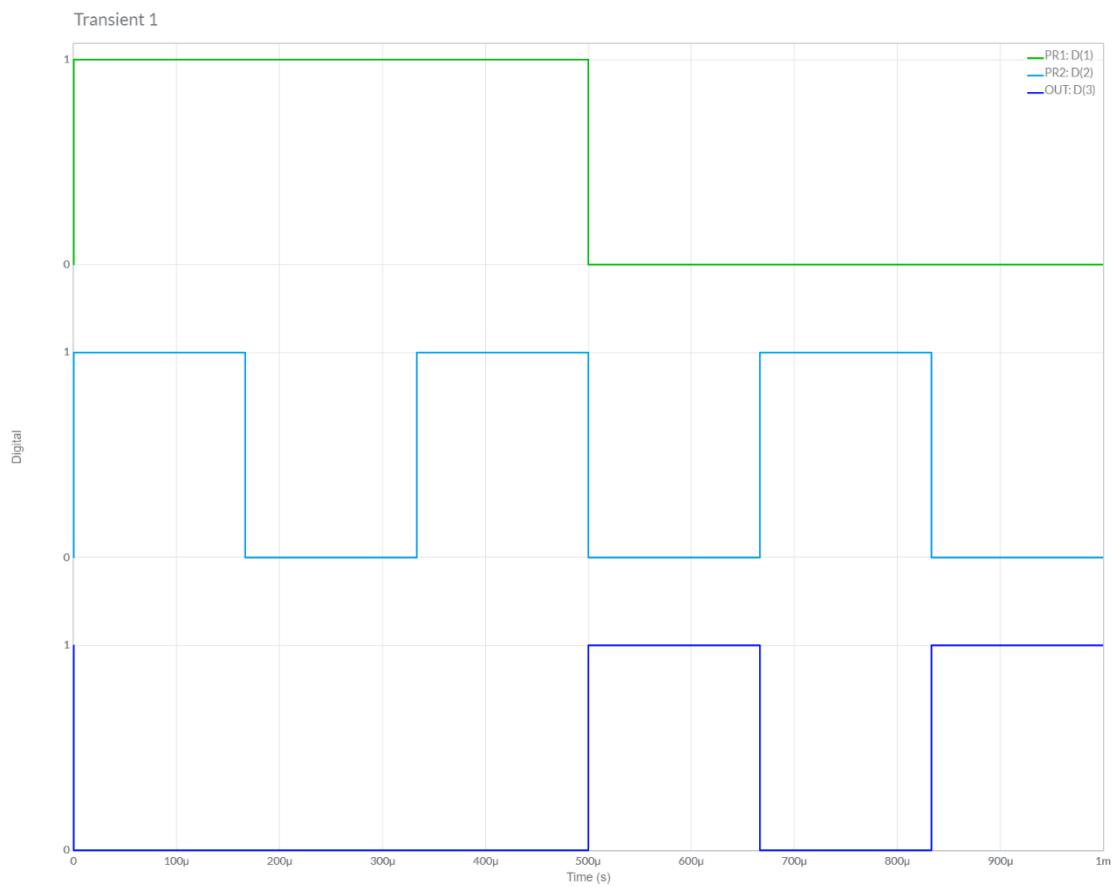
NAND GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



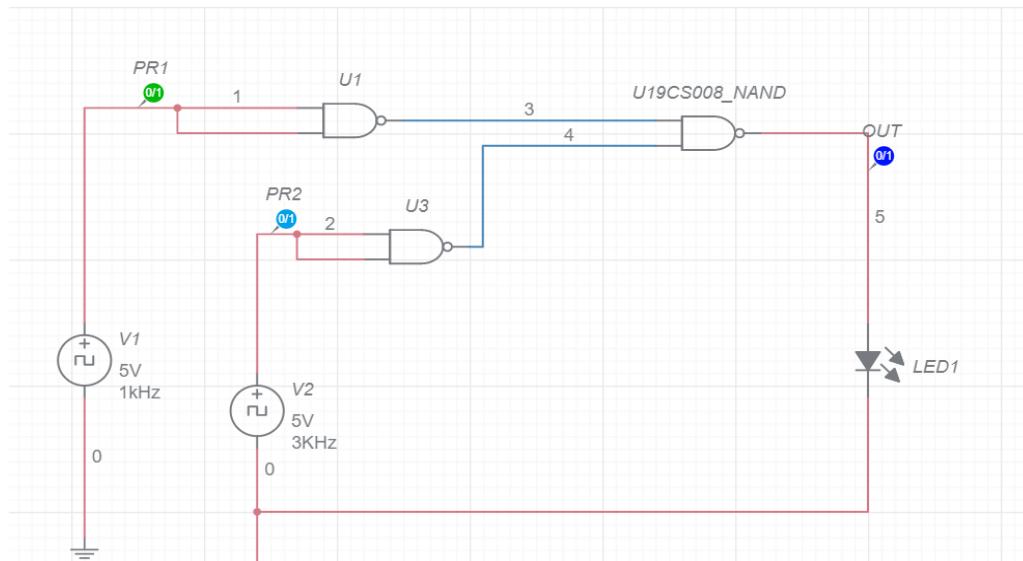
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



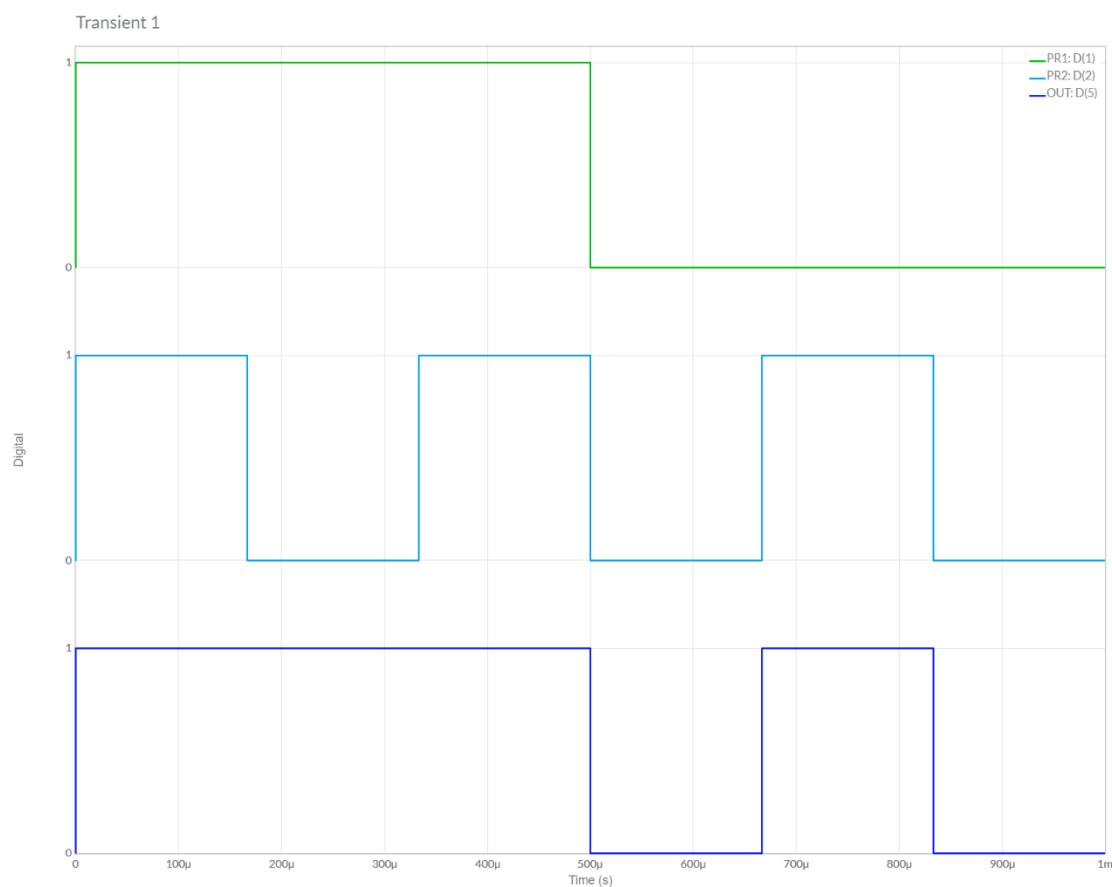
**NOR GATE****CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)****OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):**

OR GATE USING NAND

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



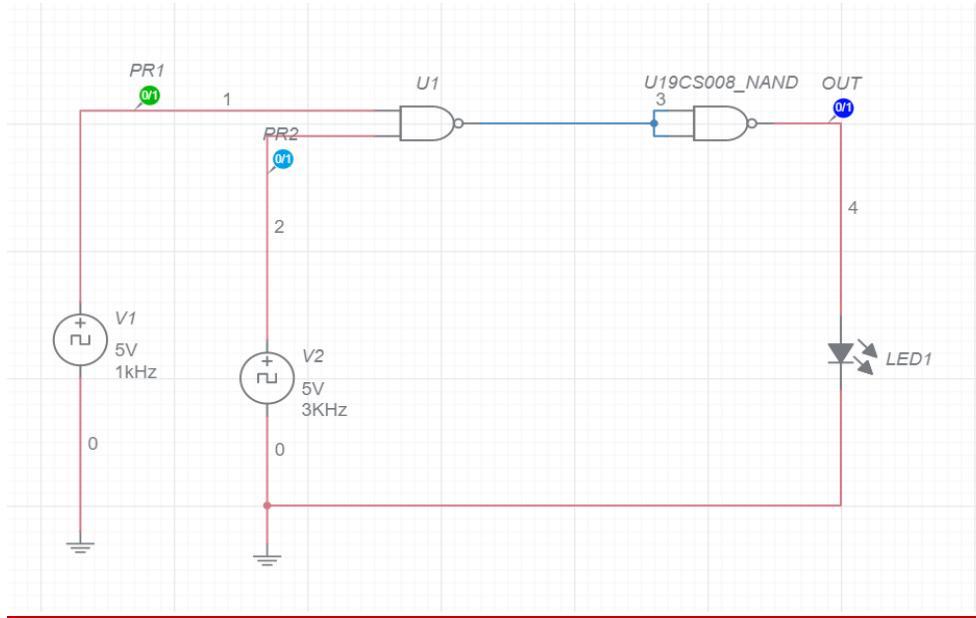
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



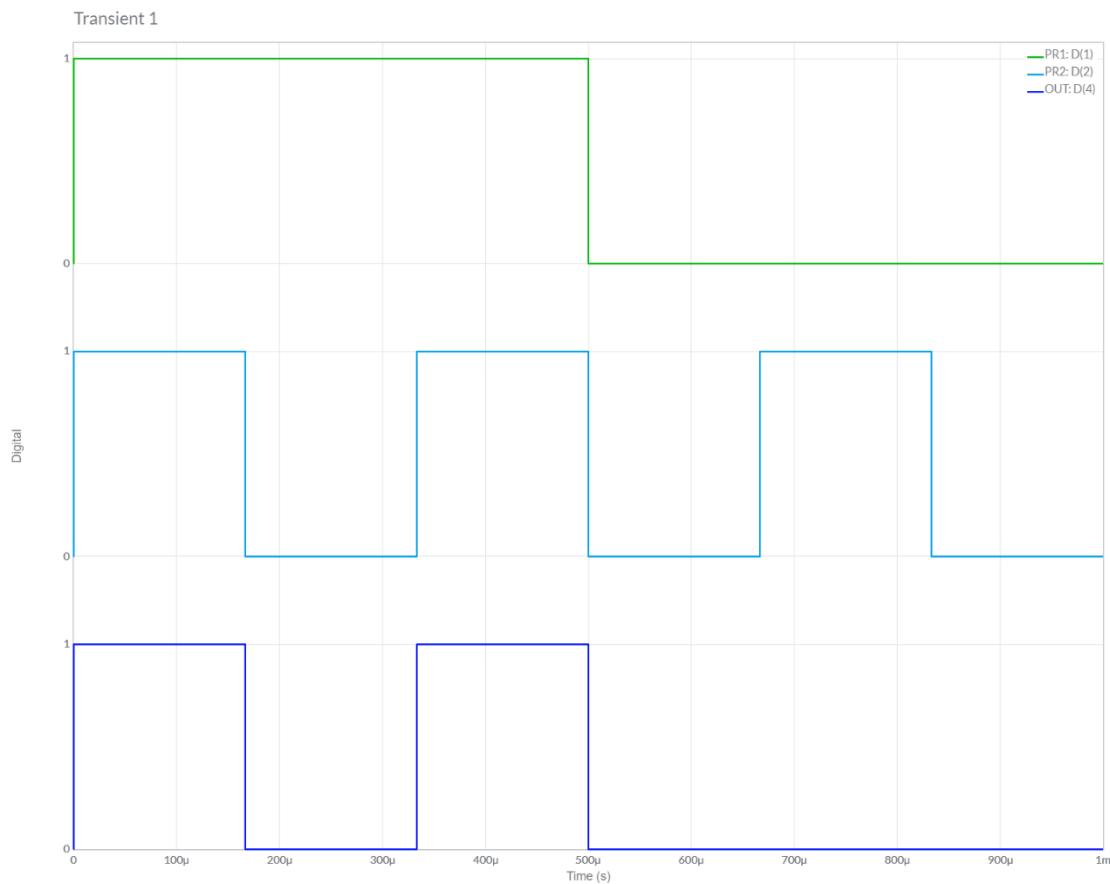


AND GATE USING NAND

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



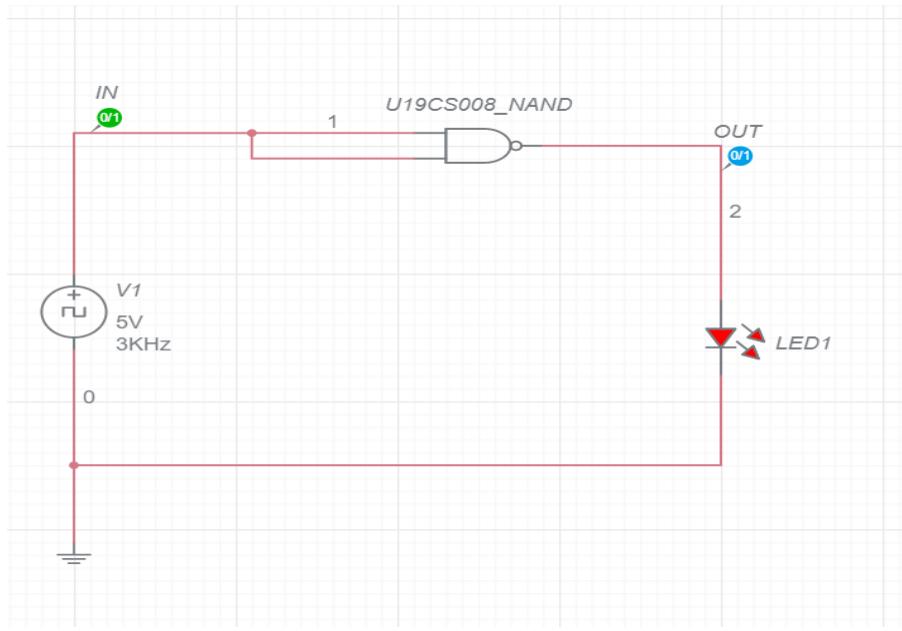
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



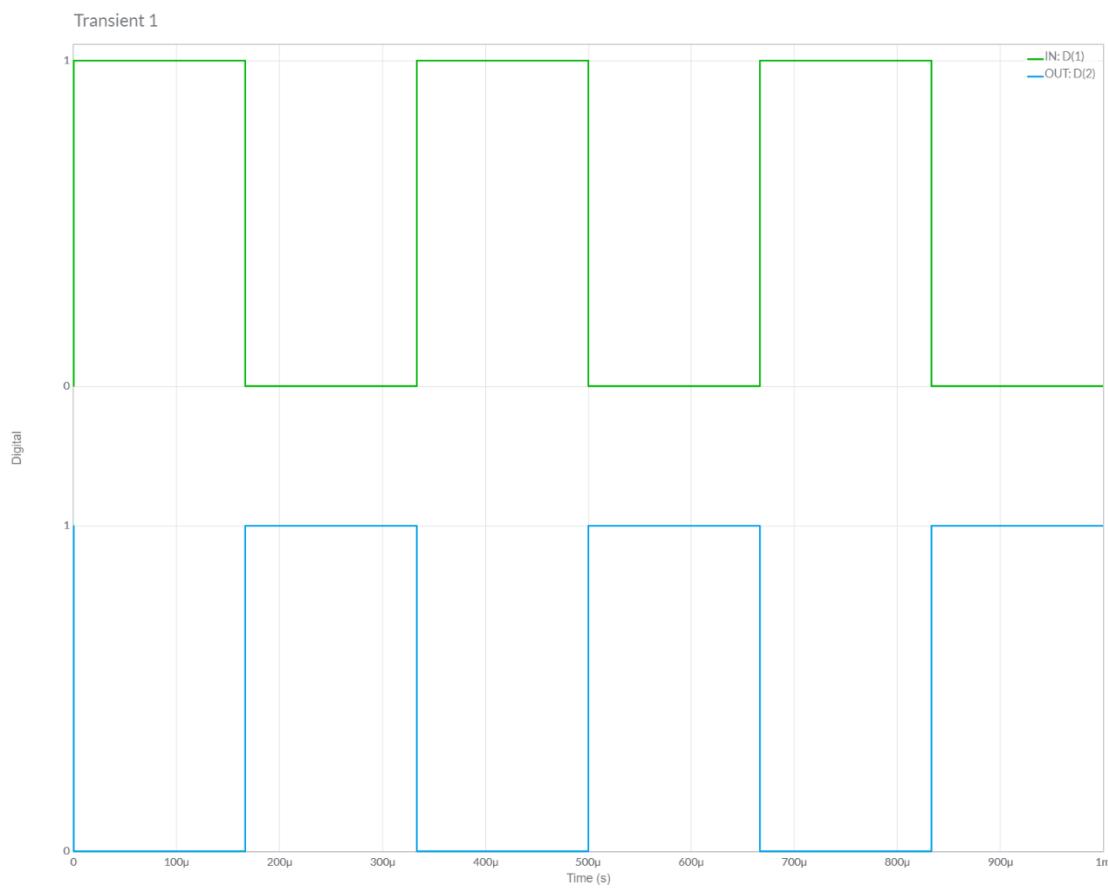


NOT GATE USING NAND

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



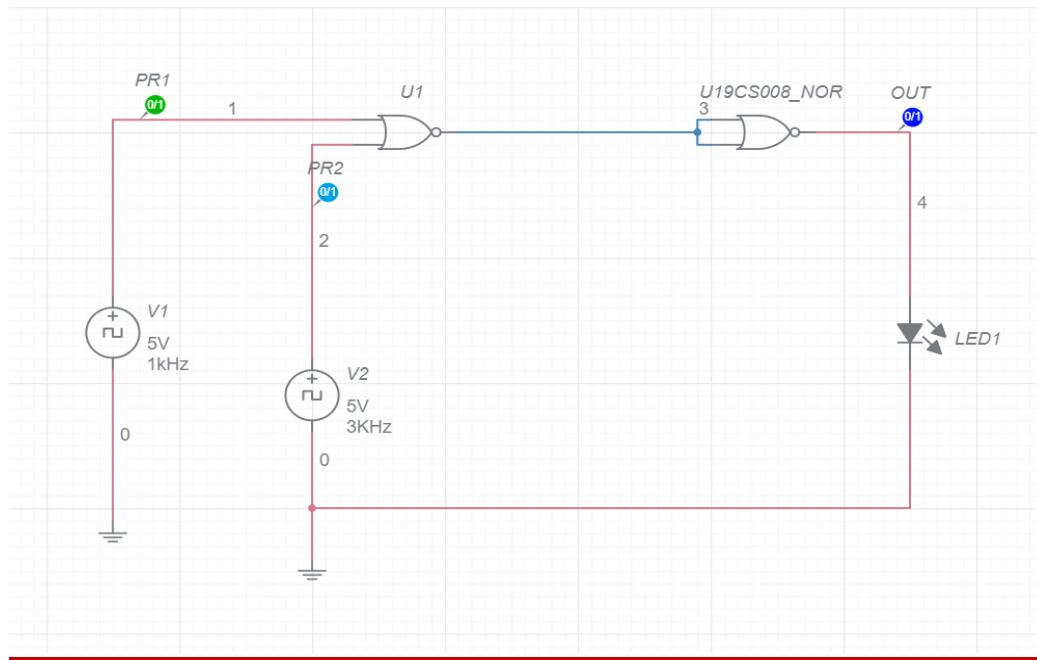
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



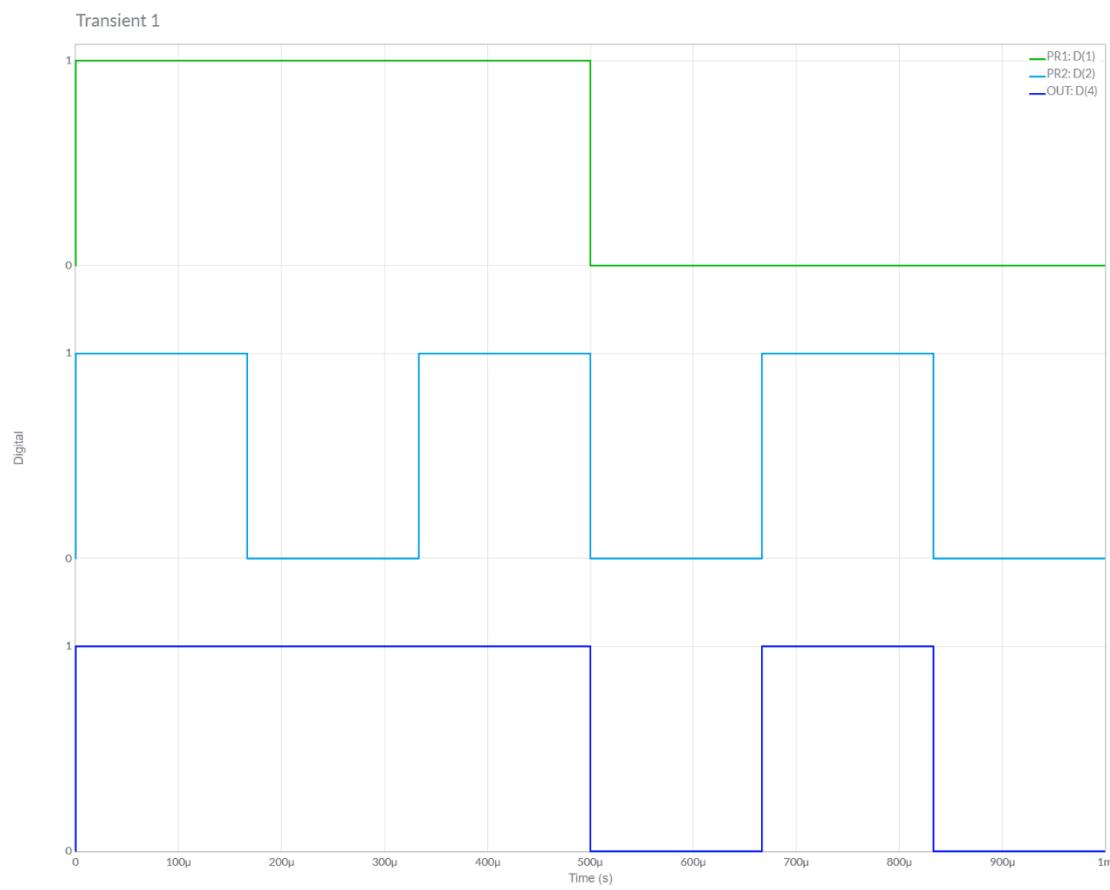


OR GATE USING NOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



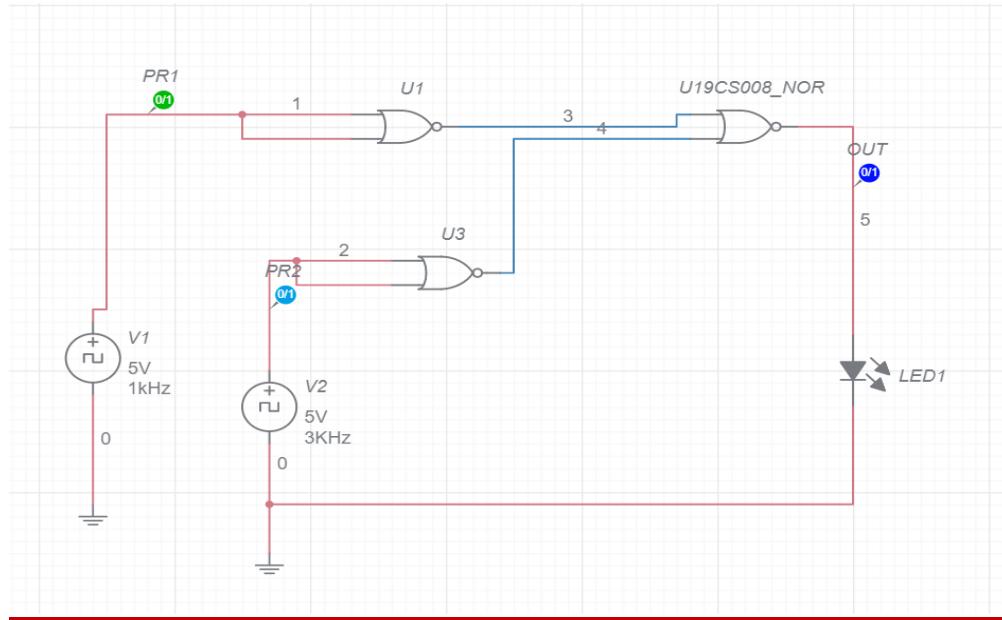
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



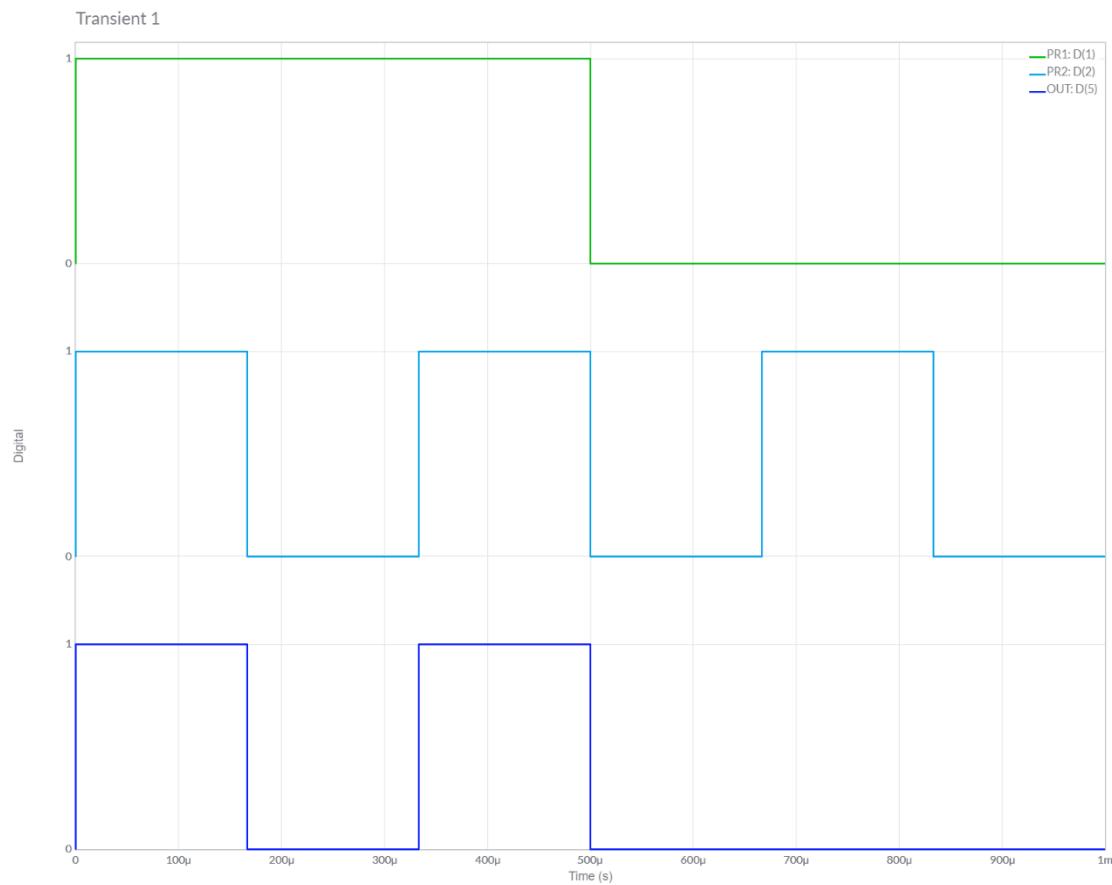


AND GATE USING NOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



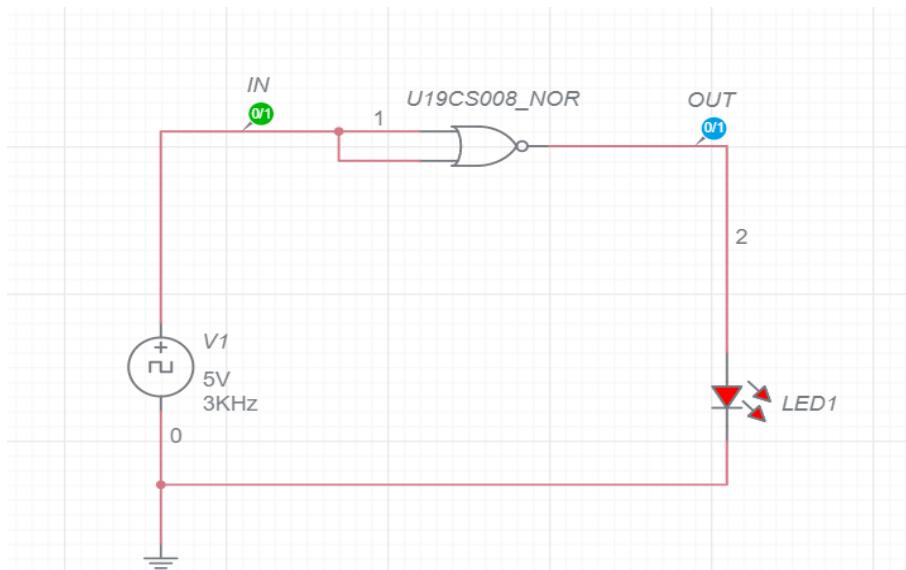
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



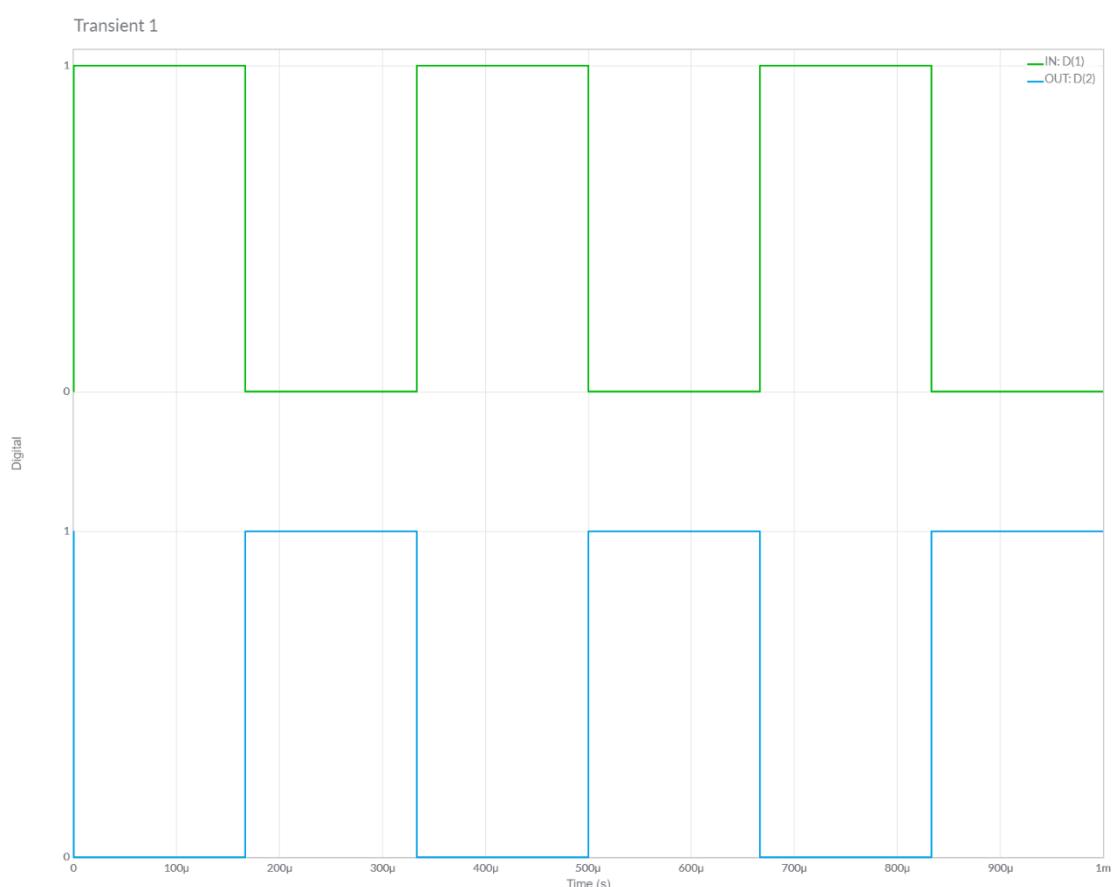


NOT GATE USING NOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



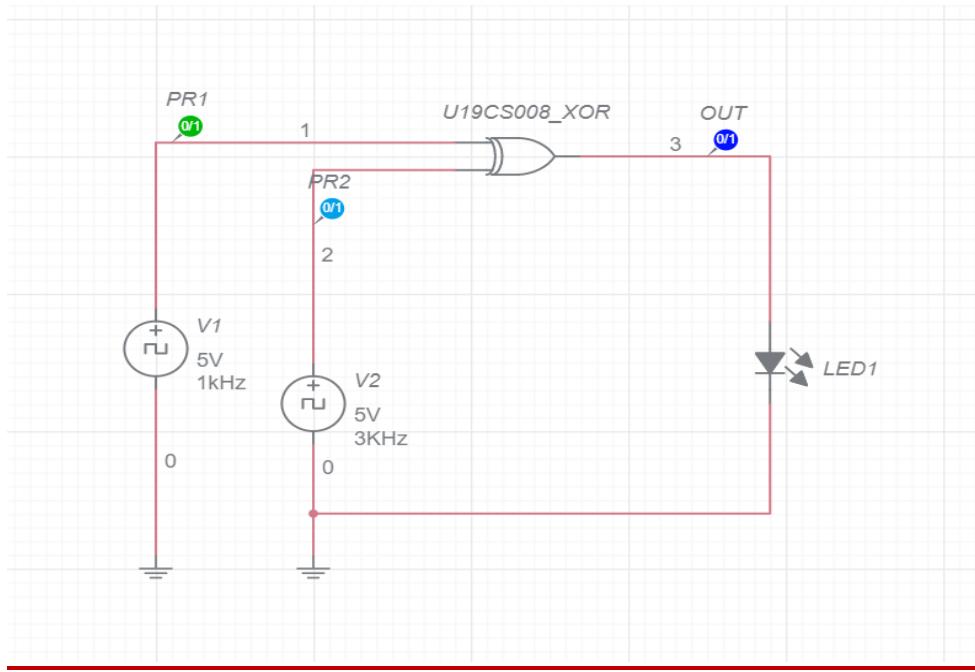
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



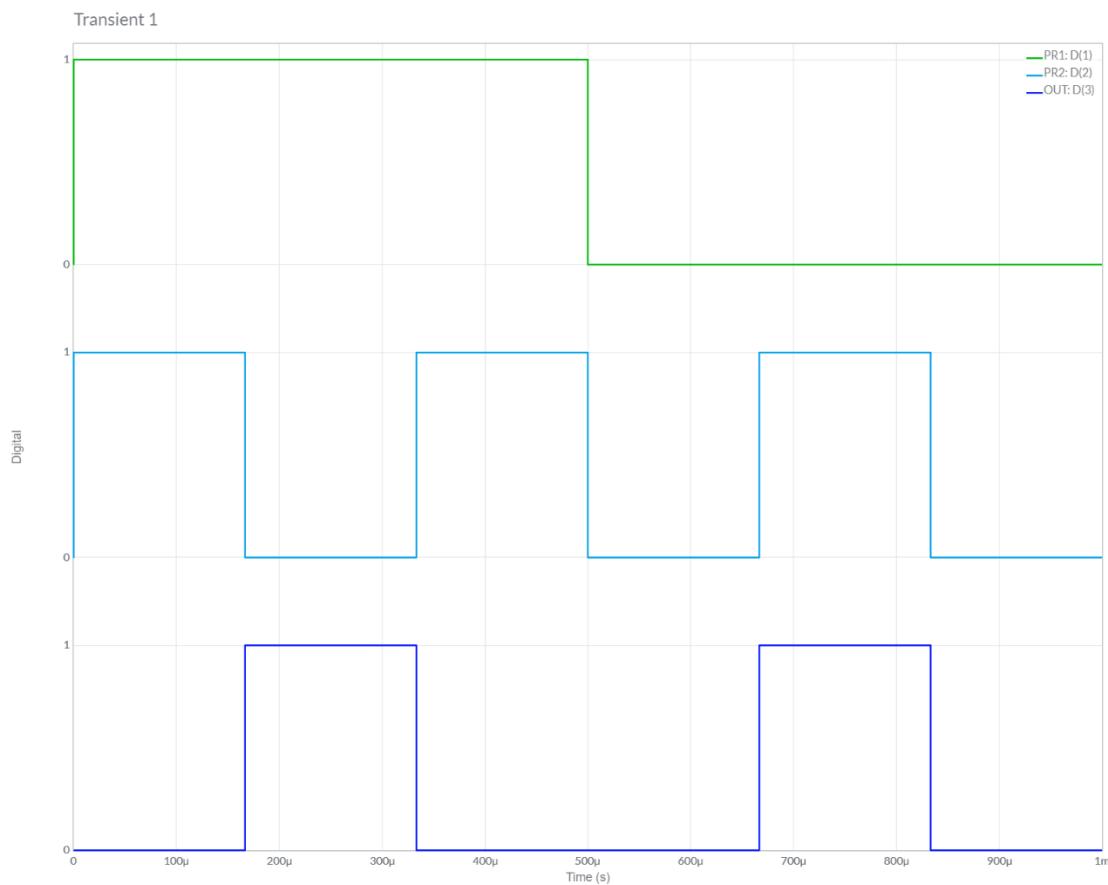


EX-OR GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



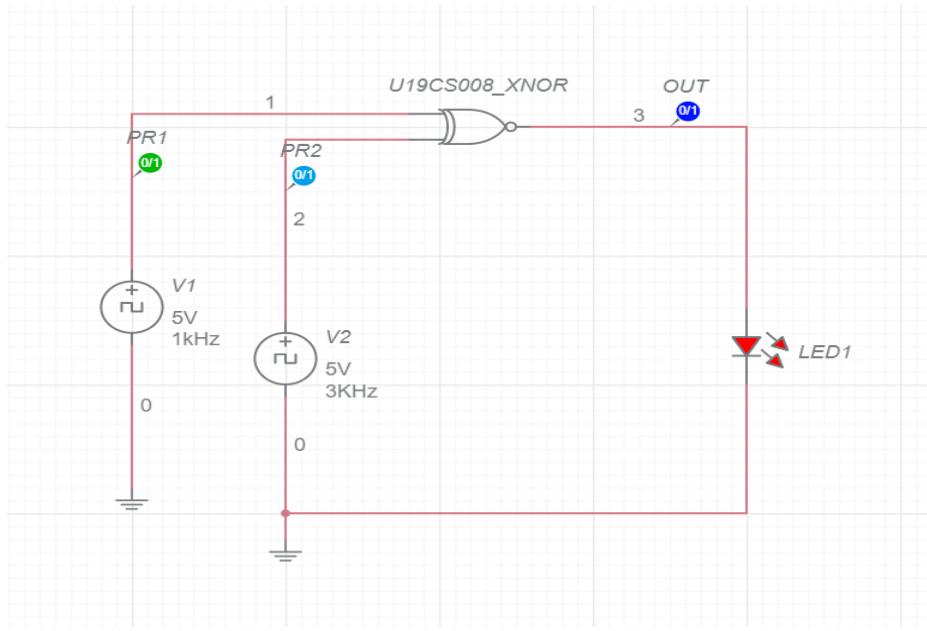
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



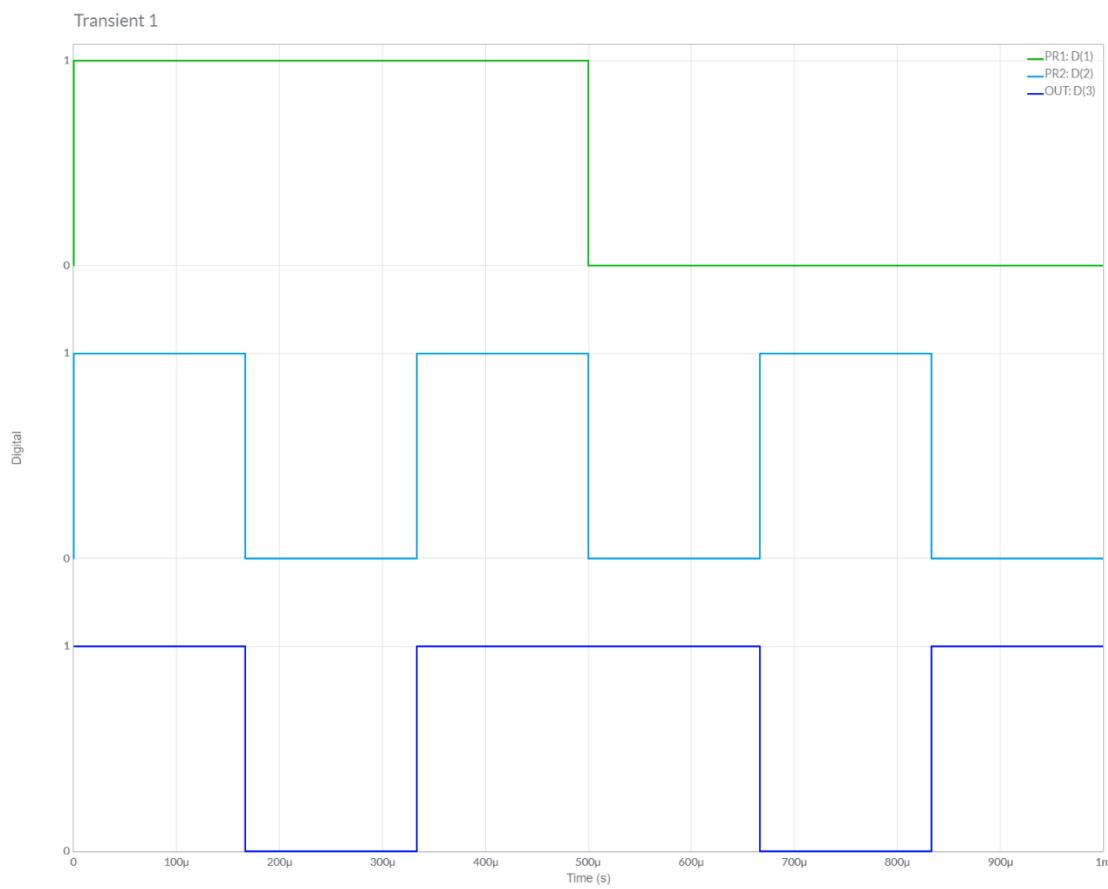


EX-NOR GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



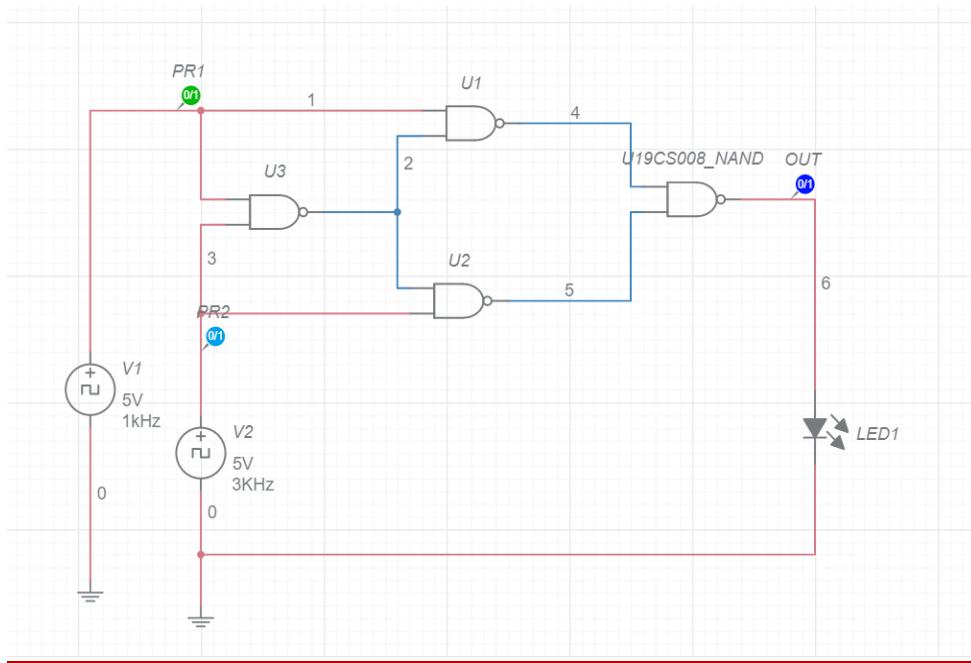
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



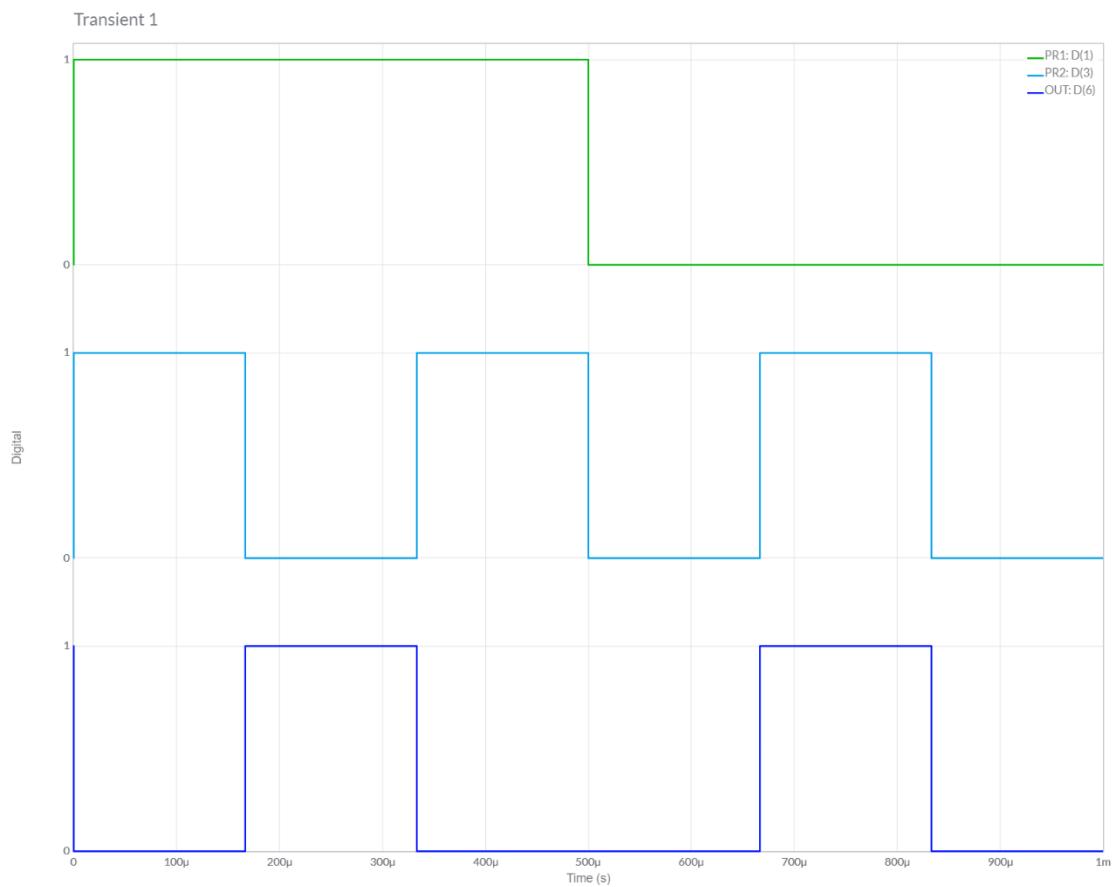


EX-OR GATE USING NAND

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



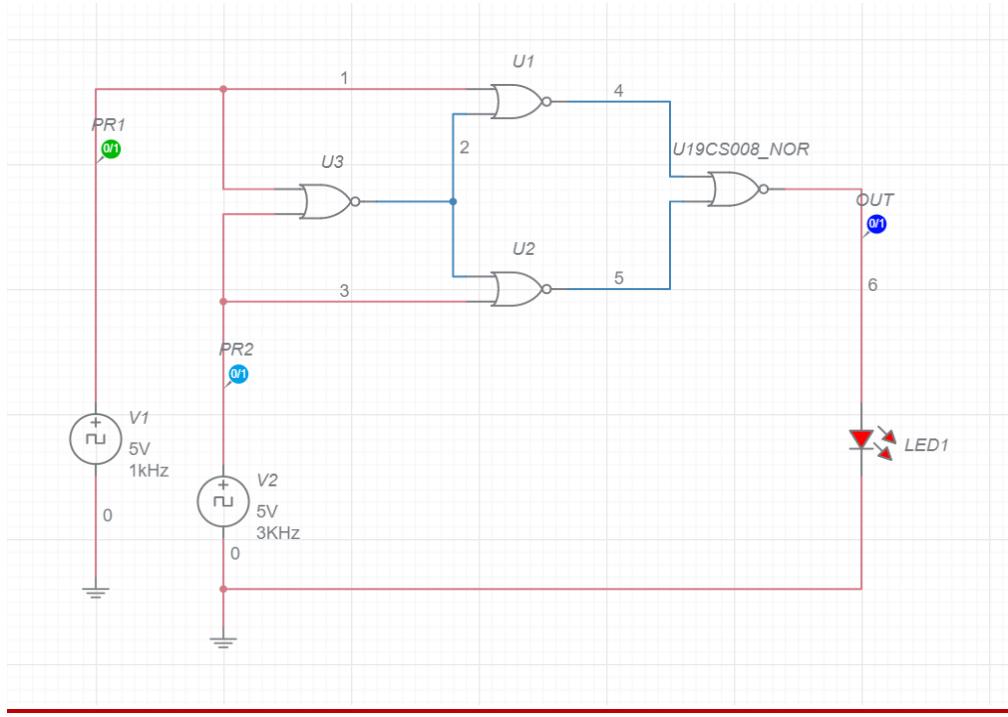
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



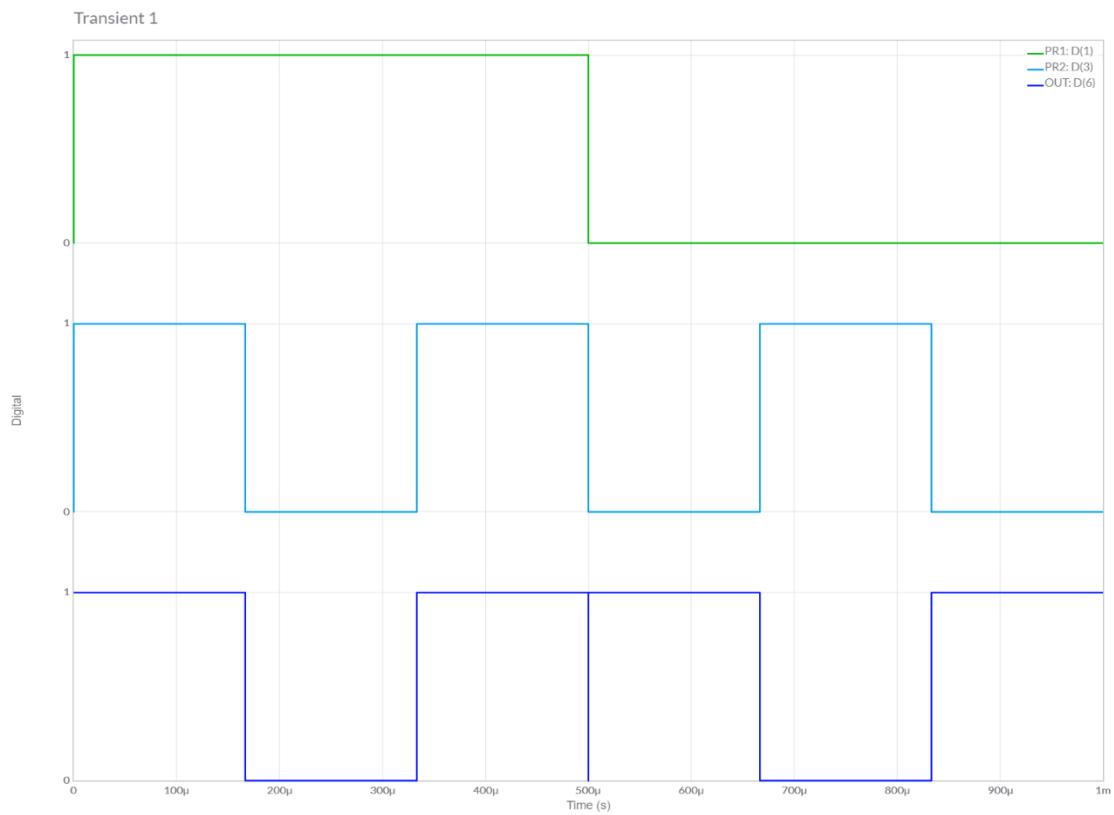


EX-NOR GATE USING NOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



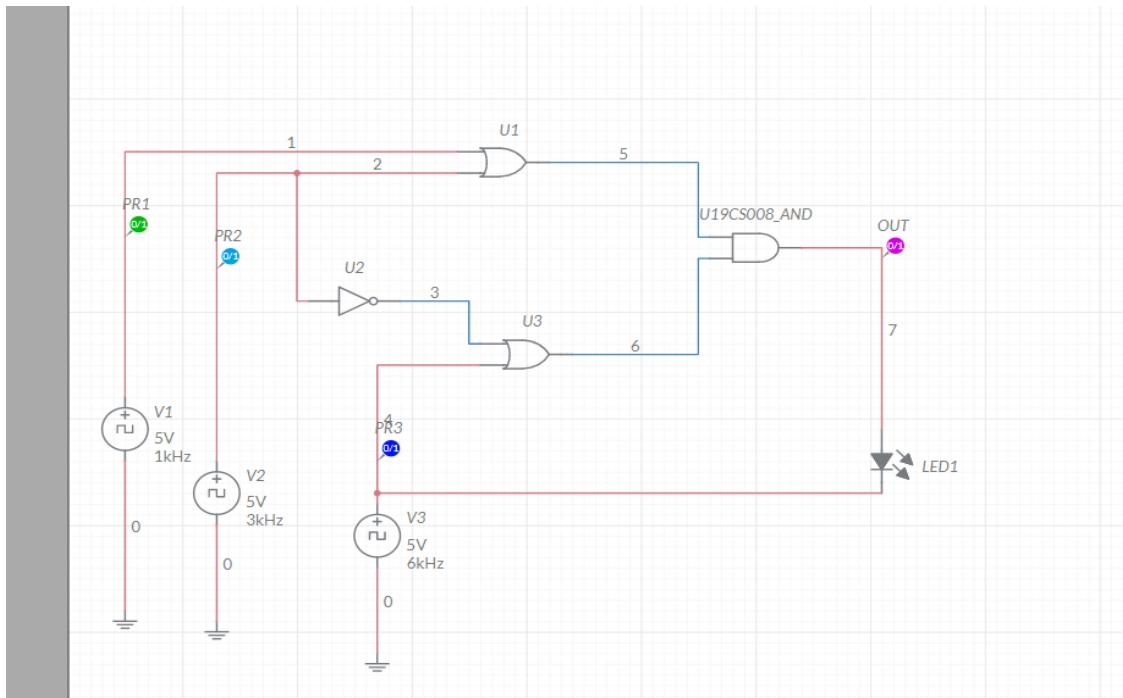


CONCLUSIONS

THE TRUTH TABLE IN THEORY AND THE SIMULATION OF THE TEST CIRCUIT ON MULTISIM LIVE BOTH ARE EQUAL. HENCE, VERIFIED...

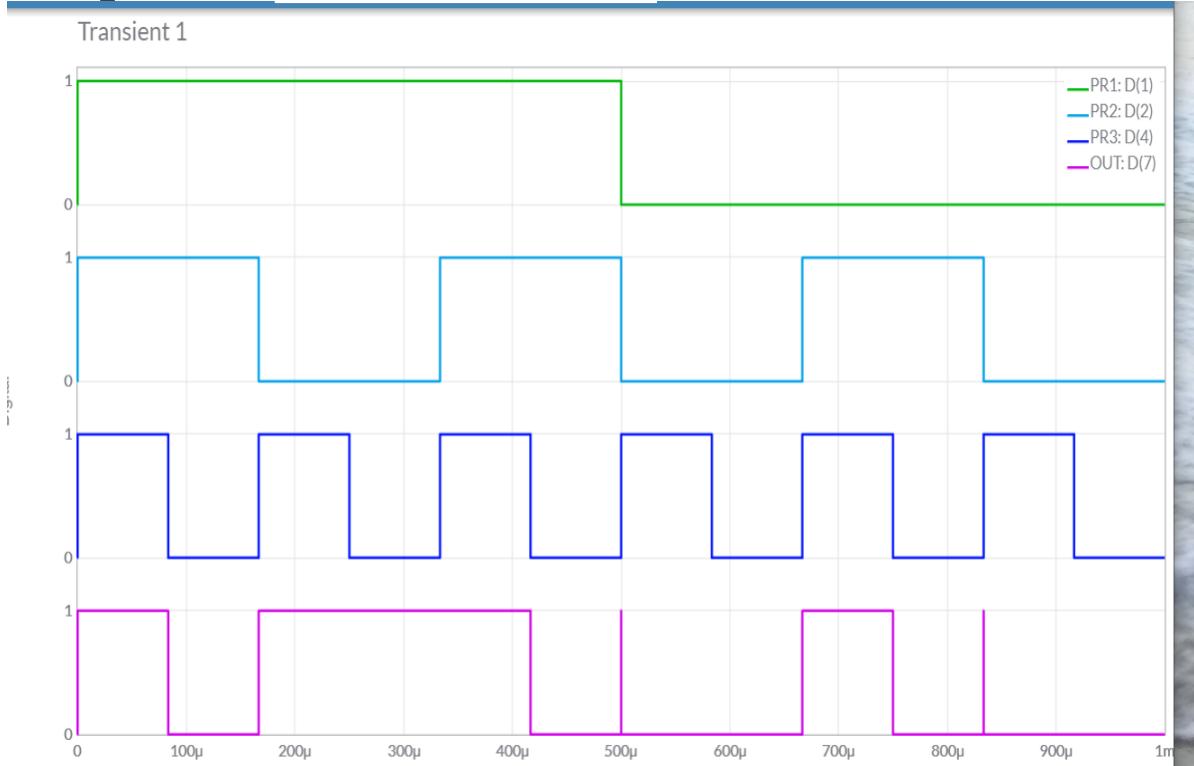
**DLED ASSIGNMENT****Name: Krina Patel****Admission number:U19CS008****Circuit-1****Truth table:**

A	B	C	B'	OUT1=A+B	OUT2=B'+C	OUT=OUT1*OUT2
1	1	1	0	1	1	1
1	1	0	0	1	0	0
1	0	1	1	1	1	1
1	0	0	1	1	1	1
0	1	1	0	1	1	1
0	1	0	0	1	0	0
0	0	1	1	0	1	0
0	0	0	1	0	1	0

Schematic:



Graph:



THE TRUTH TABLE AND TIMING DIAGRAM MATCHES. HENCE, VERIFIED.

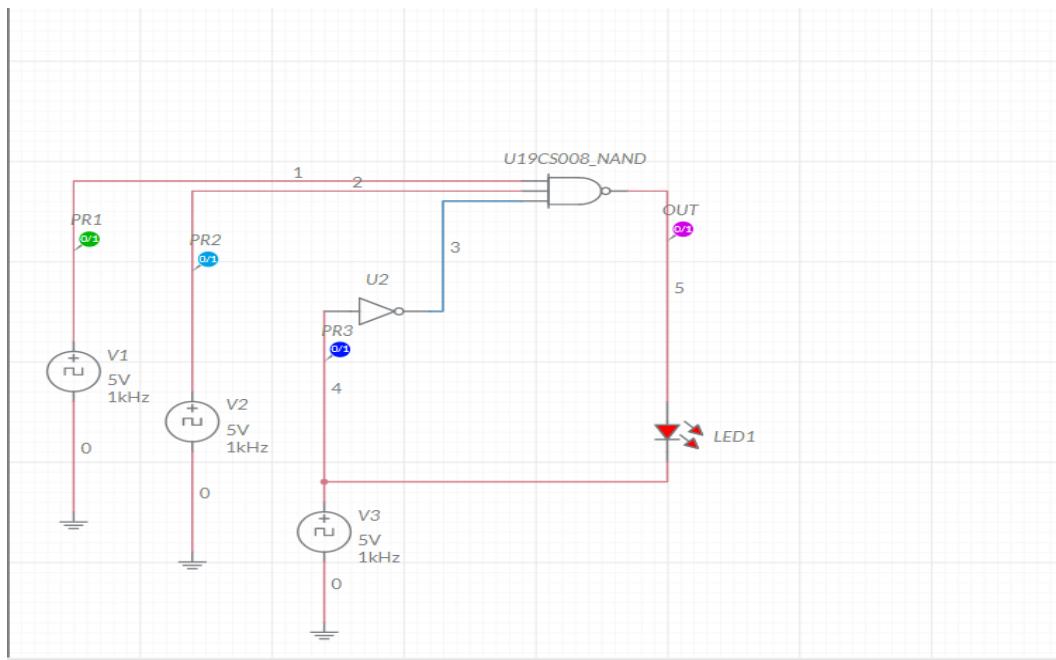
Circuit-2

Truth table:

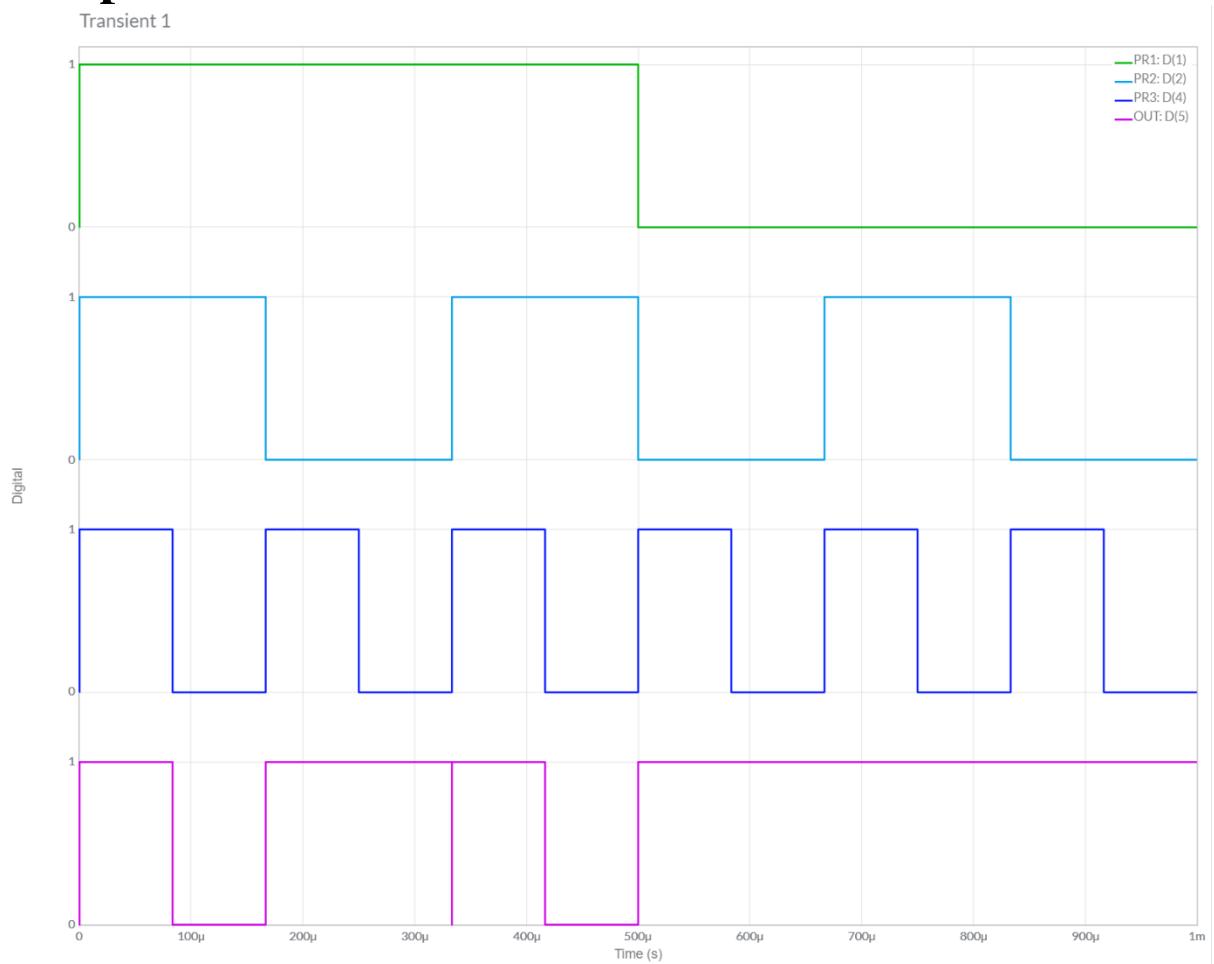
A	B	C	Out1=C'	OUT
1	1	1	0	1
1	1	0	1	0
1	0	1	0	1
1	0	0	1	1
0	1	1	0	1
0	1	0	1	1
0	0	1	0	1
0	0	0	1	1



Schematic:



Graph:





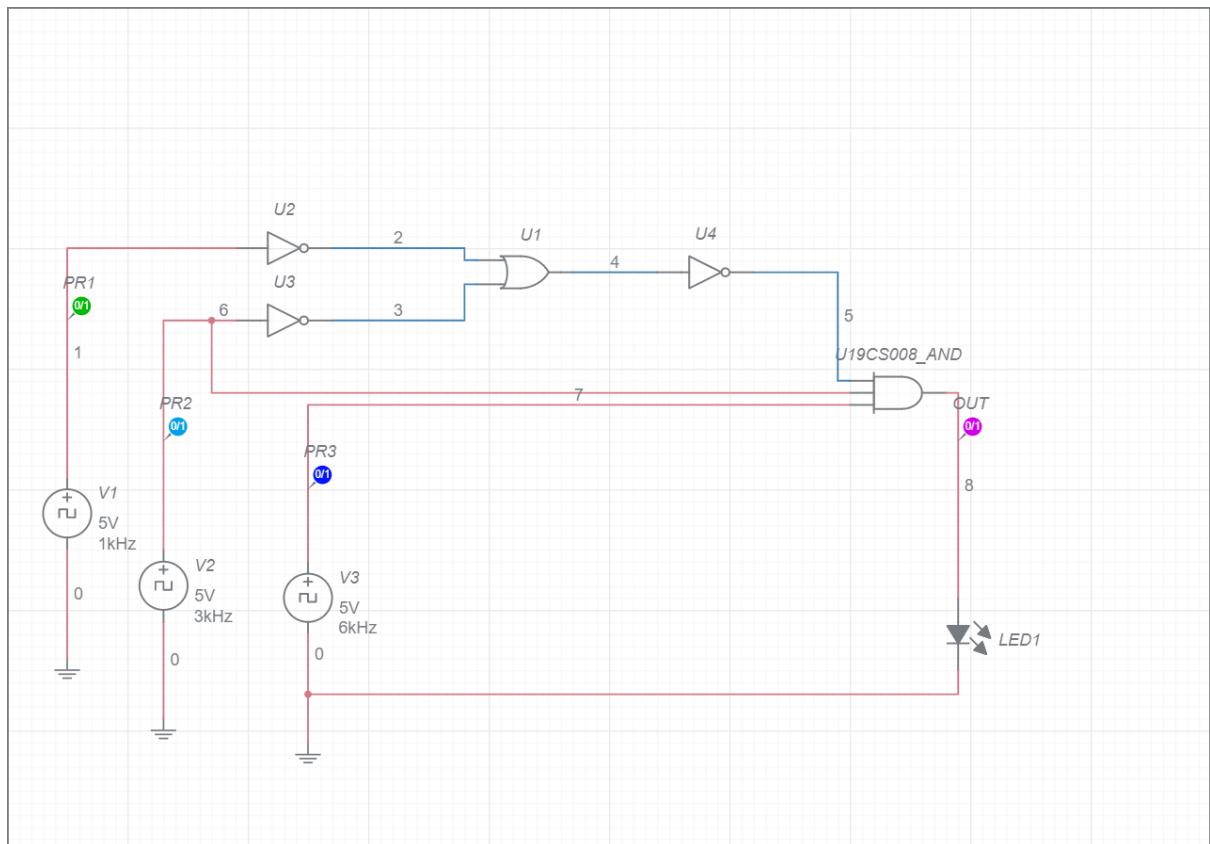
THE TRUTH TABLE AND TIMING DIAGRAM MATCHES. HENCE, VERIFIED.

Circuit-3:

Truth Table:

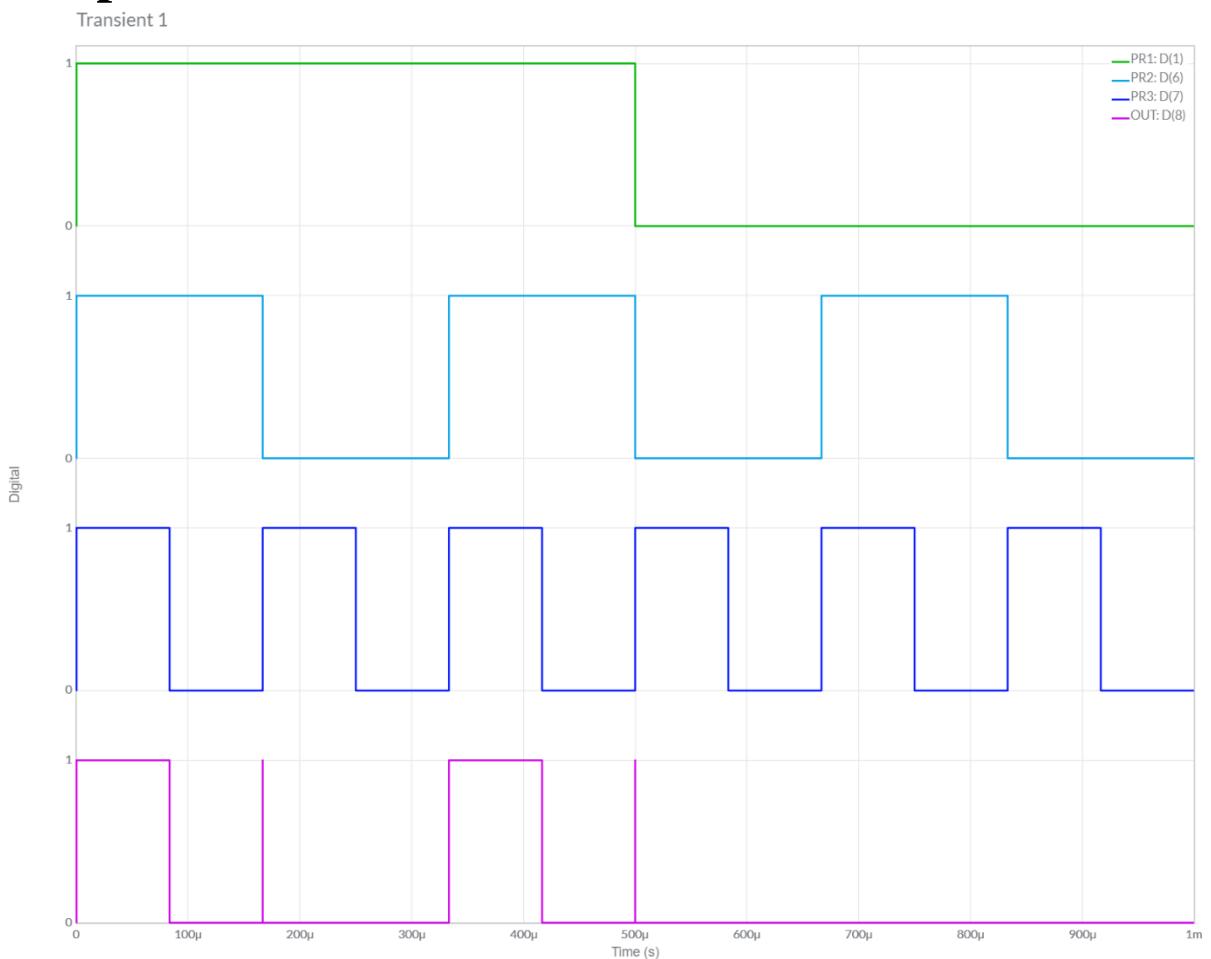
A	B	C	A'	B'	OUT1=A'+B'	OUT2=OUT1'	OUT=OUT2*B*c
1	1	1	0	0	0	1	1
1	1	0	0	0	0	1	0
1	0	1	0	1	1	0	0
1	0	0	0	1	1	0	0
0	1	1	1	0	1	0	0
0	1	0	1	0	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	1	0	0

Schematic:





Graph:



**THE TRUTH TABLE AND TIMING
DIAGRAM MATCHES. HENCE, VERIFIED.**



Expt. No:

3

Date:

27-08-2020

Half Adder and Half Subtractor

AIM: To design and implement Half Adder and Half Subtractor Circuits.

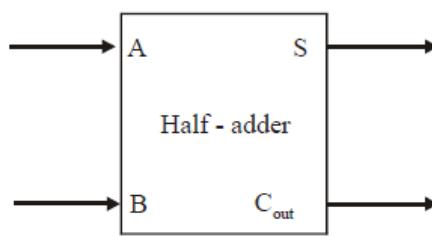
SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator
2. Logic Gates (AND, NOT and EX-OR)

THEORY:**HALF ADDER:**

Adders/Subtractors are important in computers and also in other types of digital systems in which numerical data are processed. An understanding of the basic adder/subtractor operation is fundamental to the study of digital systems.

Figure (a) below shows the logic symbol of a half-adder. As seen from this figure, we find that the half-adder accepts two binary digits on its inputs and produces two digits on its outputs: a sum bit (S) and a carry bit (C_{out}). Fig (b) shows the truth table for the half-adder.



(a) logic symbol

Inputs		Outputs	
A	B	S	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) truth table

The half-adder follows the basic rules of binary addition:

$$0 + 1 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ with a carry of } 1$$

The Boolean expression for the sum output (S) can be expressed by the equation.

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

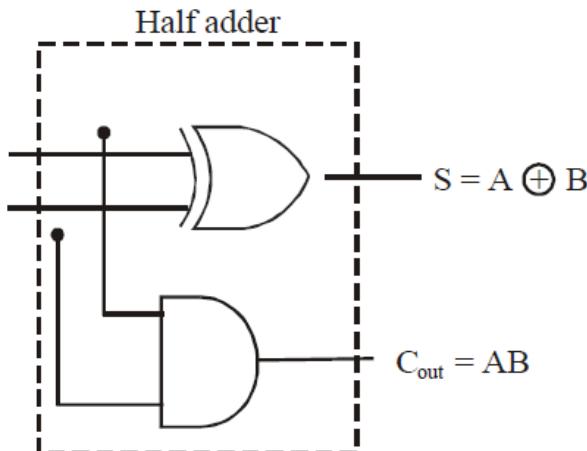
$$= A \oplus B$$



and the Boolean expression for the carry output by,

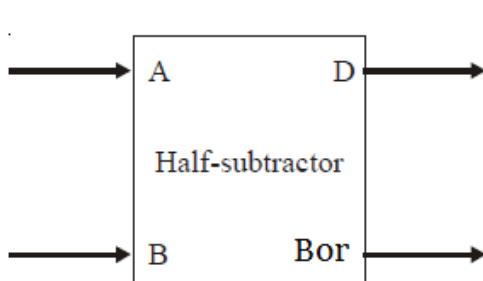
$$C_{out} = AB$$

The above two equations can be implemented by a logic circuit shown in Fig below. As seen from this figure, the sum output (S) is obtained from an exclusive-OR gate while the carry output (Cout) is the output of a two-input AND gate.



HALF SUBTRACTOR:

Fig. (a) below shows the logic symbol of a half-subtractor. As seen from this figure, we find that the half-subtractor accepts two binary digits on its inputs and produce two digits on its outputs : a difference bit (D) and a borrow bit (Bor). Fig (b) shows the truth table for the half-subtractor.



(a) logic symbol

<i>Inputs</i>		<i>Outputs</i>	
<i>A</i>	<i>B</i>	<i>D</i>	<i>Bor</i>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(b) truth table

The half-subtractor follows the basic rules for binary subtraction:

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ with a borrow of } 1$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

The Boolean expression for the difference bit (D) can be expressed by the equation.

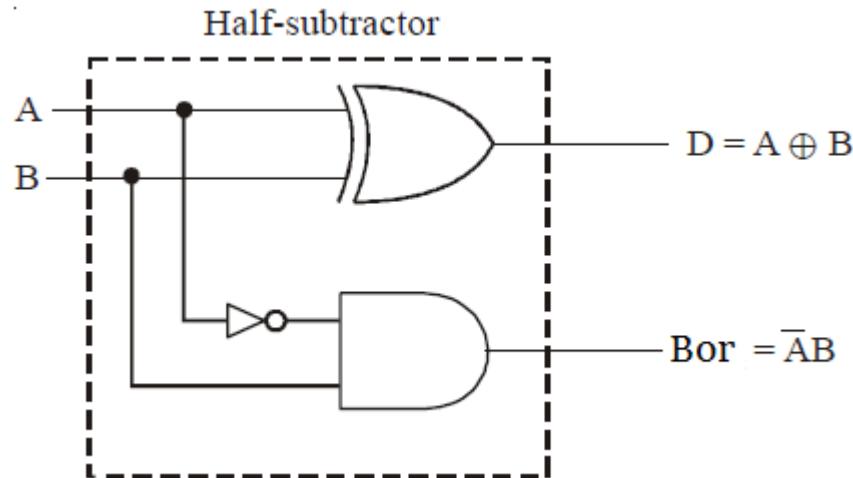


$$\begin{aligned} D &= A\bar{B} + \bar{A}B \\ &= A \oplus B \end{aligned}$$

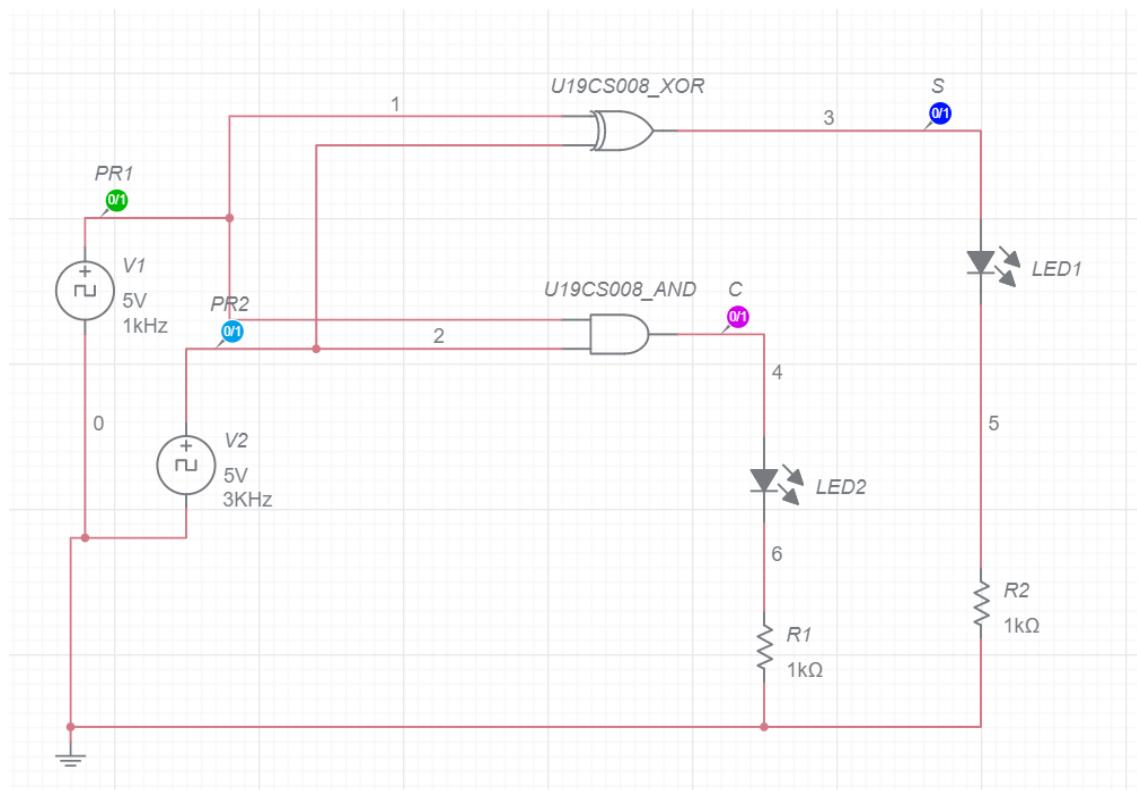
and the Boolean expression for the borrow bit,

$$\text{Bor} = \bar{A}B$$

The above two expressions can be implemented by a logic circuit shown in Fig. below. As seen from this figure, the difference output (D) is obtained from an exclusive-OR gate while the borrow bit (Bor) is the output of a two-input AND gate.

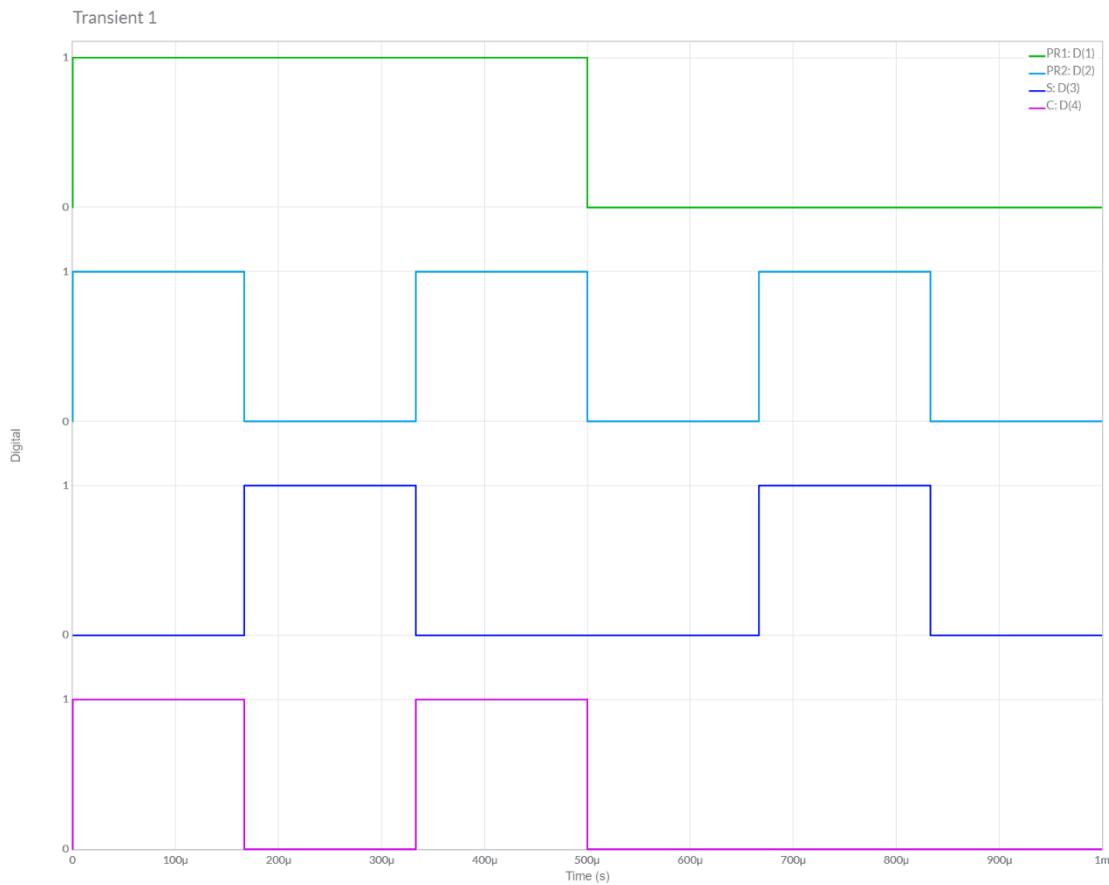


HALF ADDER: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

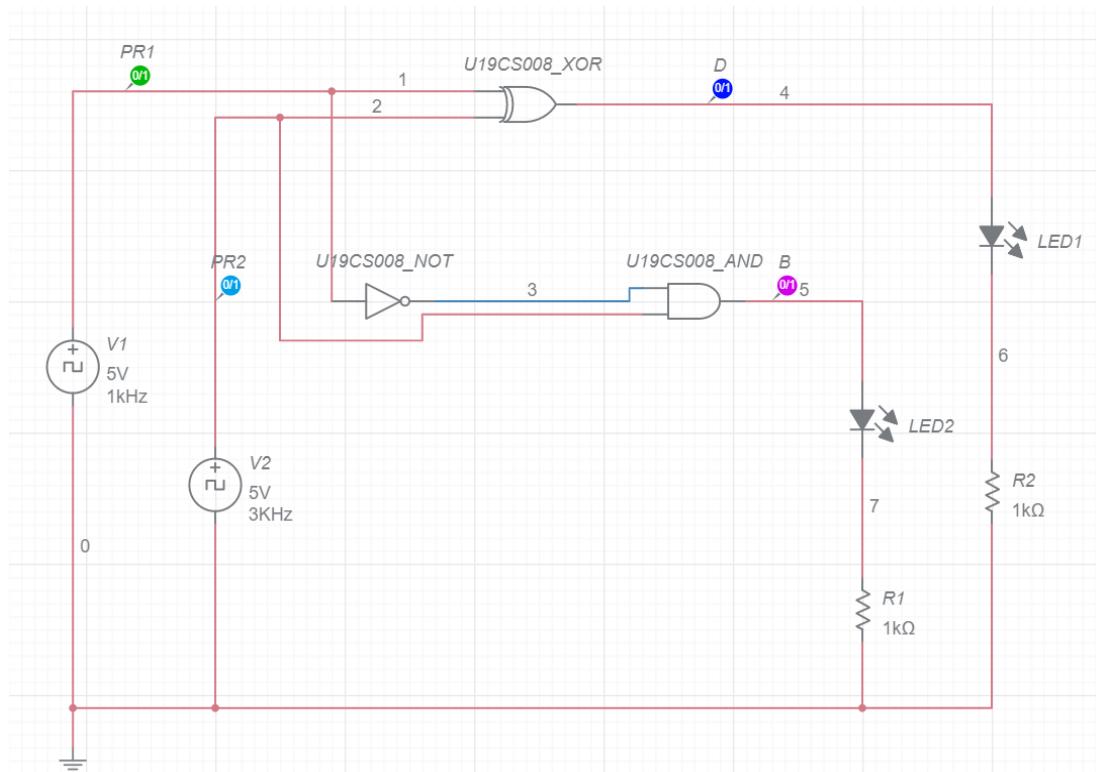


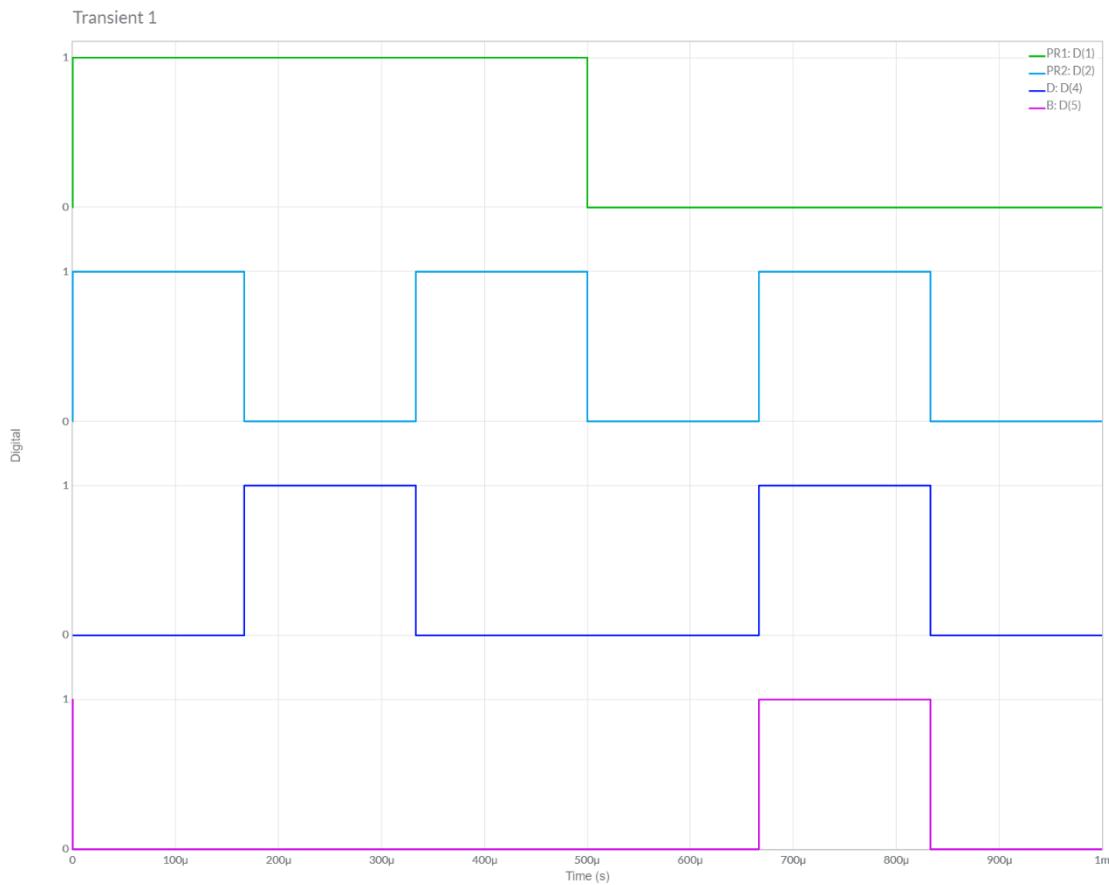


HALF ADDER: OUTPUT PLOTS/WAVEFORMS (FROM MULTISIM):



HALF SUBTRACTOR: CIRCUIT/CONNECTION DIAGRAMS (MULTISIM)



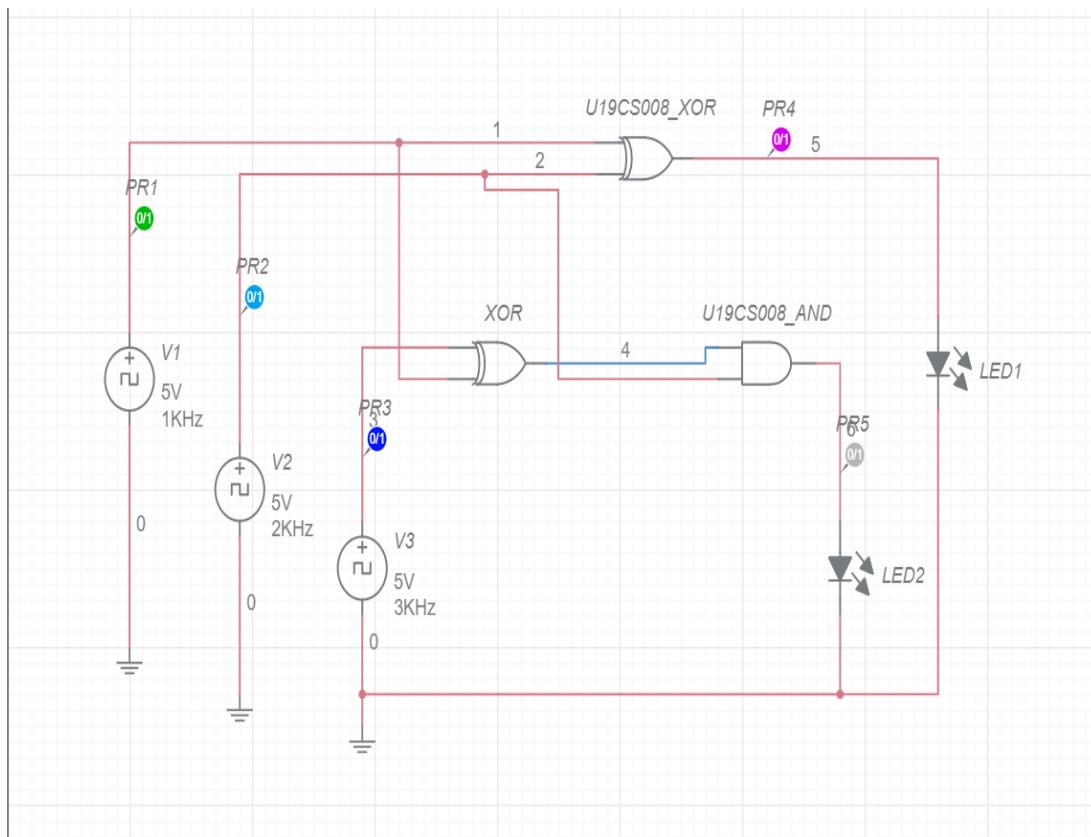
**HALF SUBTRACTOR: OUTPUT PLOTS/WAVEFORMS (MULTISIM)****CONCLUSIONS**

THE TRUTH TABLE IN THEORY AND THE SIMULATION OF THE HALF ADDER AND HALF SUBTRACTOR CIRCUIT ON MULTISIM LIVE BOTH ARE EQUAL. HENCE, VERIFIED...



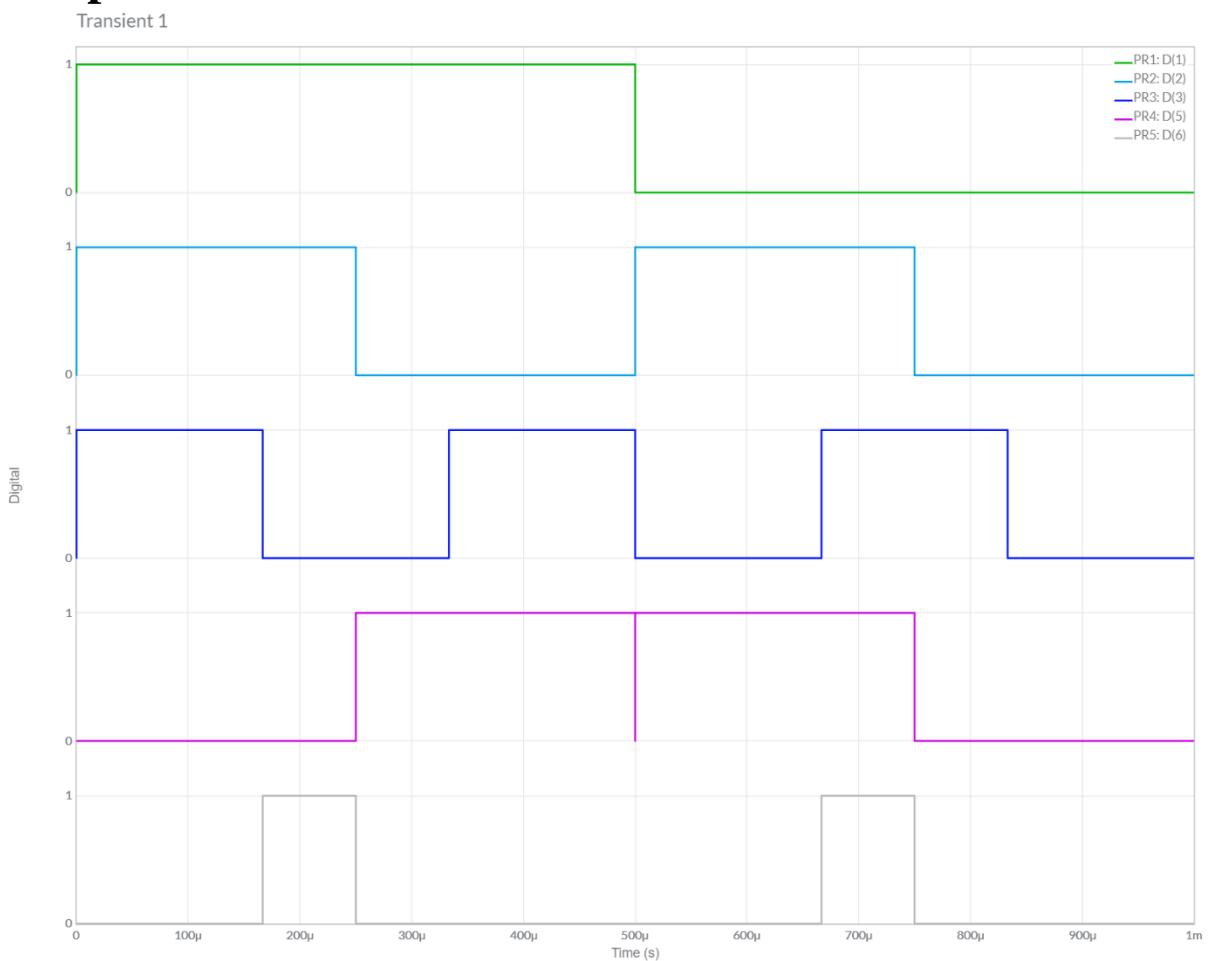
DLED ASSIGNMENT-3
NAME: KRINA PATEL
ADMISSION NO.: U19CS008

1) Half adder and half subtractor using mode control 'M'.. circuit:





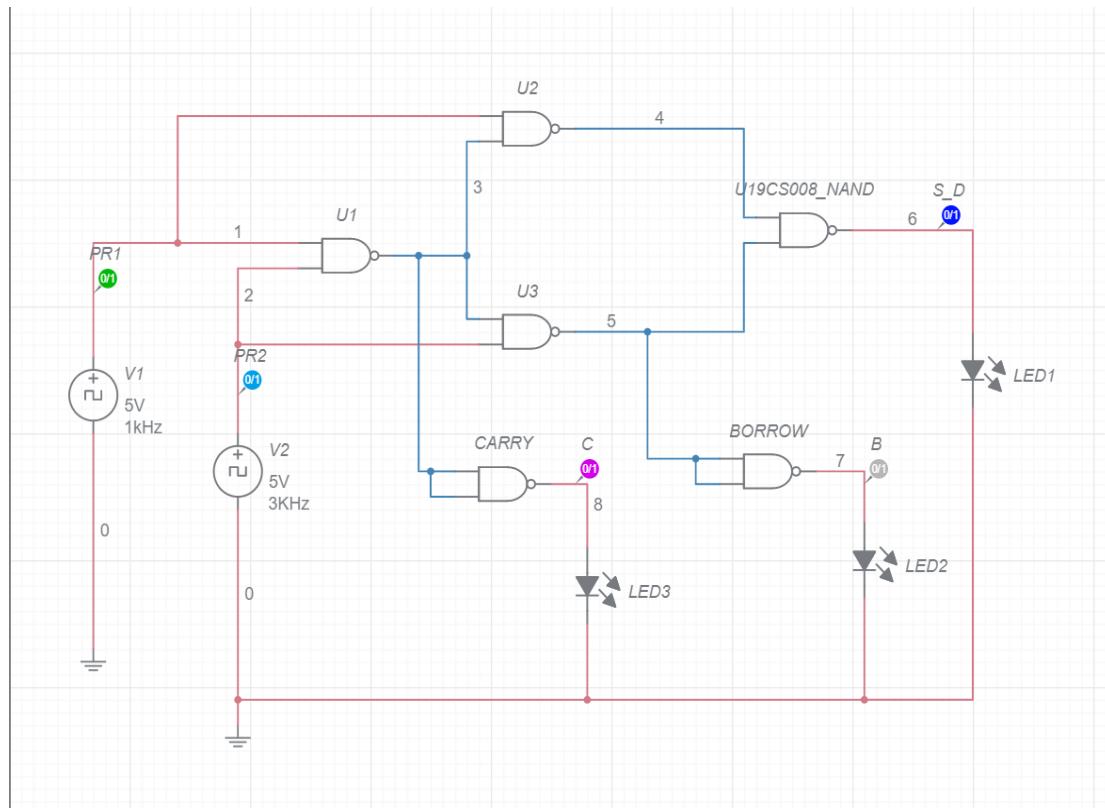
Graph :





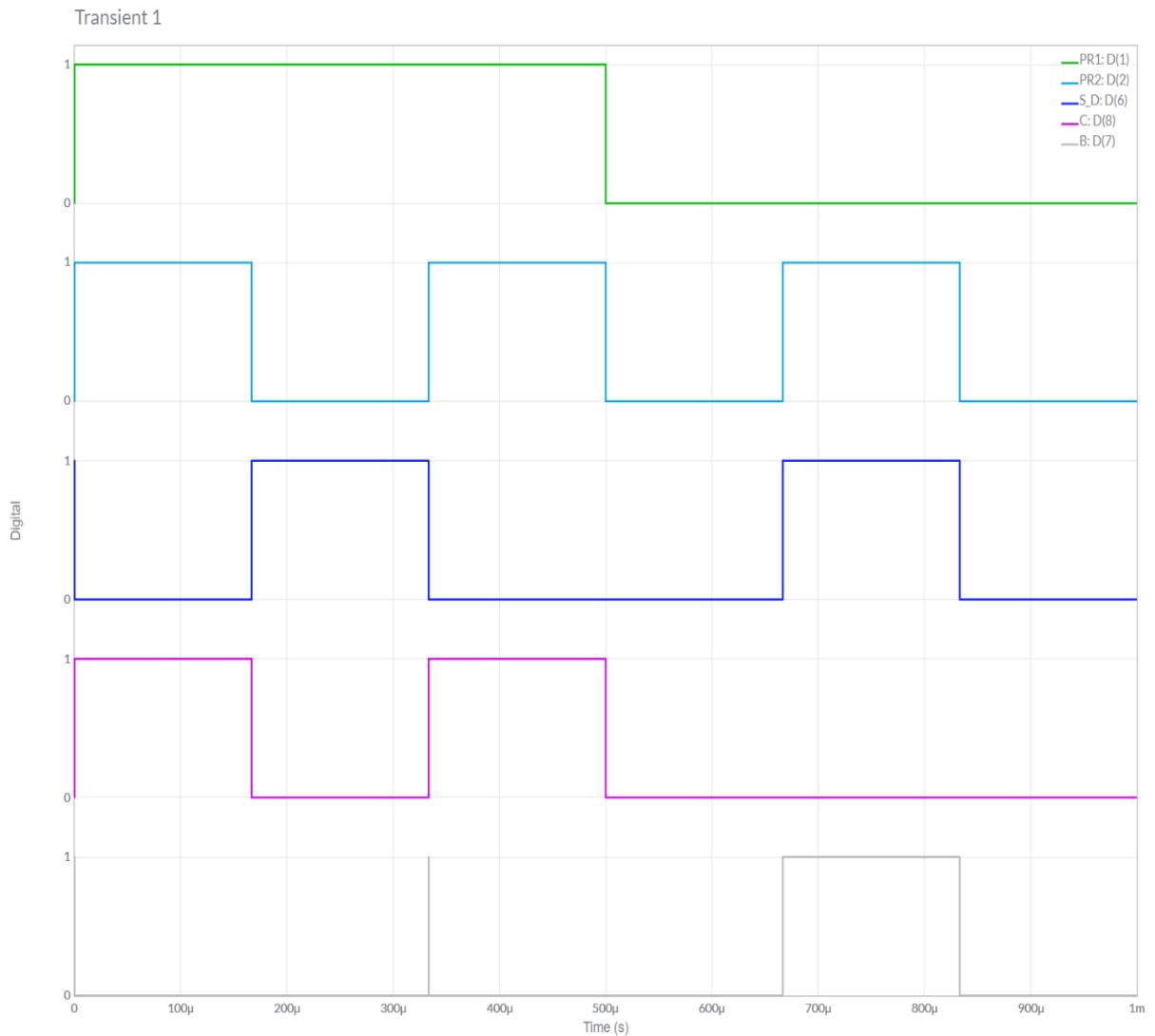
Designing above circuit using least number of NAND gates..

Circuit :





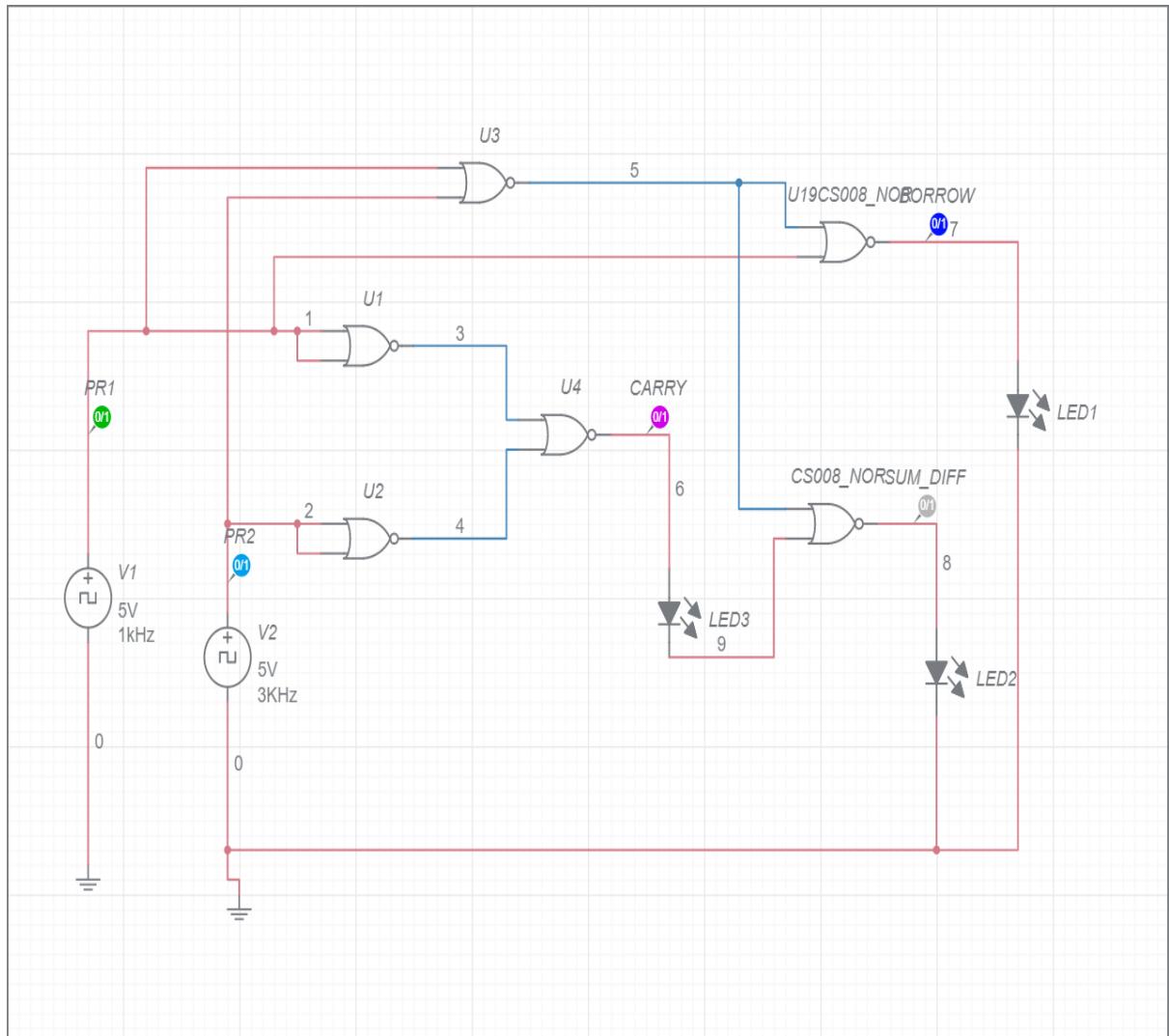
Graph :





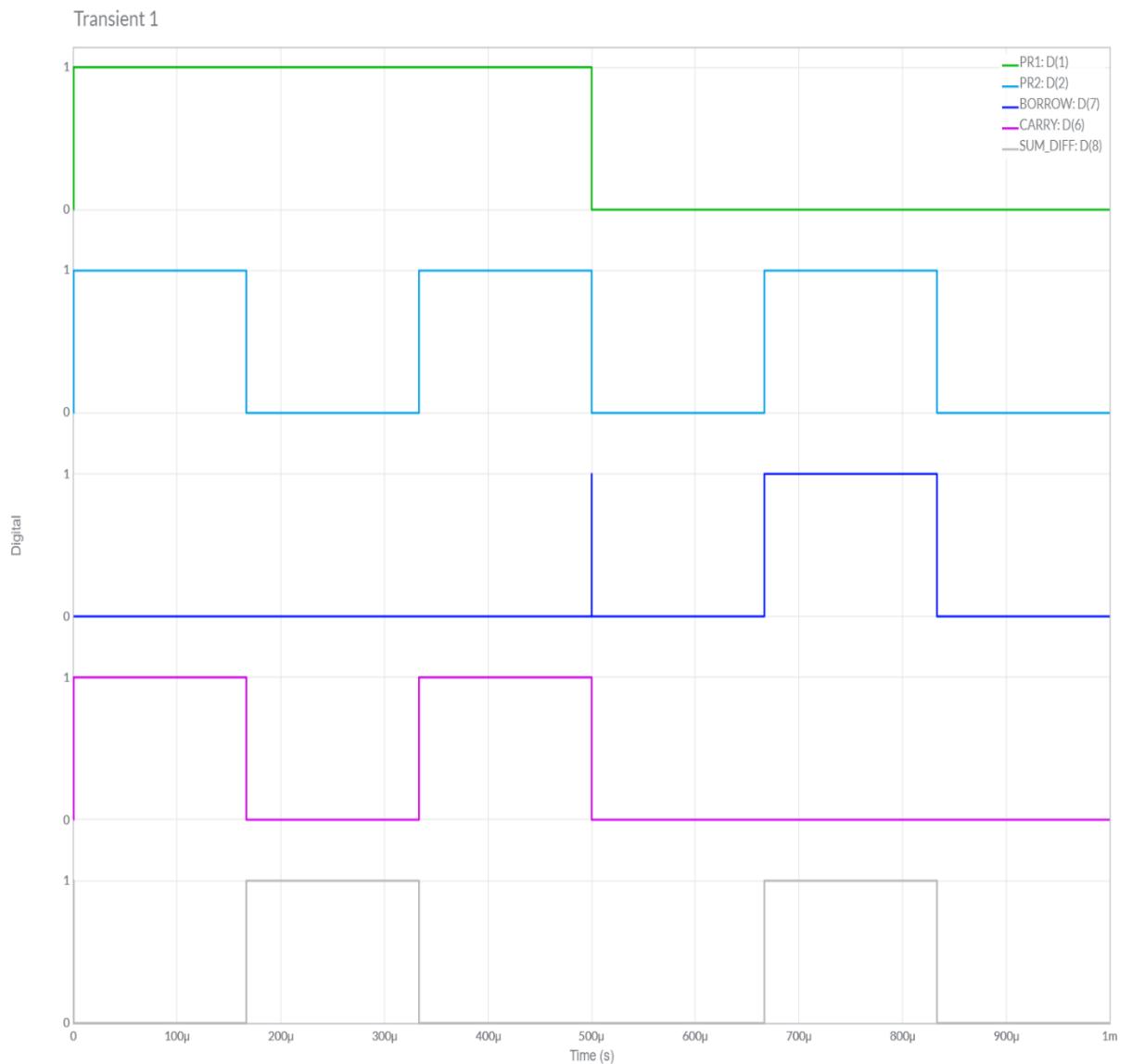
Designing above circuit using least number of NOR gates only...

Circuit:





Graph:





Expt. No:

4

Date:

3-09-2020

Full Adder and Full Subtractor

AIM: To design and implement Full Adder and Full Subtractor Circuits.

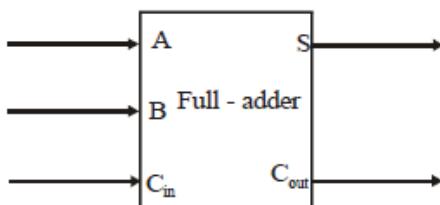
SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator
2. Logic Gates (AND, NOT and EX-OR)

THEORY:**FULL ADDER:**

Adders/Subtractors are important in computers and also in other types of digital systems in which numerical data are processed. An understanding of the basic adder/subtractor operation is fundamental to the study of digital systems.

Figure (a) below shows the logic symbol of a full-adder. As seen from this figure, we find that the full-adder accepts three binary digits on its inputs (two new bits and one carry from the previous stage) and produces two digits on its outputs: a sum bit (S) and a carry bit (C_{out}). Fig (b) shows the truth table for the full-adder.



(a) logic symbol

Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b) truth table

The full –adder also follows the same basic rules of binary addition as half-adder:



$0 + 0 + 0$	=	0	with carry	0
$0 + 0 + 1$	=	1	with carry	0
$0 + 1 + 0$	=	1	with carry	0
$0 + 1 + 1$	=	0	with carry	1
$1 + 0 + 0$	=	1	with carry	0
$1 + 0 + 1$	=	0	with carry	1
$1 + 1 + 0$	=	0	with carry	1
$1 + 1 + 1$	=	1	with carry	1

The Boolean expression for the sum output (S) can be obtained from the above truth table by summing and then simplifying the terms for which $S=1$. Thus, the sum is

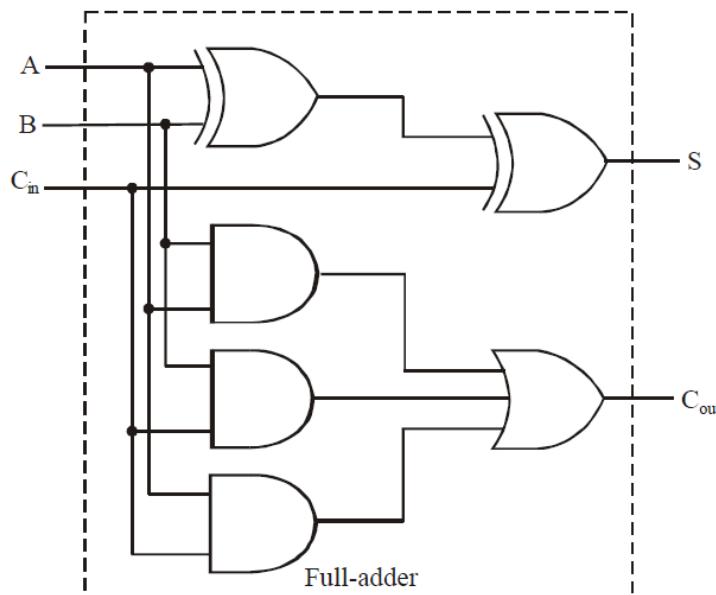
$$S = A \oplus (B \oplus C_{in})$$

$$= A \oplus B \oplus C_{in}$$

Similarly, adding up all the terms for which carry output (C_{out}) is 1 and simplifying will lead us to expression for output carry as

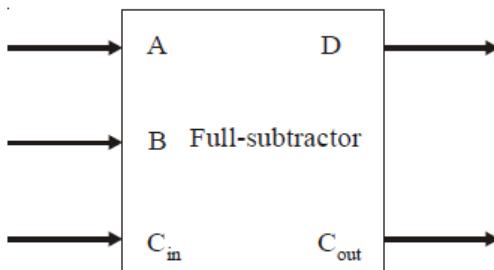
$$C_{out} = BC_{in} + AC_{in} + AB$$

The equations for Sum and Carry can be easily implemented by using logic gates. From equation of Sum we find that to implement to full-adder's sum output function, two 2-input Exclusive-OR gates can be used. The first Exclusive-OR gate generates the term $A \oplus B$, and the second has its inputs the output of the first Exclusive-OR gate and the input carry as shown in the Fig below. Similarly from equation of Carry we find that to implement the full-adder's carry output function, three 2-input AND gates followed by a 3-input OR gate can be used. The complete circuit of a full adder is shown below.



**FULL SUBTRACTOR:**

Fig. (a) below shows the logic symbol of a full-subtractor. As seen from this figure, we find that the full-subtractor accepts three inputs. Two input bits A and B and a borrow bit (B_{in}). It has two outputs : (1) a difference output (D) and a borrow output (B_{out}). Fig. (b) shows the truth table for the full-subtractor.



(a) logic symbol

Inputs			Outputs	
A	B	C_{in}	D	C_{out}
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

(b) truth table

We observe that the full-subtractor also follows the basic rules of binary subtraction as half-subtractor:

$$\begin{aligned}
 0 - 0 - 0 &= 0 \quad \text{with borrow 0} \\
 0 - 0 - 1 &= 1 \quad \text{with borrow 1} \\
 0 - 1 - 0 &= 1 \quad \text{with borrow 1} \\
 0 - 1 - 1 &= 0 \quad \text{with borrow 1} \\
 1 - 0 - 0 &= 0 \quad \text{with borrow 0} \\
 1 - 1 - 0 &= 0 \quad \text{with borrow 0} \\
 1 - 1 - 1 &= 1 \quad \text{with borrow 1}
 \end{aligned}$$

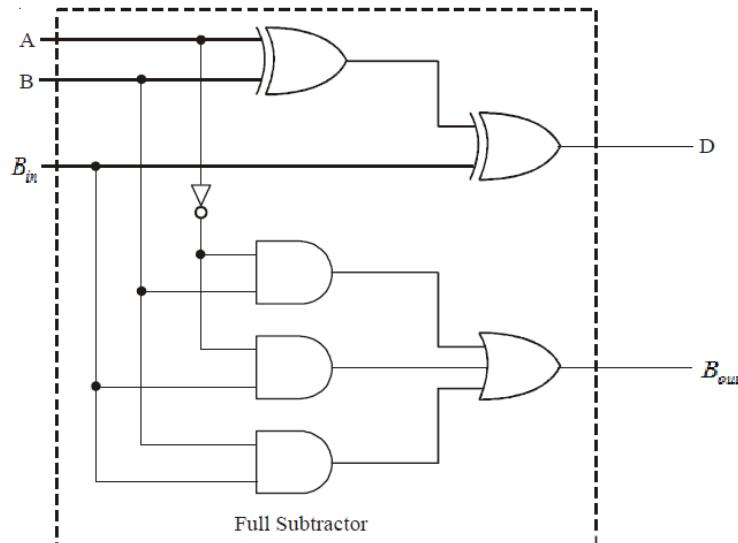
The Boolean expression for the difference bit (D) can be obtained by summing and simplifying all the input combinations from the truth table which have 1 in the corresponding difference column. The final simplified expression for difference is given by

$$D = A \oplus B \oplus B_{in}$$

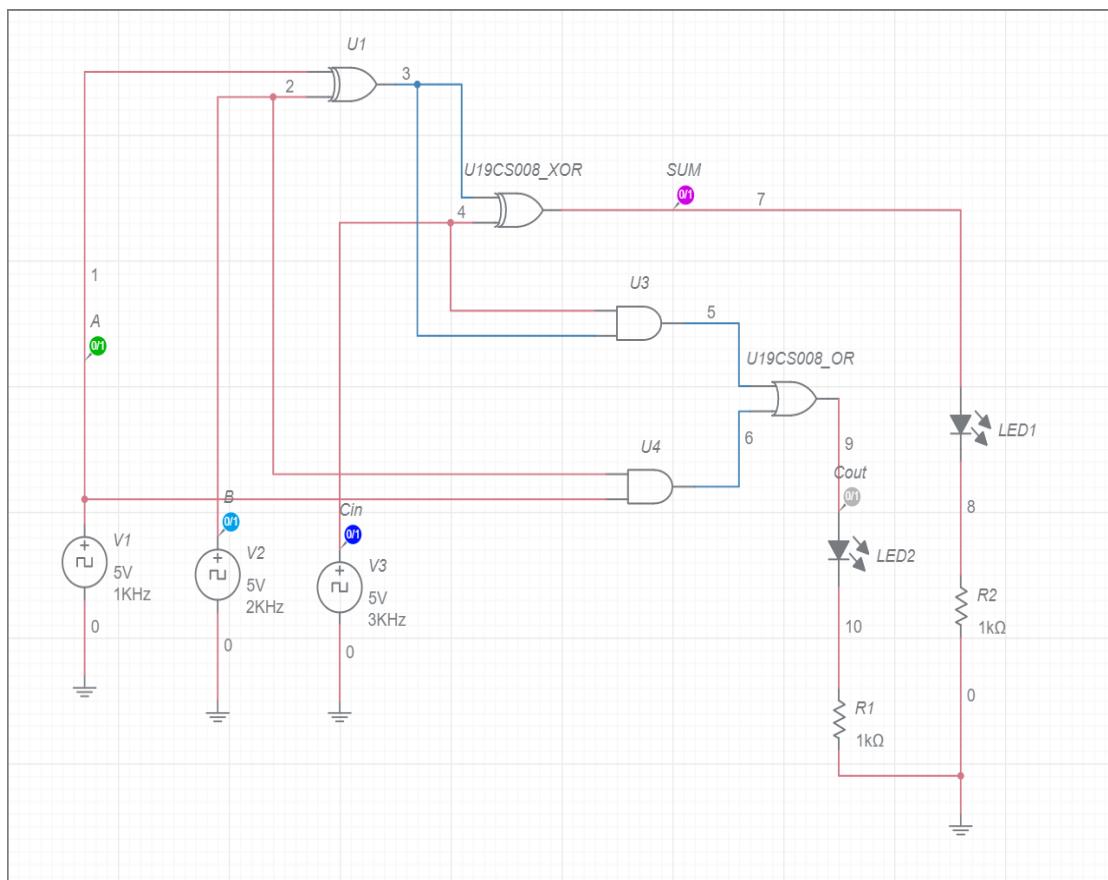
and, the Boolean expression for the borrow bit,

$$B_{out} = \overline{AB} + BB_{in} + \overline{AB}_{in}$$

From the above two expressions we find that to implement full-subtractor's difference output function, two-2 input Exclusive-OR gates can be used. Similarly from equation of borrow we find that to implement the full-subtractor's borrow output function, a NOT gate, three 2-input AND gates followed by a 3-input OR gate can be used. The complete circuit of a full-subtractor is shown below.

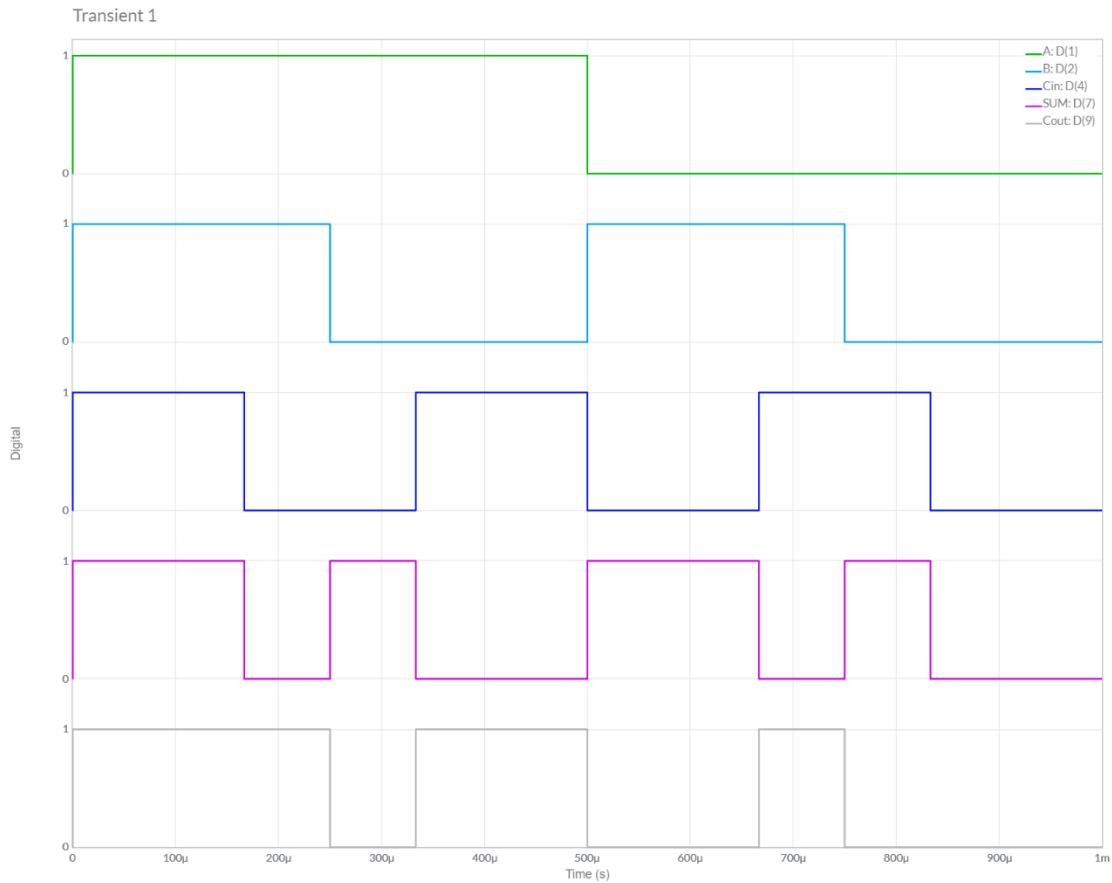


FULL ADDER: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

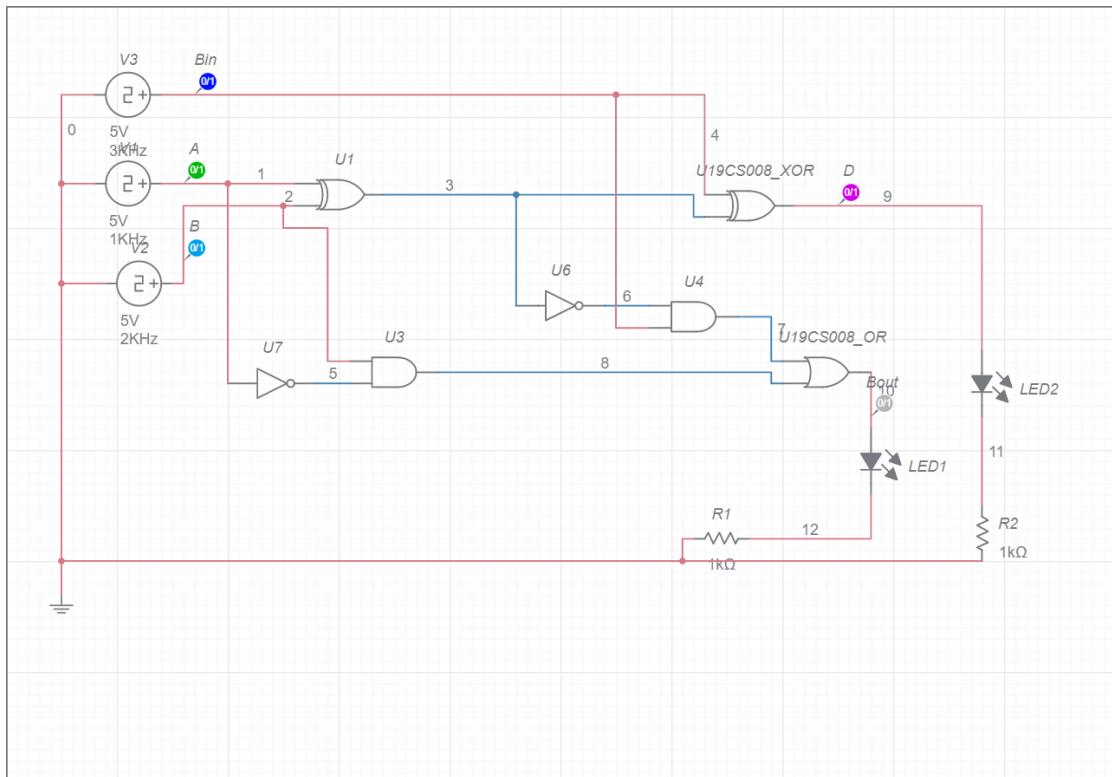


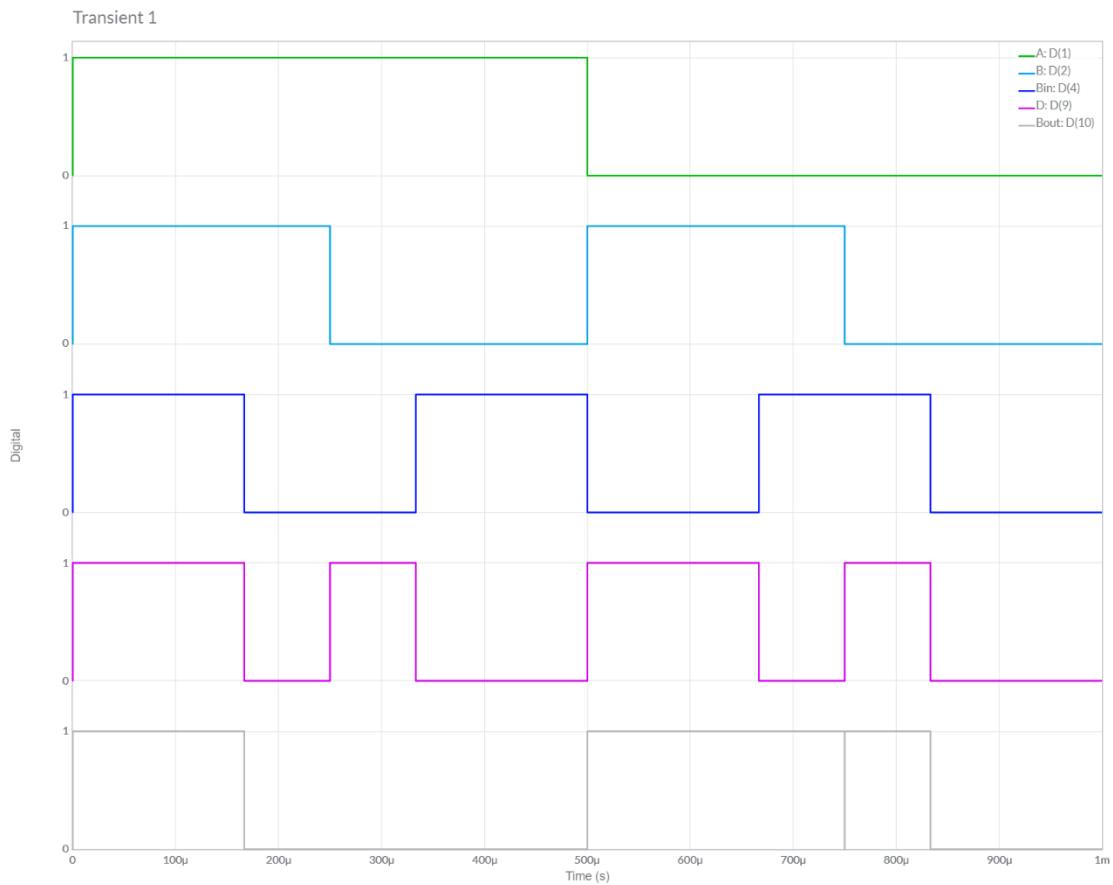


FULL ADDER: OUTPUT PLOTS/WAVEFORMS (FROM MULTISIM):



FULL SUBTRACTOR: CIRCUIT/CONNECTION DIAGRAMS (MULTISIM)

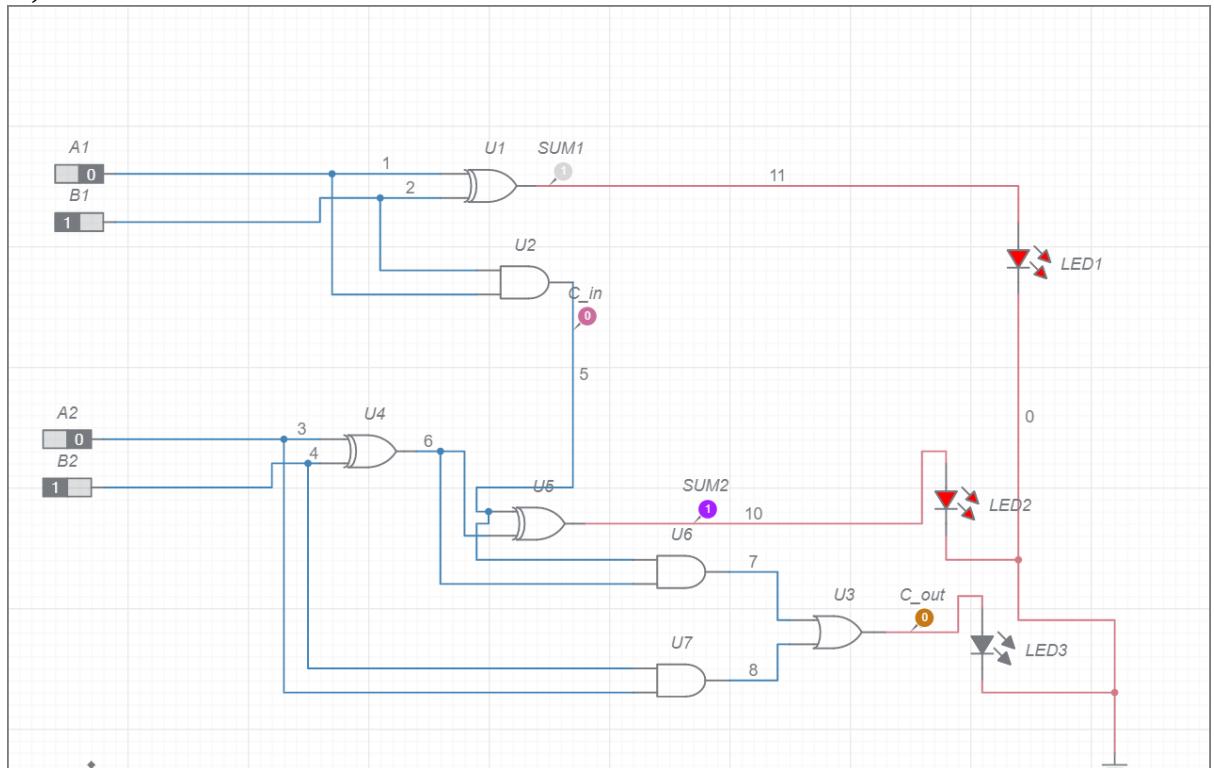


**FULL SUBTRACTOR: OUTPUT PLOTS/WAVEFORMS (MULTISIM)****CONCLUSIONS**

THE TRUTH TABLE IN THEORY AND THE SIMULATION OF THE FULL ADDER AND FULL SUBTRACTOR CIRCUIT ON MULTISIM LIVE BOTH ARE EQUAL. HENCE VERIFIED...

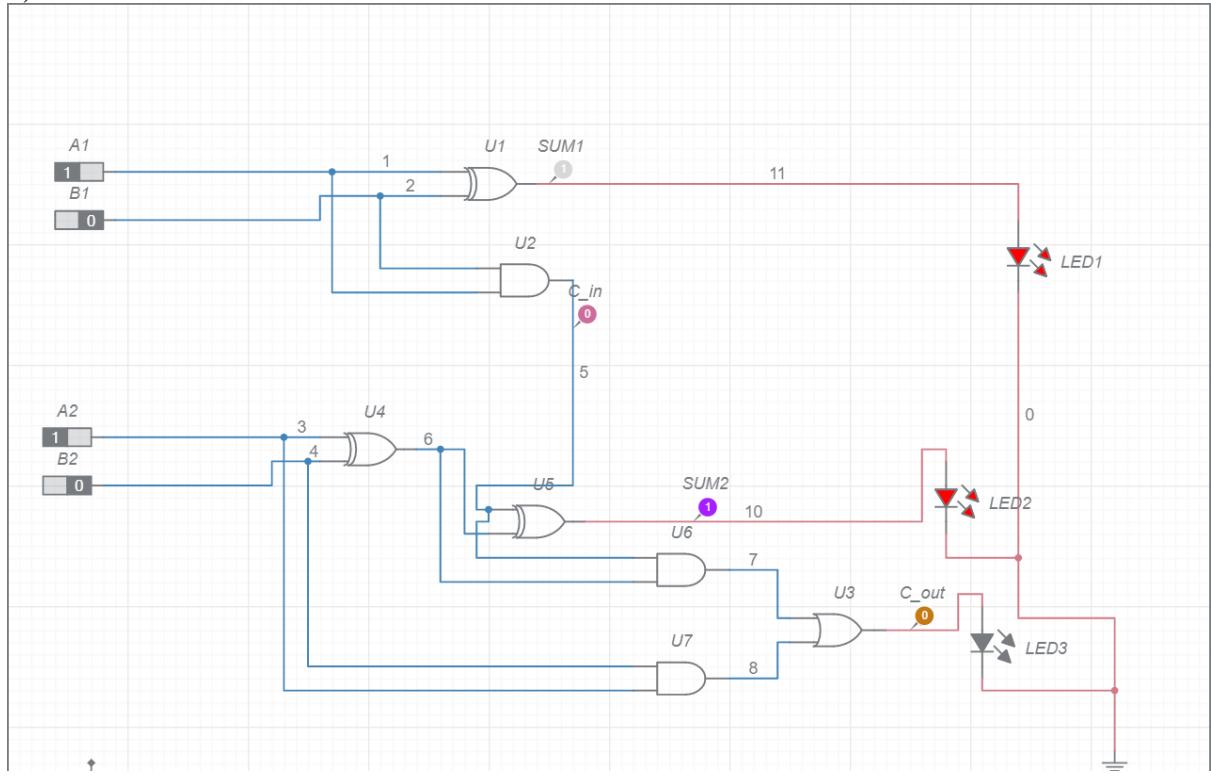
**DLED ASSIGNMENT-4****NAME : KRINA PATEL****ADMISSION NUMBER:U19CS008**

Question-1) Two bit adder circuit. Attach screenshots for any four input combination.

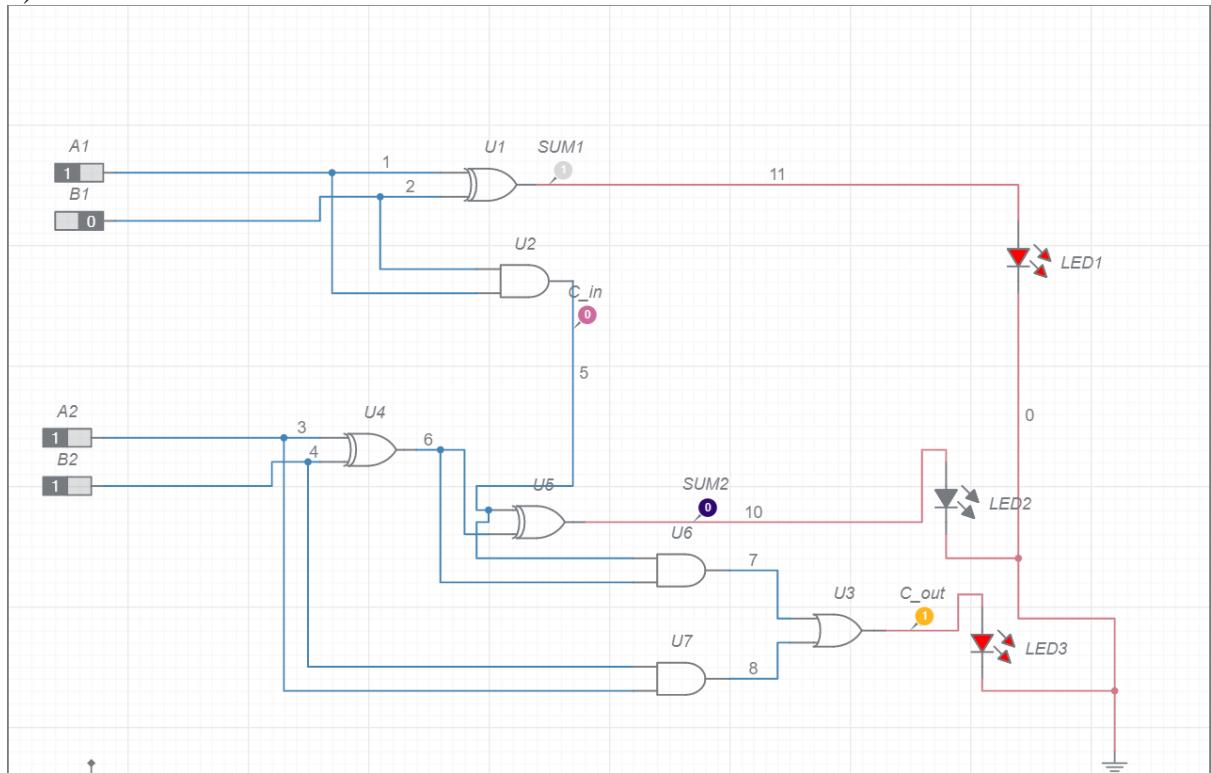
1)



2)

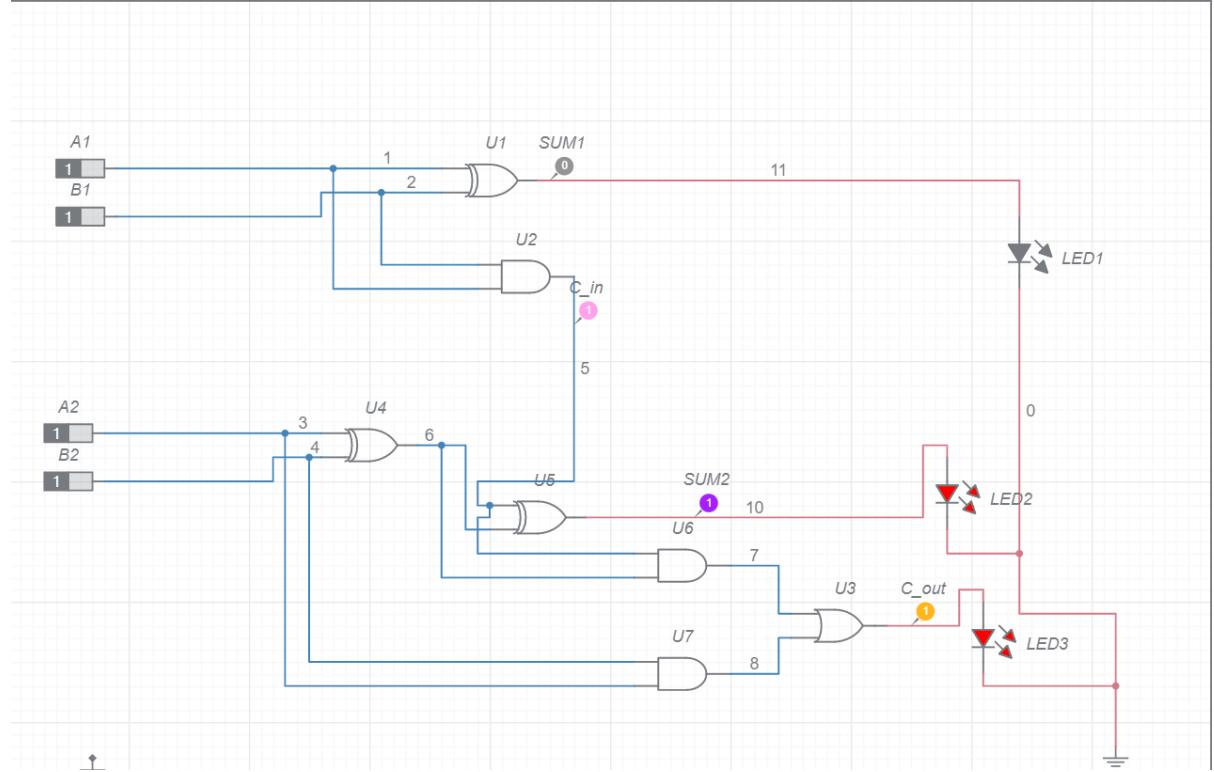


3)





4)



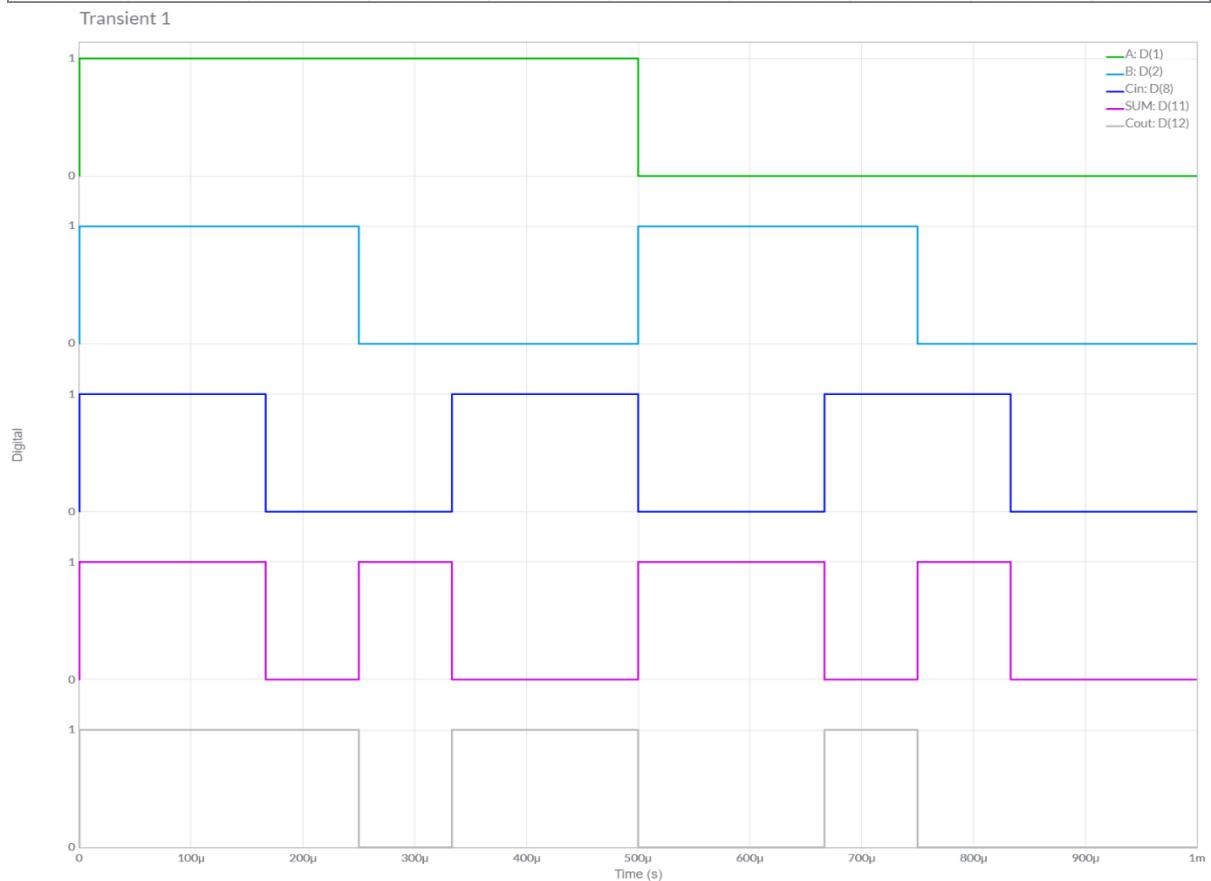
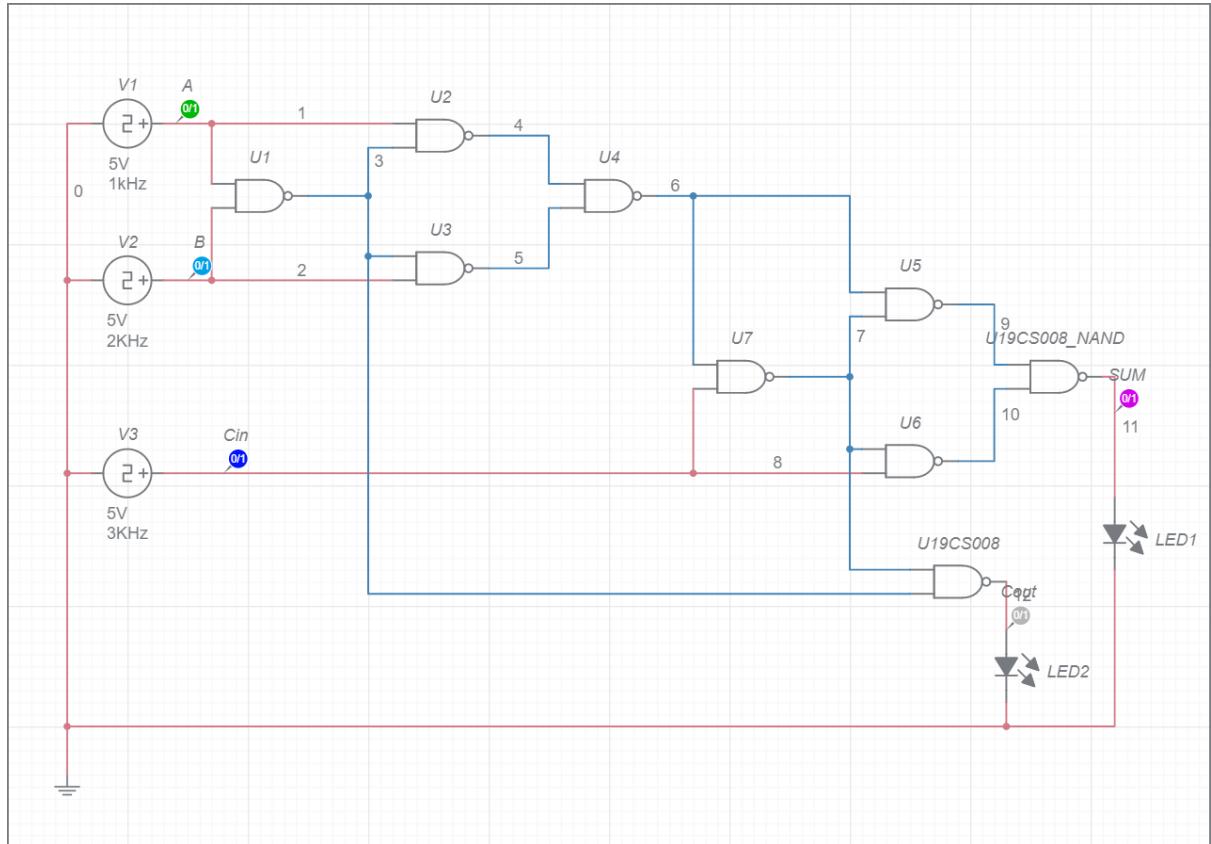
Truth table:

Ao	B1	A2	B2	SUM1	SUM2	CARRYout
0	1	0	1	1	1	0
1	0	1	0	1	1	0
1	0	1	1	1	0	1
1	1	1	1	0	1	1

HERE THEROTICAL AND PRACTICAL BOTH VALUES ARE EQUAL. HENCE VERIFIED..

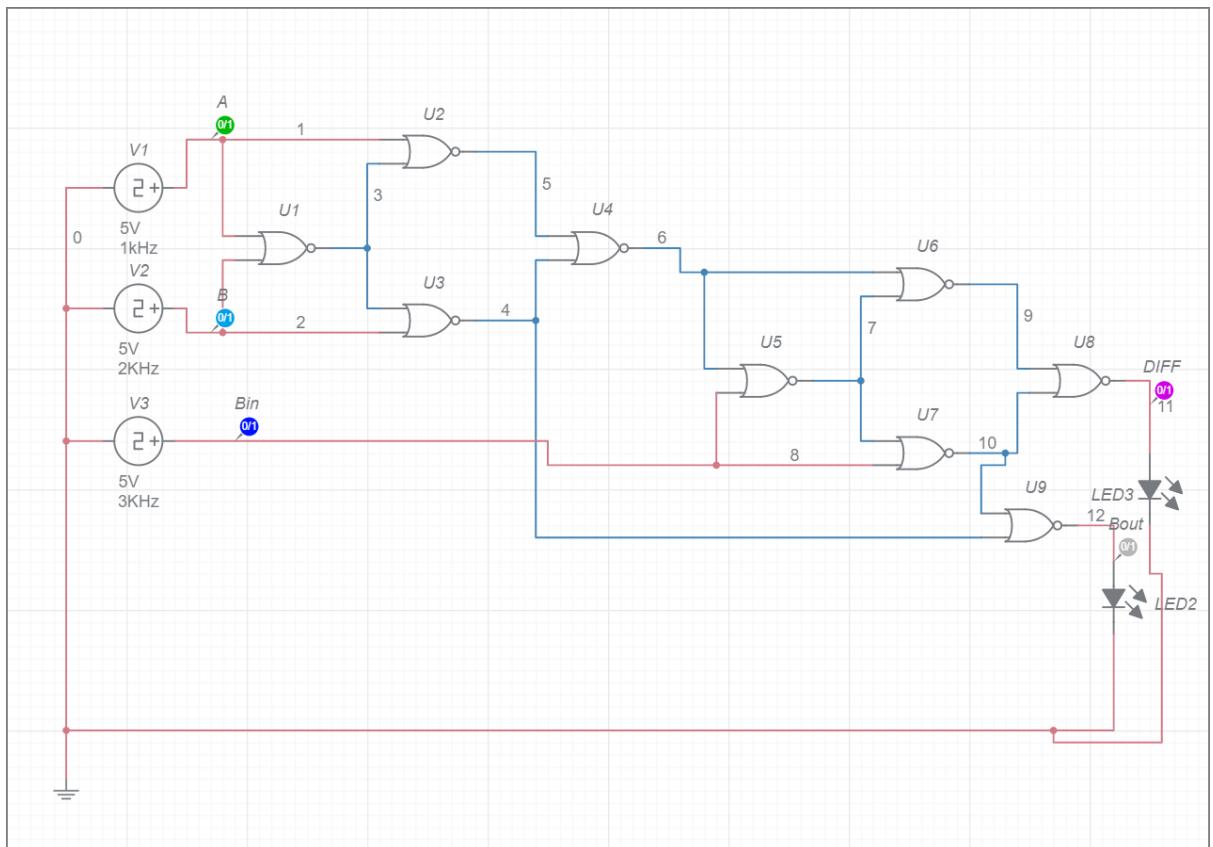


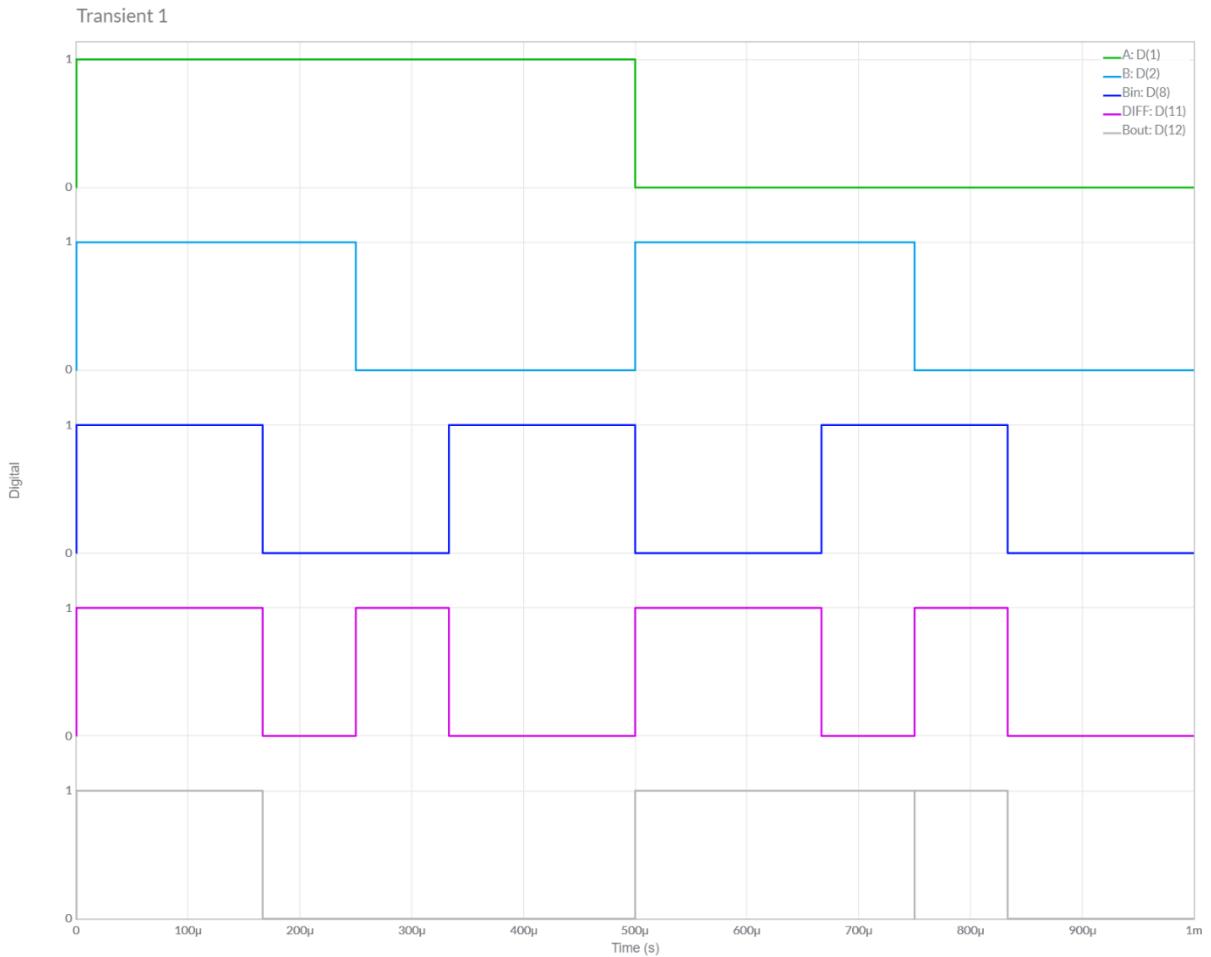
QUESTION-2) FULL ADDER USING LEAST NUMBER OF NAND GATES...



**TURTH TABLE:**

A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

QUESTION-3) FULL SUBTRACTOR USING LEAST NUMBER OF NOR GATES.

**TRUTH TABLE:**

A	B	Bin	DIFF	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

HERE IN BOTH QUESTION GRAPHER DATA AND TRUTH TABLE BOTH ARE EQUAL. HENCE VERIFIED.



Expt. No:

5

Date:

10-09-2020

V-I Characteristics of PN – Junction Diode

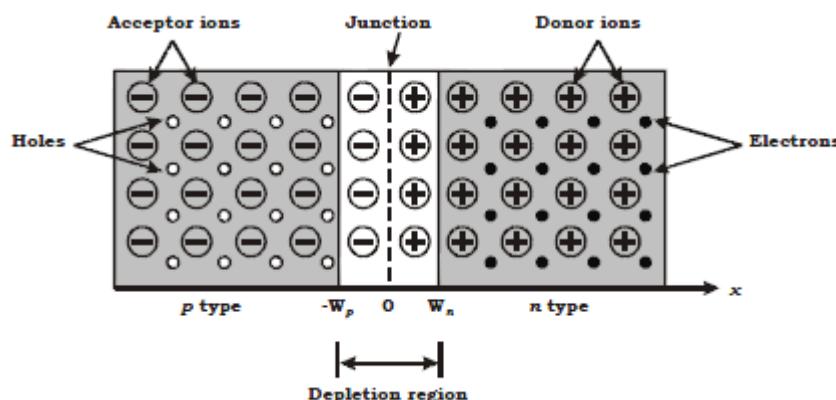
AIM: To obtain and plot both forward and reverse characteristics of PN – Junction Diode.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator/Circuit Simulator
2. PN Diode
3. Resistor (1K)
4. Variable DC Supply

THEORY:

The semiconductor diode is formed by doping P-type impurity on one side and N-type of impurity on other side of the semiconductor crystal forming a p-n junction as shown in the following figure.



At the junction initially free charge carriers from both side recombine forming negatively charged ions in P side of junction(an atom in P-side accept electron and becomes negatively charged ion) and positively charged ion on n side(an atom in n-side accepts hole i.e. donates electron and becomes positively charged ion)region. This region deplete of any type of free charge carrier is called as depletion region. Further recombination of free carrier on both side is prevented because of the depletion voltage generated due to charge carriers kept at distance by depletion (acts as a sort of insulation) layer as shown dotted in the above figure.

When voltage is not applied across the diode, depletion region is formed as shown in the above figure. When the voltage is applied between the two terminals of the diode (anode and cathode) two possibilities arise depending on polarity of DC supply.



1) Forward-Bias:

When the +Ve terminal of the battery is connected to P-type material & -Ve terminal to N-type terminal as shown in the circuit diagram, the diode is said to be forward biased. The application of forward bias voltage will force electrons in N-type and holes in P-type material to recombine. This reduces width of depletion region. This further will result in increase in majority carriers flow across the junction. If forward bias is further increased in magnitude the depletion region width will continue to decrease, resulting in exponential rise in current as shown in ideal diode characteristic curve.

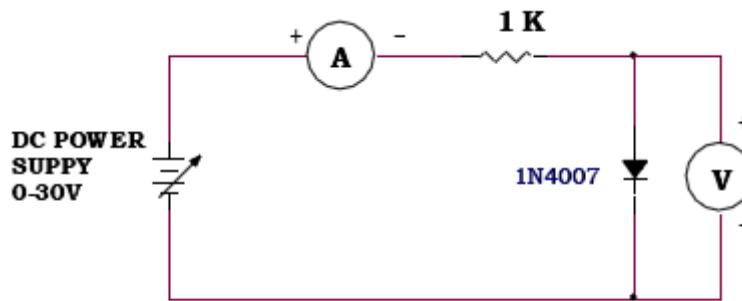
2) Reverse-biased:

If the negative terminal of battery (DC power supply) is connected with P-type terminal of diode and +Ve terminal of battery connected to N type then diode is said to be reverse biased. In this condition the free charge carriers (i.e. electrons in N-type and holes in P-type) will move away from junction widening the depletion region width. The minority carriers (i.e. -ve electrons in p-type and +ve holes in n-type) can cross the depletion region resulting in minority carrier current flow called as reverse saturation current(I_s). As no of minority carrier is very small so the magnitude of I_s is few microamperes. Ideally current in reverse bias is zero.

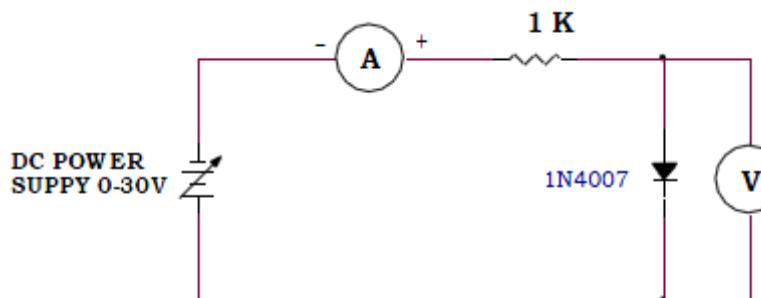
In short, current flows through diode in forward bias and does not flow through diode in reverse bias. Diode can pass current only in one direction.

CIRCUIT DIAGRAM:

Forward bias:



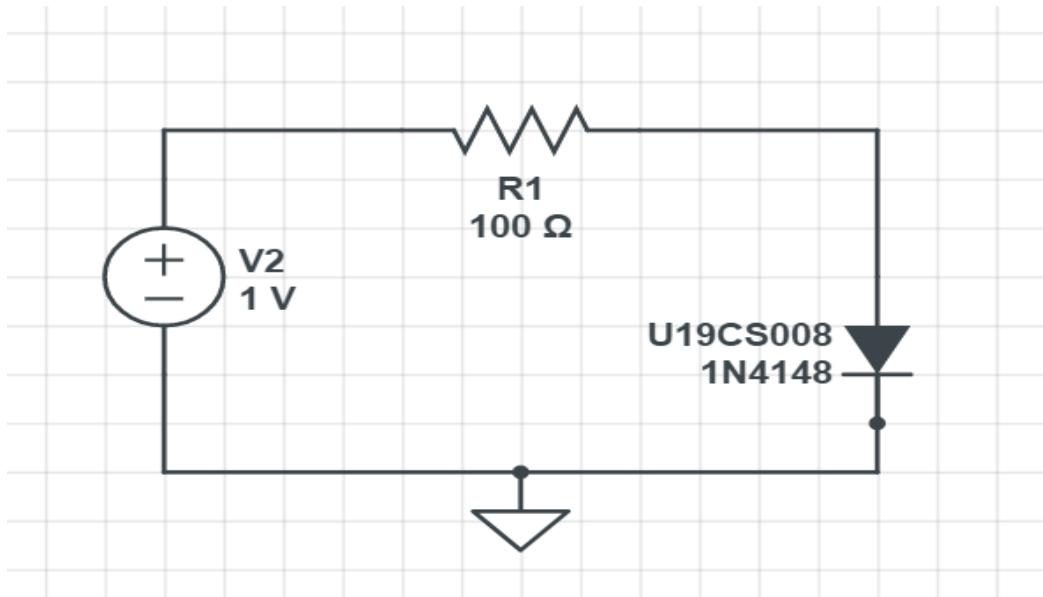
Reverse bias:



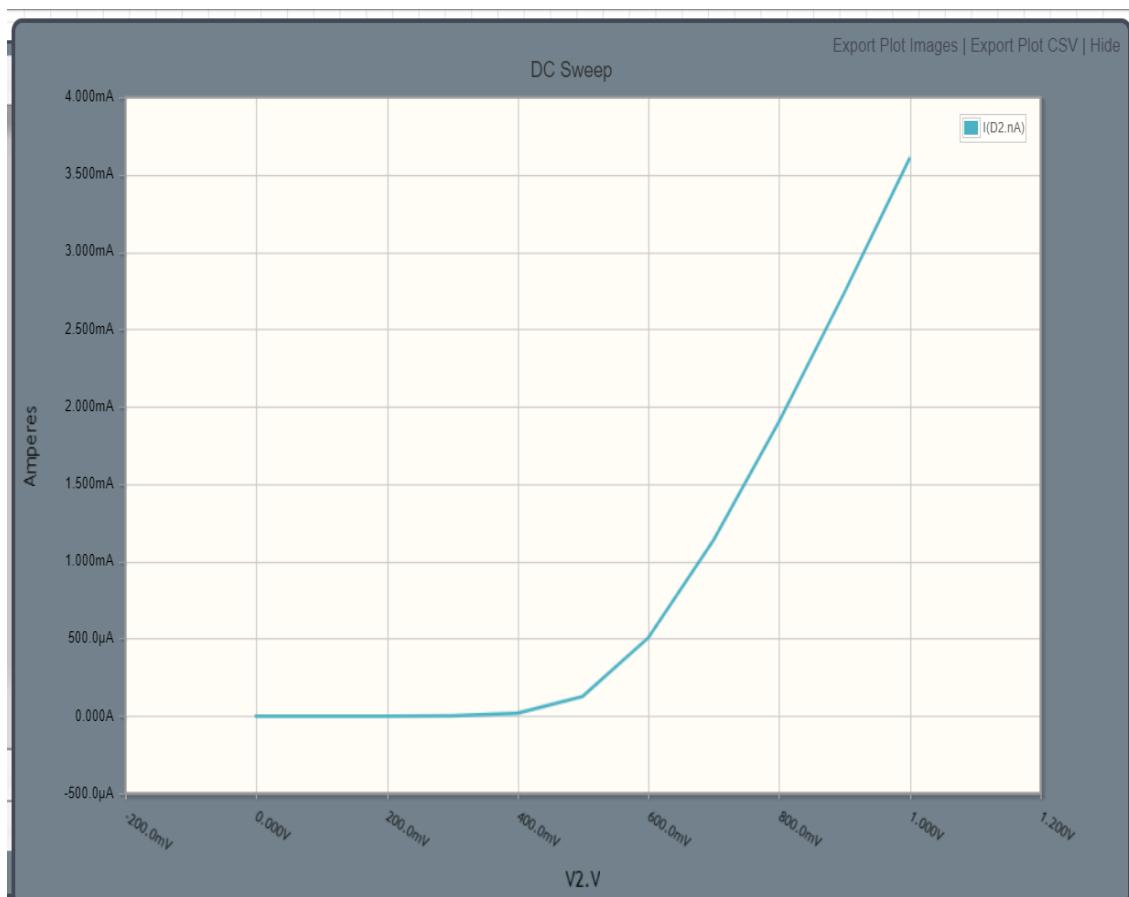


FORWARD CHARACTERISTICS:

CIRCUIT/CONNECTION DIAGRAMS (FROM SIMULATOR)



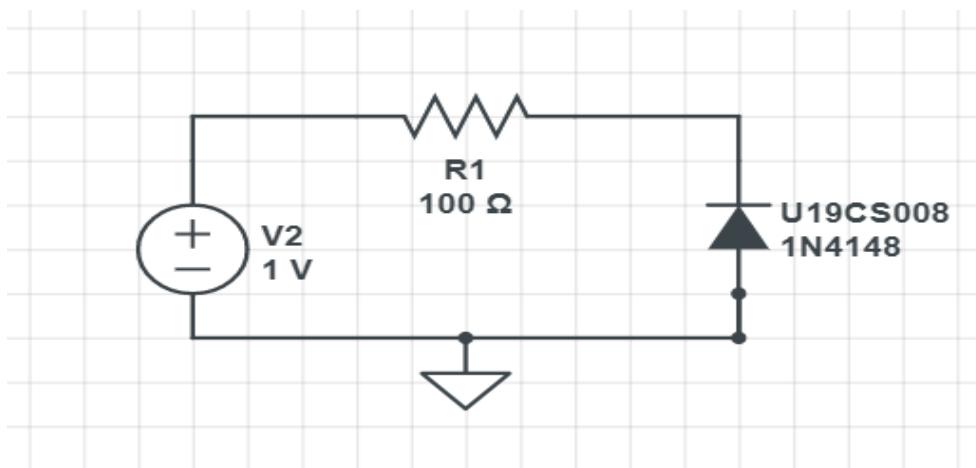
VI-PLOT (FROM SIMULATOR/GRAFH)



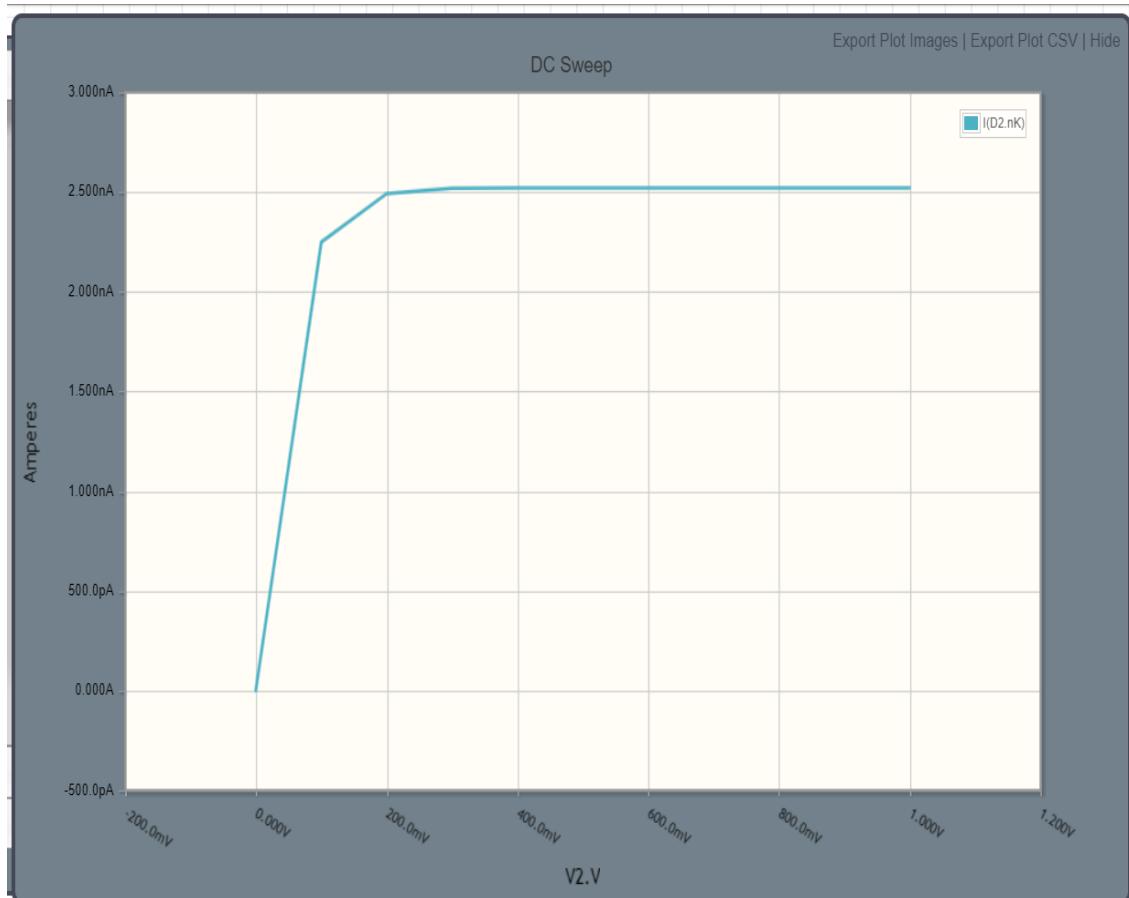


REVERSE CHARACTERISTICS:

CIRCUIT/CONNECTION DIAGRAMS (FROM SIMULATOR)



VI-PLOT (FROM SIMULATOR/GRAF)



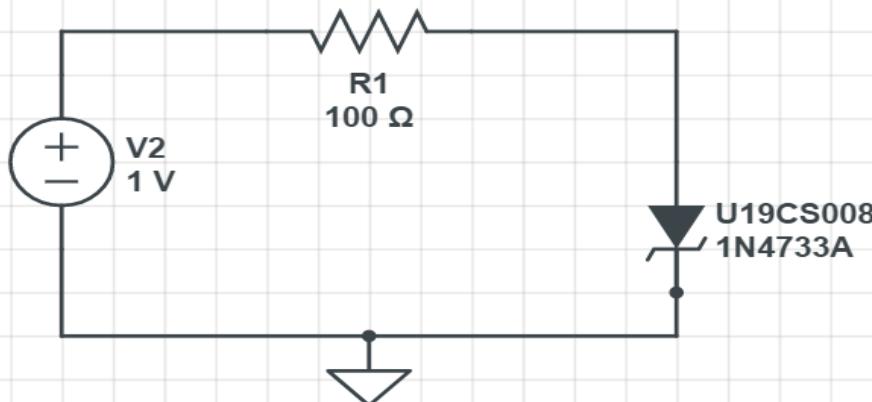
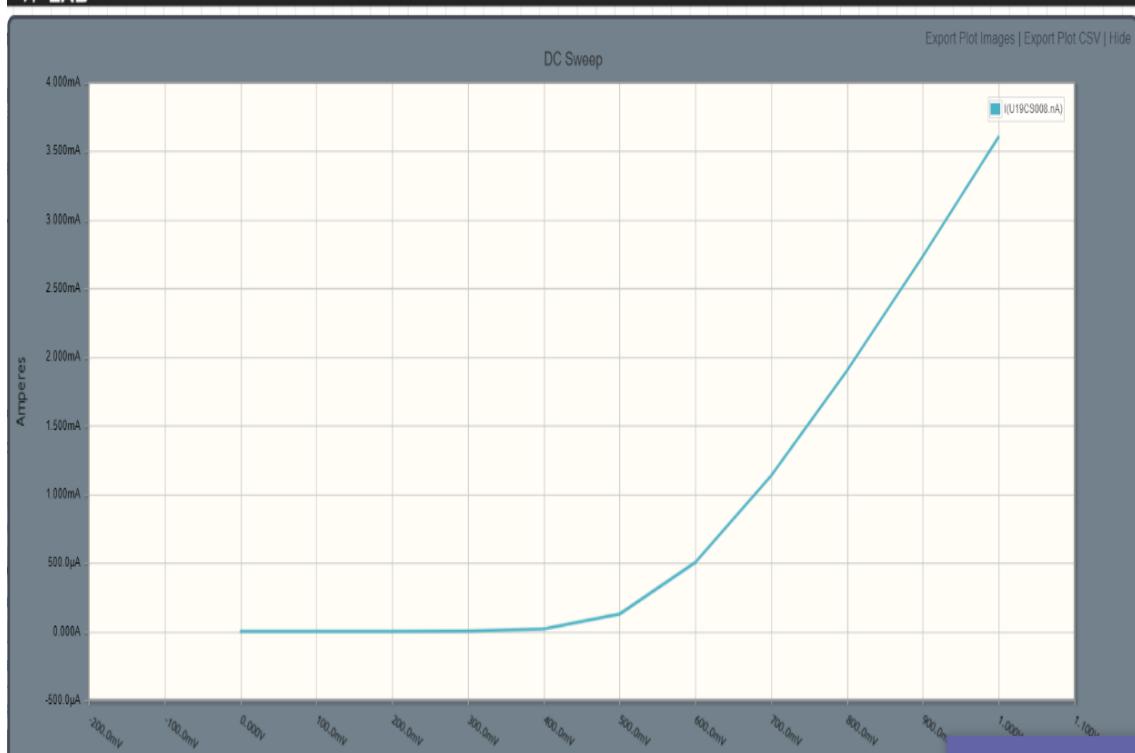


CONCLUSIONS

HERE, THE THEORITICAL AND PRACTICAL (FORWARD AND REVERSE CHARACTERISTGICS OF P-N DIDODE) BOTH ARE SAME. HENCE VERIFIED....

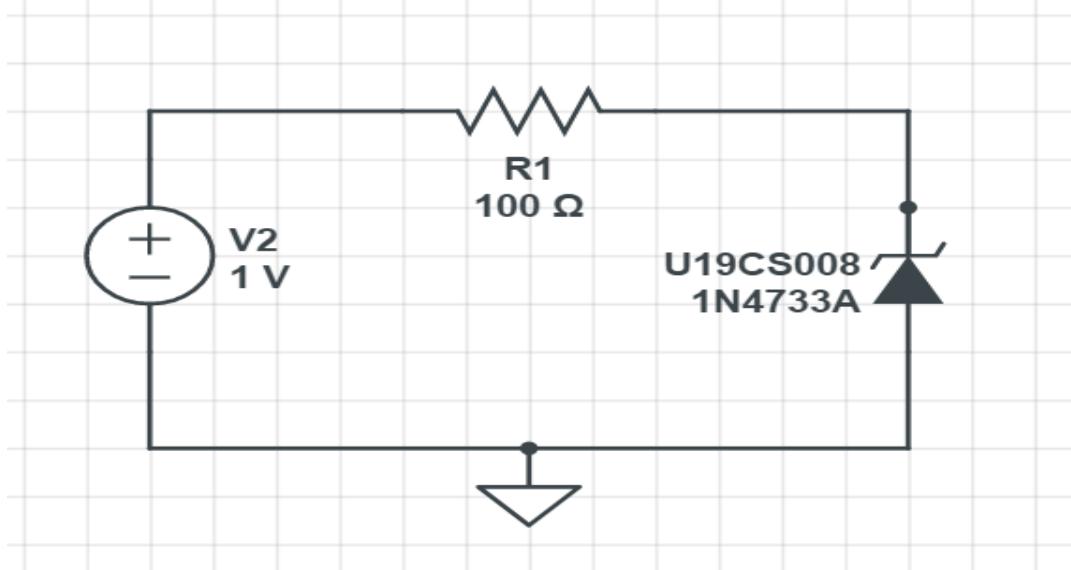
**DLED ASSIGNMENT-5****NAME: KRINA PATEL****ADMISSION NUMBER: U19CS008****QUESTION.**

Obtain and Plot the forward and reverse characteristics of Zener Diode.

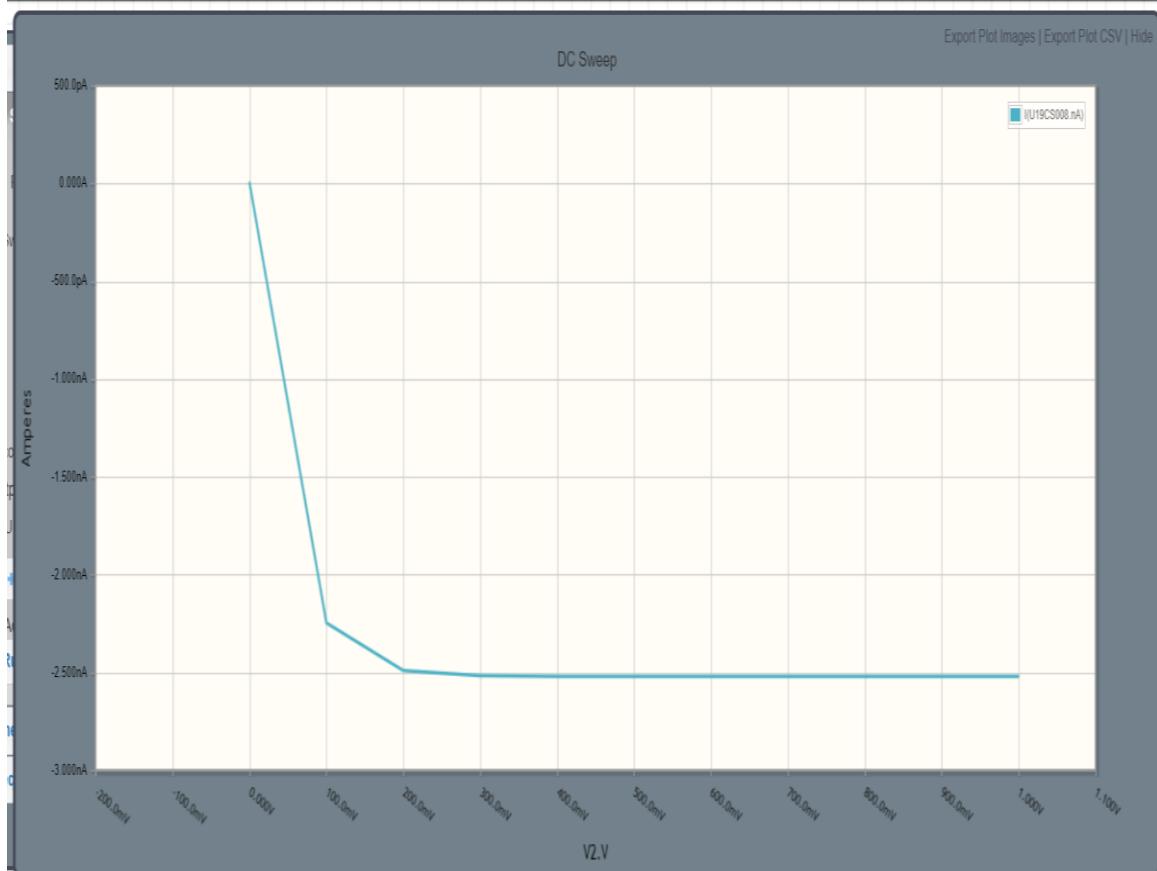
**FORWARD CHARACTERISTICS OF ZENER
DIDODE(CIRCUIT)**

**FORWARD CHARACTERISTICS OF ZENER
DIDODE(OUTPUT):**




REVERSE CHARACTERISTICS OF ZENOR DIODE(CIRCUIT):



REVERSE CHARACTERISTICS OF ZENER DIODE(OUTPUT):





Expt. No:

6

Date:

17-09-2020

Diode Clipper Circuits (Series – Configuration)

AIM: To study, design and plot the various series diode clipper circuits.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator/Circuit Simulator

THEORY:

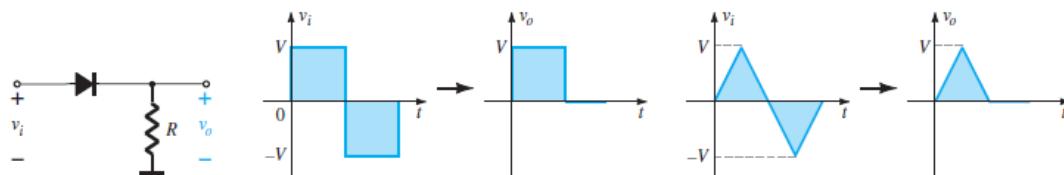
We know that when a diode is forward biased it allows current to pass through itself clamping the voltage across it to 0.7 volts (Practical Silicon Diode). While, when it is reverse biased, no current flows through it and the voltage across its terminals is unaffected, and this is the basic operation of the diode clipping circuit.

Clippers are networks that employ diodes to “clip” away a portion of an input signal without distorting the remaining part of the applied waveform.

There are two general categories of clippers: **Series** and **Parallel**. The series configuration is defined as the one where the diode is in series with the load, whereas the parallel variety has the diode in a branch parallel to the load.

SERIES CONFIGURATIONS

NEGATIVE CLIPPER



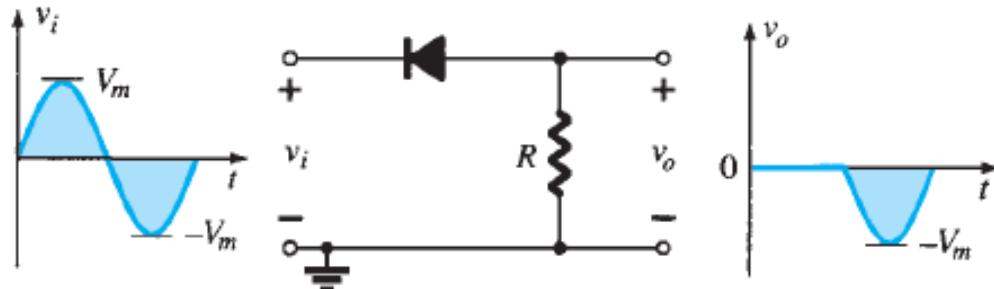
As shown above, when the positive half cycle appears, the diode being forward biased, acts as short circuit and allows the input voltage to appear across the load resistor. During the negative half cycle, the diode is reverse biased, acts as open circuit and hence we see that there is no connection between the output and input node, thereby the output voltage level remains at zero. Since the negative cycle of the input is getting clipped-off, the configuration in the above circuit is known as negative clipper.

Likewise when the polarity of the diode is reversed, we can clip off the positive half of the input cycle. In this case, during the positive half cycle, the diode remains reverse biased thereby disconnecting the output node from input node and the output voltage level remains at zero. But when the negative half cycle appears, the diode gets forward biased and allows the entire input to appear across the output load resistance.

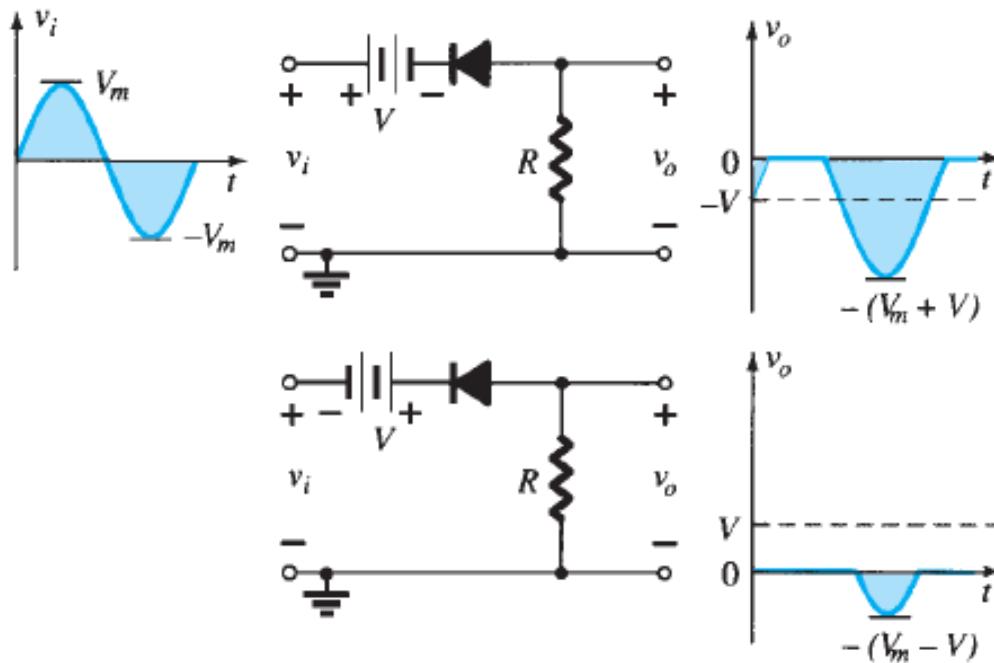


FEW SERIES DIODE CLIPPER CONFIGURATIONS

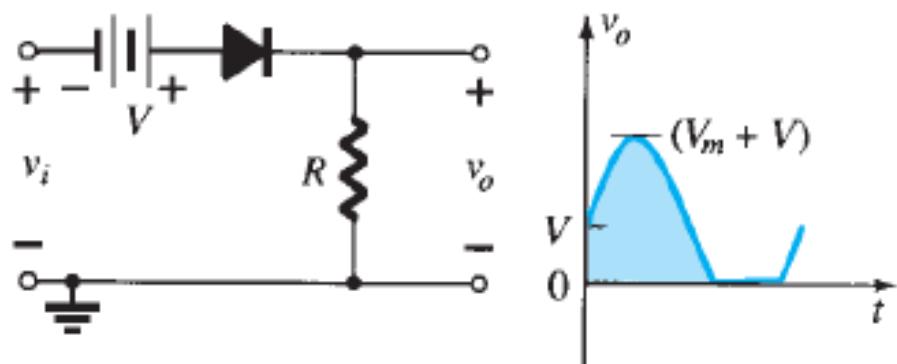
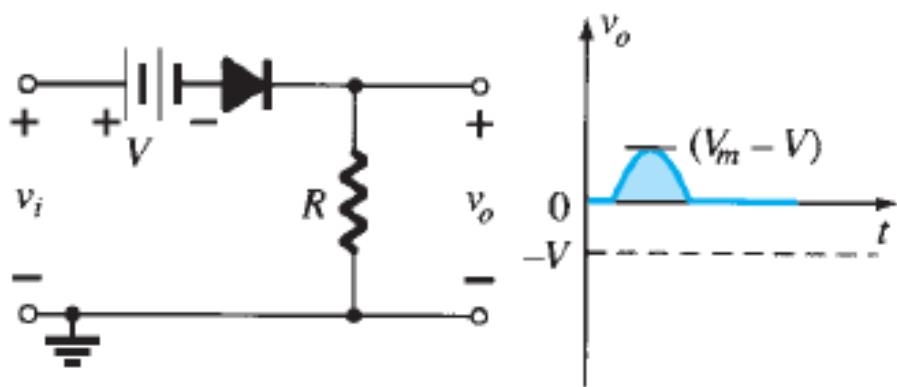
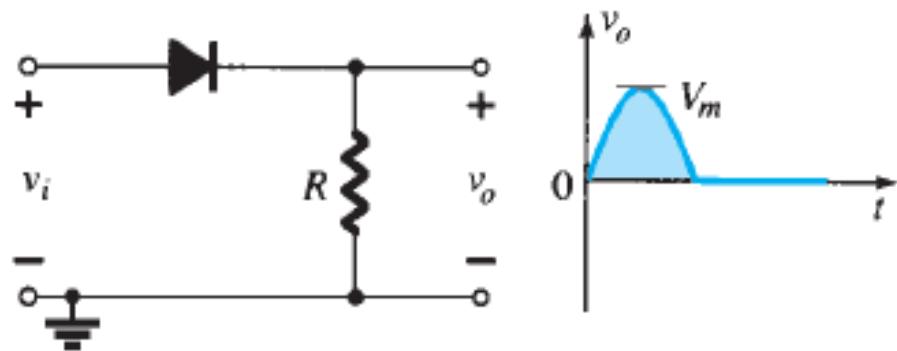
POSITIVE



Biased Series Clippers (Ideal Diodes)



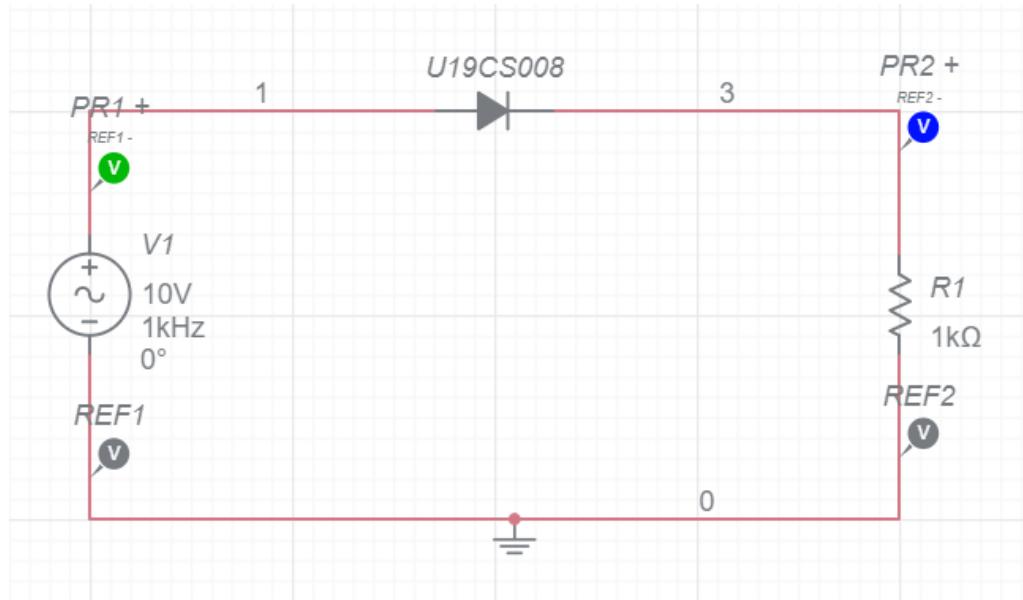
NEGATIVE



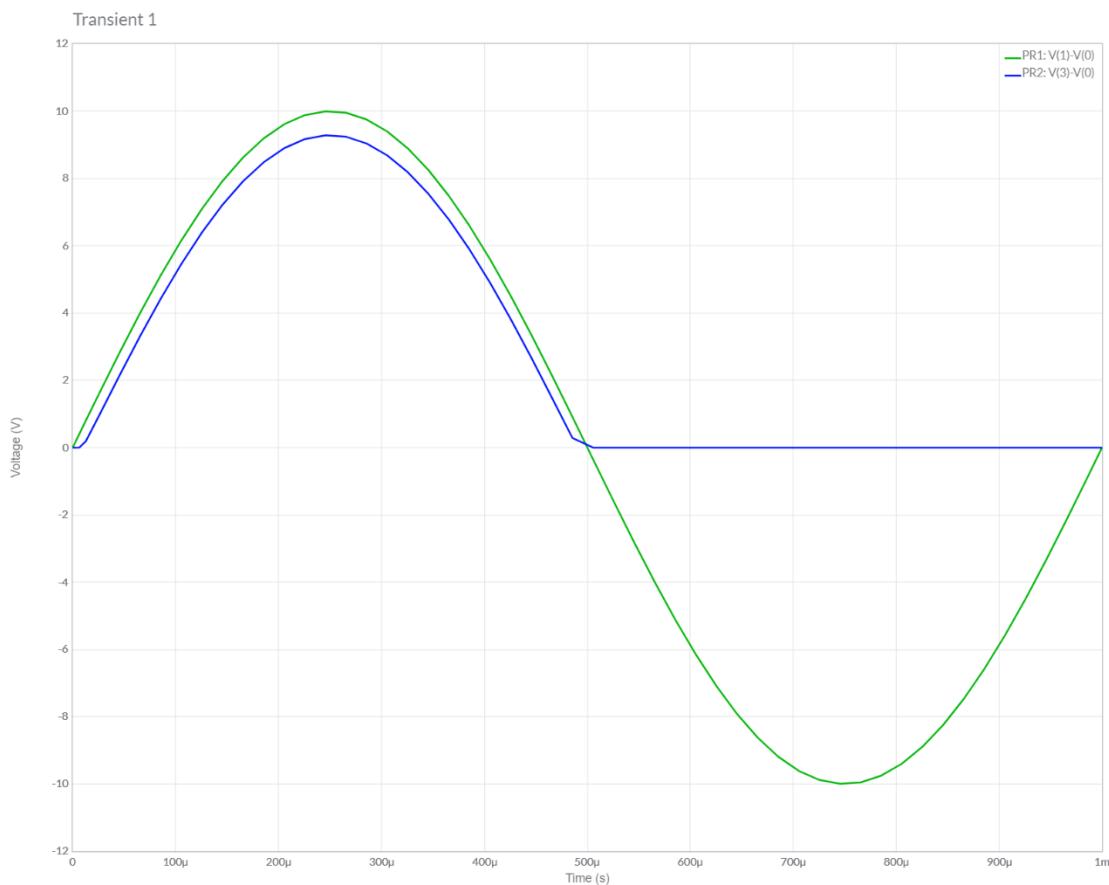


CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

NEGATIVE SERIES CLIPPER



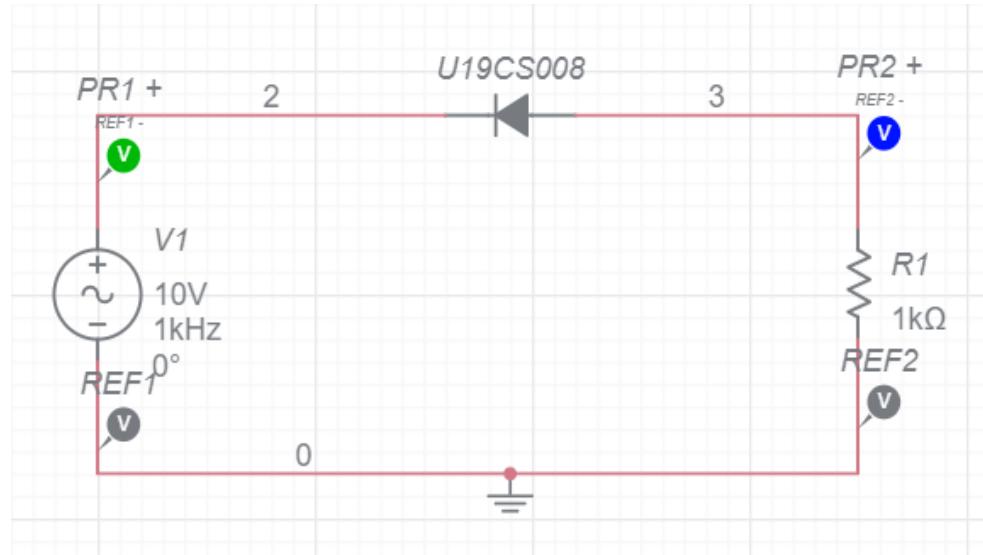
WAVEFORMS (FROM MULTISIM)



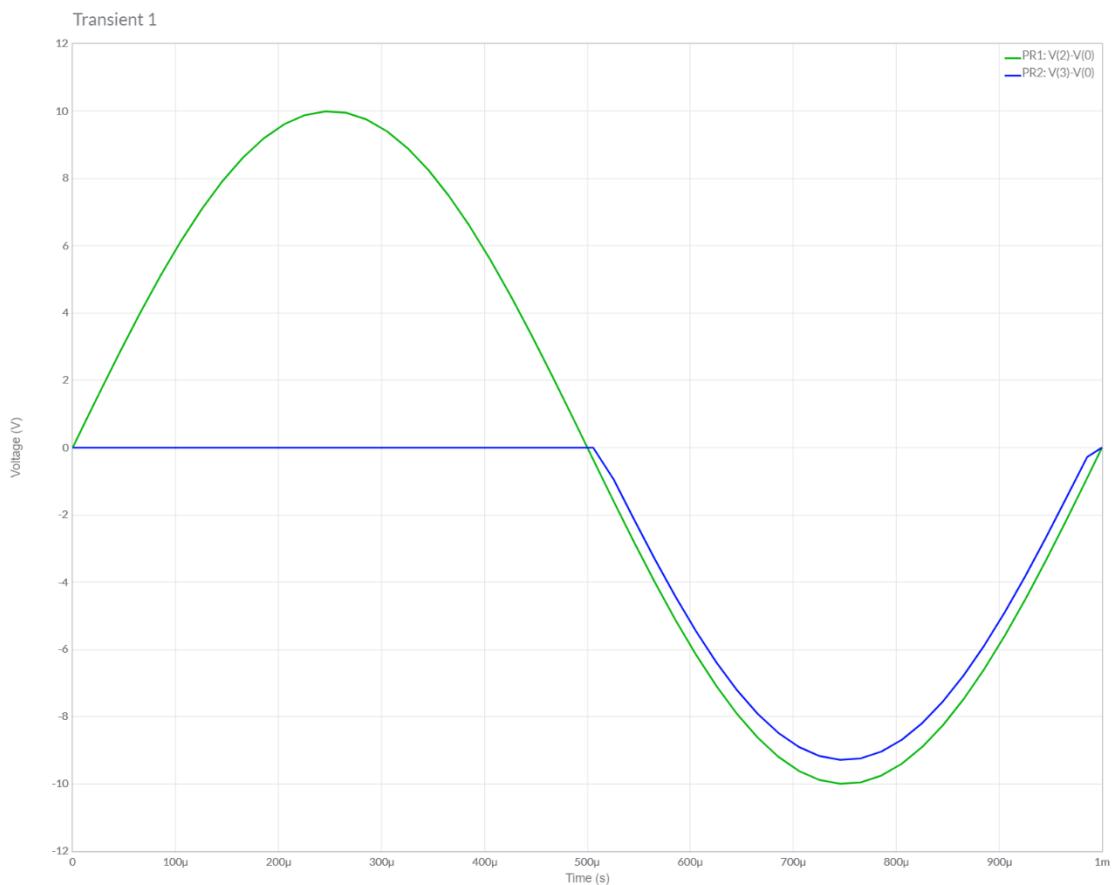


CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

POSITIVE SERIES CLIPPER



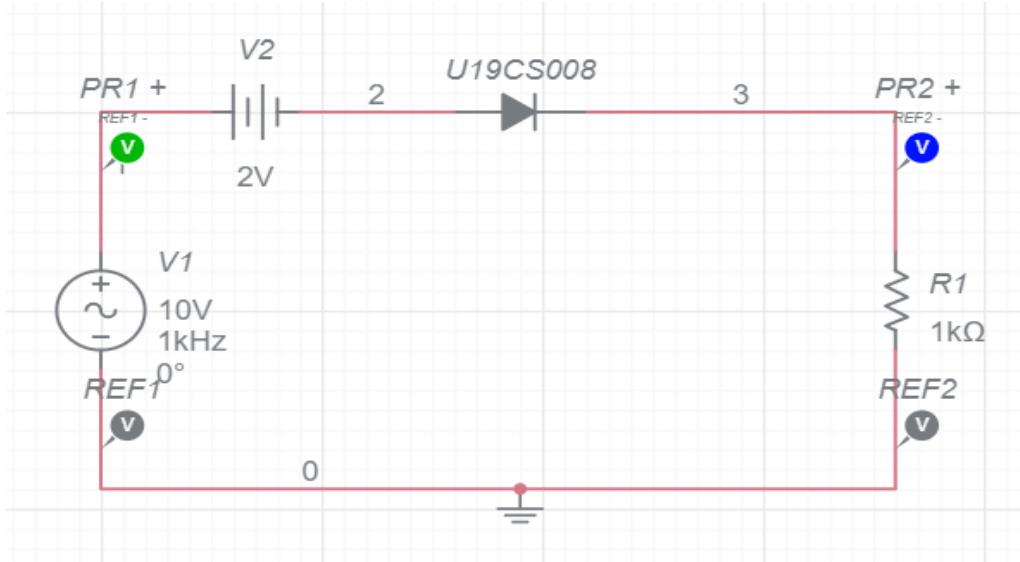
WAVEFORMS (FROM MULTISIM)



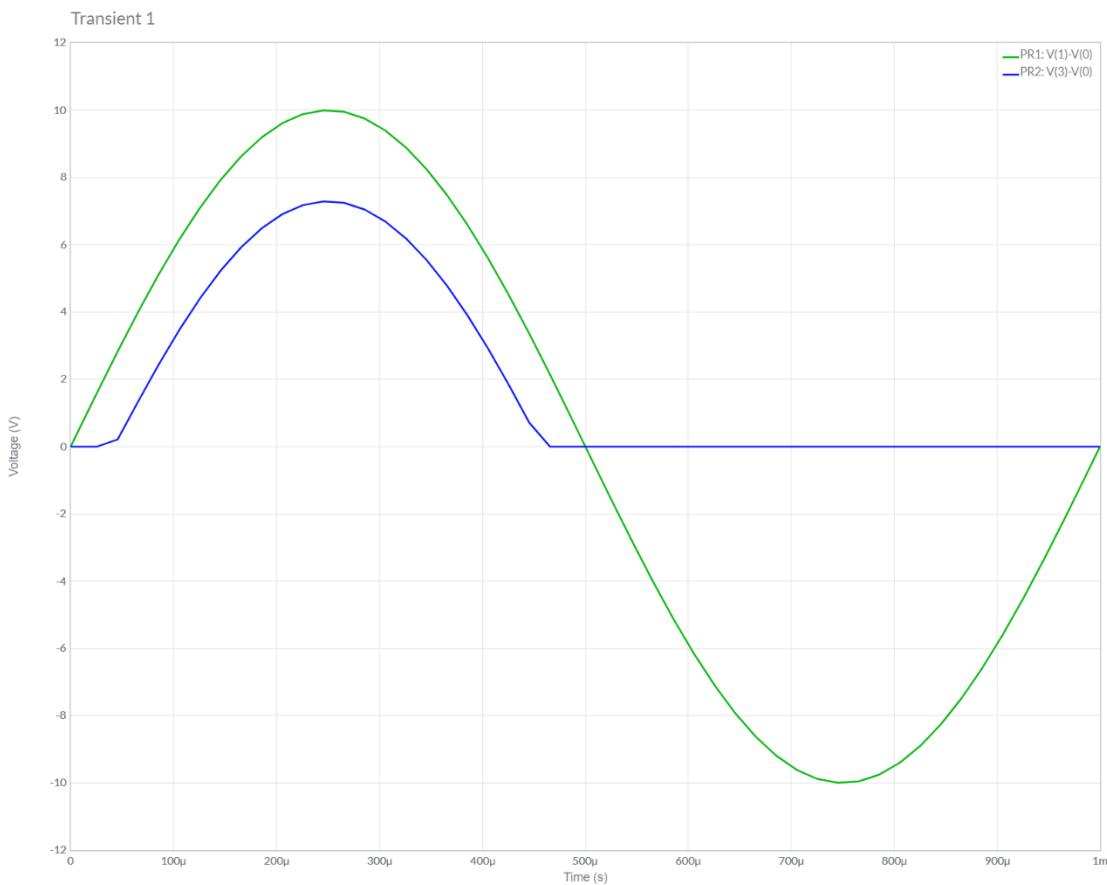


CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

NEGATIVE SERIES CLIPPER WITH BIAS-I



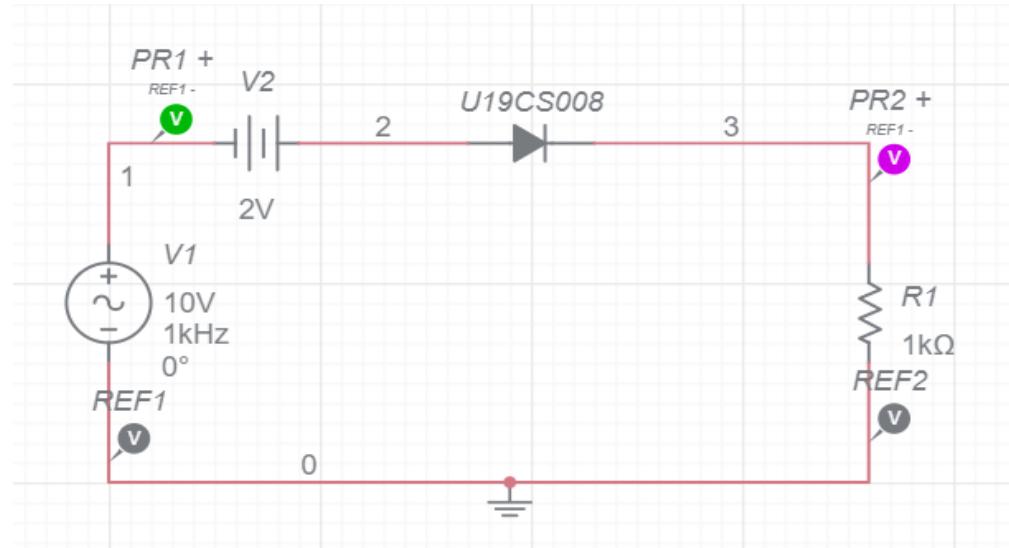
WAVEFORMS (FROM MULTISIM)



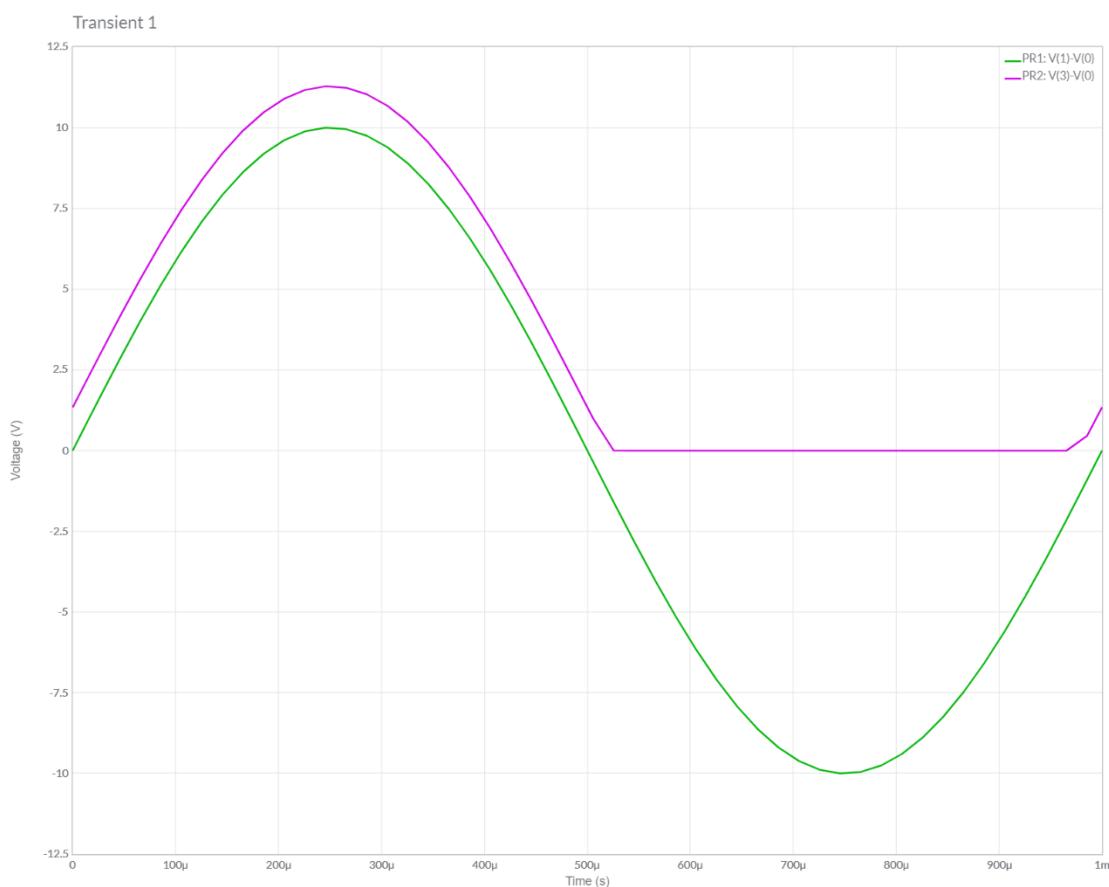


CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

NEGATIVE SERIES CLIPPER WITH BIAS-II



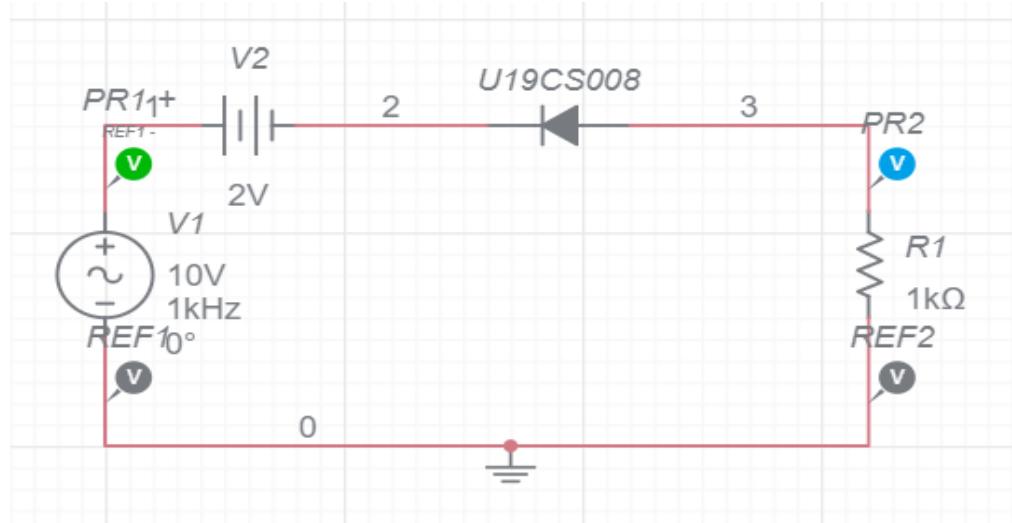
WAVEFORMS (FROM MULTISIM)



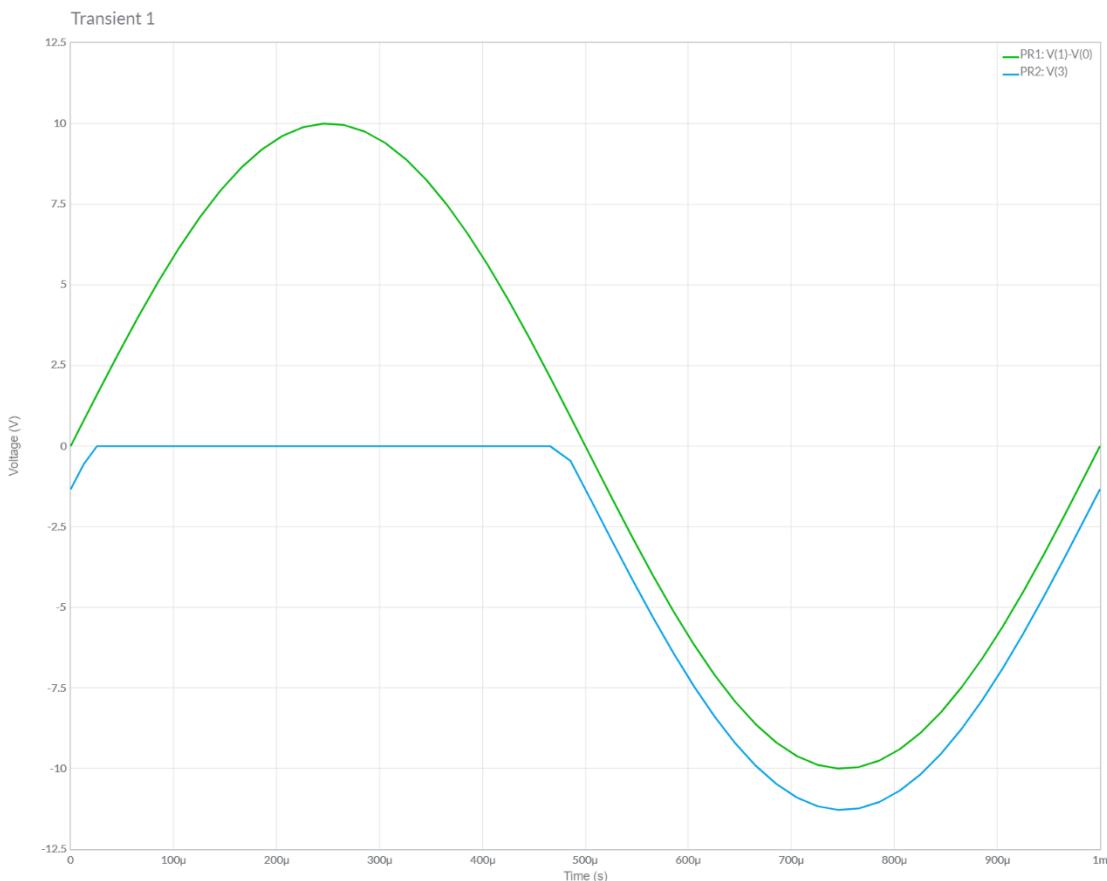


CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

POSITIVE SERIES CLIPPER WITH BIAS-I



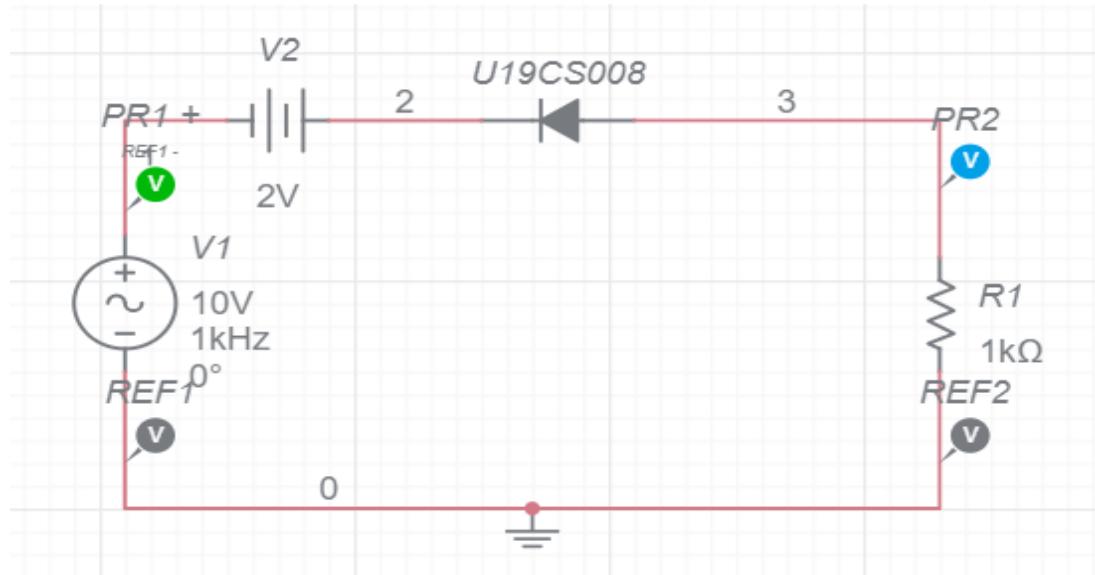
WAVEFORMS (FROM MULTISIM)



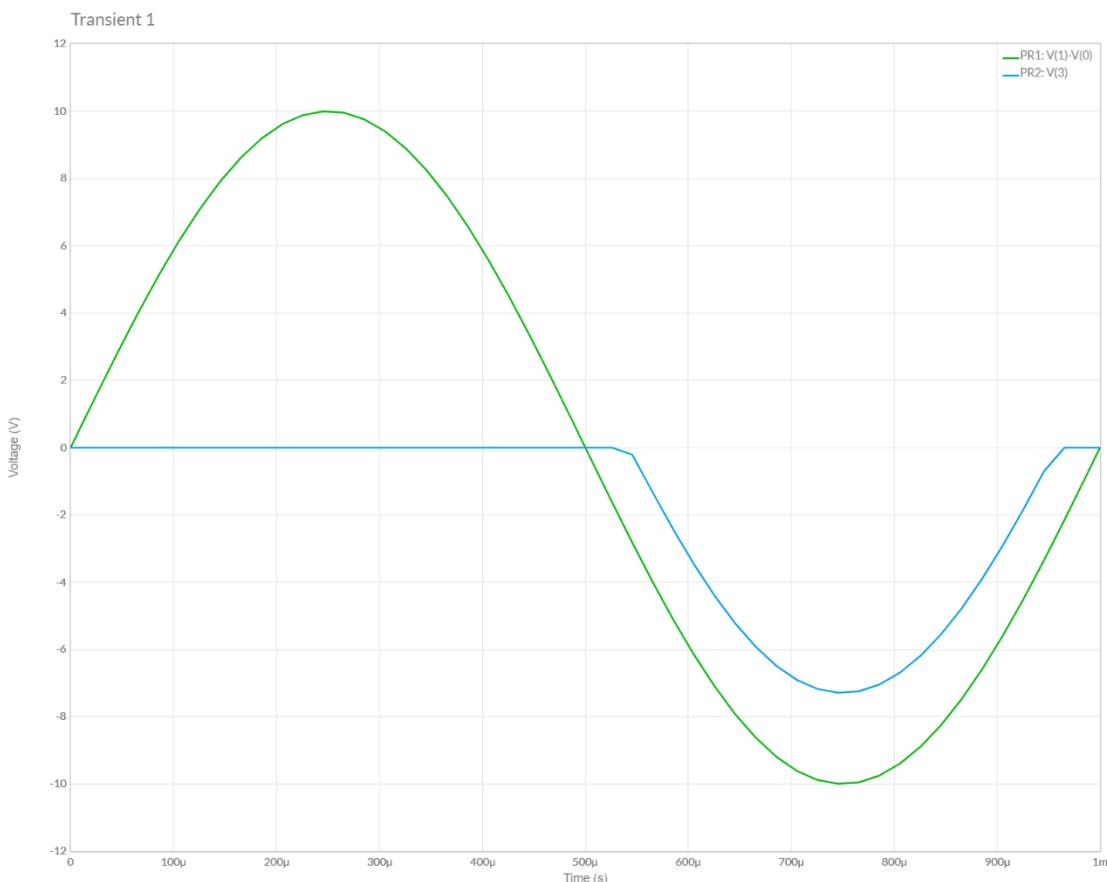


CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

POSITIVE SERIES CLIPPER WITH BIAS-II



WAVEFORMS (FROM MULTISIM)





CONCLUSIONS

**HERE, THE THEORITICAL AND PRACTICAL CHARACTERISTICS OF
VERIES NEGATIVE AND POSITIVE SERIES CLIPPER (WITH AND
WITHOUT BAISED) ARE SAME. HENCE VERIFIED.....**



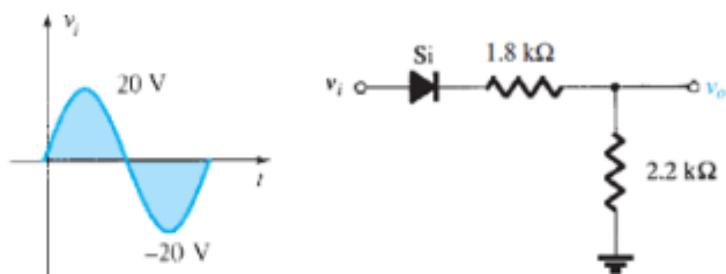
DLED ASSIGNMENT-6

NAME: KRINA PATEL

ADMISSION NUMBER: U19CS008

QUESTION-1

1. Determine and plot the output voltage for the given circuit. Also verify the same using Multisim.





CALCULATION AND THEORITICAL OUTPUT VOLTAGE:

Q-1)

calculation:

using ~~Kirchhoff's law~~:

$$\rightarrow V_i = 0.7 + (1.8)I + (2.2)I = 0$$

~~$\therefore 2.0 + 0.7 = (4.0)I$~~

~~$I = \frac{2.7}{4.0}$~~

$$I = \frac{19.3}{4.0} \text{ mA}$$

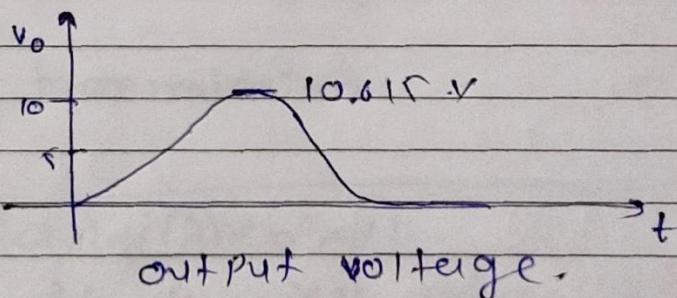
$$I = 4.825 \text{ mA}$$

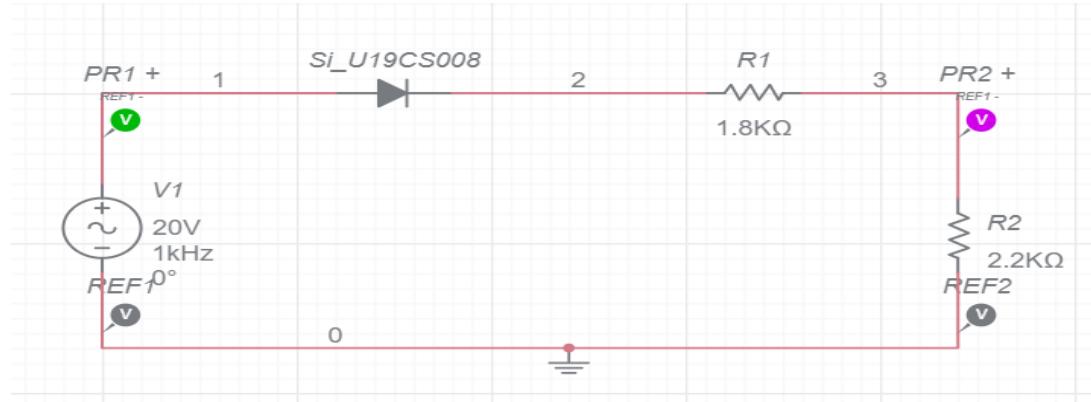
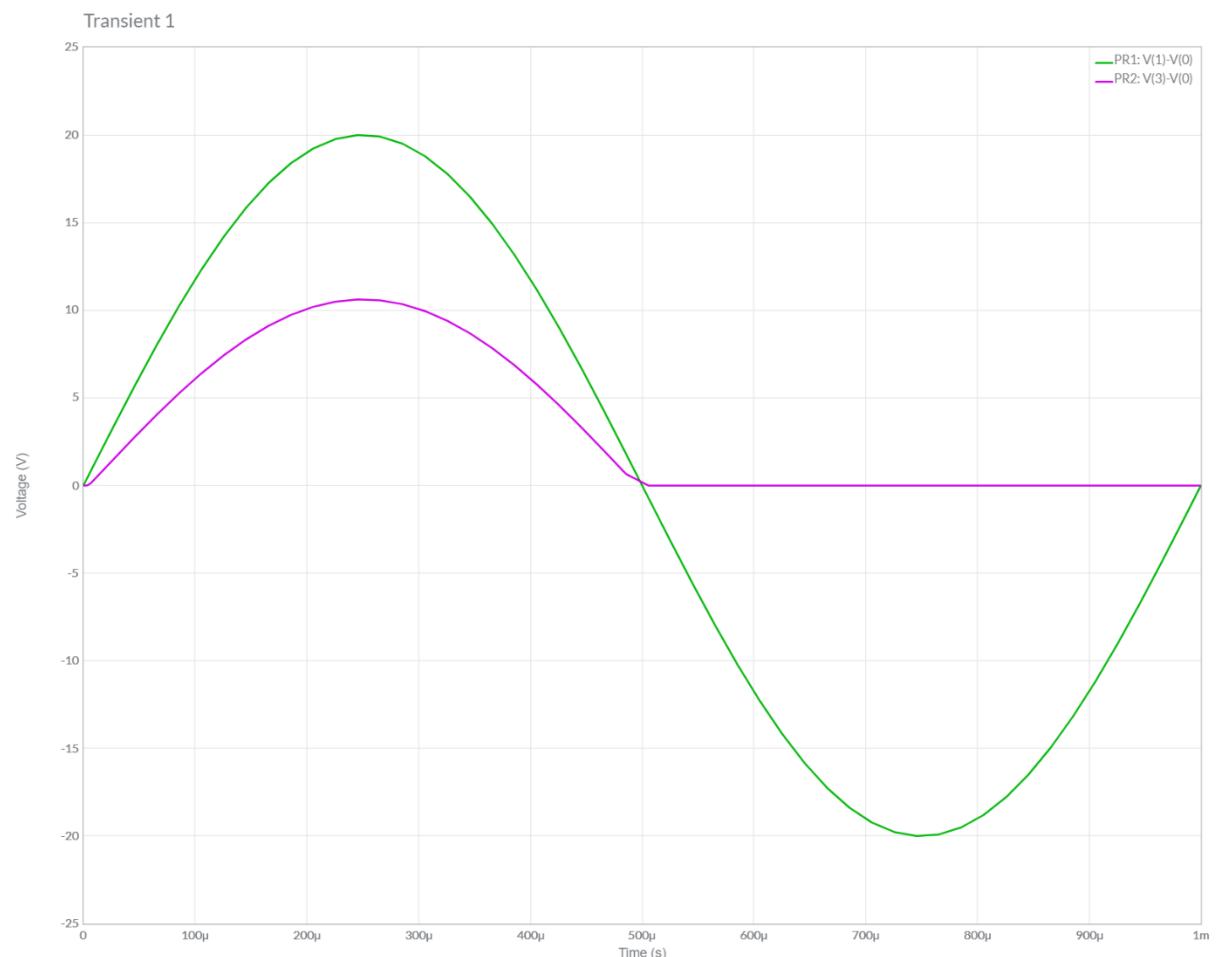
$$\therefore V_o = (2.2 \text{ k}\Omega) \cdot I$$

$$= (2.2 \times 4.825)$$

$$\boxed{V_o = 10.615 \text{ V}}$$

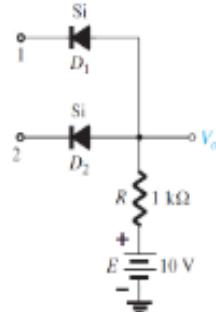
∴ Output voltage is 10.615 V.

graph

**MULTISIM CIRCUIT:****GRAPH:**

**QUESTION-2**

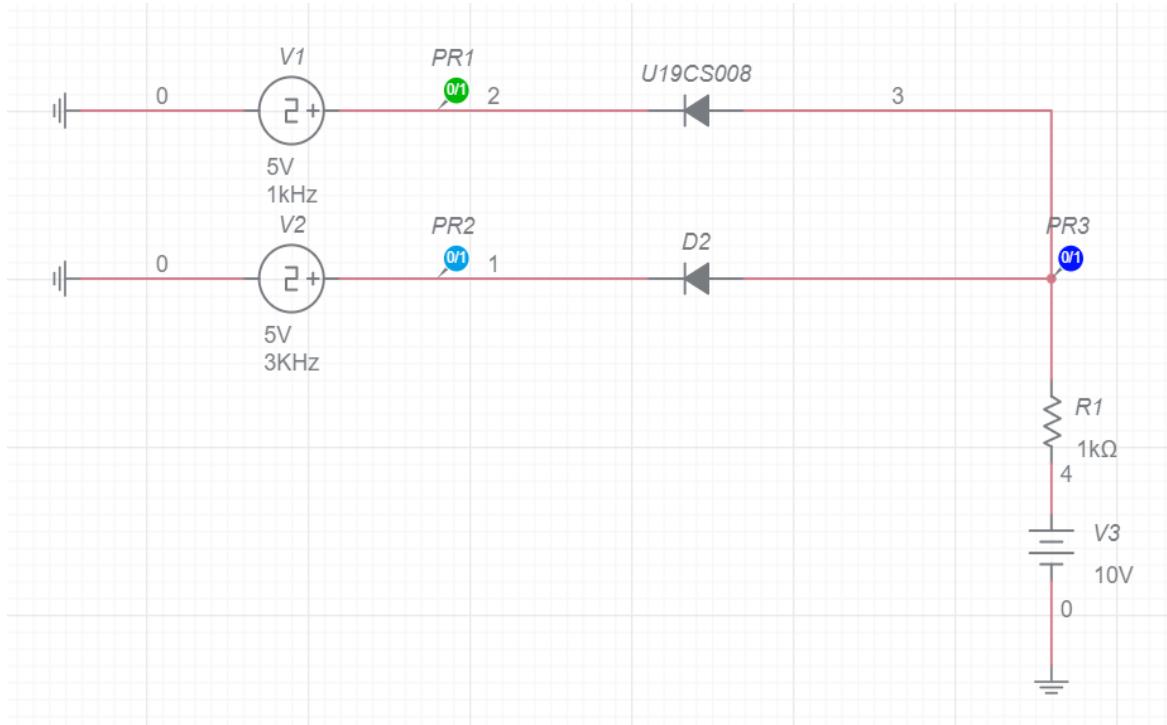
- 2.** Identify the type of Logic Gate implemented by the below diode configuration.
Also verify it using multisim.

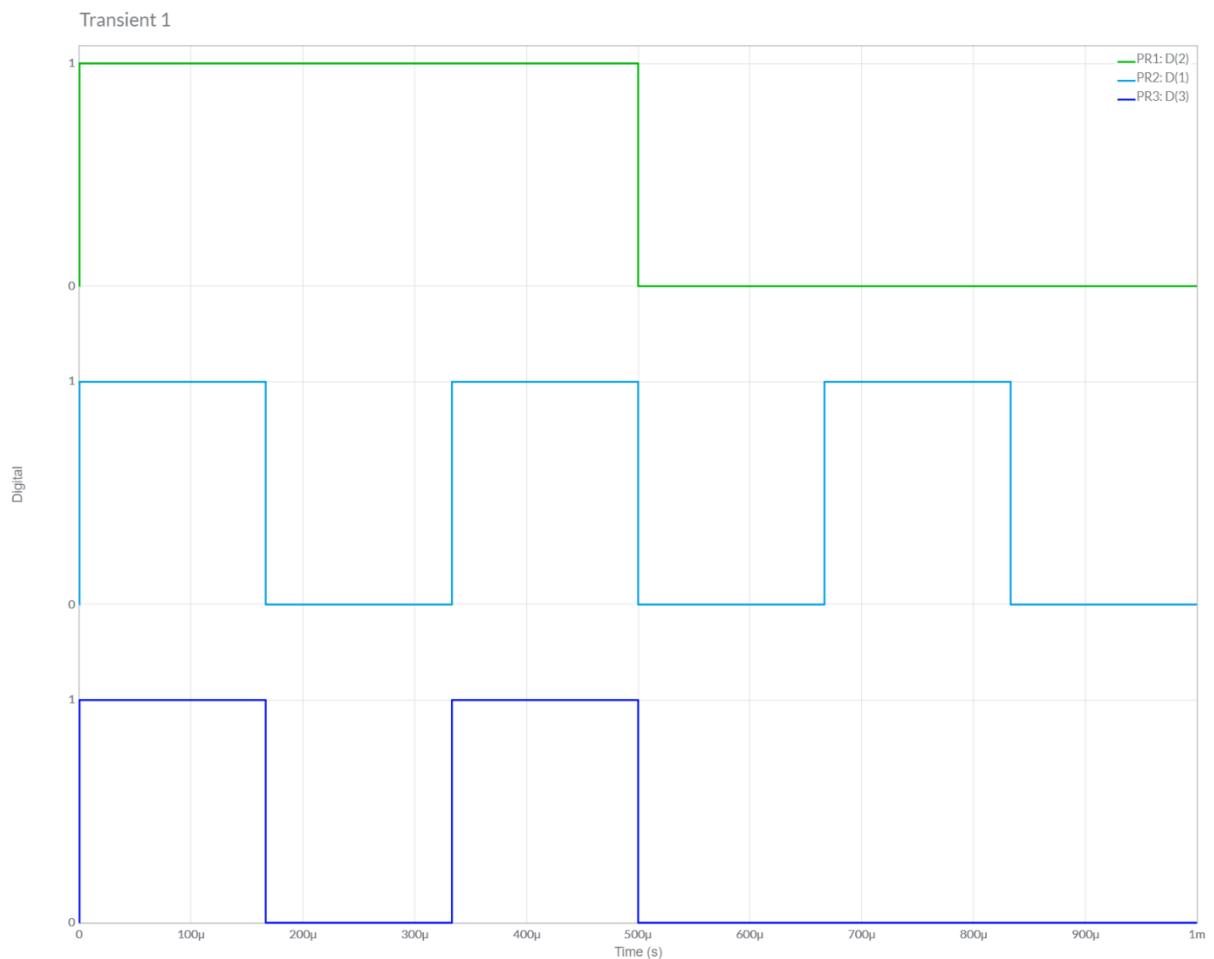
**ANALYSIS:**

When both inputs are at 10V , the two diodes are reverse biased and there is no current flowing to ground. Therefore the output is logic “1” because there is no voltage drop across the resister R.

If one of the inputs is 0V , the current will flow through the corresponding diode and through the resistor. Thus the diode anode (the output) will be logic “0”.

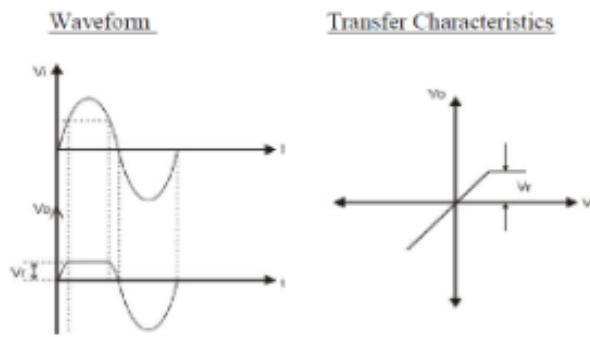
Thus, this circuit is behave like AND gate.

CIRCUIT:

**GRAPH:****QUESTION-3**

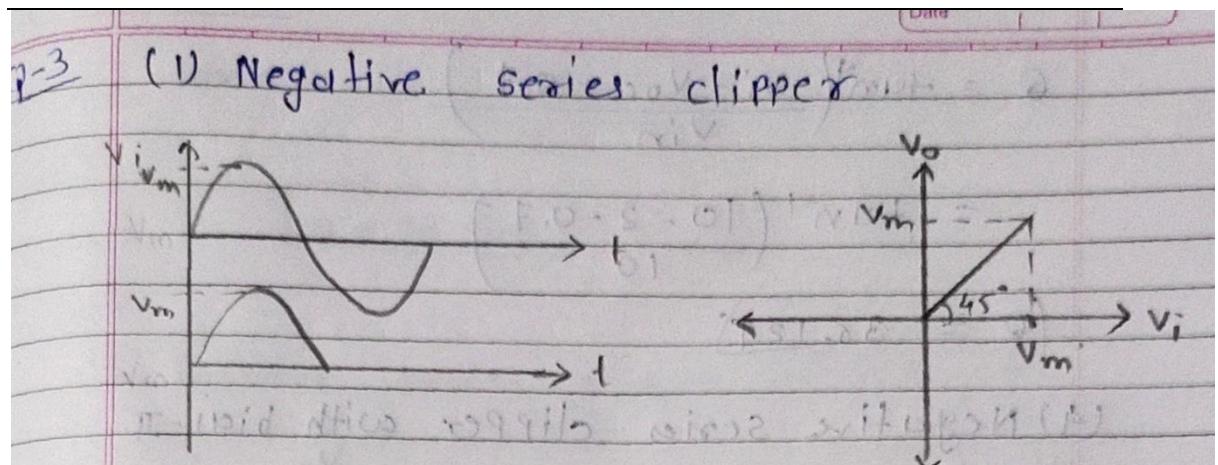
3. Draw the transfer characteristics for all the clipper configurations which are part of your today's practical (Practical - 6).

Ref. Example



ASSUMING THAT;

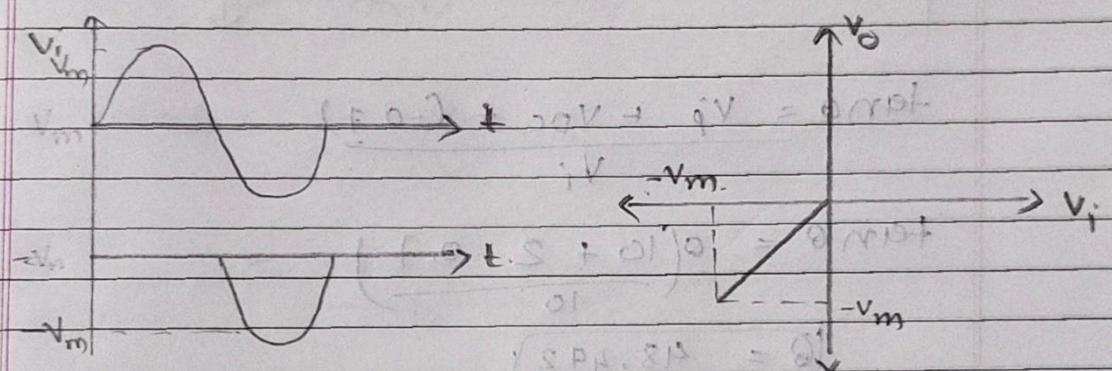
Vi=10V, 10KHz DC Voltage=2V



$$\tan \theta = \frac{V_i - 0.7}{V_i} = \frac{10 - 0.7}{10}$$

$$\therefore \theta = \tan^{-1}(0.93) = 42.9228^\circ$$

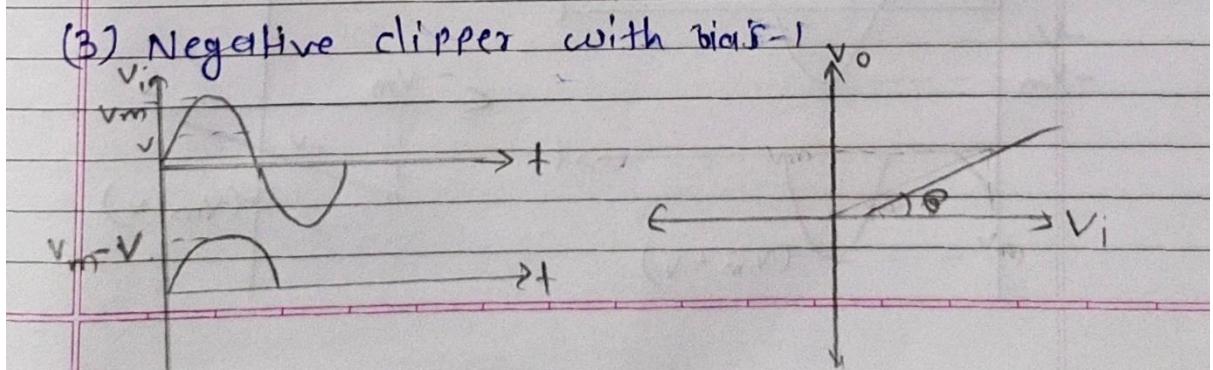
(2) Positive series clipper.



$$\tan \theta = \frac{V_i - 0.7}{V_i} = \frac{10 - 0.7}{10}$$

$$\therefore \theta = 42.9228^\circ$$

(3) Negative clipper with bias -1





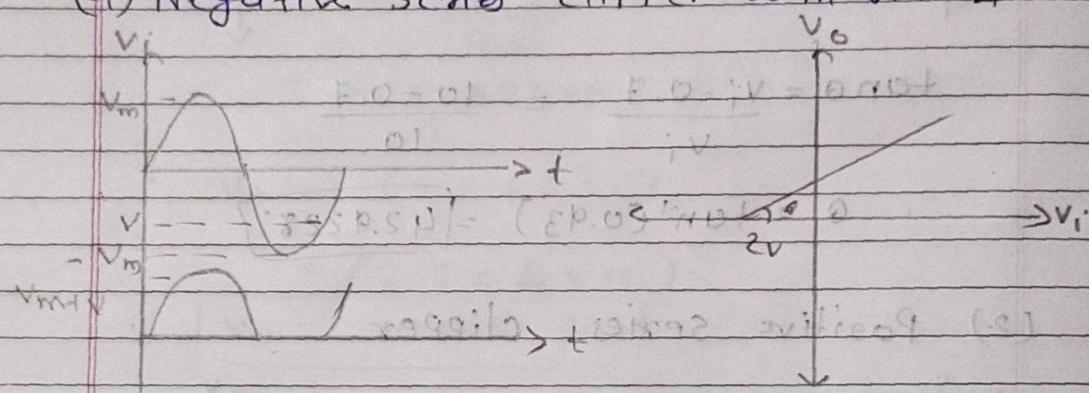
Date _____

$$\theta = \tan^{-1} \left(\frac{v_i - v_{DC} - 0.7}{v_i} \right)$$

$$= \tan^{-1} \left(\frac{10 - 2 - 0.7}{10} \right)$$

$$\theta = 36.129^\circ$$

(4) Negative series clipper with bias-II

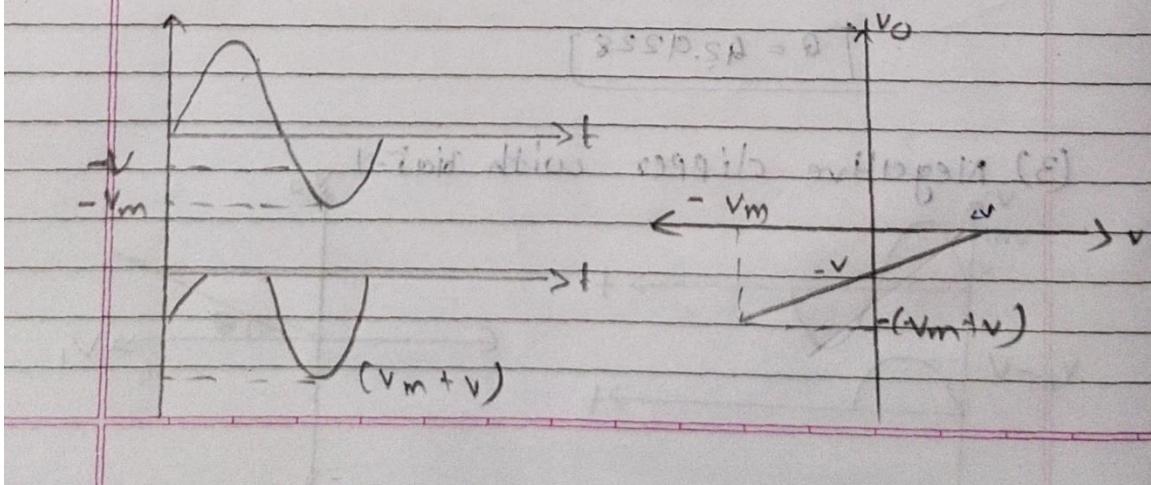


$$\tan \theta = \frac{v_p + v_{DC} + (-0.7)}{v_i}$$

$$\tan \theta = \frac{0(10 + 2 - 0.7)}{10}$$

$$\theta = 48.492^\circ$$

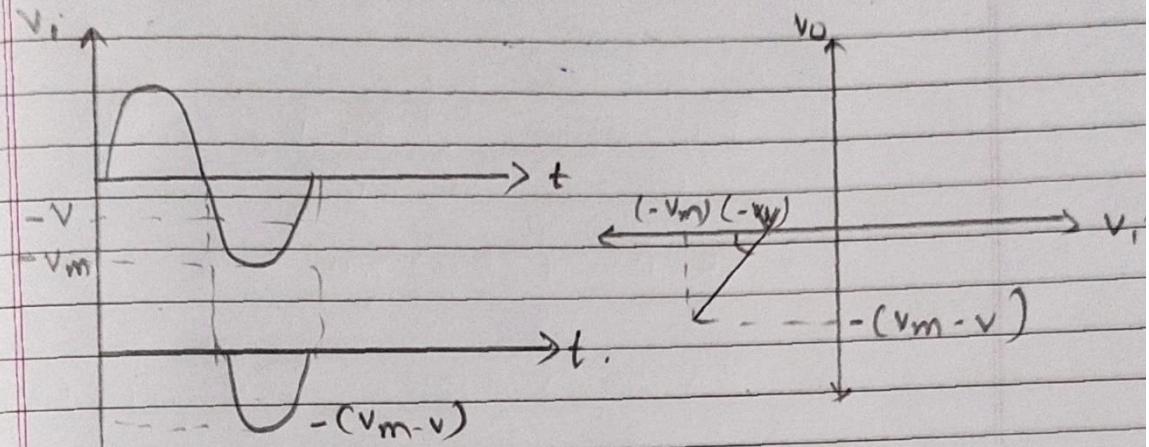
(5) Positive Series clipper with bias-I





$$\tan \theta = \frac{V_i - 2V_{DC} - 0.7}{V_{in}}$$

$$\theta = 36.129^\circ$$

(6) Positive series clipper with bias - π 

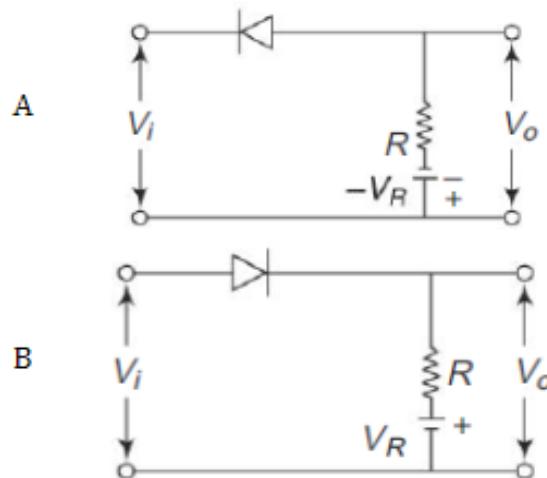
$$\tan \theta = \frac{V_i + V_{DC} - 0.7}{V_i}$$

$$= \frac{10 + 2 - 0.7}{10}$$

$$\theta = 48.492^\circ$$

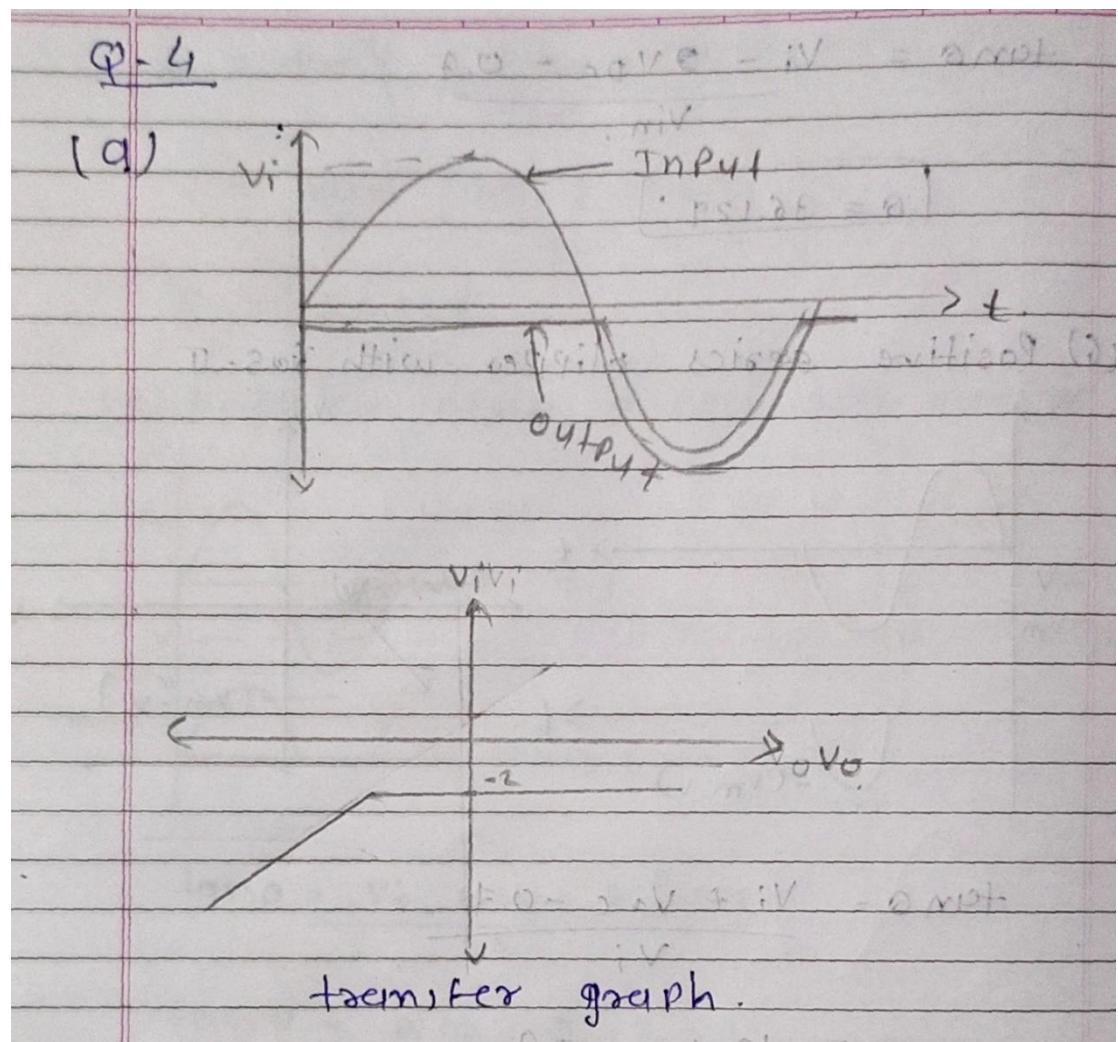
**QUESTION-4**

4. Assuming Symmetrical Sine wave input with peak value greater than the reference voltage, predict the output and plot the Transfer Characteristics for the following Clipper Circuits:

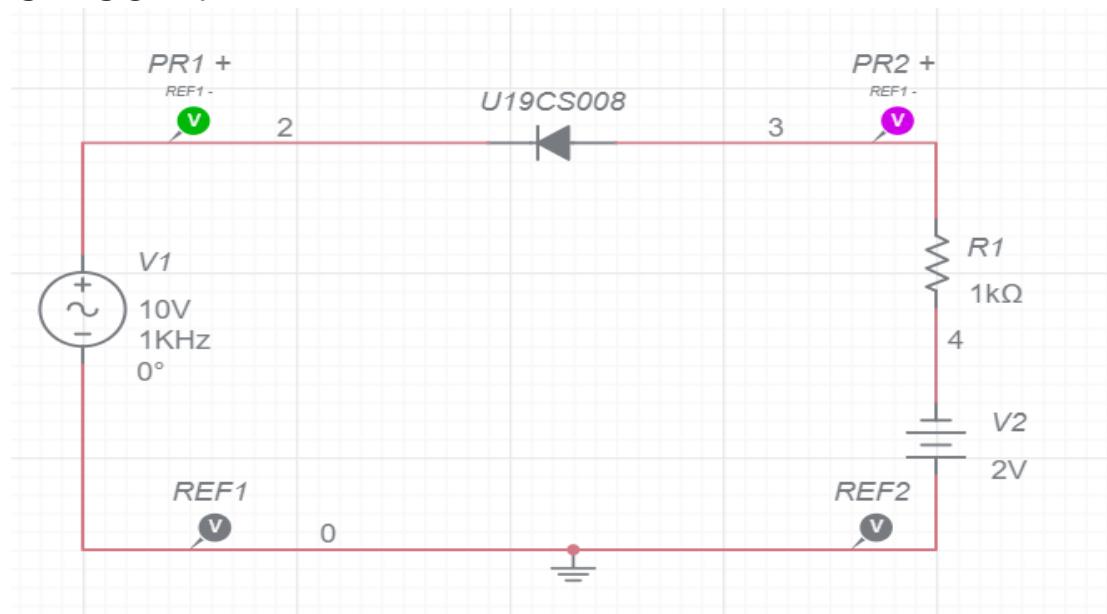


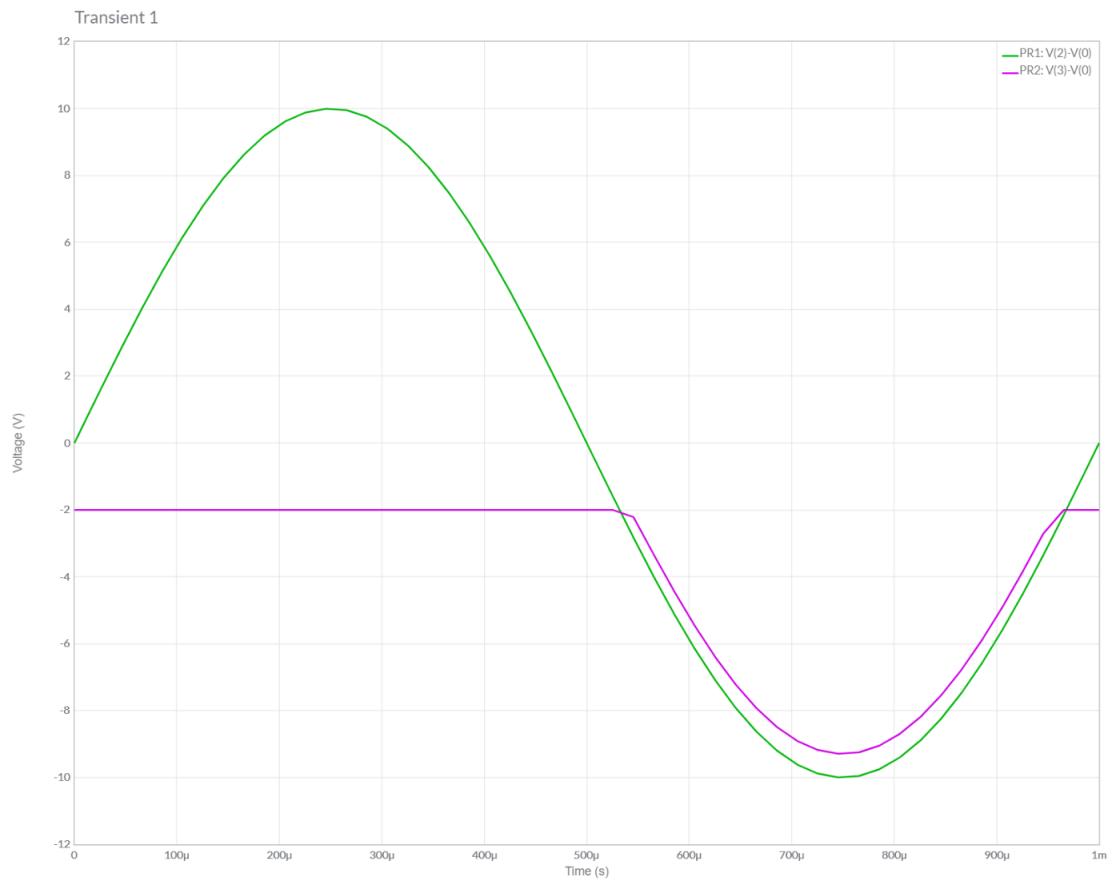


A) PREDICTED OUTPUT AND TRANSFER DIAGRAM:



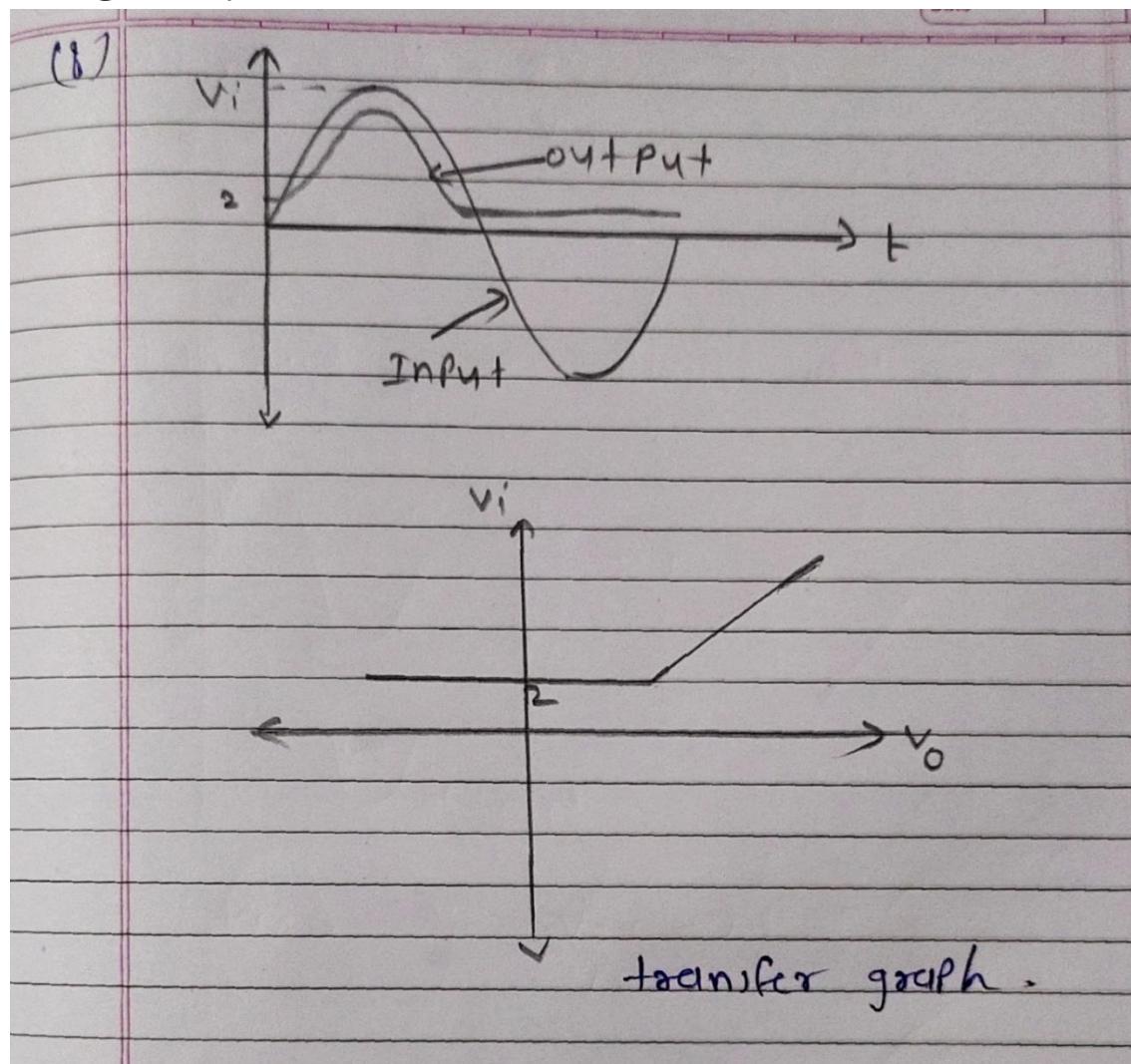
CIRCUIT:



**GRAPH:**

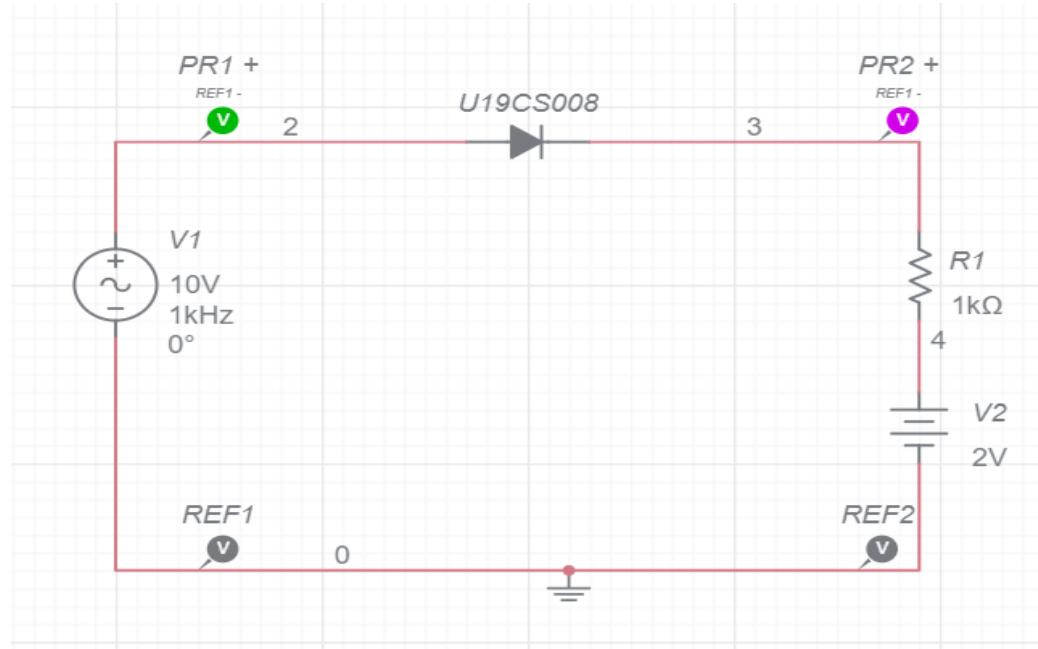
**B)**

**PREDICTED OUTPUT AND TRANSFER
DIAGRAM:**

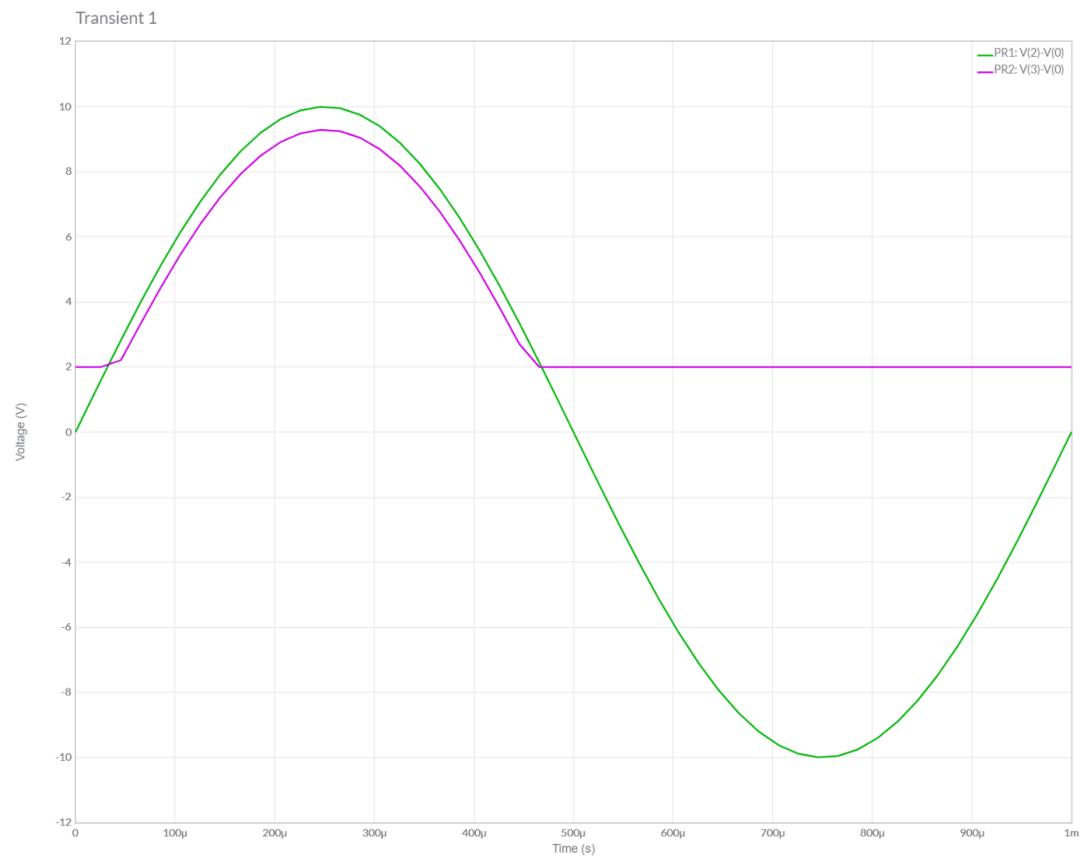




CIRCUIT:



GRAPH:



**HERE, IN QUESTION-1 , QUESTION-2 AND QUESTION-4
BOTH THEORITICAL AND PRACTICAL VALUES AND
WAVEFORMS ARE SAME. HENCE VERIFIED...**



Expt. No:

7

Date:

24-09-2020

Diode Clipper Circuits (Shunt – Configuration)

AIM: To study, design and plot the various shunt diode clipper circuits.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

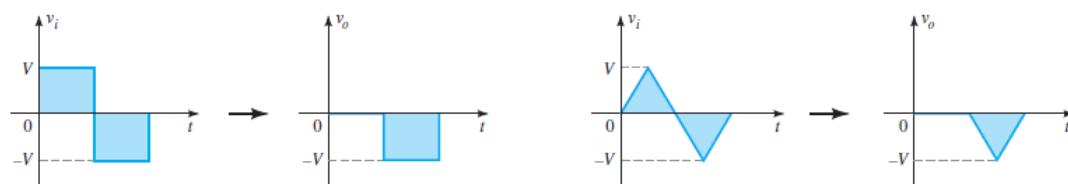
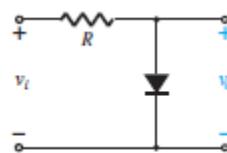
1. Multisim Simulator/Circuit Simulator

THEORY:

We know that when a diode is forward biased it allows current to pass through itself clamping the voltage across it to 0.7 volts (Practical Silicon Diode). While, when it is reverse biased, no current flows through it and the voltage across its terminals is unaffected, and this is the basic operation of the diode clipping circuit.

Clippers are networks that employ diodes to “clip” away a portion of an input signal without distorting the remaining part of the applied waveform.

There are two general categories of clippers: **Series** and **Parallel**. The series configuration is defined as the one where the diode is in series with the load, whereas the parallel variety has the diode in a branch parallel to the load.

SHUNT CONFIGURATIONS**SHUNT POSITIVE CLIPPER**

As shown above, when the positive half cycle appears, the diode being forward biased, acts as short circuit and thus the output voltage remains at zero level. During the negative half cycle, the diode is reverse biased, acts as open circuit and hence we see that the output node comes into direct contact with the input node, thereby the output

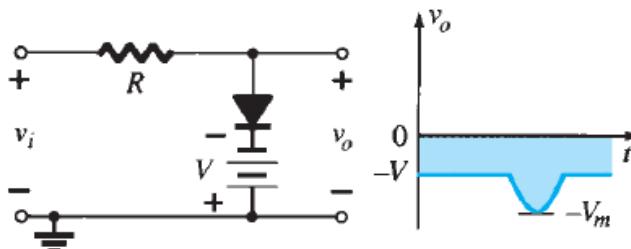
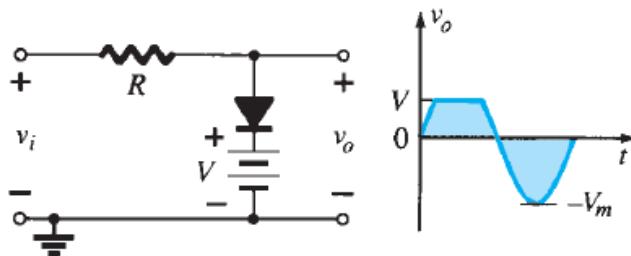


follows the input. Since the positive cycle of the input is getting clipped-off, the configuration in the above circuit is known as shunt positive clipper.

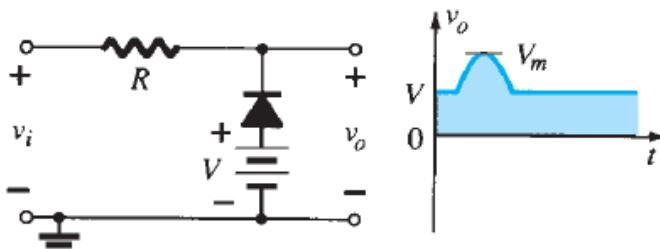
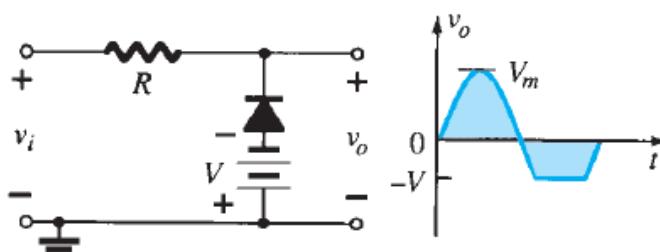
Likewise if the polarity of the diode is reversed; we can clip-off the negative half of the input cycle. In this case, during the positive half cycle, the diode remains reverse biased thereby connecting the output node with input node and the output voltage follows the input. But when the negative half cycle appears, the diode gets forward biased creating a short across the output nodes resulting into a zero voltage at the output. The level will be 0.7 if a silicon diode is considered instead of an non-ideal diode.

FEW SHUNT DIODE CLIPPER CONFIGURATIONS

POSITIVE

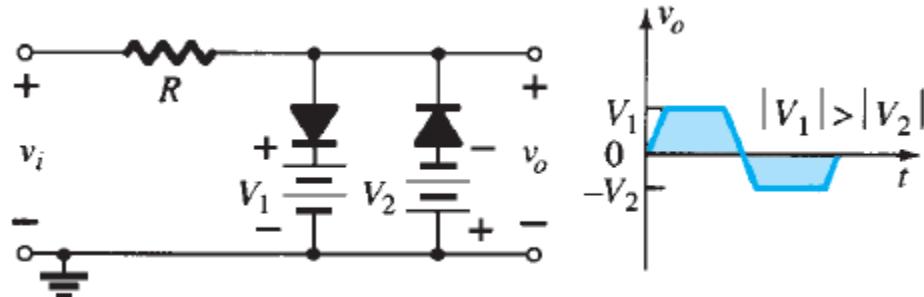


NEGATIVE



**TWO LEVEL CLIPPERS**

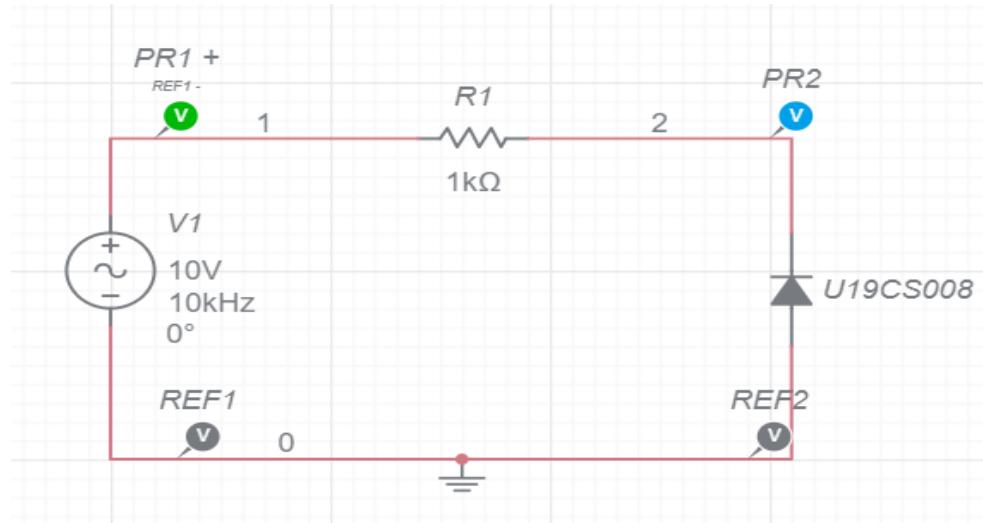
These circuits employ clipping in both the directions (Positive as well as Negative Half Cycles) as shown in figure below:



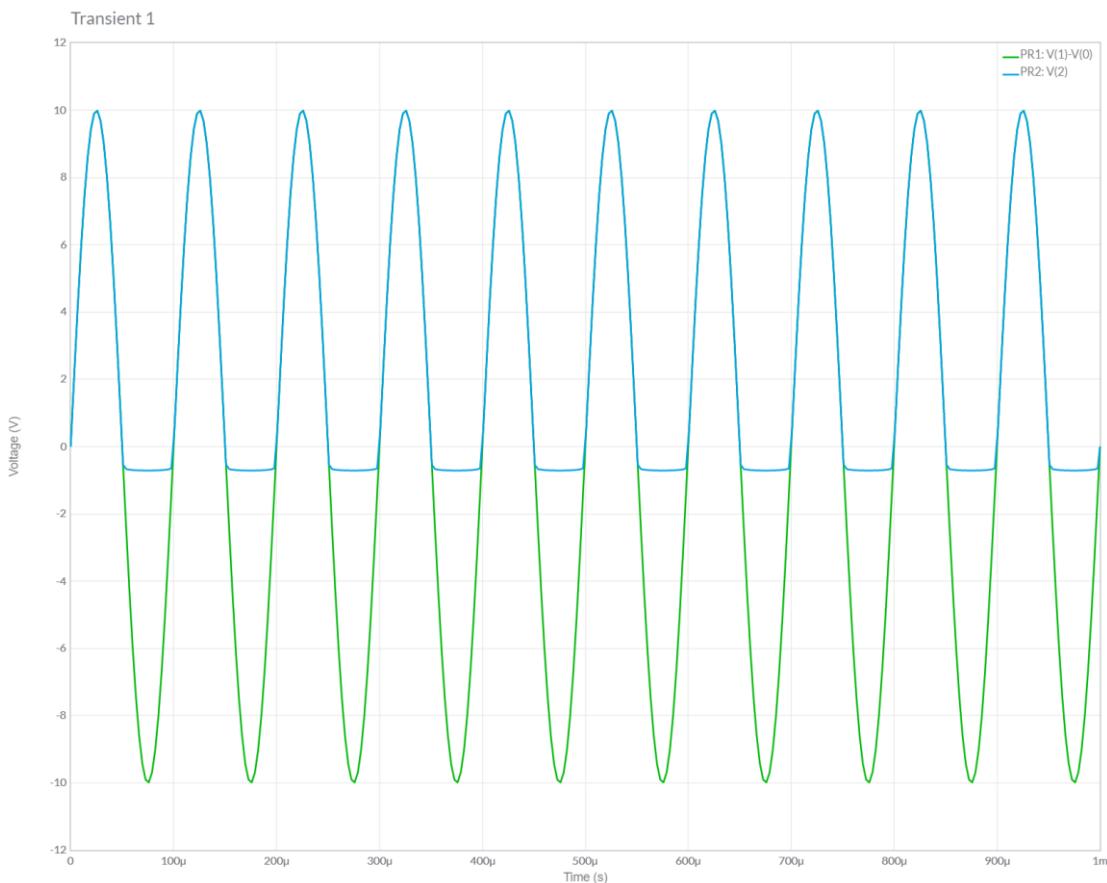


NEGATIVE SHUNT CLIPPER

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



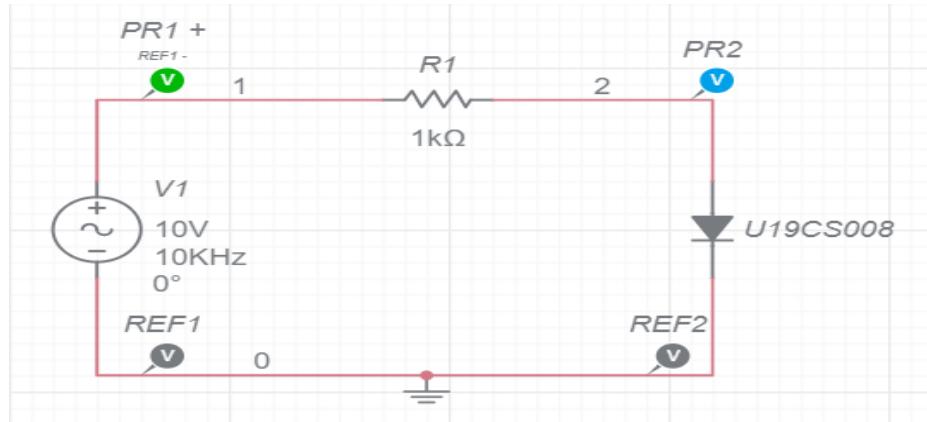
WAVEFORMS (FROM MULTISIM)



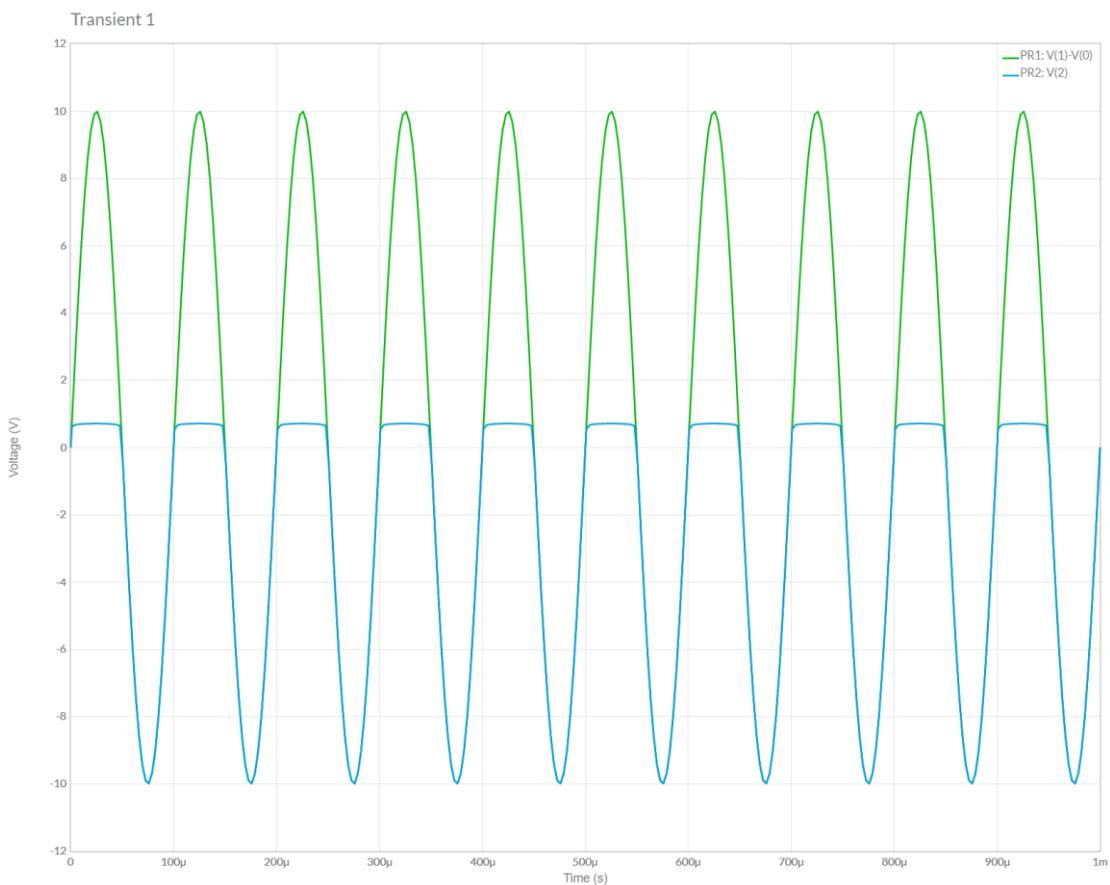


POSITIVE SHUNT CLIPPER

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



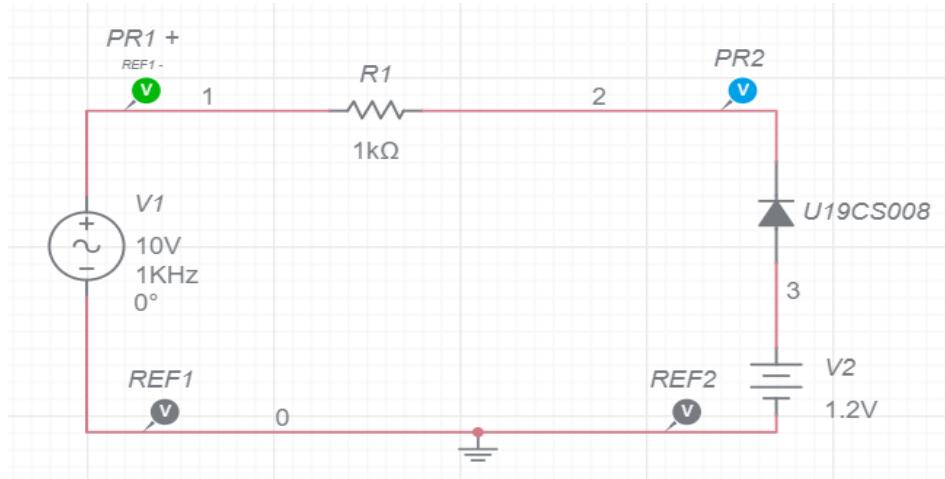
WAVEFORMS (FROM MULTISIM)



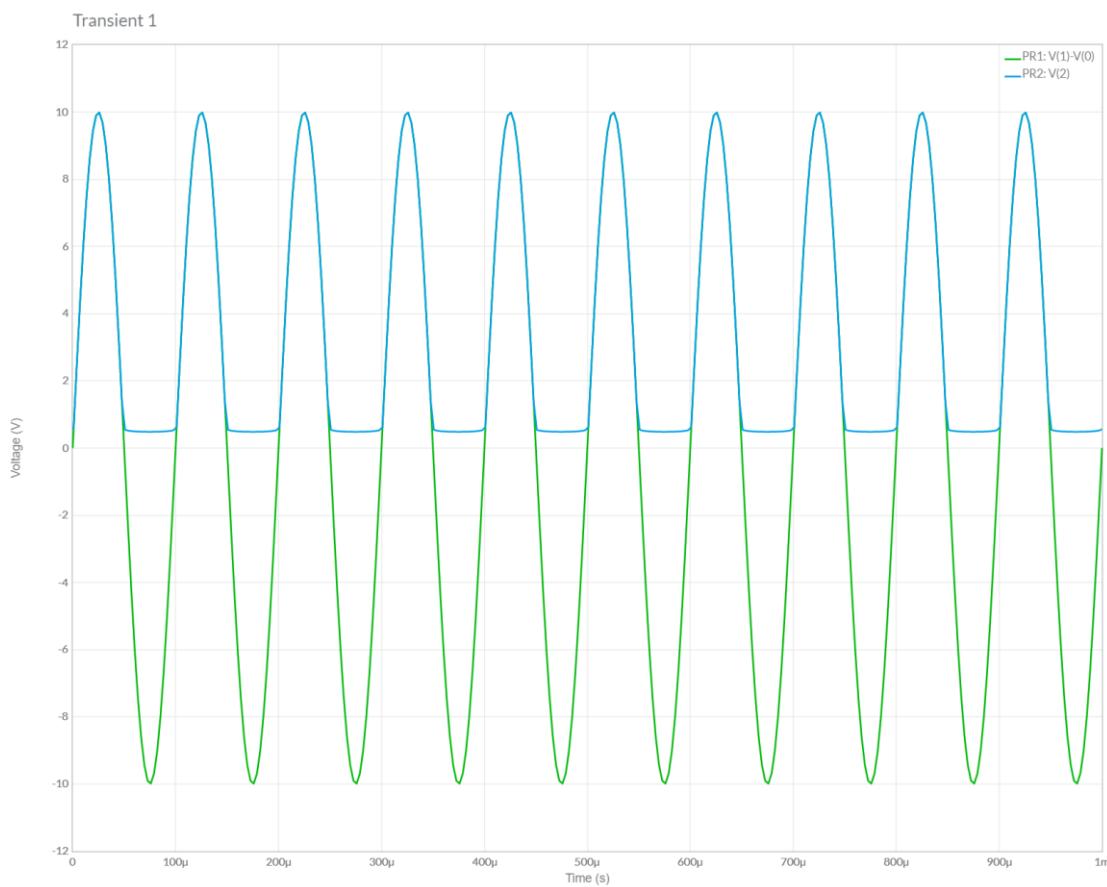


NEGATIVE SHUNT CLIPPER WITH BIAS-I

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



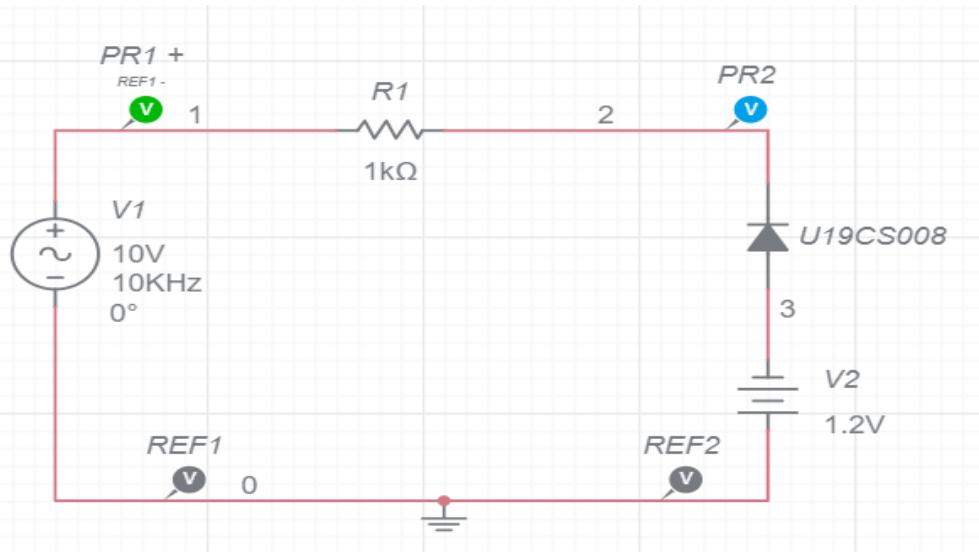
WAVEFORMS (FROM MULTISIM)



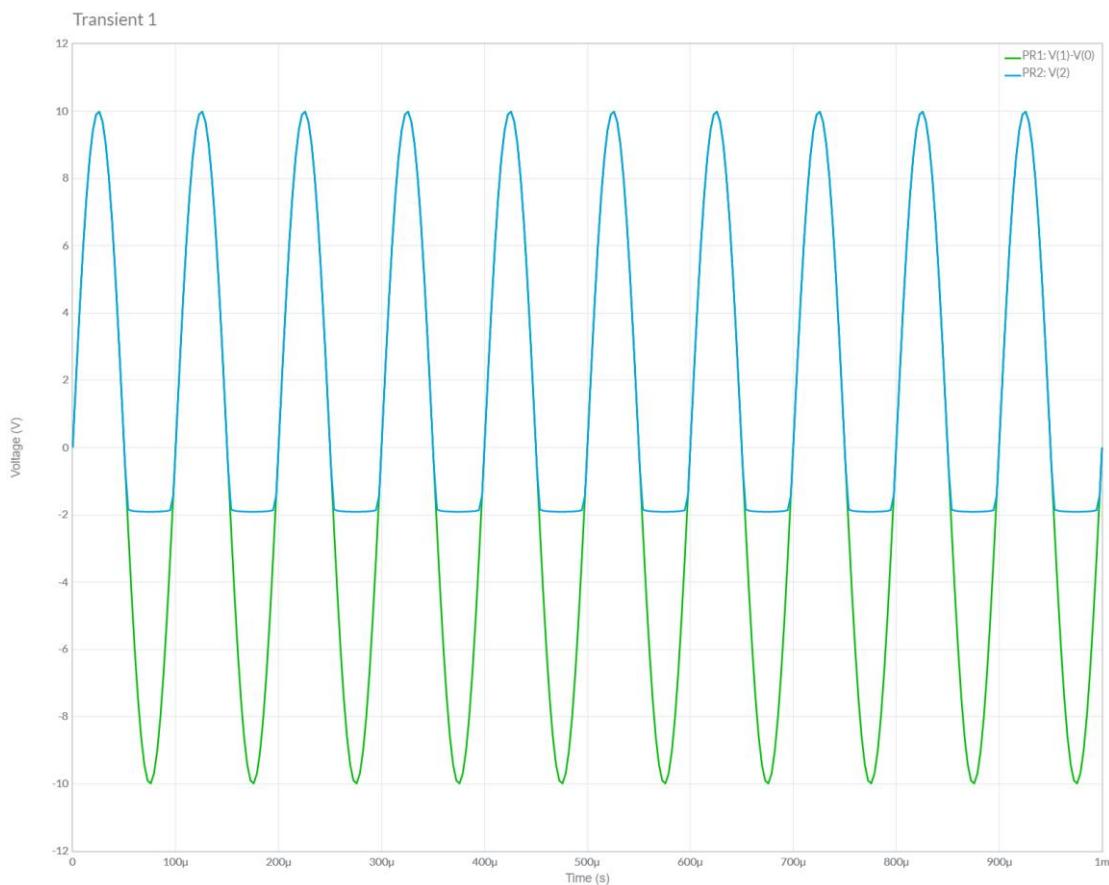


NEGATIVE SHUNT CLIPPER WITH BIAS-II

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



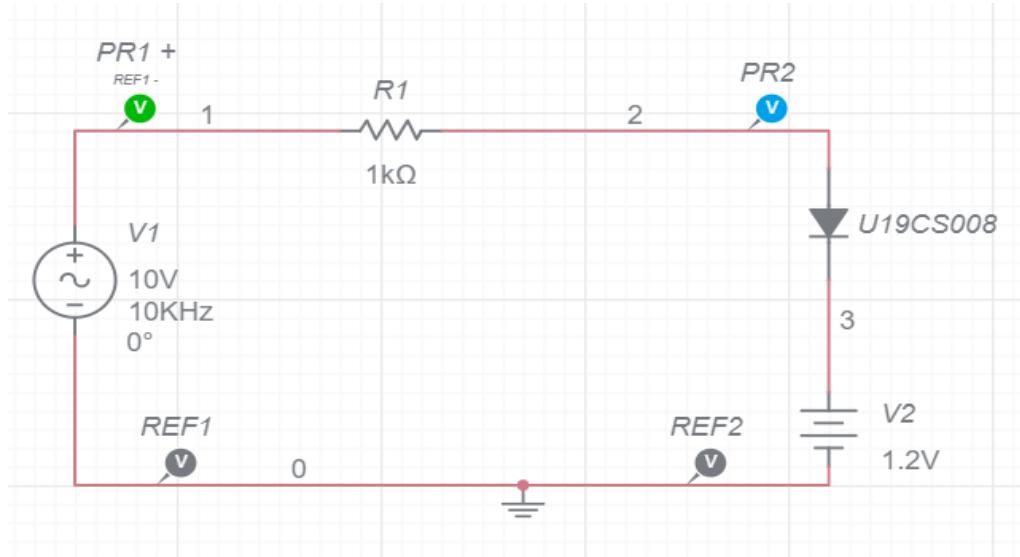
WAVEFORMS (FROM MULTISIM)



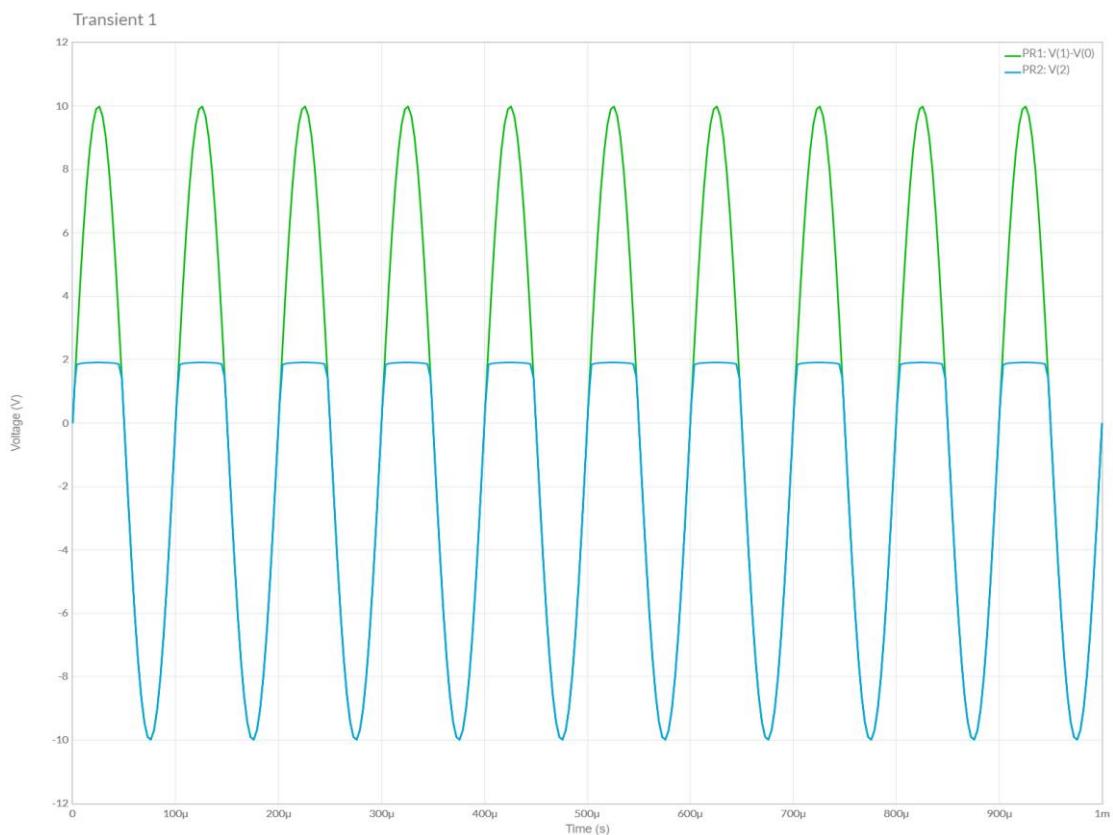


POSITIVE SHUNT CLIPPER WITH BIAS-I

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



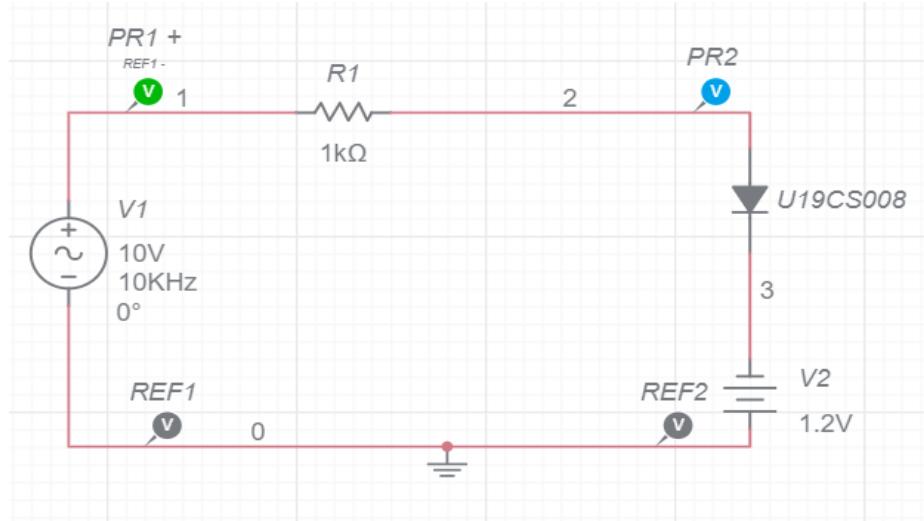
WAVEFORMS (FROM MULTISIM)



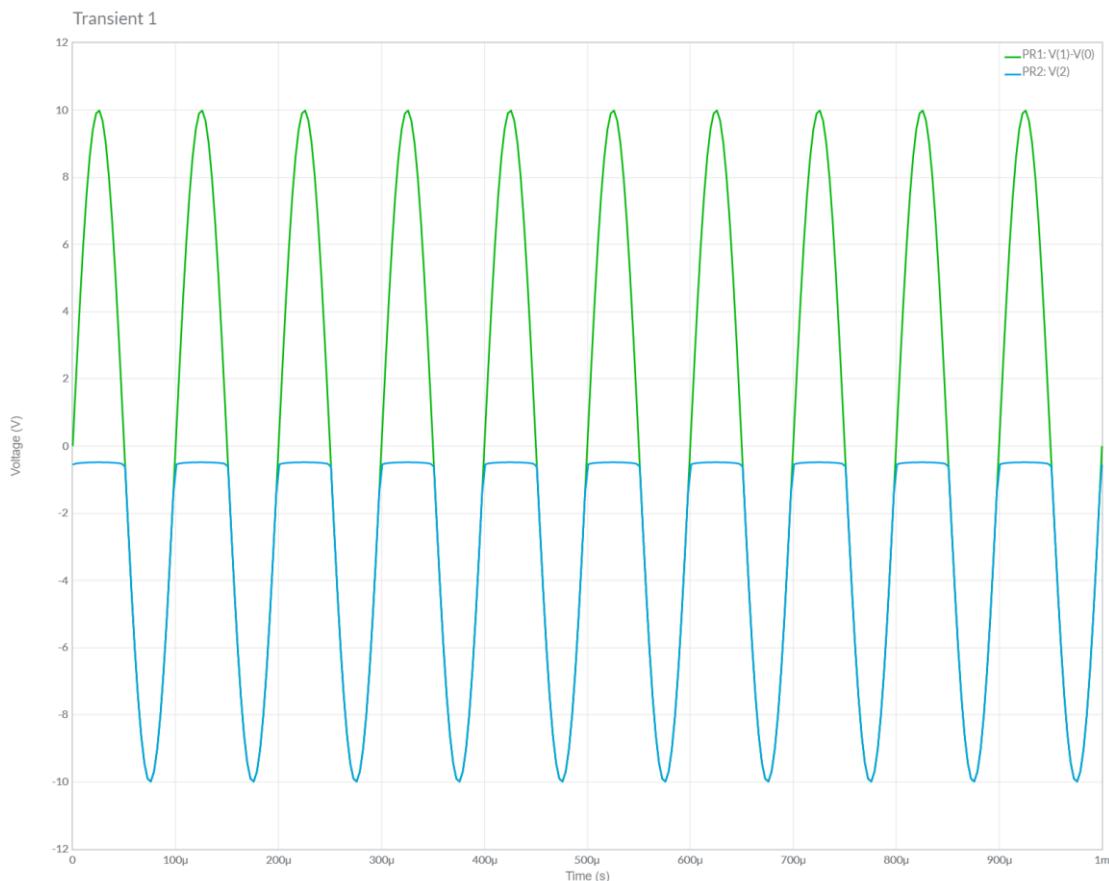


POSITIVE SHUNT CLIPPER WITH BIAS-II

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



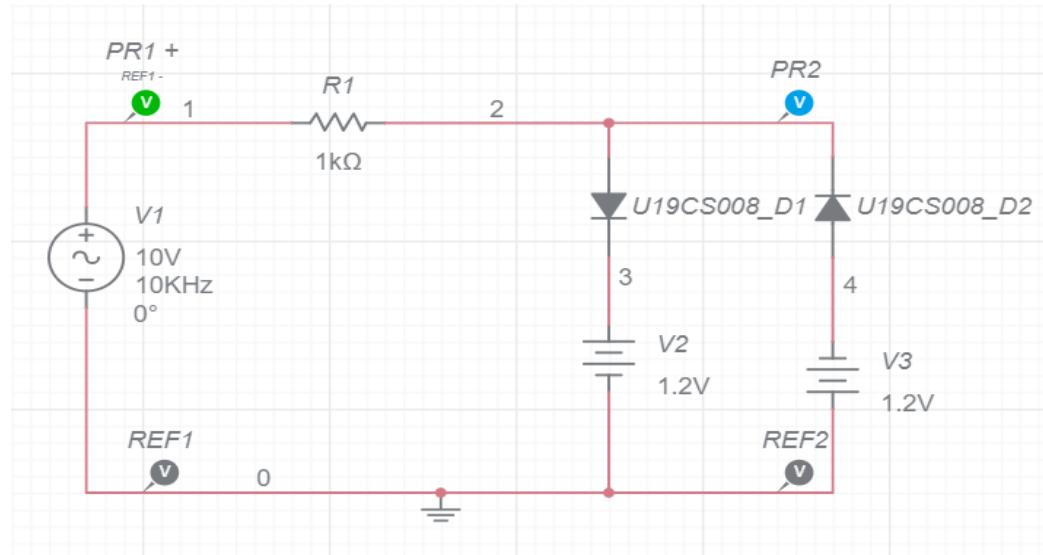
WAVEFORMS (FROM MULTISIM)



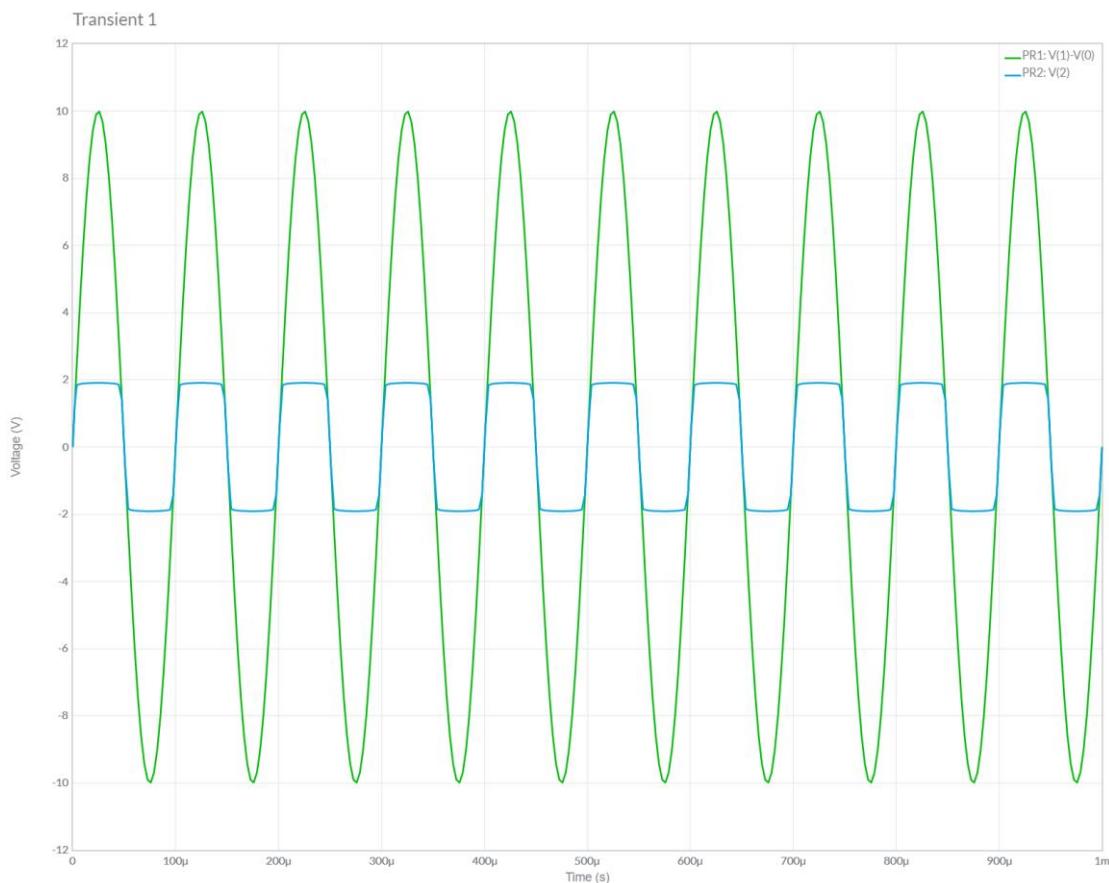


DUAL SHUNT CLIPPER

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



WAVEFORMS (FROM MULTISIM)



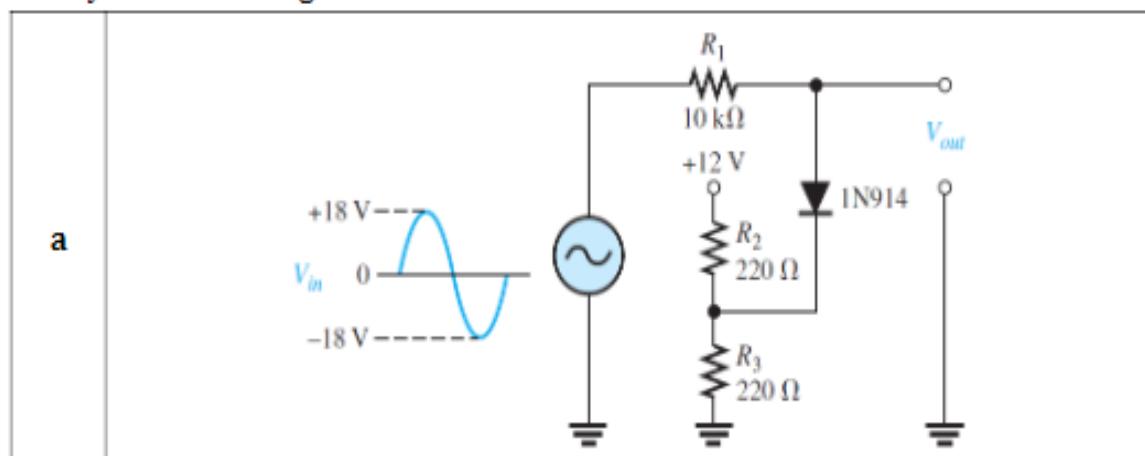


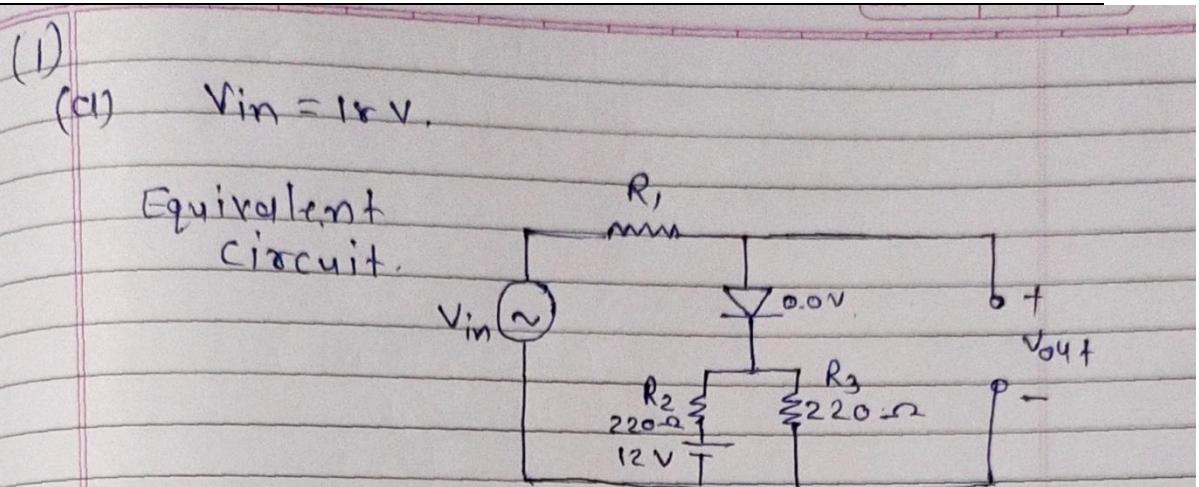
CONCLUSIONS

HERE, THE PRACTICAL AND THEORITICAL CHARACTERISTICS OF NEGATIVE AND POSITIVE SHUNT CLIPPER (WITH AND WITHOUT BIASED) ARE SAME. HENCE VERIFIED.....

**DLED ASSIGNMENT-7****NAME: KRINA PATEL****ADMISSION NUMBER: U19CS008**

1. Determine and plot the output voltage waveform for the given circuits. Also verify the same using Multisim.

**THEORITICAL:**



Equivalent voltage due to R_1 & R_3
& and 12V battery across
 220Ω is,

$$= \left(\frac{220}{220+220} \right) \times 12 \text{ V}$$

$$= 6 \text{ V} \quad \text{now},$$

which is Bias voltage.

considering ideal diode

V_{in}

6.7 V

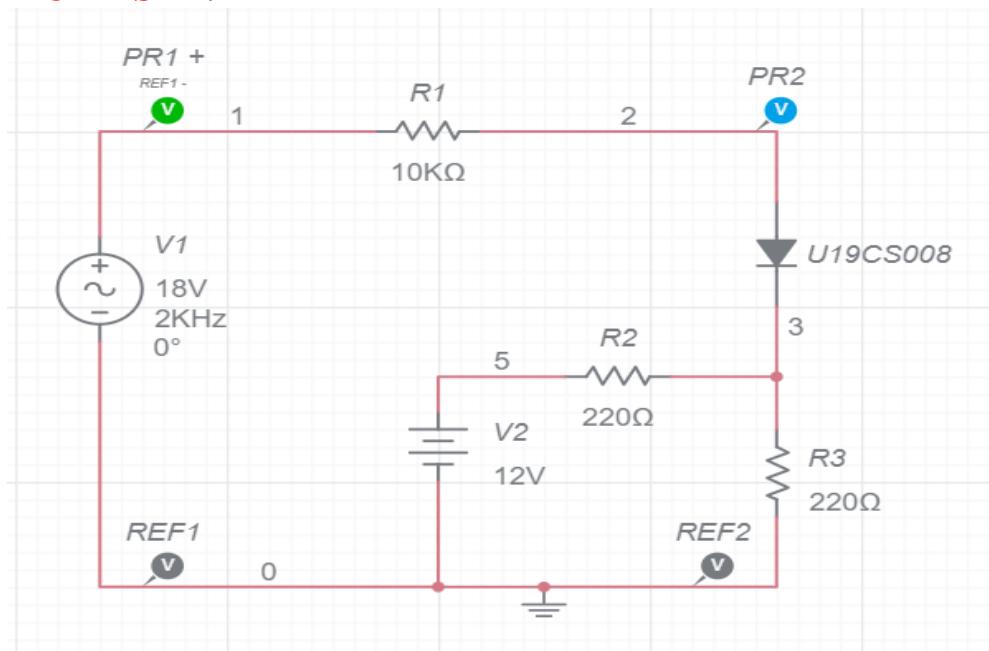
-18 V

$$\therefore V_{out} = 6.7 + 0.7 \\ = 6 + 0.7 \\ = 6.7 \text{ V}$$

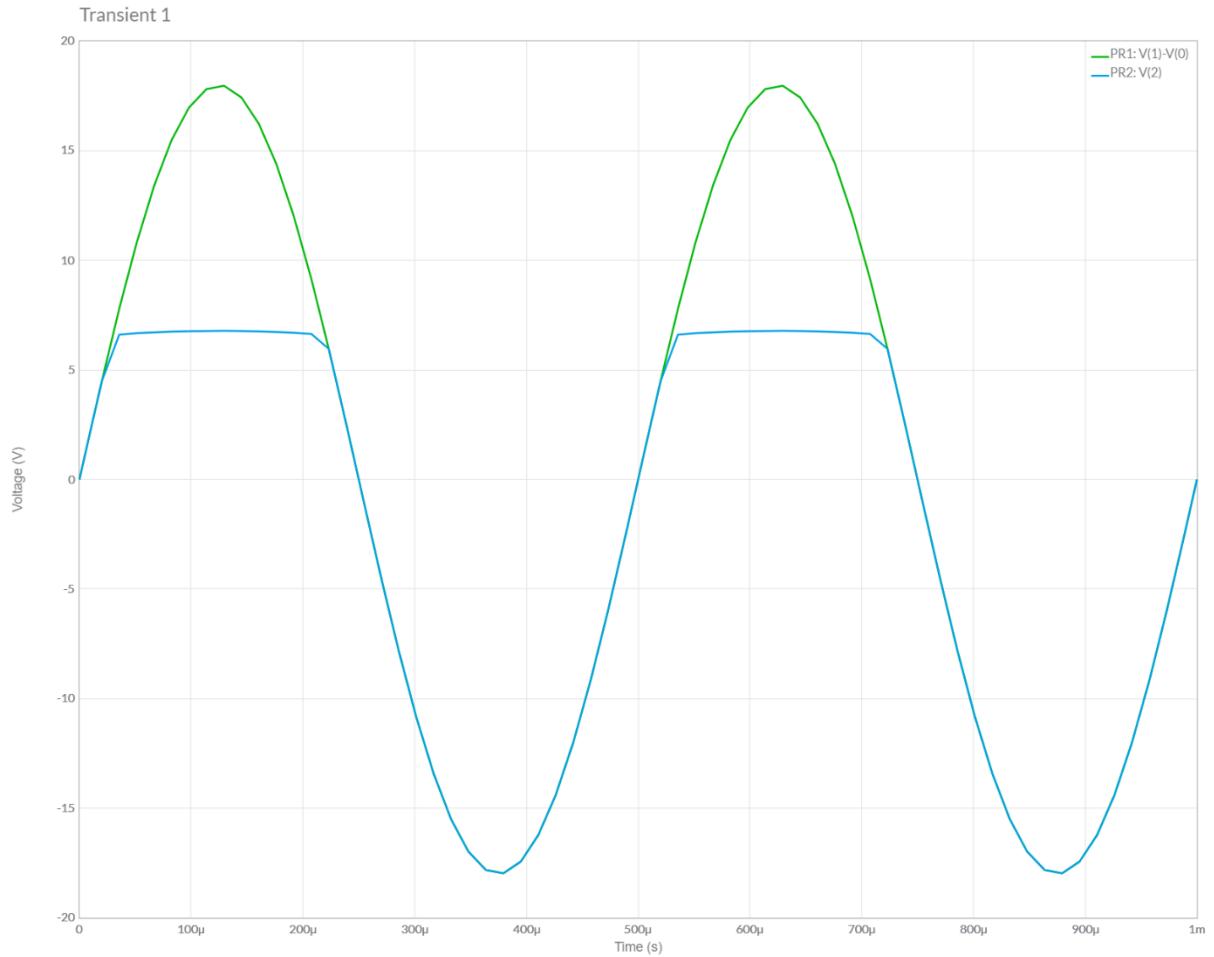
$$\therefore V_{out} = 6.7 \text{ V}$$



MULTISIM:

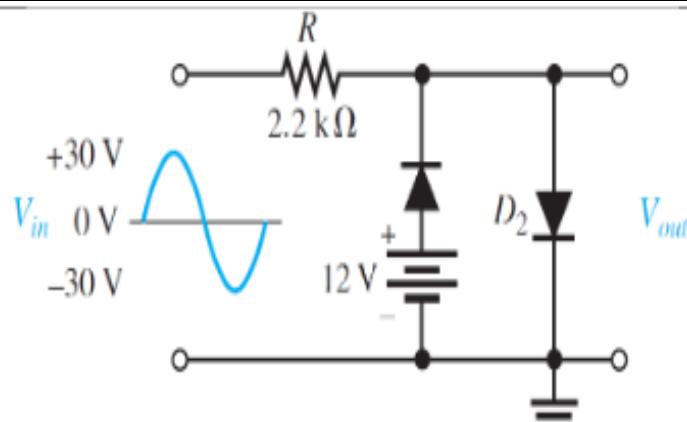


GRAPH:



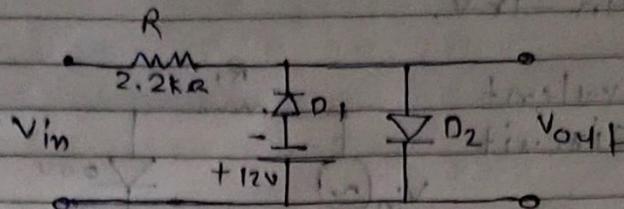


b





THEORITICAL:

(1) (b) $V_{in} = +30 \text{ V}$ 

considering both ideal diode

In +ve half cycle, D_1 will be reverse biased and D_2 forward biased.

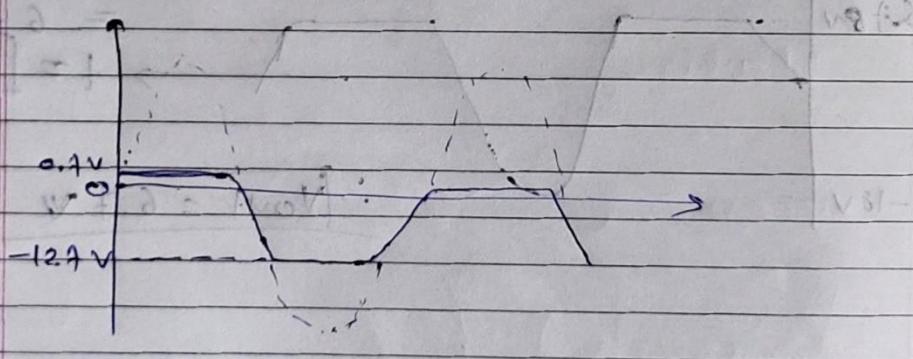
So, $V_{out} = V_D = 0.7 \text{ V}$

In -ve half cycle, D_2 will be reverse biased (open circuit) & D_1 forward biased so, V_{input} will get clipped at $V_{ref} + V_D$.

$\therefore V_{out} = -(+12 + 0.7) \text{ V}$

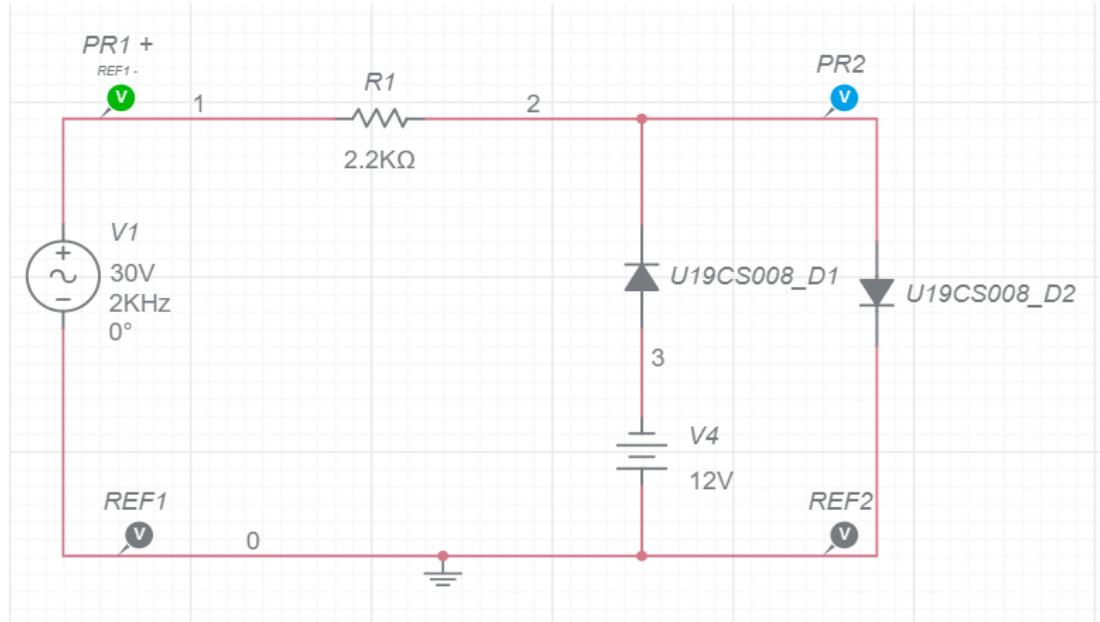
$V_{out} = -12.7 \text{ V}$

Output waveform.

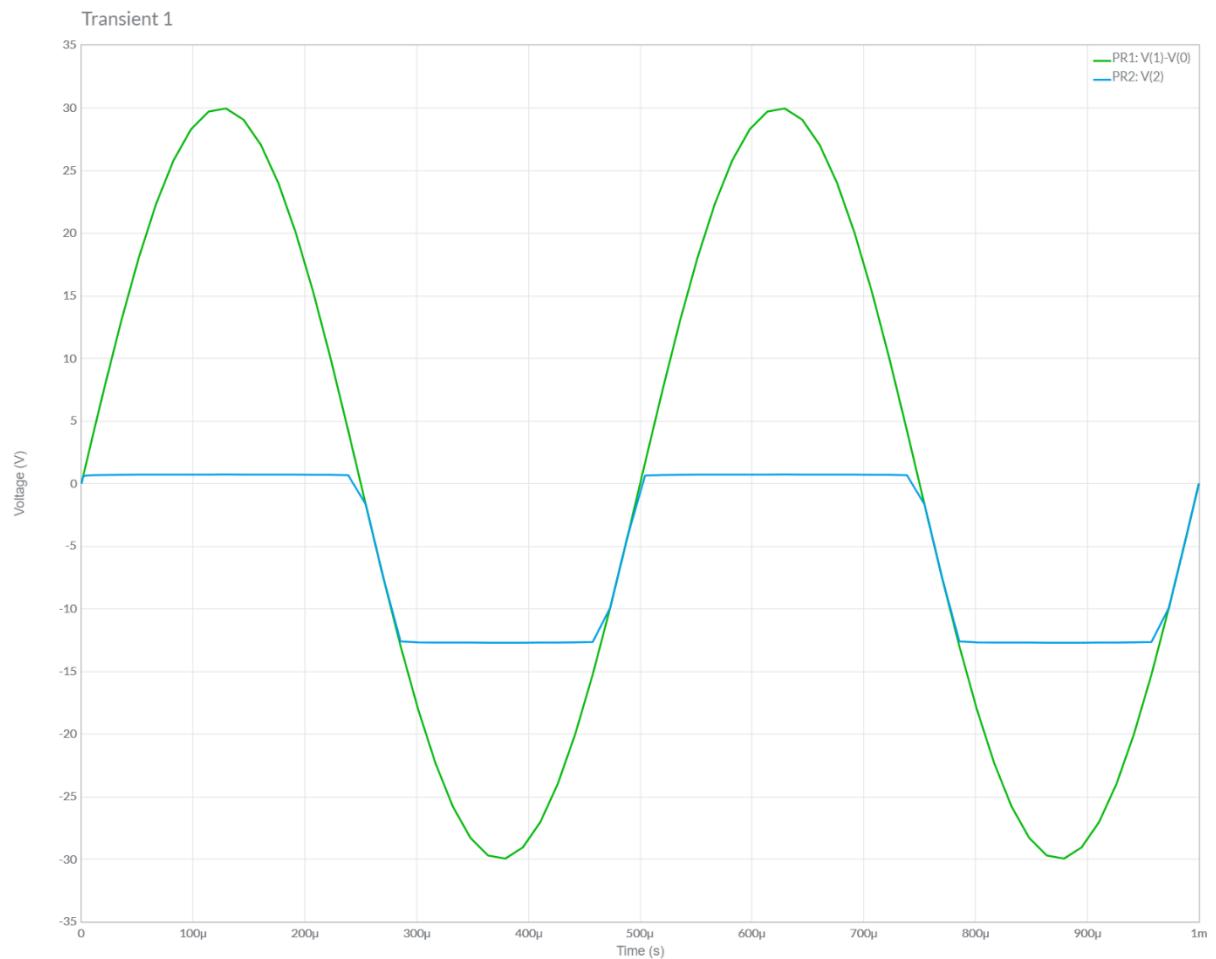




MULTISIM:

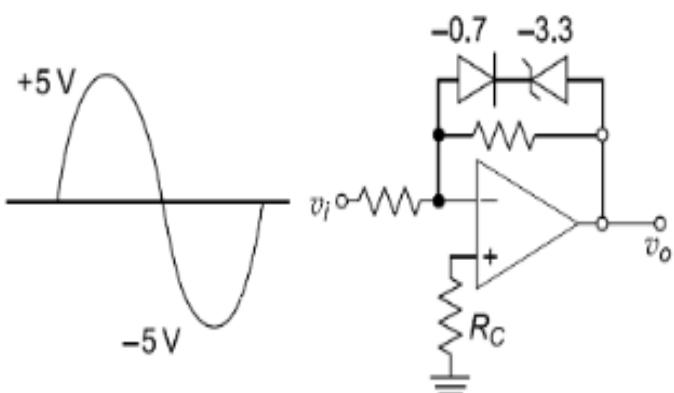


GRAPH:



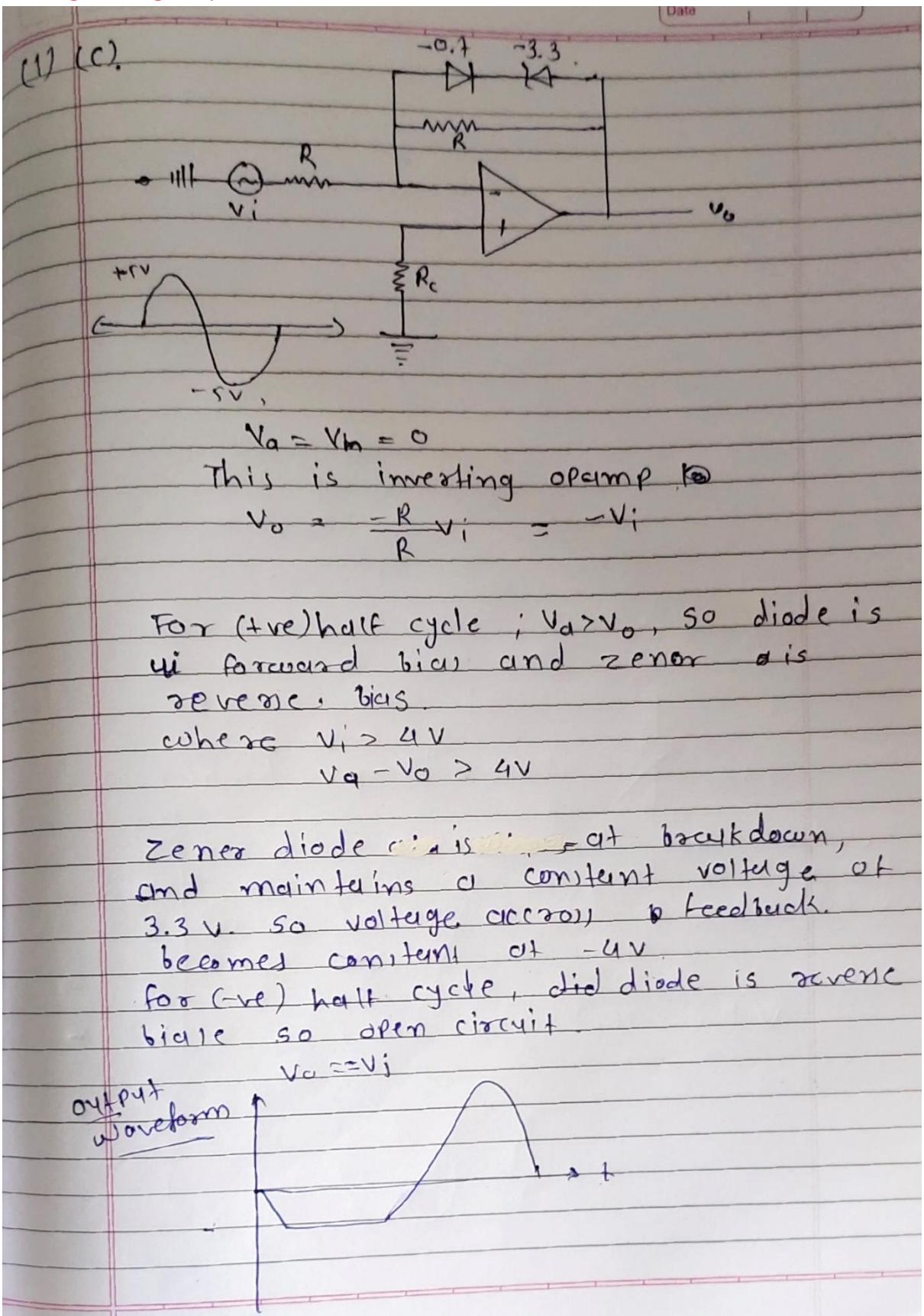


c



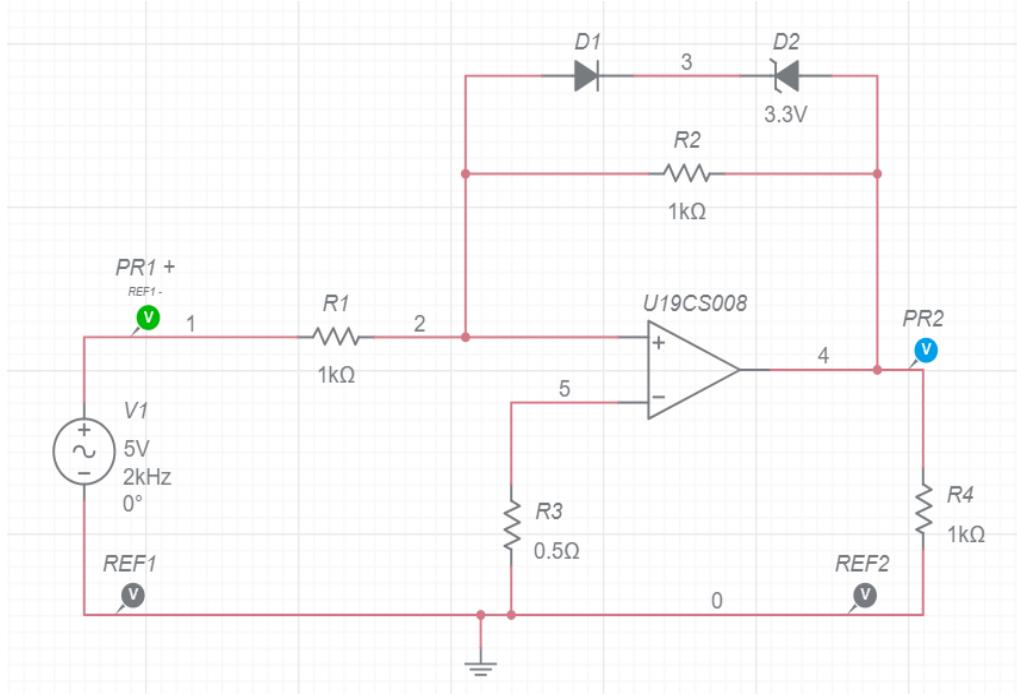


THEORITICAL:

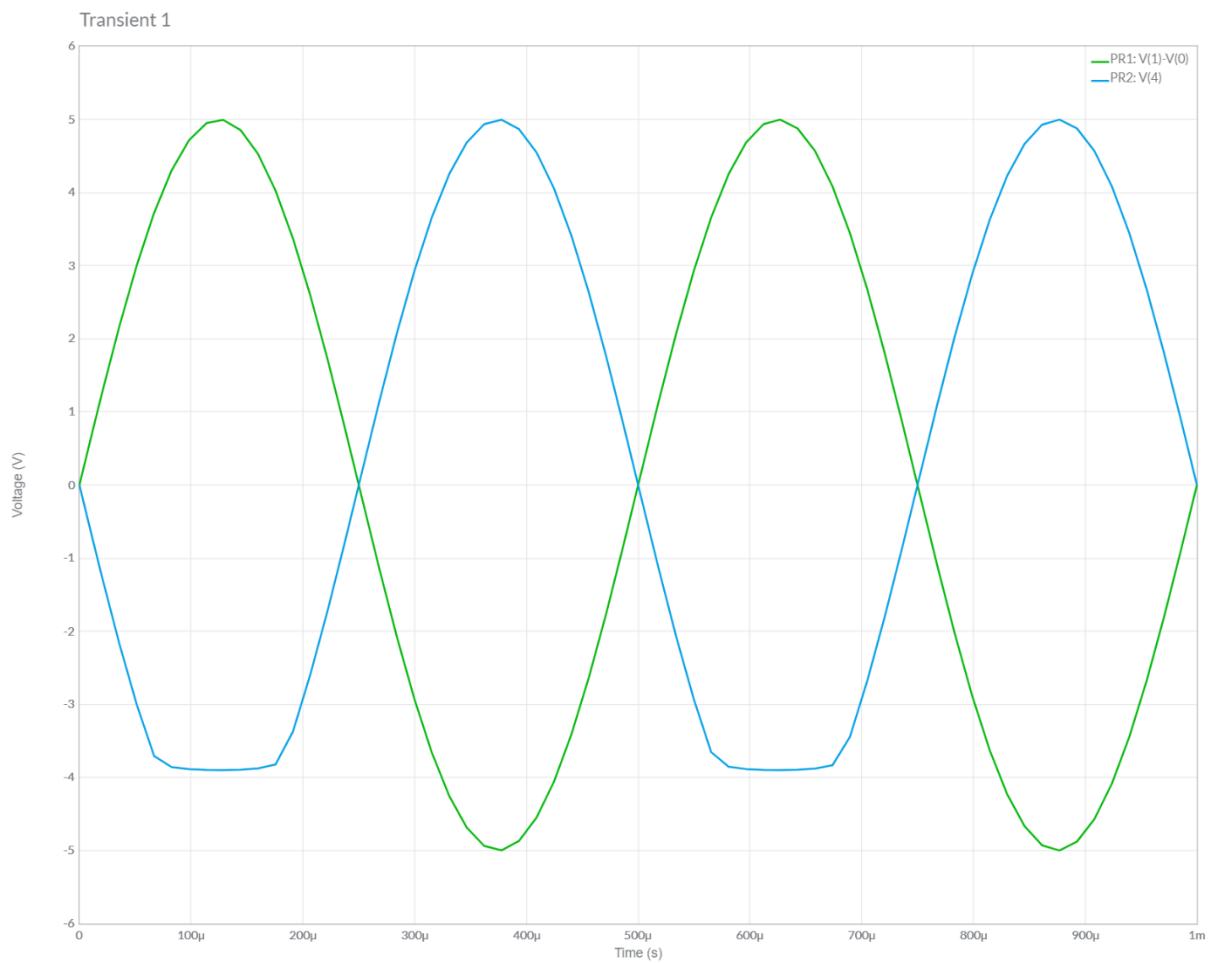




MULTISIM:



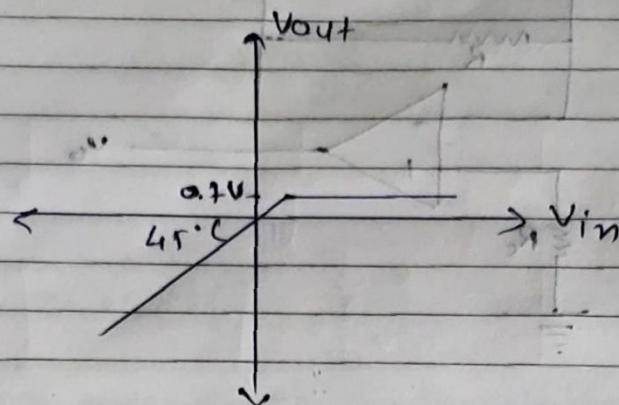
GRAPH:



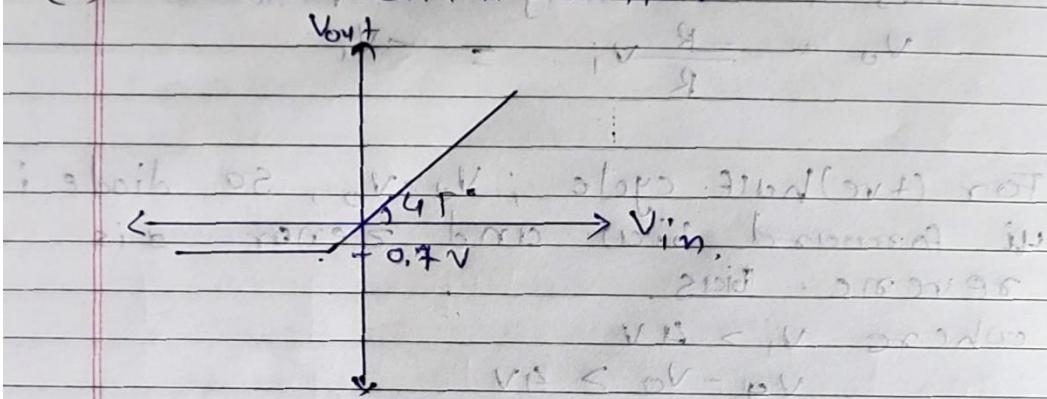


2. DRAW THE TRANSFER CHARACTERISTICS FOR ALL THE CLIPPER CONFIGURATIONS WHICH ARE PART OF YOUR TODAY'S PRACTICAL (PRCTICAL-7)

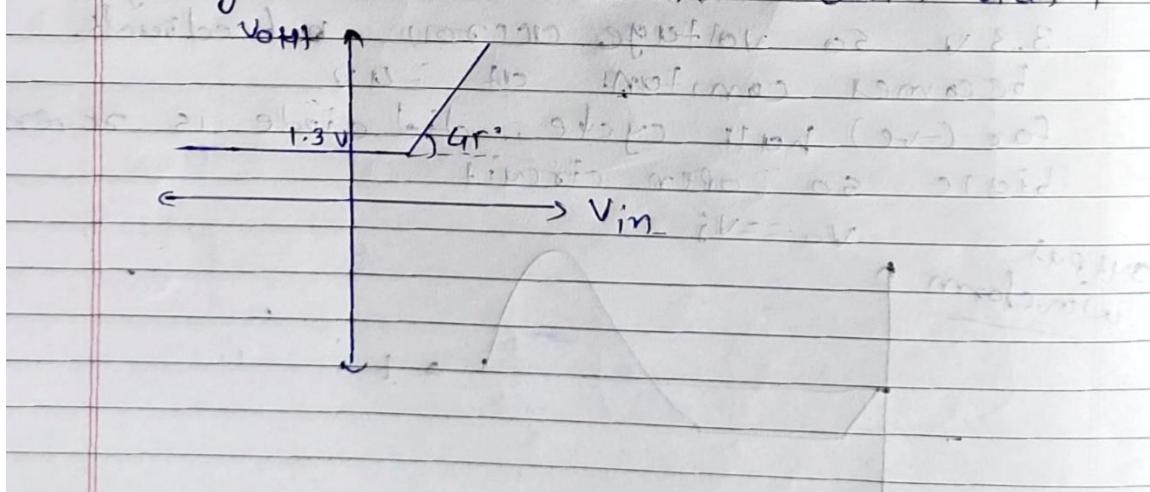
(1) Shunt negative clipper



(2) Positive shunt clipper

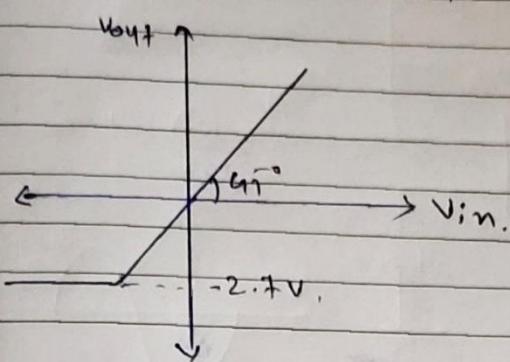


(3) Negative shunt clipper with bias - 1.3V

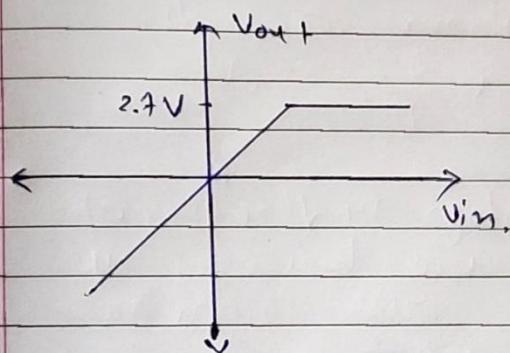




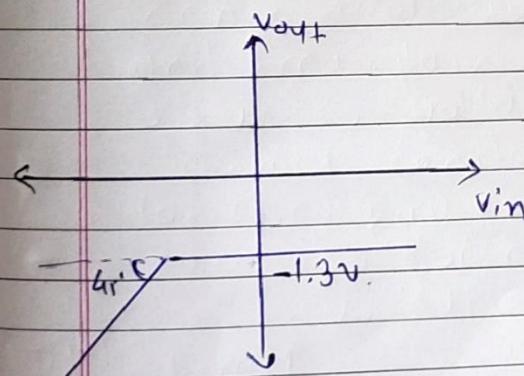
(4) Negative shunt clipper with $b_{ic1} = 2$



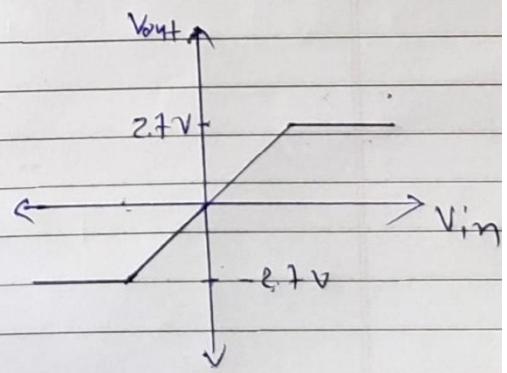
(5) Positive shunt clipper with $b_{ic1} = 1$



(6) Positive shunt clipper with $b_{ic1} = 2$

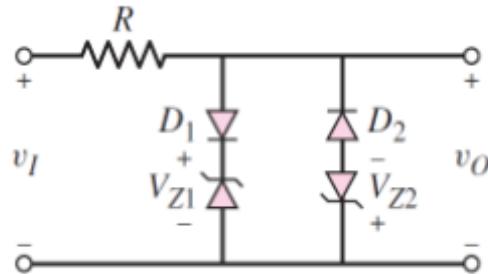


(7) Dual shunt clipper





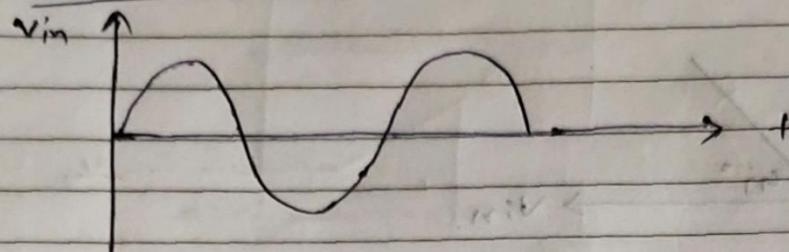
3. Assuming Symmetrical Sine wave input with peak value greater than the Zener reference voltage, predict the output and plot the Transfer Characteristics for the following Clipper Circuits:



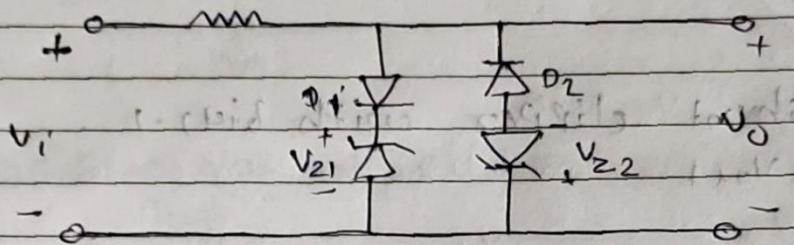


Q-3

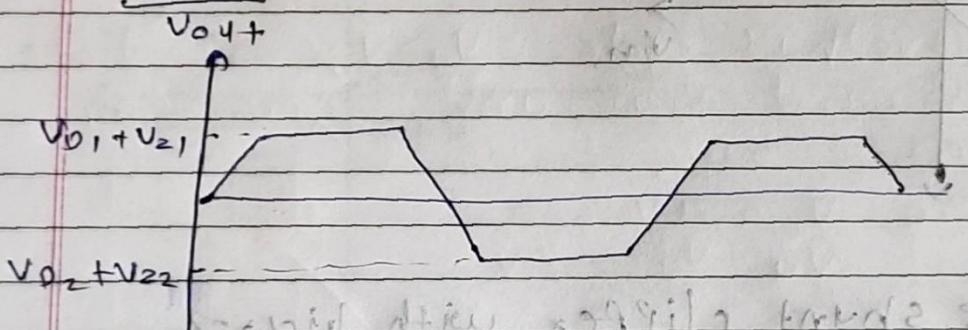
Input signal.



Circuit



Output



$v_{out}(+)$

$v_{D1} + v_{Z1}$

$-(v_{D2} + v_{Z2})$

$v_{in}(+)$



Expt. No:

8

Date:

22-10-2020

Diode Clamper Circuits

AIM: To study, design and plot the various Clamper Circuits.

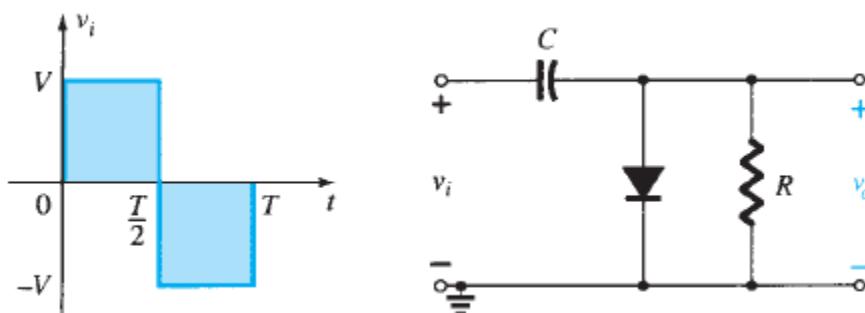
SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator/Circuit Simulator

THEORY:

A clamper is a network constructed of a diode, a resistor, and a capacitor that shifts a waveform to a different dc level without changing the appearance of the applied signal. Additional shifts can also be obtained by introducing a dc supply to the basic structure. The resistor and capacitor of the network must be chosen such that the time constant determined by $t=RC$ is sufficiently large to ensure that the voltage across the capacitor does not discharge significantly during the interval the diode is not conducting. Throughout the analysis we assume that for all practical purposes the capacitor fully charges or discharges in five time constants.

The simplest of clamper networks is shown in figure below. It is important to note that the capacitor is connected directly between input and output signals and the resistor and the diode are connected in parallel with the output signal. The diode is also in parallel with the output signal but may or may not have a series dc supply as an added element.

**Analysis**

Step 1: Start the analysis by examining the response of the portion of the input signal that will forward bias the diode.

Step 2: During the period that the diode is in the “ON” state, assume that the capacitor will charge up instantaneously to a voltage level determined by the surrounding network.



For the network shown above the diode will be forward biased for the positive portion of the applied signal. For the interval 0 to $T/2$ the network will appear as shown in **Fig. a** below. The short-circuit equivalent for the diode will result in $V_o=0$ V for this time interval. During this same interval of time, the time constant determined by $t=RC$ is very small because the resistor R has been effectively “shorted out” by the conducting diode and the only resistance present is the inherent (contact, wire) resistance of the network. The result is that the capacitor will quickly charge to the peak value of V volts as shown in **Fig. a** with the polarity indicated in the figure below.

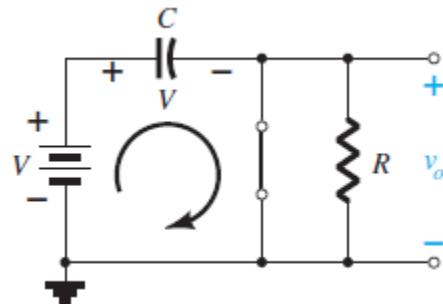


Fig. a

Step 3: Assume that during the period when the diode is in the “off” state the capacitor holds on to its established voltage level.

Step 4: Throughout the analysis, maintain a continual awareness of the location and defined polarity for V_o to ensure that the proper levels are obtained.

When the input switches to the $-V$ state, the network will appear as shown in **Fig.b**, with the open-circuit equivalent for the diode determined by the applied signal and stored voltage across the capacitor—both “pressuring” current through the diode from cathode to anode. Now that R is back in the network the time constant determined by the RC product is sufficiently large to establish a discharge period $5t$, much greater than the period T , and it can be assumed on an approximate basis that the capacitor holds onto all its charge and, therefore, voltage (since $V = Q/C$) during this period.

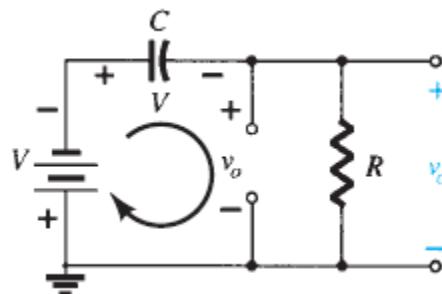


Fig. b

Since V_o is in parallel with the diode and resistor, it can also be drawn in the alternative position shown in **Fig.b**. Applying Kirchhoff's voltage law around the input loop results in

$$\begin{aligned} -V - V - v_o &= 0 \\ v_o &= -2V \end{aligned}$$

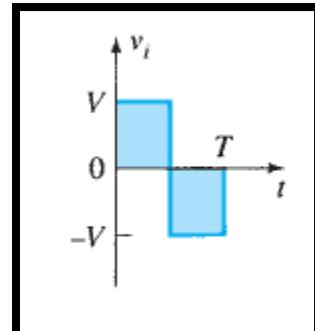
The negative sign results from the fact that the polarity of $2V$ is opposite to the polarity defined for V_o .

Step 5: Check that the total swing of the output matches that of the input.

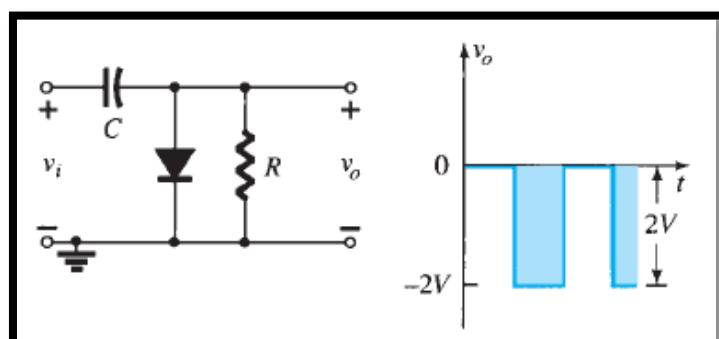
This is a property that applies for all clamping networks, giving an excellent check on the results obtained.



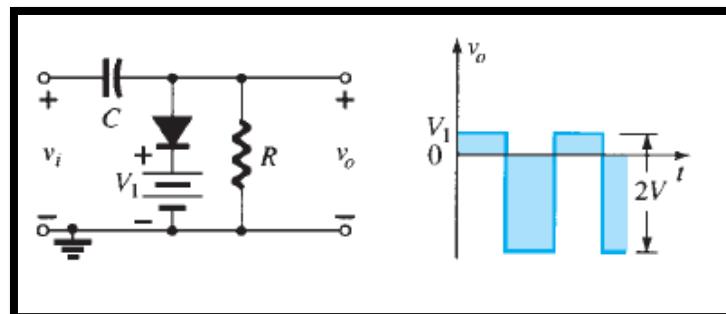
Few Clamper Configurations



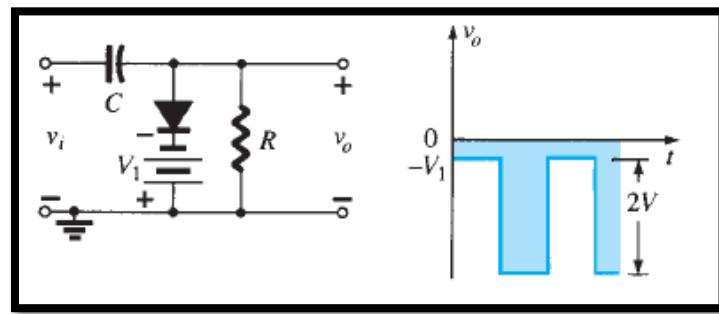
Common Input for all below circuits



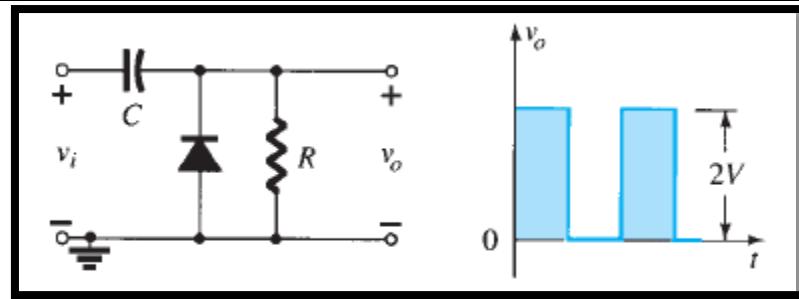
Negative Clammer with Zero Bias



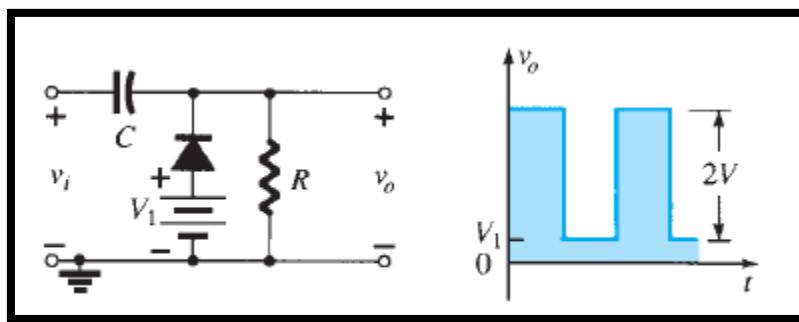
Negative Clammer with Positive Bias



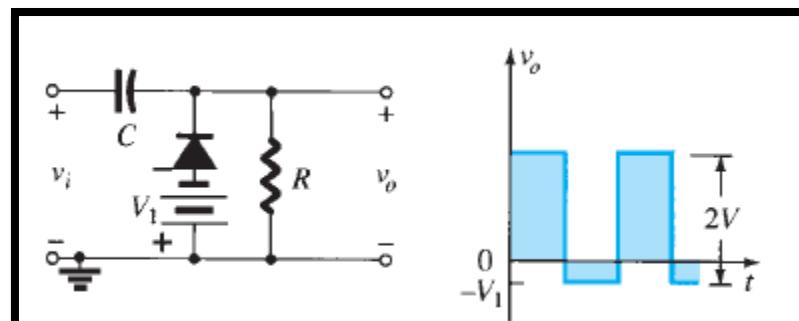
Negative Clammer with Negative Bias



Positive Clamper with Zero Bias



Positive Clamper with Positive Bias

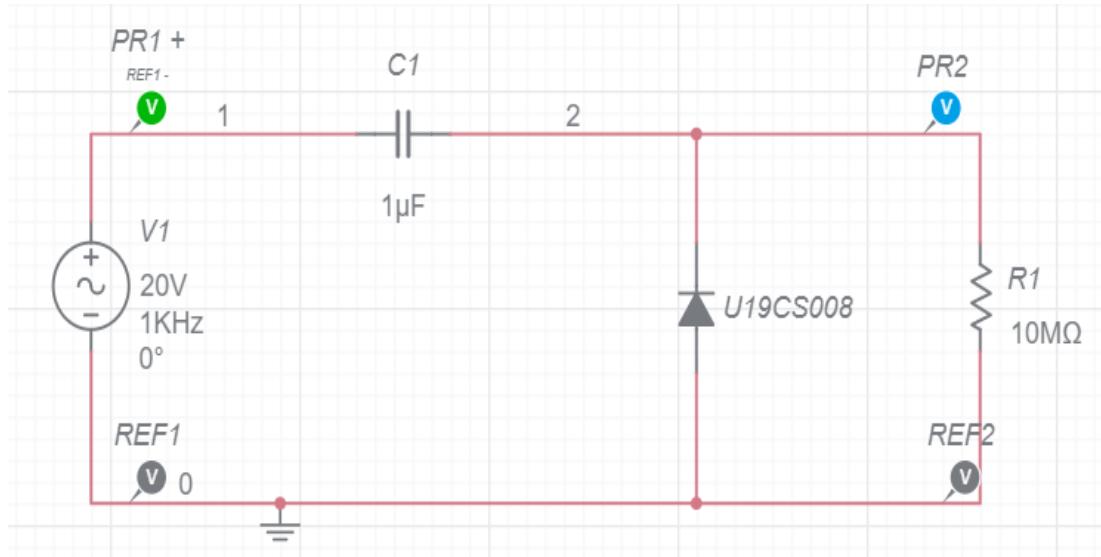


Positive Clamper with Negative Bias

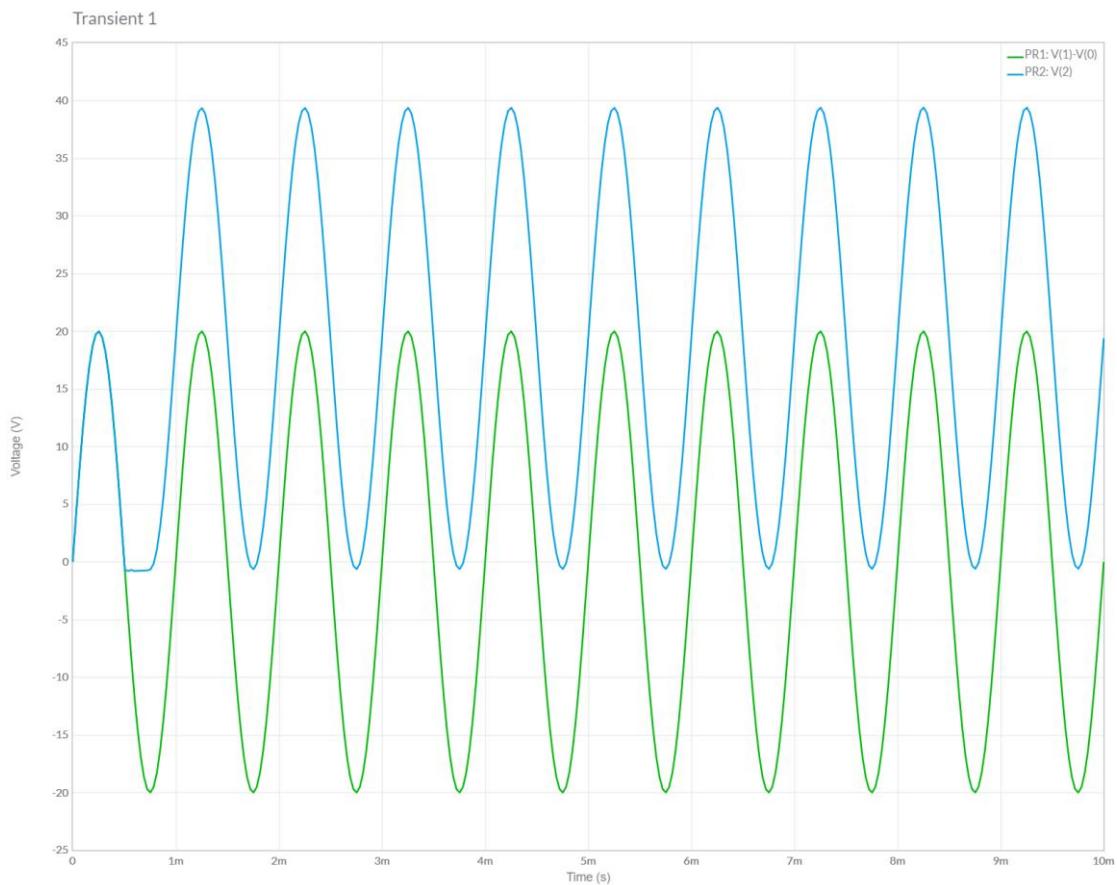


POSITIVE CLAMPER

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



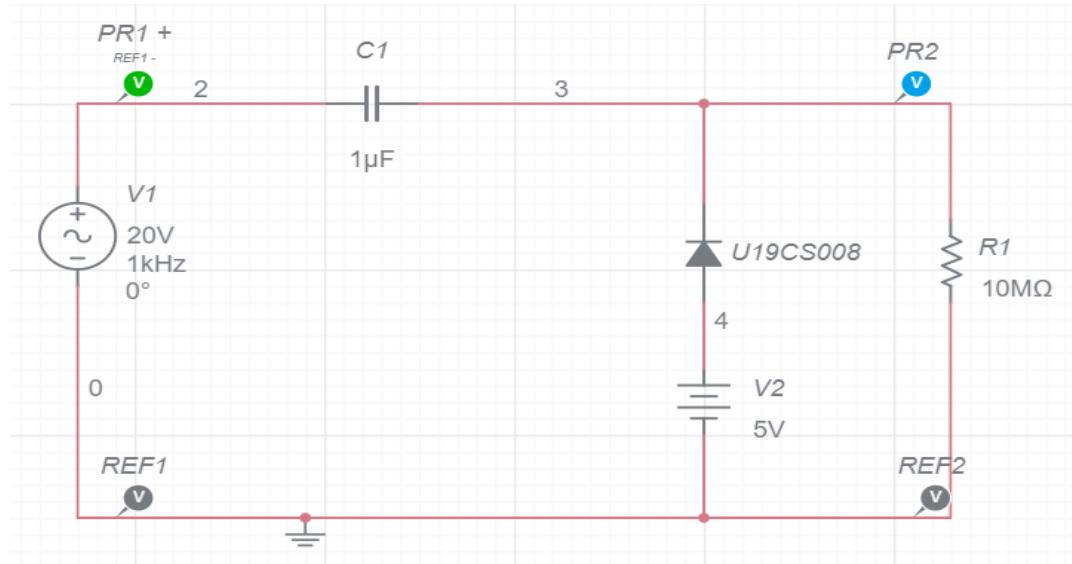
WAVEFORMS (FROM MULTISIM)



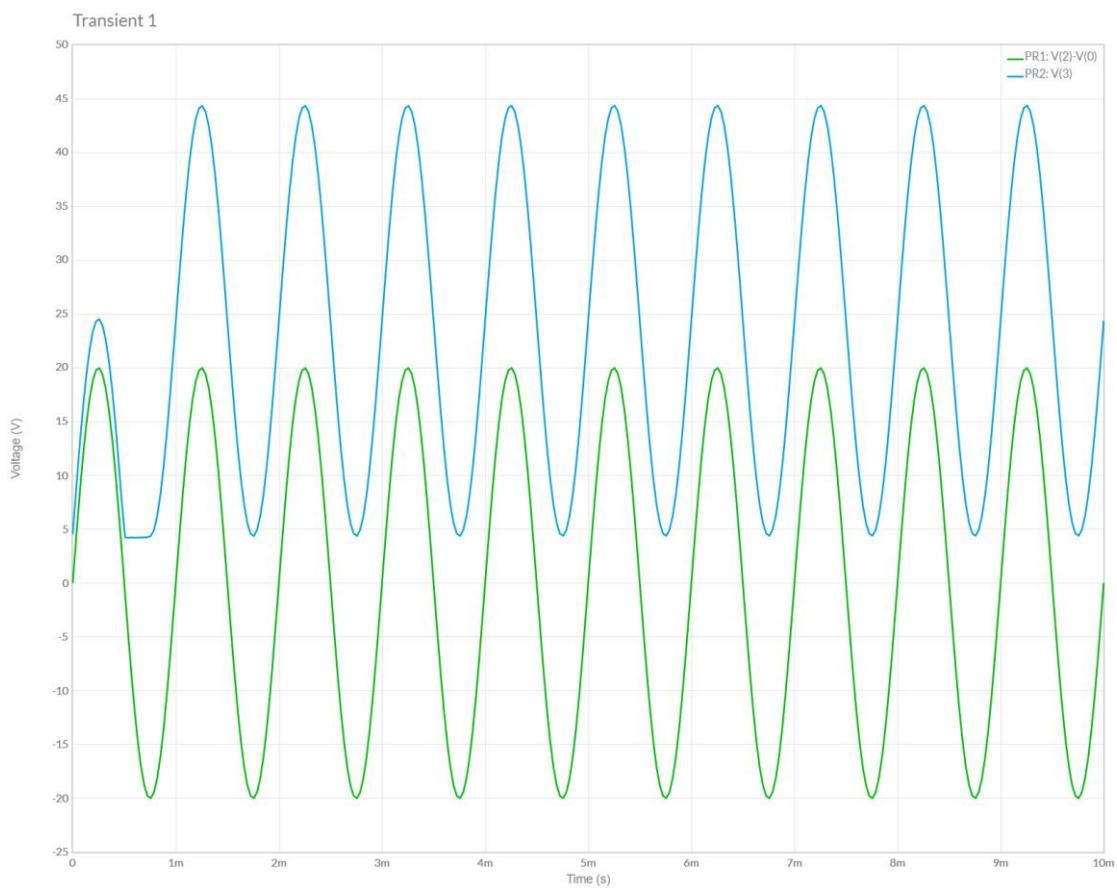


POSITIVE CLAMPER WITH POSITIVE BIAS

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



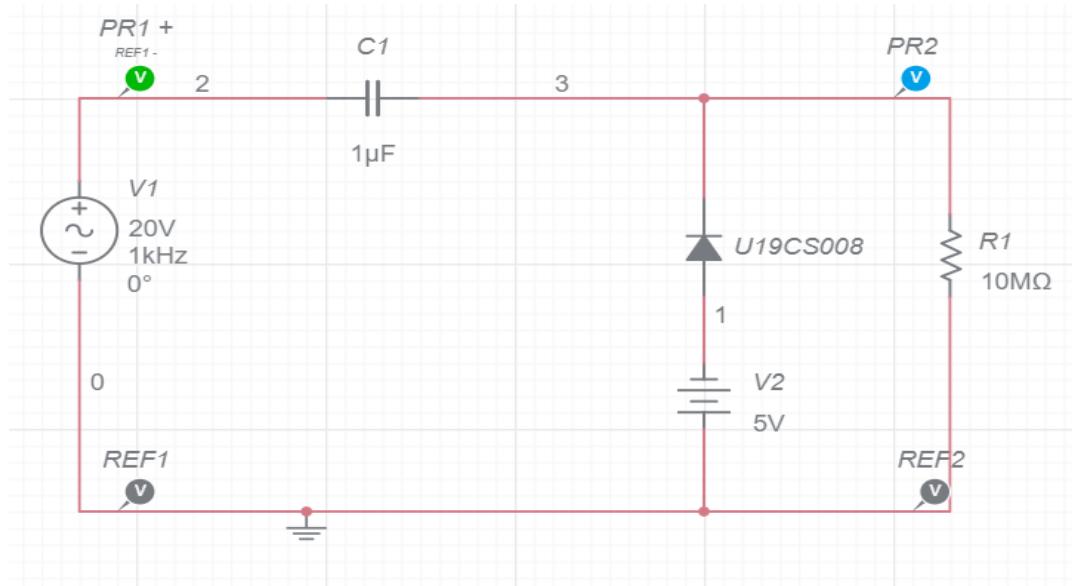
WAVEFORMS (FROM MULTISIM)



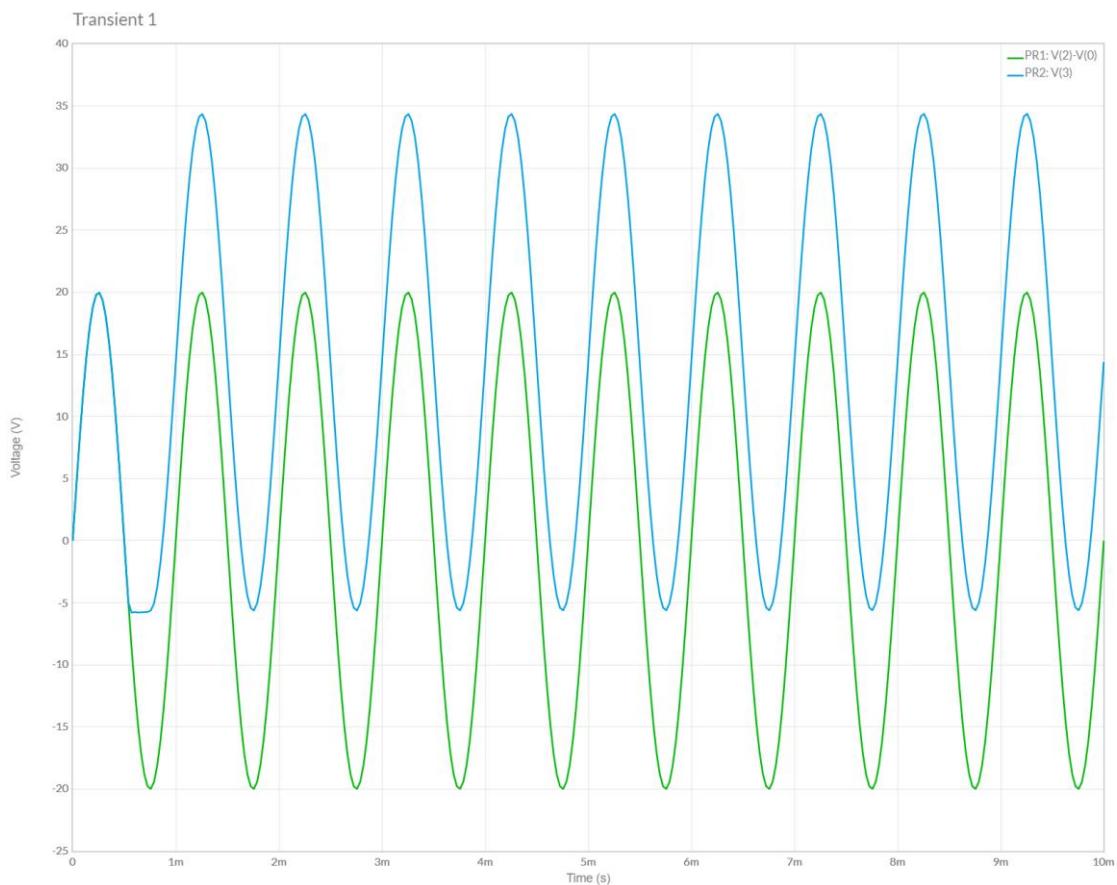


POSITIVE CLAMPER WITH NEGATIVE BIAS

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



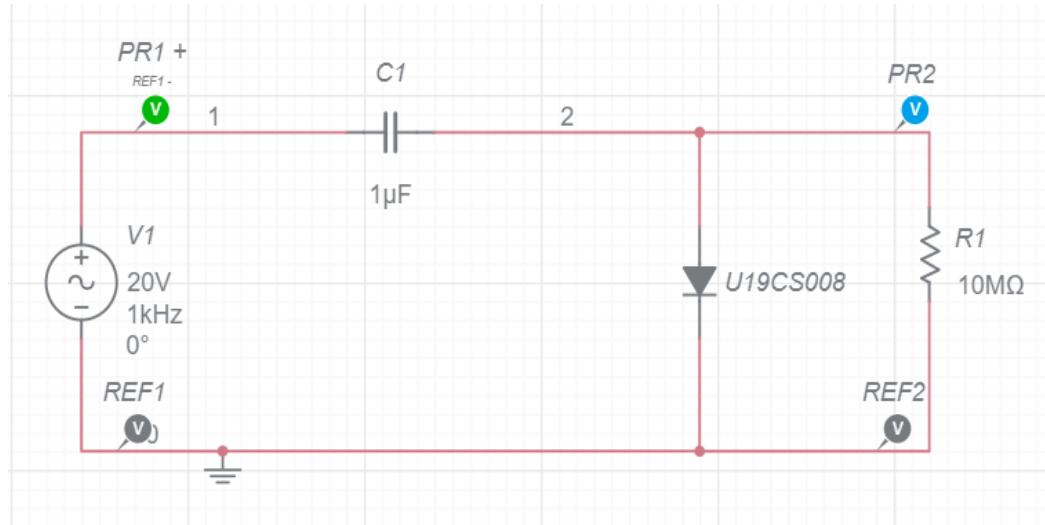
WAVEFORMS (FROM MULTISIM)



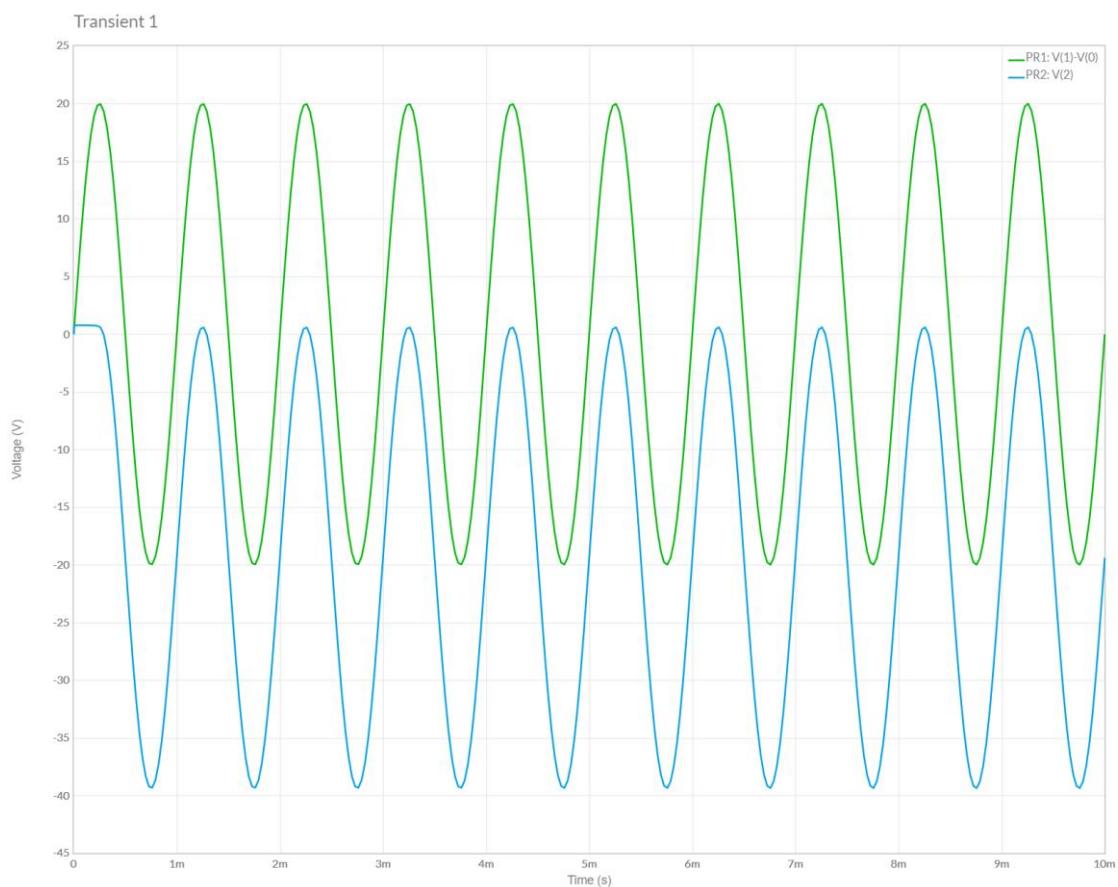


NEGATIVE CLAMPER

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



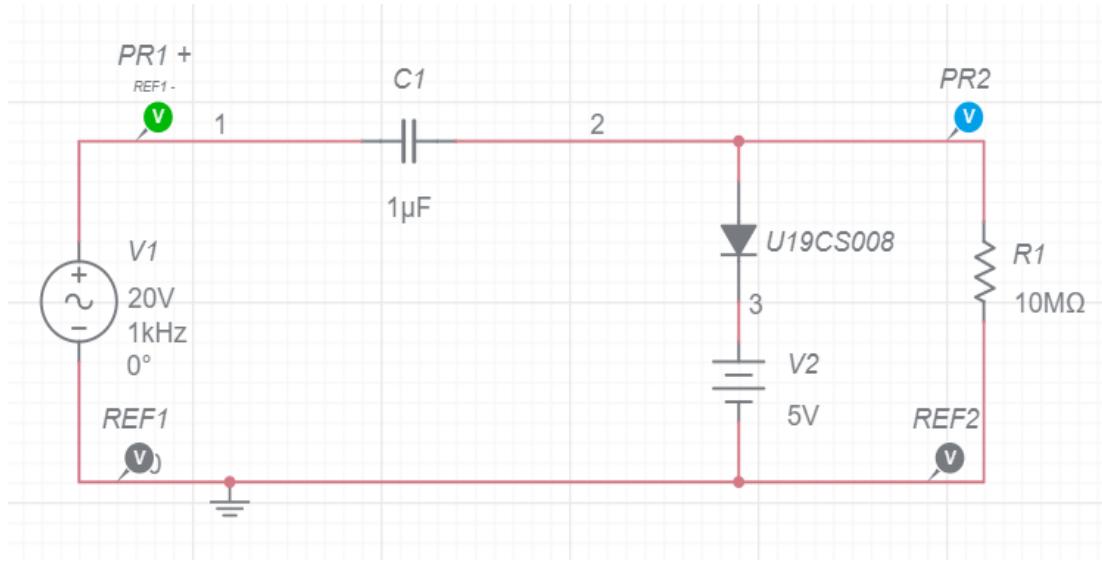
WAVEFORMS (FROM MULTISIM)



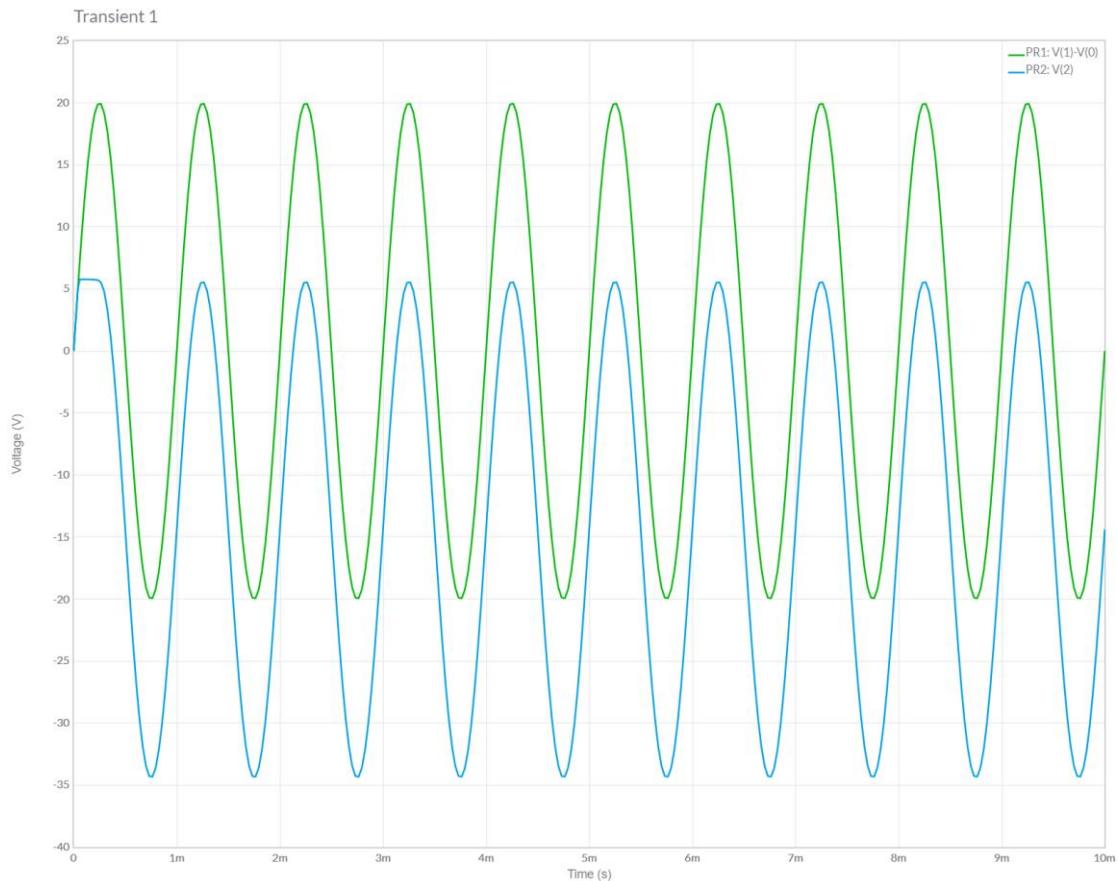


NEGATIVE CLAMPER WITH POSITIVE BIAS

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



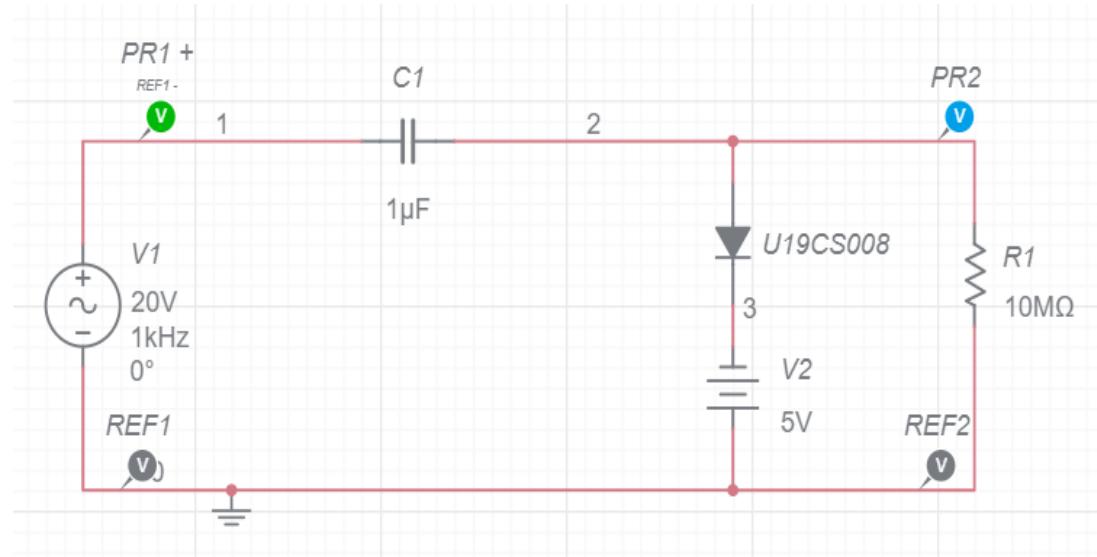
WAVEFORMS (FROM MULTISIM)



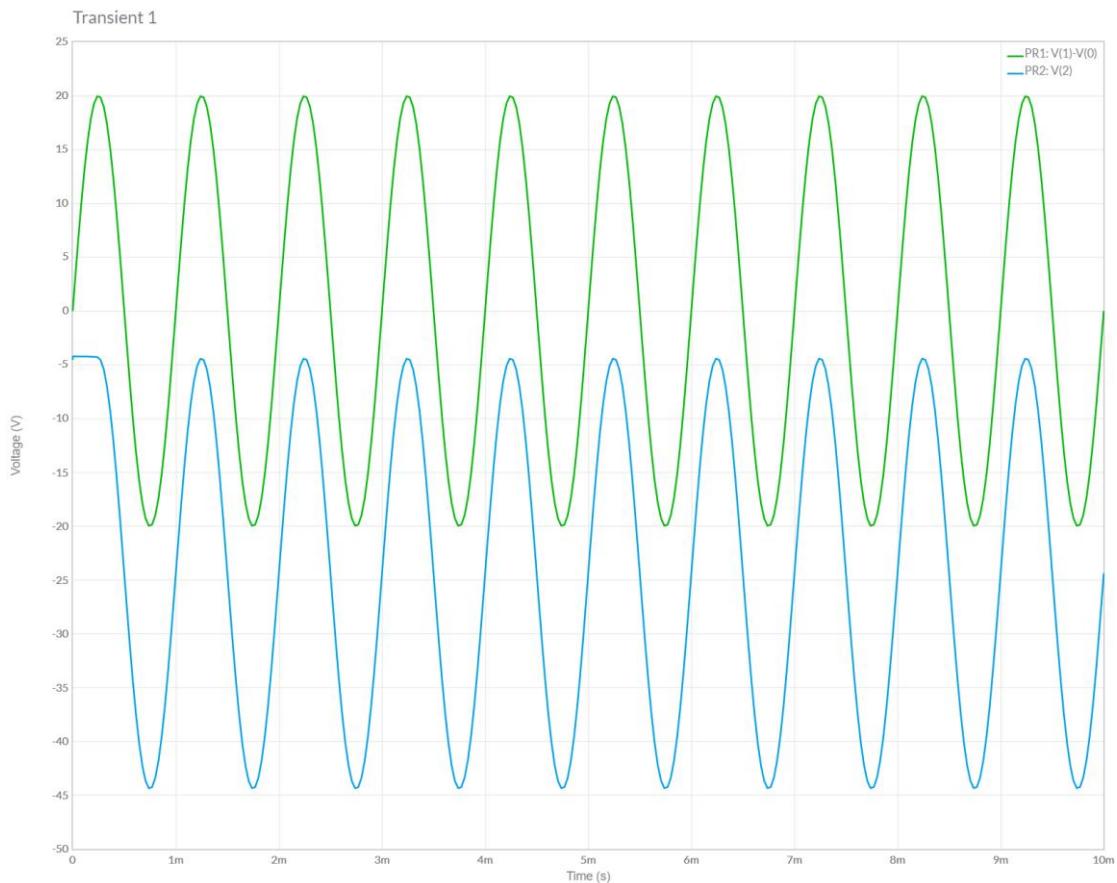


NEGATIVE CLAMPER WITH NEGATIVE BIAS

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



WAVEFORMS (FROM MULTISIM)





CONCLUSIONS

**HERE, THE PRACTICAL AND THEORITICAL CHARACTERISTICS OF
VERIOUS NEGATIVE AND POSITIVE CLAMPER (WITH AND WITHOUT
BIAS) CIRCUITS ARE SAME. HENCE VERIFIED.....**



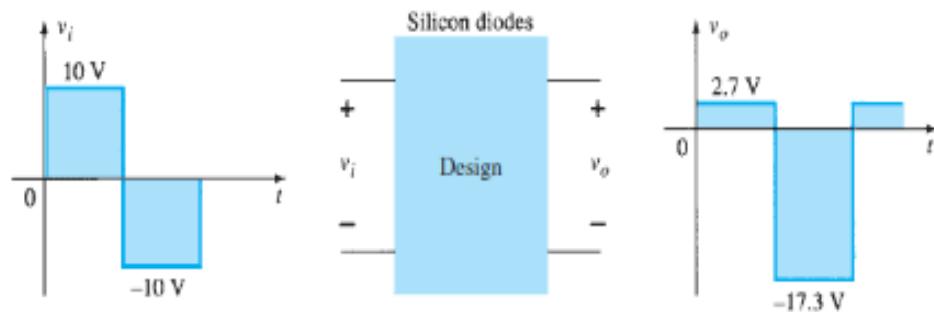
DLED ASSIGNMENT – 8

NAME: KRINA PATEL

ADMISSION NO: U19CS008

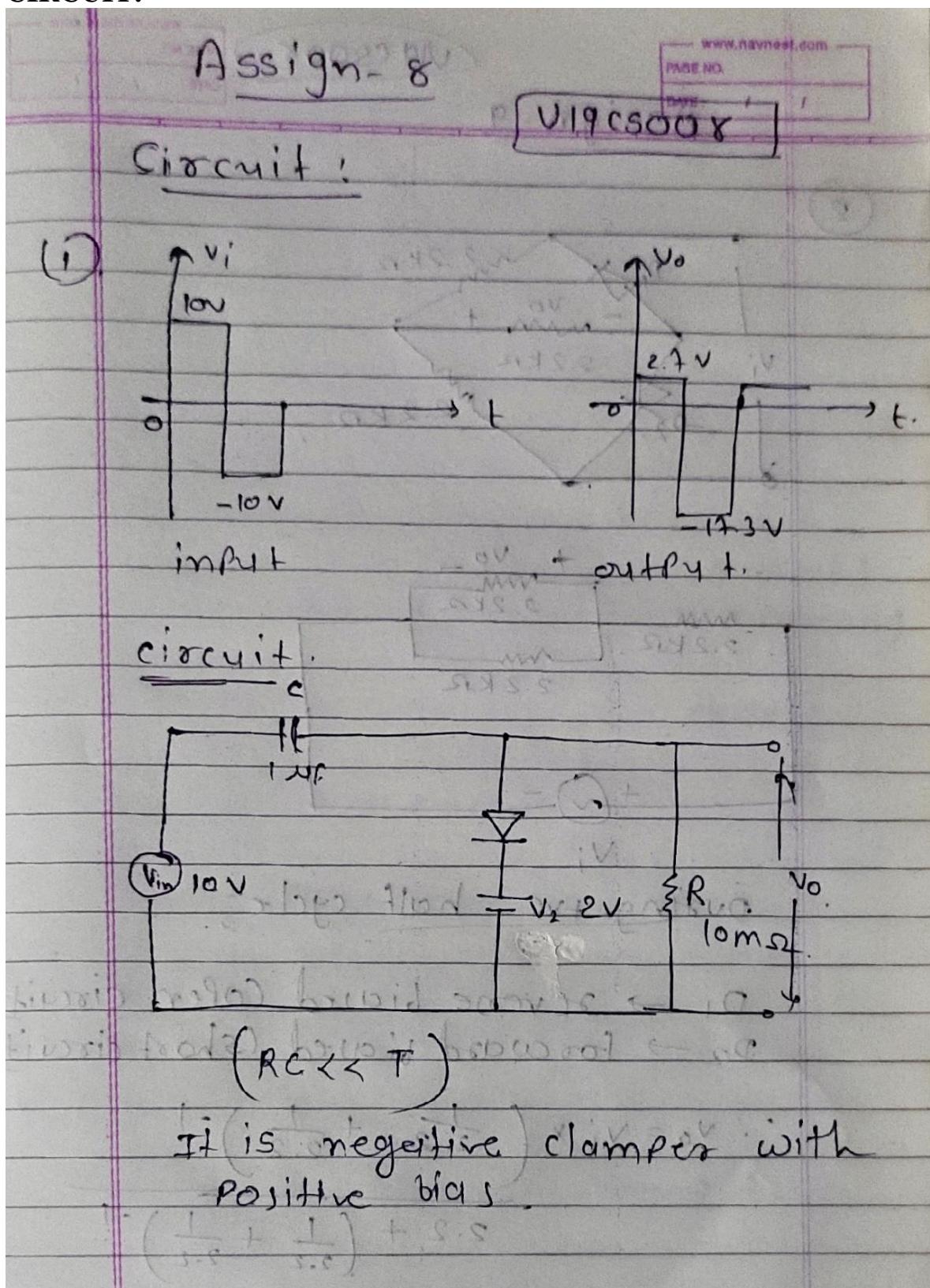
QUESTION – 1

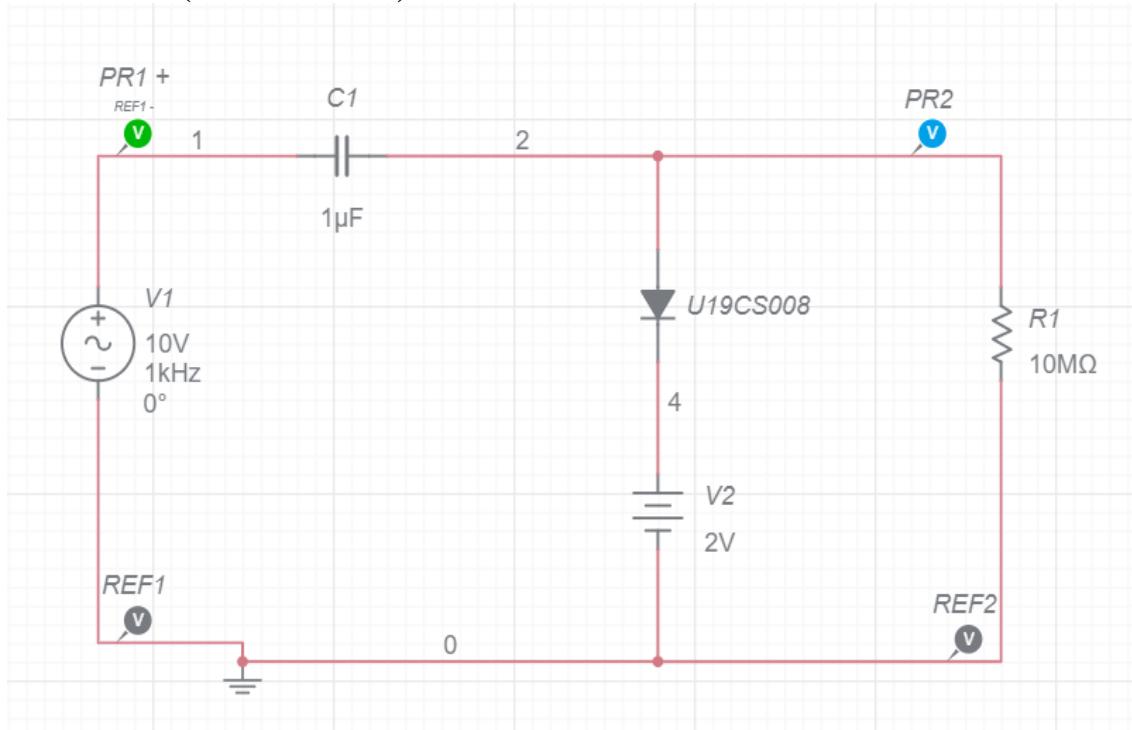
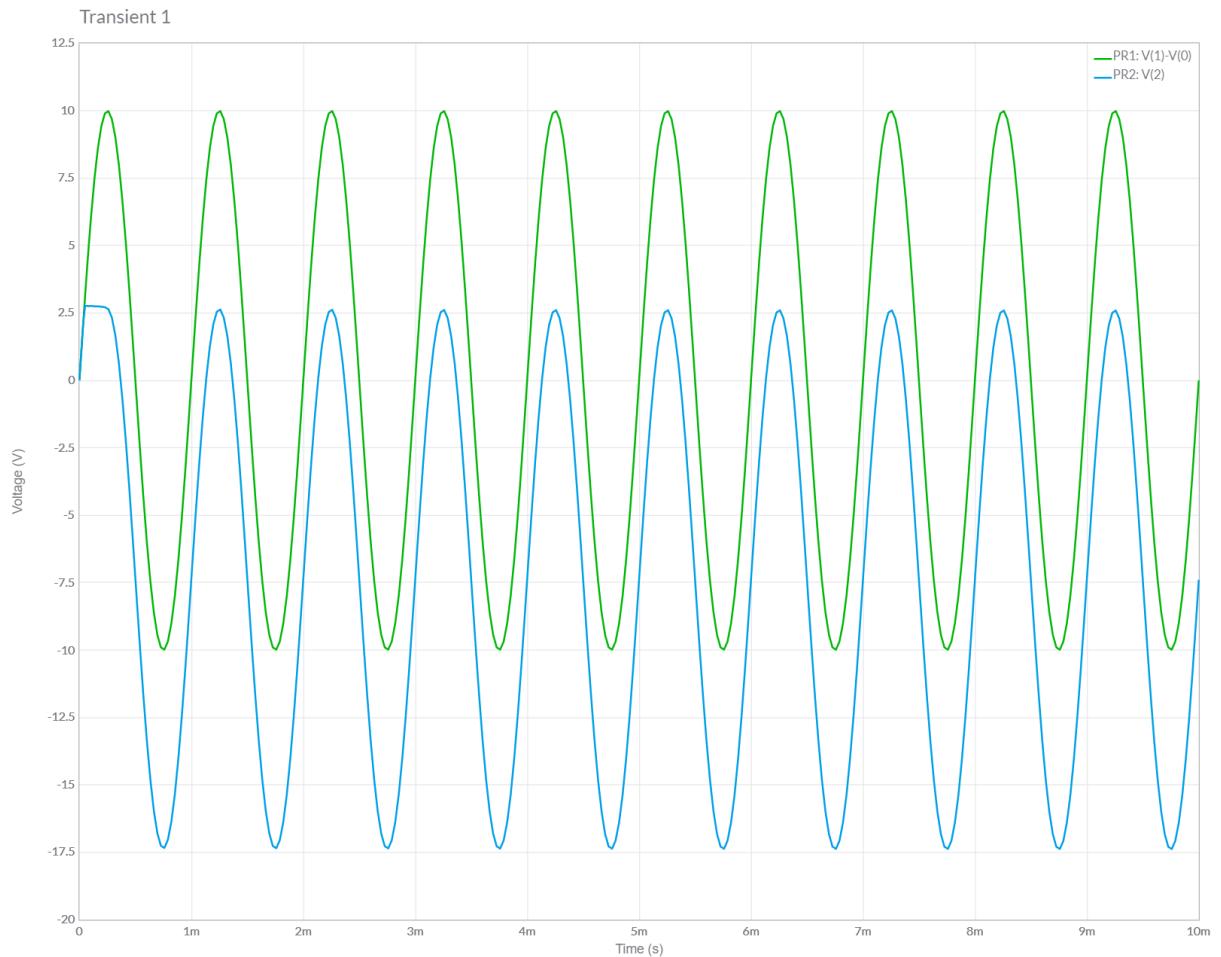
1. Design a Clamper to perform the following operation.





CIRCUIT:



**CIRCUIT (MULTISIM):****GRAPH:**



Expt. No:

9

Date:

22-10-2020

Full Wave Rectifier

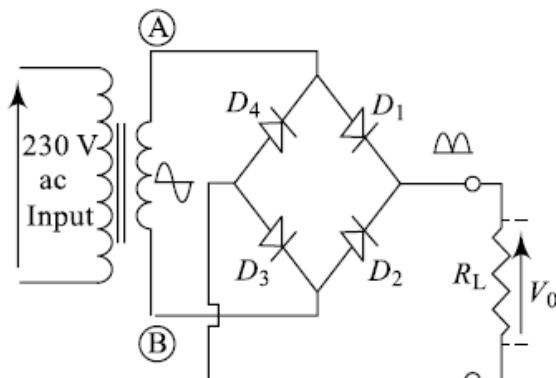
AIM: To study, design and implement Full Wave Rectifier Circuit.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

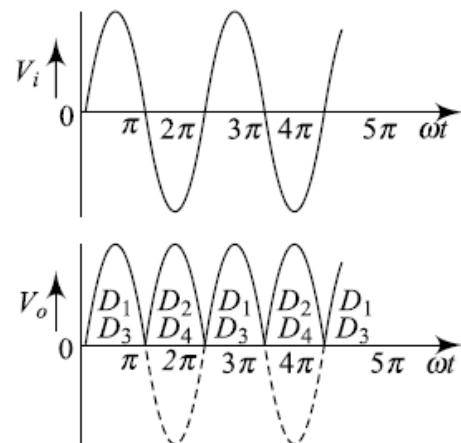
1. Multisim Simulator/Circuit Simulator

THEORY:

The circuit diagram of the Bridge Rectifier along with input-output waveforms is shown in figure below. The four diodes D1, D2, D3 and D4 are arranged in a bridge configuration and hence the name. The circuit contains a transformer that steps down the input ac magnitude depending on the requirement and provides the necessary isolation avoiding any risk of shocks.



(a) Bridge rectifier circuit

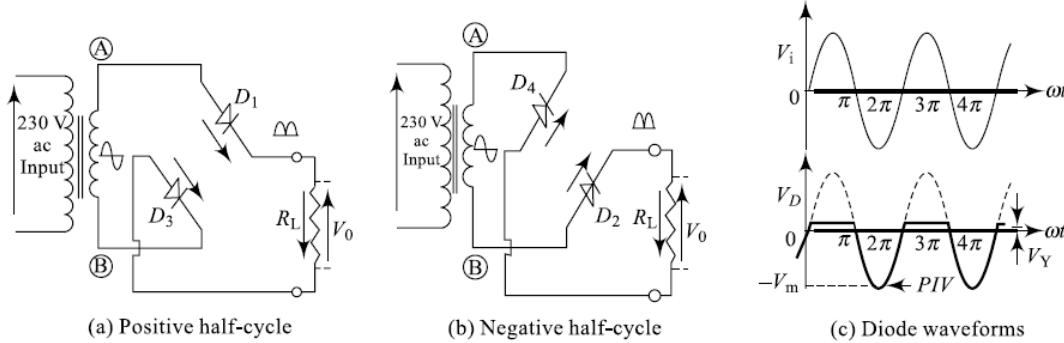


(b) Input/output waveforms

Let $V_i = V_m \sin \omega t$ be the input signal at the transformer secondary, it is a sinusoidal signal with maximum amplitude V_m . During the positive half-cycle of the input, the point A being positive with respect to the point B, the diodes D1 and D3 will be forward biased; however, the diodes D2 and D4 will be reverse biased. The pair of diodes D1 and D3 start conduction resulting in a current I_D flowing through the load resistor R_L in the direction marked for the entire positive half-cycle, i.e. from $\omega t = 0$ to π and the diodes D2 and D4 will be in OFF condition. During the negative half-cycle of the input, the point B will be positive with respect to the point A and the diodes D2 and D4 will be forward biased, and the diodes D1 and D3 will be reverse biased. Diodes D2 and D4 start conduction resulting in a current I_D flowing through the load resistor R_L again in the same direction (as earlier) for the entire negative half-cycle, i.e.

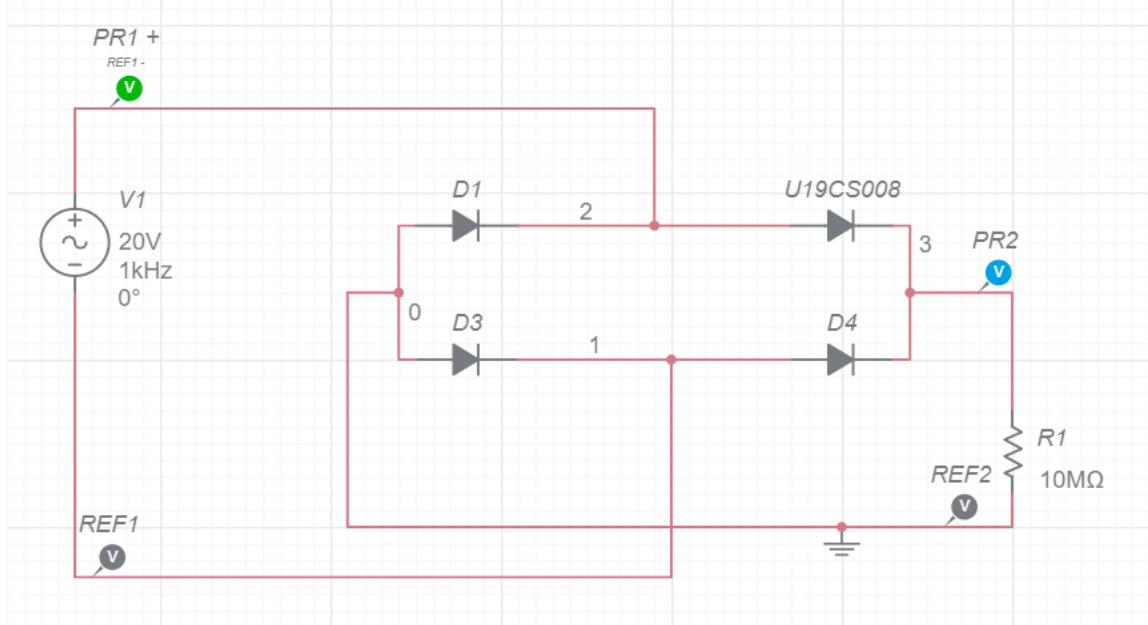


from $\omega t = \pi$ to 2π and the diodes D1 and D3 will be in OFF condition. Thus, between $\omega t = 0$ to π , D1 and D3 conduct and result in an output, between $\omega t = \pi$ to 2π , D2 and D4 conduct and result in an output V_o as indicated in Fig. (b). It can therefore be observed that for both cycles of input, there is a current flowing, hence the name full-wave rectifier.

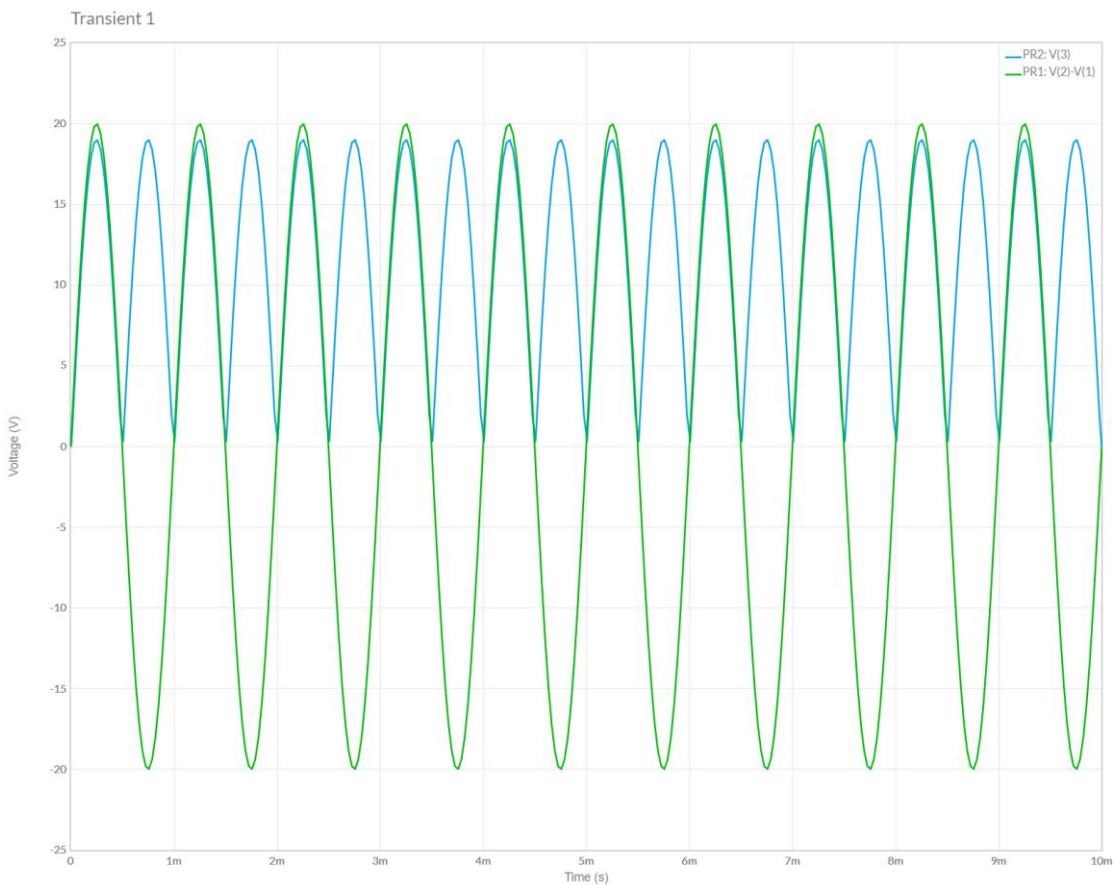




CIRCUIT DIAGRAM (FROM MULTISIM)



WAVEFORMS (FROM MULTISIM)





CONCLUSIONS

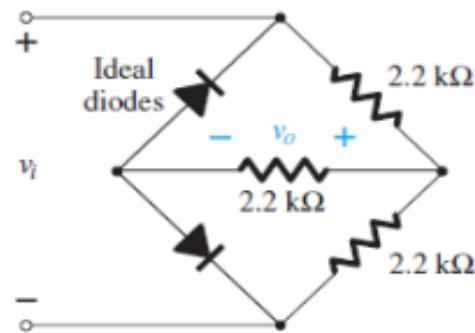
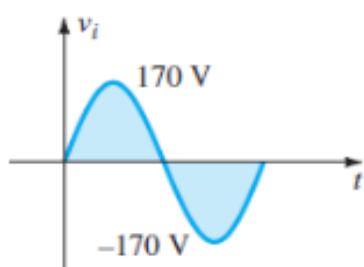
**HERE, THE PRACTICAL AND THEORITICAL CHARACTERISTICS OF
FULL WAVE RECTIFIER CIRCUIT ARE SAME. HENCE VERIFIED.....**



DLED ASSIGNMENT – 9
NAME: KRINA PATEL
ADMISSION NO: U19CS008

QUESTION:

2. Calculate and Plot V_o for the following circuit.

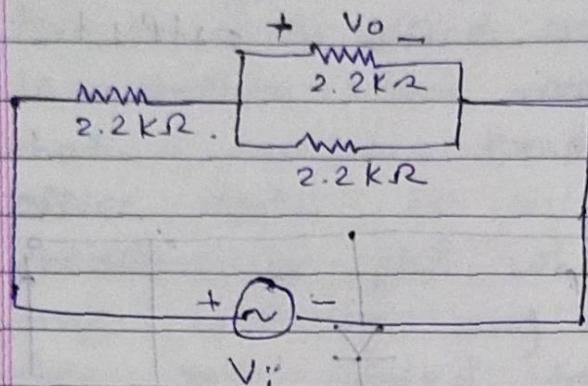
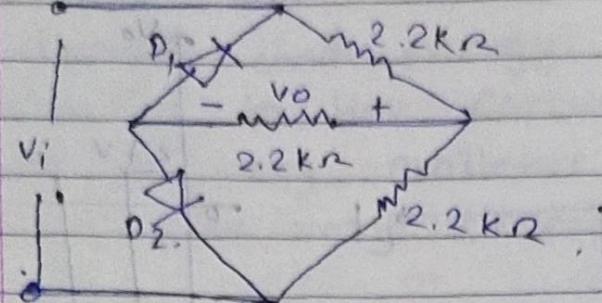




V19 CS008

www.navneet.com
PAGE NO. / /
DATE: / /Assign - 9.

(2)

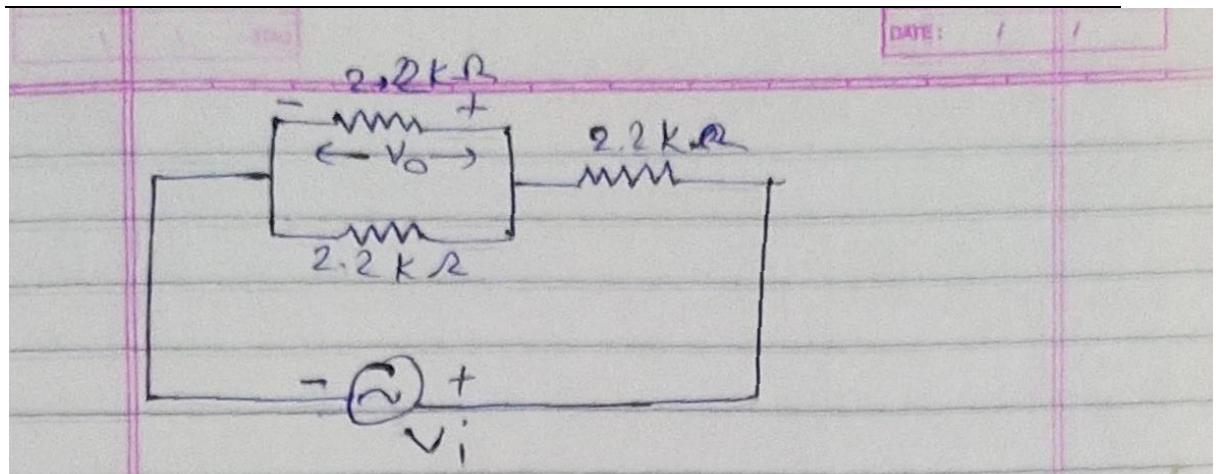
During +ve half cycle,D₁ → reverse biased (open circuit)D₂ → forward biased (short circuit)

$$\therefore V_o = V_i \times \left(\frac{1}{2.2} + \frac{1}{2.2} \right)^{-1}$$

$$= \frac{2.2 + 2.2}{2.2 + \left(\frac{1}{2.2} + \frac{1}{2.2} \right)} = \frac{4.4}{2.4} = 1.8333$$

$$= \frac{170}{3}$$

$$\boxed{V_o = 56.667 \text{ V.}}$$



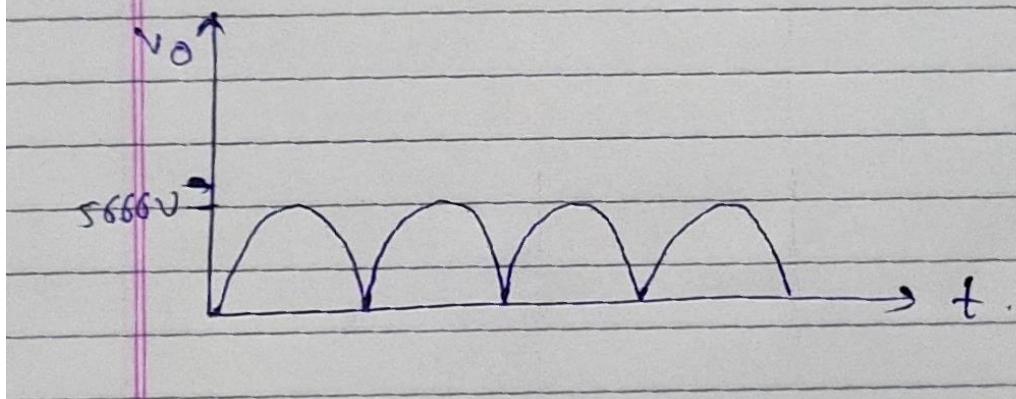
During +ve half cycle,
 $D_1 \rightarrow$ forward biased (short circuit)

$D_2 \rightarrow$ reverse biased (open circuit)

$$\therefore v_o = v_i \times \frac{1.1}{3.3} = 170 \times \frac{1}{3}$$

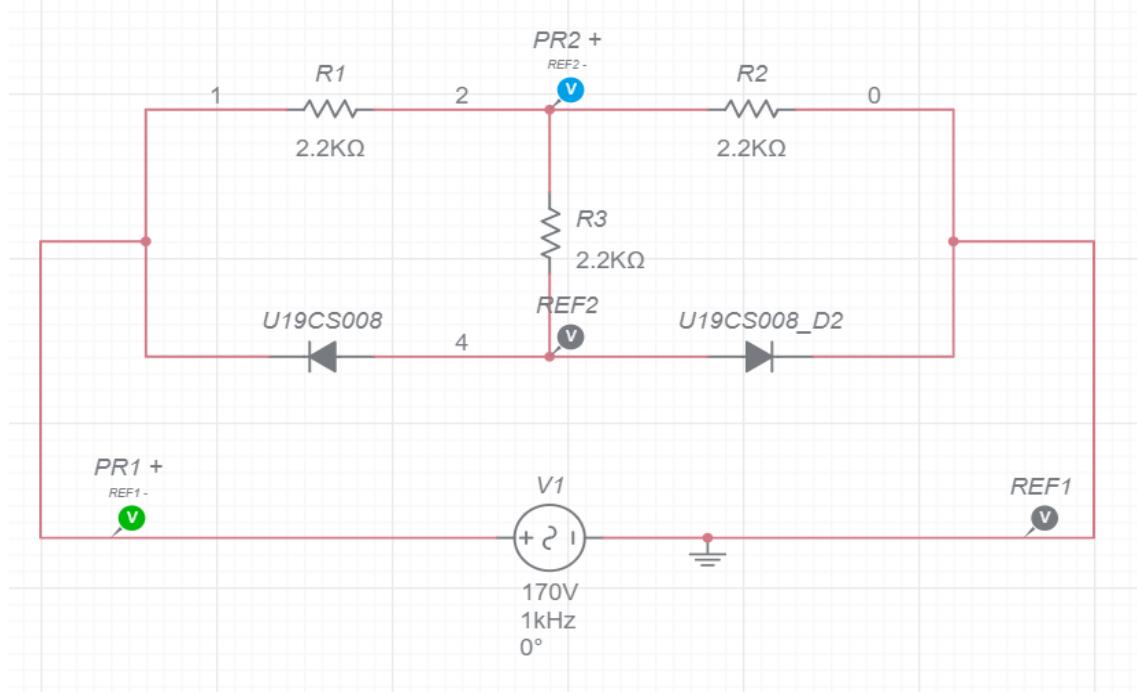
$$v_o = 56.667 \text{ V}$$

So, output graph will be.

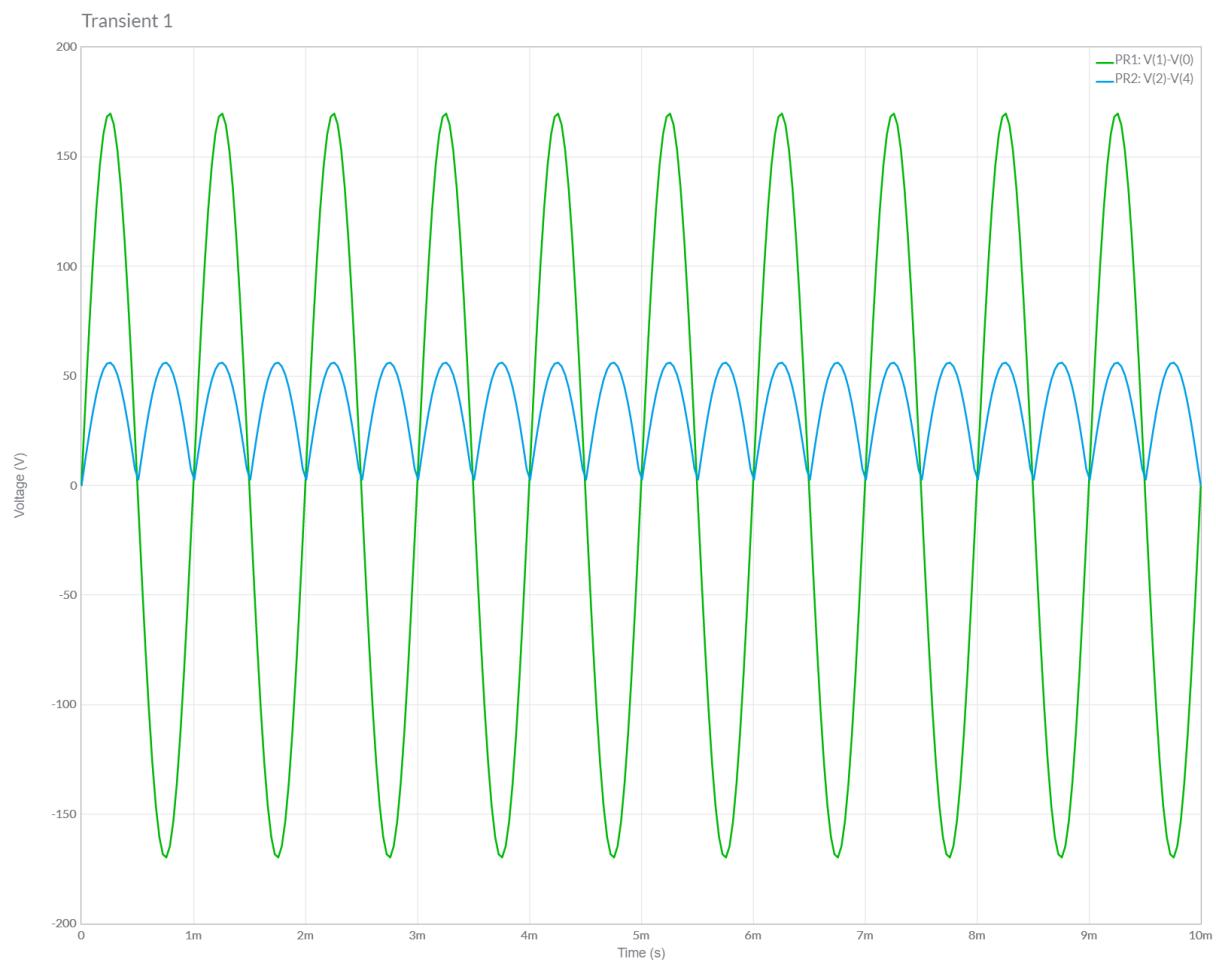




CIRCUIT (MULTISIM)



GRAPH:





Expt. No:

10

Date:

30/10/2020

Common Emitter Characteristics & Common Emitter Amplifier

AIM: To study, the Input-Output characteristics of a BJT in Common Emitter Configuration. Also implement Common Emitter Amplifier.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

- Multisim Simulator/Circuit Simulator

THEORY:

The most frequently encountered transistor configuration appears in Fig.10.1 for the pnp and npn transistors. It is called the common-emitter configuration because the emitter is common to both the input and output terminals (in this case common to both the base and collector terminals). Two sets of characteristics are again necessary to describe fully the behavior of the common-emitter configuration: one for the input or base-emitter circuit and one for the output or collector-emitter circuit. Both are shown in Fig. 10.2 (a) and 10.2 (b) respectively.

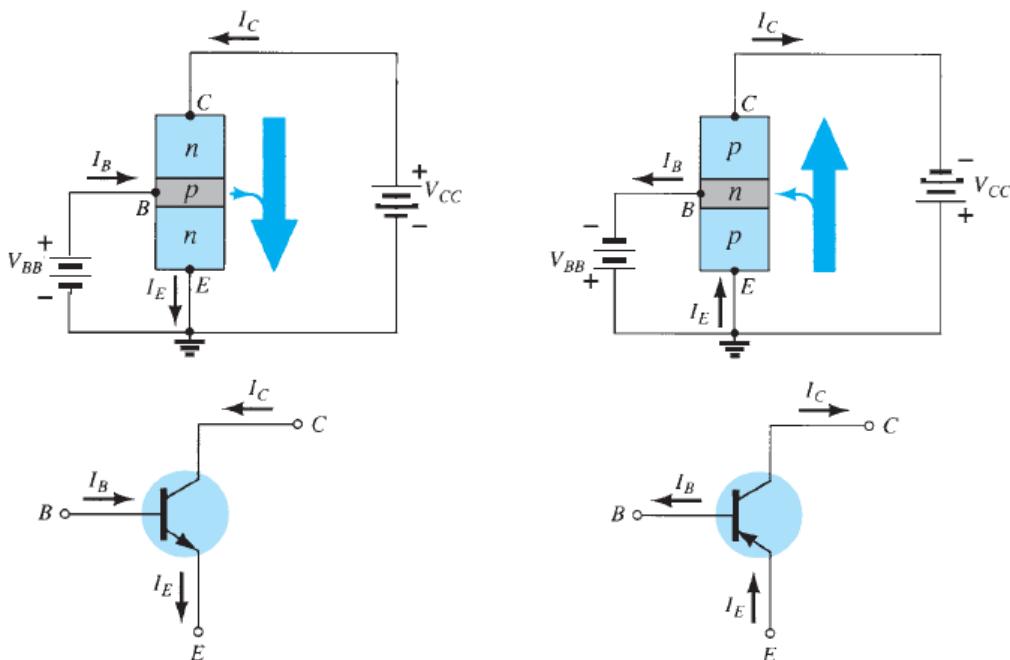


Fig. 10.1

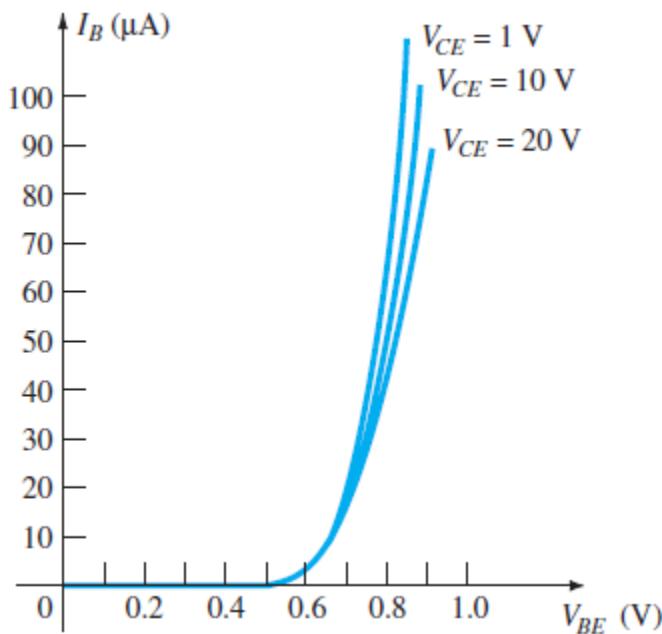


Fig. 10.2 (a) CE Input Characteristics

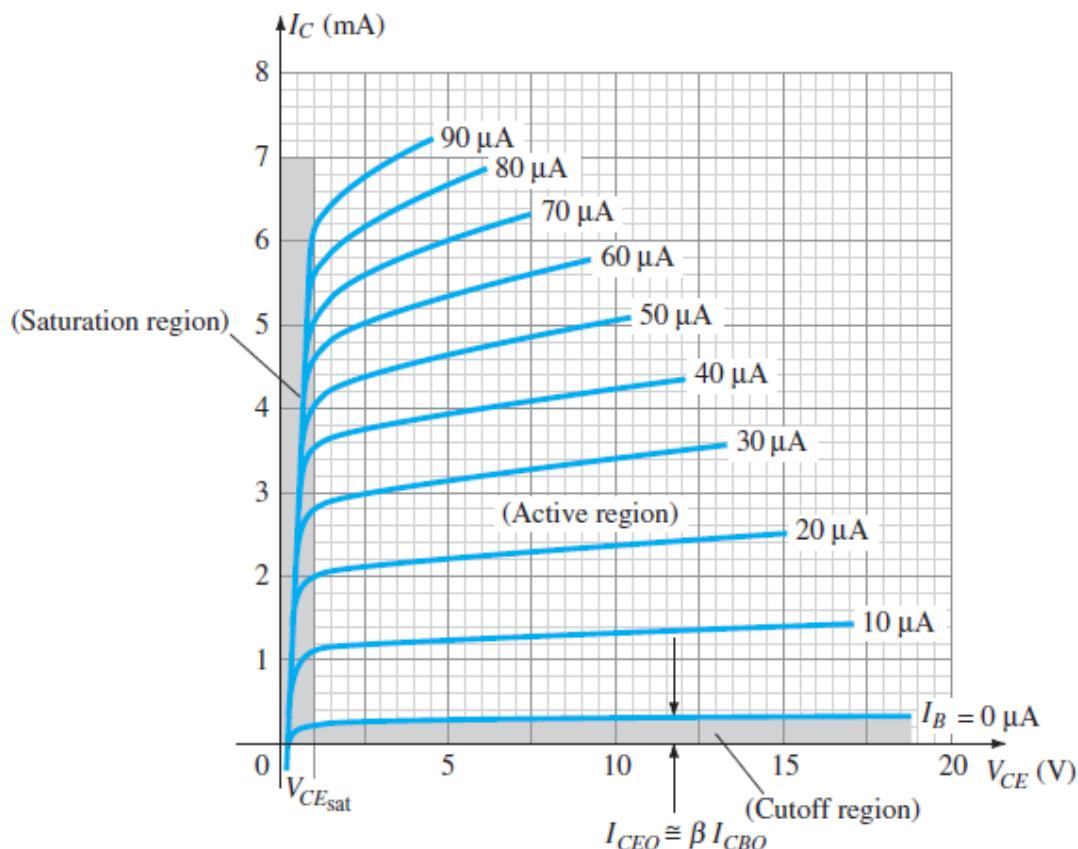


Fig. 10.2 (b) CE Output Characteristics



INPUT CHARACTERISTICS

The input characteristics are a plot of the input current (I_B) versus the input voltage (V_{BE}) for a range of values of output voltage (V_{CE}). The curve describes the changes in the values of input current with respect to the values of input voltage keeping the output voltage constant.

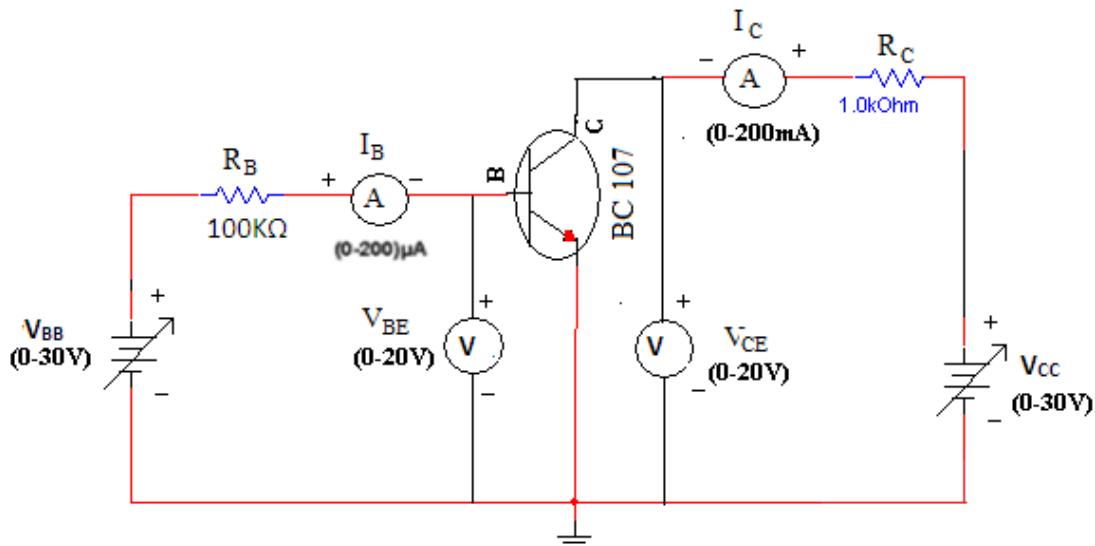


Fig. 10. 3 Circuit Diagram to obtain CE Input/Output Characteristics

PROCEDURE

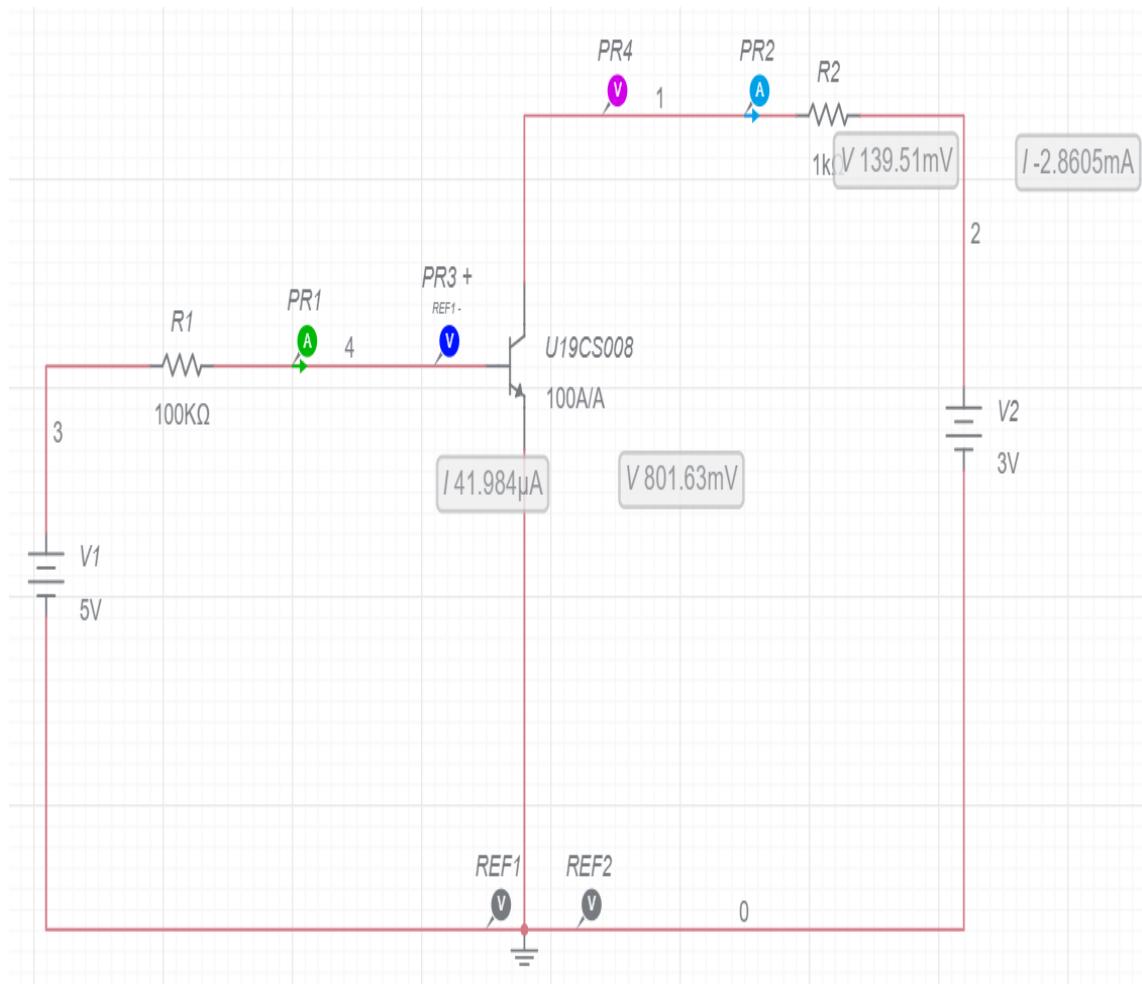
1. CONNECT THE CIRCUIT AS SHOWN IN THE CIRCUIT DIAGRAM.
2. KEEP OUTPUT VOLTAGE $V_{CE} = 0V$ BY VARYING V_{CC} .
3. VARYING V_{BB} GRADUALLY, NOTE DOWN BASE CURRENT I_B AND BASE-EMITTER VOLTAGE V_{BE} .
4. STEP SIZE IS NOT FIXED BECAUSE OF NON LINEAR CURVE. INITIALLY VARY V_{BB} IN STEPS OF 0.1V. ONCE THE CURRENT STARTS INCREASING VARY V_{BB} IN STEPS OF 1V UP TO 5V.
5. REPEAT ABOVE PROCEDURE (STEP 3) FOR $V_{CE} = 3V$.

OUTPUT CHARACTERISTICS

The output characteristics are a plot of the output current (I_C) versus output voltage (V_{CE}) for a range of values of input current (I_B). The curve describes the changes in the values of output current against output voltage keeping the input current constant.

**PROCEDURE**

1. CONNECT THE CIRCUIT AS SHOWN IN THE CIRCUIT DIAGRAM.
2. KEEP Emitter CURRENT $I_B = 0\mu A$ BY VARYING V_{BB} .
3. VARYING V_{CC} GRADUALLY IN STEPS OF 1V UP TO 5V AND NOTE DOWN COLLECTOR CURRENT I_c AND COLLECTOR-EMITTER VOLTAGE(V_{CE}).
4. REPEAT ABOVE PROCEDURE (STEP 3) FOR $I_B = 20\mu A$ AND $60\mu A$.

INPUT CHARACTERISTICS**CIRCUIT DIAGRAM (FROM MULTISIM)**

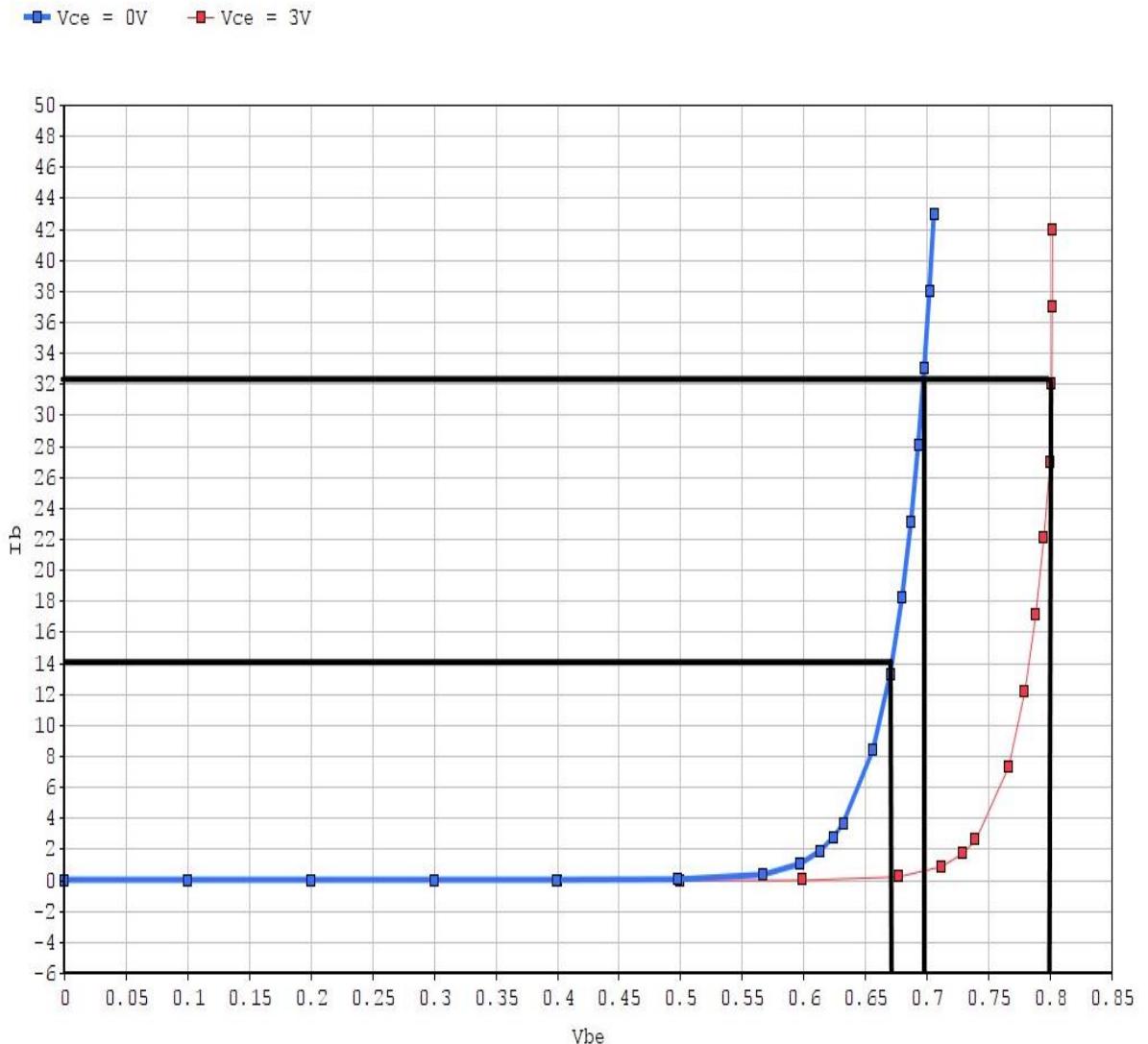


OBSERVATION TABLE

V_{BB}	$V_{CE} = 0V$		$V_{CE} = 3V$	
	V_{BE} (IN VOLTS)	I_B (IN μA)	V_{BE} (IN VOLTS)	I_B (IN μA)
0	0	0	0.00000030001	-0.0000030001
0.1	0.1	0.000000206	0.1	-0.0000027991
0.2	0.2	0.000000632	0.2	-0.0000025958
0.3	0.3	0.0000116	0.3	-0.000002281
0.4	0.39995	0.0000526	0.4	0.0000030102
0.5	0.4977	0.0230	0.49998	0.0000024647
0.6	0.56696	0.330	0.59886	0.011367
0.7	0.59702	1.0298	0.67683	0.23169
0.8	0.61316	1.8684	0.71150	0.88505
0.9	0.62398	2.7602	0.72859	1.7141
1.0	0.63211	3.6789	0.73942	2.6058
1.5	0.65648	8.4352	0.76620	7.3380
2.0	0.67033	13.297	0.77936	12.206
2.5	0.67995	18.200	0.78811	17.119
3.0	0.68727	23.127	0.79466	22.053
3.5	0.69314	28.069	0.79988	27.001
4.0	0.69801	33.020	0.80116	31.988
4.5	0.70218	37.978	0.80144	36.986

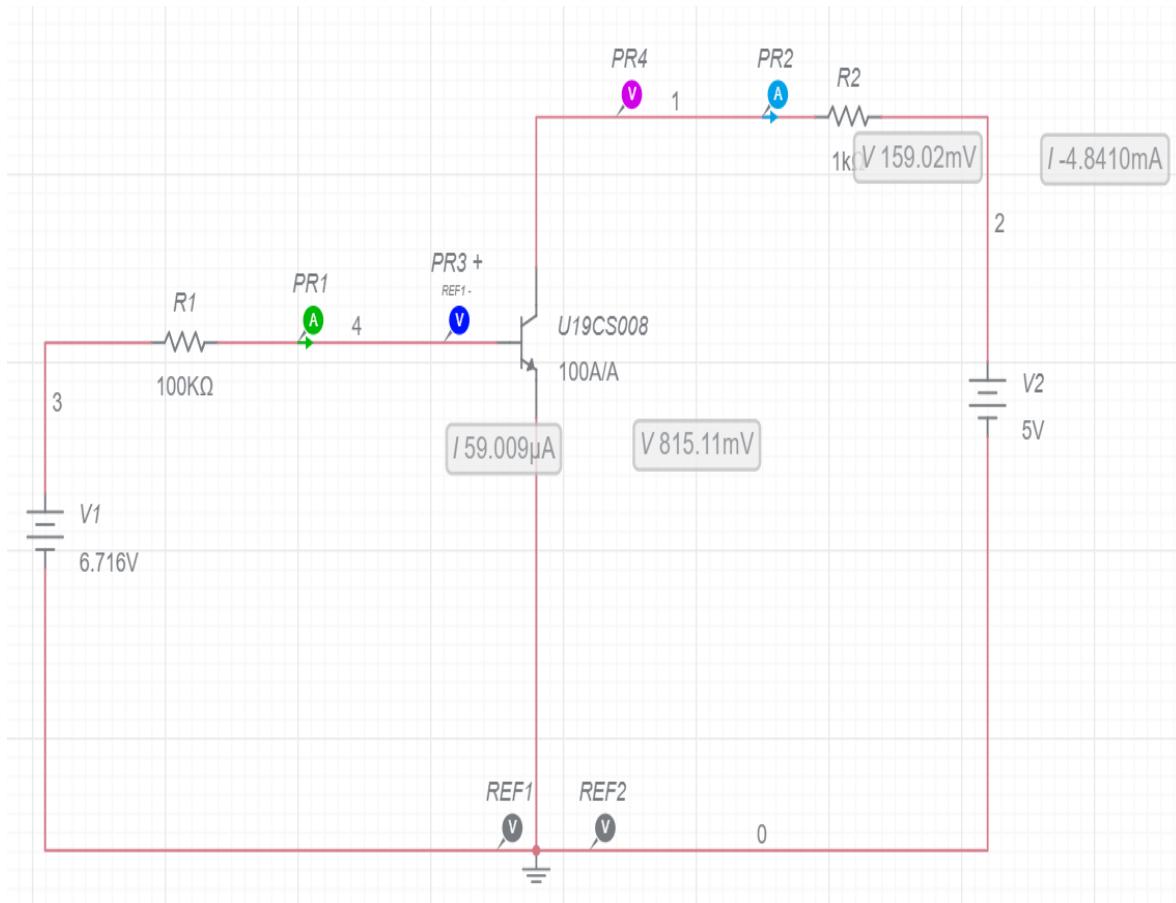


5.0	0.7058	42.942	0.80163	41.984
-----	--------	--------	---------	--------

GRAPH**CALCULATIONS**

$$\begin{aligned}
 \text{Input impedance} &= h_{ie} = R_i = \frac{\Delta V_{BE}}{\Delta I_B} \quad (\text{VCE} = \text{constant}) \\
 &= \frac{(6.87 - 6.56) * 10^{-1} * 10^6}{23.1 - 8.44} \\
 &= 0.33 * 10^5 \\
 &= 14.66 \\
 &= 2.25 \text{ k}\Omega
 \end{aligned}$$

$$\begin{aligned}
 \text{Reverse voltage gain} &= h_{re} = \frac{\Delta V_{EB}}{\Delta V_{CE}} \quad (\text{IB} = \text{constant}) \\
 &= \frac{3 - 0}{(6.9 - 7.98) * 10^{-1}} \\
 &= 3.6 * 10^2
 \end{aligned}$$

OUTPUT CHARACTERISTICS
CIRCUIT DIAGRAM (FROM MULTISIM)

OBSERVATION TABLE

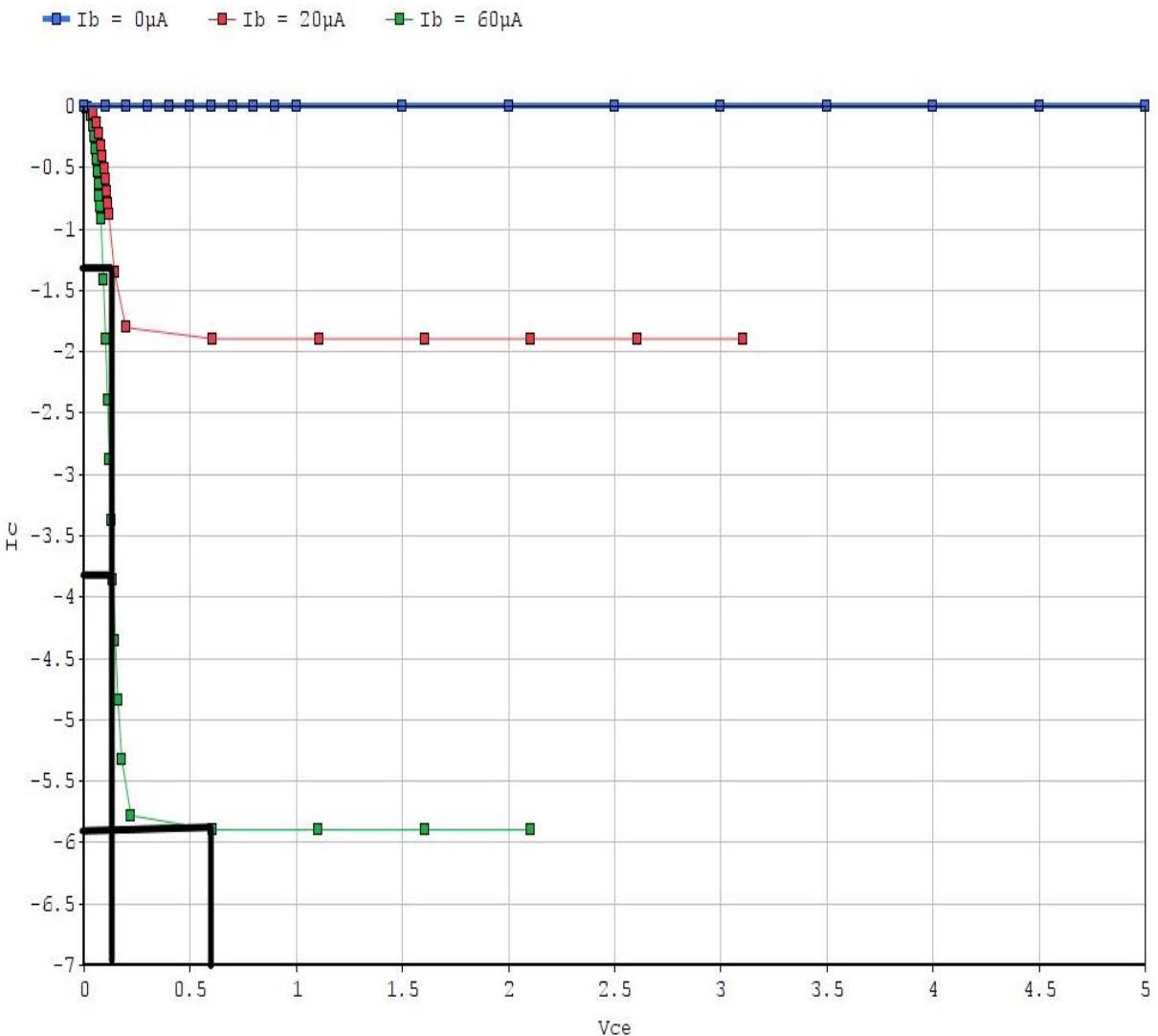
V_{CC}	$I_B = 0 \mu A$		$I_B = 20 \mu A$		$I_B = 60 \mu A$	
	V_{CE} (IN VOLTS)	I_c (IN mA)	V_{CE} (IN VOLTS)	I_c (IN mA)	V_{CE} (IN VOLTS)	I_c (IN mA)
0	0	0	0.010196	-0.01019	0.014528	-0.014528
0.1	0.1	-0.00000000020020	0.042075	-0.057925	0.030123	-0.069877



0.2	0.2	-0.00000000040020	0.059311	-0.14069	0.040552	-0.15945
0.3	0.3	-0.00000000060020	0.071064	-0.22894	0.048373	-0.25163
0.4	0.4	-0.00000000080020	0.080140	-0.31986	0.054647	-0.34535
0.5	0.5	-0.0000000010002	0.087688	-0.41231	0.059905	-0.44010
0.6	0.6	-0.0000000012002	0.094282	-0.50572	0.064448	-0.53555
0.7	0.7	-0.0000000014002	0.10025	-0.59975	0.068464	-0.63154
0.8	0.8	-0.0000000016002	0.10581	-0.69419	0.072076	-0.72792
0.9	0.9	-0.0000000018002	0.11110	-0.78890	0.075369	-0.82463
1	1.0	-0.0000000020002	0.11625	-0.88375	0.078405	-0.92159
1.5	1.5	-0.0000000030002	0.14336	-1.3566	0.091037	-1.4090
2.0	2.0	-0.0000000040002	0.19658	-1.8034	0.10124	-1.8988
2.5	2.5	-0.0000000050002	0.60573	-1.8943	0.11028	-2.3897
3.0	3.0	-0.0000000060002	1.1057	-1.8943	0.11882	-2.8812
3.5	3.5	-0.0000000070002	1.6057	-1.8943	0.12736	-3.3726
4.0	4.0	-0.0000000080002	2.1057	-1.8943	0.13637	-3.8636
4.5	4.5	-0.0000000090002	2.6057	-1.8943	0.14651	-4.3535
5	5.0	-0.000000001	3.1057	-1.8943	0.15902	-4.8410



GRAPH



CALCULATIONS

Output admittance $1/h_{oe} = R_o = \Delta I_C / \Delta V_{CE}$ (IB is constant)

$$= \underline{(5.8954 - 4.841)} * 10^{-3}$$

$$1.1045 - 0.159$$

$$= 1.1152 * 10^{-3} \text{ } \text{V}$$

Forward current gain $= h_{fe} = \Delta I_C / \Delta I_B$ (VCE = constant)

$$= \underline{(5.8954 - 2.1057)} * 10^{-3}$$

$$(60 - 20) * 10^{-6}$$

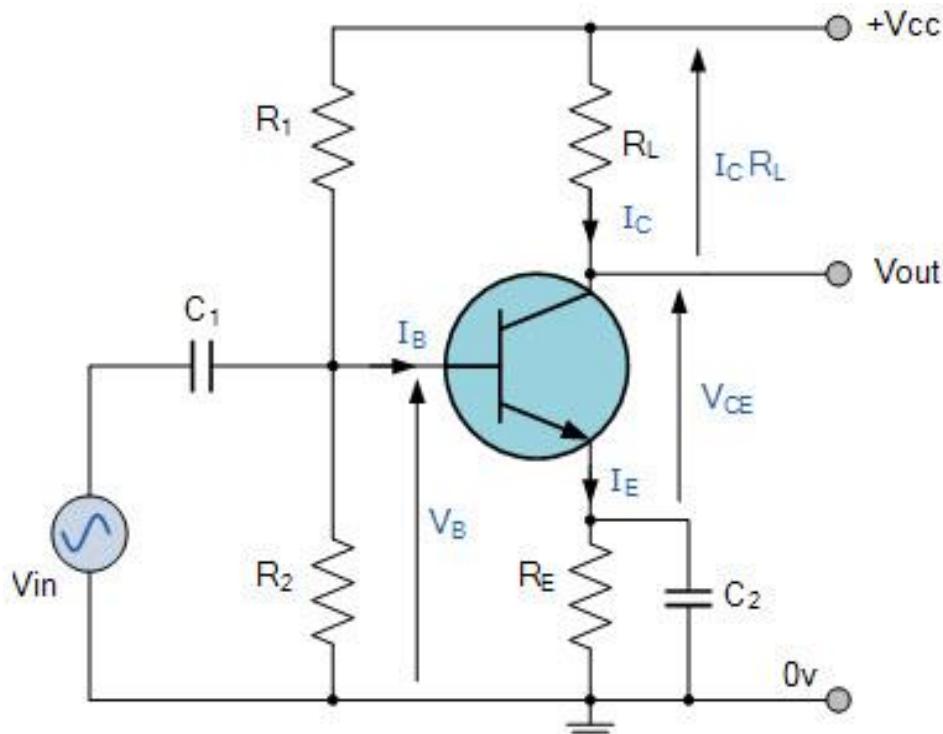
$$= 94.743$$

**PART – B****COMMON Emitter AMPLIFIER**

All types of transistor amplifiers operate using AC signal inputs which alternate between a positive value and a negative value so some way of “presetting” the amplifier circuit to operate between these two maximum or peak values is required. This is achieved using a process known as Biasing. Biasing is very important in amplifier design as it establishes the correct operating point of the transistor amplifier ready to receive signals, thereby reducing any distortion to the output signal.

The aim of any small signal amplifier is to amplify all of the input signal with the minimum amount of distortion possible to the output signal, in other words, the output signal must be an exact reproduction of the input signal but only bigger (amplified).

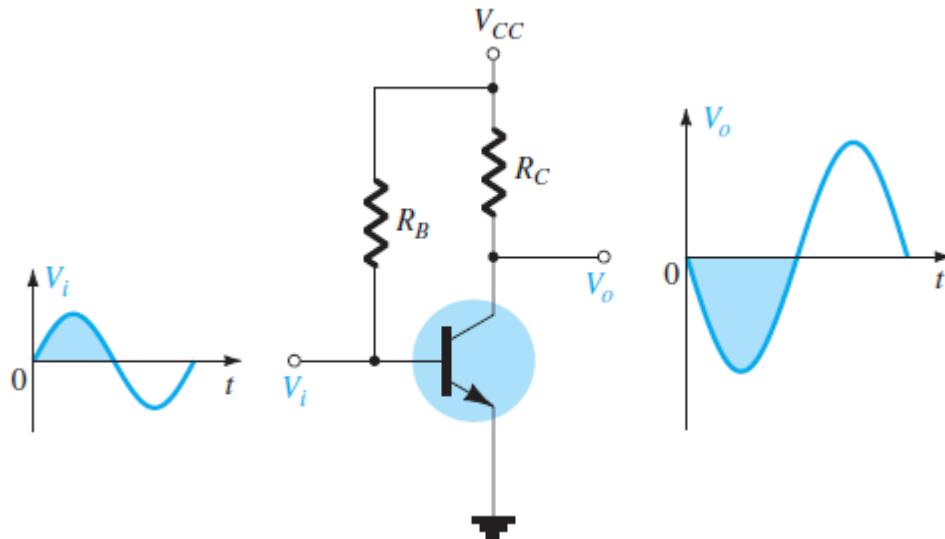
To obtain low distortion when used as an amplifier the operating quiescent point needs to be correctly selected. This is in fact the DC operating point of the amplifier and its position may be established at any point along the load line by a suitable biasing arrangement. The best possible position for this Q-point is as close to the center position of the load line.



Common Emitter Amplifier Circuit

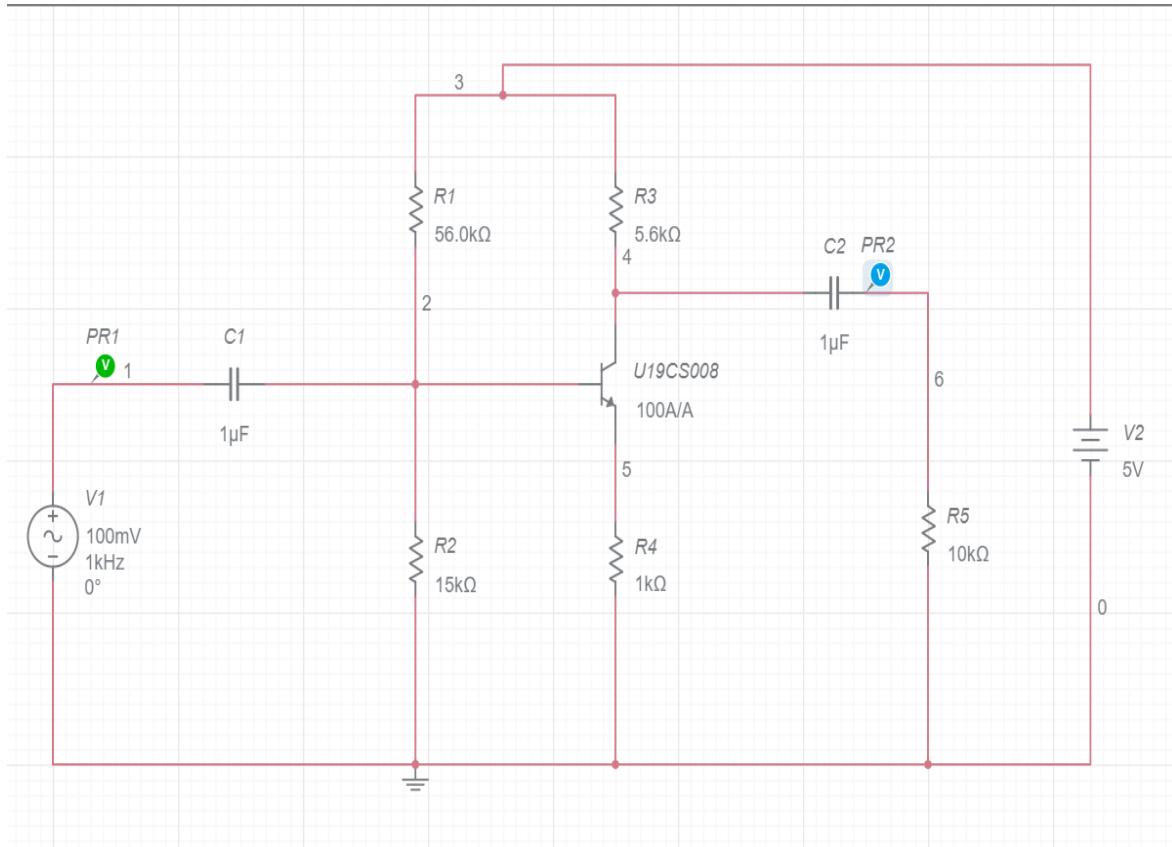
**180 DEGREE PHASE SHIFT**

In CE amplifier configuration, there will always a phase-shift of 180 degrees between the input and output as described in figure below.

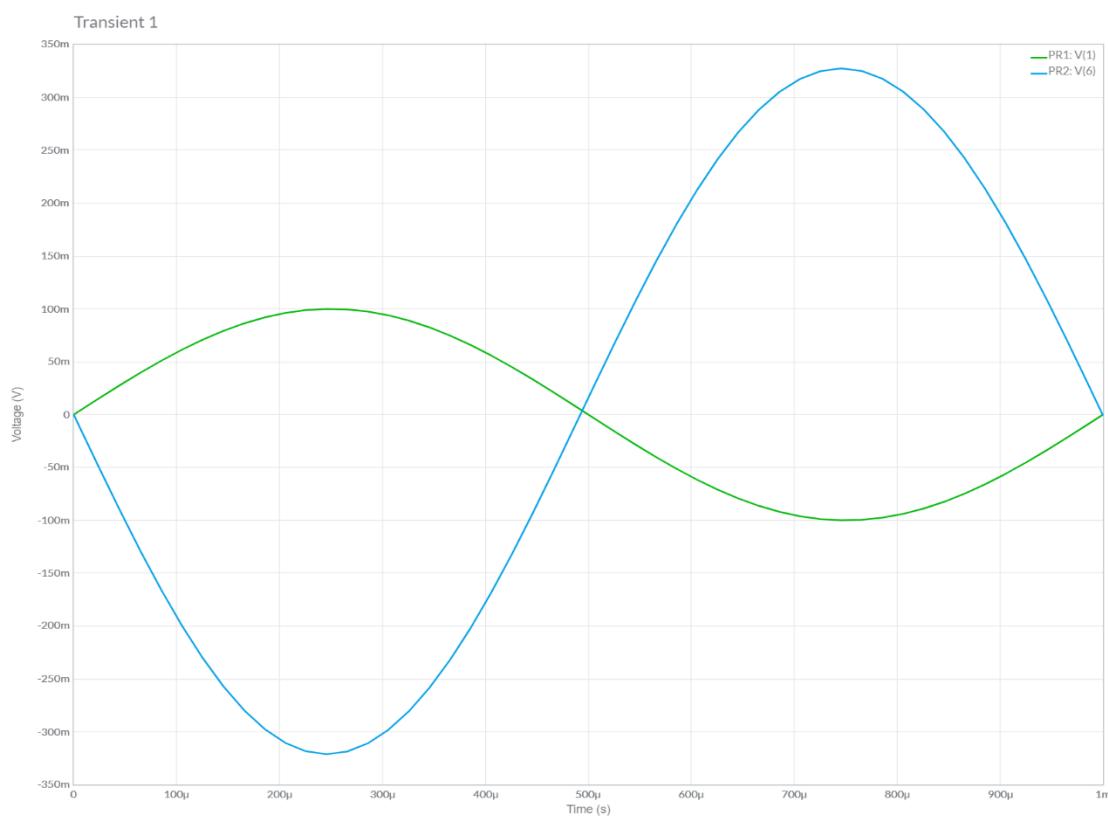




CIRCUIT DIAGRAM FROM MULTISIM



INPUT – OUTPUT WAVEFORMS





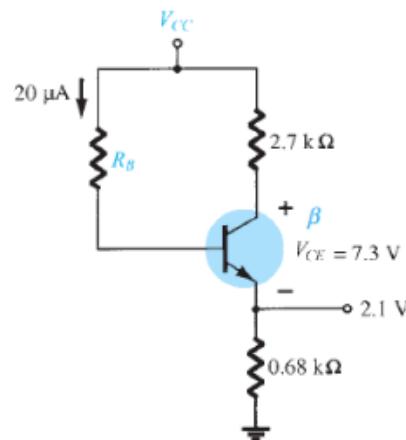
CONCLUSIONS

- ⇒ THE INPUT AND OUTPUT CHARACTERISTICS OF THE BJT TRANSISTOR ARE PLOTTED FROM THE SIMULATION DATA. THUS THE CHARACTERISTICS OF TRANSITOR ARE VERIFIED FROM THE PRACTICAL DATA.

- ⇒ FROM THE OUTPUT OF CE AMPLIFIER GRAPH IT CAN BE OBSERVED THAT THE OUTPUT IS THE INVERTED AND AMPLIFIED FORM OF INPUT.

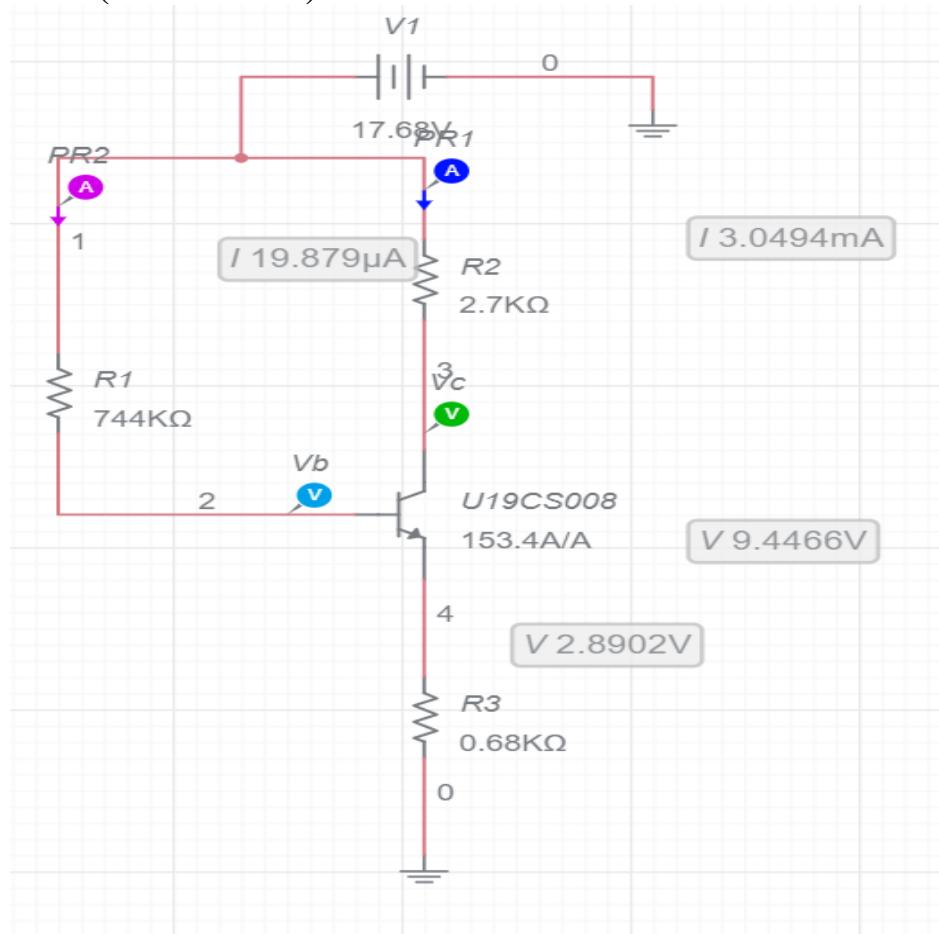
**DLED ASSIGNMENT-10****NAME: KRINA PATEL****ADMISSION NUMBER: U19CS008****PROBLEM STATEMENT-1**

- Using the information provided in the figure below, determine the values of Beta, V_{cc} and R_b theoretically. Also compute and verify the values of I_c , V_b , and V_c by implementing the circuit on Multisim.





CIRCUIT (MULTISIM)



PRACTICAL DATA

Signal	Value
$V_c: V(3)$	9.4466V
$V_b: V(2)$	2.8902V
$PR_1: I(R_2)$	3.0494mA
$PR_2: I(R_1)$	$19.879\mu\text{A}$

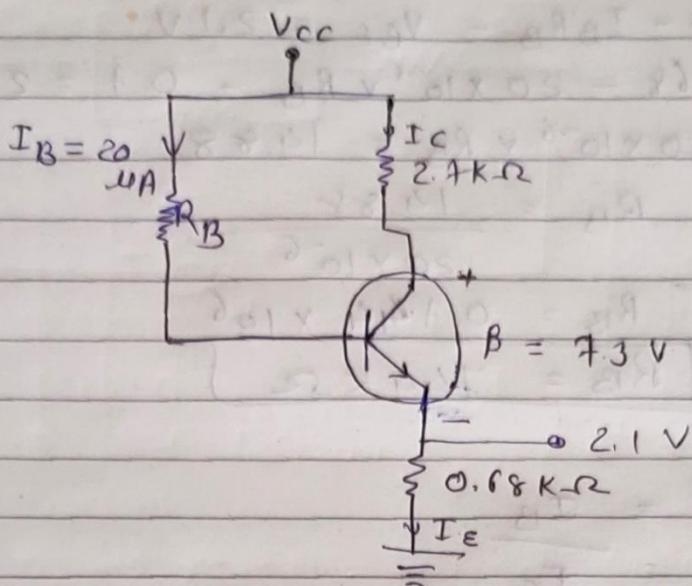
**GRAPH:****CALCULATIONS:**



U19CS008

Assign-10

(1)



$$I_E = \frac{2.1 - 0}{0.68 \times 10^3}$$

$$= 3.088 \text{ mA}$$

now,

$$I_E = I_B + I_C$$

$$\therefore 3.088 \text{ mA} = 0.02 \text{ mA} + I_C$$

$$\therefore I_C = 3.068 \text{ mA}$$

Applying KVL

$$V_{CC} - I_C R_C - V_{CE} = 2.1 V$$

$$\therefore V_{CE} = 9.4 + (3.068 \times 2.7)$$

$$\boxed{V_{CC} = 17.68 V}$$

now,



$$V_{cc} - I_B R_B - V_{BE} = 2.1 \text{ V}$$

$$17.68 - 20 \times 10^{-6} \times R_B - 0.7 = 2.1 \text{ V}$$

$$\therefore 20 \times 10^{-6} \times R_B = 14.88$$

$$R_B = \frac{14.88}{20 \times 10^{-6}}$$

$$R_B = 0.744 \times 10^6$$

$$\therefore R_B = 744 \text{ k}\Omega$$

$$\beta = \frac{I_B}{I_C}$$

$$= \frac{3.068 \times 10^{-3}}{20 \times 10^{-6}} = \frac{3068}{20}$$

$$\therefore \beta = 153.4$$

$$V_B = V_{cc} - I_B R_B$$

$$= 17.68 - (20 \times 10^{-6} \times 744 \times 10^3)$$

$$= 17.68 - 14.83$$

$$\therefore V_B = 2.8 \text{ V}$$

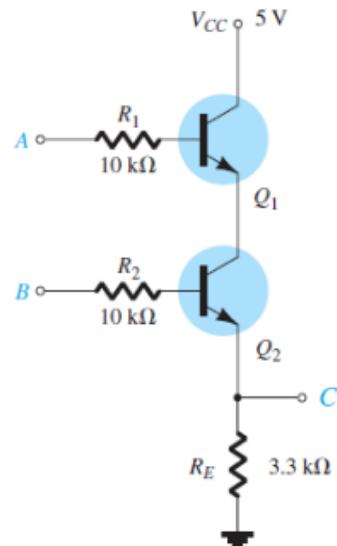
$$V_C = V_{cc} - I_C R_C$$

$$V_C = 17.68 - 8.28$$

$$\therefore V_C = 9.4 \text{ V}$$

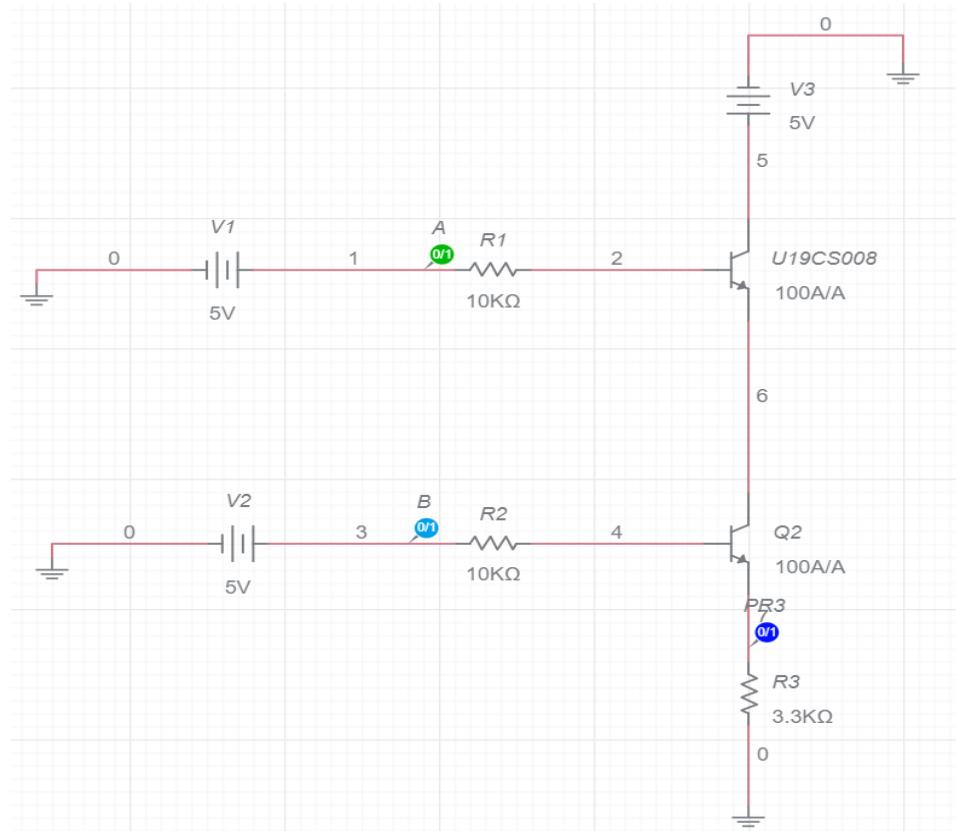
**PROBLEM STATEMENT-2**

2. Simulate the below given circuit on Multisim and predict the type of Digital Logic implemented by the same. Use the default transistor available in Multisim. Attach all the four screenshots (00,05,50,55).

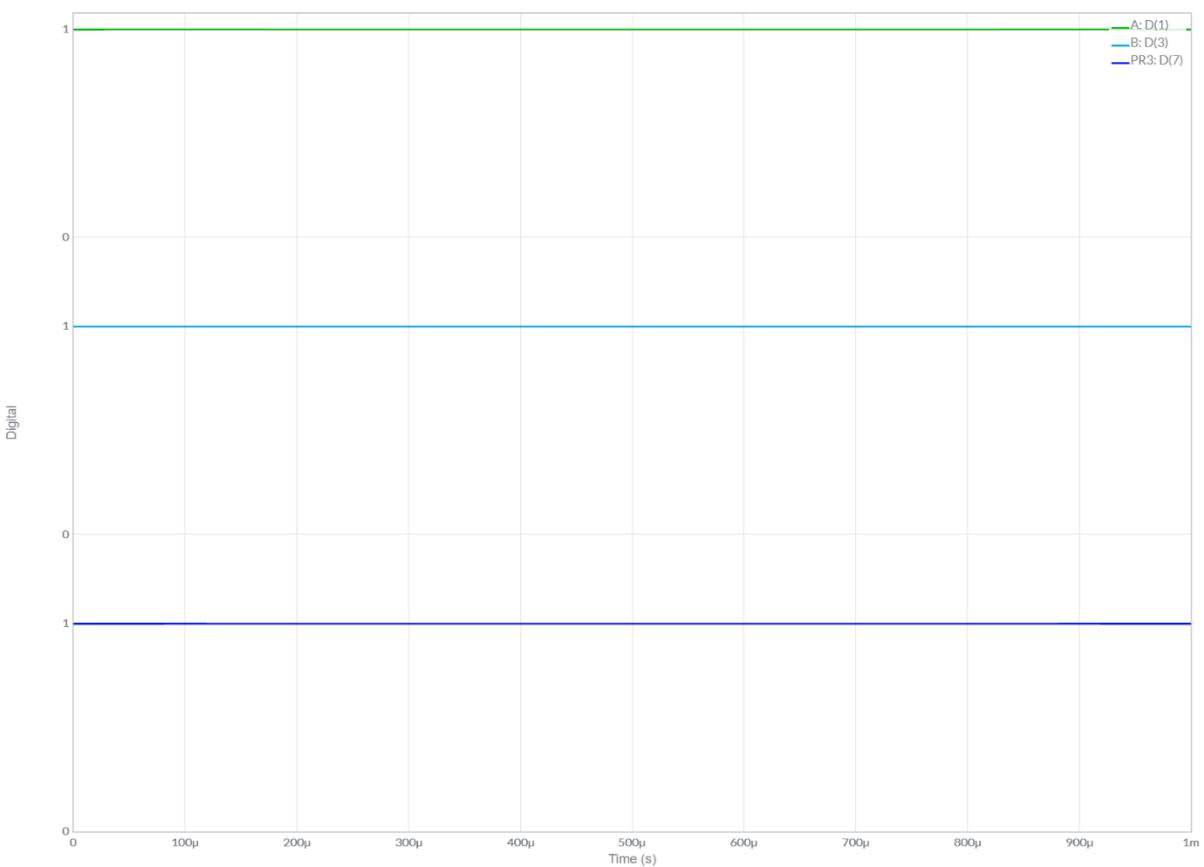




PART-1

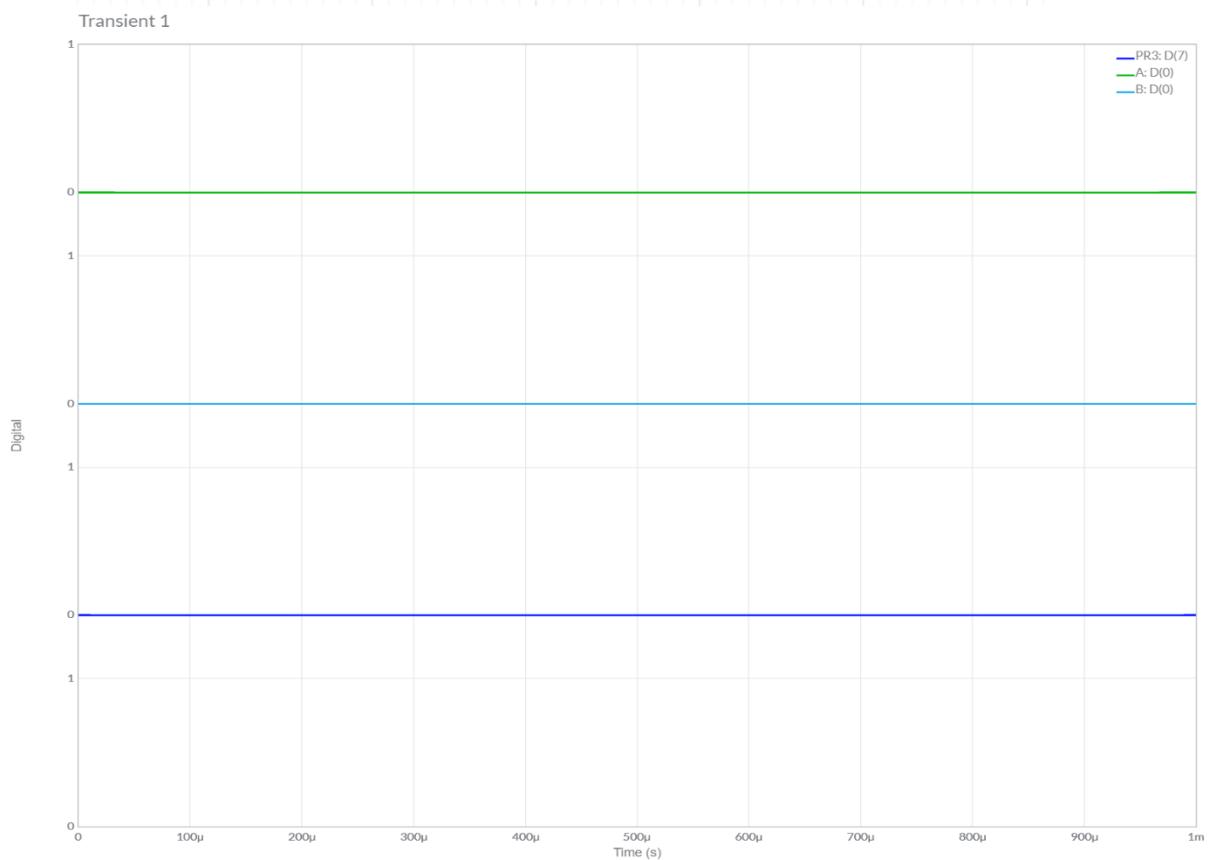
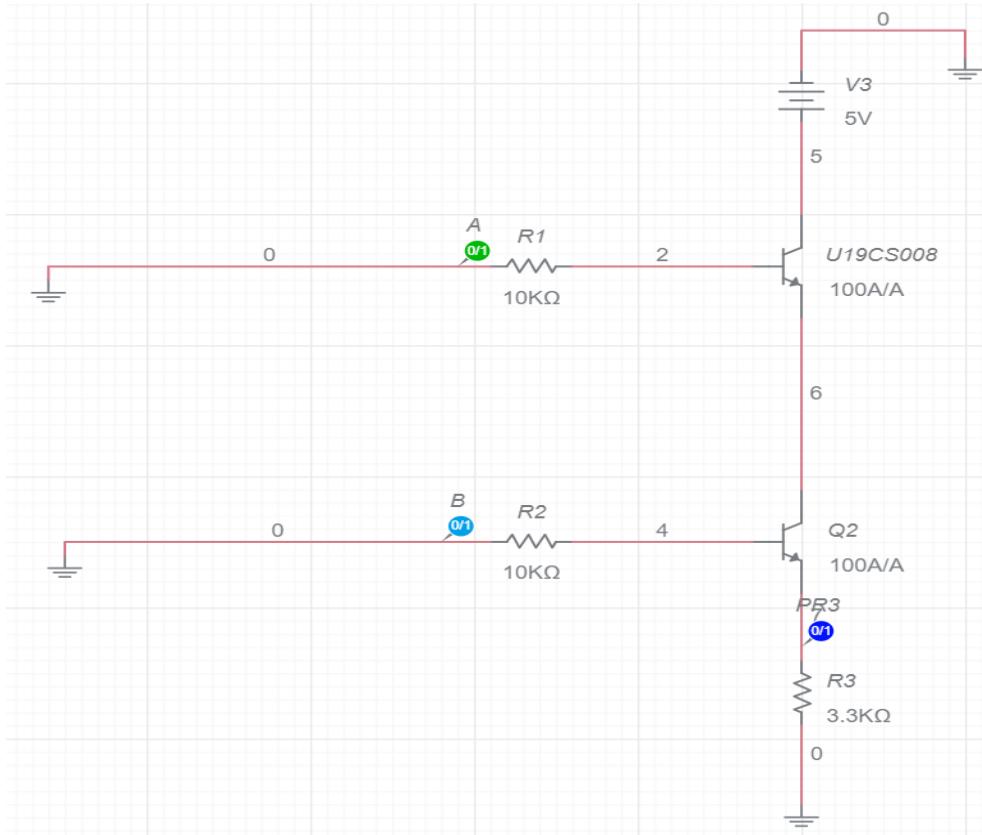


Transient 1



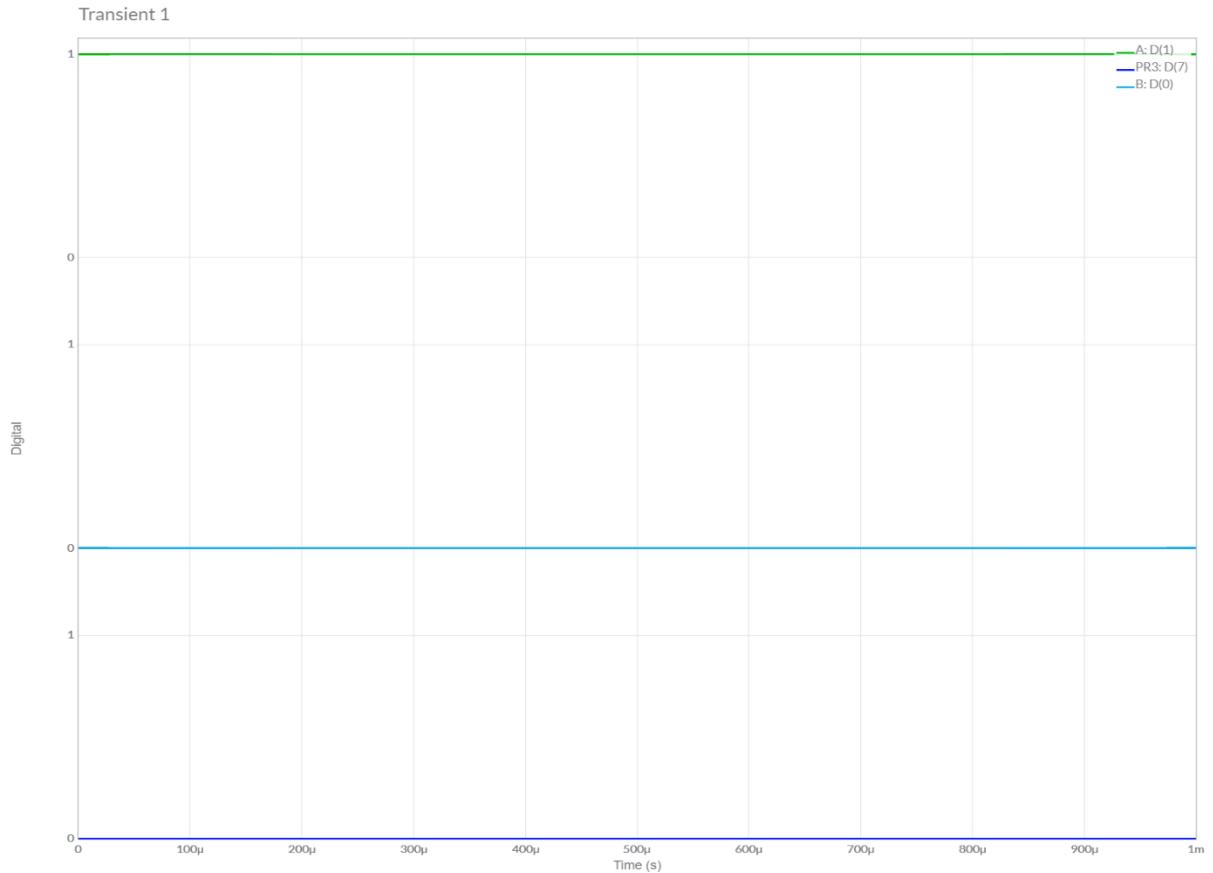
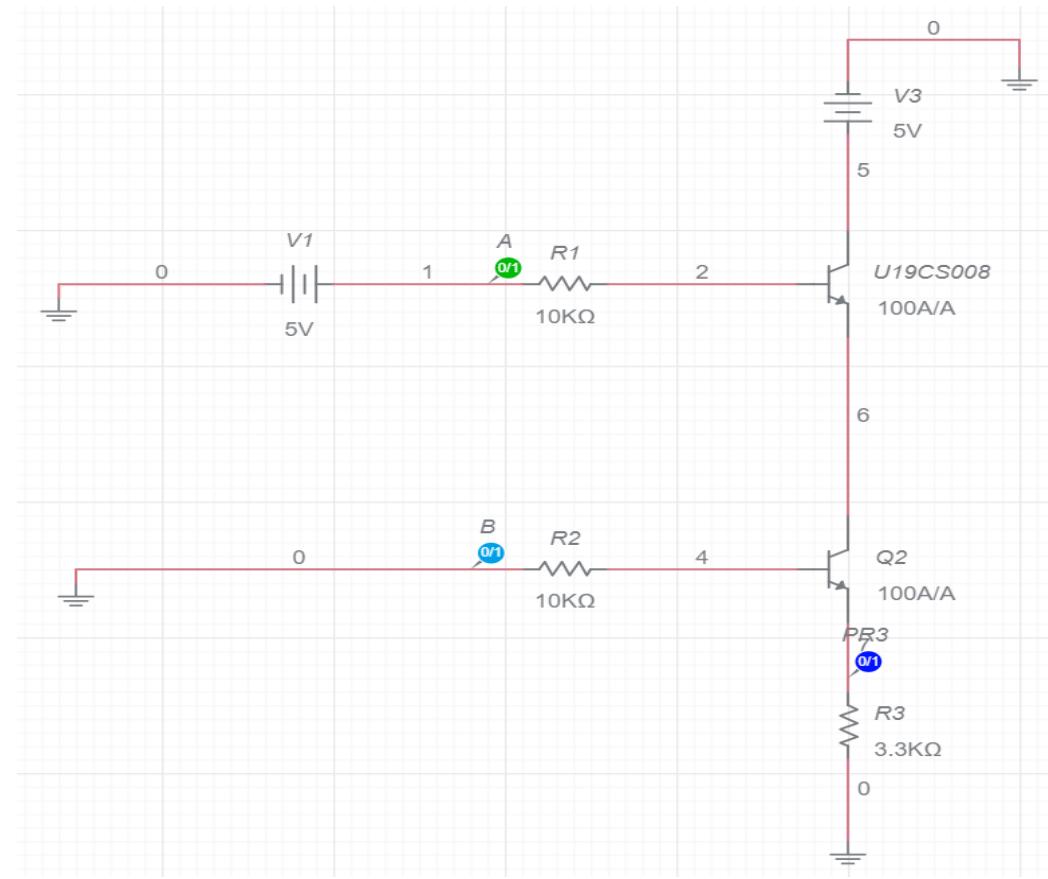


PART-2



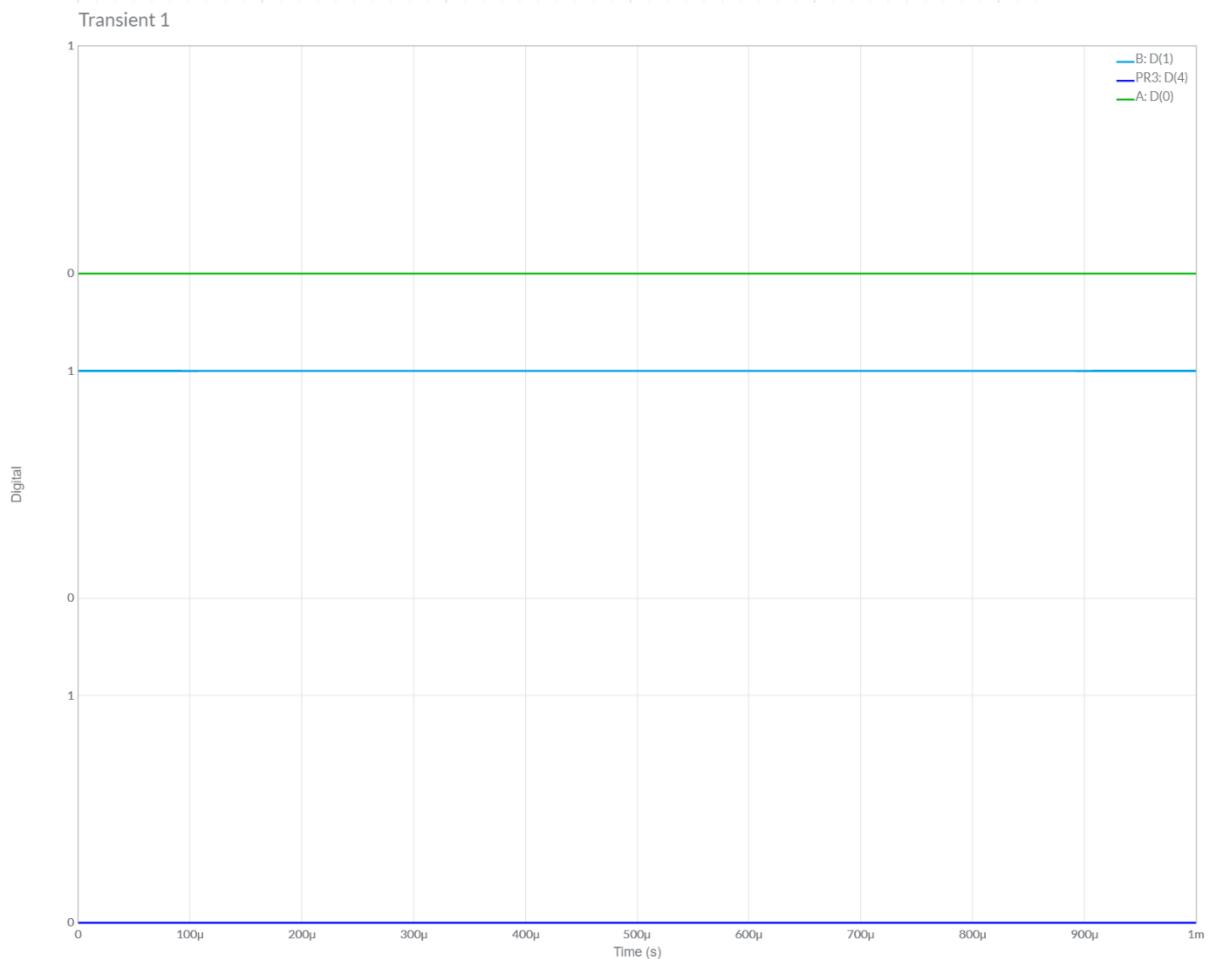
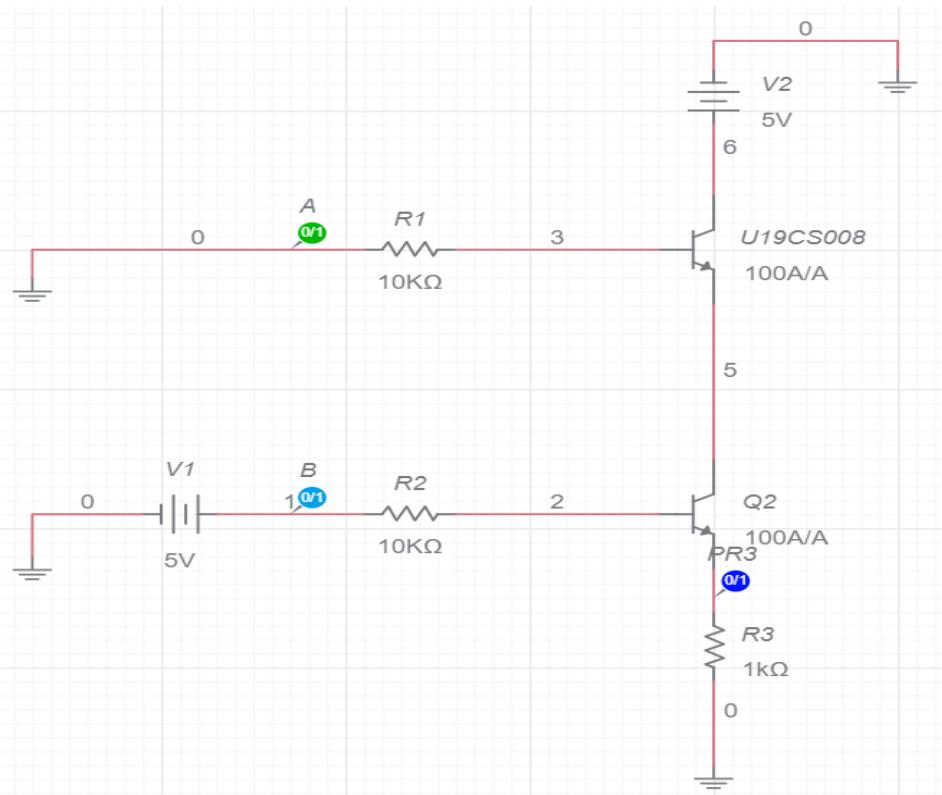


PART-3





PART-4



**TRUTH TABLE:**

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

THIS IS THE TRUTH TABLE OF AND GATE.SO LOOKING GRAPH AND TRUTH TABLE WE CAN SAY THAT THIS CIRCUIT IS BEHAVE AS **AND GATE**.



Expt. No:

11

Date:

07/11/2020**Registers and Counters**

AIM: To study, design and implement 3 – Bit Up Counter, Mod-7 Counter, 4 – Bit Shift Right Register, 4 – Bit Shift Left Register.

SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator/Circuit Simulator

THEORY:**Counters**

A special type of sequential circuit used to count the pulse is known as a counter, or a collection of flip flops where the clock signal is applied is known as counters.

The counter is one of the widest applications of the flip flop. Based on the clock pulse, the output of the counter contains a predefined state. The number of the pulse can be counted using the output of the counter.

Modulus:

To determine the number of flipflops requires to build a counter having a given modulus, identify the smallest integer m that is either equal to or greater than the desired modulus and is also equal to an integral power of 2. For instance, if the desired modulus is 10, which is the case in a decode counter, the smallest integer greater than or equal to 10 and which is also an integral power of 2 is 16. The number of flip-flops in this case would be 4, as $16 = 2^4$.

Modulus $\geq 2^N$

Truth Table

Clock	Counter output		State number	Decimal counter output
	Q _B	Q _A		
Initially	0	0	-	0
1 st	0	1	1	1
2 nd	1	0	2	2
3 rd	1	1	3	3
4 th	0	0	4	0



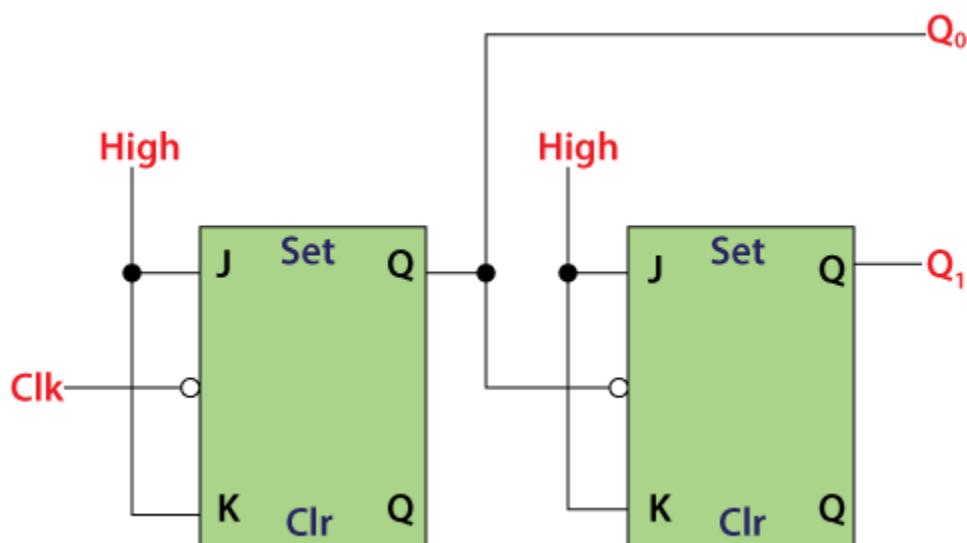
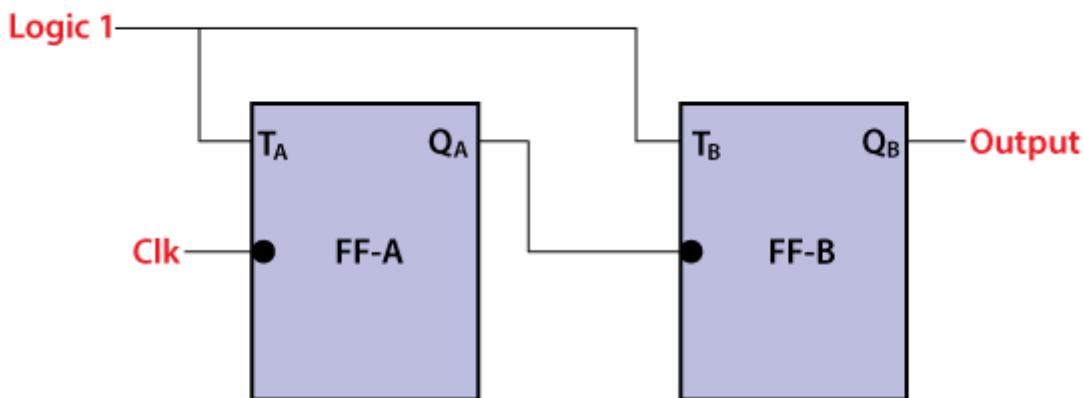
There are the following types of counters:

- Asynchronous Counters
- Synchronous Counters

Asynchronous or ripple counters

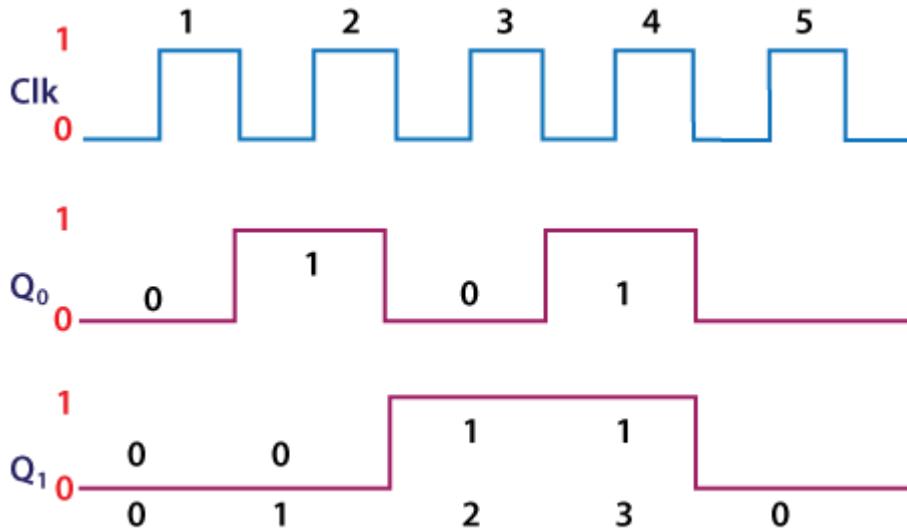
The **Asynchronous counter** is also known as the **ripple counter**. Below is a diagram of the 2-bit **Asynchronous counter** in which we used two T flip-flops. Apart from the T flip flop, we can also use the [JK flip flop](#) by setting both of the inputs to 1 permanently. The external clock pass to the clock input of the first flip flop, i.e., FF-A and its output, i.e., is passed to clock input of the next flip flop, i.e., FF-B.

Block Diagram





Signal Diagram



Operation

1. **Condition 1:** When both the flip flops are in reset condition.

Operation: The outputs of both flip flops, i.e., Q_A Q_B , will be 0.

2. **Condition 2:** When the first negative clock edge passes.

Operation: The first flip flop will toggle, and the output of this flip flop will change from 0 to 1. The output of this flip flop will be taken by the clock input of the next flip flop. This output will be taken as a positive edge clock by the second flip flop. This input will not change the second flip flop's output state because it is the negative edge triggered flip flop.

So, $Q_A = 1$ and $Q_B = 0$

3. **Condition 3:** When the second negative clock edge is applied.

Operation: The first flip flop will toggle again, and the output of this flip flop will change from 1 to 0. This output will be taken as a negative edge clock by the second flip flop. This input will change the second flip flop's output state because it is the negative edge triggered flip flop.

So, $Q_A = 0$ and $Q_B = 1$.

4. **Condition 4:** When the third negative clock edge is applied.

Operation: The first flip flop will toggle again, and the output of this flip flop will change from 0 to 1. This output will be taken as a positive edge clock by the second flip flop. This input will not change the second flip flop's output state because it is the negative edge triggered flip flop.

So, $Q_A = 1$ and $Q_B = 1$

5. **Condition 5:** When the fourth negative clock edge is applied.

Operation: The first flip flop will toggle again, and the output of this flip flop will change from 1 to 0. This output will be taken as a negative edge clock by the second flip flop. This input will change the output state of the



second flip flop.

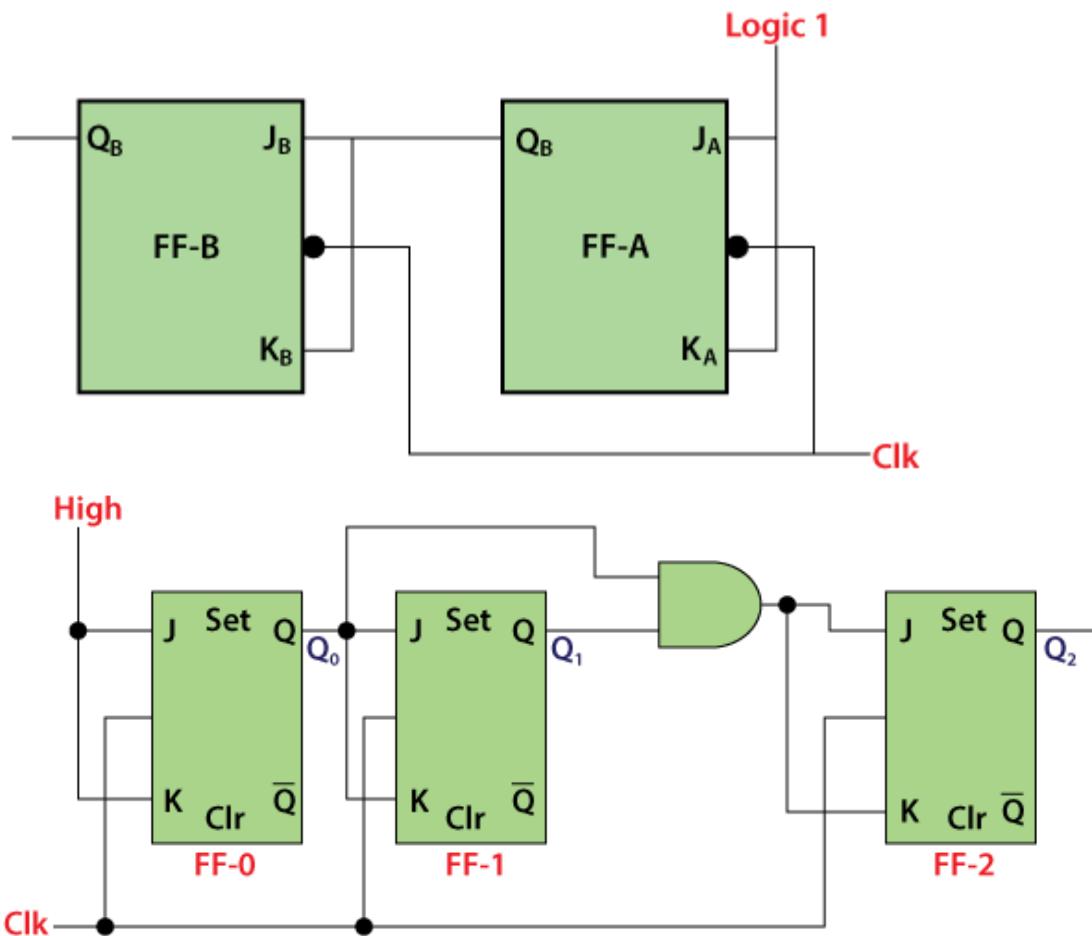
$$\text{So, } Q_A = 0 \text{ and } Q_B = 0$$

Synchronous counters

In the **Asynchronous counter**, the present counter's output passes to the input of the next counter. So, the counters are connected like a chain. The drawback of this system is that it creates the counting delay, and the propagation delay also occurs during the counting stage. The **synchronous counter** is designed to remove this drawback.

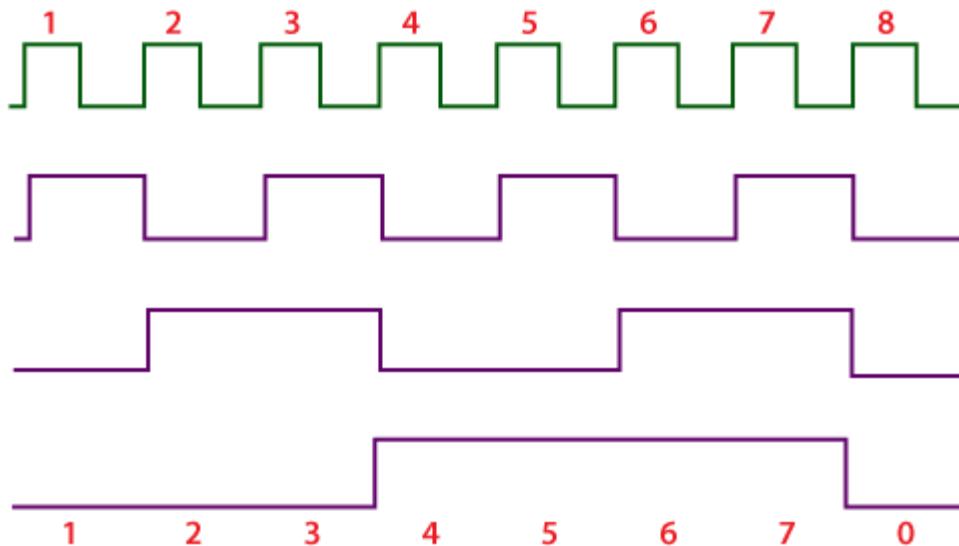
In the **synchronous counter**, the same clock pulse is passed to the clock input of all the flip flops. The clock signals produced by all the flip flops are the same as each other. Below is the diagram of a 2-bit synchronous counter in which the inputs of the first flip flop, i.e., FF-A, are set to 1. So, the first flip flop will work as a toggle flip-flop. The output of the first flip flop is passed to both the inputs of the next JK flip flop.

Logical Diagram





Signal Diagram



Operation

1. **Condition 1:** When both the flip flops are in reset condition.

Operation: The outputs of both flip flops, i.e., Q_A Q_B , will be 0.

So, $Q_A = 0$ and $Q_B = 0$

2. **Condition 2:** When the first negative clock edge passes.

Operation: The first flip flop will be toggled, and the output of this flip flop will be changed from 0 to 1. When the first negative clock edge is passed, the output of the first flip flop will be 0. The clock input of the first flip flop and both of its inputs will set to 0. In this way, the state of the second flip flop will remain the same.

So, $Q_A = 1$ and $Q_B = 0$

3. **Condition 2:** When the second negative clock edge is passed.

Operation: The first flip flop will be toggled again, and the output of this flip flop will be changed from 1 to 0. When the second negative clock edge is passed, the output of the first flip flop will be 1. The clock input of the first flip flop and both of its inputs will set to 1. In this way, the state of the second flip flop will change from 0 to 1.

So, $Q_A = 0$ and $Q_B = 1$

4. **Condition 2:** When the third negative clock edge passes.

Operation: The first flip flop will toggle from 0 to 1, but at this instance, both the inputs and the clock input set to 0. Hence, the outputs will remain the same as before.

So, $Q_A = 1$ and $Q_B = 1$

5. **Condition 2:** When the fourth negative clock edge passes.

Operation: The first flip flop will toggle from 1 to 0. At this instance, the inputs and the clock input of the second flip flop set to 1. Hence, the



outputs will change from 1 to 0.

So, $Q_A = 0$ and $Q_B = 0$

Classification of counters

Depending on the way in which the counting progresses, the synchronous or asynchronous counters are classified as follows –

- Up counters
- Down counters
- Up/Down counters

UP/DOWN Counter

Up counter and down counter is combined together to obtain an UP/DOWN counter. A mode control (M) input is also provided to select either up or down mode. A combinational circuit is required to be designed and used between each pair of flip-flop in order to achieve the up/down operation.

- Type of up/down counters
- UP/DOWN ripple counters
- UP/DOWN synchronous counter

UP/DOWN Ripple Counters

In the UP/DOWN ripple counter all the FFs operate in the toggle mode. So either T flip-flops or JK flip-flops are to be used. The LSB flip-flop receives clock directly. But the clock to every other FF is obtained from ($Q = Q \bar{b}$) output of the previous FF.

- **UP counting mode (M=0)** – The Q output of the preceding FF is connected to the clock of the next stage if up counting is to be achieved. For this mode, the mode select input M is at logic 0 (M=0).
- **DOWN counting mode (M=1)** – If M = 1, then the Q bar output of the preceding FF is connected to the next FF. This will operate the counter in the counting mode.

Example

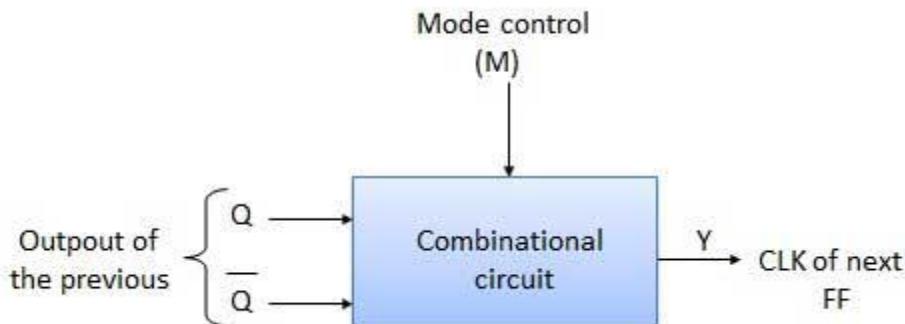
3-bit binary up/down ripple counter.

- 3-bit – hence three FFs are required.
- UP/DOWN – So a mode control input is essential.
- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.
- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.
- For a ripple down counter, the Q bar output of preceding FF is connected to the clock input of the next one.



- Let the selection of Q and Q bar output of the preceding FF be controlled by the mode control input M such that, If M = 0, UP counting. So connect Q to CLK. If M = 1, DOWN counting. So connect Q bar to CLK.

Block Diagram



Truth Table

Inputs			Outputs
M	Q	\bar{Q}	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
	0	1	1
	1	0	0
	1	1	1

} $Y = Q$
 for up
 counter
 } $Y = \bar{Q}$
 for up
 counter

Modulus Counter (MOD-N Counter)

The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So in general, an n-bit ripple counter is called as modulo-N counter. Where, MOD number = 2^n .

Type of modulus

- 2-bit up or down (MOD-4)
- 3-bit up or down (MOD-8)
- 4-bit up or down (MOD-16)

Application of counters

- Frequency counters
- Digital clock



- Time measurement
- A to D converter
- Frequency divider circuits
- Digital triangular wave generator.

Registers:

Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a **Register**. The **n-bit register** will consist of **n** number of flip-flop and it is capable of storing an **n-bit word**.

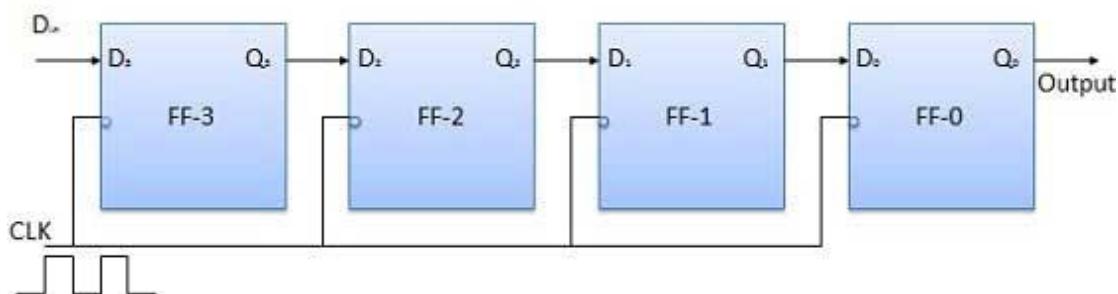
The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.

- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

Serial Input Serial Output

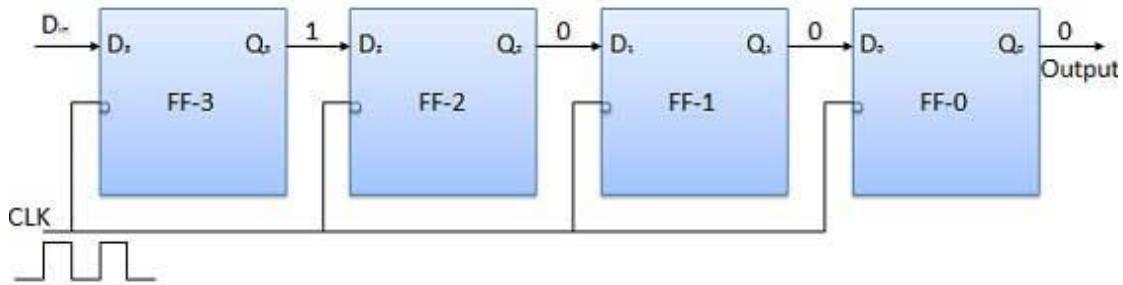
Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to D_{in} bit with the LSB bit applied first. The D input of FF-3 i.e. D_3 is connected to serial data input D_{in} . Output of FF-3 i.e. Q_3 is connected to the input of the next flip-flop i.e. D_2 and so on.

Block Diagram

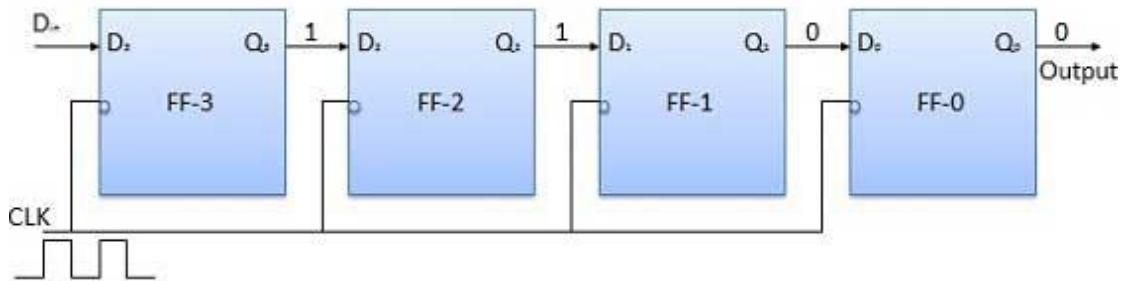


Operation

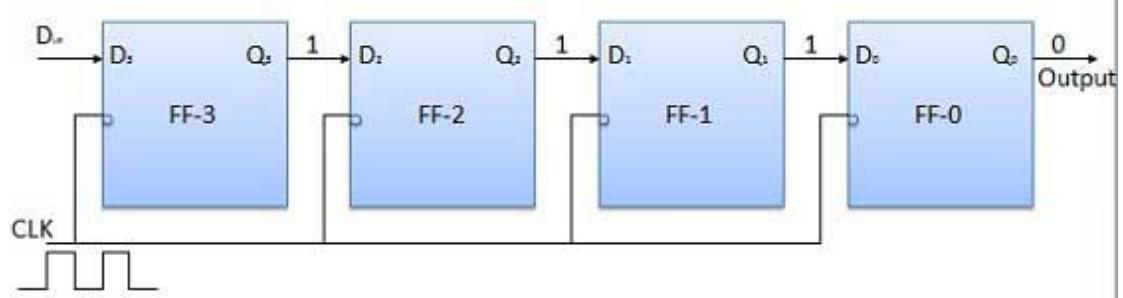
Before application of clock signal, let $Q_3 Q_2 Q_1 Q_0 = 0000$ and apply LSB bit of the number to be entered to D_{in} . So $D_{in} = D_3 = 1$. Apply the clock. On the first falling edge of clock, the FF-3 is set, and stored word in the register is $Q_3 Q_2 Q_1 Q_0 = 1000$.



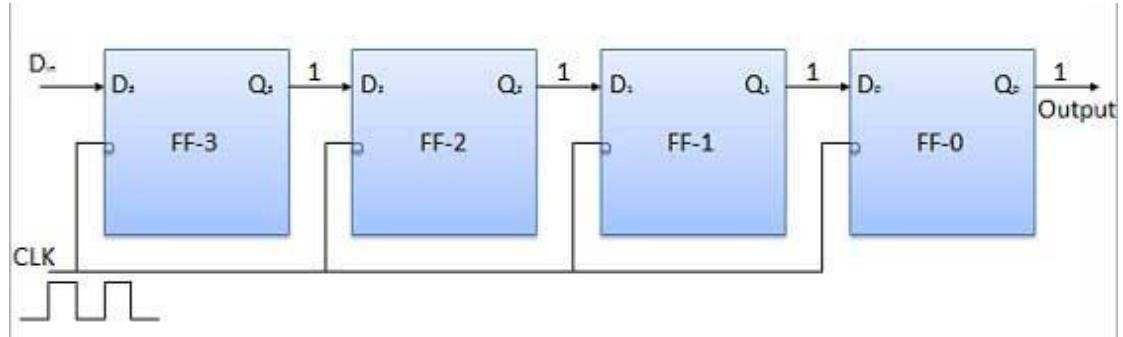
Apply the next bit to D_{in} . So $D_{in} = 1$. As soon as the next negative edge of the clock hits, FF-2 will set and the stored word change to $Q_3 Q_2 Q_1 Q_0 = 1100$.



Apply the next bit to be stored i.e. 1 to D_{in} . Apply the clock pulse. As soon as the third negative clock edge hits, FF-1 will be set and output will be modified to $Q_3 Q_2 Q_1 Q_0 = 1110$.



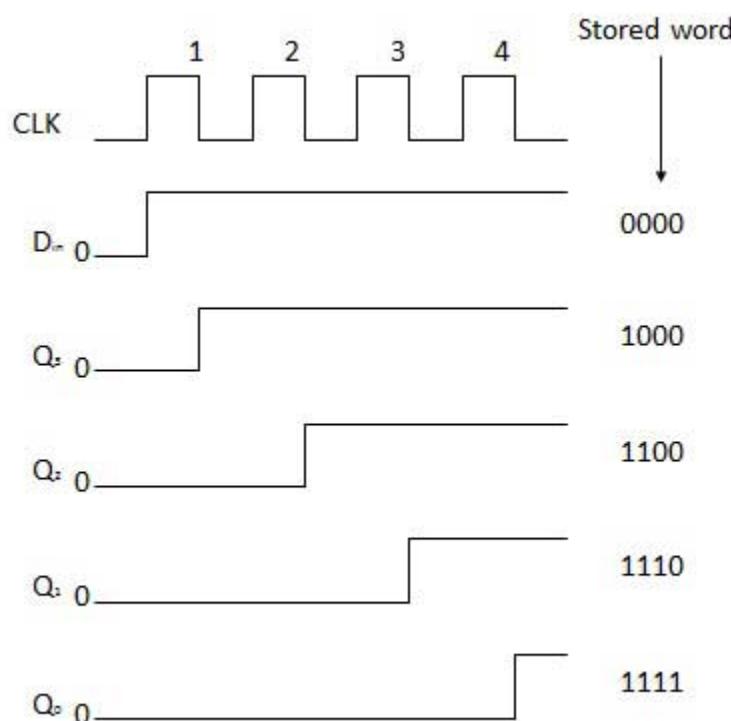
Similarly with $D_{in} = 1$ and with the fourth negative clock edge arriving, the stored word in the register is $Q_3 Q_2 Q_1 Q_0 = 1111$.



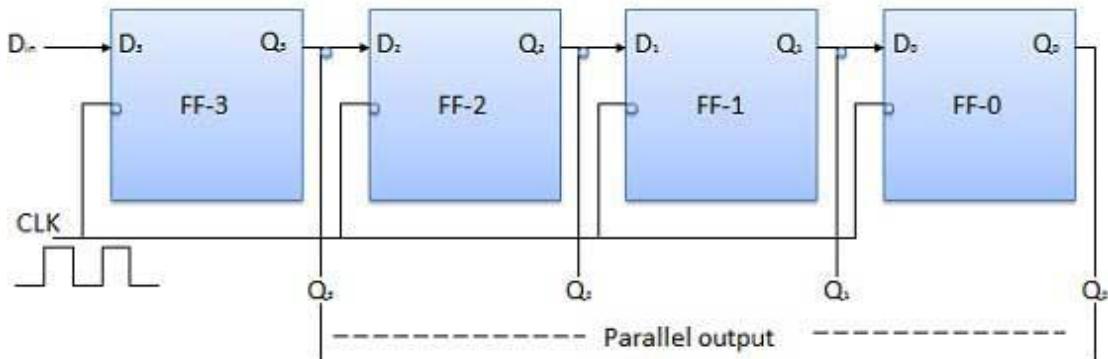
**Truth Table**

Initially	CLK	$D_n = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

→ Direction of data travel

Waveforms**Serial Input Parallel Output**

- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.

**Block Diagram****Parallel Input Serial Output (PISO)**

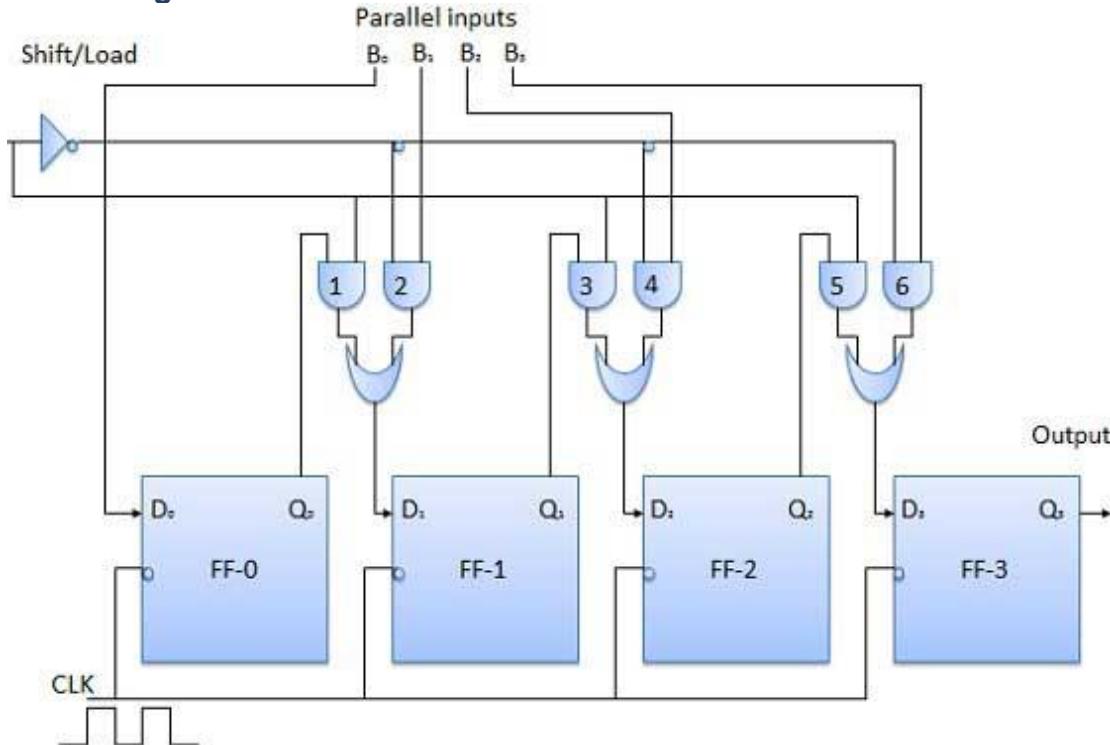
- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied through the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

Load mode

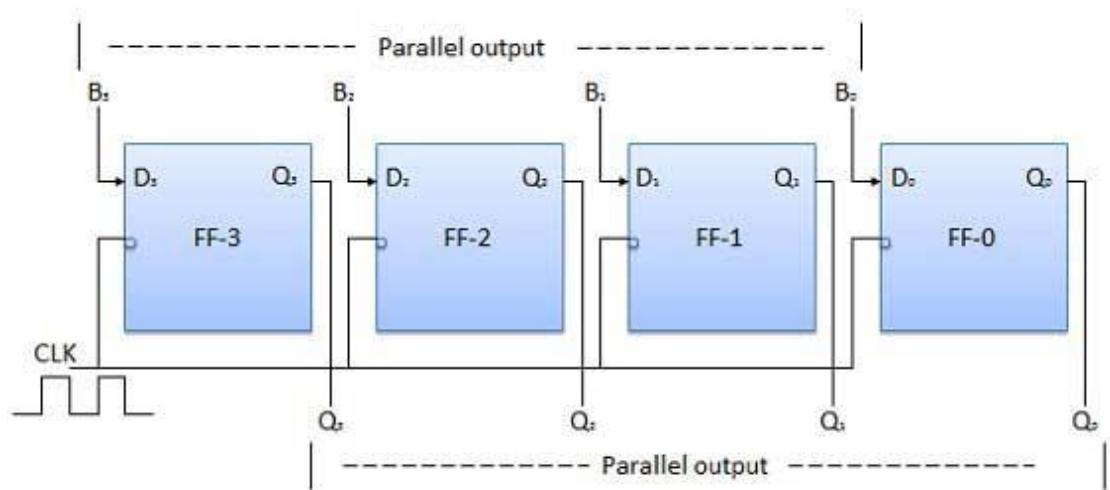
When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

When the shift/load bar line is high (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

**Block Diagram****Parallel Input Parallel Output (PIPO)**

In this mode, the 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.

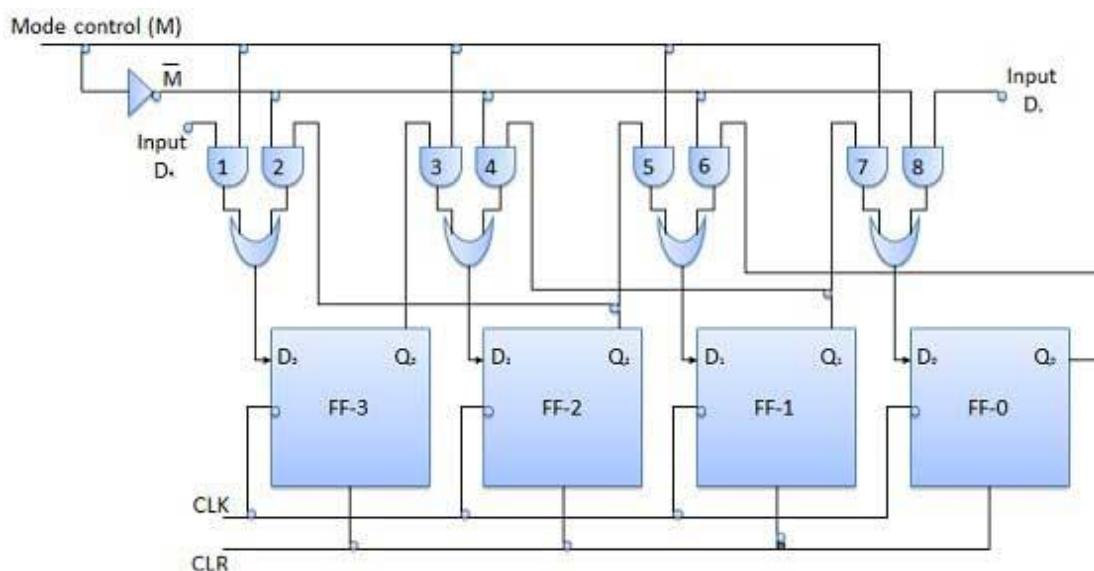
Block Diagram**Bidirectional Shift Register**

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is

shifted right by one position then it is equivalent to dividing the original number by 2.

- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
 - Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.
 - There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

Block Diagram



S.N.	Condition	Operation
1	With M = 1 – Shift right operation	<p>If M = 1, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled.</p> <p>The data at D_R is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with M = 1 we get the serial right shift operation.</p>
2	With M = 0 – Shift left operation	<p>When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled.</p>



The data at D_L is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with $M = 0$ we get the serial right shift operation.

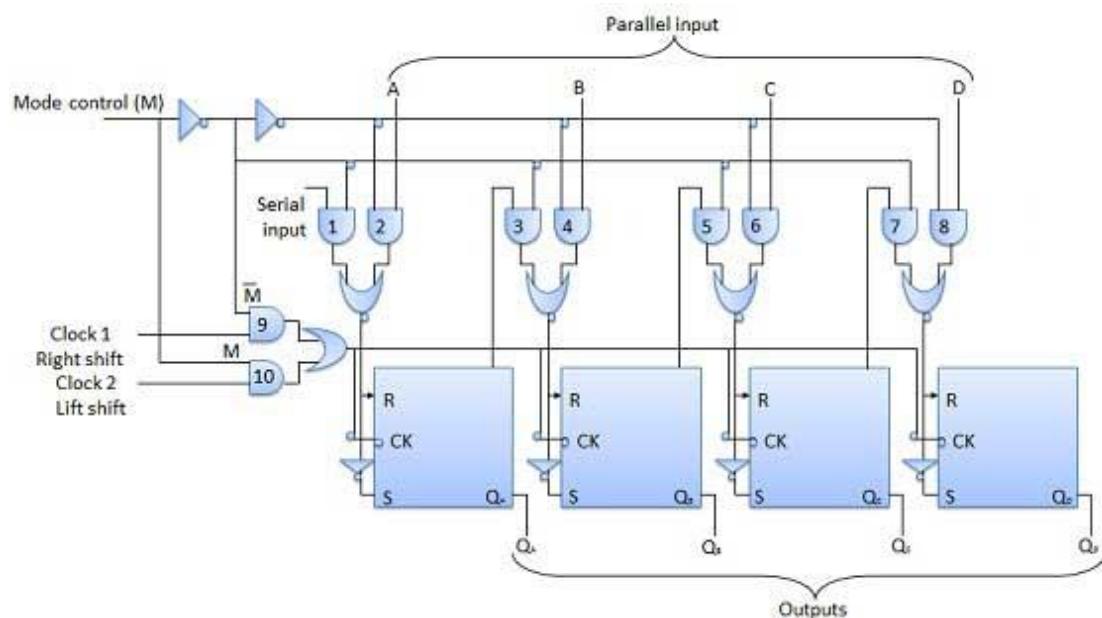
Universal Shift Register

A shift register which can shift the data in only one direction is called a uni-directional shift register. A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallelly, is known as a universal shift register. The shift register is capable of performing the following operation –

- Parallel loading
- Left Shifting
- Right shifting

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.

Block Diagram



Shift Register Counter –

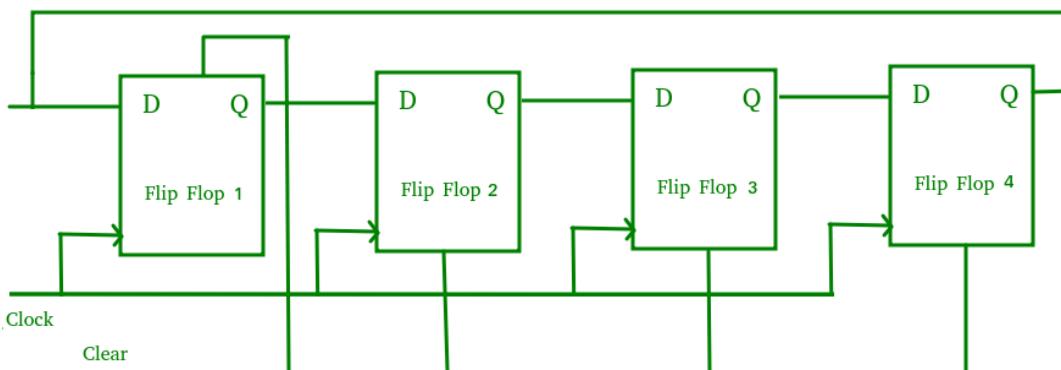
Shift Register Counters are the shift registers in which the outputs are connected back to the inputs in order to produce particular sequences. These are basically of two types:

**1. Ring Counter -**

A ring counter is basically a shift register counter in which the output of the first flip flop is connected to the next flip flop and so on and the output of the last flip flop is again fed back to the input of the first flip flop, thus the name ring counter. The data pattern within the shift register will circulate as long as clock pulses are applied.

The logic circuit given below shows a Ring Counter. The circuit consists of four D flip-flops which are connected. Since the circuit consists of four flip flops the data pattern will repeat after every four clock pulses as shown in the truth table below:

Clock Pulse	Q1	Q2	Q3	Q4
0	1	0	0	1
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1



A Ring counter is generally used because it is self-decoding. No extra decoding circuit is needed to determine what state the counter is in.

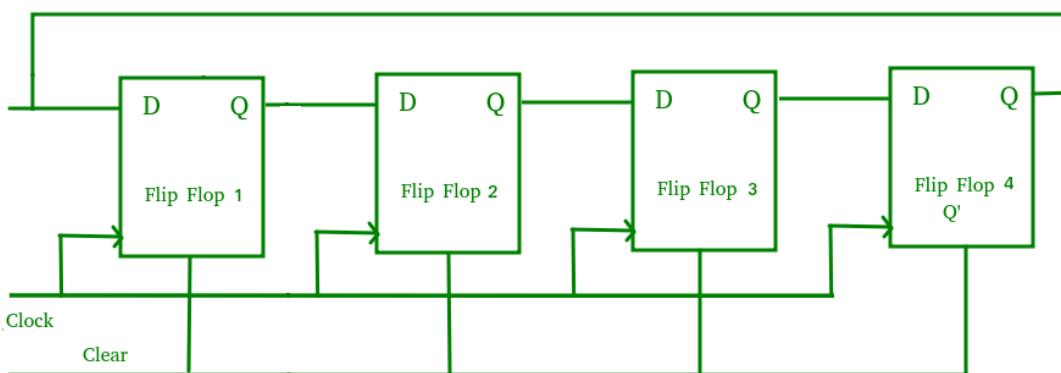
Johnson Counter -

A Johnson counter is basically a shift register counter in which the output of the first flip flop is connected to the next flip flop and so on and the inverted output of the last flip flop is again fed back to the input of the first flip flop. They are also known as twisted ring counters.

The logic circuit given below shows a Johnson Counter. The circuit consists of four D flip-flops which are connected. An n-stage Johnson counter yields a count sequence of $2n$ different states, thus also known as a mod- $2n$ counter. Since the circuit consists of four flip flops the data pattern will repeat every eight clock pulses as shown in the truth table below:



Clock Pulse	Q1	Q2	Q3	Q4
0	0	0	0	1
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1



The main advantage of Johnson counter is that it only needs n number of flip-flops compared to the ring counter to circulate a given data to generate a sequence of $2n$ states.

Applications of shift Registers –

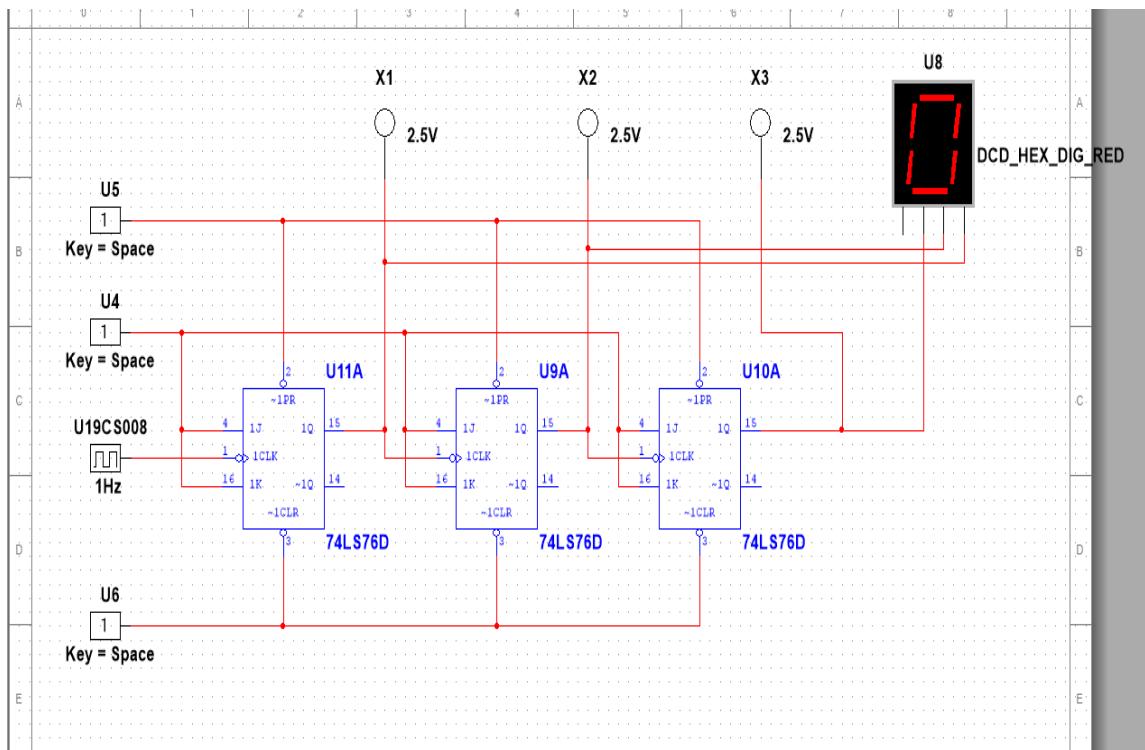
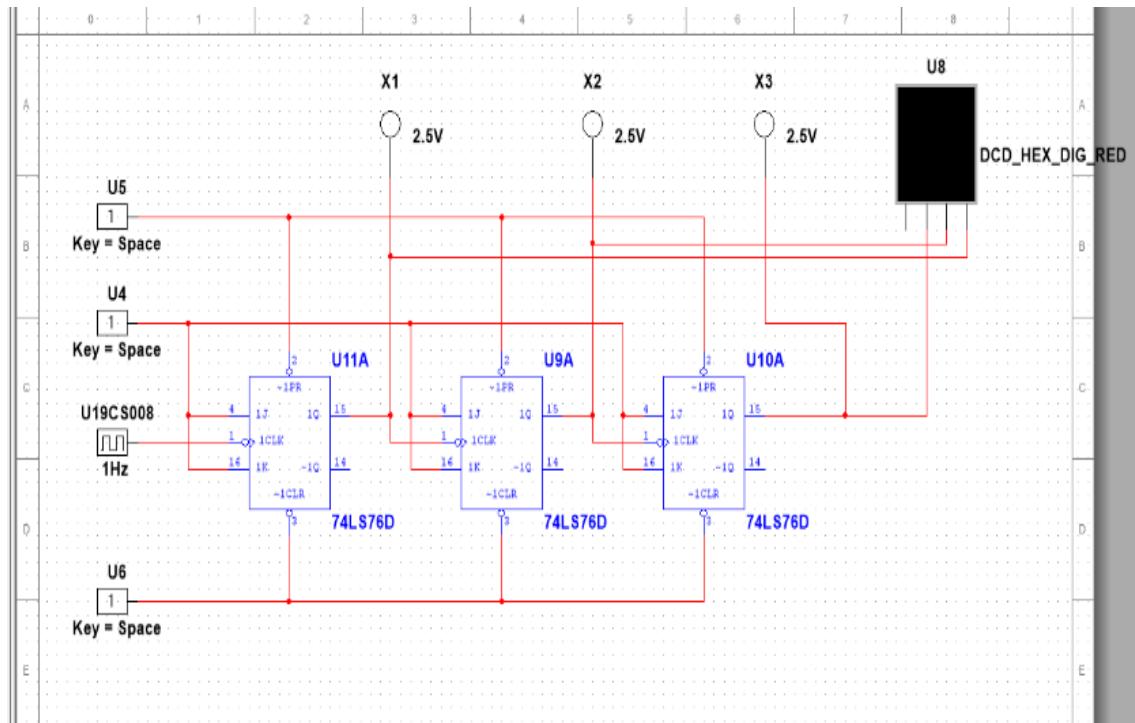
- The shift registers are used for temporary data storage.
- The shift registers are also used for data transfer and data manipulation.
- The serial-in serial-out and parallel-in parallel-out shift registers are used to produce time delay to digital circuits.
- The serial-in parallel-out shift register is used to convert serial data into parallel data thus they are used in communication lines where demultiplexing of a data line into several parallel line is required.

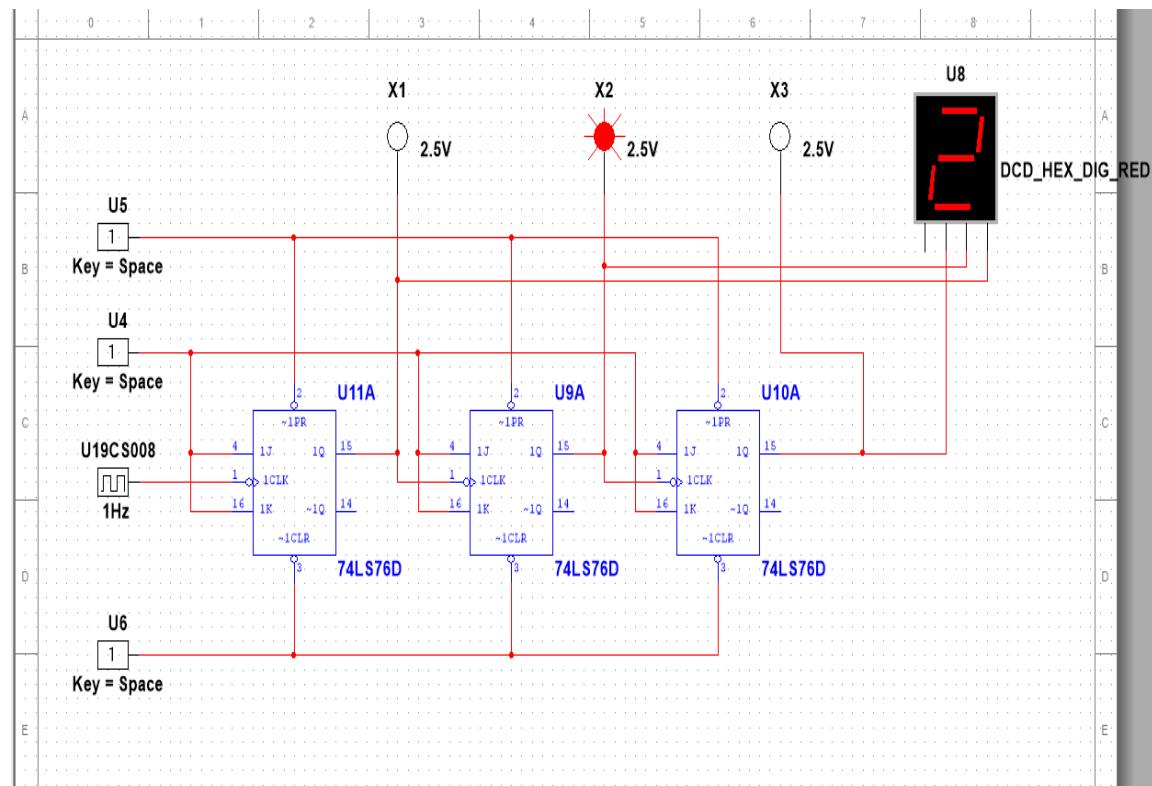
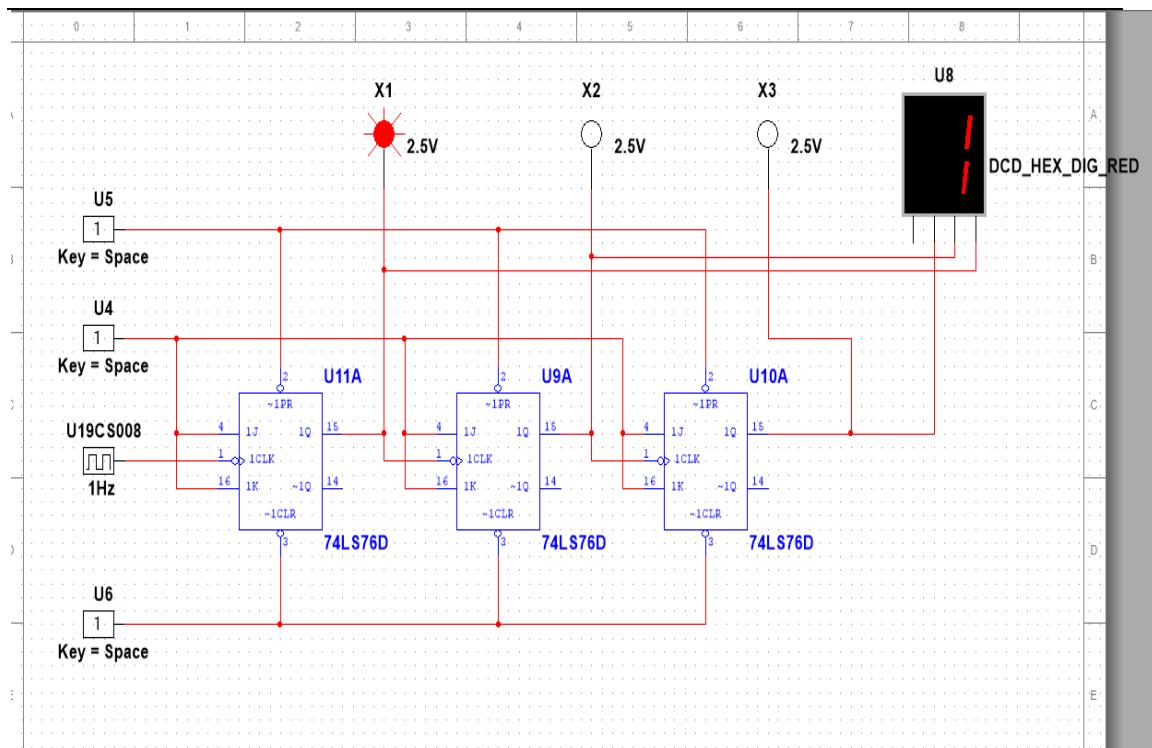


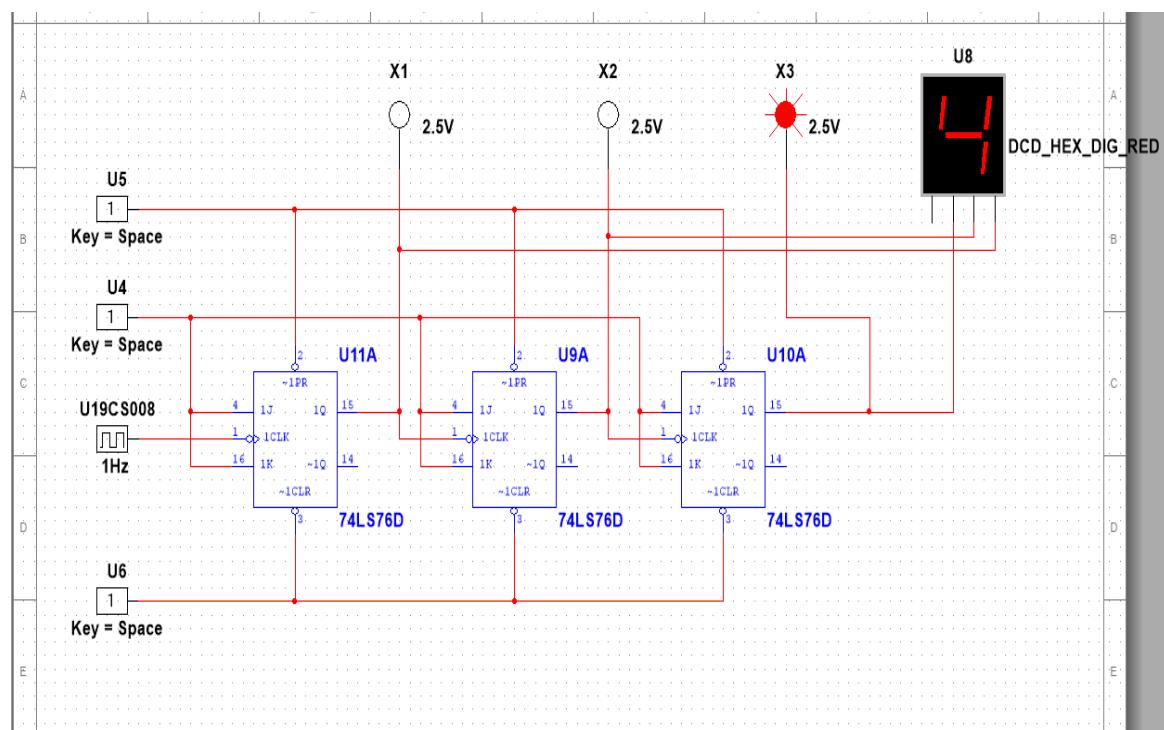
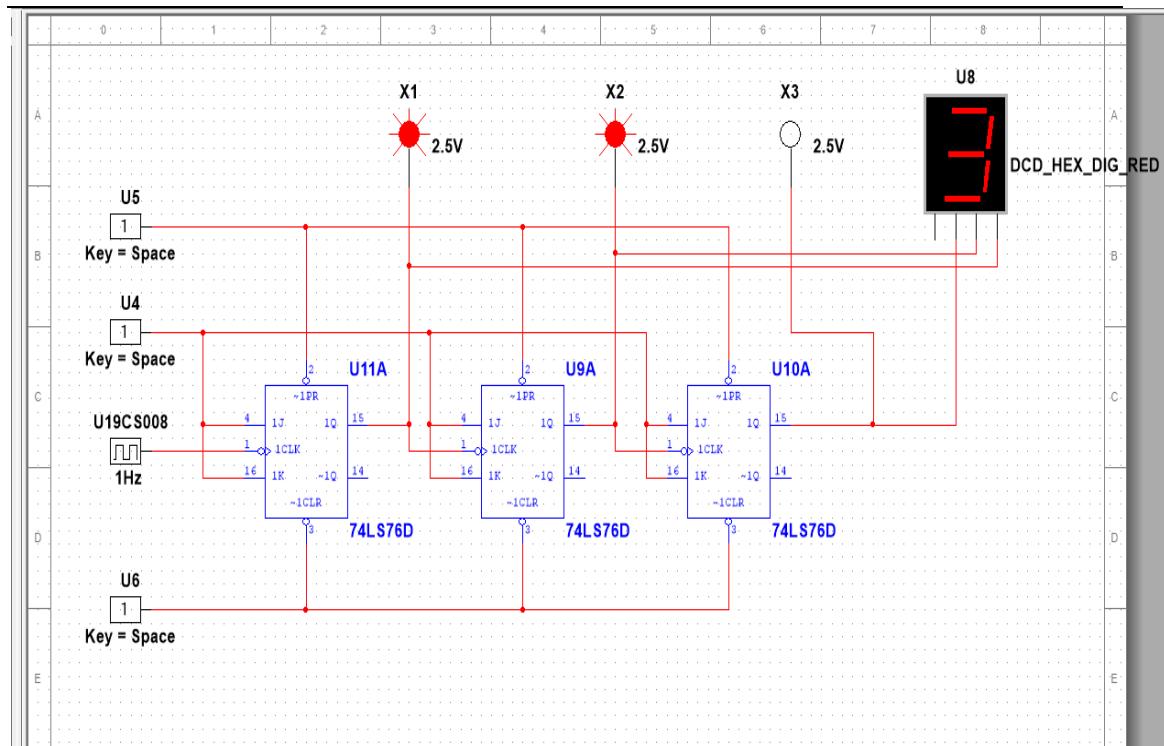
- A Parallel in Serial out shift register us used to convert parallel data to serial data.

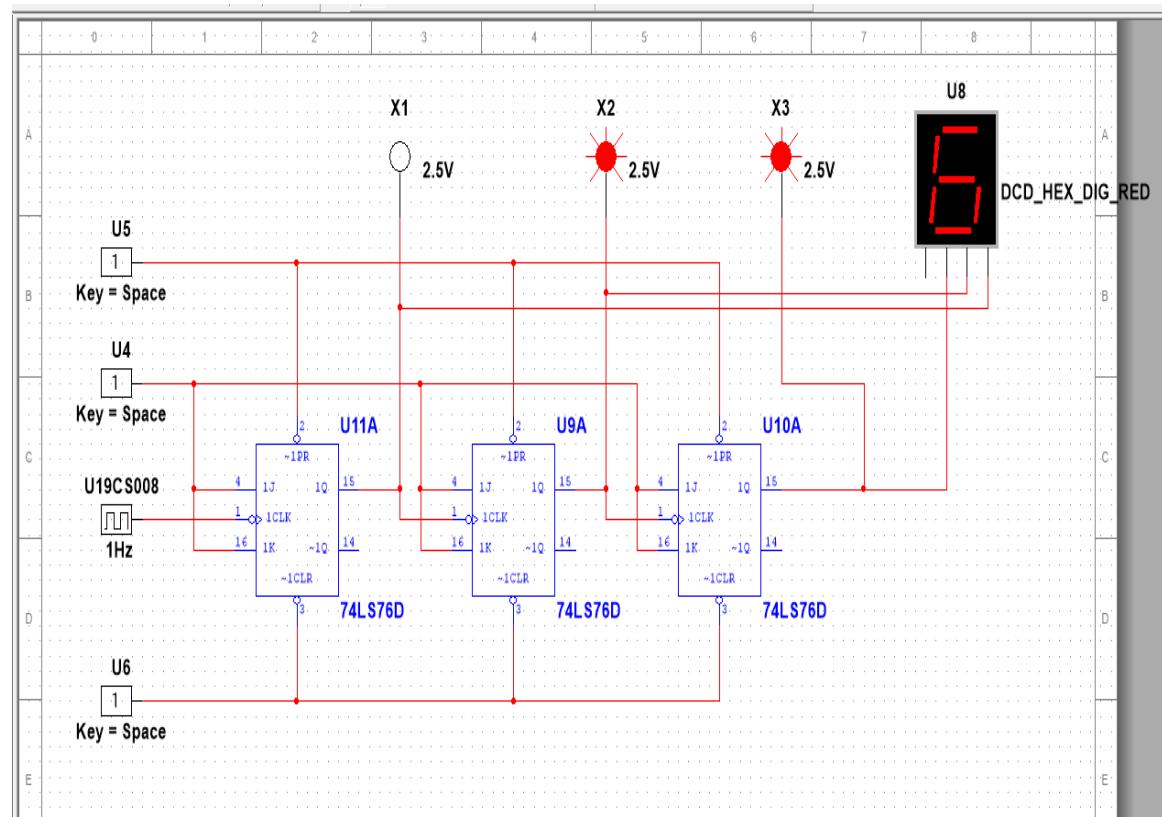
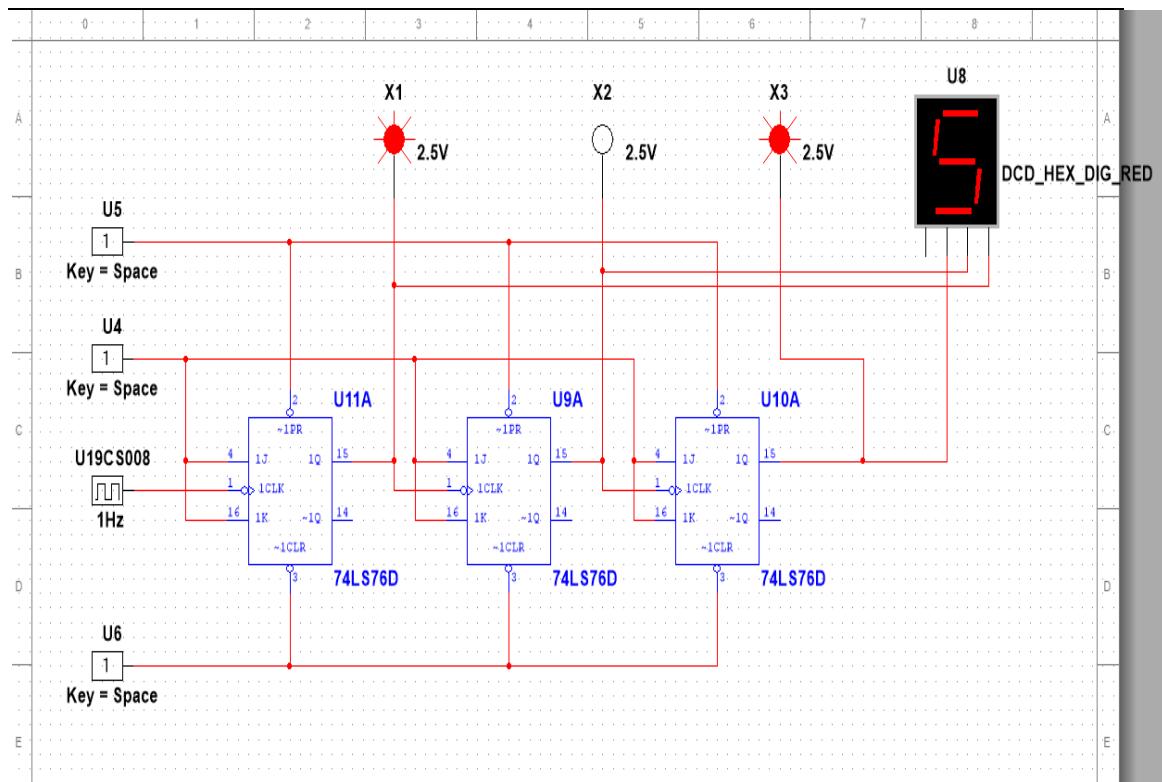
SIMULATION SCREENSHOTS

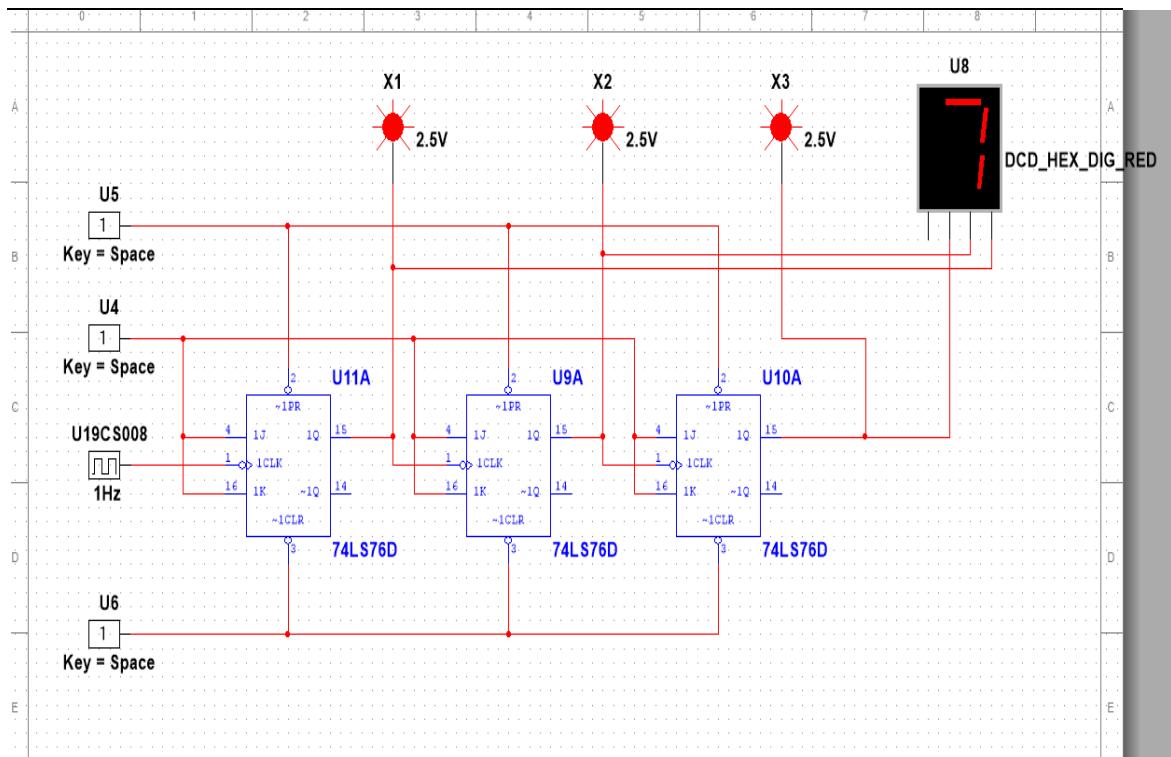
3-BIT UP COUNTER



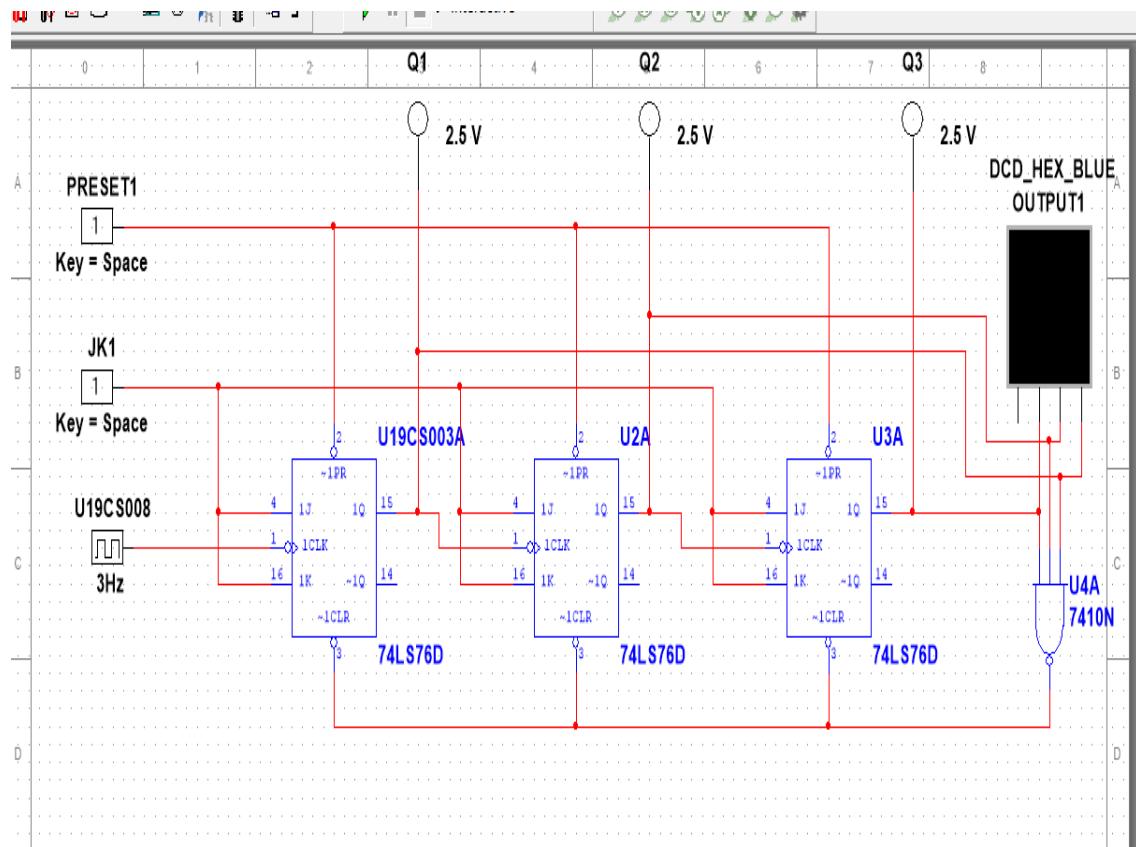


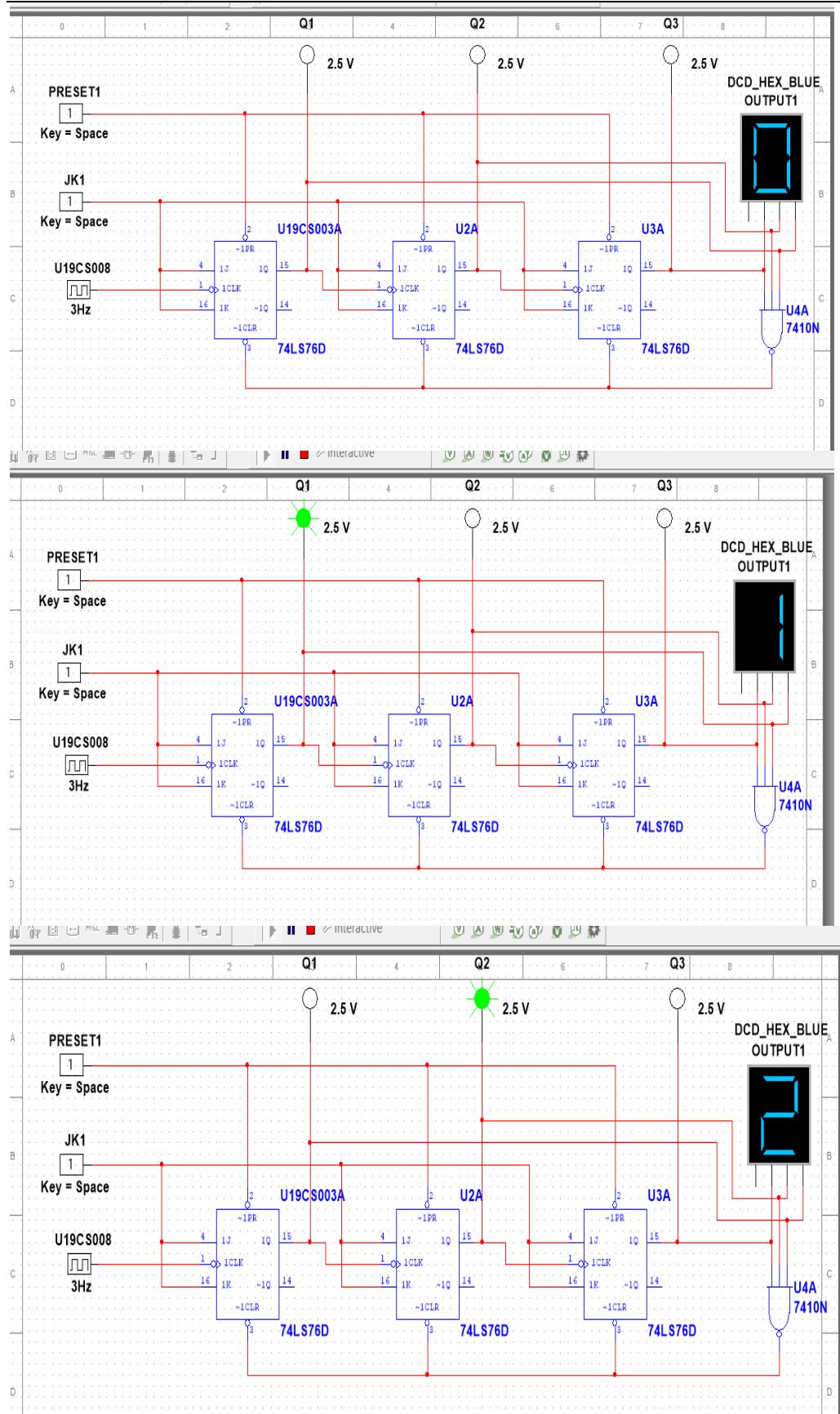


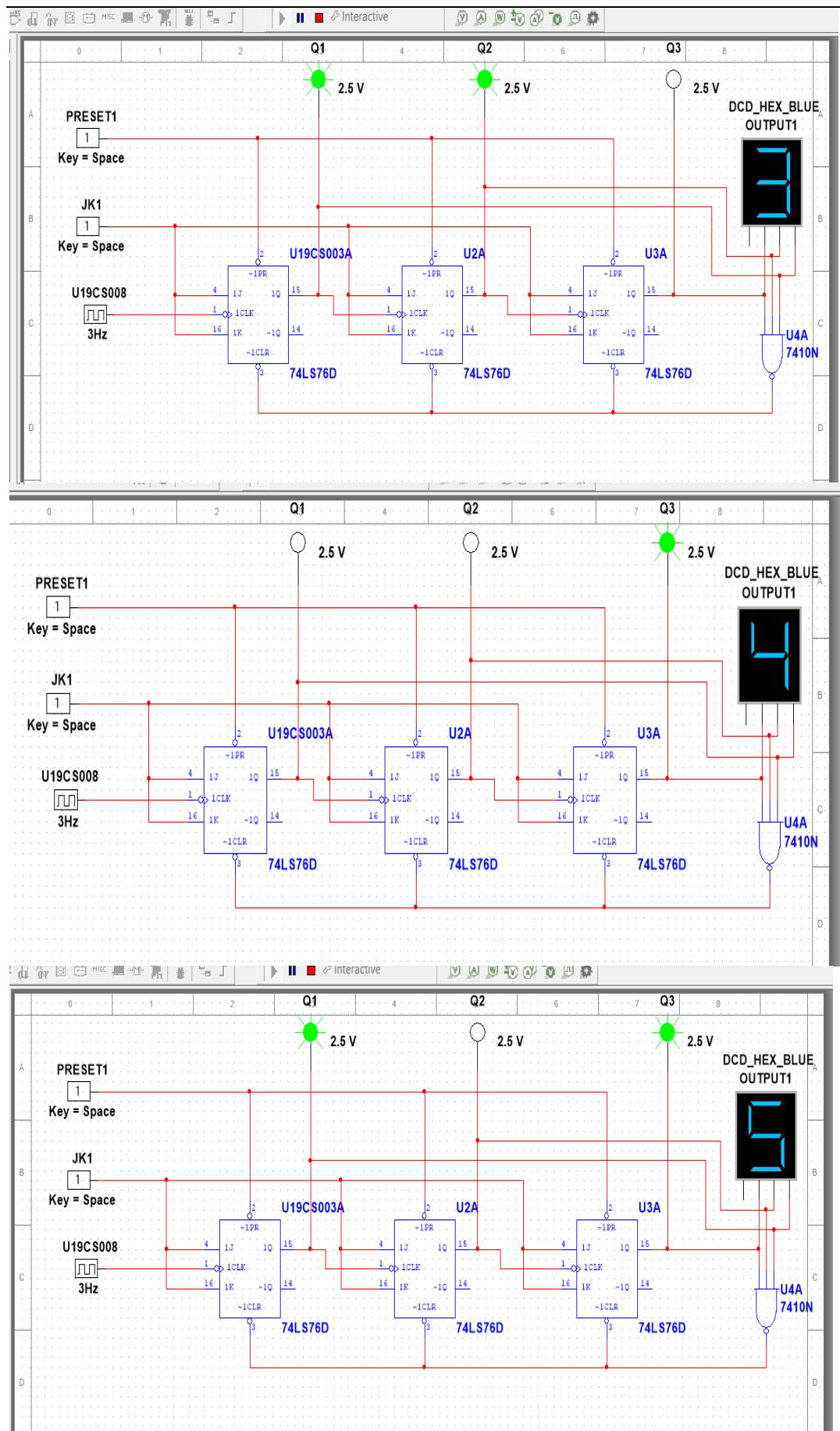


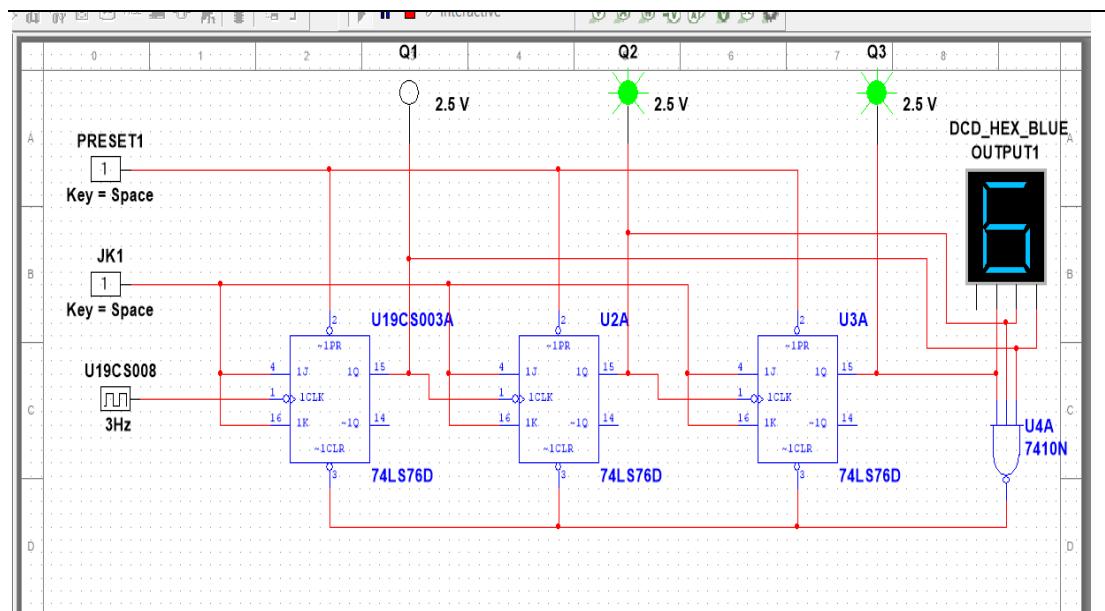


MOD-7 COUNTER

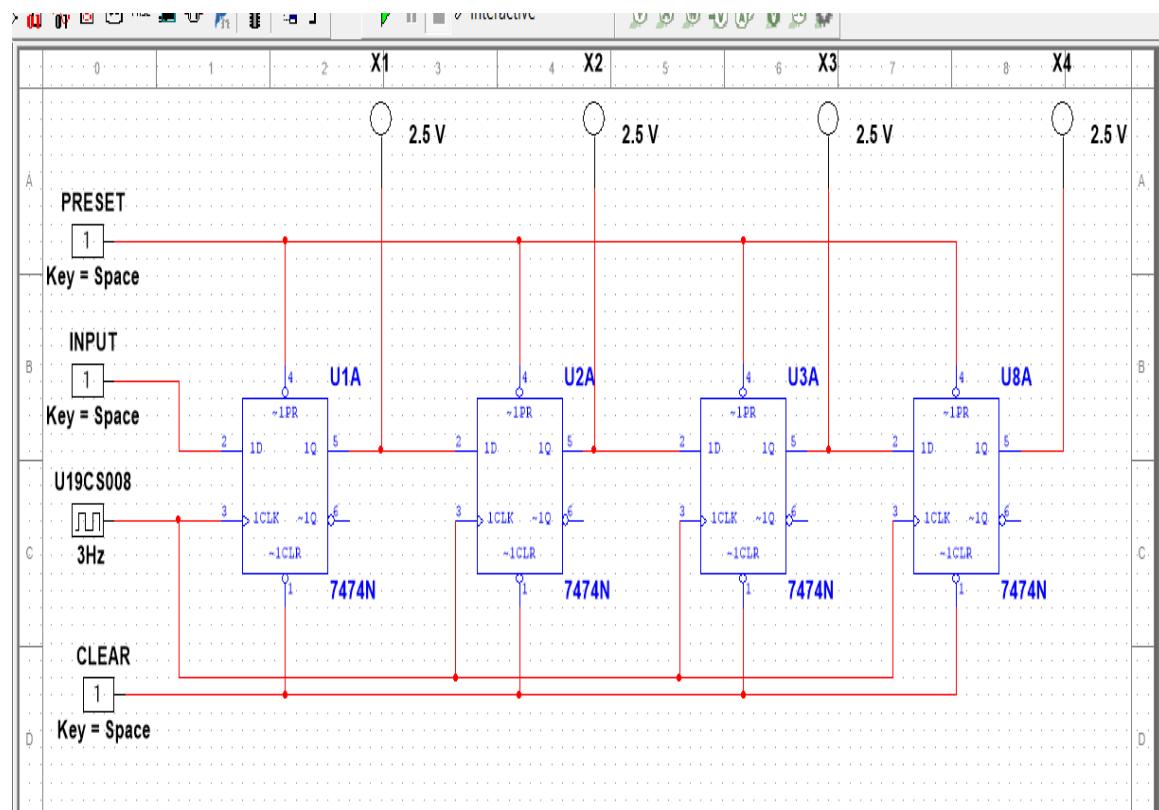


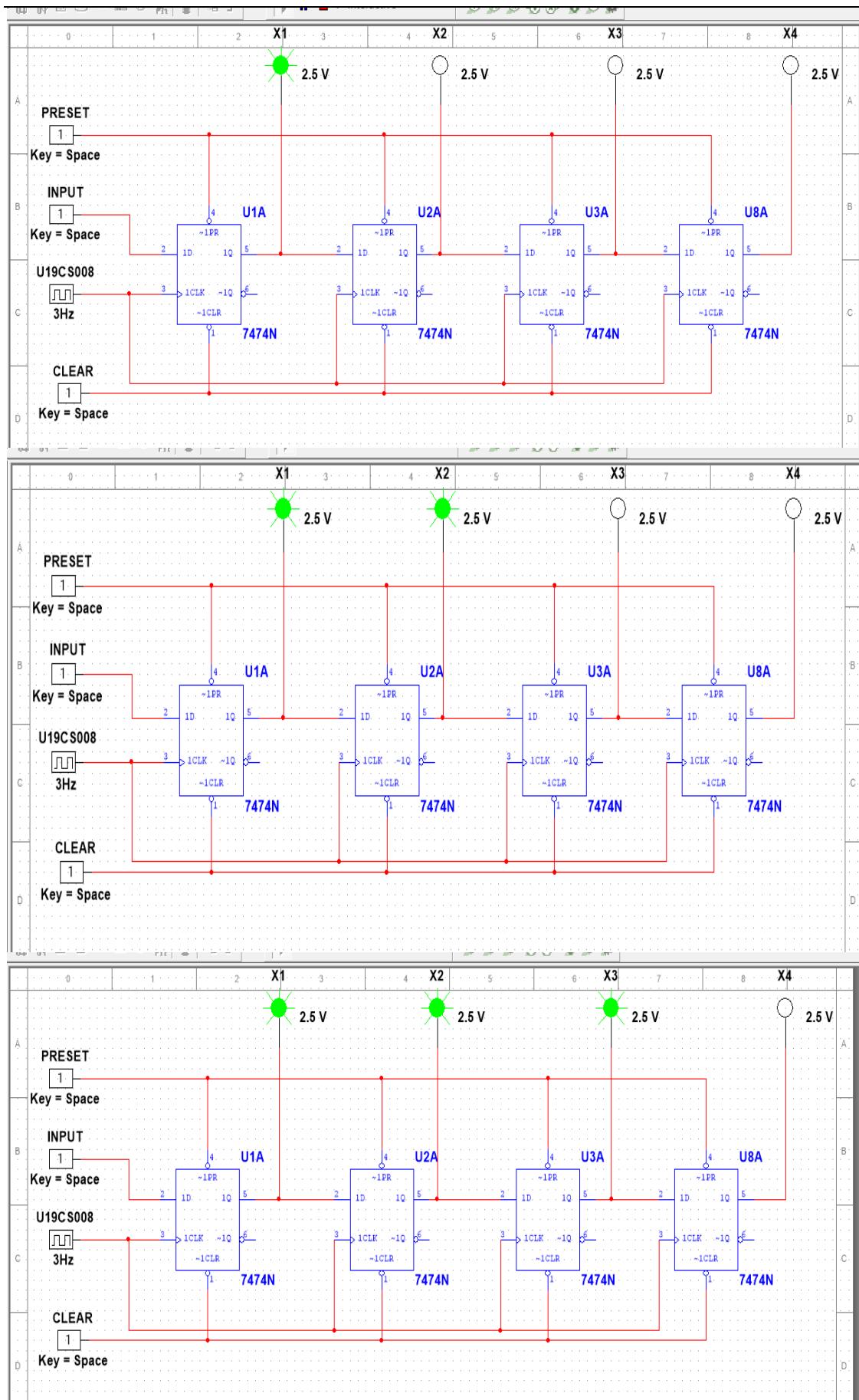


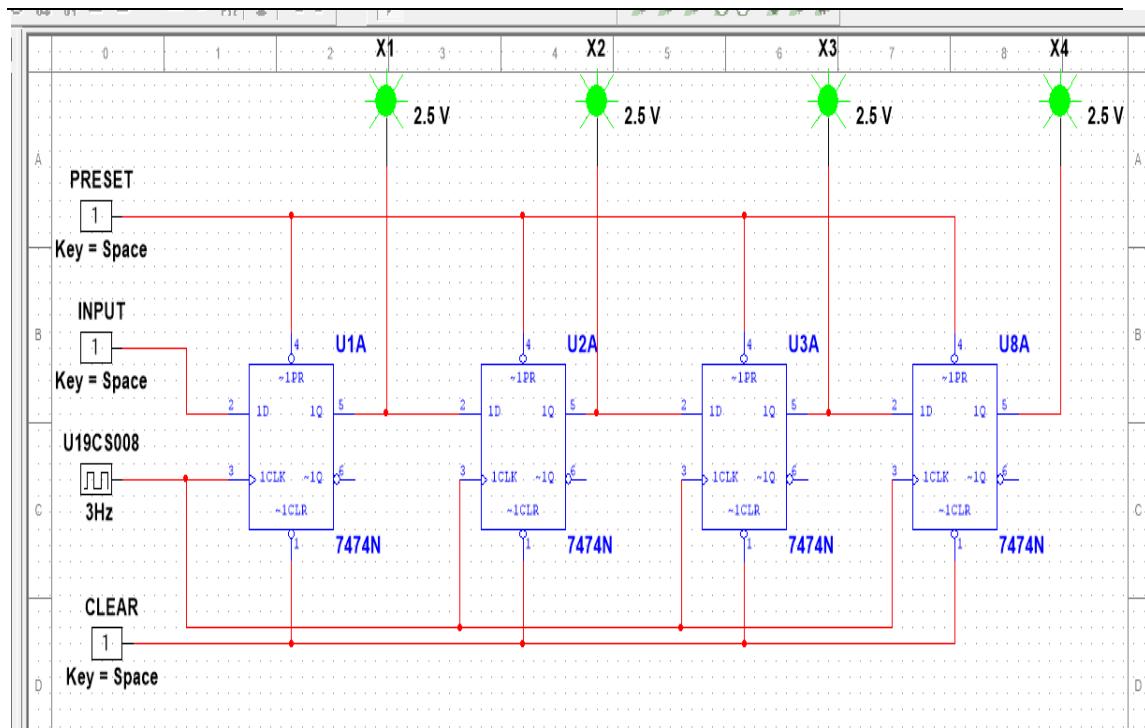




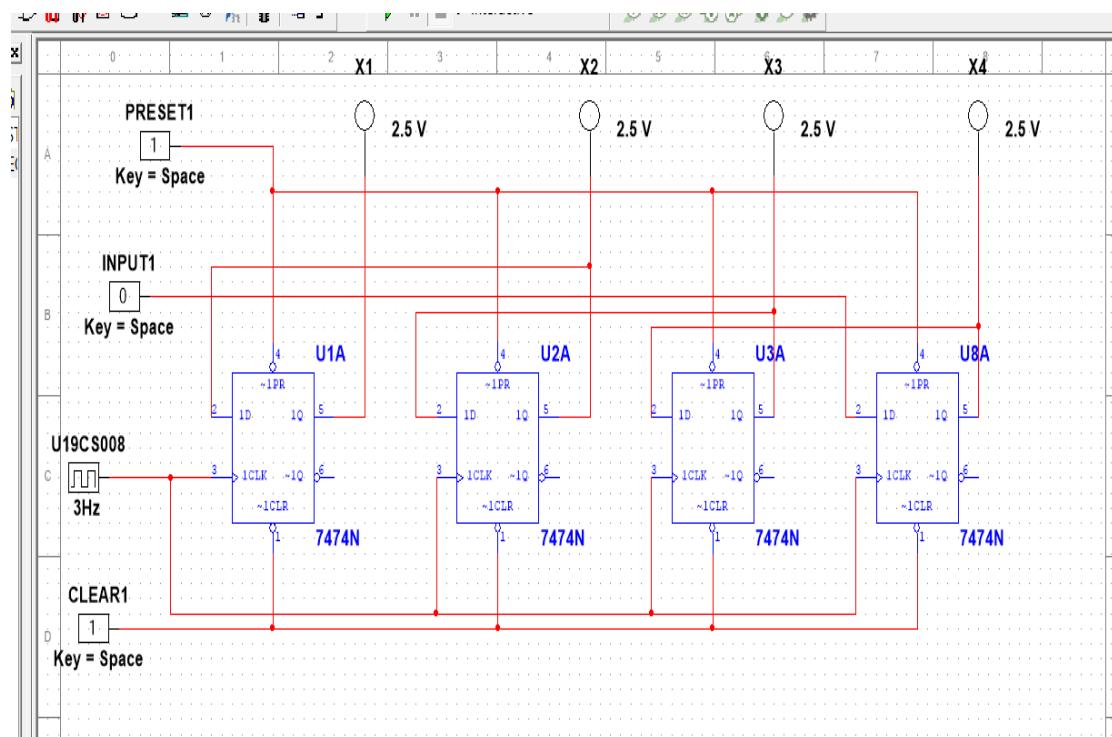
SHIFT RIGHT REGISTER

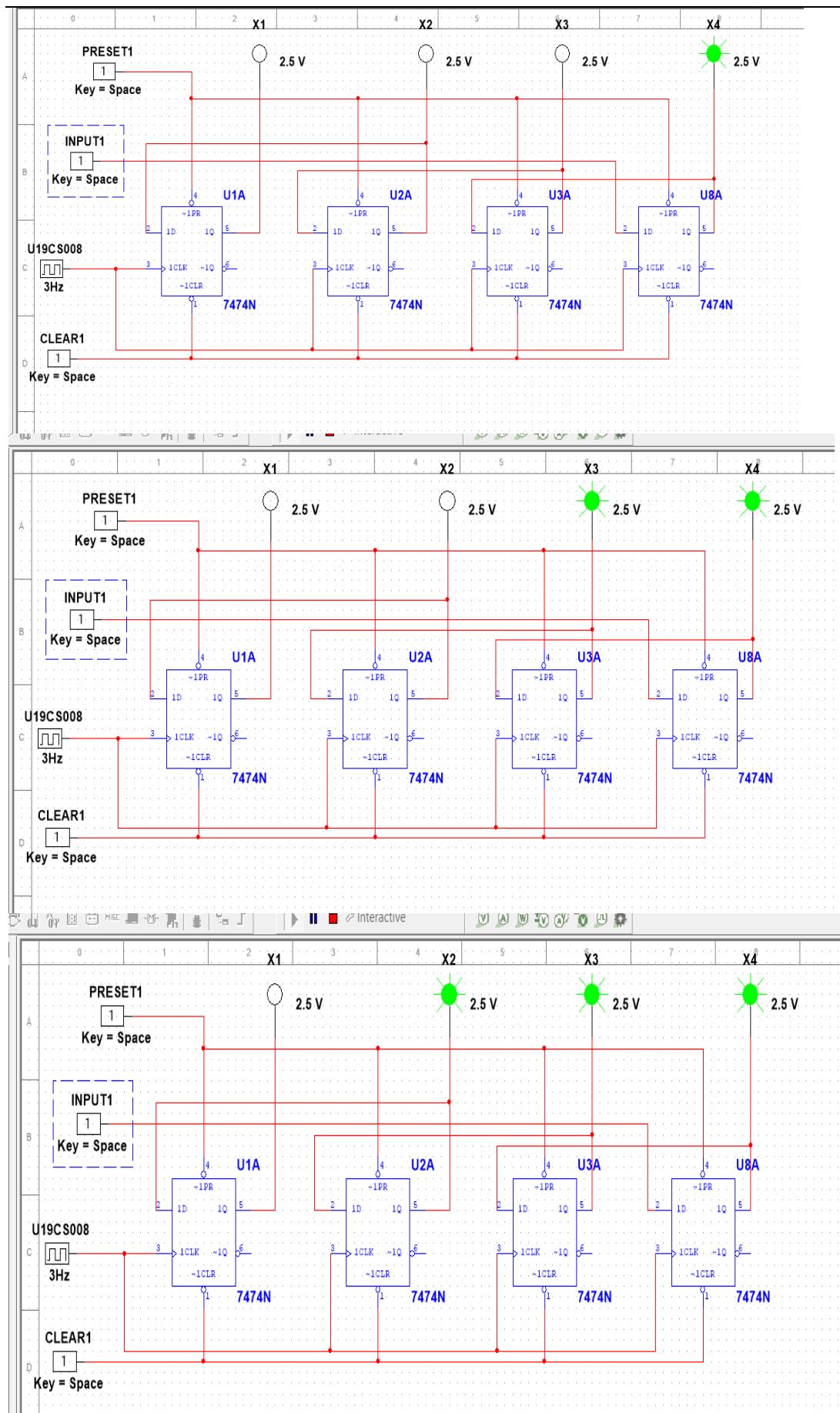


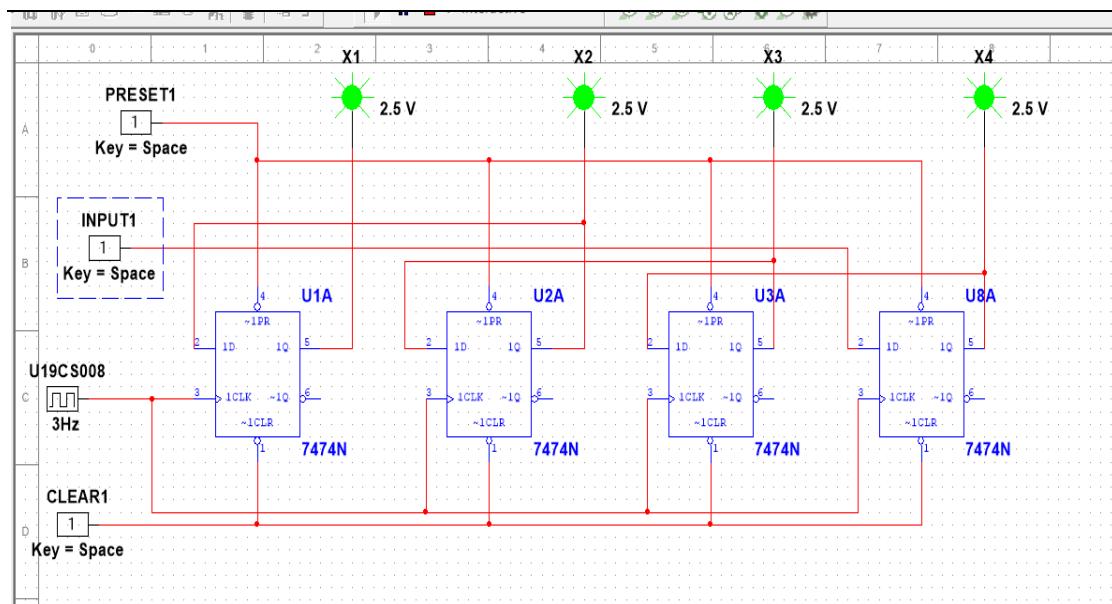




SHIFT LEFT REGISTER





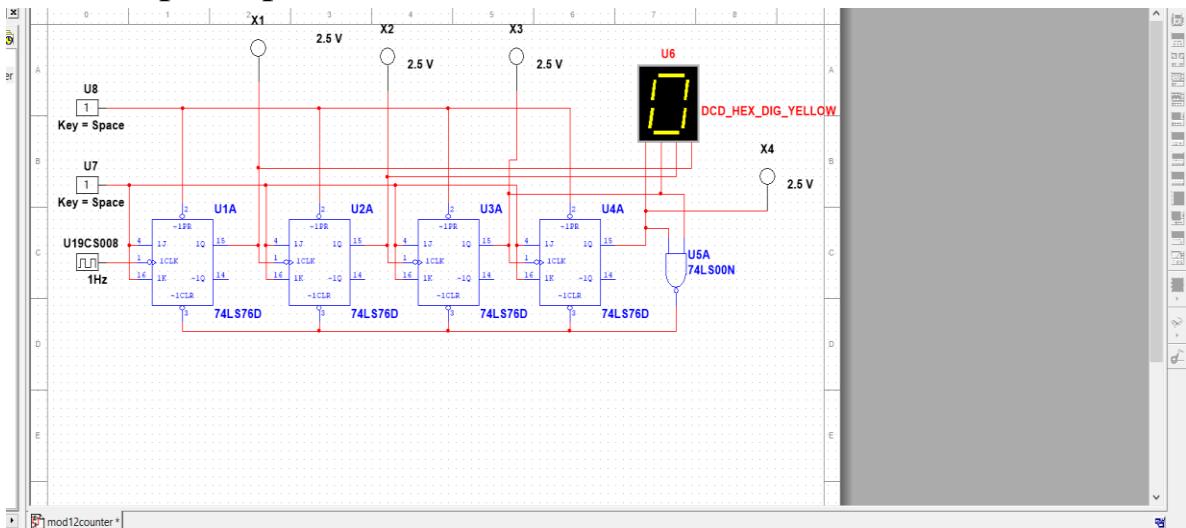


CONCLUSIONS

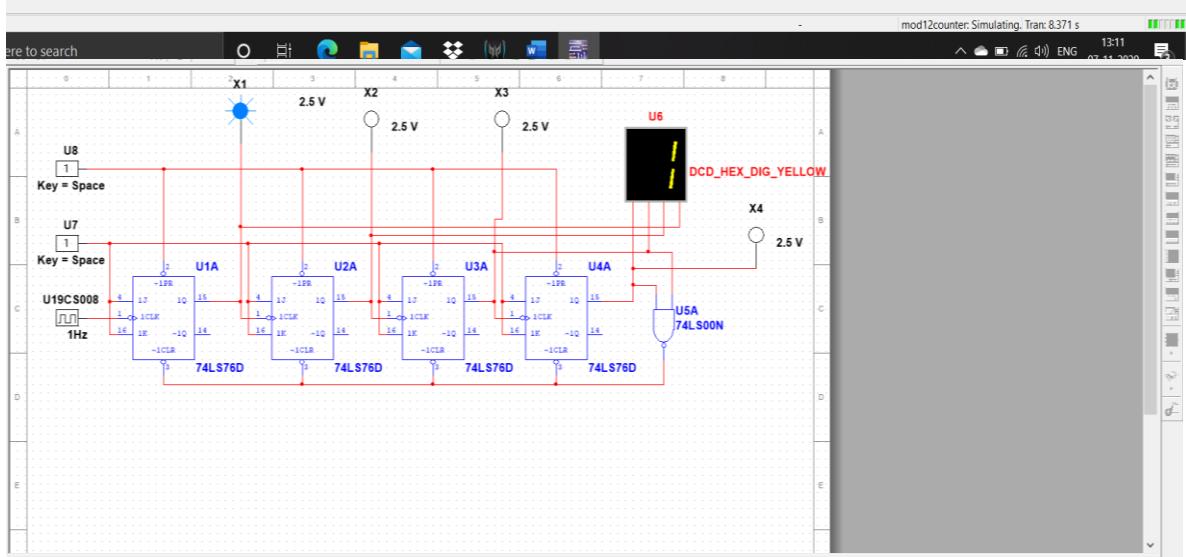
Successfully observed and implemented 3 – Bit Up Counter, Mod-7 Counter, 4 – Bit Shift Right Register, 4 – Bit Shift Left Register using Multisim software.

DLED ASSIGNMENT- 11NAME: KRINA PATELADMISSION NUMBER: U19CS008

1. Design and implement in Multisim Mod – 12 Counter using JK Flip- Flops.

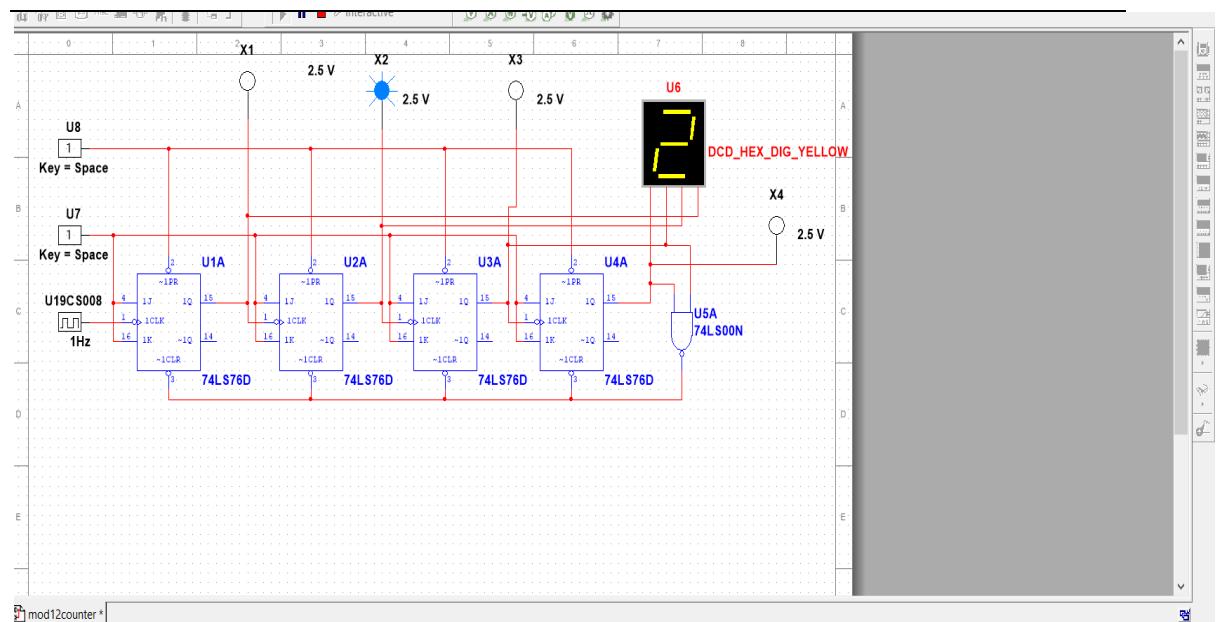


November 2020, 13:03:53
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.



November 2020, 13:03:53
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
 ↳ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.





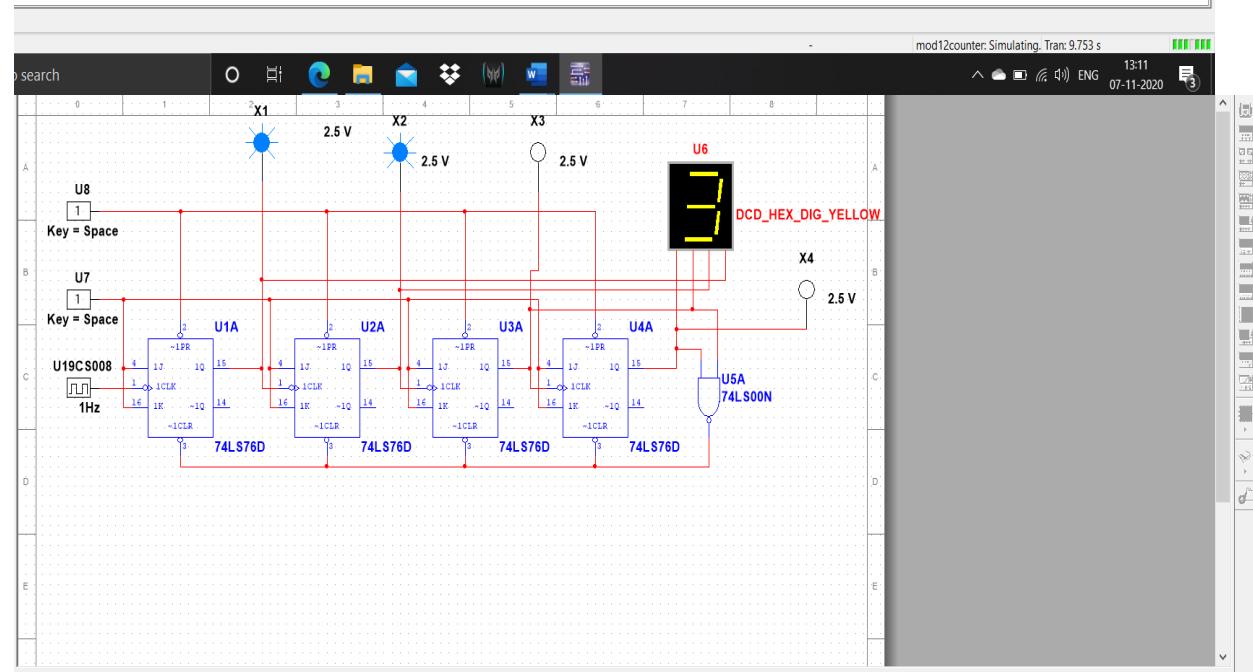
ber 2020, 13:03:51

The Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

The Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

The Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.



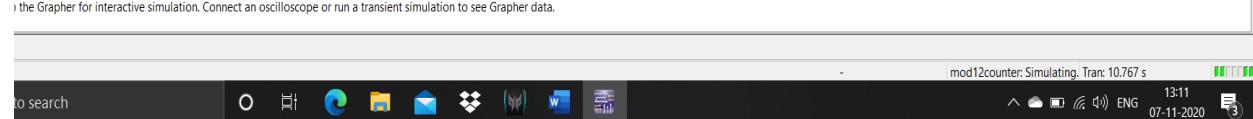
Module Counter

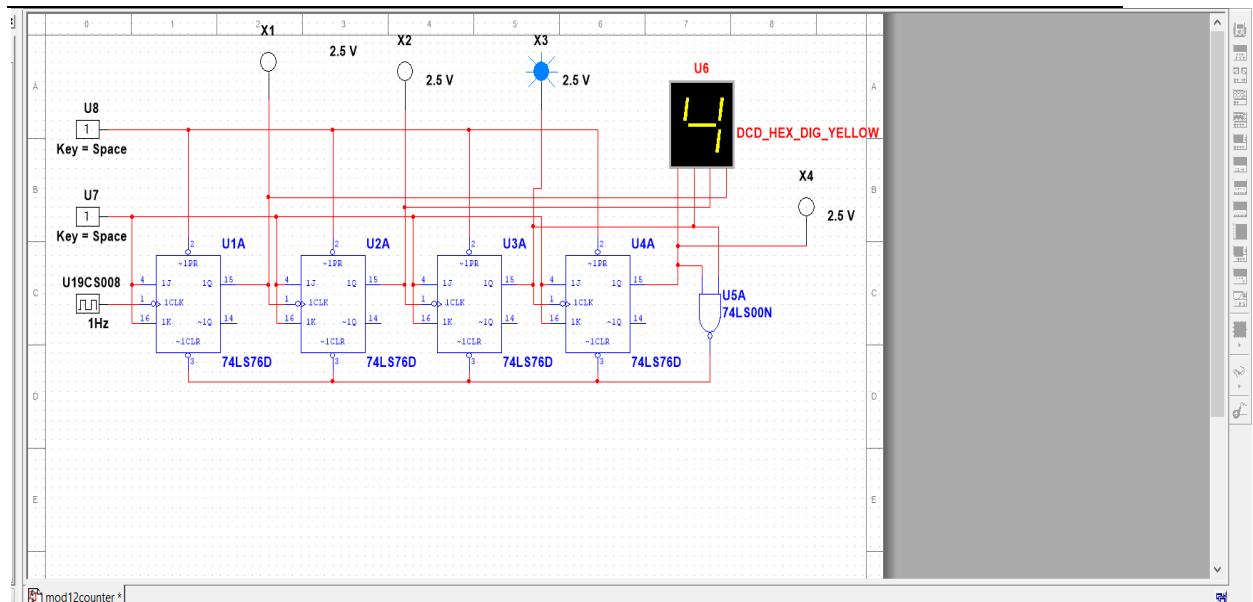
nber 2020, 13:03:53

→ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

› the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

→ the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.





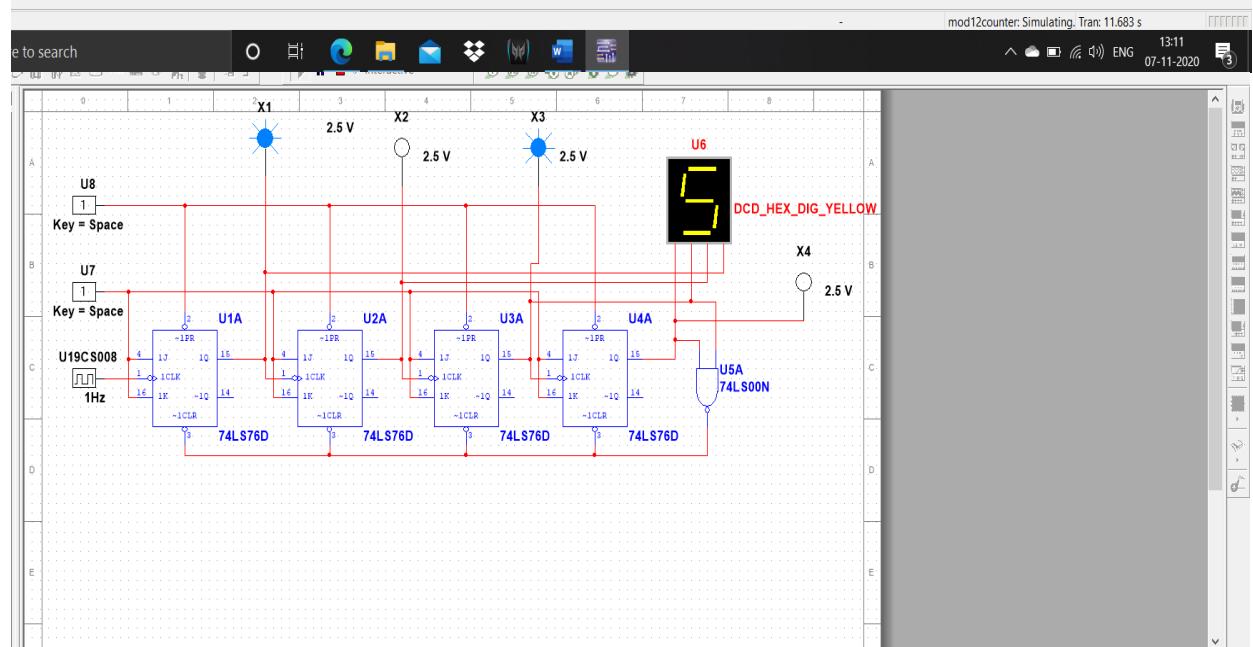
ember 2020, 13:03:53

t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.



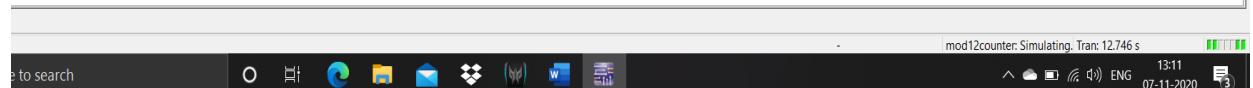
ember 2020, 13:03:53

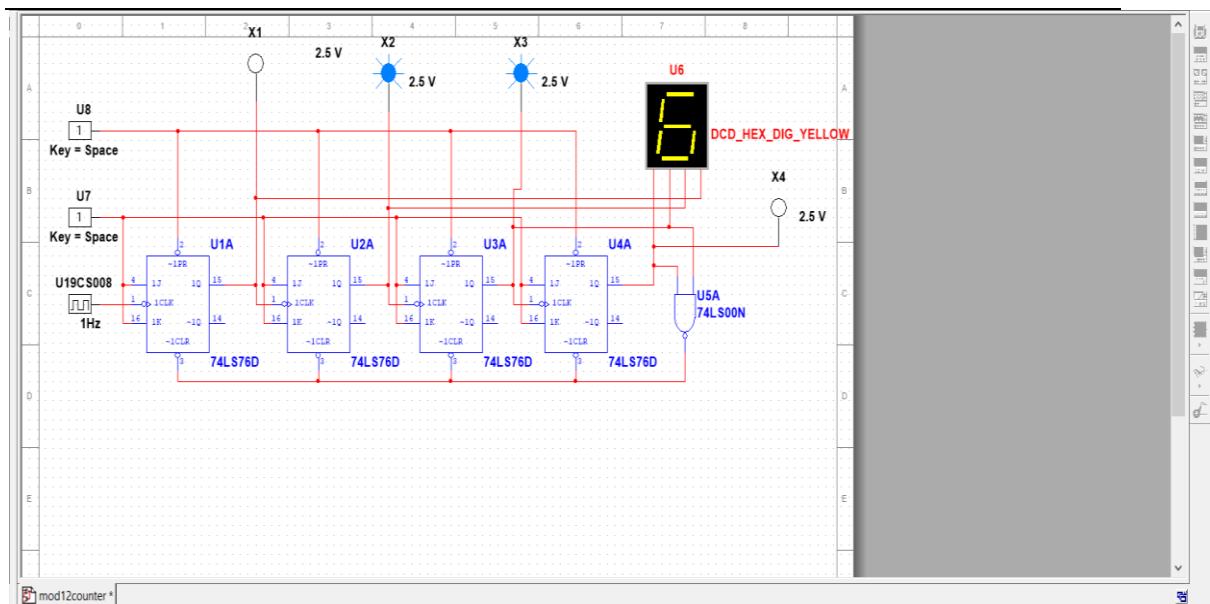
t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

t to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.





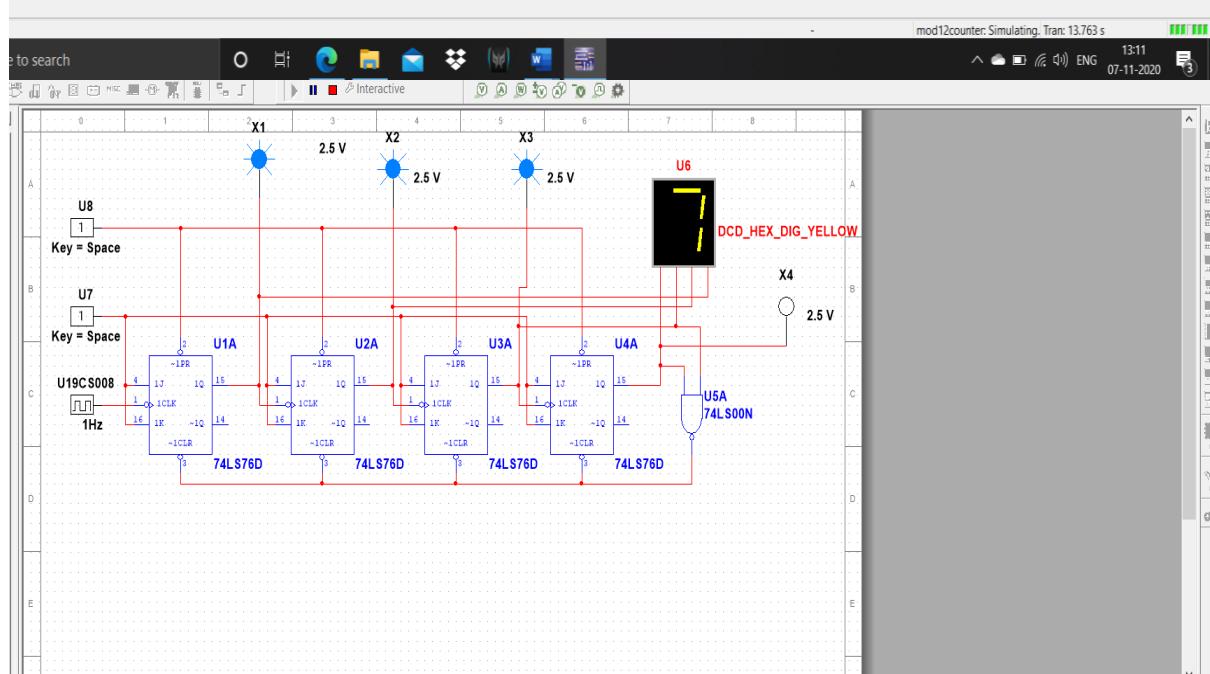
ember 2020, 13:03:53

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.



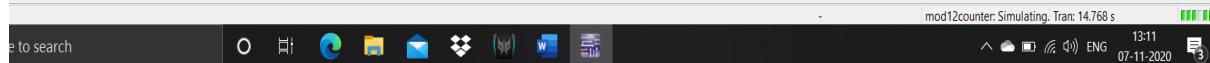
ember 2020, 13:03:53

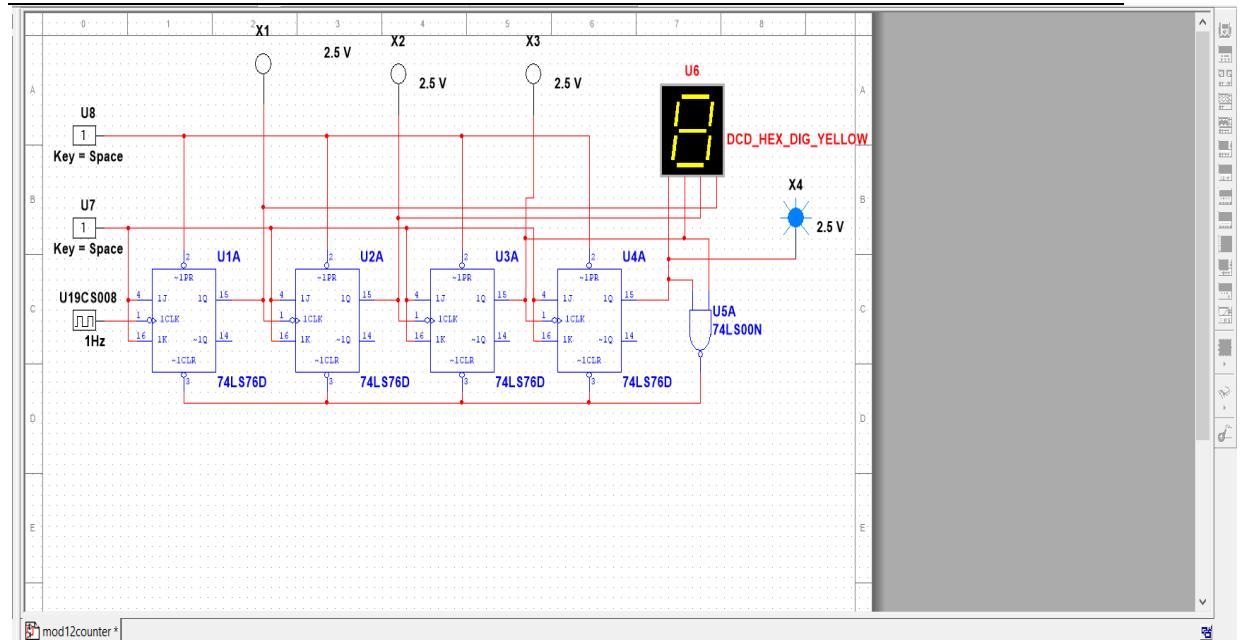
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.





ember 2020, 13:03:53

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

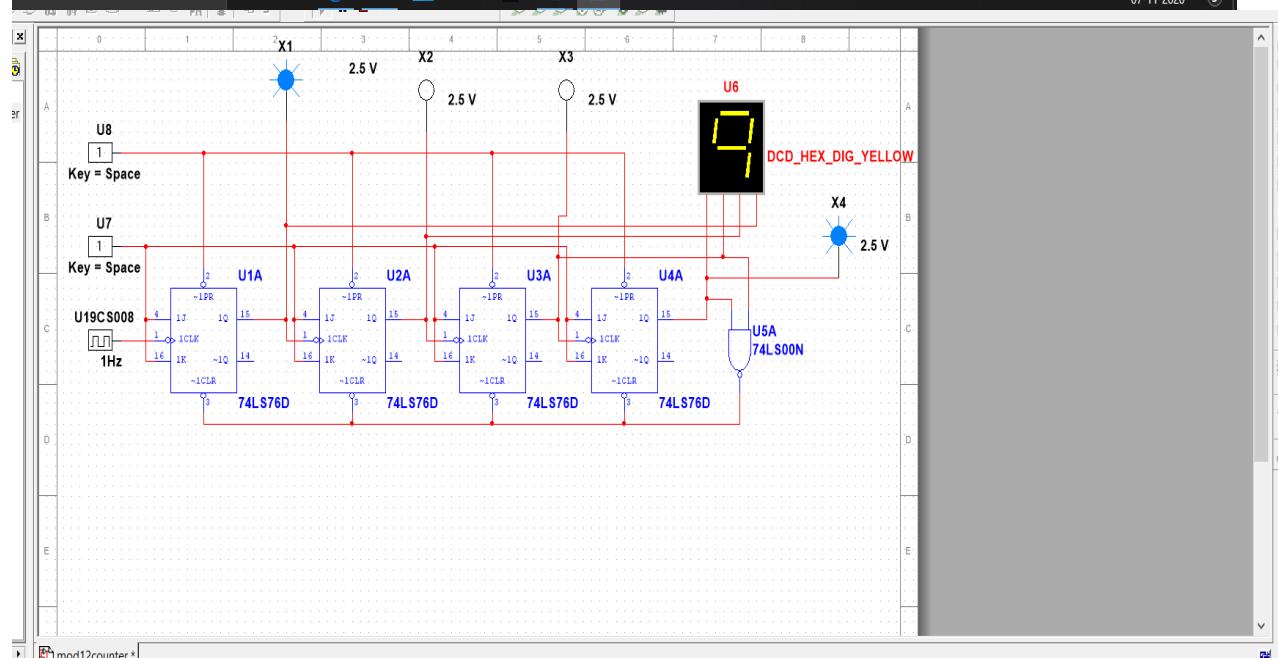
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

mod12counter: Simulating, Tran: 15.757 s

13:11 07-11-2020



ovember 2020, 13:03:53

ot to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

ot to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

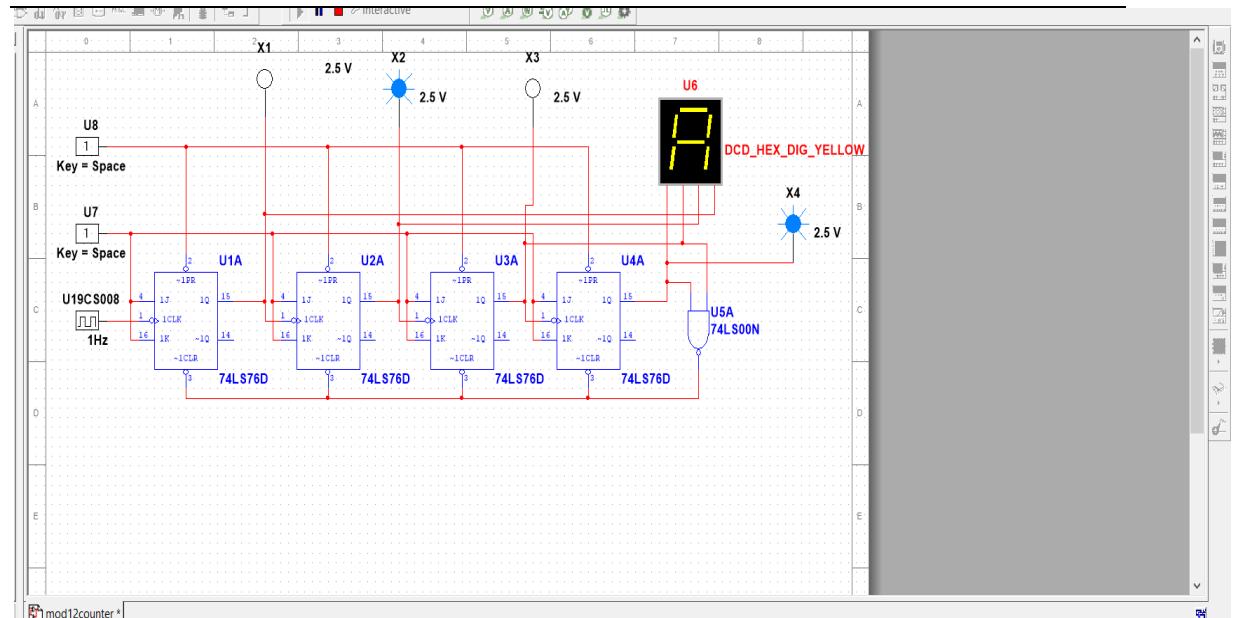
ot to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

ot to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

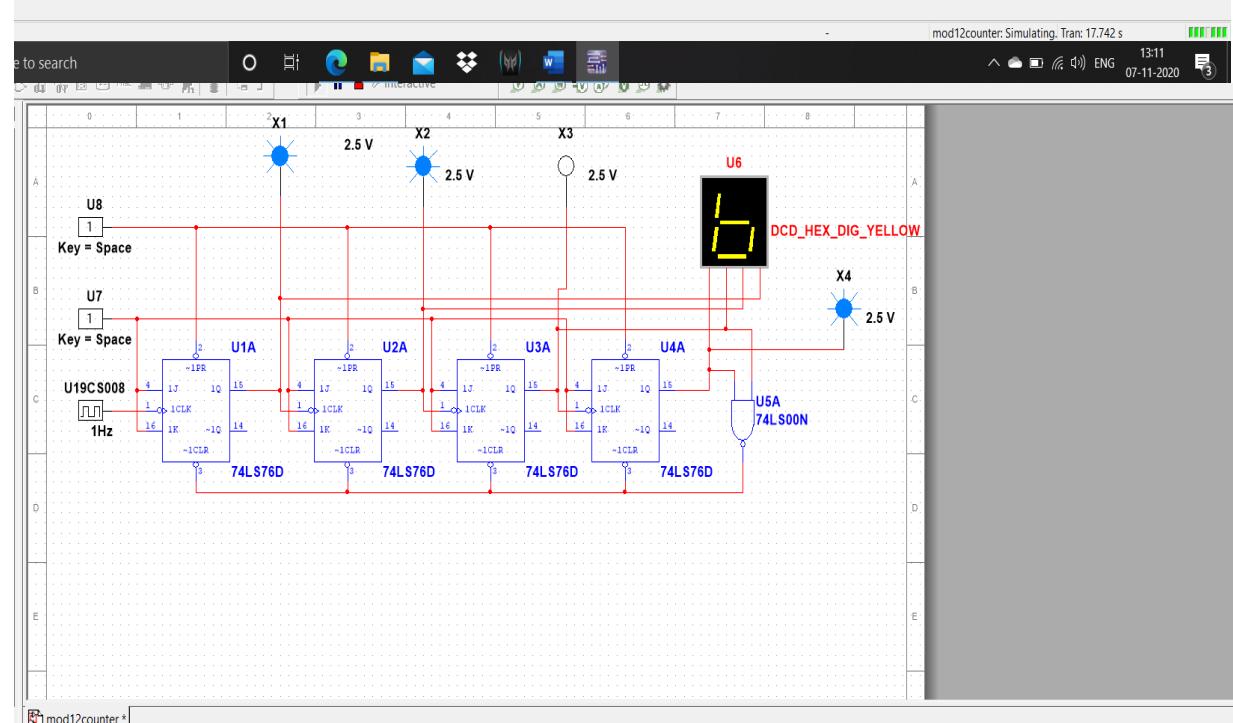
mod12counter: Simulating, Tran: 16.775 s

13:11 07-11-2020

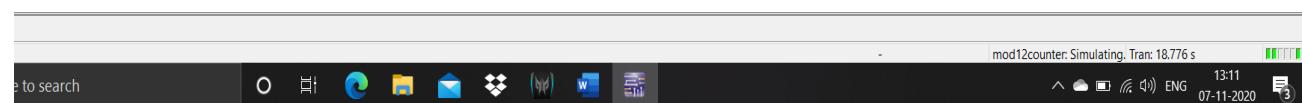




ember 2020, 13:03:53
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

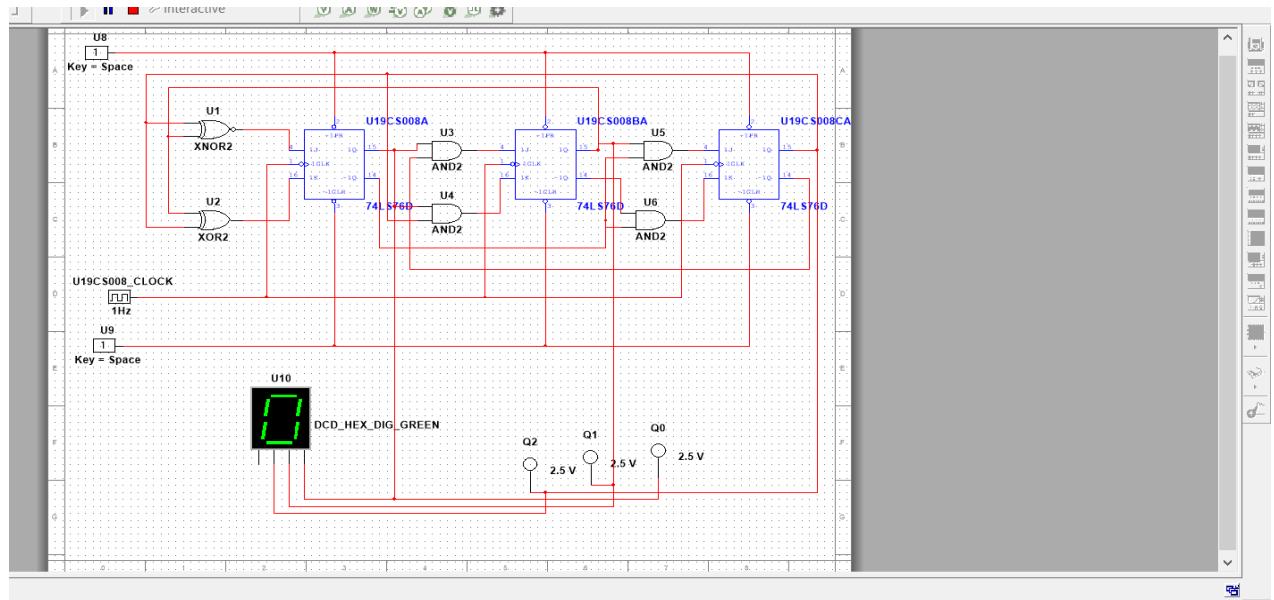


ember 2020, 13:03:53
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.
to the Grapher for interactive simulation. Connect an oscilloscope or run a transient simulation to see Grapher data.

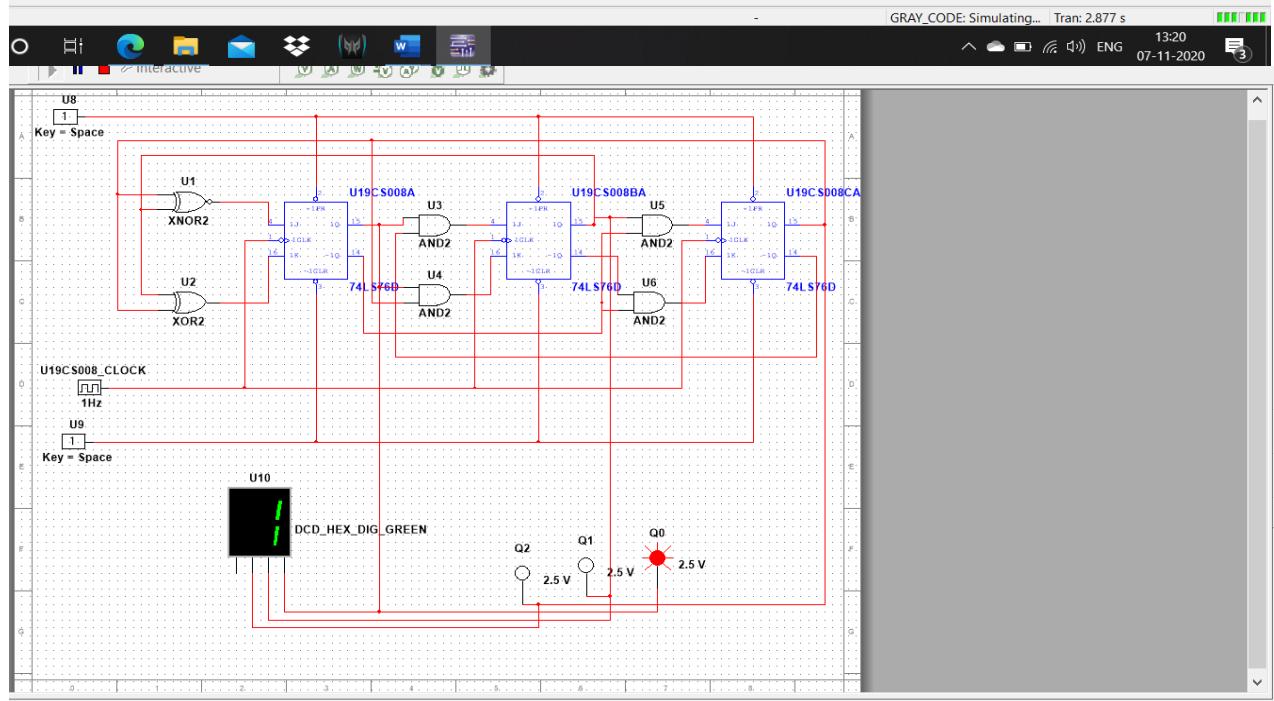




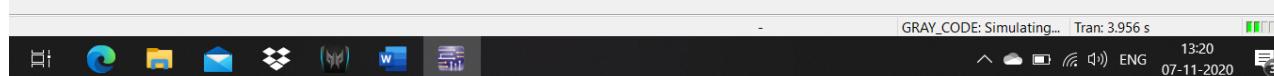
2. Design and implement 3-Bit Gray Code Up-Counter.

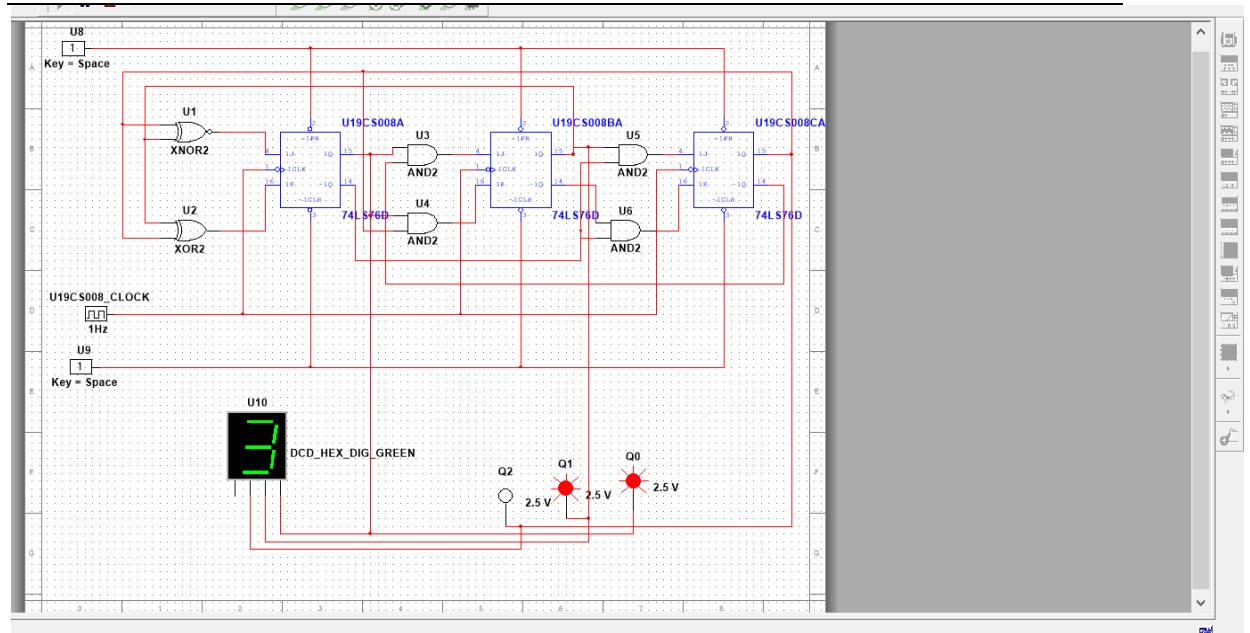


than an oscilloscope or run a transient simulation to see Grapher data.
 than an oscilloscope or run a transient simulation to see Grapher data.
 than an oscilloscope or run a transient simulation to see Grapher data.
 than an oscilloscope or run a transient simulation to see Grapher data.
 than an oscilloscope or run a transient simulation to see Grapher data.

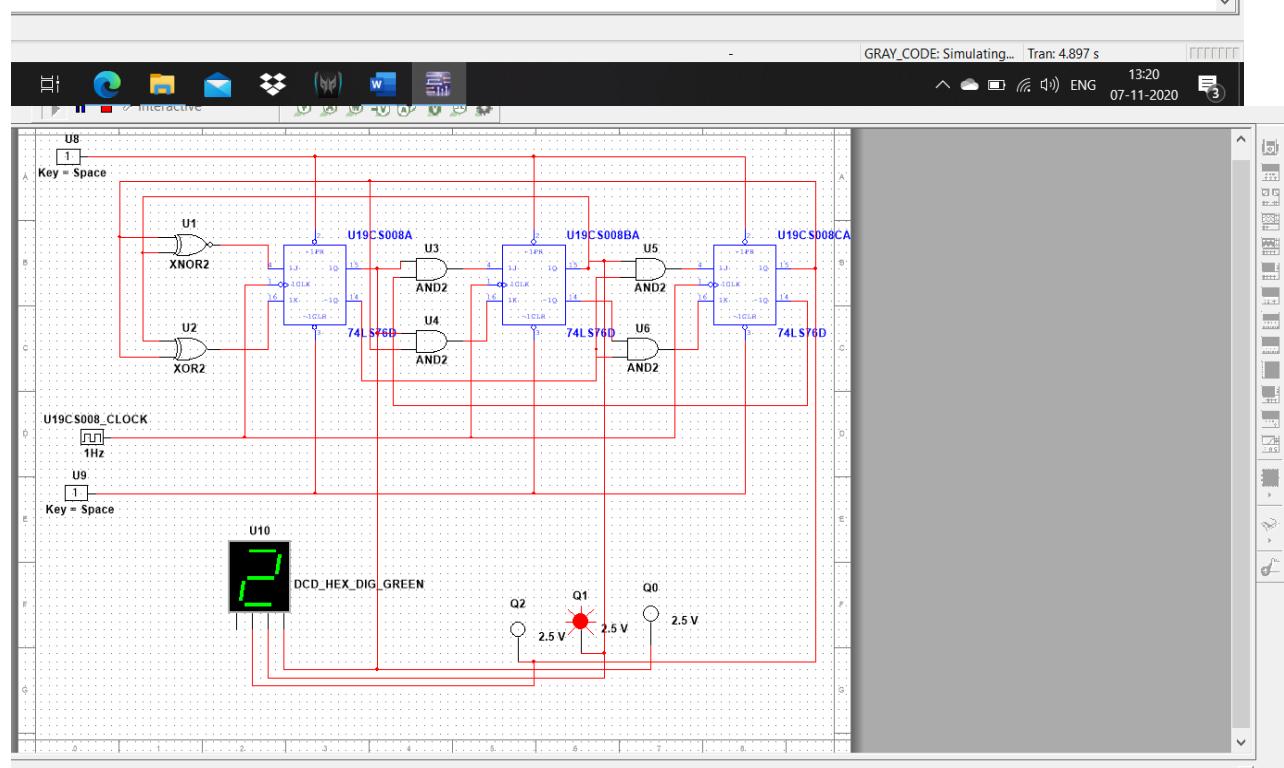


oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.

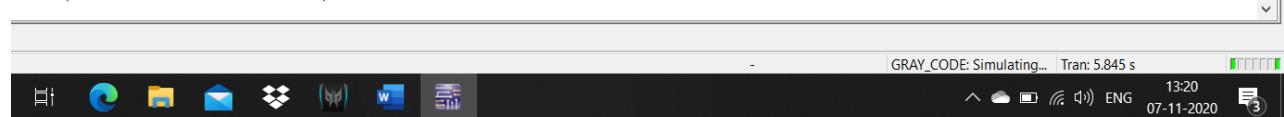


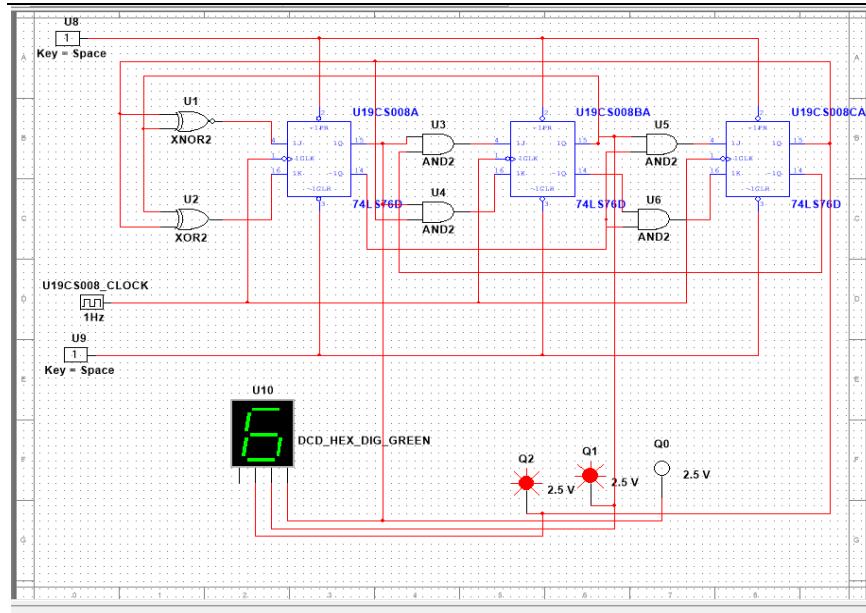


oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.



oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.





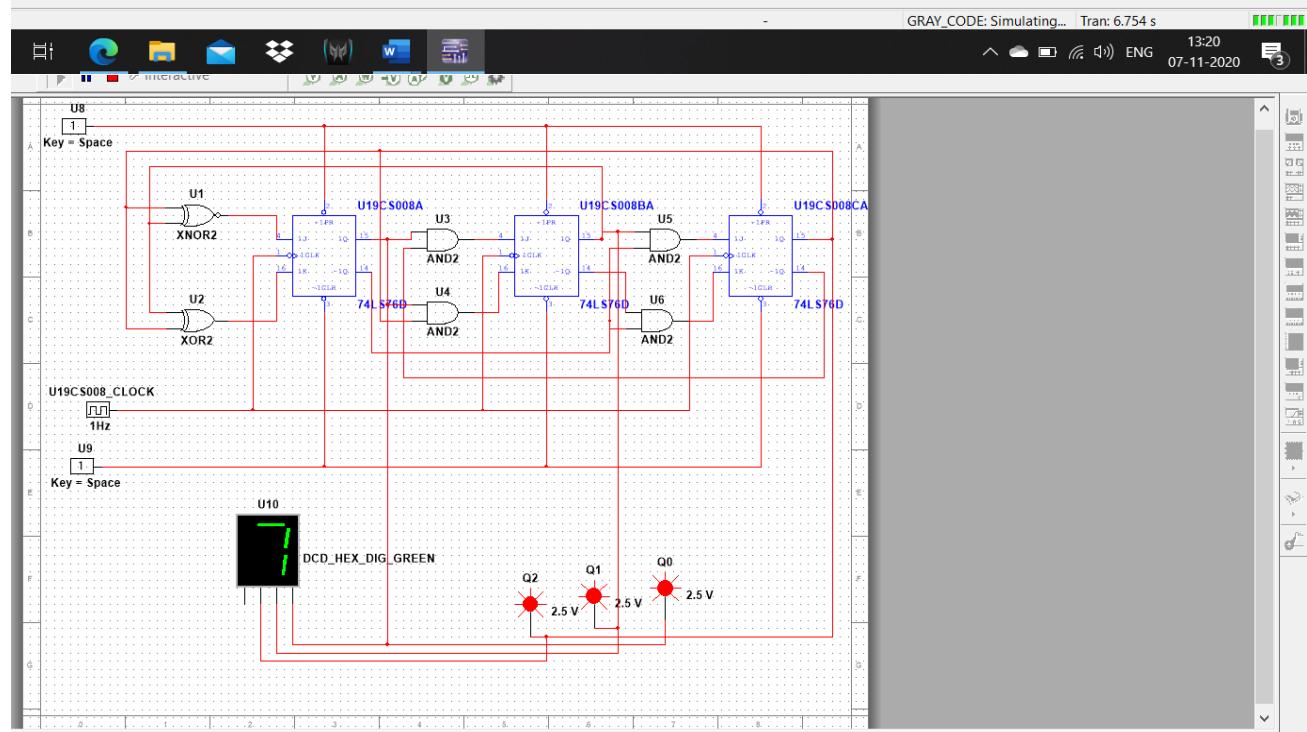
oscilloscope or run a transient simulation to see Grapher data.

oscilloscope or run a transient simulation to see Grapher data.

oscilloscope or run a transient simulation to see Grapher data.

oscilloscope or run a transient simulation to see Grapher data.

oscilloscope or run a transient simulation to see Grapher data.



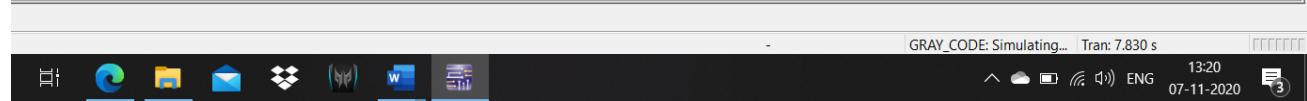
oscilloscope or run a transient simulation to see Grapher data.

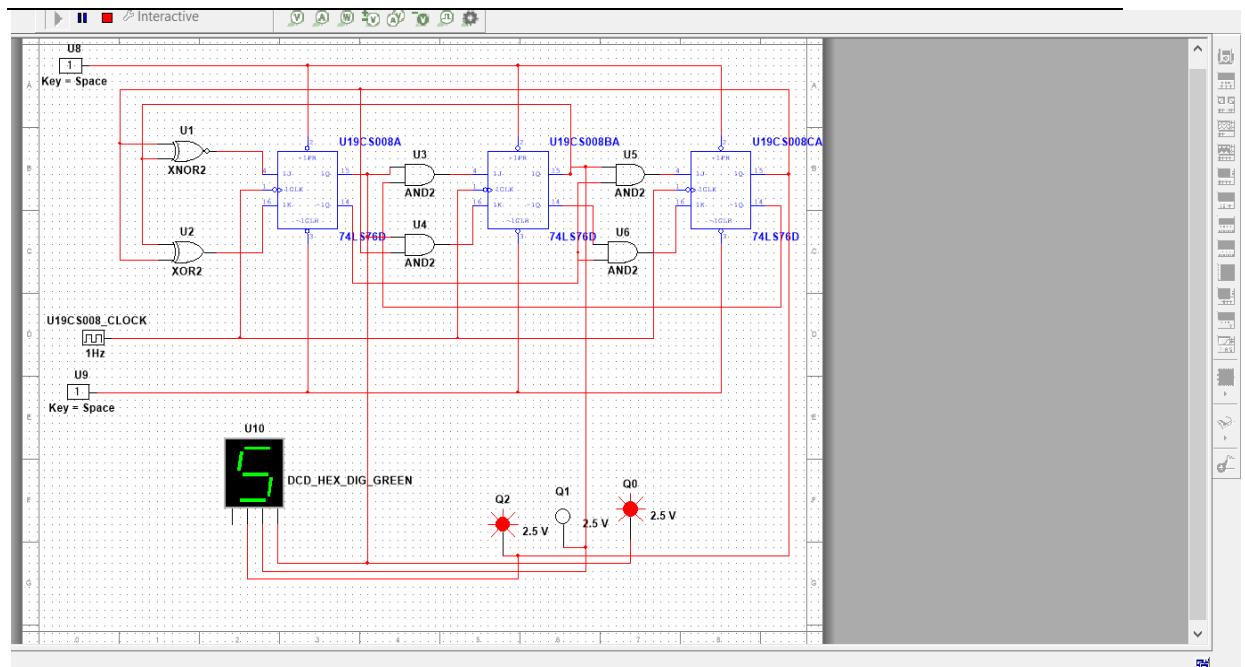
oscilloscope or run a transient simulation to see Grapher data.

oscilloscope or run a transient simulation to see Grapher data.

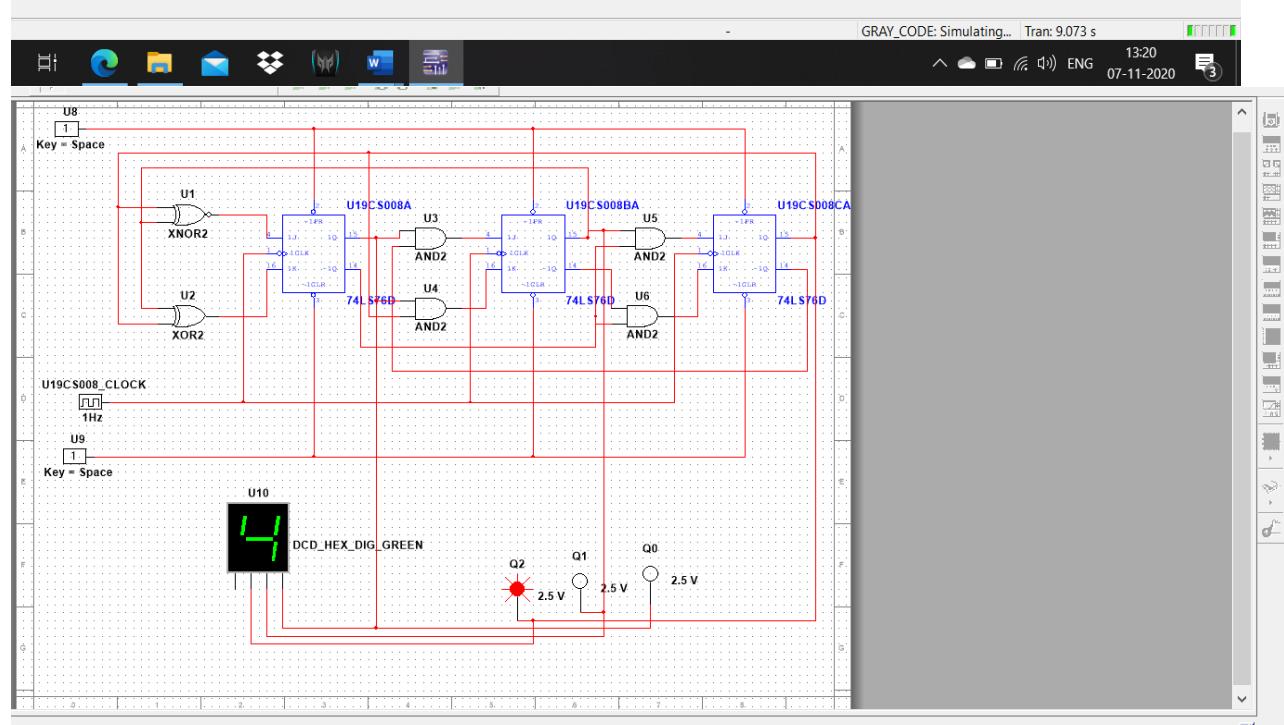
oscilloscope or run a transient simulation to see Grapher data.

oscilloscope or run a transient simulation to see Grapher data.

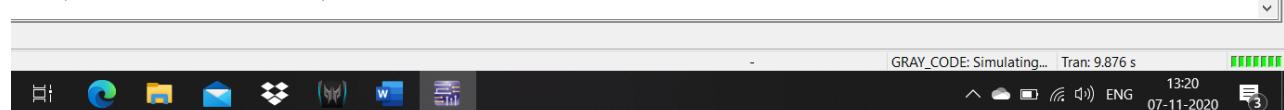




oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.



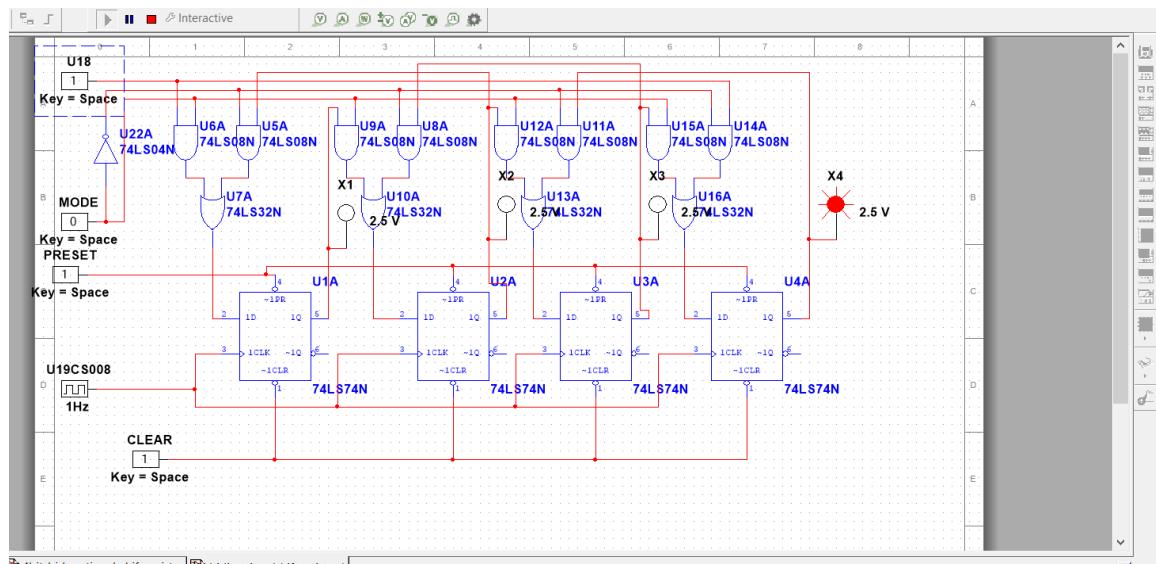
oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.
 oscilloscope or run a transient simulation to see Grapher data.



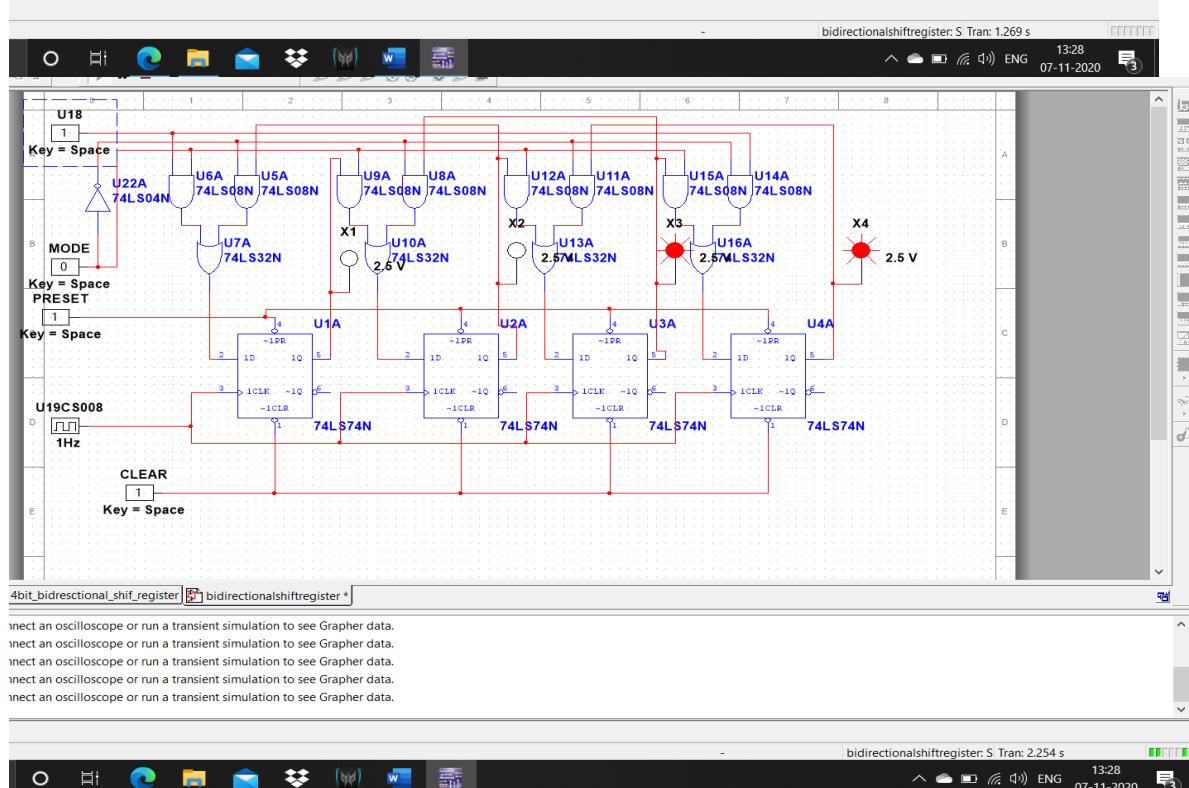


3. Design and implement in Multisim 4-Bit Bidirectional Shift Register.

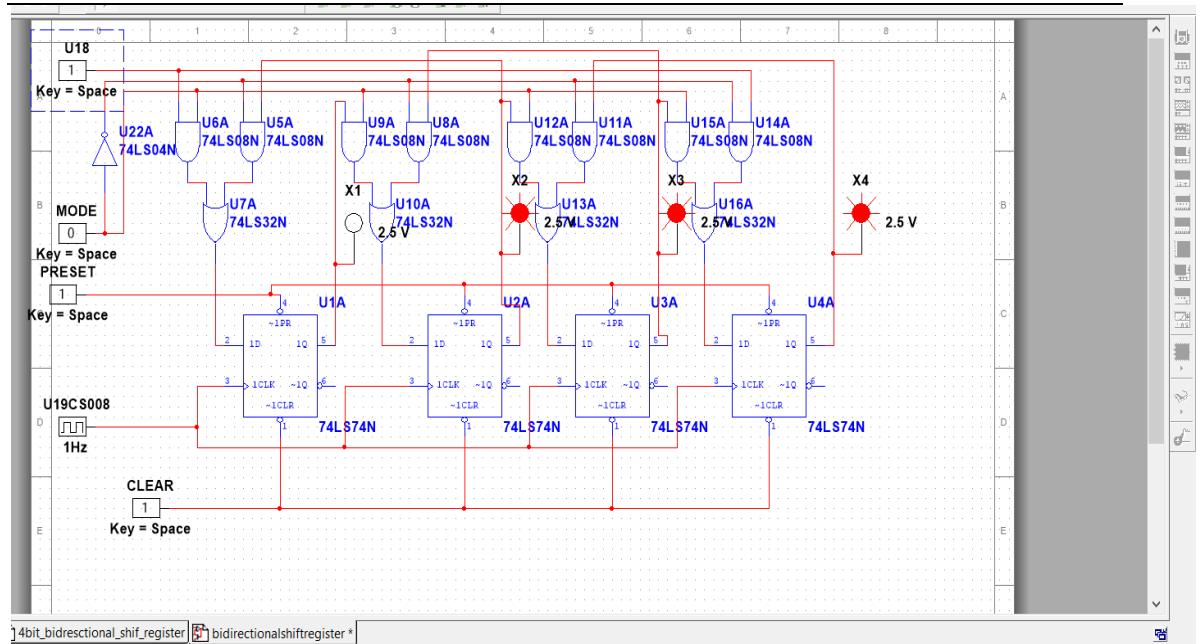
Mode=0 → SHIFT LEFT REGISTER



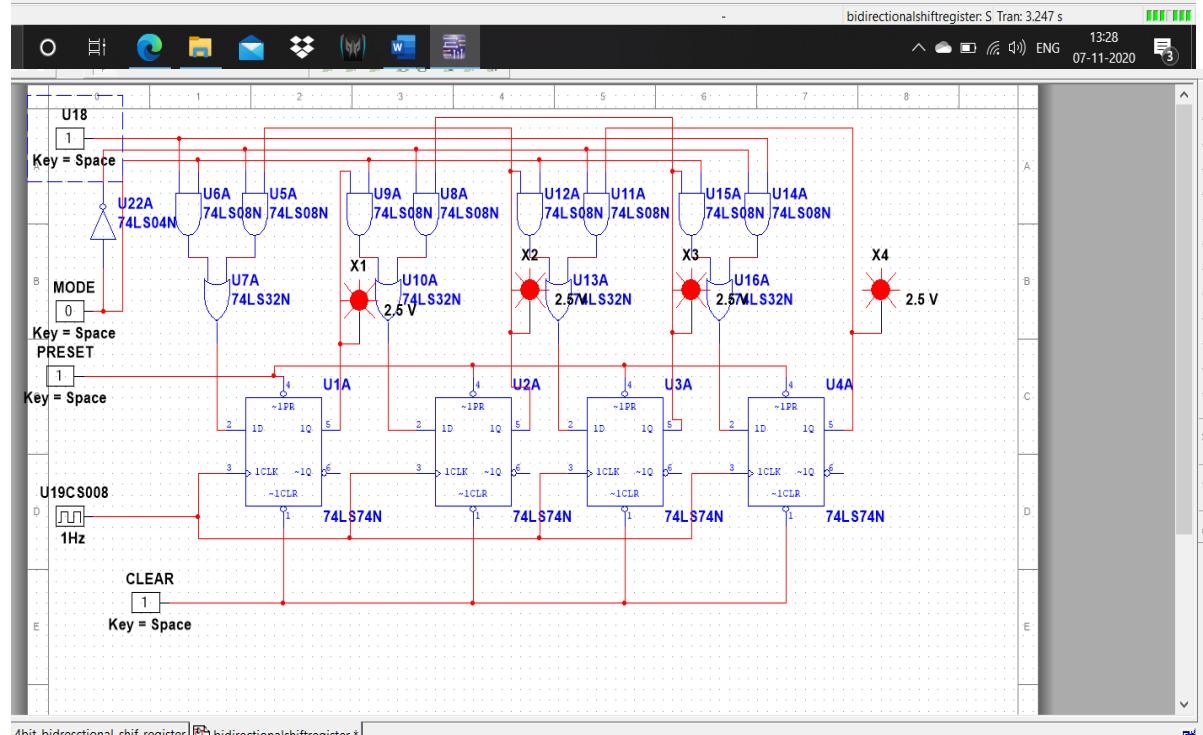
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.



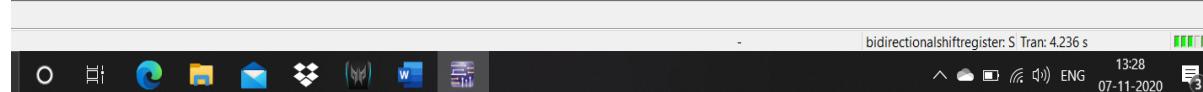
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.



Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.

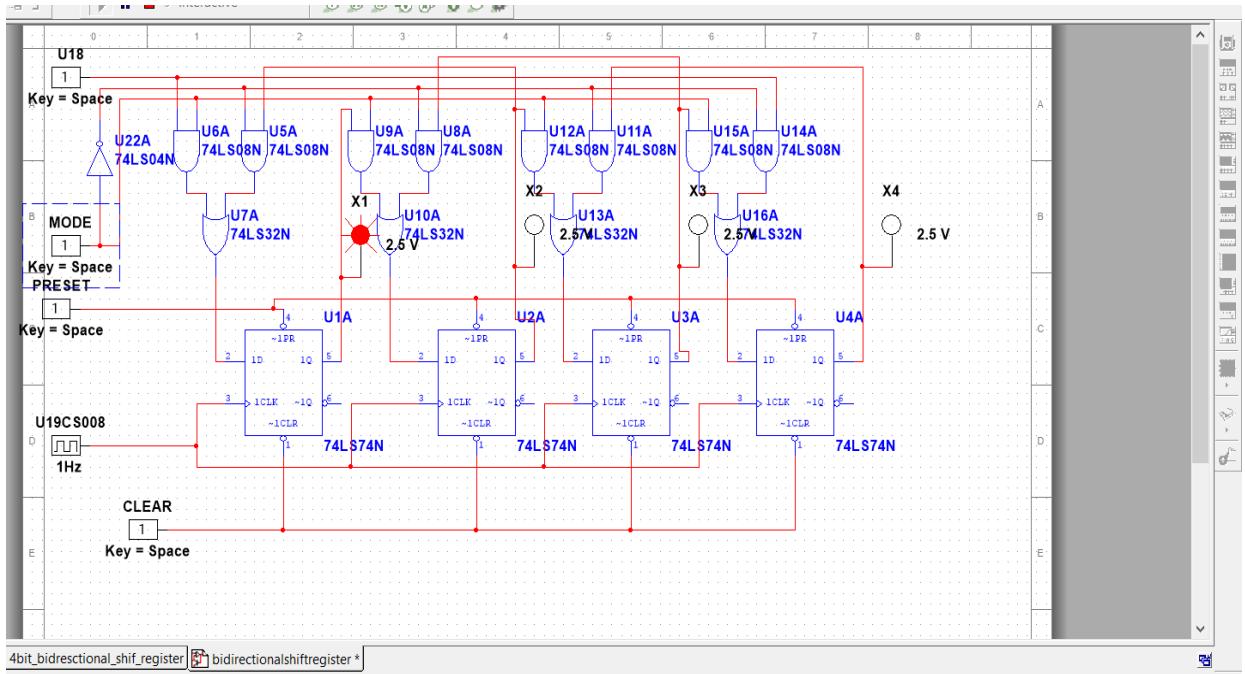


Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.
 Connect an oscilloscope or run a transient simulation to see Grapher data.





MODE=1 SHIFT RIGHT REGISTER



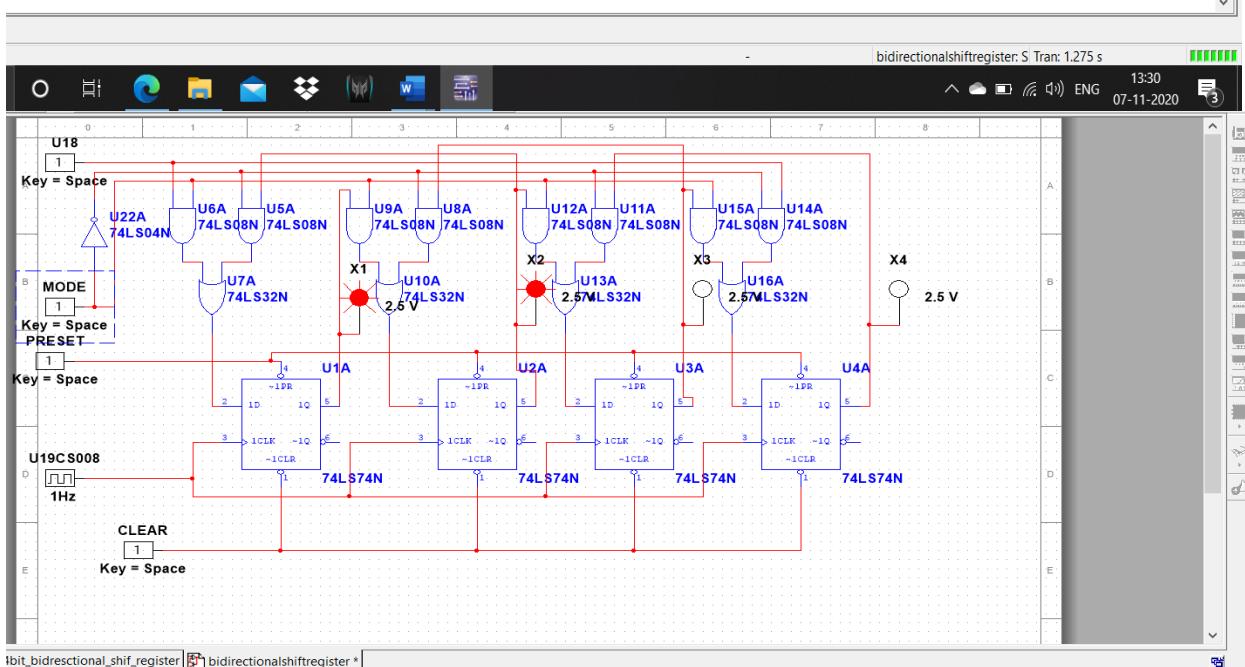
nect an oscilloscope or run a transient simulation to see Grapher data.

nect an oscilloscope or run a transient simulation to see Grapher data.

nect an oscilloscope or run a transient simulation to see Grapher data.

nect an oscilloscope or run a transient simulation to see Grapher data.

nect an oscilloscope or run a transient simulation to see Grapher data.



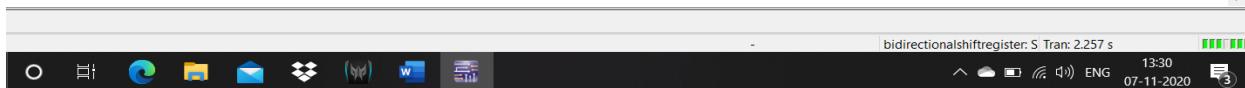
nect an oscilloscope or run a transient simulation to see Grapher data.

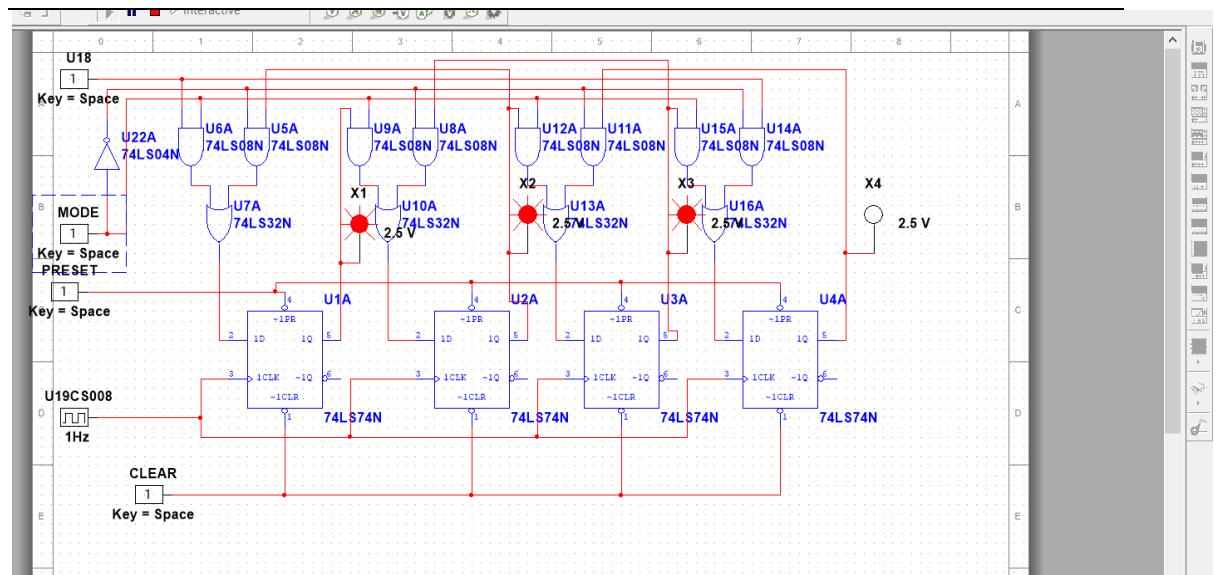
nect an oscilloscope or run a transient simulation to see Grapher data.

nect an oscilloscope or run a transient simulation to see Grapher data.

nect an oscilloscope or run a transient simulation to see Grapher data.

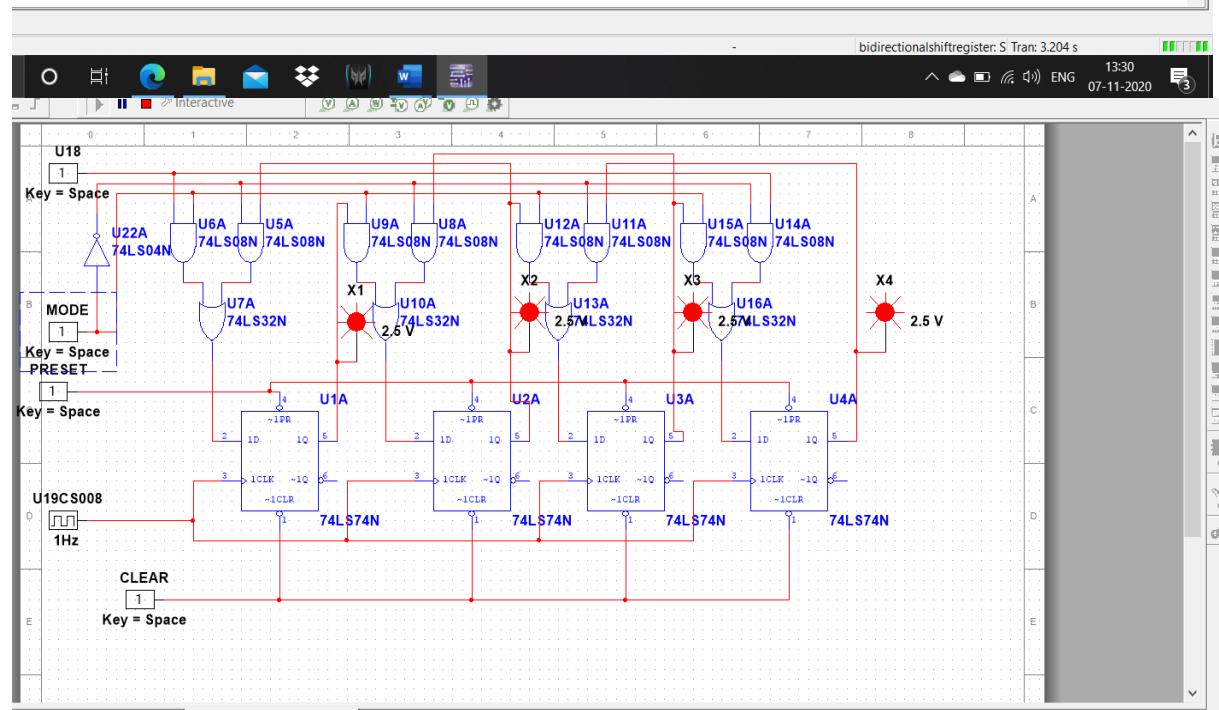
nect an oscilloscope or run a transient simulation to see Grapher data.



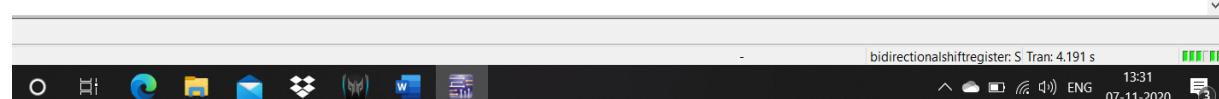


14bit_bidirectional_shif_register bidirectionalshiftregister *

Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.



Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.
Connect an oscilloscope or run a transient simulation to see Grapher data.





Expt. No:

12

Date:

11/11/2020**Multiplexers and Code Converters****AIM:** To study, design and implement:

1. Binary to Gray and Gray to Binary Code Converter (2-Bit, 3-Bit and 4-Bit)
2. Multiplexer using basic Gates (2x1 and 4x1)
3. Realise all the basic gates using 2x1 Multiplexer
4. Function Implementation using Multiplexers

SOFTWARE TOOLS / OTHER REQUIREMENTS:

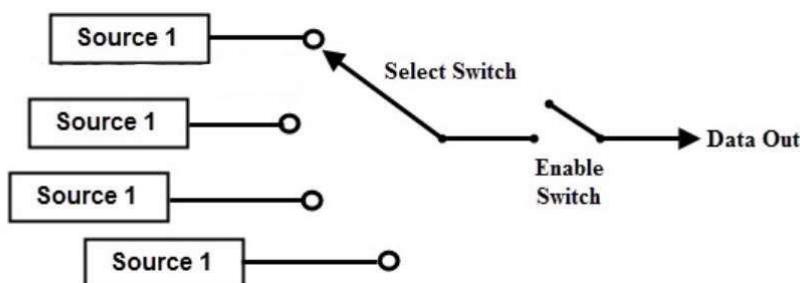
1. Multisim Simulator/Circuit Simulator

THEORY:**Multiplexer**

Multiplexing is the property of combining one or more signals and transmitting on a single channel. This is achieved by the device multiplexer. A multiplexer is the most frequently used combinational circuits and important building block in many in digital systems.

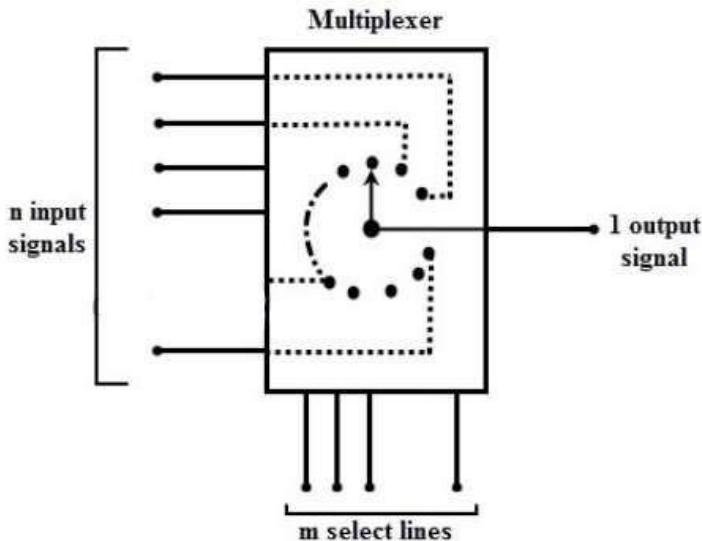
These are mostly used to form a selected path between multiple sources and a single destination. A basic multiplexer has various data input lines and a single output line. These are found in many digital system applications such as data selection and data routing, logic function generators, digital counters with multiplexed displays, telephone network, communication systems, waveform generators, etc. In this article we are going to discuss about types of multiplexers and its design.

The multiplexer or MUX is a digital switch, also called as data selector. It is a combinational circuit with more than one input line, one output line and more than one select line. It allows the binary information from several input lines or sources and depending on the set of select lines, particular input line, is routed onto a single output line. The basic idea of multiplexing is shown in figure below in which data from several sources are routed to the single output line when the enable switch is ON. That is how the multiplexers are also called as ‘many to one’ combinational circuits.





The below figure shows the block diagram of a multiplexer consisting of n input lines, m selection lines and one output line. If there are m selection lines, then the number of possible input lines is 2^m . Alternatively we can say that if the number of input lines is equal to 2^m , then m selection lines are required to select one of n (consider $2^m = n$) input lines.



This type of multiplexer is referred to as $2^n \times 1$ multiplexer or 2^n -to-1 multiplexer. For example, if one of the 4 input lines has to be selected, then two select lines are required. Similarly, to select one of 8 input lines, three select lines are required.

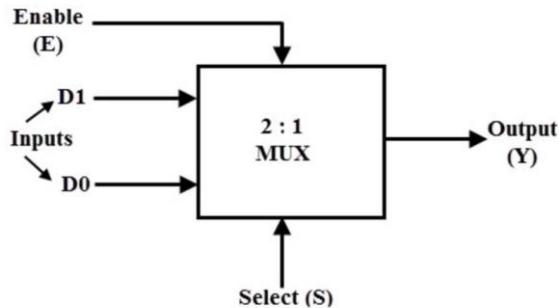
Generally, the number of data inputs to a multiplexer is a power of two such as 2, 4, 8, 16, etc. Some of the mostly used multiplexers include 2-to-1, 4-to-1, 8-to-1 and 16-to-1 multiplexers.

These multiplexers are available in IC forms with different input and select line configurations. Some of the available multiplexer ICs include 74157 (2-to-1 MUX), 78158 (2-to-1 MUX), 74352 (4-to-1 MUX), 74153 (4-to-1 MUX), 74152 (8-to-1 MUX) and 74150 (16-to-1 MUX).

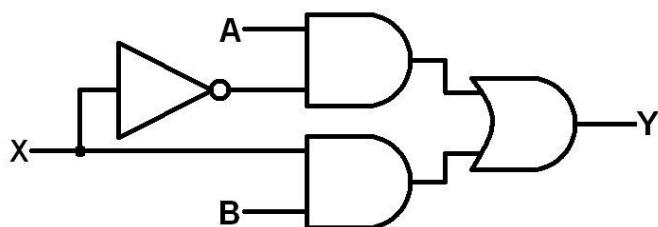
2x1 MUX

A 2-to-1 multiplexer consists of two inputs D0 and D1, one selects input S and one output Y. Depends on the select signal, the output is connected to either of the inputs. Since there are two input signals only two ways are possible to connect the inputs to the outputs, so one selects is needed to do these operations.

If the select line is low, then the output will be switched to D0 input, whereas if select line is high, then the output will be switched to D1 input. The figure below shows the block diagram of a 2-to-1 multiplexer which connects two 1-bit inputs to a common destination.



Depending on the selector switching the inputs are produced at outputs, i.e., D0, D1 and are switched to the output for S=0 and S=1 respectively. Thus, the Boolean expression for the output becomes D0 when S=0 and output is D1 when S=1.



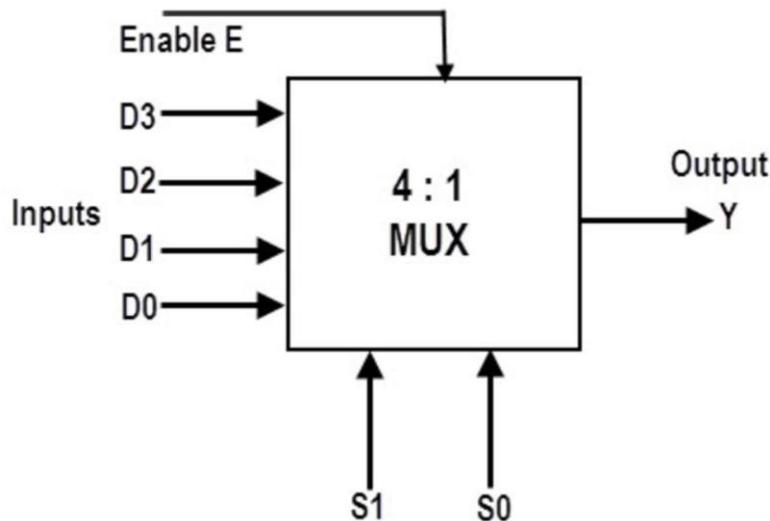
From the above output expression, the logic circuit of 2-to-1 multiplexer can be implemented using logic gates as shown in figure. It consists of two AND gates, one NOT gate and one OR gate. When the select line, S=0, the output of the upper AND gate is zero, but the lower AND gate is D0.

Thus, the output generated by the OR gate is equal to D0. Similarly, when S=1, the output of the lower AND gate is zero, but the output of upper AND gate is D1. Therefore, the output of the OR gate is D1. Thus, the above given Boolean expression is satisfied by this circuit.

Select	Inputs		Output
0	0	0	0
0	0	1	1
1	1	0	1
1	1	1	1

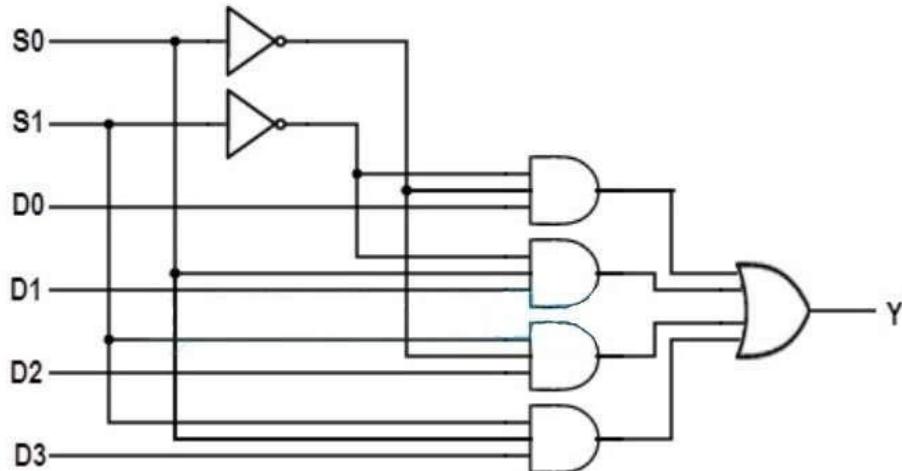
In some cases, two or more multiplexers are fabricated on a single IC because simple logic gates can implement the multiplexer. Generally, four 2 line to 1-line multiplexers are fabricated in a single IC as shown in figure below. Some of these ICs of 2 to 1 multiplexer include IC 74157 and IC 74158. The selection line controls the input lines to the output in all four multiplexers.

The output Y1 can be selected such that its value may be equal to A1 or B1, Y2 can be either A2 or B2 and so on. The control input E enables and disables all the multiplexers, i.e., when E=1, outputs of all the multiplexer is zero irrespective of the value of S.

**4X1 Mux**

A 4-to-1 multiplexer consists of four data input lines as D0 to D3, two select lines as S0 and S1 and a single output line Y. The select lines S1 and S0 select one of the four input lines to connect the output line. The particular input combination on select lines selects one of input (D0 through D3) to the output.

The figure below shows the block diagram of a 4-to-1 multiplexer in which the multiplexer decodes the input through select line.



The truth table of a 4-to-1 multiplexer is shown below in which four input combinations 00, 10, 01 and 11 on the select lines respectively switches the inputs D0, D2, D1 and D3 to the output. That means when S1=0 and S0 =0, the output at Y is D0, similarly Y is D1 if the select inputs S1=0 and S0= 1 and so on.



Select Data Inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

To get the total data output from the multiplexer, all these product terms are to be summed and then the final Boolean expression of this multiplexer is given as

From the above expression of the output, a 4-to-1 multiplexer can be implemented by using basic logic gates. The below figure shows the logic circuit of 4:1 MUX which is implemented by four 3-inputs AND gates, two 1-input NOT gates, and one 4-inputs OR gate.

In this circuit, each data input line is connected as input to an AND gate and two select lines are connected as other two inputs to it. The AND gate output is connected to with inputs of OR gate so as to produce the output Y.

Generally, this type of multiplexers is available in dual IC forms and most common type is IC 74153 which is a dual 4-to-1 line multiplexer. It consists of two identical and independent 4-to-1 multiplexers. It has two separate enable or strobe inputs to switch ON or OFF the multiplexers.

Application of Multiplexer

In all types of digital system applications, multiplexers find its immense usage. Since these allows multiple inputs to be connected independently to a single output, these are found in variety of applications including data routing, logic function generators, control sequencers, parallel-to-serial converters, etc.

Codes and code converters

Coding is the process of translating the input information which can be understandable by the machine or a particular device. Coding can be used for security purpose to protect the information from stealing or interrupting. Actually this is not the latest trend, in previous days also the king of the kingdom used to send the information to other kingdom which some code words.

The code converters are used to convert the information in to the code which we want. These are basically encoders and decoders which converts the data in to an encoded form. The below explains some digital codes used in digital electronics.



Excess-3 code:

It is also known as self-complementary code as the complement of any number (0-9) will be available within these 10 numbers. As the name implies it is excess of 3 for the regular BCD code i.e. if u add 3(0011) to the BCD Addition u can get Excess-3 code.

Grey code

It is also called as unit distance code and reflective code word. It is not arithmetic code that there are no specific weights assigned to each bit positions.

The successive code words differ only one bit this nature of the code is called unit distance code

The above table shows the Decimal, BCD and Grey code, the last column shows grey code of Decimal 1, 2, 3 & 4. The grey code for 1 & 2 differs only one bit same for 2&3 etc. that's y this code is called unit distance code as the distance between the successive codes Is one bit.

Switching activity is reduced because of one-digit change in consequence numbers, hence low power consumption and fast response.

It is also called as reflective code as the code of n bits are again reflected from n bit code. In the below table u can find 2-bit code which shows the reflective nature of 0 is 2 and 1 is 3. Same is applicable for 3-bit code.

The above table shows the Decimal, BCD and Grey code, the last column shows grey code of Decimal 1, 2, 3 & 4. The grey code for 1 & 2 differs only one bit same for 2&3 etc. that's y this code is called unit distance code as the distance between the successive codes Is one bit.

Switching activity is reduced because of one-digit change in consequence numbers, hence low power consumption and fast response.

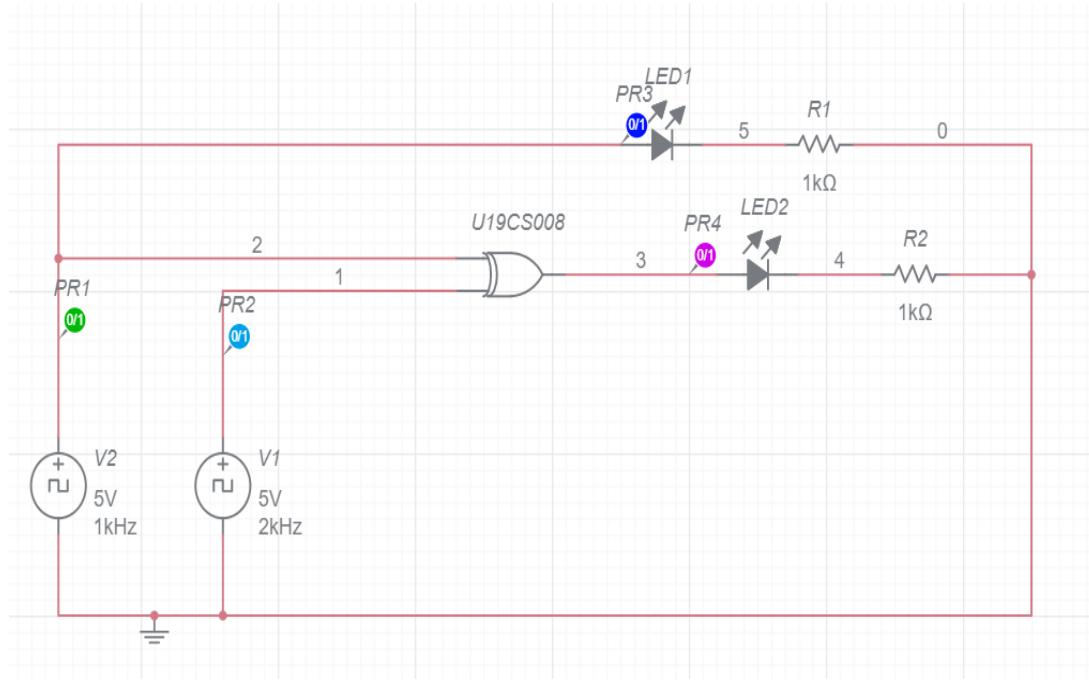
It is also called as reflective code as the code of n bits are again reflected from n bit code. In the below table u can find 2-bit code which shows the reflective nature of 0 is 2 and 1 is 3. Same is applicable for 3-bit code.



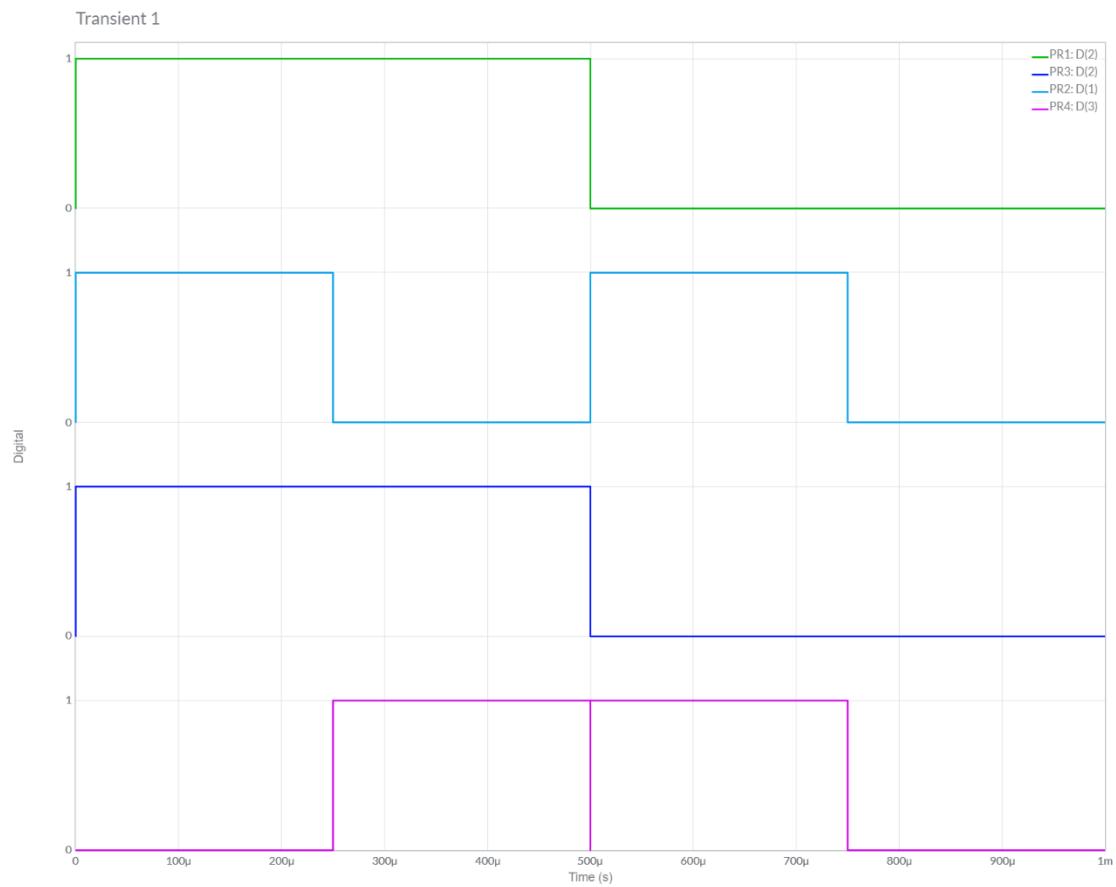
SIMULATION SCREENSHOTS

**BINARY TO GRAY CODE
2- BIT**

CIRCUIT DAIGRAM:



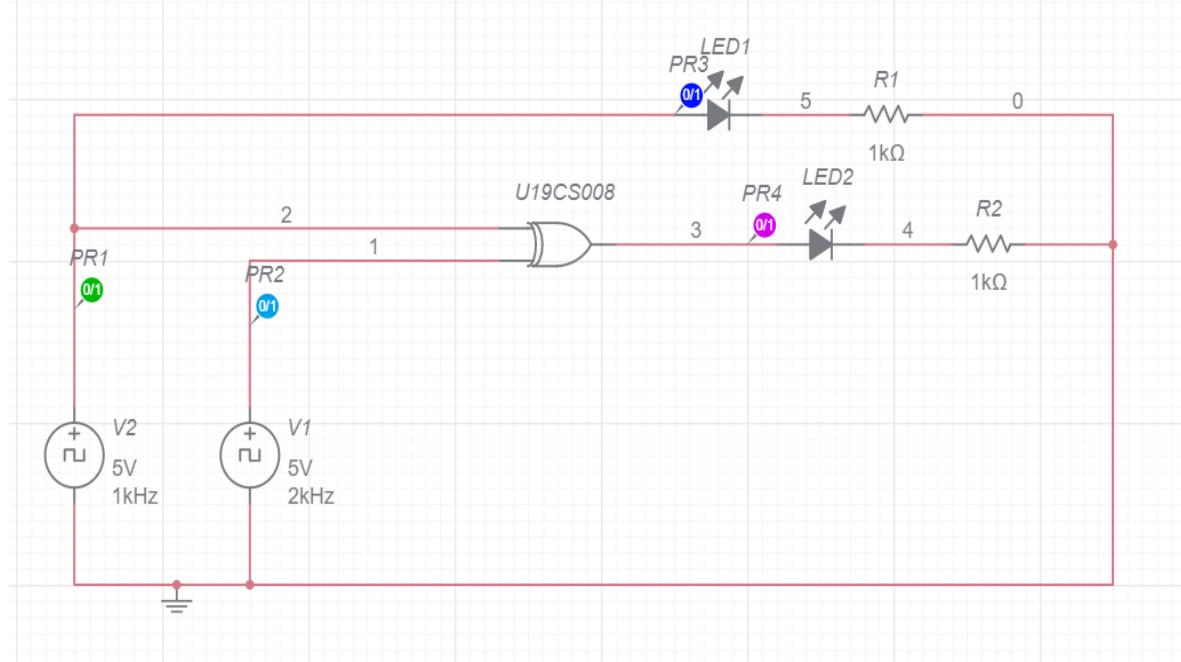
GRAPH:



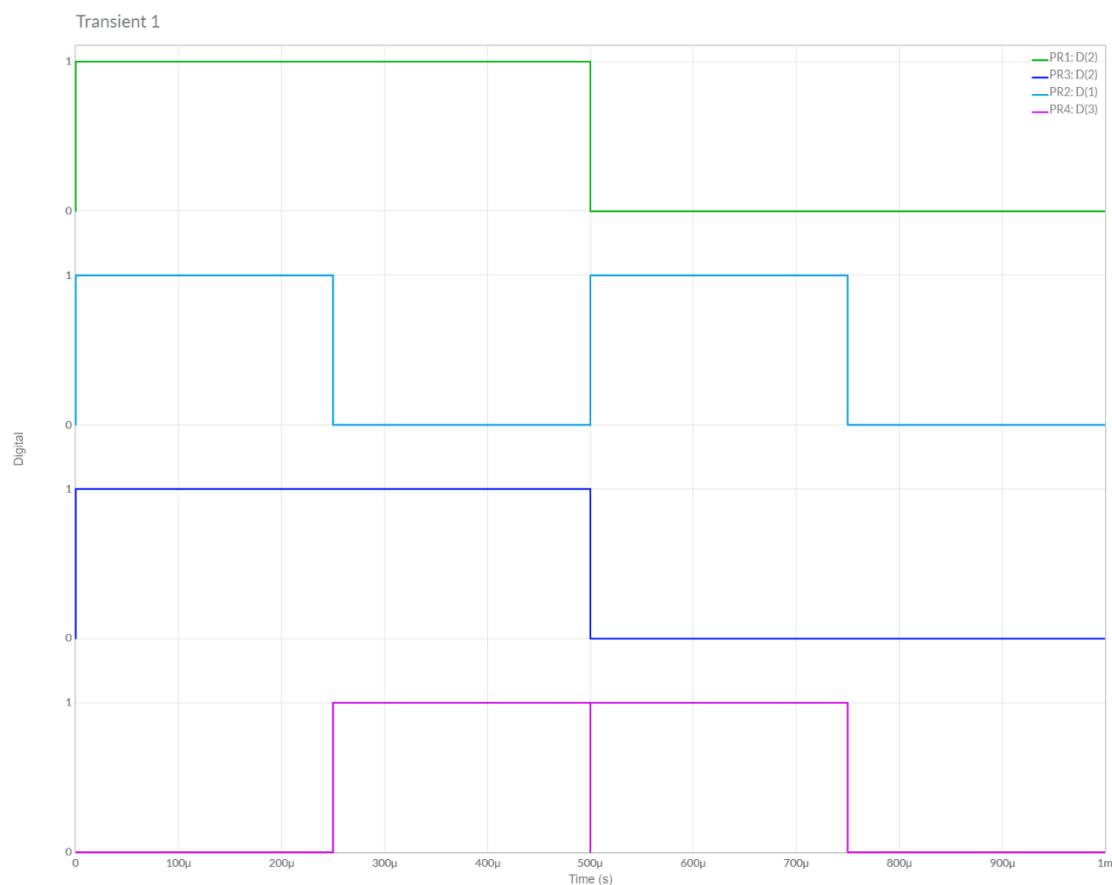


**GRAY TO BINARY CODE
2- BIT**

CIRCUIT DAIGRAM:



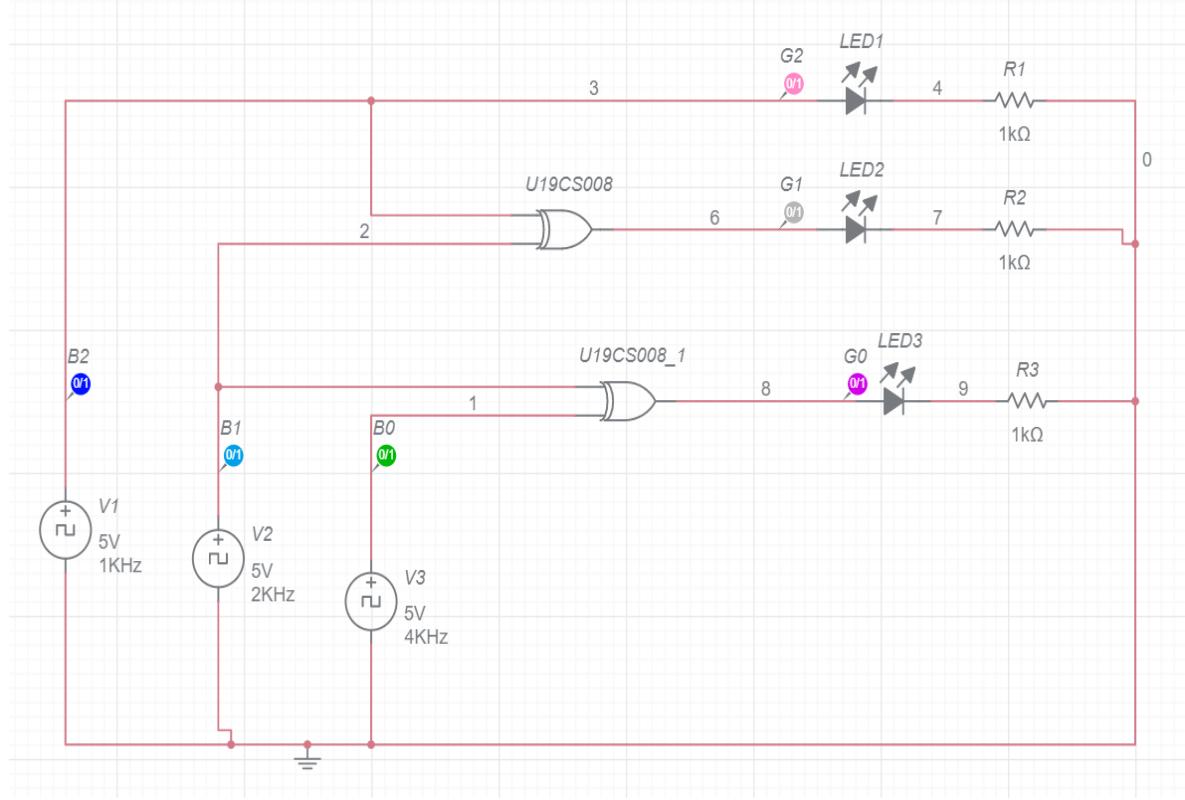
GRAPH:



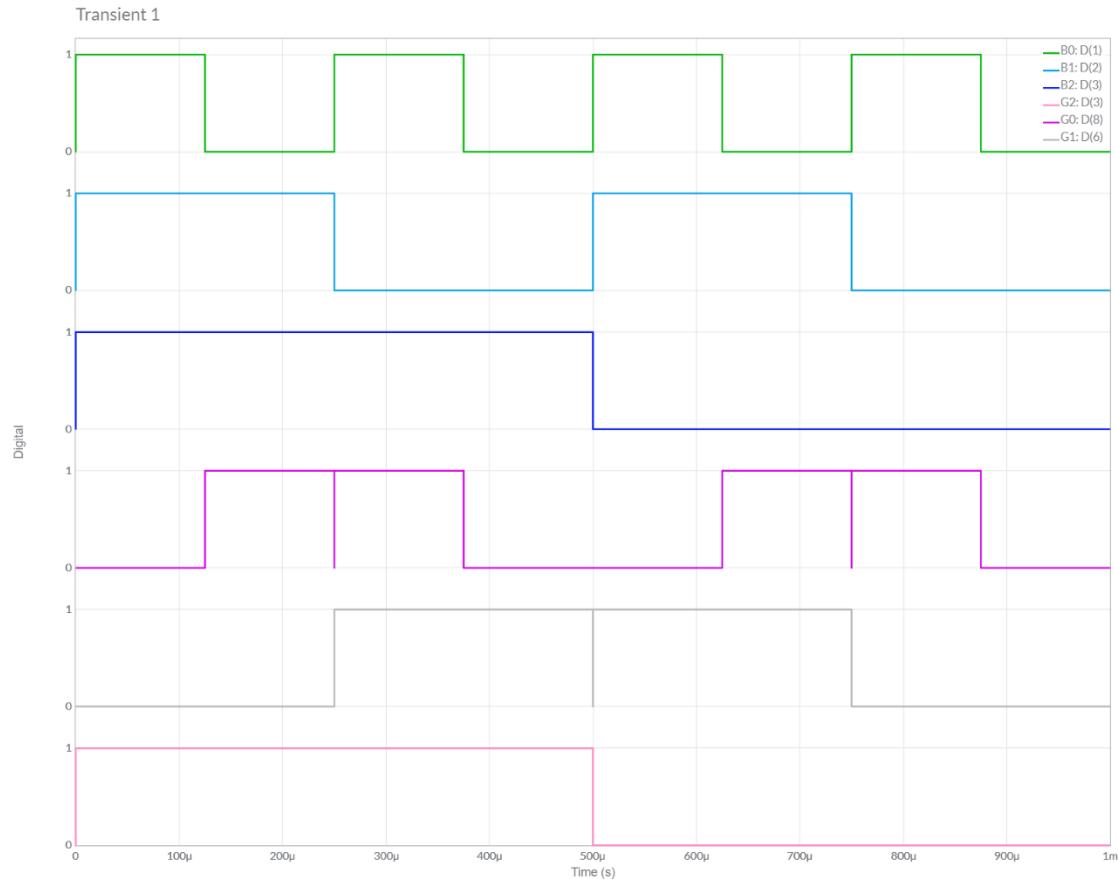


**BINARY TO GRAY CODE
3 - BIT**

CIRCUIT DAIGRAM:



GRAPH:

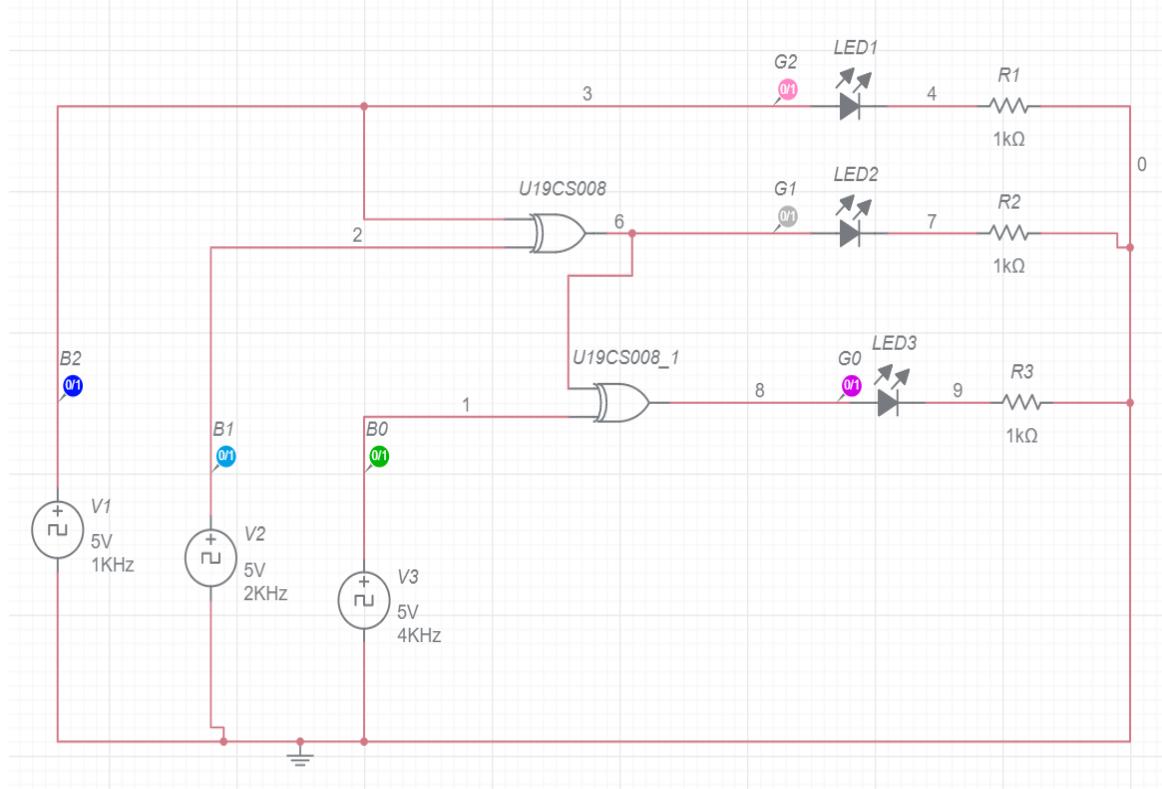




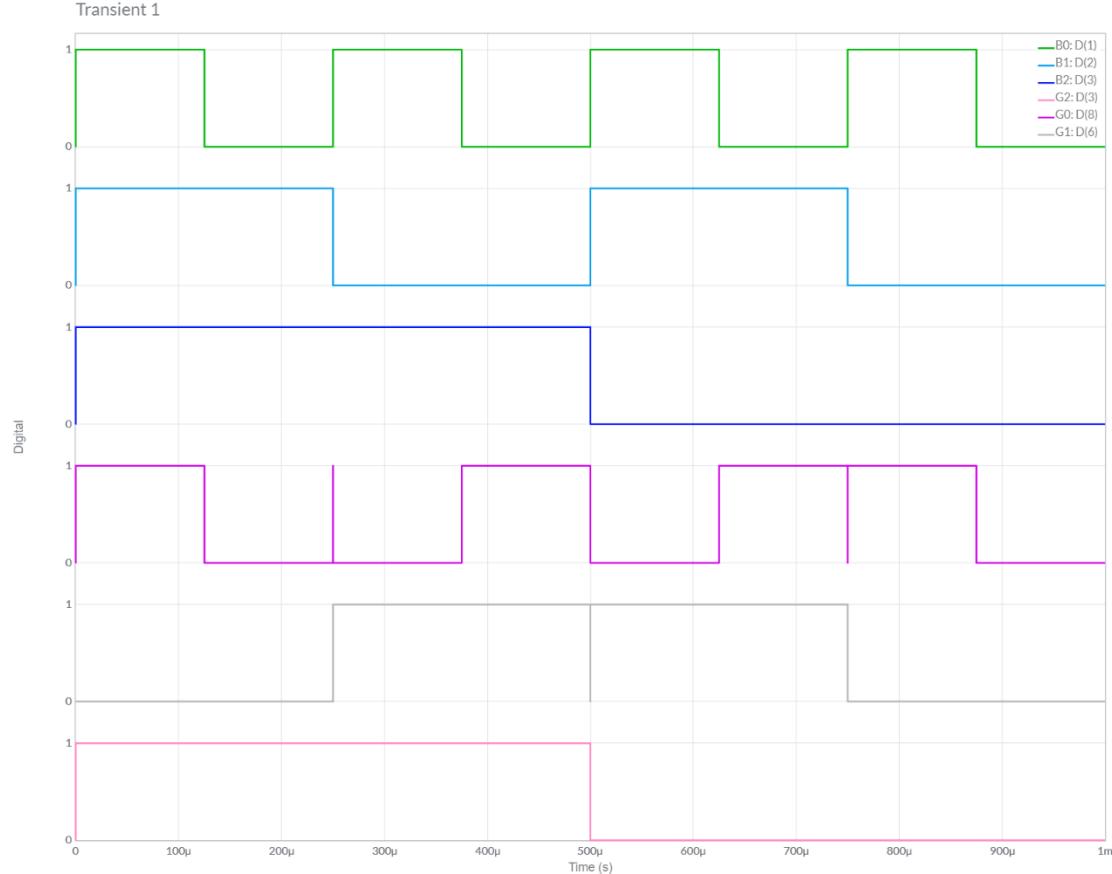
GRAY TO BINARY CODE

3 - BIT

CIRCUIT DAIGRAM:



GRAPH:

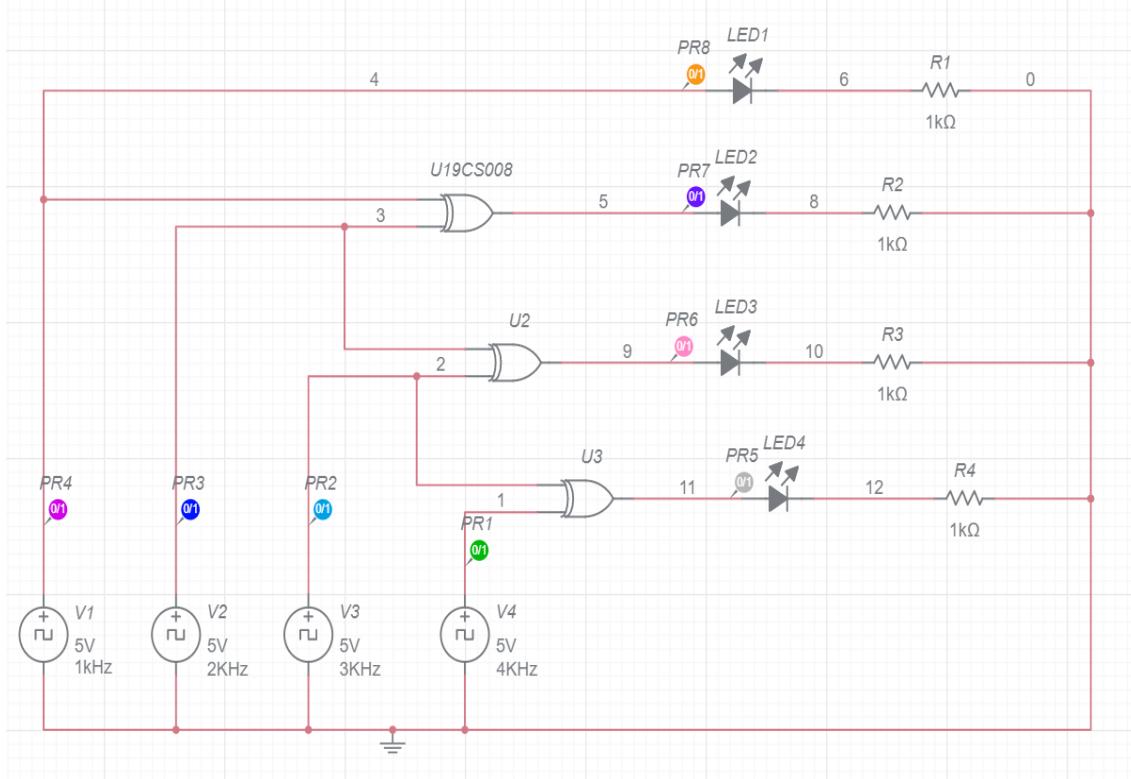




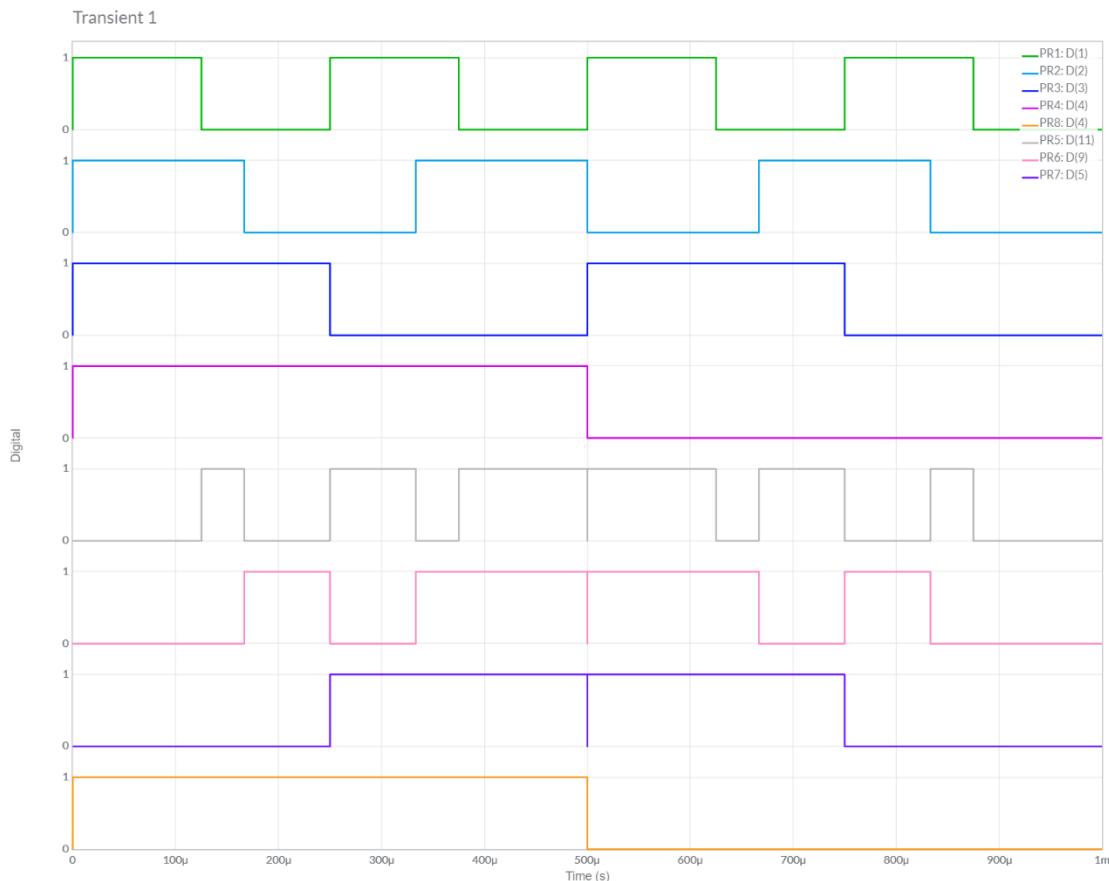
BINARY TO GRAY CODE

4 - BIT

CIRCUIT DAIGRAM:



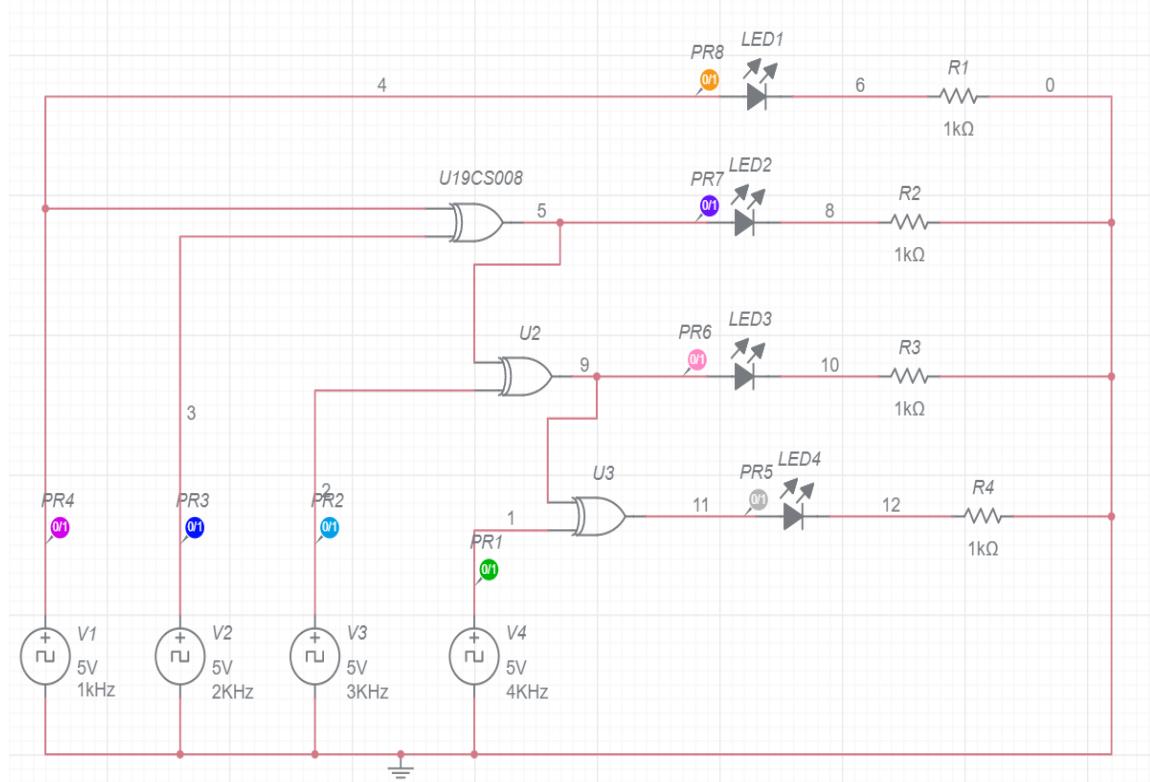
GRAPH:



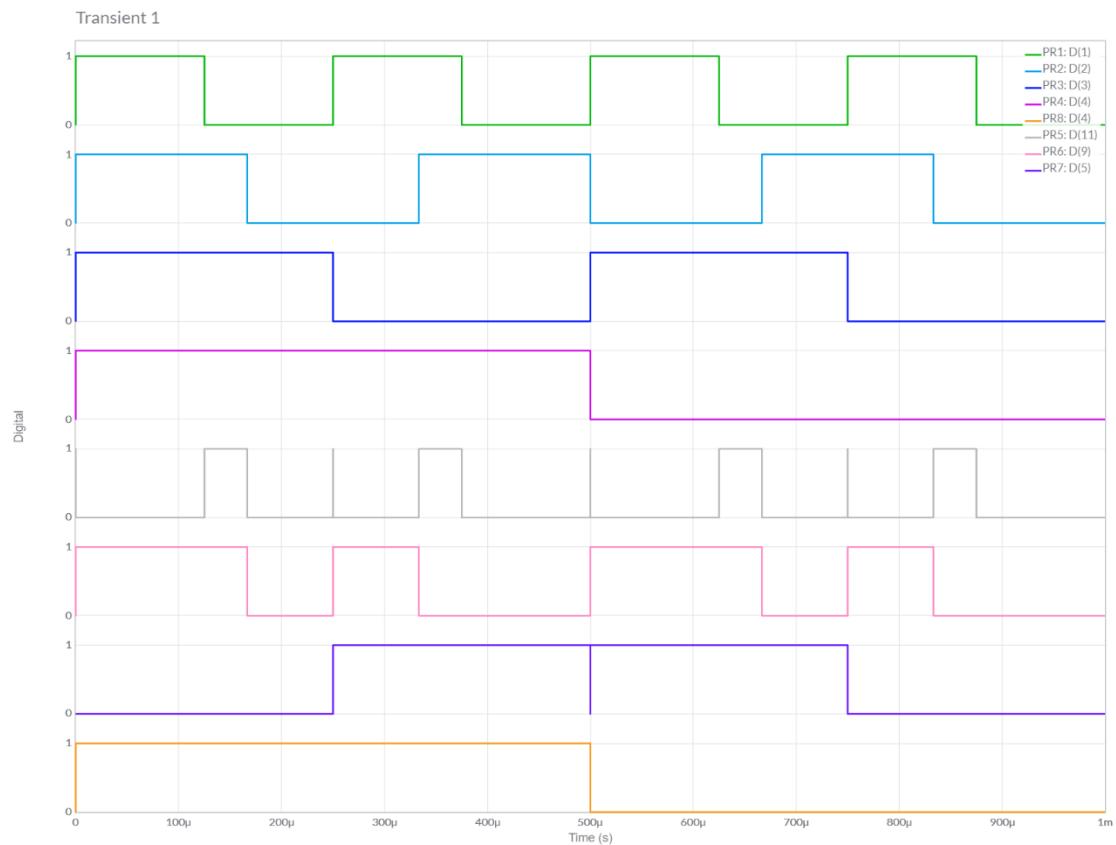


**GRAY TO BINARY CODE
4 - BIT**

CIRCUIT DAIGRAM:



GRAPH:

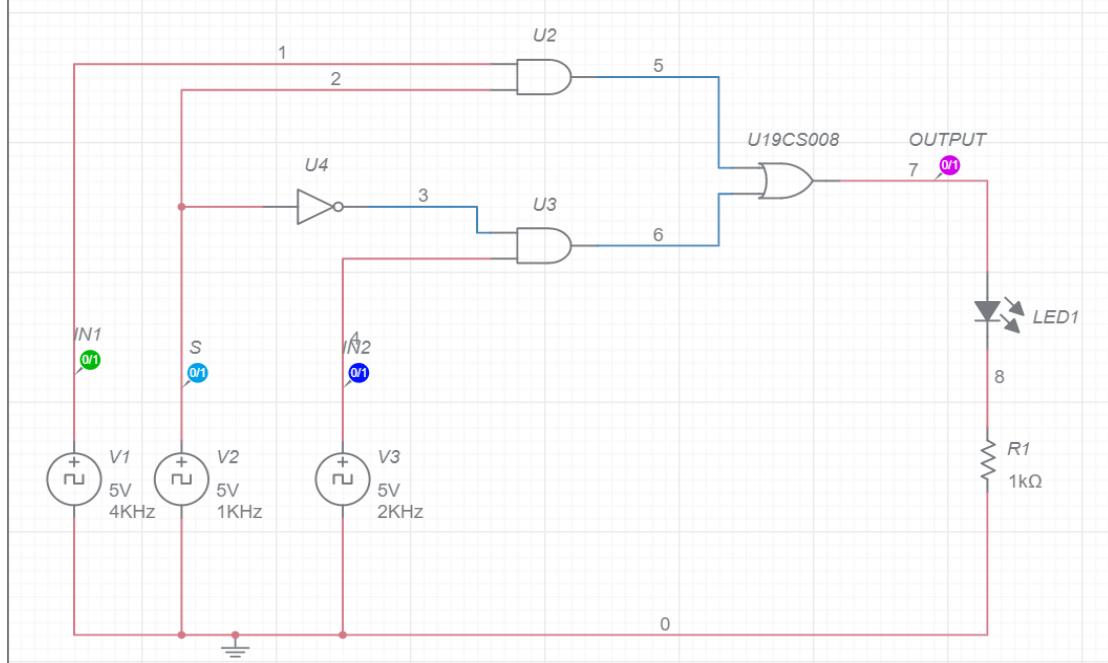




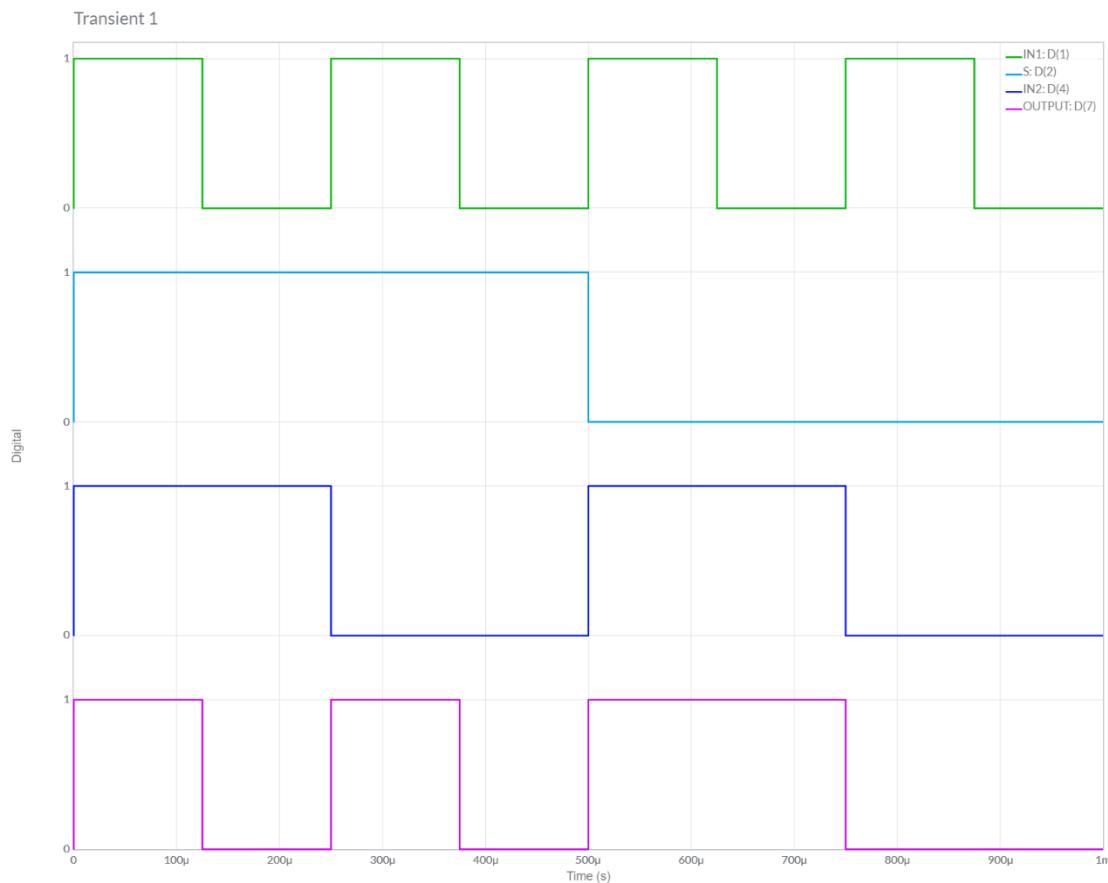
MULTIPLEXER USING BASIC GATES

2*1 MULTIPLEXER

CIRCUIT DAIGRAM:



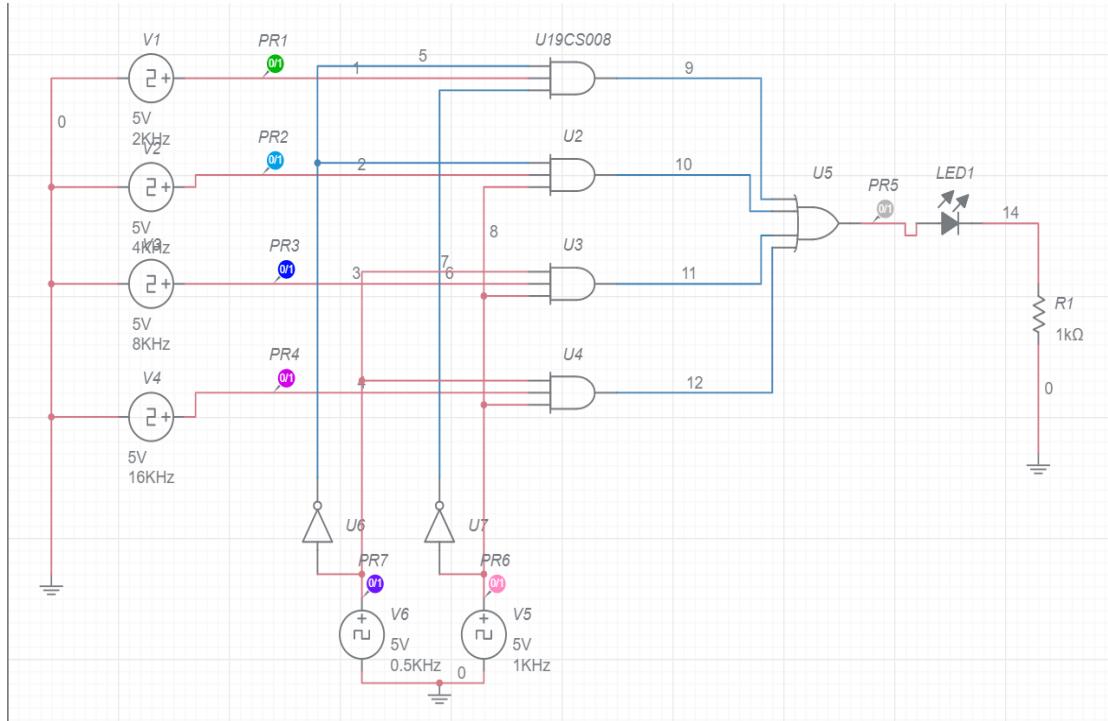
GRAPH:



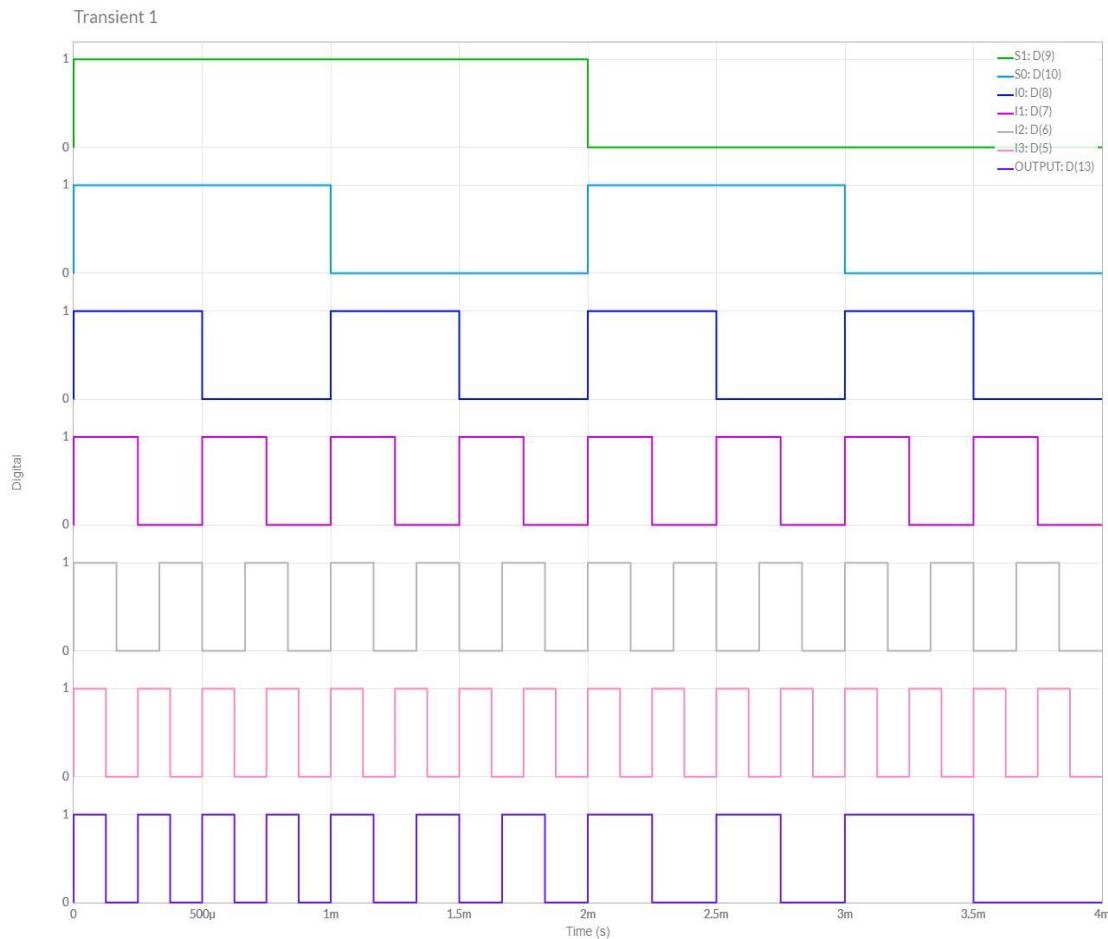


4*1 MULTIPLEXER

CIRCUIT DAIGRAM:



GRAPH:



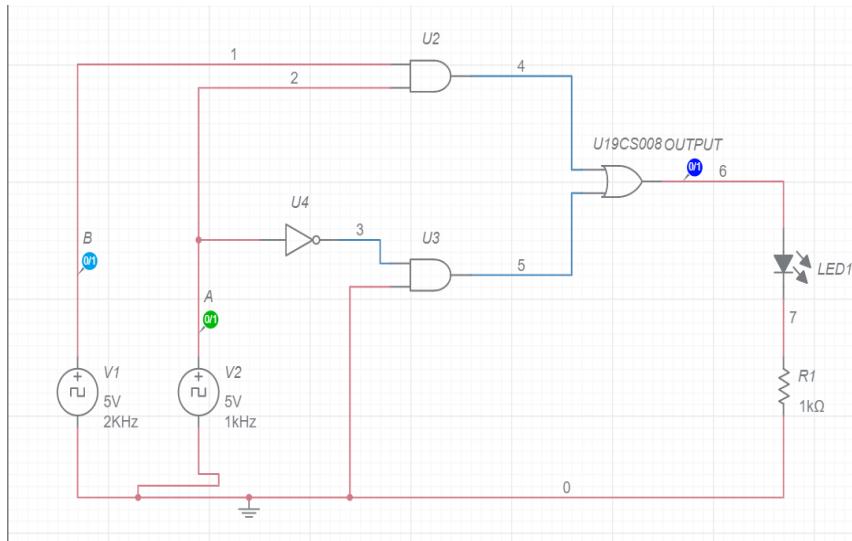


3.) Realise all the basic gates using 2x1 Multiplexer

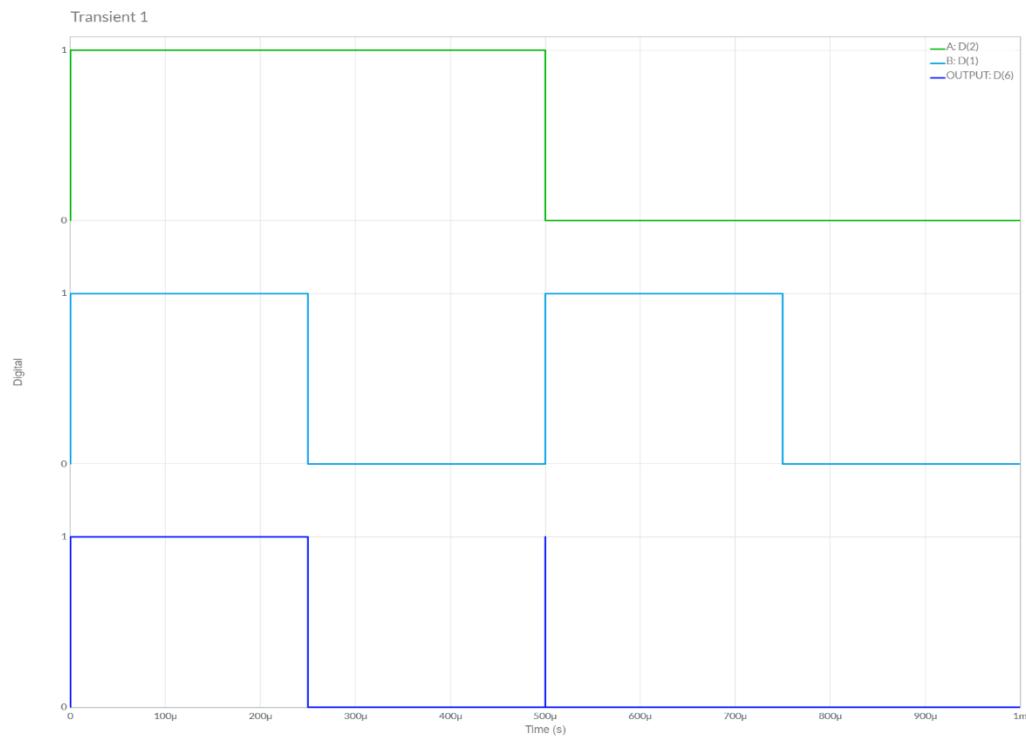
A.) AND GATE

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

CIRCUIT DAIGRAM:



GRAPH:

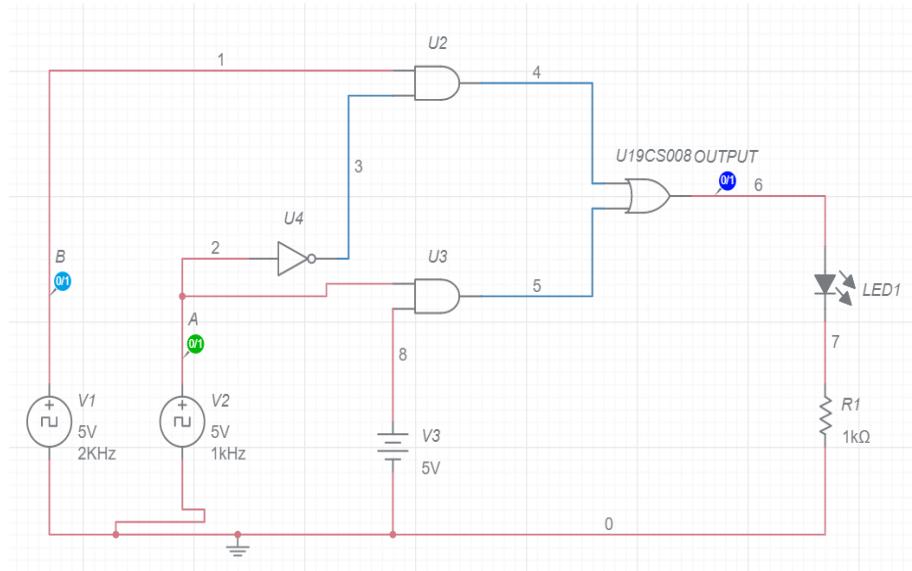




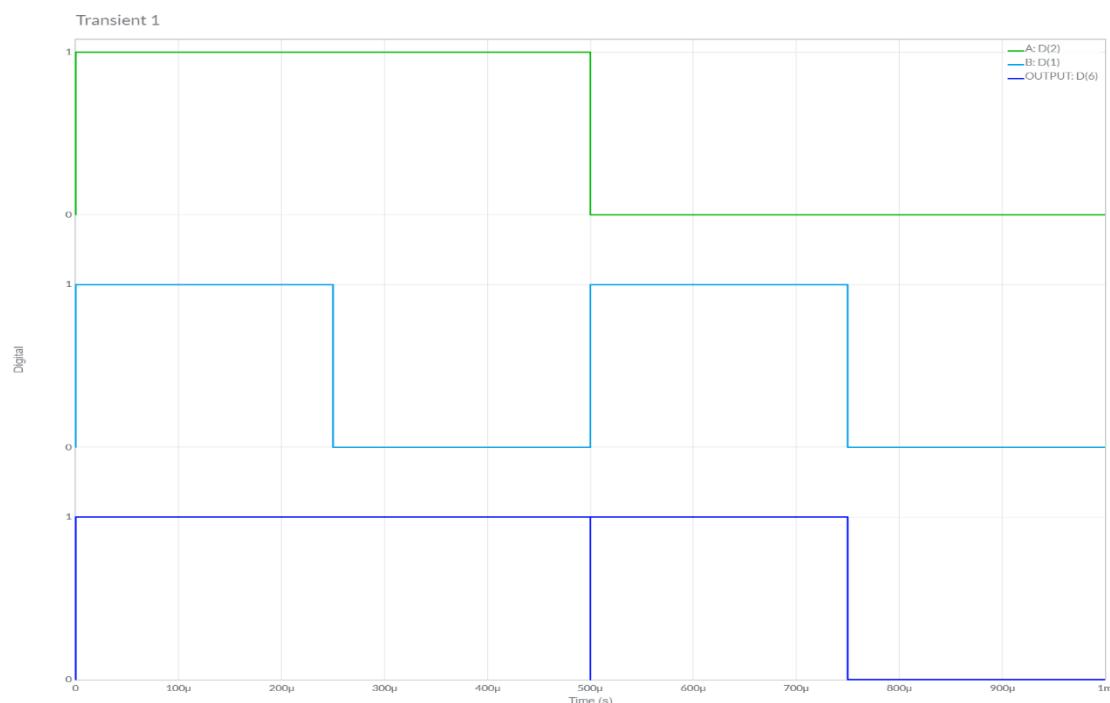
B.) OR GATE

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

CIRCUIT DAIGRAM:



GRAPH:

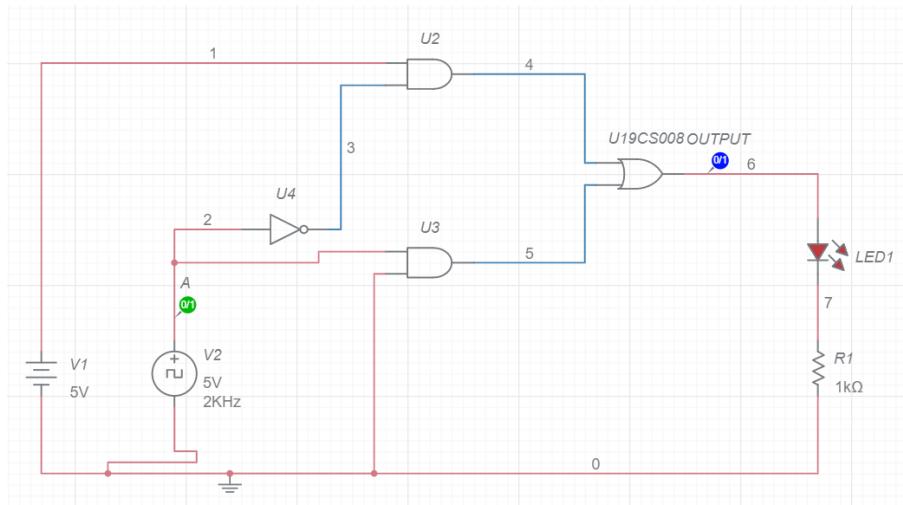




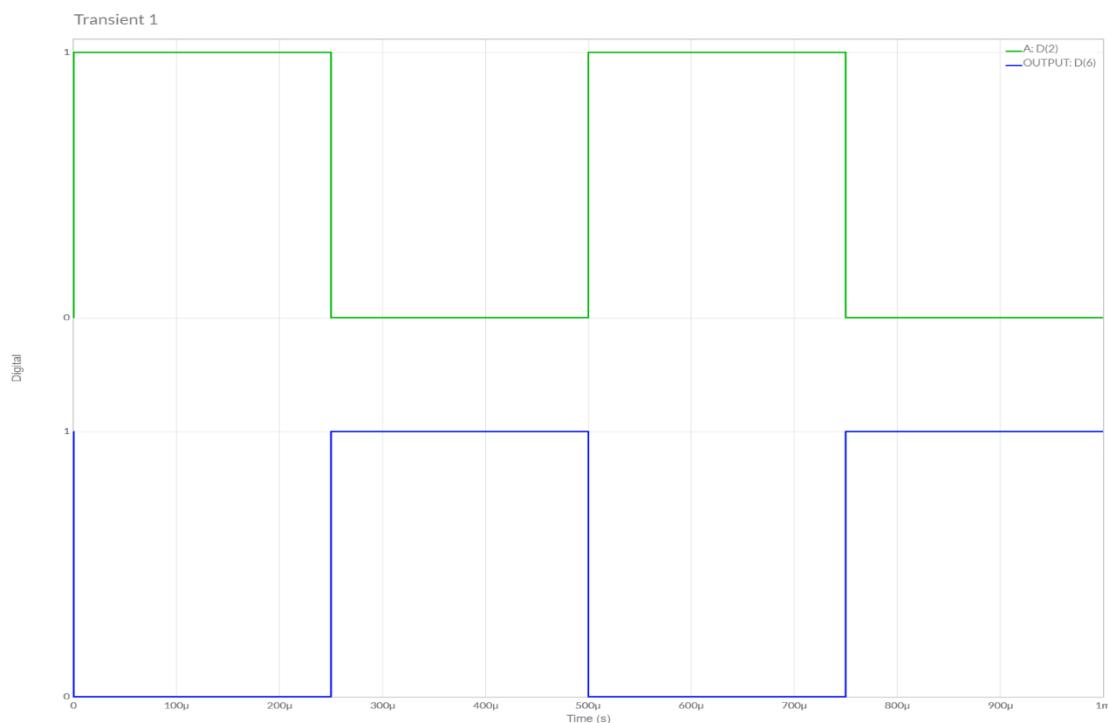
C.) NOT GATE

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

CIRCUIT DAIGRAM:



GRAPH:

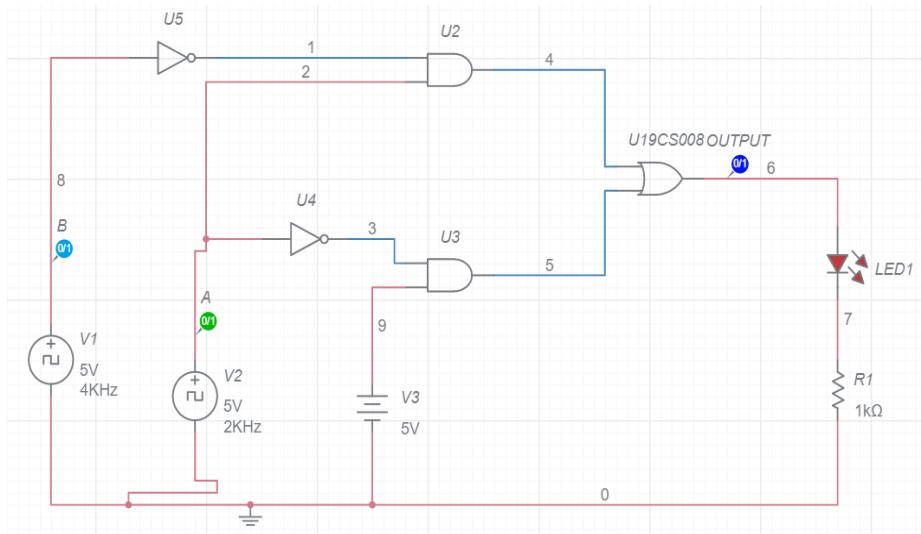




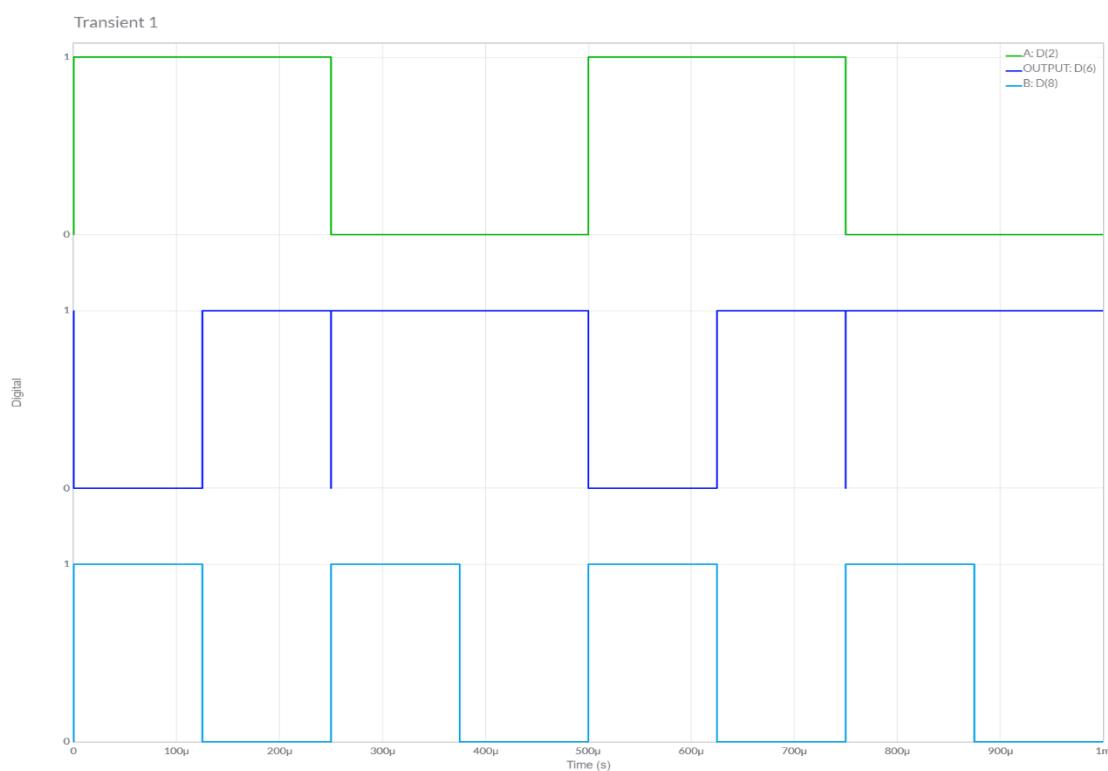
D.) NAND GATE

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

CIRCUIT DAIGRAM:



GRAPH:

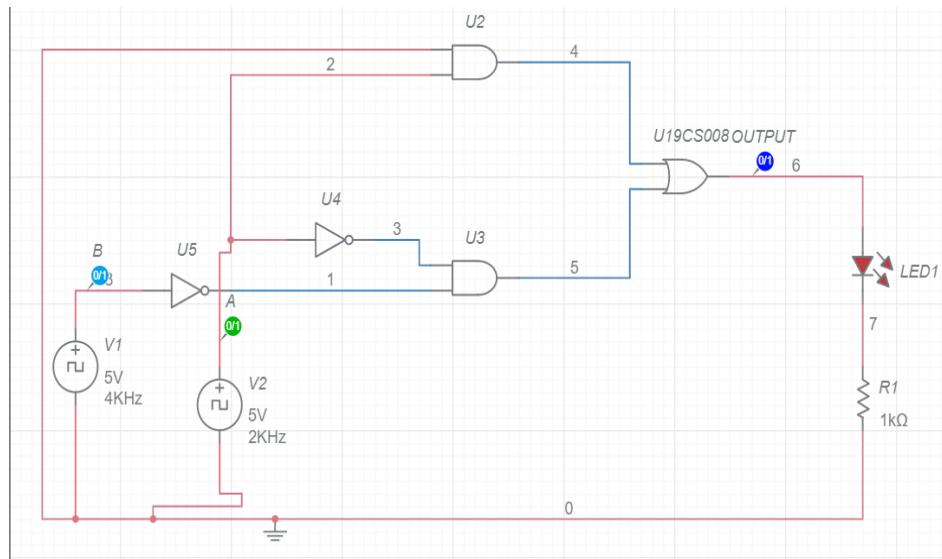




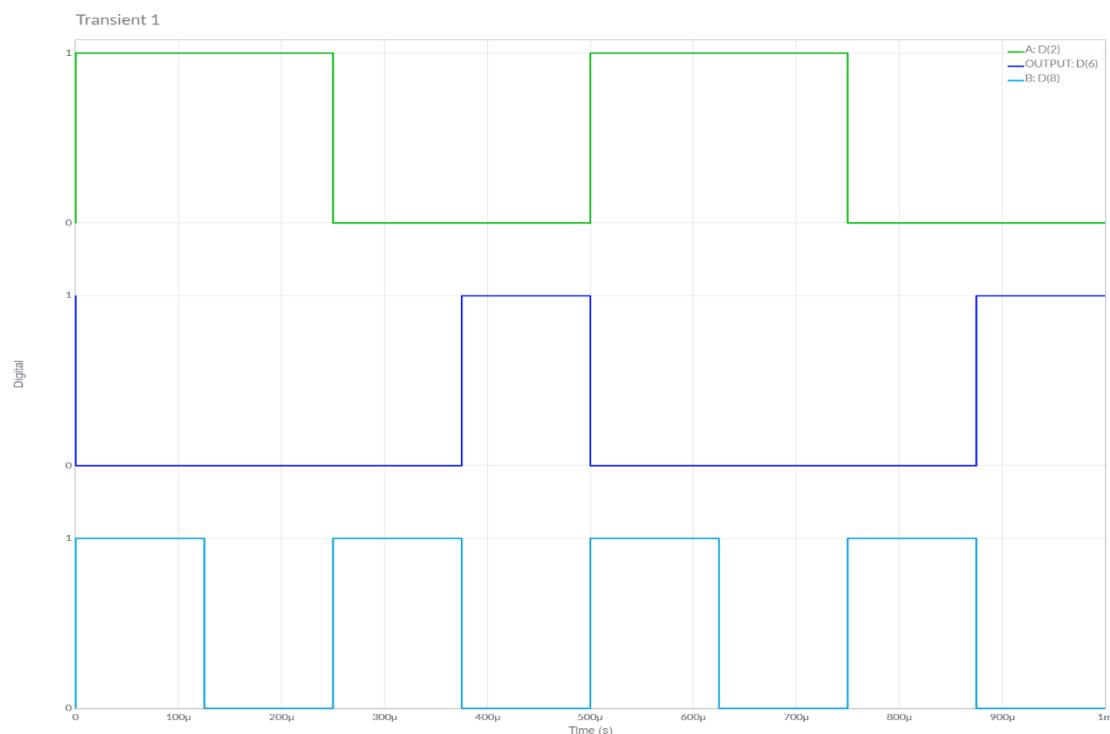
E.) NOR GATE

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

CIRCUIT DAIGRAM:



GRAPH:

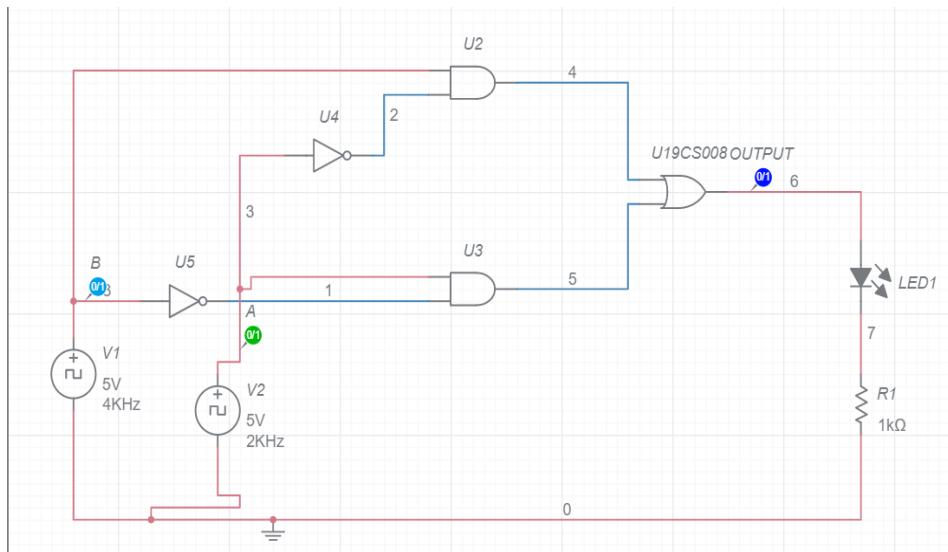




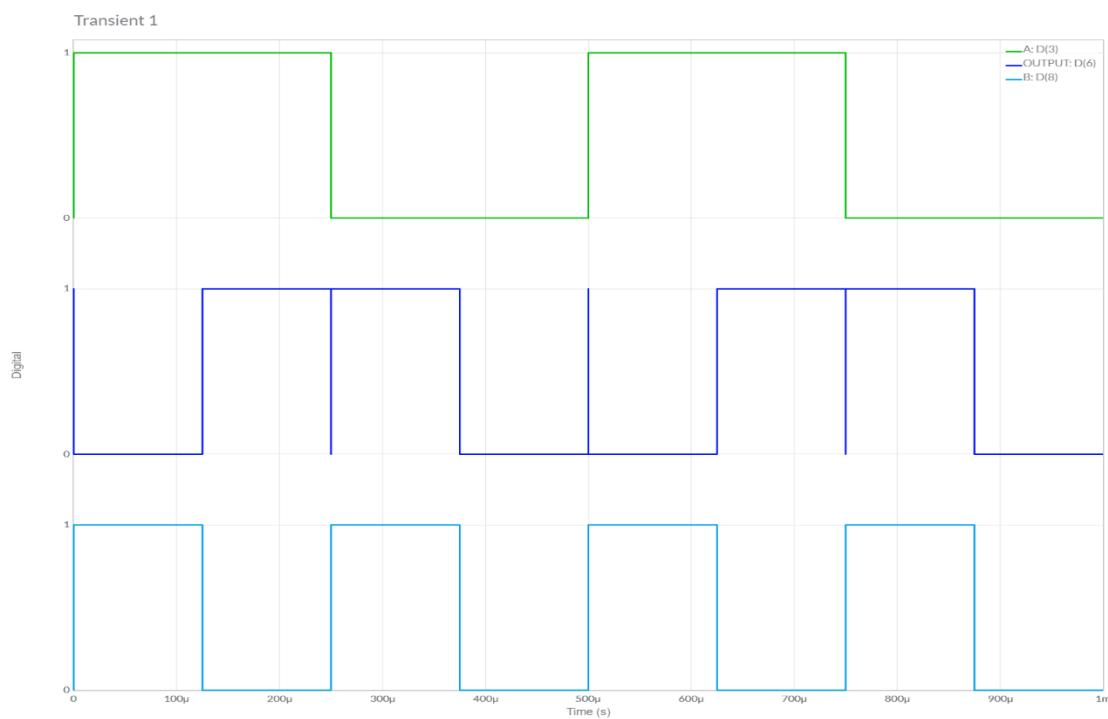
F) XOR GATE

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

CIRCUIT DAIGRAM:



GRAPH:

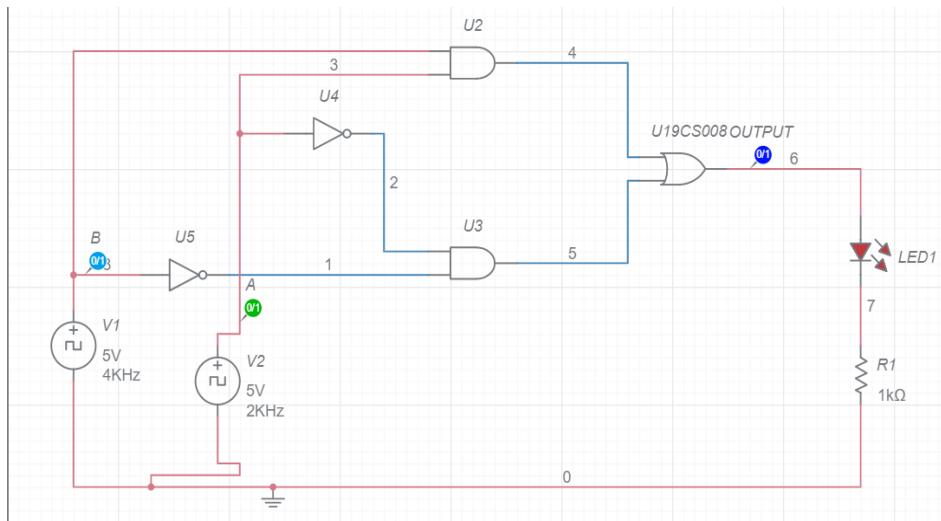




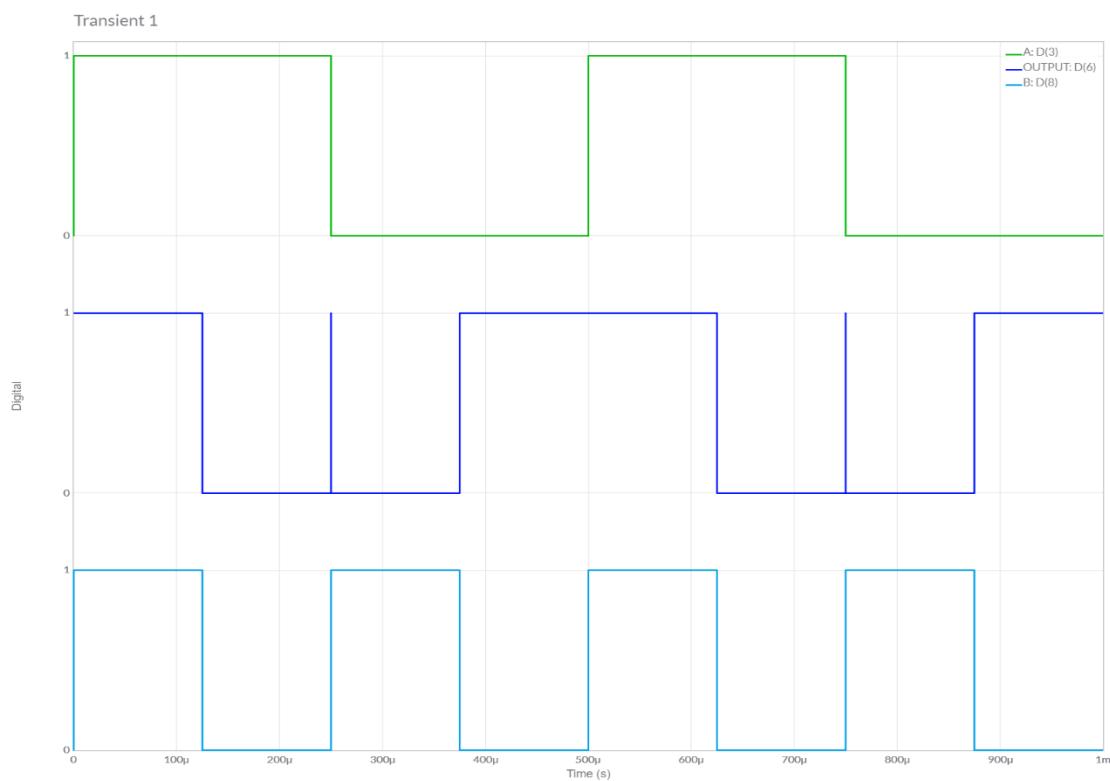
G.) XNOR GATE

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

CIRCUIT DAIGRAM:



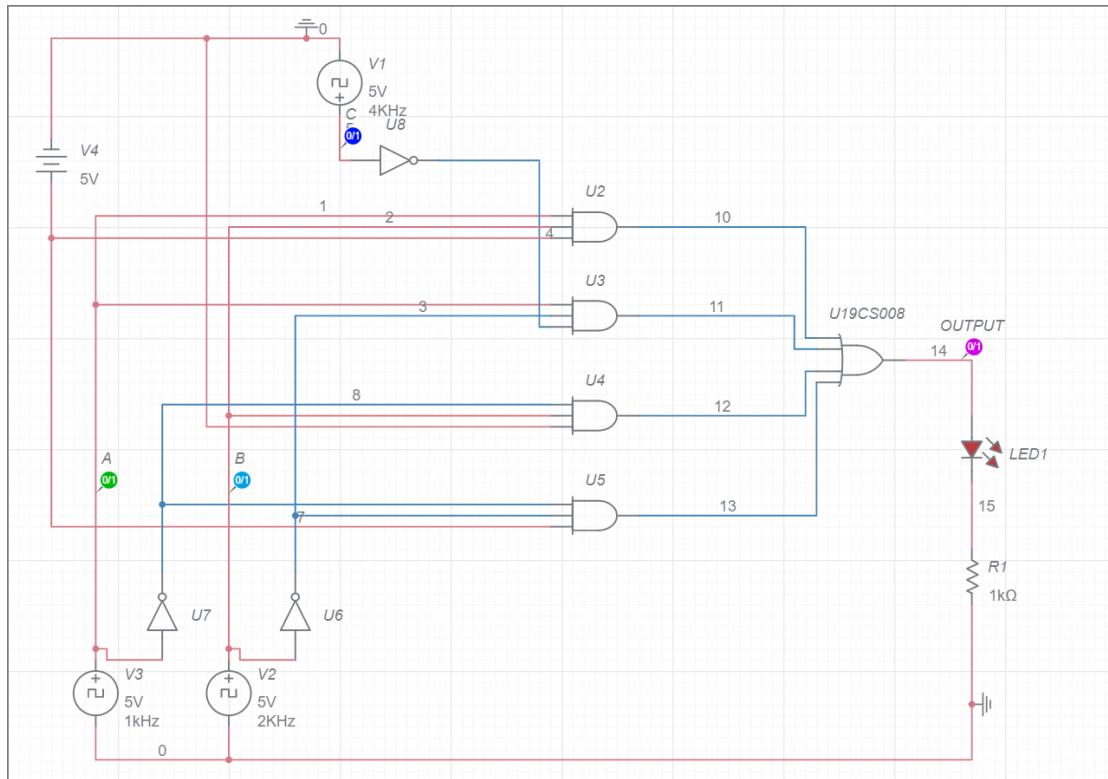
GRAPH:



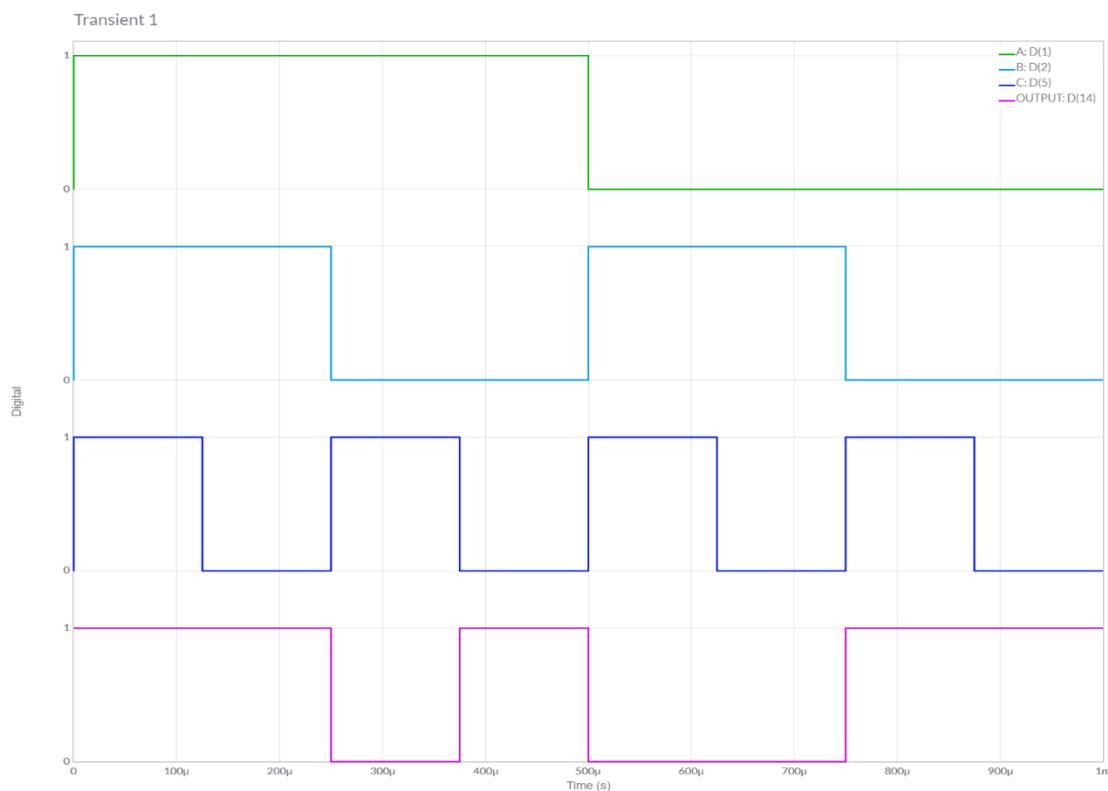


4.) Function Implementation using Multiplexers 1.) $f(A,B,C) = m(0,1,4,6,7)$

CIRCUIT DAIGRAM:



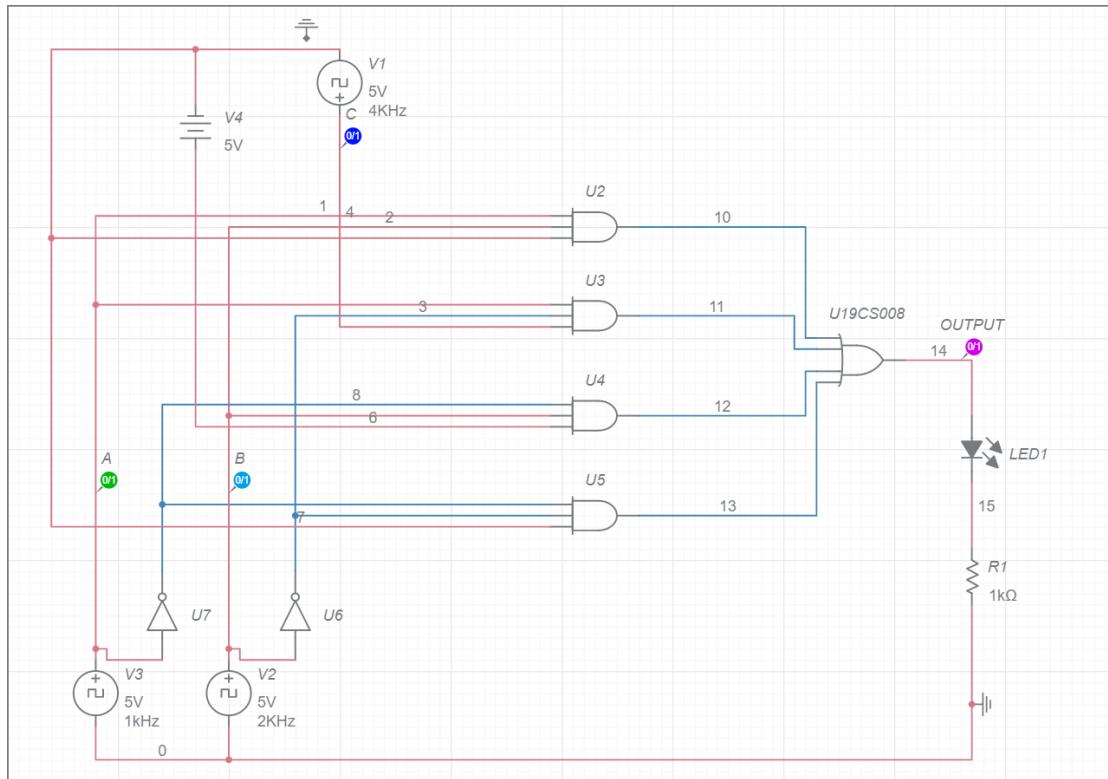
GRAPH:



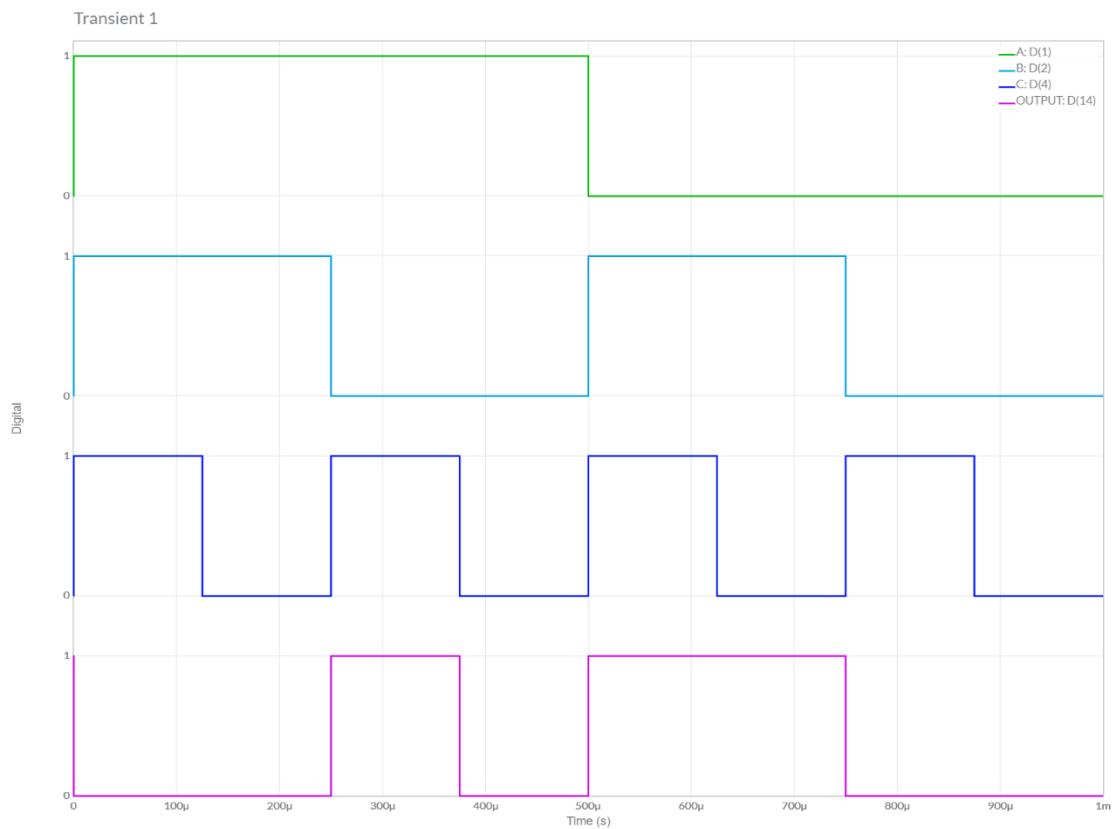


2.) $F(A,B,C)=M(0,1,4,6,7)$

CIRCUIT DAIGRAM:



GRAPH:

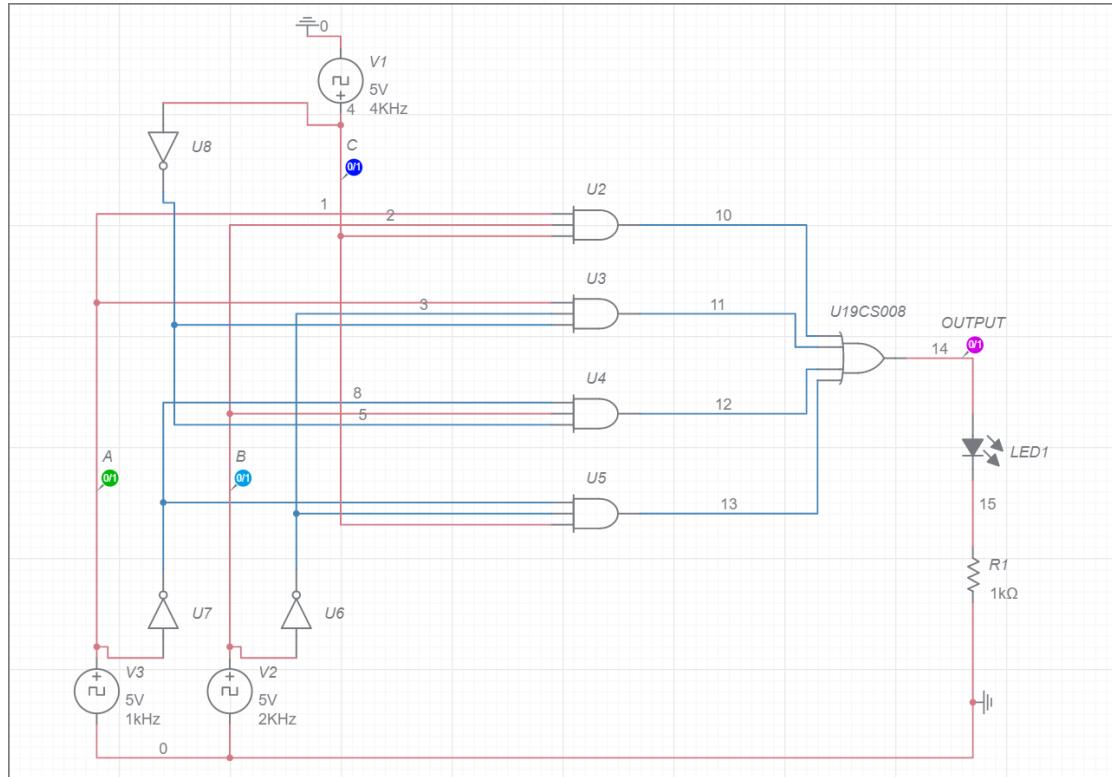




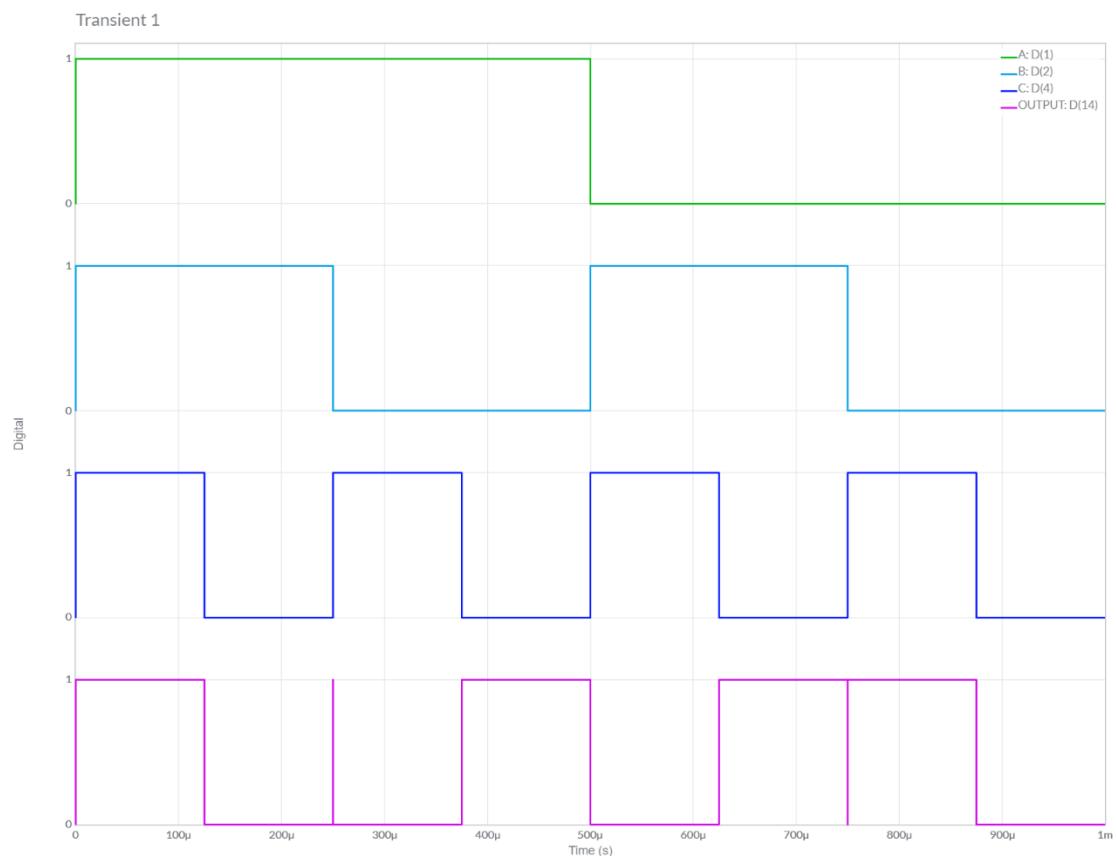
3.) Full Adder Circuit

SUM:

CIRCUIT DIAGRAM:



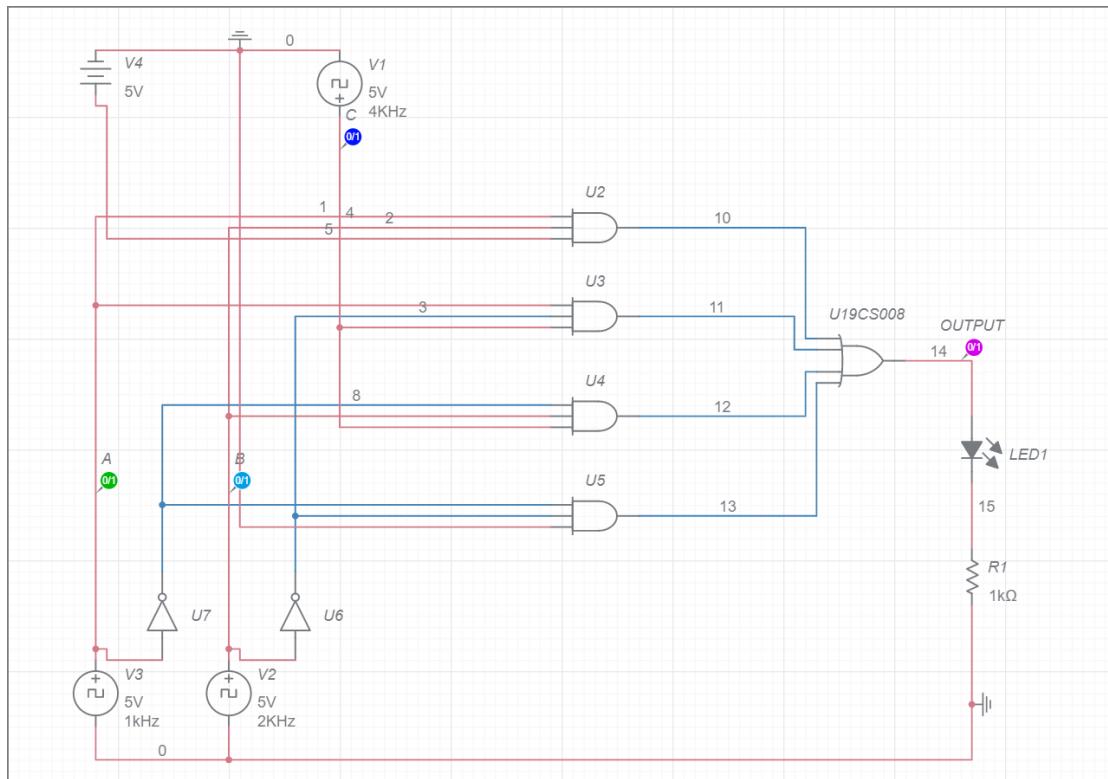
GRAPH:



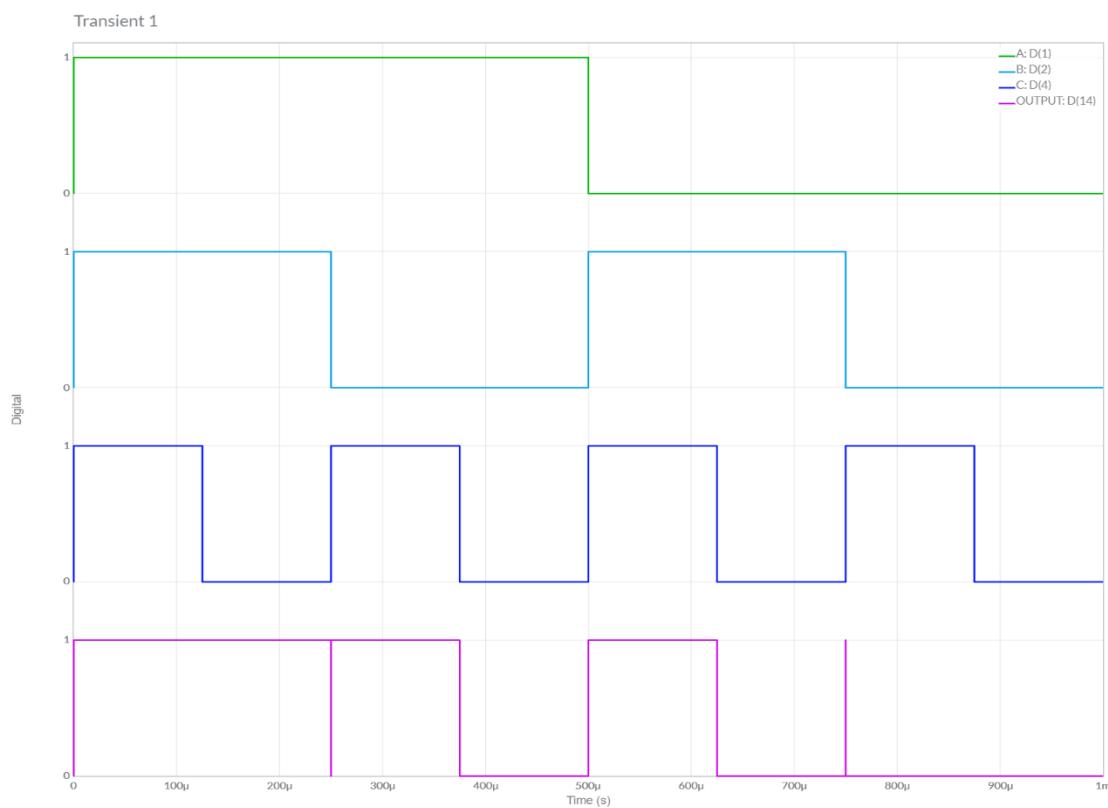


CARRY:

CIRCUIT DIAGRAM:



GRAPH:





CONCLUSIONS:

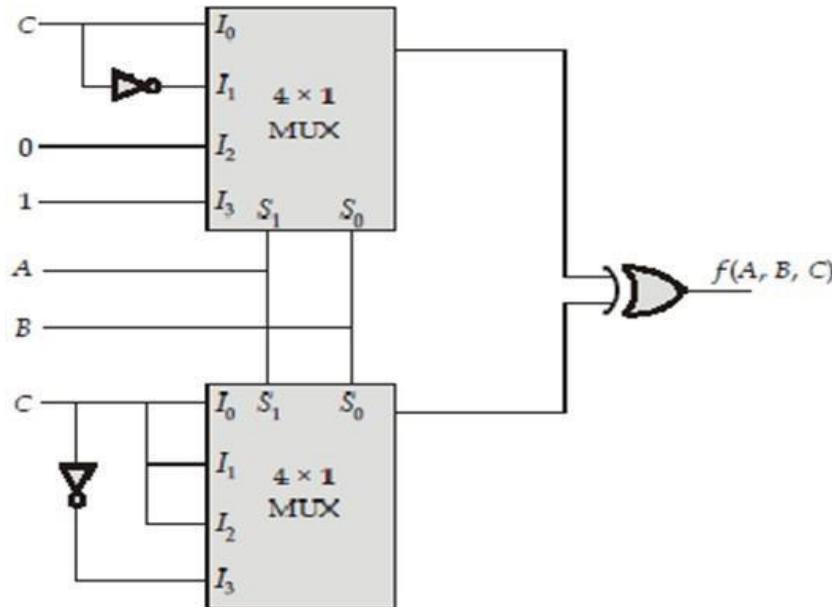
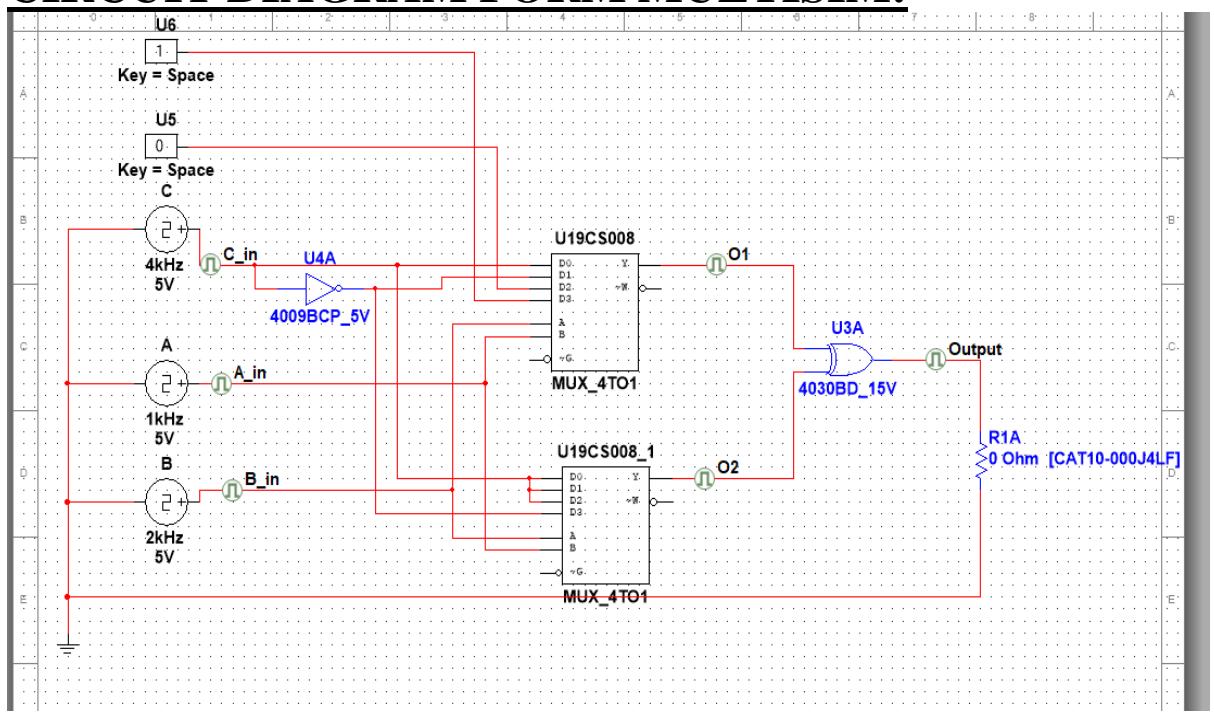
SUCCESSFULLY DESIGNED AND IMPLEMENTED

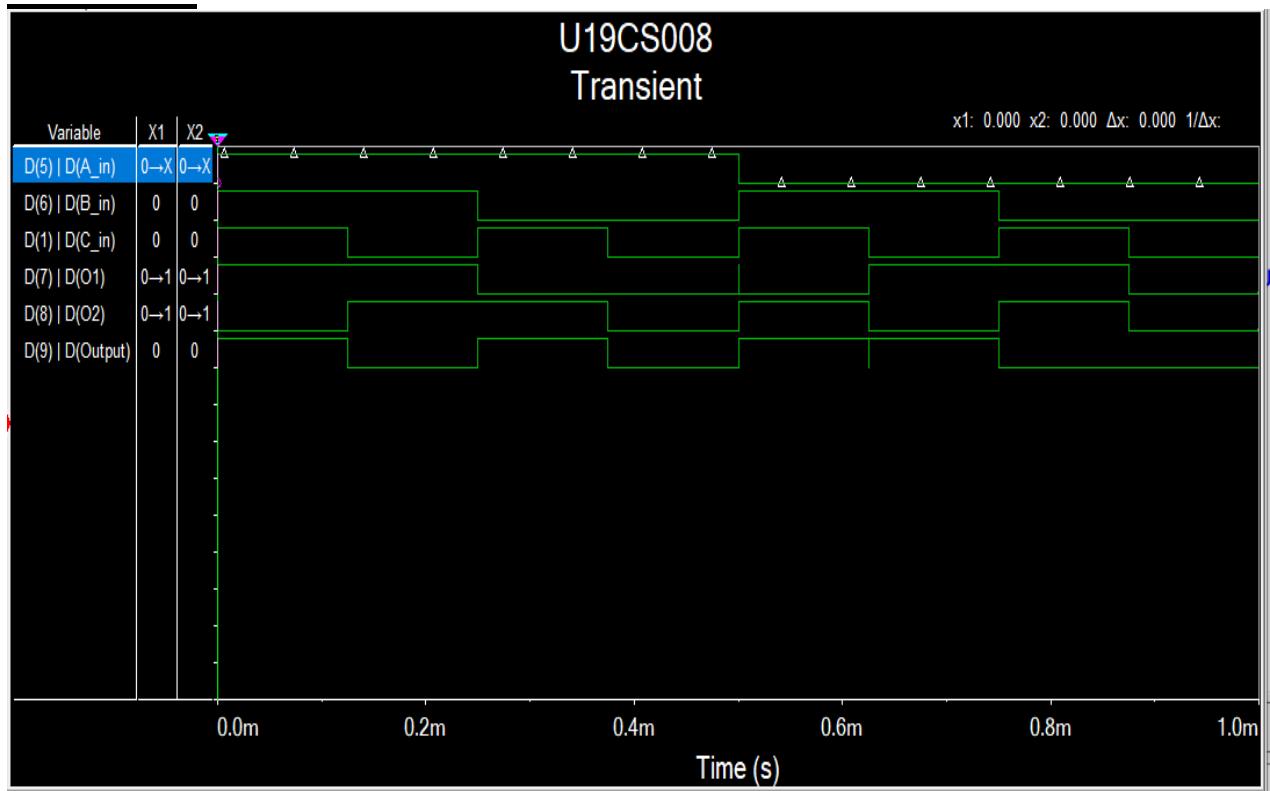
1. Binary to Gray and Gray to Binary Code Converter (2-Bit, 3-Bit and 4-Bit)
2. Multiplexer using basic Gates (2x1 and 4x1)
3. Realise all the basic gates using 2x1 Multiplexer
4. Function Implementation using Multiplexers

We obtained the desired output which was theoretically calculated. The experiment was successfully performed on Multi-Sim. The simulations were verified visually via observing the graphs and matching it with Truth Table.

**DLED ASSIGNMENT-12****NAME : KRINA PATEL****ADMISSION NUMBER: U19CS008**

1. Solve for output Function/Functions. Also verify the same using Multisim.

**CIRCUIT DIAGRAM FORM MULTISIM:**

**GRAPH:**

THEORITICAL:

U19CS008 .
DLED Practical - 12.

~~Q-1~~

A	B	C	O ₁	O ₂	F = O ₁ + O ₂
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	1	0	1
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	0	1

F	A	$\cancel{B^c}$	$B^c c'$	$B^c c$	B^c	$B^c c'$
A'				1	1	
A			1	1		

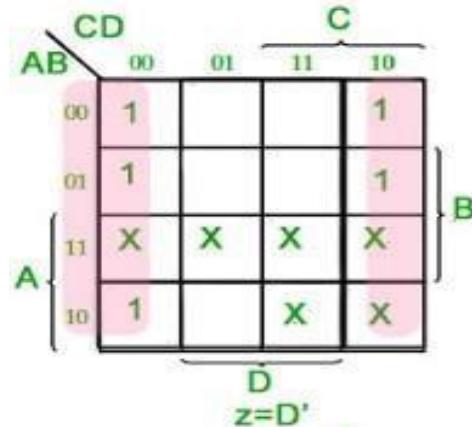
$F = A'B + AC$

Hence, theoretical and practical values obtained from multisim are identical.



2.Design, implement and verify using Multisim:

Batch - A : BCD to Excess – 3 Code Converter



BCD(8421)				Excess-3				
A	B	C	D	w	x	y	z	
0	0	0	0	0	0	1	1	
0	0	0	1	0	1	0	0	
0	0	1	0	0	1	0	1	
0	0	1	1	0	1	1	0	
0	1	0	0	0	1	1	1	
0	1	0	1	1	0	0	0	
0	1	1	0	1	0	0	1	
0	1	1	1	1	0	1	0	
1	0	0	0	1	0	1	1	
1	0	0	1	1	1	0	0	
1	0	1	0	X	X	X	X	
1	0	1	1	X	X	X	X	
1	1	0	0	X	X	X	X	
1	1	0	1	X	X	X	X	
1	1	1	0	X	X	X	X	
1	1	1	1	X	X	X	X	

THEORITICAL:

U19CS008

$g = 2$ BCD to excess-3 code

	$c'd'$	$c'd$	cd'	cd
$A'B'$	1	1	1	1
$A'B$	0	1	1	1
AB	\times	\times	\times	\times
AB'	1	1	\times	\times

W = $A + BD + BC$
 $= A + B(C + D)$

X

	$c'd'$	$c'd$	cd'	cd
$A'B'$	1	1	1	1
$A'B$	1			
AB	\times	\times	\times	\times
AB'	1	\times	\times	

$x = Bc'd' + b'd + b'c$
 $= b'(c + d) + bc'$

y

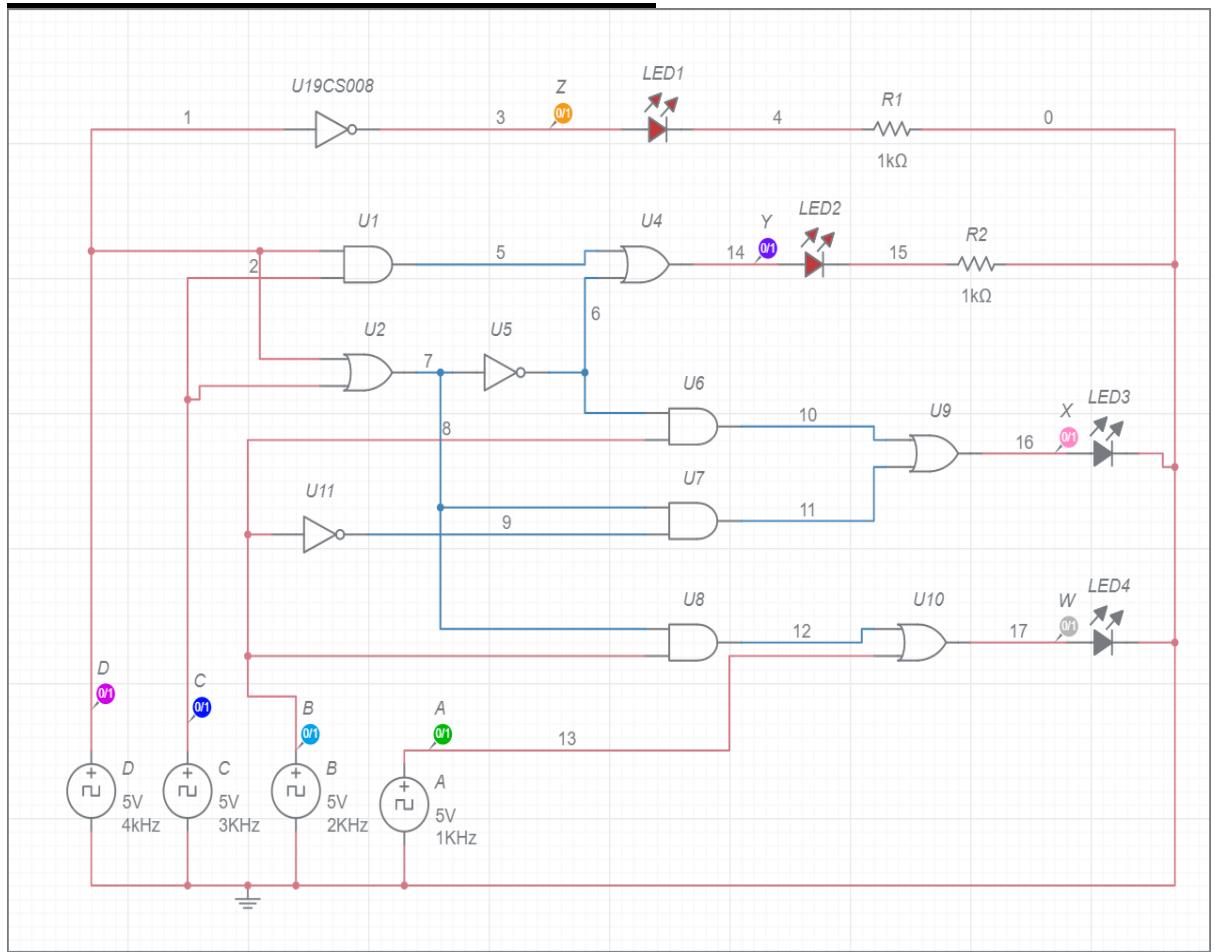
	$c'd'$	$c'd$	cd'	cd
$A'B'$	1		1	
$A'B$	1		1	\times
AB	\times	\times	\times	\times
AB'	1		\times	\times

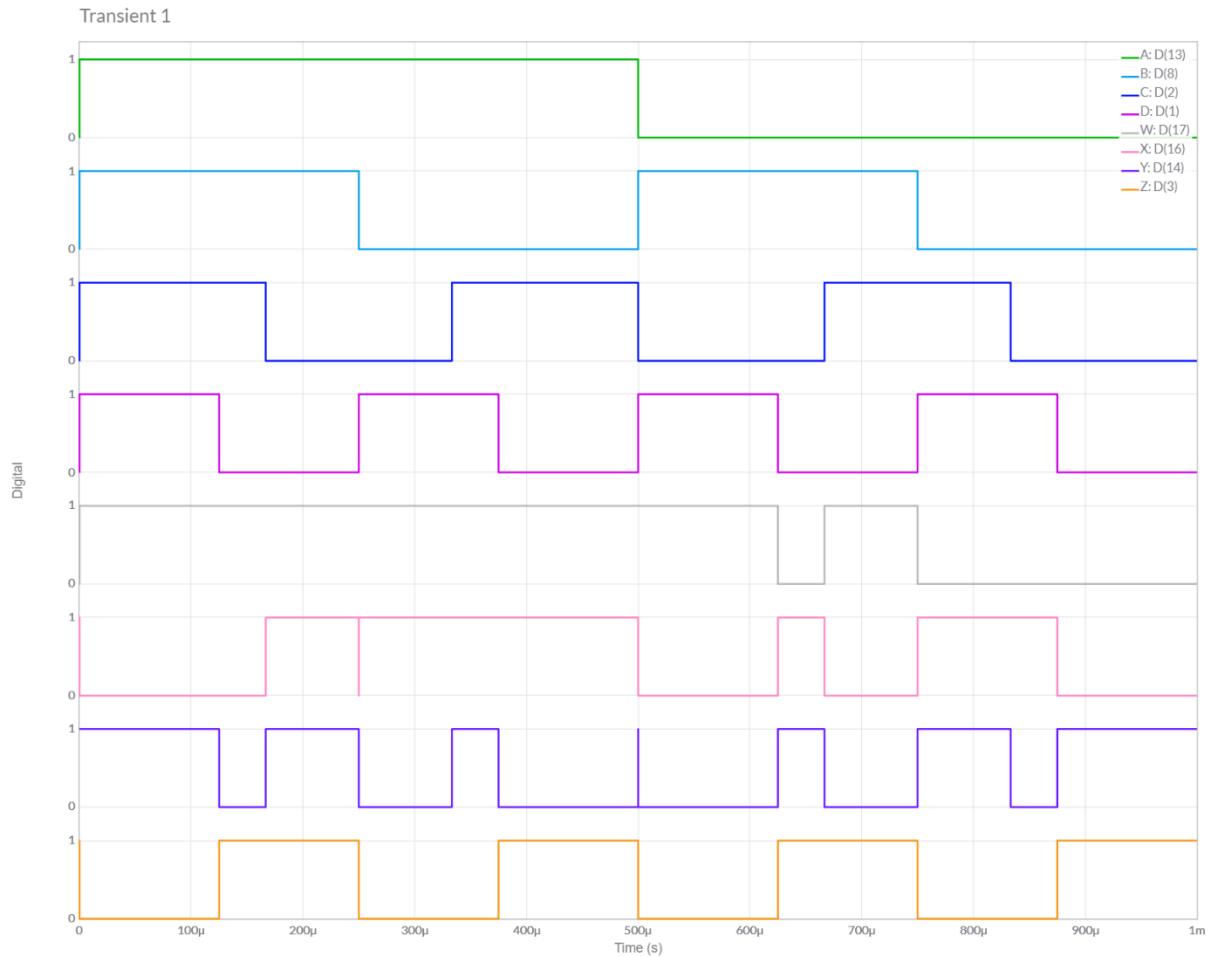
$y = c'd' + cd$
 $= \overline{c \oplus d}$
 $= (c \oplus d)'$



CD	X	X	1	$'AA$
$A'B$	$c'b'$	$c'b$	cd	cd'
$A'b'$	1	1	1	$c'd$
$A'b$	1	1	1	$'d'b = 'd'$
AB	x	x	x	$'AA$
$A'B$	1	x	x	x

CIRCUIT FROM MULTISIM:



**GRAPH:**

Hence , Theoretical and Practical Values Obtained from Multisim Are Identical.



Expt. No:

13

Date:

26/11/2020

High Pass and Low Pass Filters

AIM: To study, design and implement:

1. Passive RC – High Pass Filter
2. Passive RC – Low Pass Filter
3. Observe the Working of Low Pass Filter as an Integrator
4. Observe the working of High Pass Filter as a Differentiator

SOFTWARE TOOLS / OTHER REQUIREMENTS:

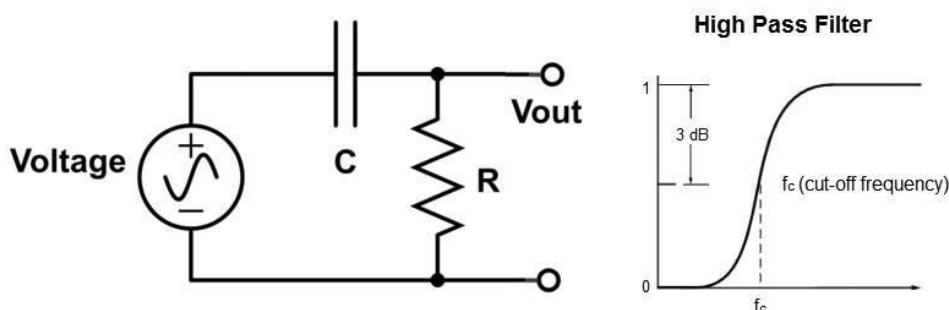
1. Multisim Simulator/Circuit Simulator

THEORY:

RC – High Pass Filter:

High pass filter is a circuit which passes only High frequency signals and rejects low frequency signals. The below figure shows the diagram of High pass RC circuit.

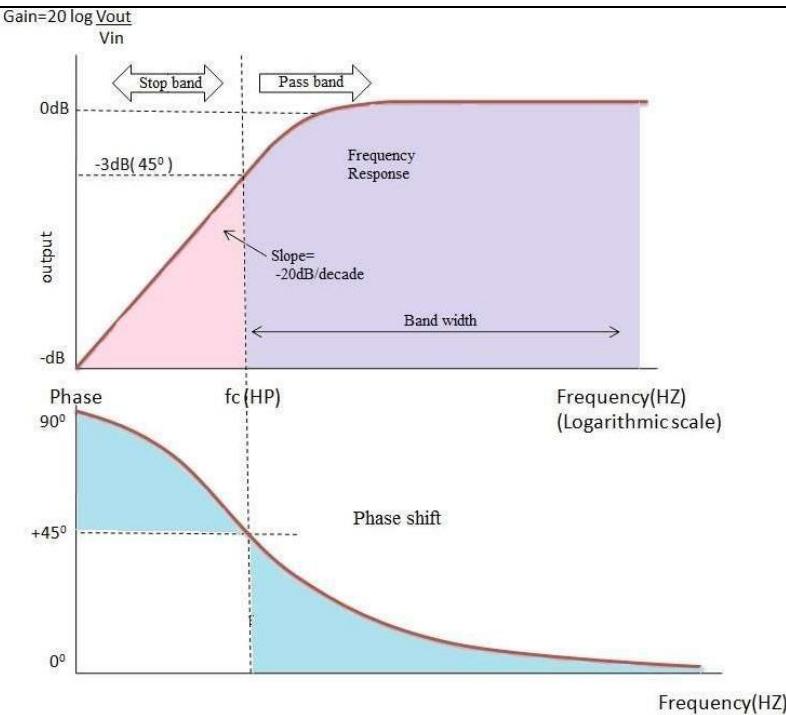
At Zero frequency the capacitor offers high amount of reactance hence as the capacitor is at the input it acts as a blocking capacitor. This circuit acts as capacitive coupling circuit and provides dc isolation between input and output. At high frequency capacitor offers less amount of reactance and a small amount of signal appears across the capacitor. Hence almost all the input applied appears across the output i.e. gain is unity.



High Pass Filter- Circuit and Frequency Response

Frequency Response of High Pass Filter

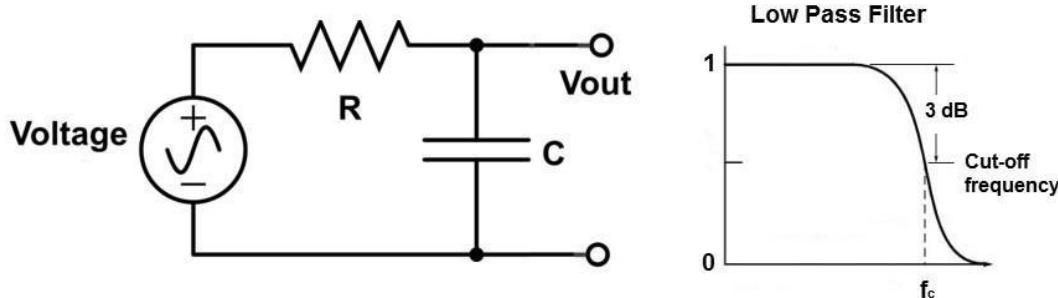
The response curves with respect to frequency and the capacitive reactance are given below



This response curve shows that the high pass filter is exactly opposite to the low pass filter. In high pass filter till the cut off frequency all the low frequency signals are blocked by the capacitor resulting in the decrease of output voltage. At the cut off frequency point the value of the resistor 'R' and the reactance of the capacitor 'X_c' are equal thus the output voltage increases at a rate of -20 dB/decade and the output signal levels are -3 dB of the input signal levels.

RC – Low Pass Filter:

Low pass filter is a circuit which passes only low frequency signals and attenuates high frequency signals when passed through a network over certain cutoff frequency and it is determined by the RC time constant. The below figure shows the circuit of Low pass RC circuit. Here we are going to discuss about different types of input signals given to the low pass circuit and observe the output response.



**Determination of Cut-off Frequency:**

A simple Low pass RC circuit consists of a resistor and capacitor across the input and capacitor across the output. As we know that a Capacitor exhibits Reactance (opposition to the flow of alternating current). By analyzing the low pass RC circuit at zero frequency the capacitor offers

high amount of reactance almost acts as an open circuit. Now the entire signal passes across the output terminals hence the signal is unattenuated. We can say in terms of frequency response as the gain is unity at zero frequency. Similarly at high frequencies we can say that the capacitor offers less amount of reactance and hence the output decreases as frequency increases. This concept can be clearly understood by considering the following notations.

$$X_c = \frac{1}{2\pi f C}$$

Where X_c =Capacitive reactance measured in ohms, f = frequency in hertz, C = capacitance in farad.

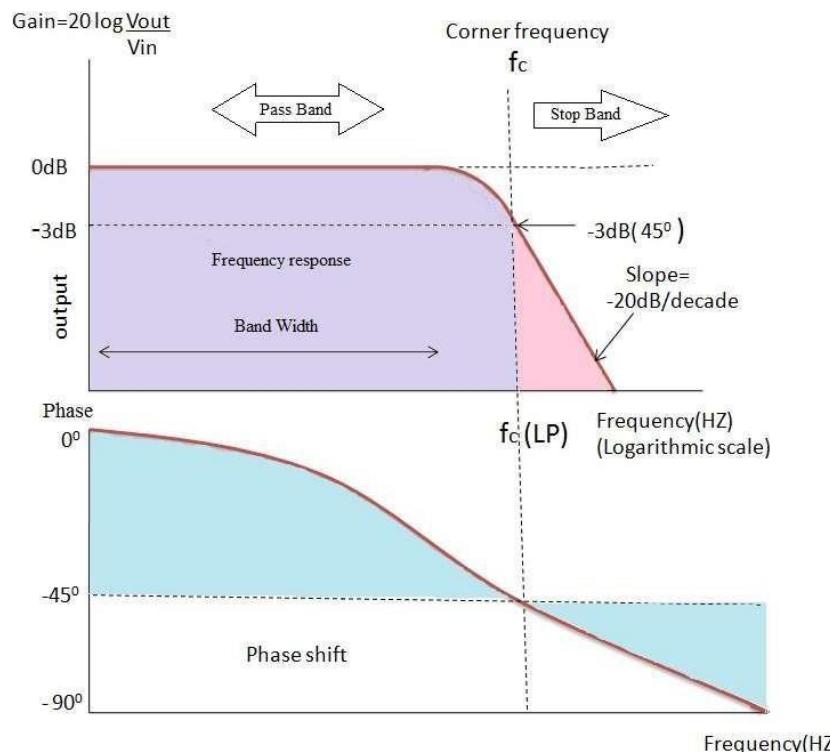
When $f = 0$

$X_c = \infty$ (Open Circuit)

$X_c = 0$ (Short Circuit)

Frequency Response:

It is defined as the plot drawn between magnitude of output voltage versus frequency. It defines the range of frequencies for which the circuit is operated.

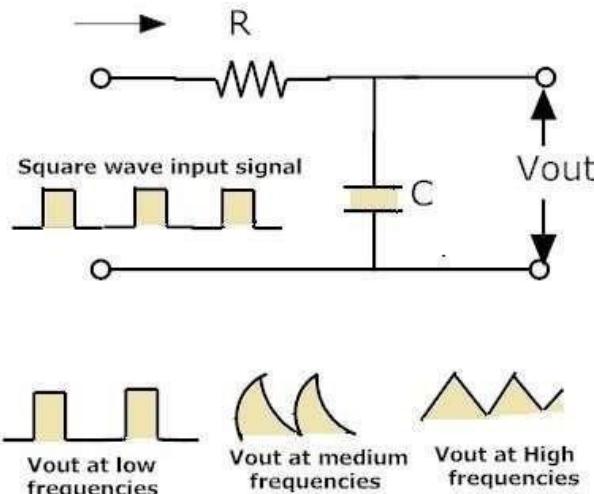


**Low Pass as an Integrator:**

In order to act circuit as an integrator it depends upon the time constant of the RC circuit. The time constant should be large enough when compared with the input signal so that the capacitor charges slowly and all the input voltage which is applied appears across the resistor. In this circuit the output is an integral function of the input voltage. In other words we can say that output is directly proportional to the input signal.

Applications of Integrator:

- To analyse the signals and perform the integration.
- To convert a square wave input into
- a triangular wave To convert a rectangular wave input into a sawtooth wave.

**High Pass as a Differentiator:**

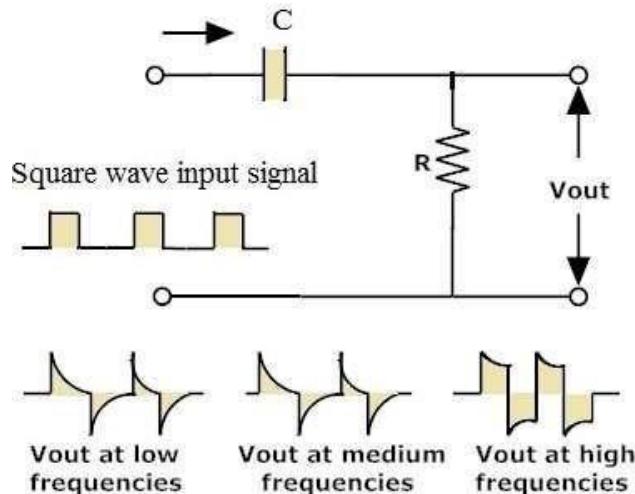
The high pass circuit acts as a differentiator depending upon the time constant of the RC network. It varies depending upon the circuit. The time constant of the circuit should be small so that the capacitor charges quickly and all the voltage is taken by the capacitor and small amount of voltage appears across the resistor. The current in the circuit is decided by the capacitance.

The output equation in terms of capacitance is given by:

$$\begin{aligned}
 V_o &= \frac{q}{C} = \frac{1}{C} \int i \, dt \\
 &= \int \frac{V_i}{C} dt \quad (\because i = \frac{V_i}{R}) \\
 &= \frac{1}{RC} \int V_i \, dt
 \end{aligned}$$

**Applications of Differentiator:**

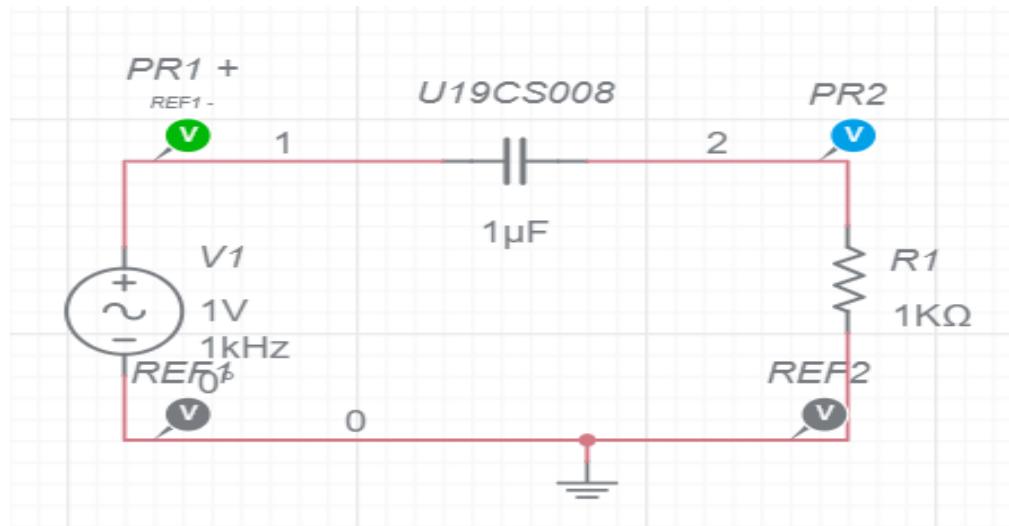
- It converts triangular wave into square wave.
- It converts ramp input into step wave
- Generates spikes from square or triangular waves which are used in for triggering the circuits or for synchronization of clock circuits.



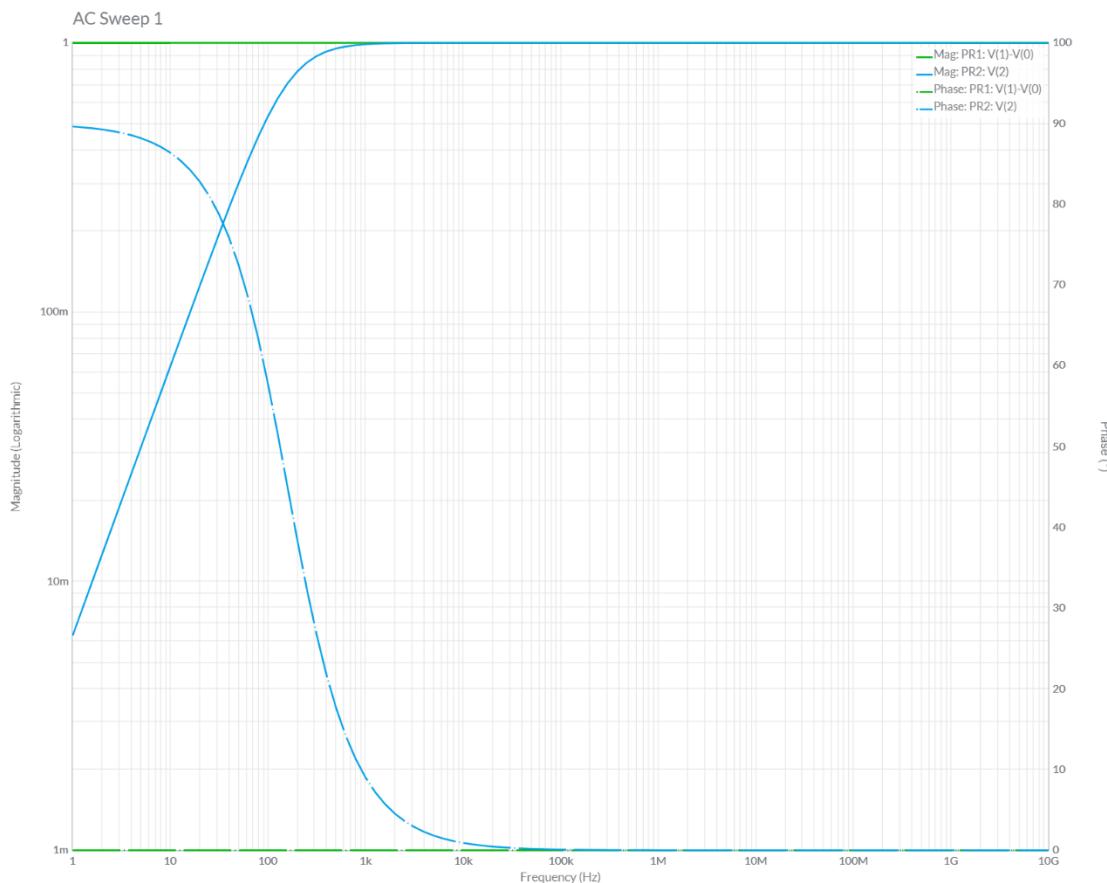


SIMULATION SCREENSHOTS

Circuit Diagram of RC – High Pass Filter:



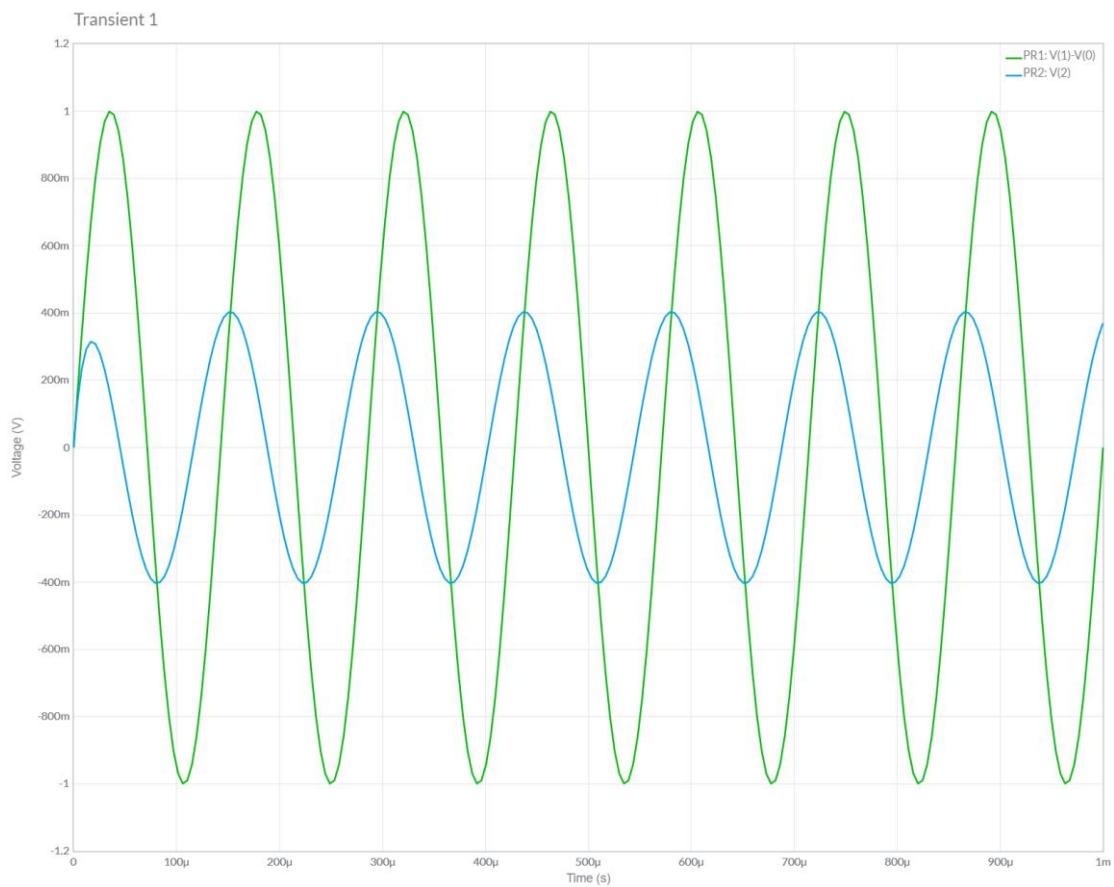
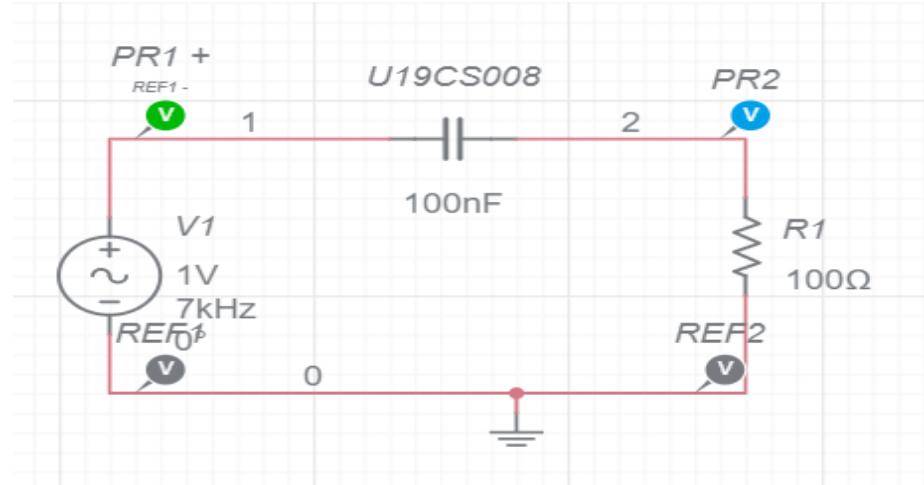
Frequency Response Plot of High Pass Filter:





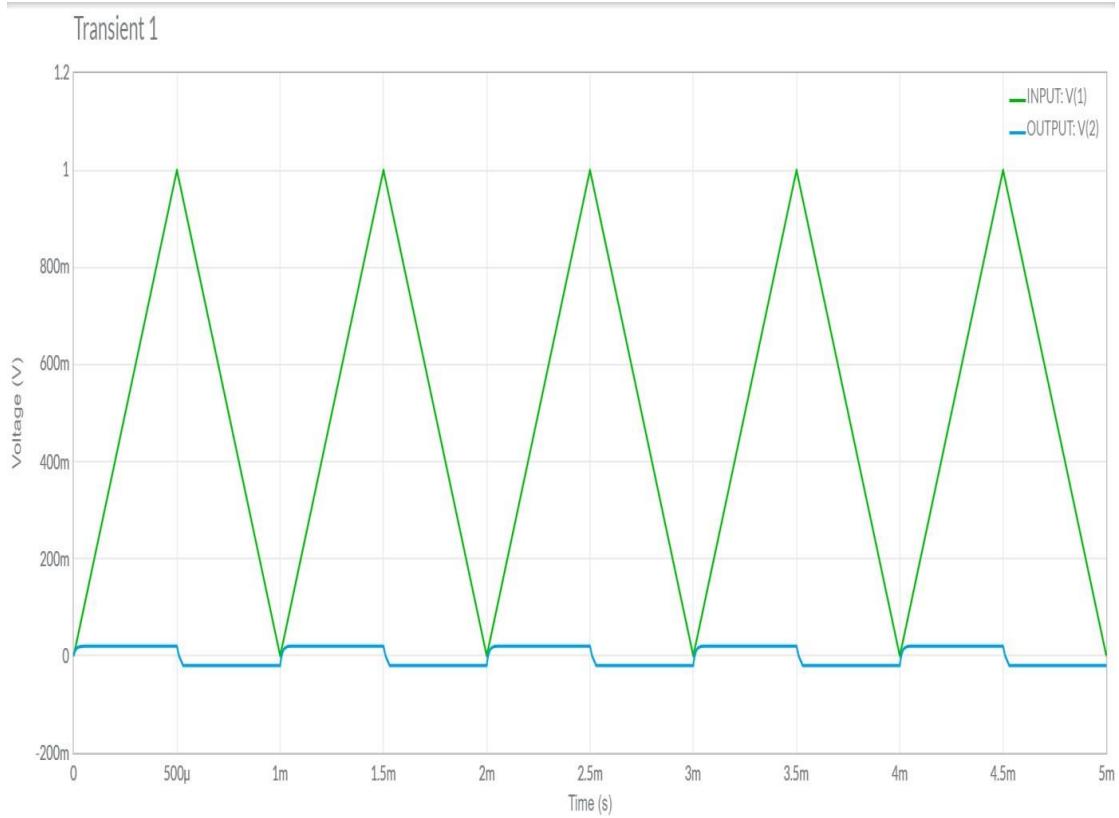
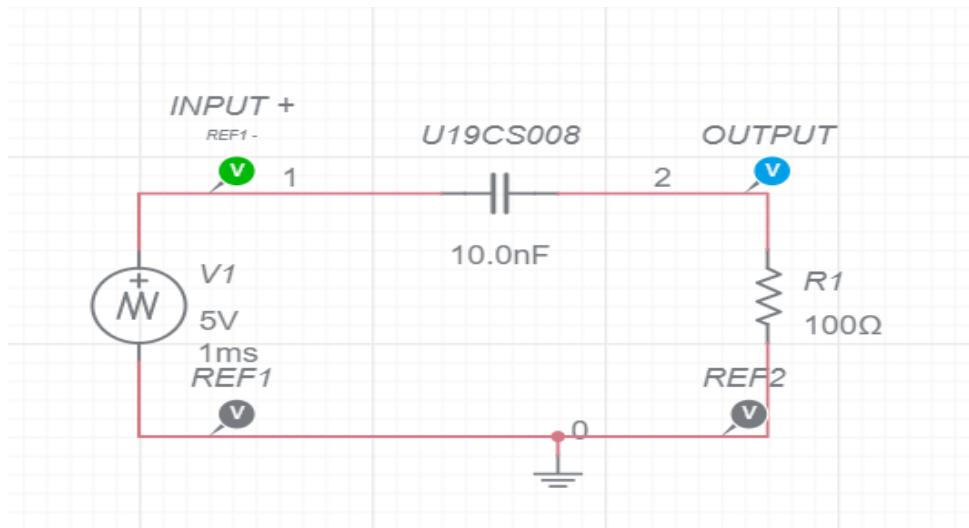
Output of Differentiator:

Input – Sine Wave:



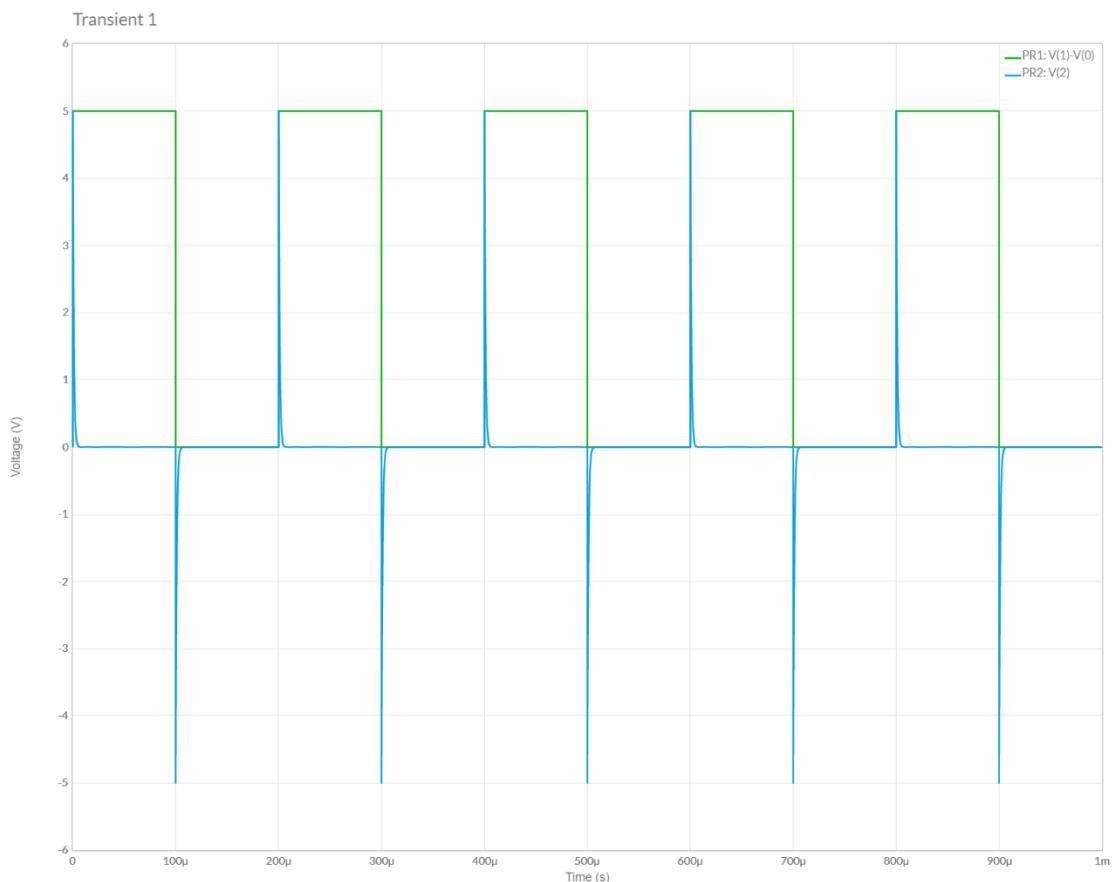
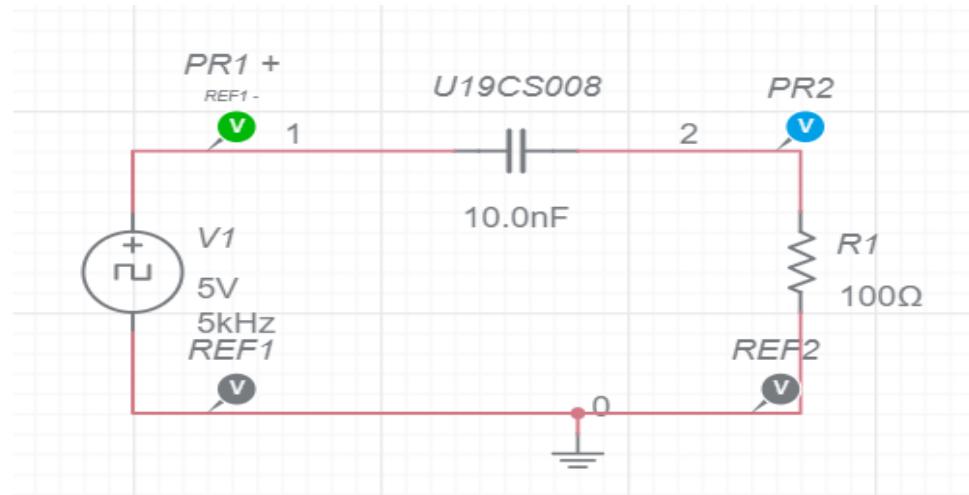


Input – Triangular Wave:



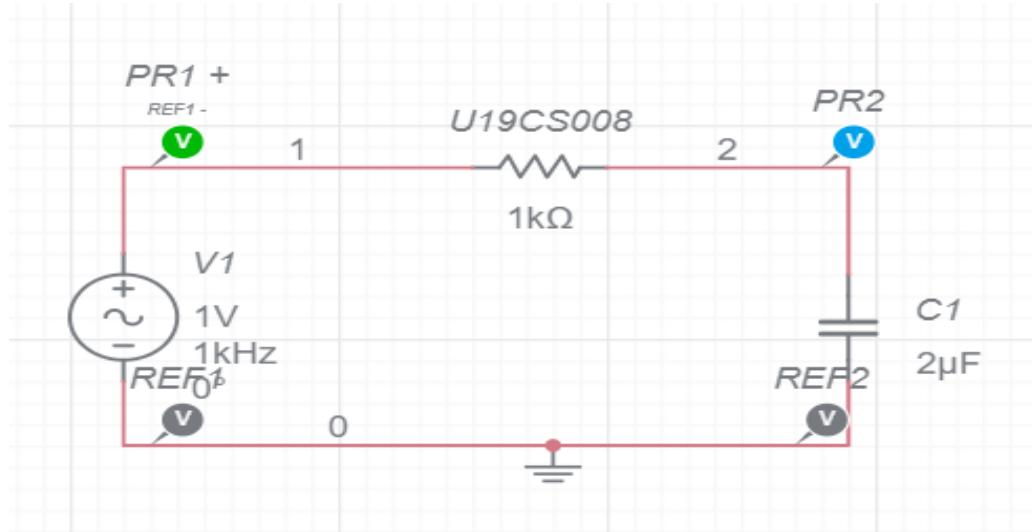


Input – Square Wave:

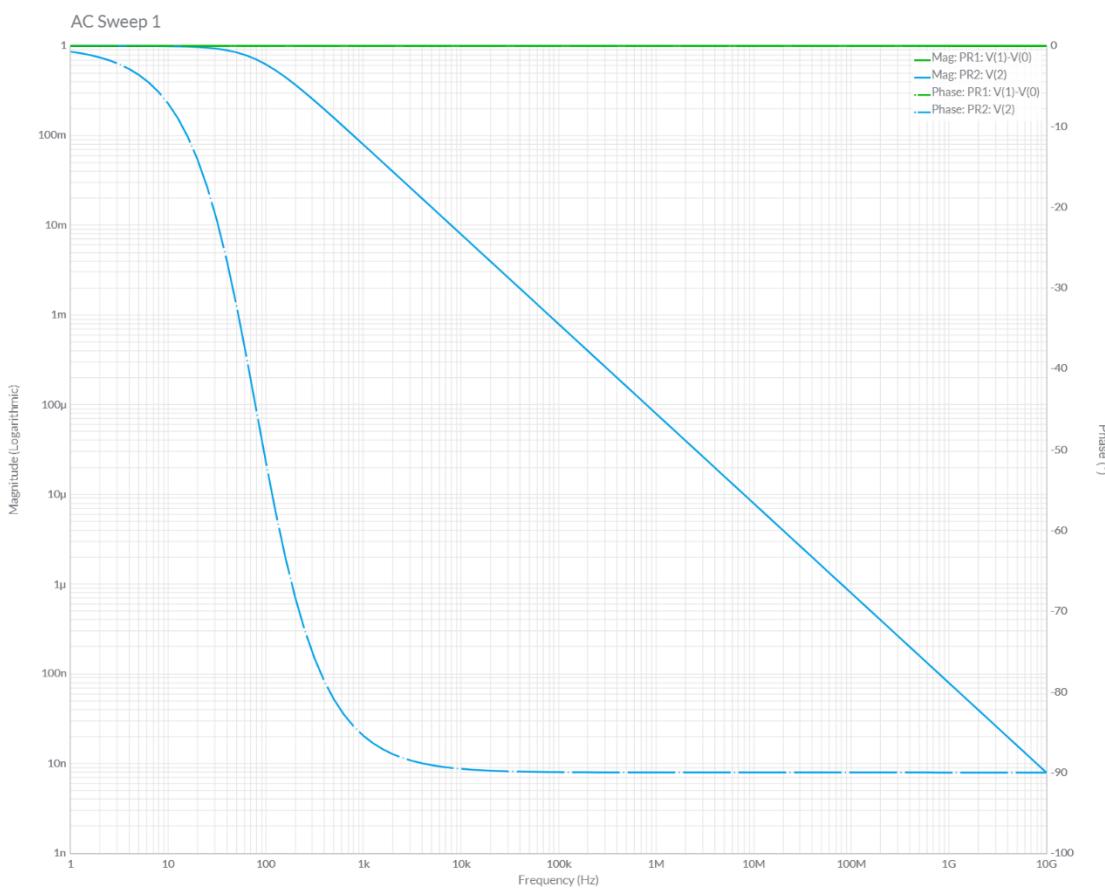




Circuit Diagram of RC – Low Pass Filter:



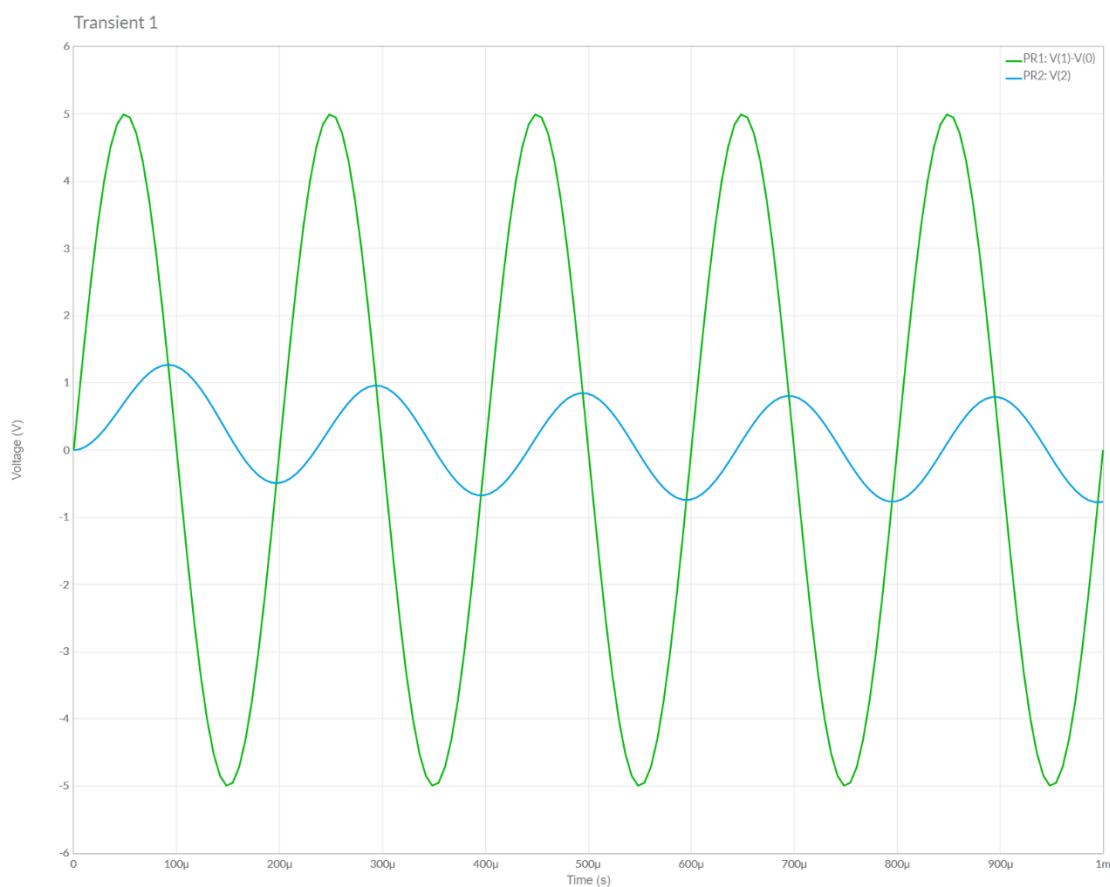
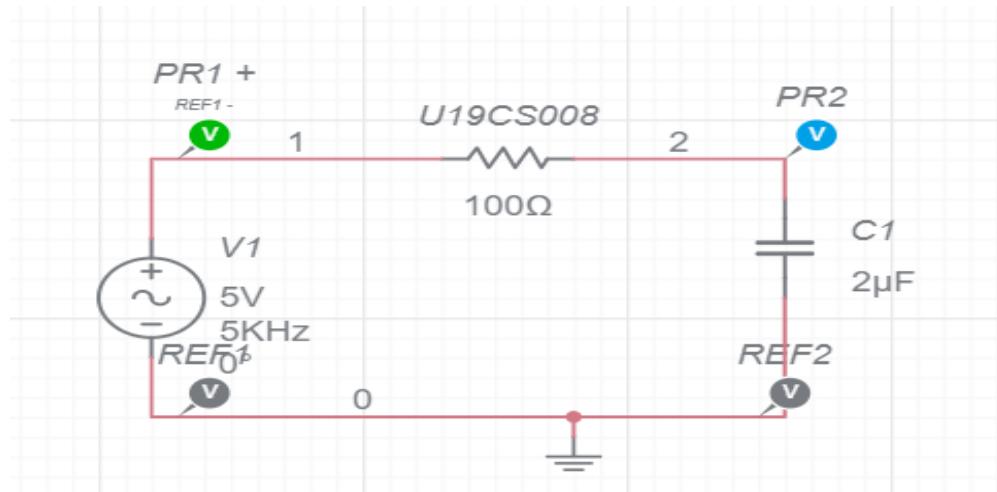
Frequency Response Plot of Low Pass Filter:





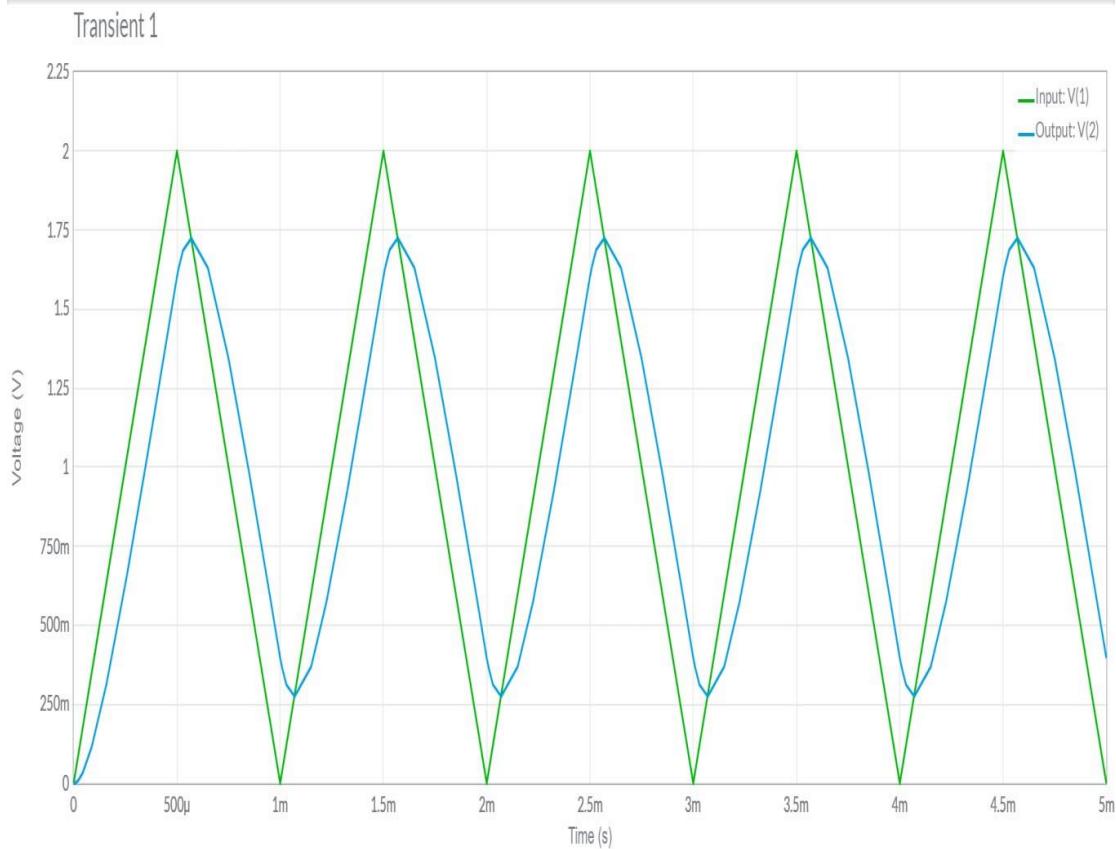
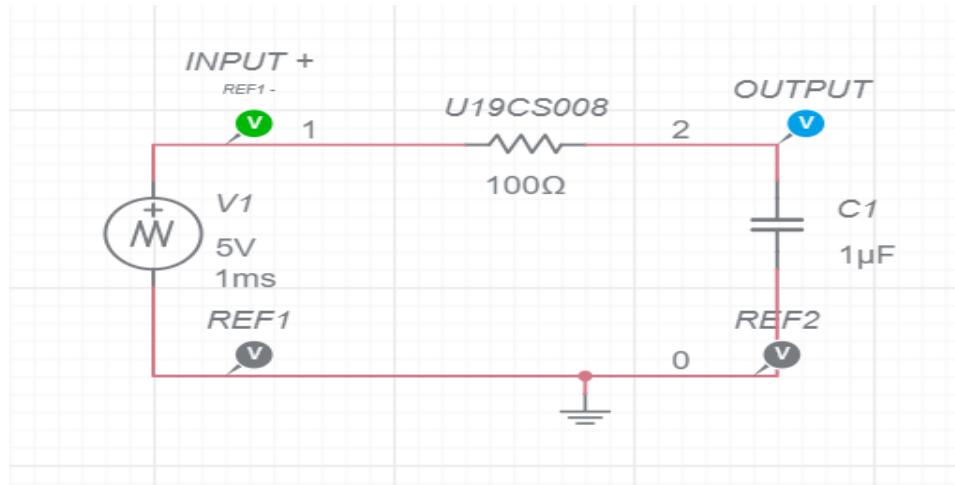
Output of Integrator:

Input – Sine Wave:



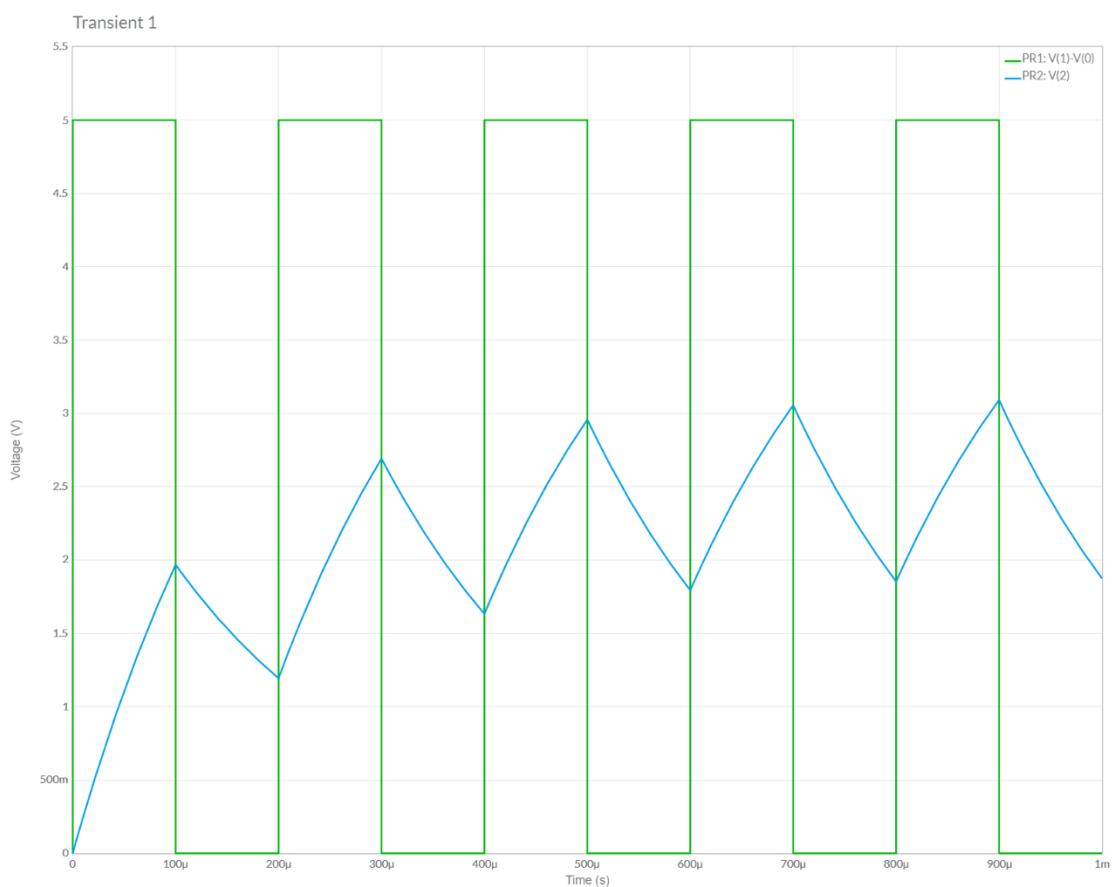
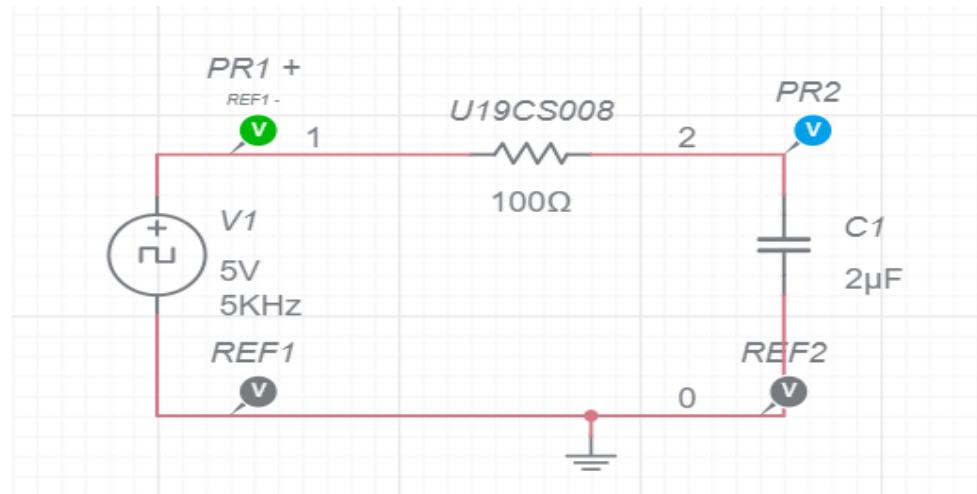


Input – Triangular Wave:





Input – Square Wave:





CONCLUSIONS

Thus, we successfully implemented RC- High pass as well as Low pass circuit Using Multisim online and obtained the Frequency at -3dB in both the cases.

Also verified the waveform graphs in following three input cases:

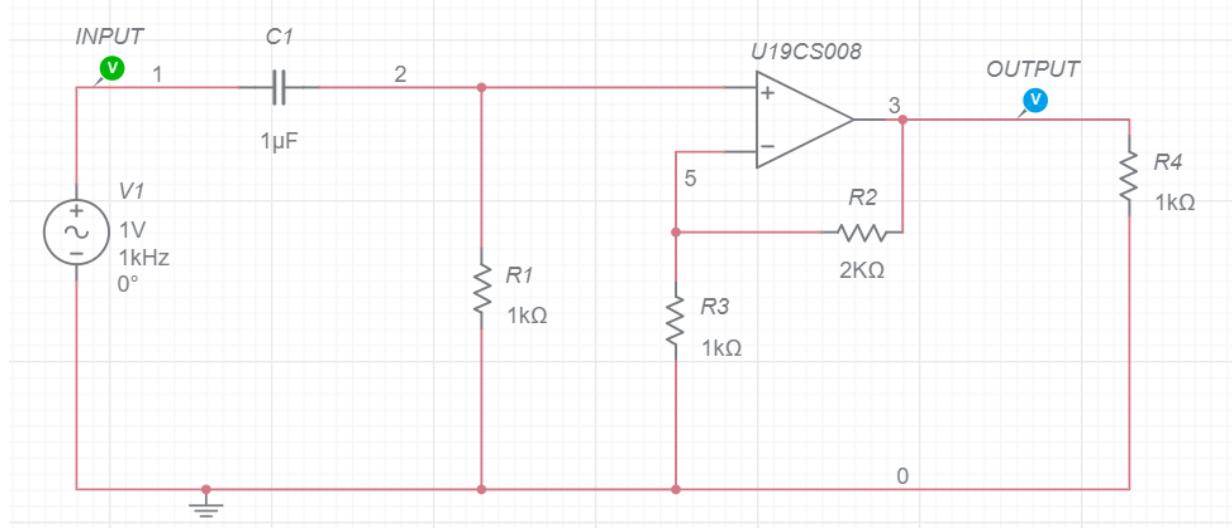
- 1) Sine
- 2) Square
- 3) Triangular Input Voltage

for RC – Low Pass Filter circuit as Integrator and for RC – High Pass Filter circuit as Differentiator.

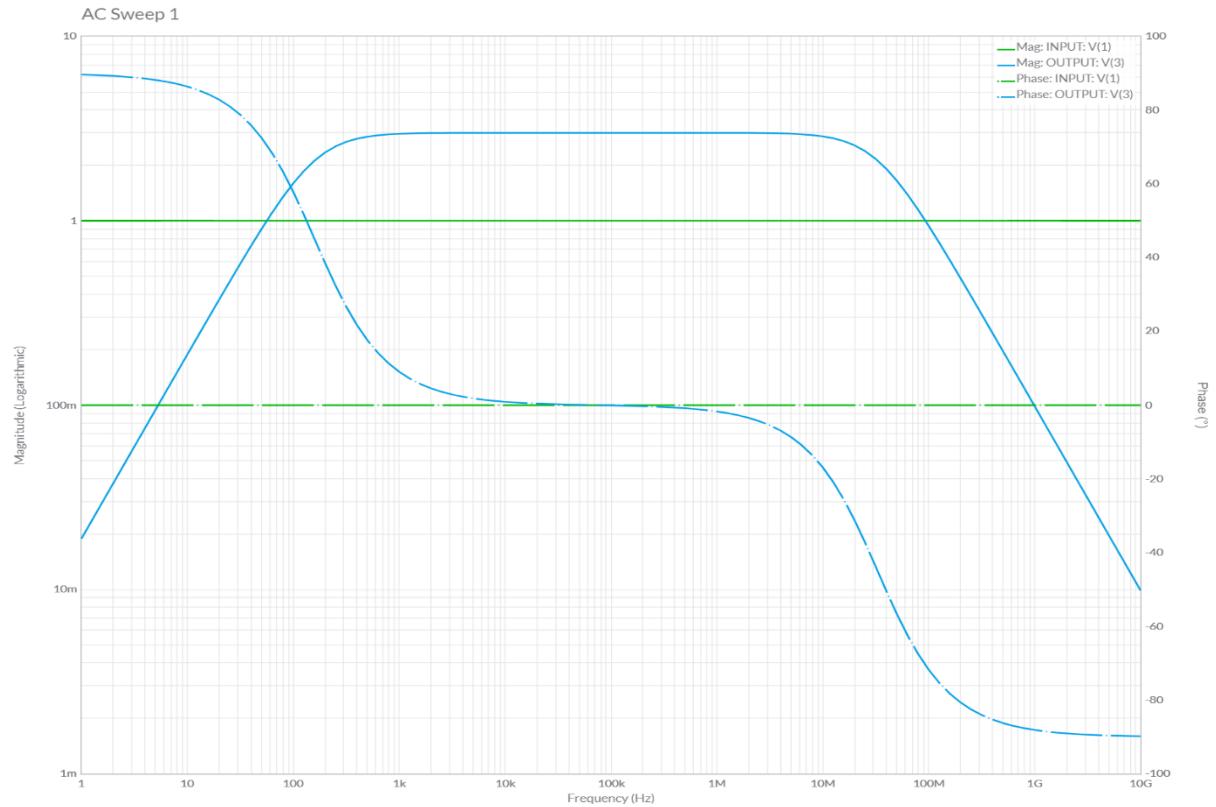
**DLED ASSIGNMENT-13****NAME: KRINA PATEL****ADMISSION NUMBER: U19CS008**

Q) SIMULATE ACTIVE HIGH PASS FILTER CIRCUIT USING OPAMP.

→ CIRCUIT:



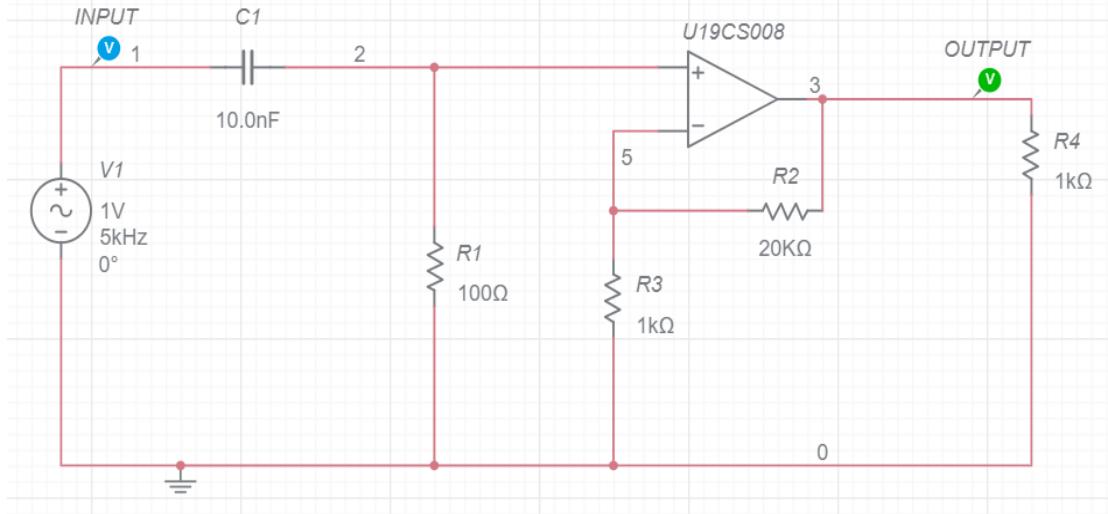
→ GRAPH:



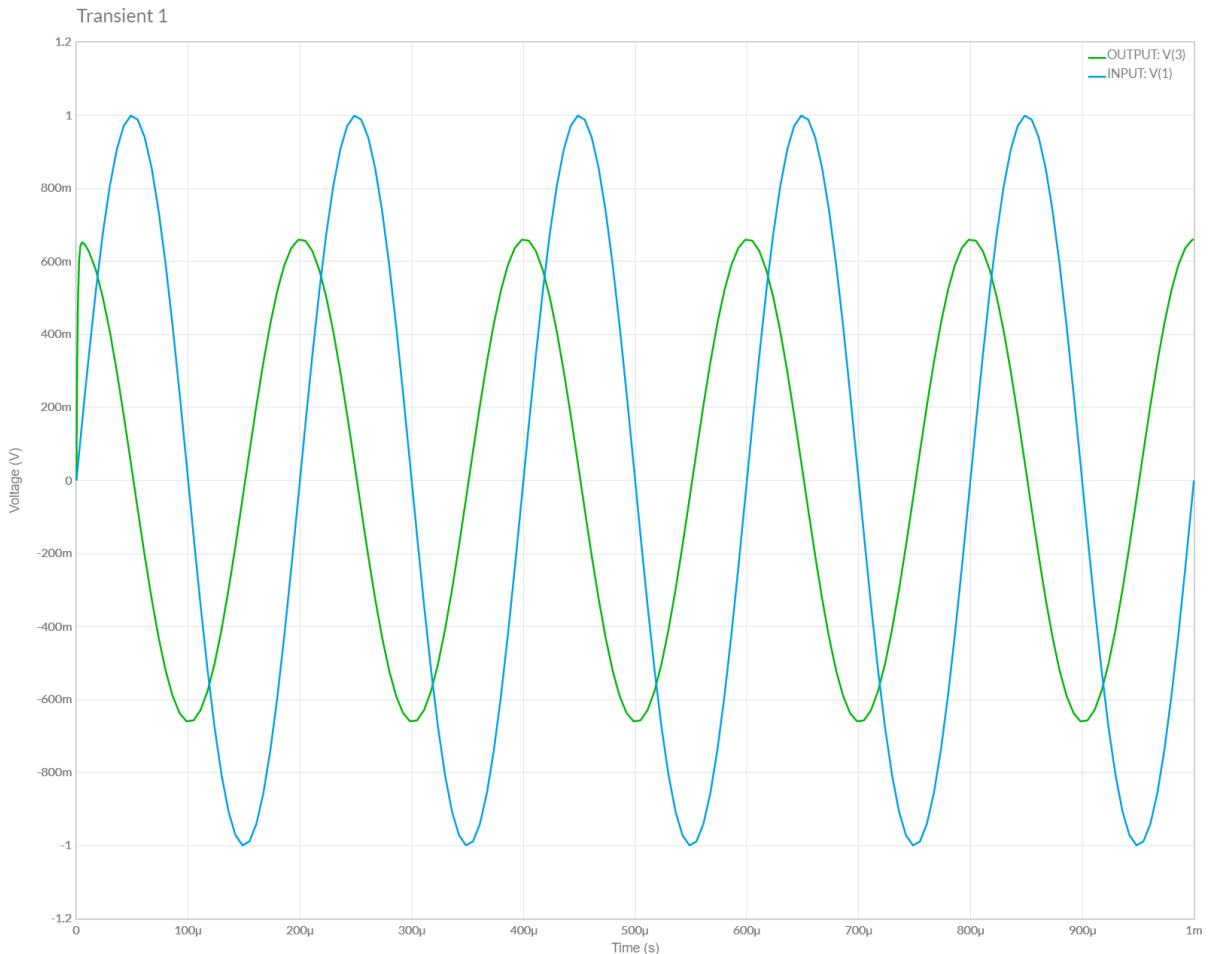


• INPUT SINE WAVE

→ GRAPH:

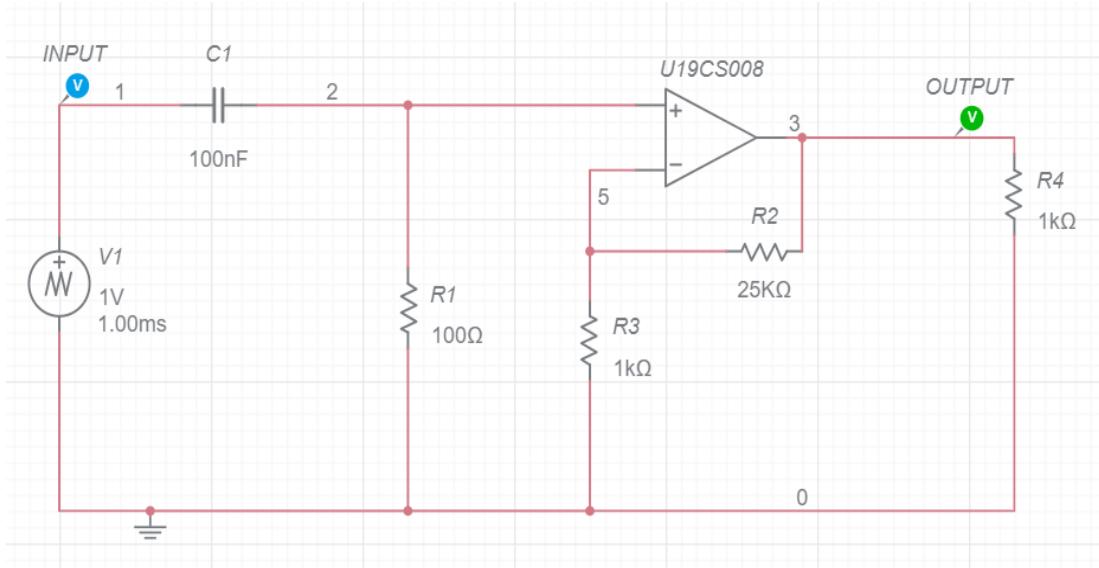


→ GRAPH:

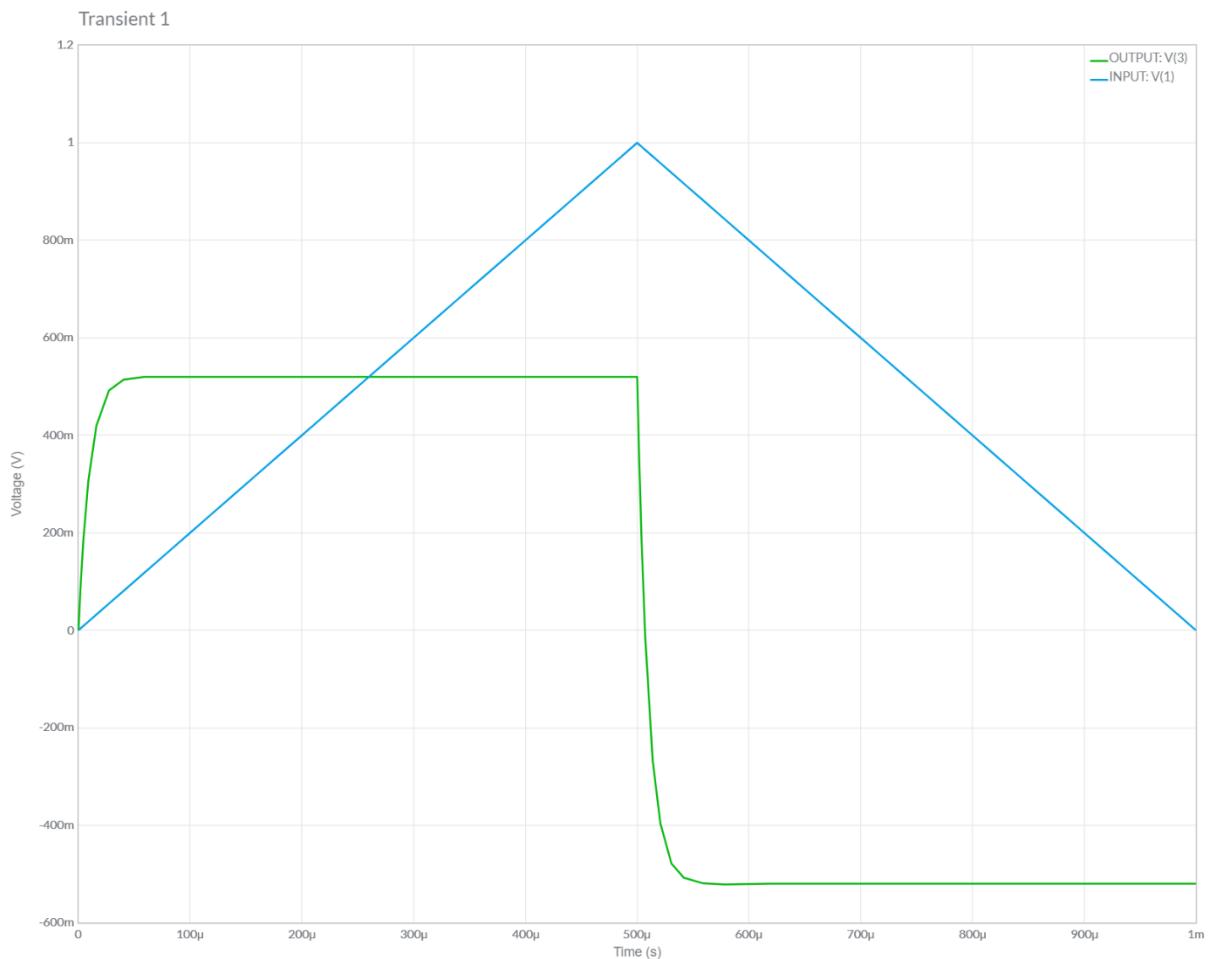




**• INPUT TRIANGULAR WAVE:
→ GRAPH:**

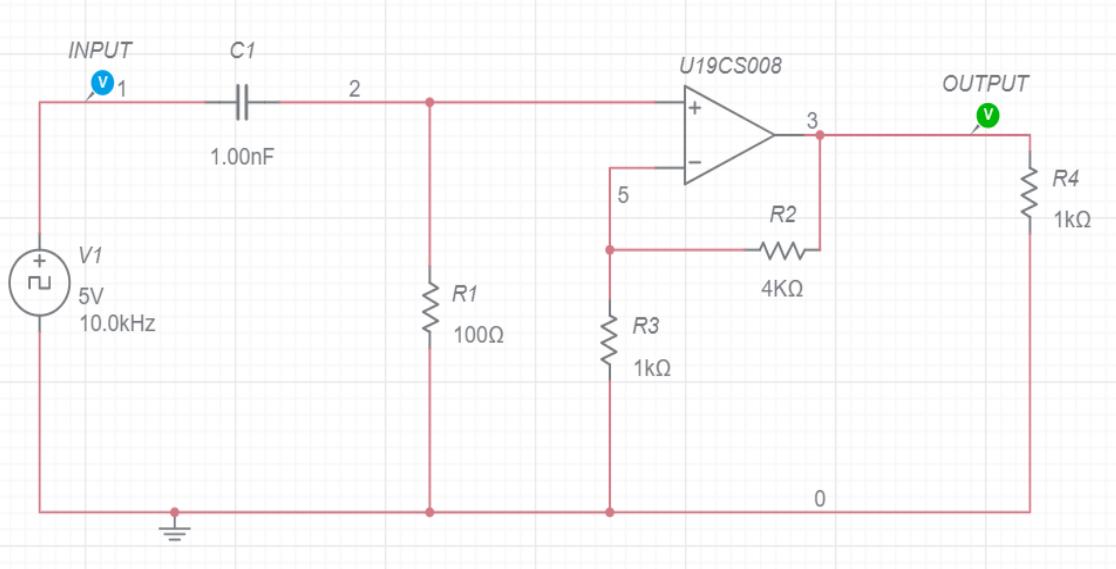


→ GRAPH:





**• INPUT SQUARE WAVE:
→ CIRCUIT:**



→ GRAPH:

