

## LABORATORY JOURNAL

*Submitted in fulfillment of the requirement*

*For the Subject*

**“DIGITAL ELECTRONICS AND LOGIC DESIGN”**

**(EC 207)**

Prepared & Submitted By

**Mr. BHAGYA VINOD RANA**

**(Admission No. U19CS012)**

**B. TECH. II (CSE) 3rd Semester**

**(Academic Year: 2020-21 [Aug-Dec 2020])**

ONLINE MODE

Laboratory Teachers

Manoj Sir

Dr. Shilpi Gupta Mam

Sudhanshu Sir

# Certificate

This is to Certify That

**Mr. BHAGYA VINOD RANA** of B.Tech II<sup>nd</sup> Computer Admission No. U19CS012 has Satisfactorily Completed His course work in **Digital Electronics and Logic Design Laboratory** during the 3<sup>rd</sup> 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**

# INDEX

Expt. No.	Experiment Name	Page No.
1	Introduction to Multisim	13/08/2020 [4]
2	Study of Basic and Universal Gates	20/08/2020 [23]
3	Half Adder and Half Subtractor	27/08/2020 [56]
4	Full Adder and Full Subtractor	03/09/2020 [69]
5	V-I Characteristics of PN – Junction Diode	10/09/2020 [83]
6	Diode Clipper Circuits (Series – Configuration)	17/09/2020 [91]
7	Diode Clipper Circuits (Shunt – Configuration)	24/09/2020 [115]
8	Diode Clamper Circuits	22/10/2020 [139]
9	Full Wave Rectifier	22/10/2020 [152]
10	Common Emitter Characteristics & Common Emitter Amplifier	29/10/2020 [158]
11	Registers and Counters	07/11/2020 [178]
12	Multiplexers and Code Converters	11/11/2020 [220]
13	High Pass and Low Pass Filters	26/11/2020 [255]



Expt. No:

1

Date:

13-08-2020

**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.

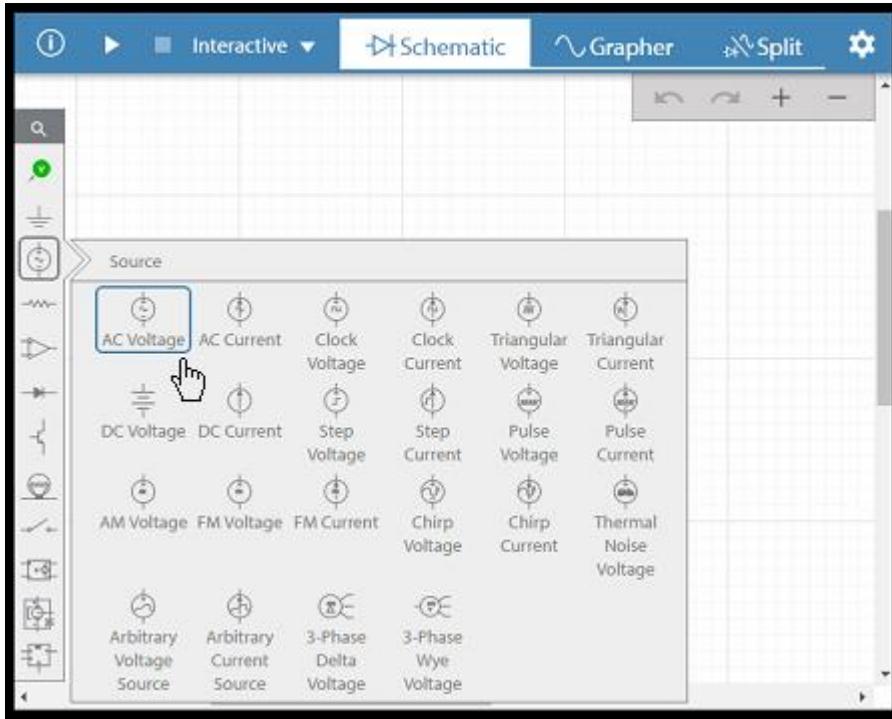
Multisim Live is a free online circuit simulator that includes SPICE software, which lets you create, learn and share electronics circuits online.

**Creating Circuits on Multisim:**

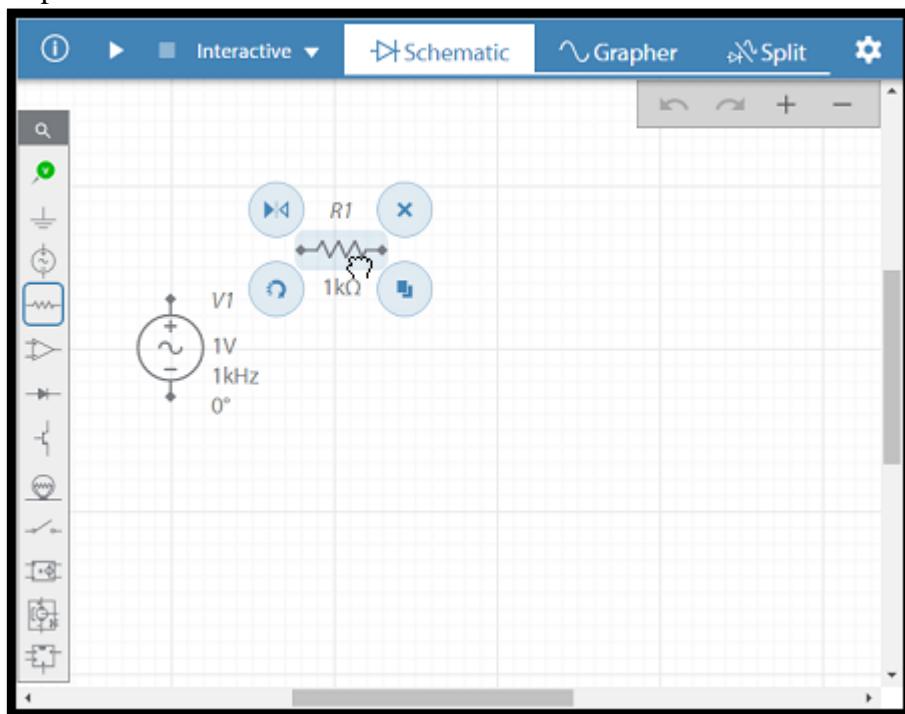
UI appears as shown.  
Click on Create Circuit

**Placing Voltage Source:**

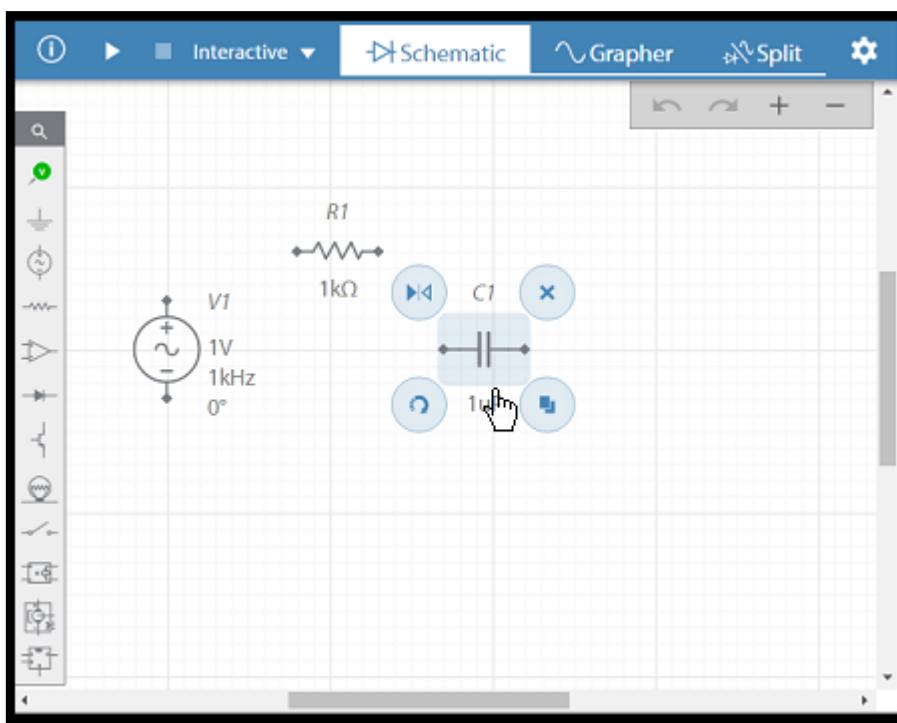
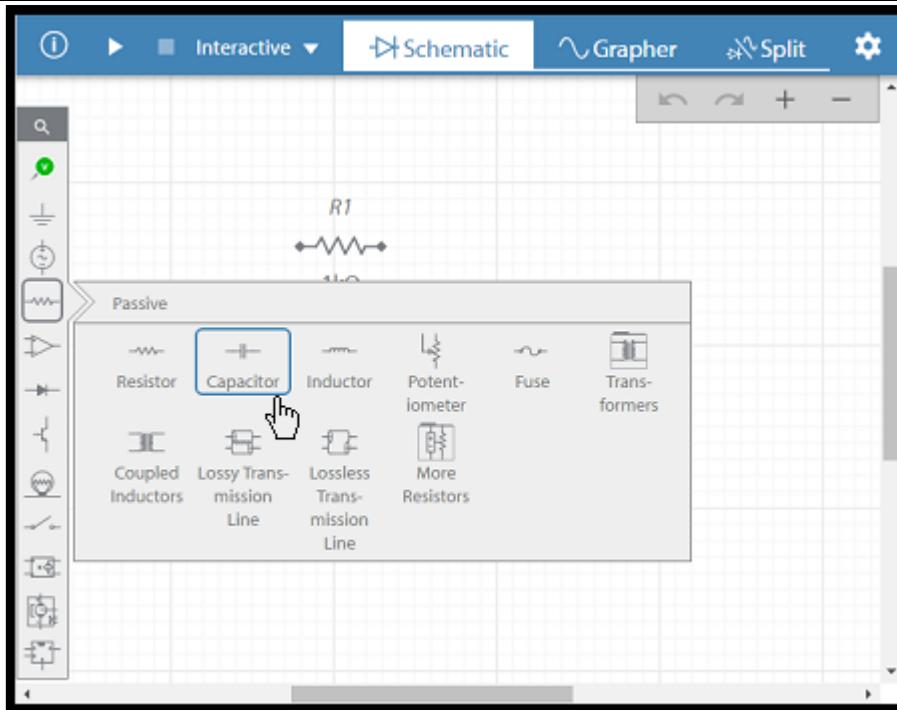
Tap the Source subpalette and tap AC Voltage and tap on the workspace or Type V if you are using a device with a keyboard, and tap to place the source.

**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.

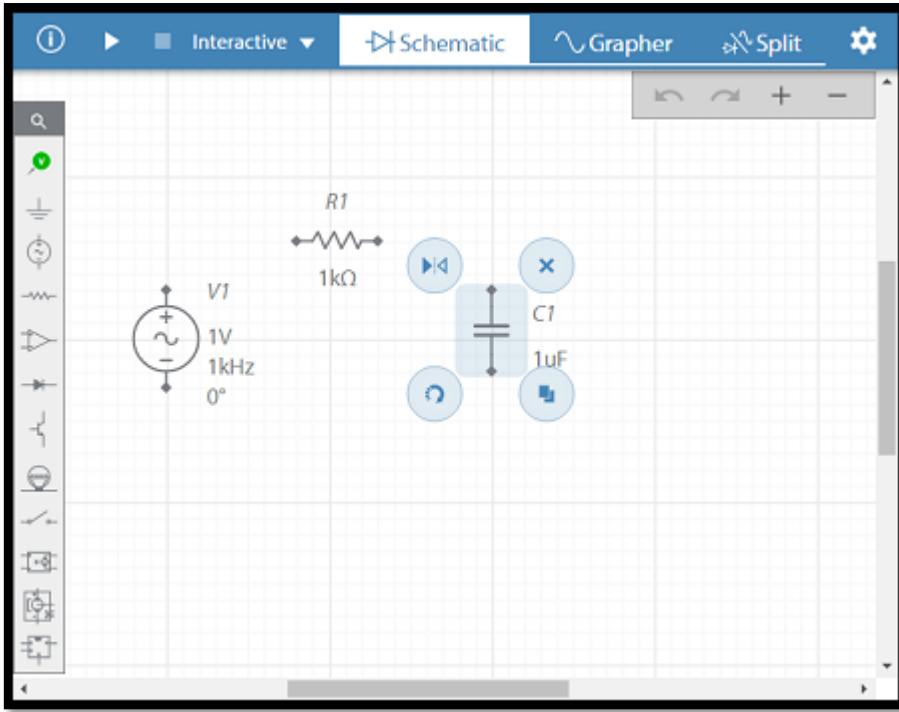
**Placing Capacitor:**

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



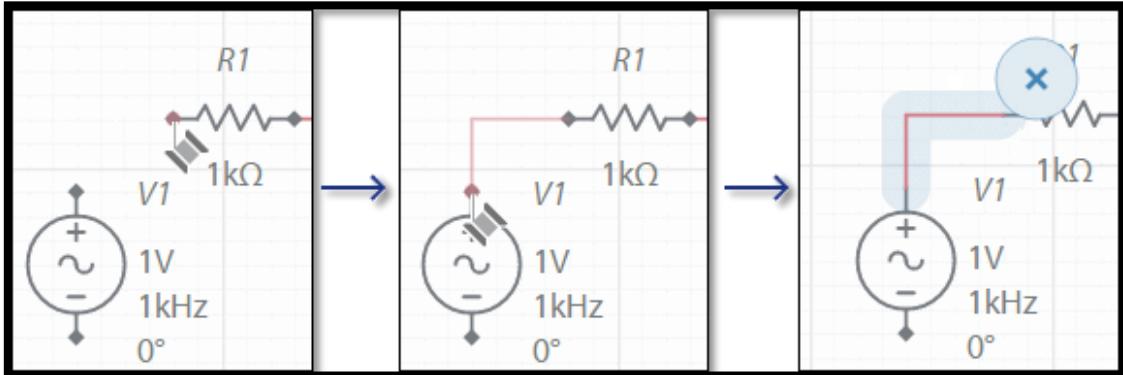
### Rotating Components:

Tap to rotate the capacitor and other components.



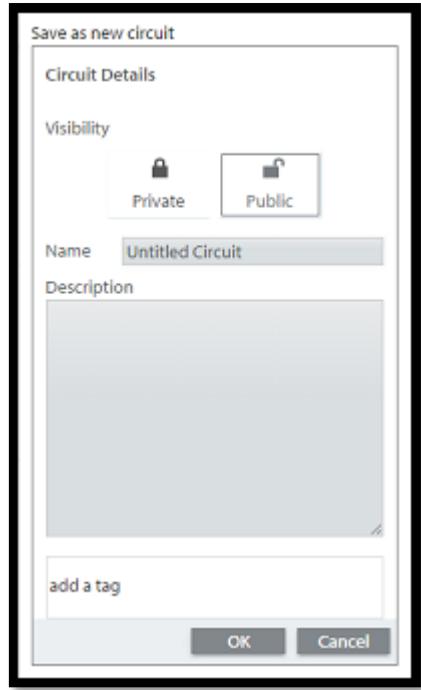
### **Wiring the components:**

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:**

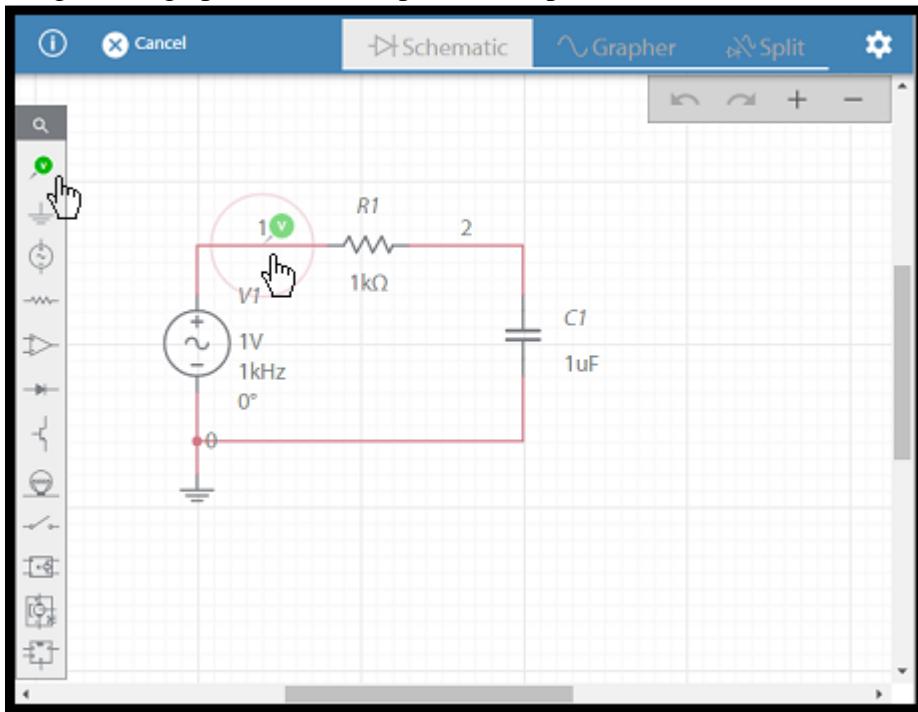
Tap in the title bar and select Save as.



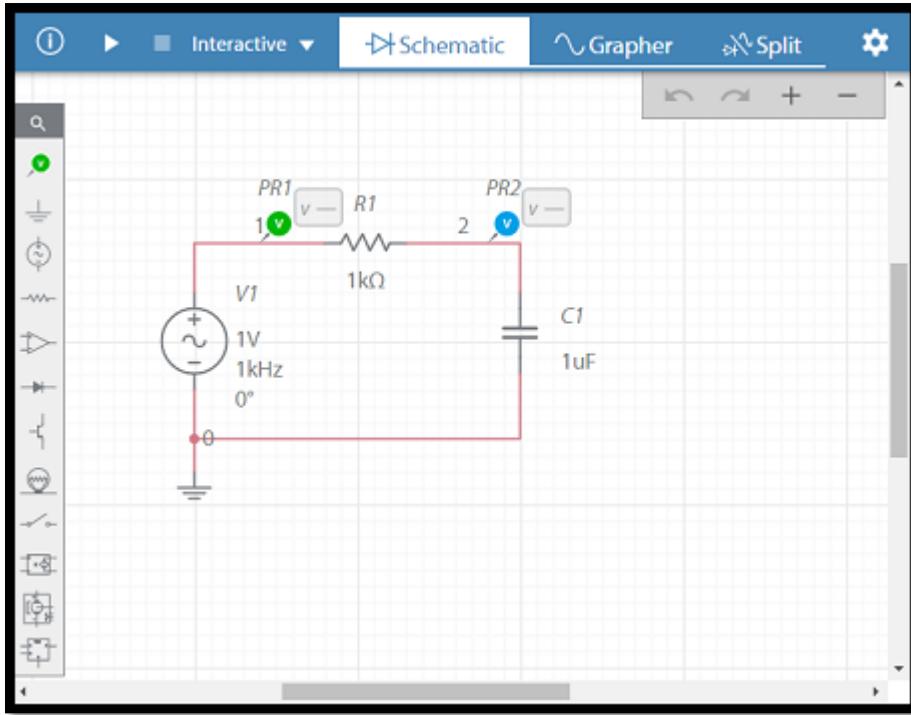
### Simulating a Design:

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

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

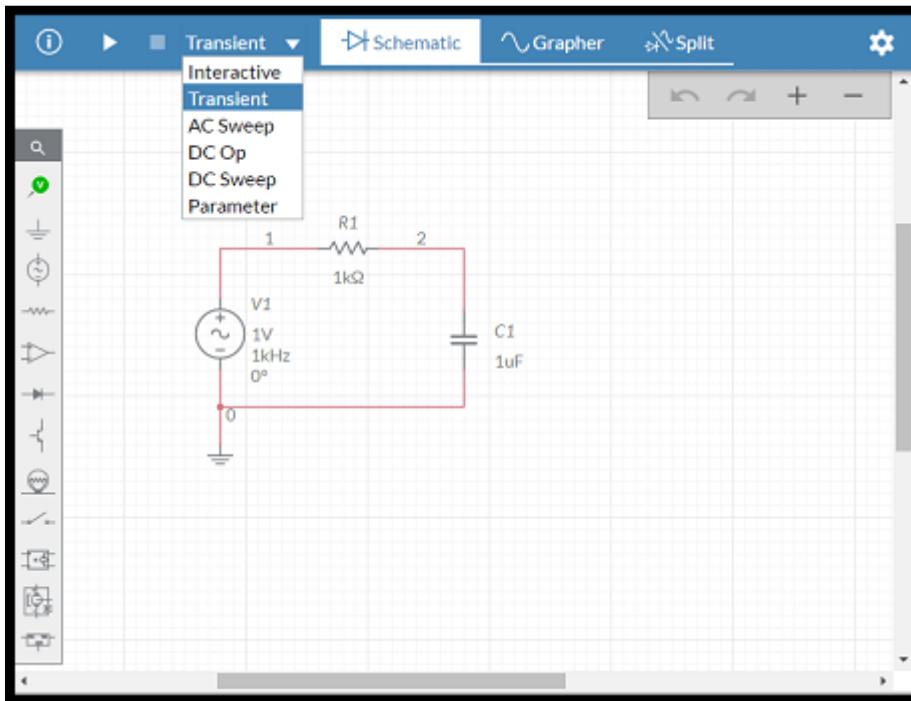


2. Place a second probe.

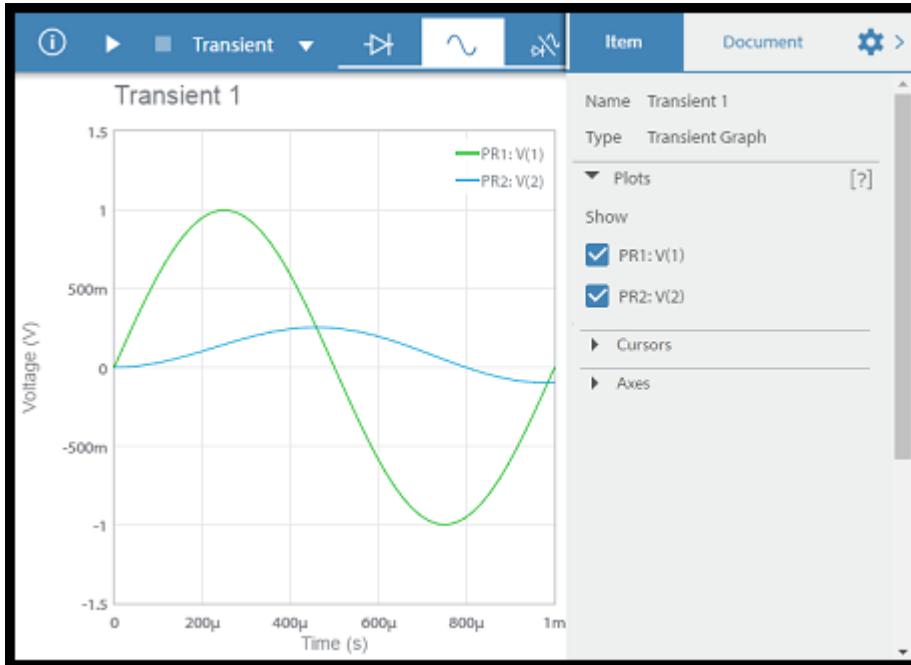


### Select and run simulation

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.



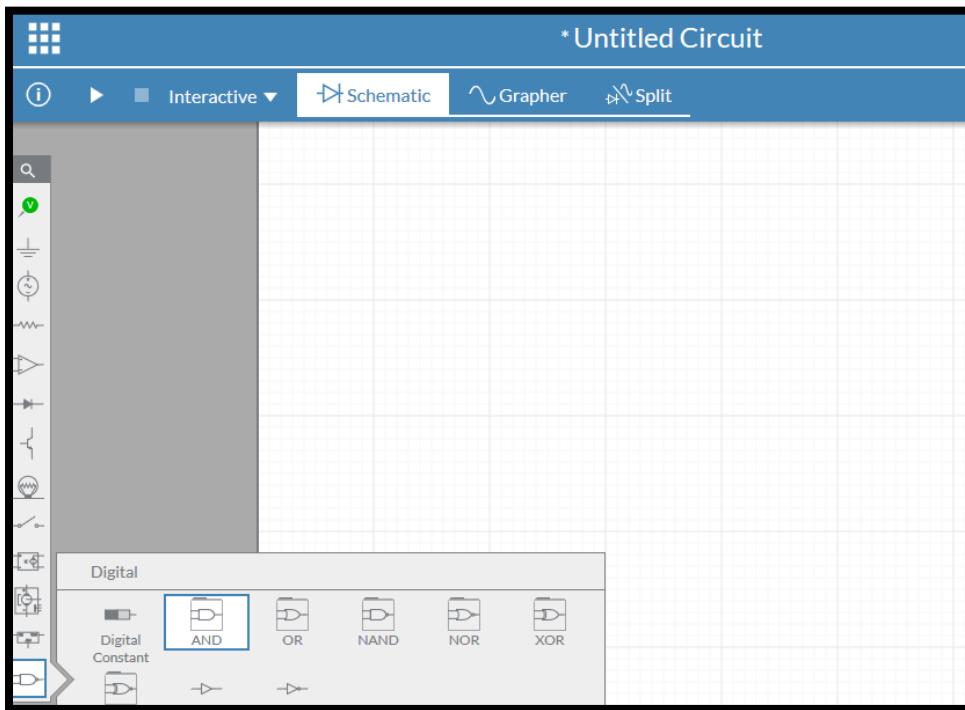
5. Switch the simulation type to AC Sweep and tap



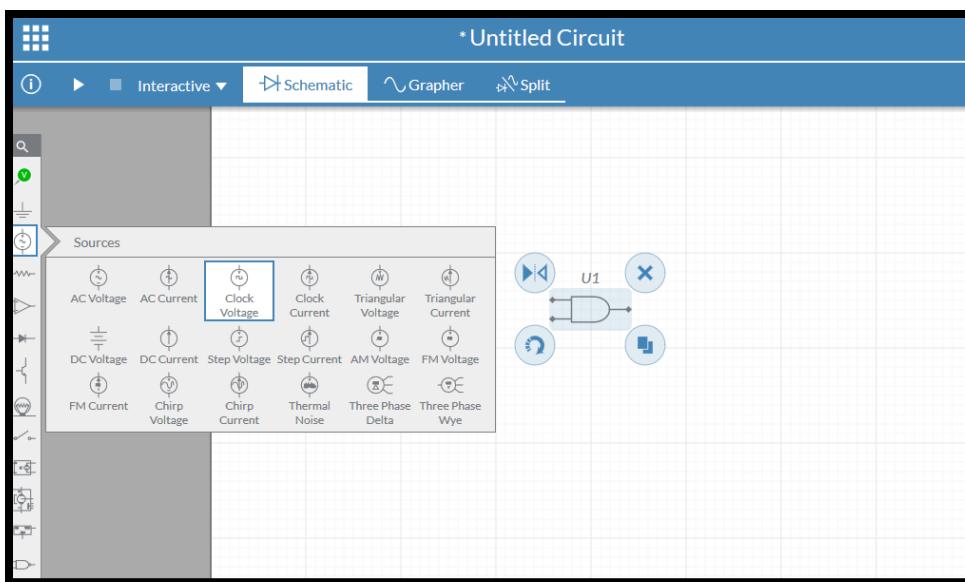
**Simulating a simple Digital Circuit:**



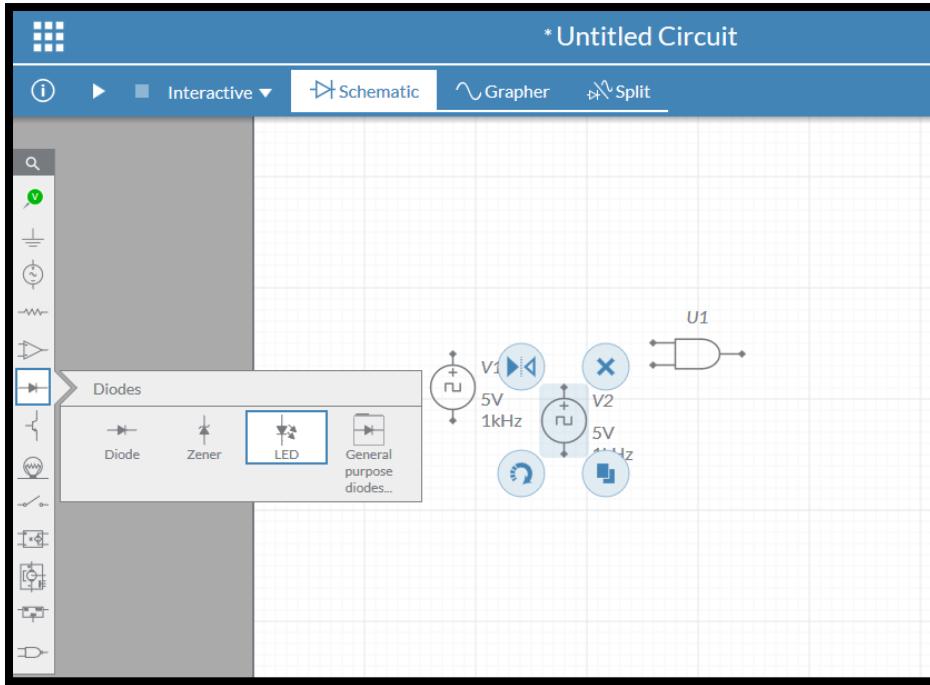
**Step 1: Selecting AND Gate**



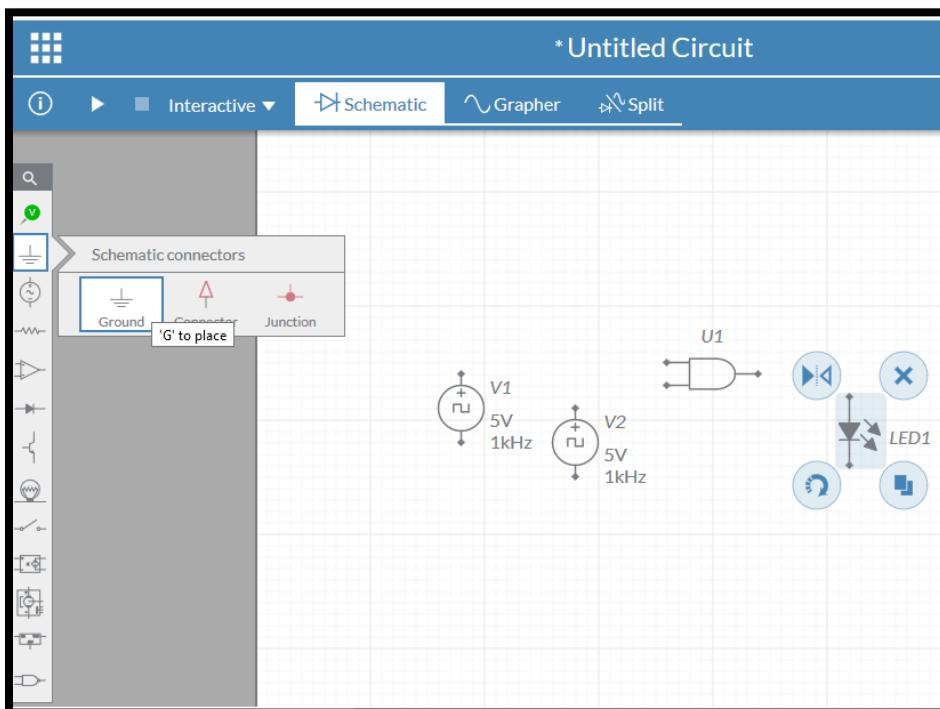
**Step 2: Adding Source (Clock Voltages)**



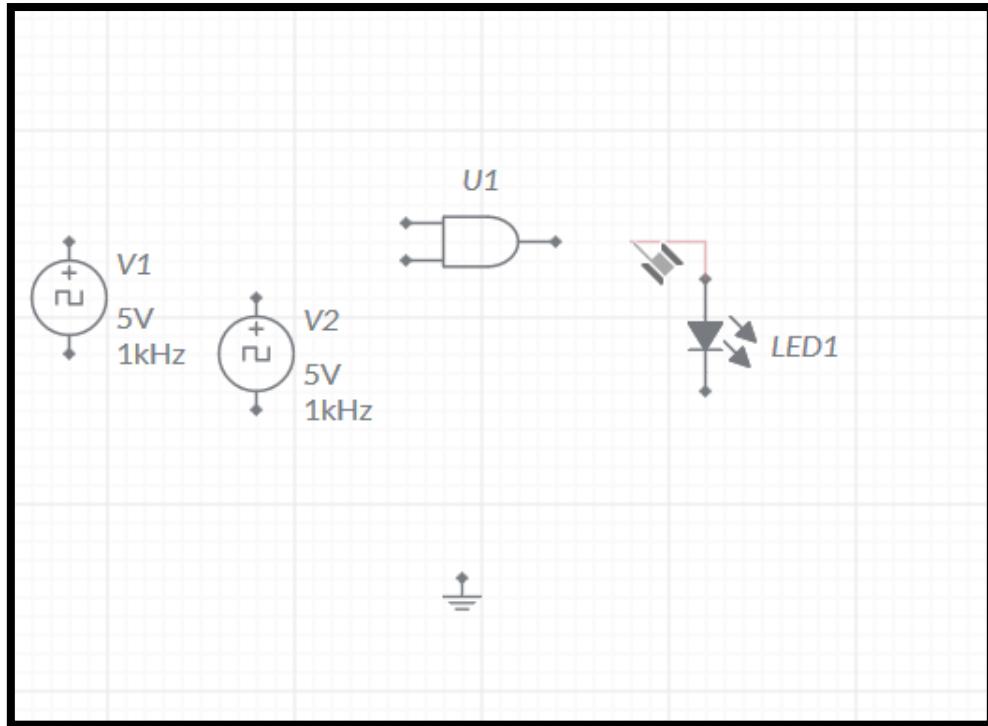
**Step 3: Adding Load (LED)**



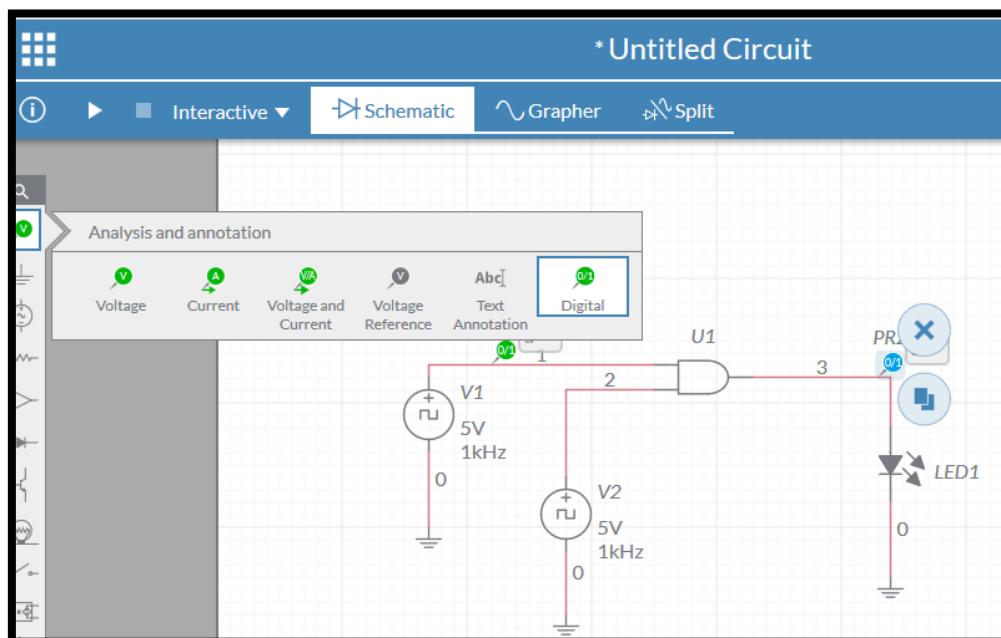
#### **Step 4: Grounding the Components:**



#### **Step 5: Connecting Components**



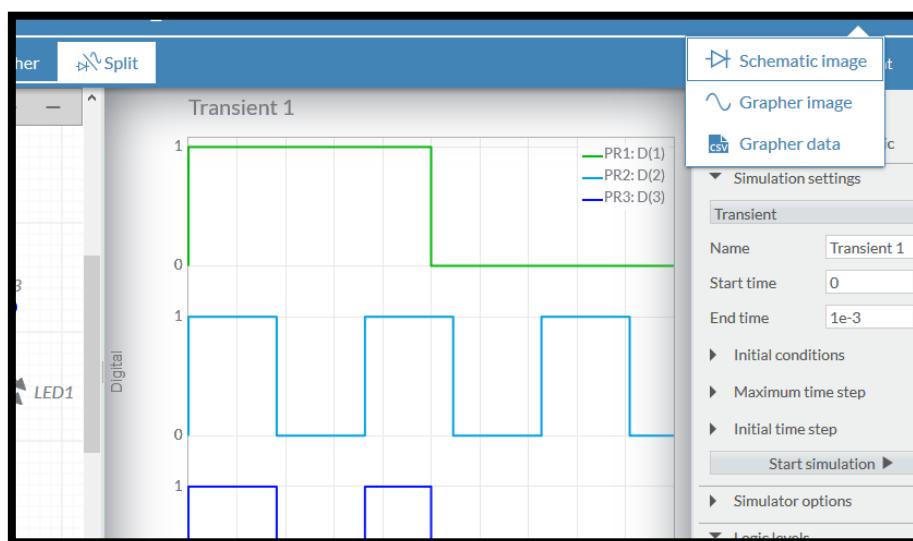
### Step 6: Adding Probes



### Step 7: Finally we Simulate the Design



### Step 8: Exporting Schematic/Grapher Images/Screenshots

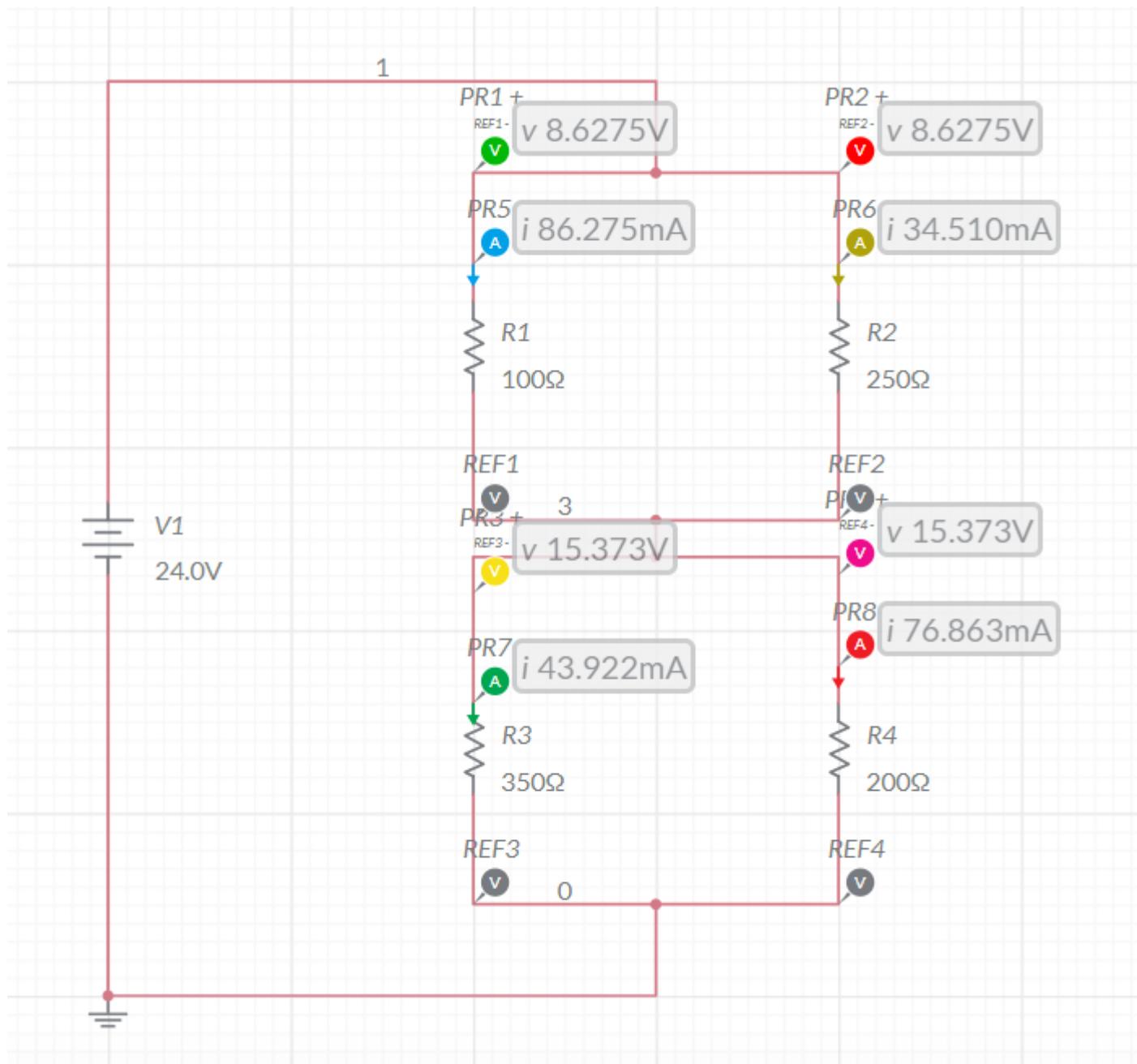


### CONCLUSIONS:

- 1.) We learned and Implemented Multisim Software interface and used different electronic tools available in the simulator to create circuits.
- 2.) We used *resistor*, *wires*, *D.C. voltage source*, *Voltmeter*, *Ammeter* and other electronic devices to verify current and voltage across resistor by *theoretical* and simulated data with Multisim values obtained.
- 3.) We also learned how to Export Schematic Image, grapher Image and its Data from Multisim.

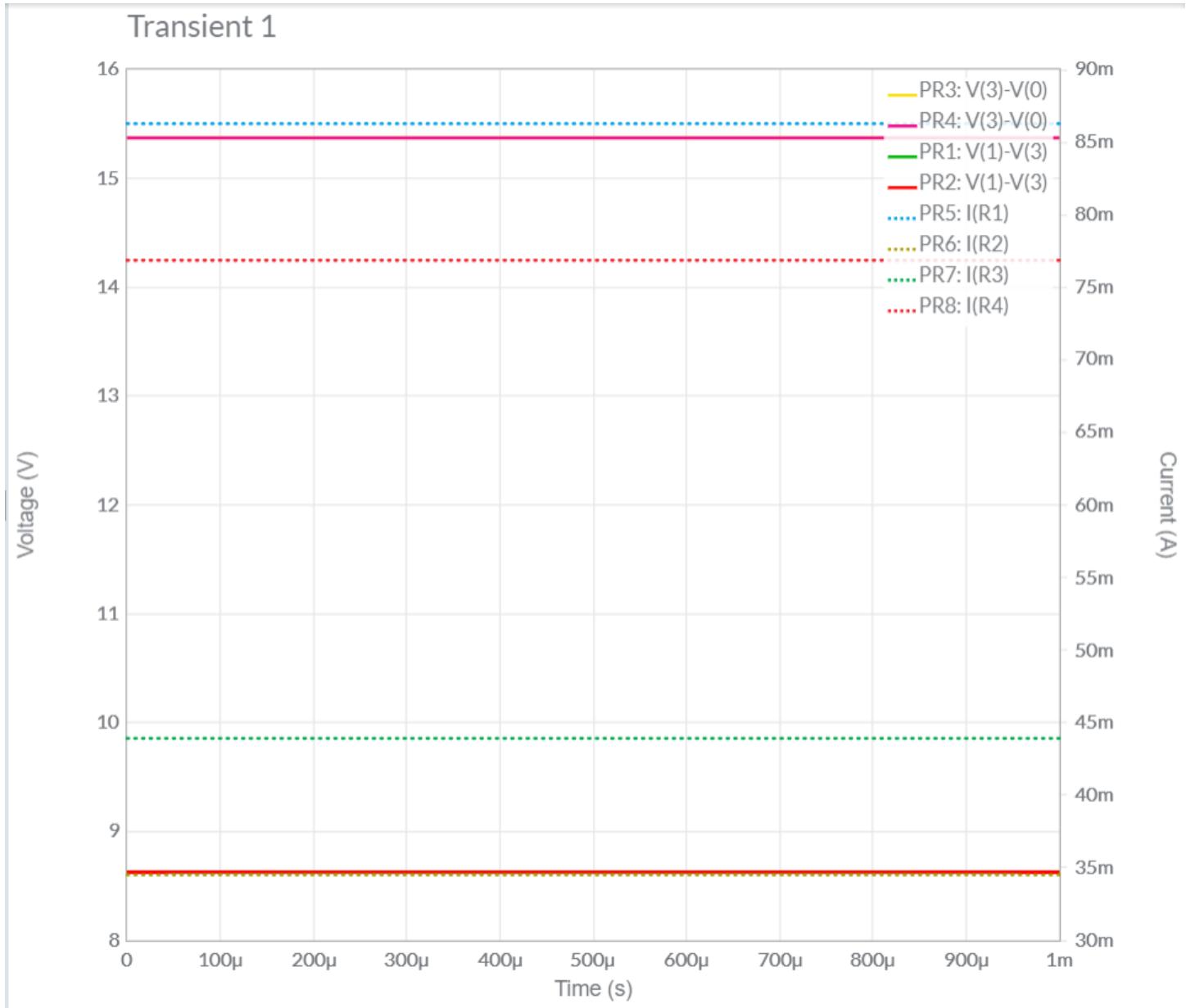
Assignment - 1 Q1

a.) Implement the circuit as shown in Figure in Multisim online.





b.) Evaluate the current and voltage across each resistor using simulator.



DC OP 1

Signal	Value
PR3: V(3)-V(0)	15.373V
PR4: V(3)-V(0)	15.373V
PR1: V(1)-V(3)	8.6275V
PR2: V(1)-V(3)	8.6275V
PR5: I(R1)	86.275mA
PR6: I(R2)	34.510mA
PR7: I(R3)	43.922mA
PR8: I(R4)	76.863mA

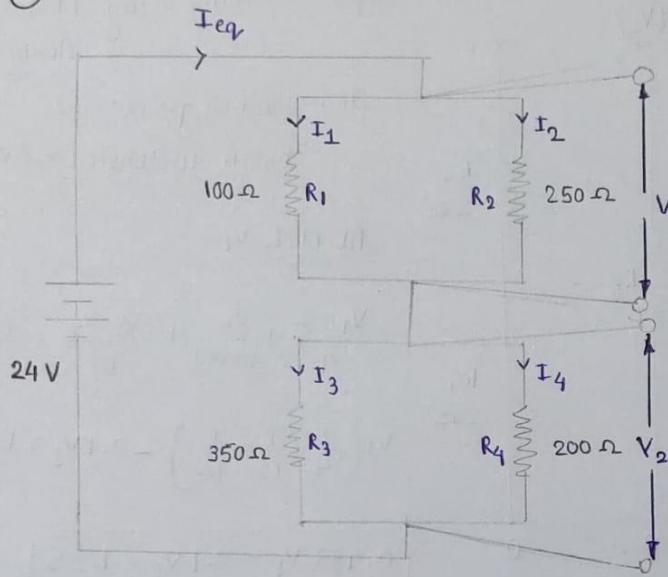


c.) Compare with theoretical values.

ASSIGNMENT - 1

( UI9CS012 )

(1)



$$\text{Step 1 : } R_{eq} = \left( \frac{R_1 \times R_2}{(R_1 + R_2)} \right) + \left( \frac{R_3 \times R_4}{(R_3 + R_4)} \right) \Omega$$

$$\begin{aligned} R_{eq} &= \left( \frac{250 \times 100}{350} + \frac{350 \times 200}{550} \right) \Omega \\ &= 71.4286 + 127.27 \Omega \\ &\approx 198.7013 \Omega \\ &\approx \underline{\underline{198.7 \Omega}} \end{aligned}$$

$$\begin{aligned} \text{Step 2 : } I_{eq} &= \frac{V}{R_{eq}} = \frac{24V}{198.7 \Omega} \\ &= 0.120785 A \\ &= \underline{\underline{120.785 mA}} \end{aligned}$$

Step 3 : Parallel

$$V_1 = I_1 R_1 \quad (\text{i}) \quad V_1 = I_2 R_2 \quad (\text{ii})$$

$$\therefore I_1 R_1 = I_2 R_2 \quad \text{--- (1)}$$

$$I_1 + I_2 = I_{eq} \quad \text{--- (2)}$$

Derivation (Just for info)

$$I_1 = I_{eq} - \left( \frac{I_1 R_1}{R_2} \right) \Rightarrow I_1 \left( 1 + \frac{R_1}{R_2} \right) = I_{eq} \Rightarrow \left[ I_1 = \frac{I_{eq} (R_2)}{R_1 + R_2} \right]$$

$$\text{Similarly } \left[ I_2 = \frac{I_{eq} (R_1)}{R_1 + R_2} \right]$$

$$I_1 = \frac{(I_{eq}) R_2}{(R_1 + R_2)} = (120.785) \text{ mA} \times \frac{(250)}{350} = \boxed{86.275 \text{ mA}}$$

$$I_2 = \frac{(I_{eq}) R_1}{(R_1 + R_2)} = (120.785) \text{ mA} \times \frac{(100)}{350} = \boxed{34.51 \text{ mA}}$$

$$V_1 = I_1 R_1 = (86.275 \times 10^{-3} \text{ A}) \times (100 \Omega) = \boxed{8.6275 \text{ V}}$$

Similarly;

$$I_3 = \frac{(I_{eq}) R_4}{(R_3 + R_4)} = (120.785) \text{ mA} \times \frac{(200)}{550} = 43.9218 \text{ mA} \approx \boxed{43.922 \text{ mA}}$$

$$I_4 = \frac{(I_{eq}) R_3}{(R_3 + R_4)} = (120.785) \text{ mA} \times \frac{(350)}{550} = \boxed{76.8632 \text{ mA}}$$

$$V_2 = I_3 R_3 = (43.922 \times 10^{-3} \text{ A}) \times 350 \Omega = 15.3727 \text{ V} \approx \boxed{15.373 \text{ V}}$$



## d.) Final Result and Conclusion

Resistor	Voltage (V)		Current (mA)	
	Multism	Theoretical	Multism	Theoretical
R1	8.6275	8.6275	86.275	86.275
R2	8.6275	8.6275	34.51	34.51
R3	15.373	15.373	43.922	43.922
R4	15.373	15.373	76.863	76.8632

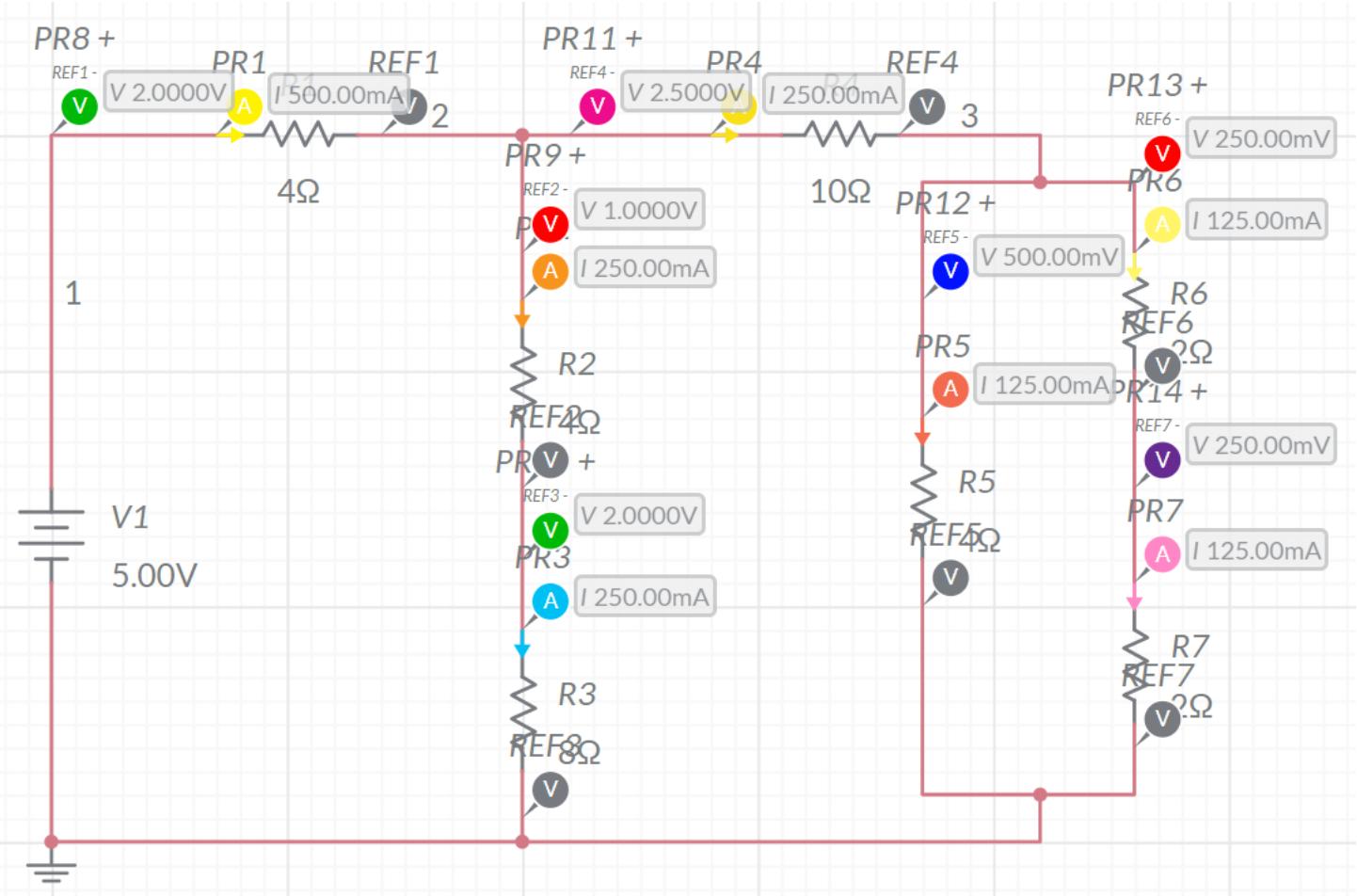
*Conclusion:*

We can observe from Above Table, Both the *Theoretical* and *Multisim* Values of Current and Voltage are **Equal**.  
 Hence, Experiment is Performed Successfully (without any Error).



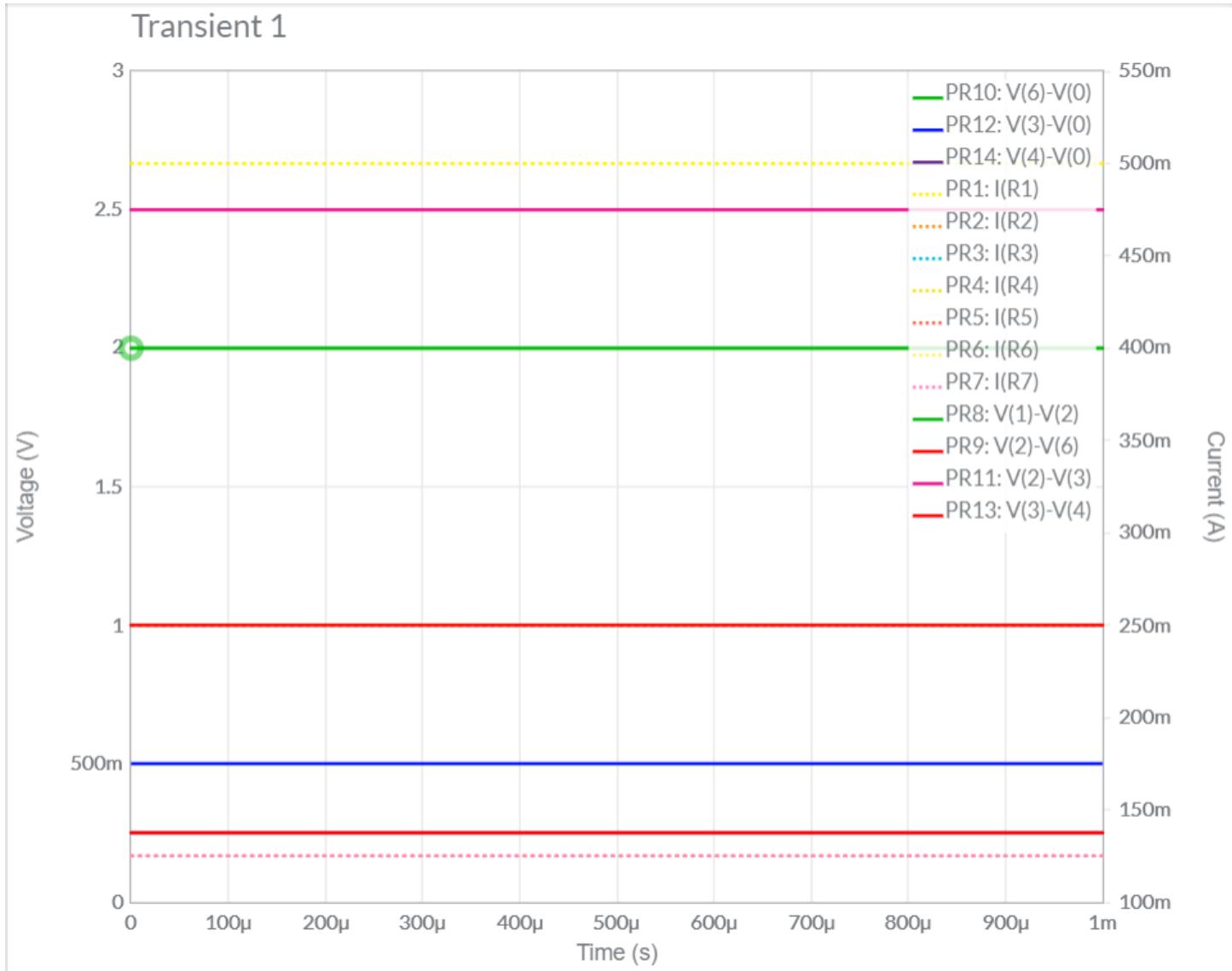
## Assignment -1 Q2

a.) Implement the circuit as shown in Figure in Multisim online.





*b.) Evaluate the current and voltage across each resistor using simulator.*



**DC OP 1**

Signal	Value
PR10: V(6)-V(0)	2.0000V
PR12: V(3)-V(0)	500.00mV
PR14: V(4)-V(0)	250.00mV
PR1: I(R1)	500.00mA
PR2: I(R2)	250.00mA
PR3: I(R3)	250.00mA
PR4: I(R4)	250.00mA
PR5: I(R5)	125.00mA
PR6: I(R6)	125.00mA
PR7: I(R7)	125.00mA
PR8: V(1)-V(2)	2.0000V
PR9: V(2)-V(6)	1.0000V
PR11: V(2)-V(3)	2.5000V
PR13: V(3)-V(4)	250.00mV



c.) Compare with theoretical values.

ASSIGNMENT - 2			(V19CS012)																							
			Step 1 : Solving using Node Analysis																							
<p>Two Junction points as shown in figure (<math>V_1</math> &amp; <math>V_3</math>)</p> <p>At Node <math>V_1</math></p> $\frac{V_1 - 5}{4} + \frac{V_1}{(4+8)} + \frac{V_1 - V_3}{10} = 0$ $V_1 \left( \frac{1}{4} + \frac{1}{12} + \frac{1}{10} \right) - 0.1 V_3 = 1.25$			$0.433 V_1 - 0.1 V_3 = 1.25 \quad \text{Eqn 1}$																							
<p>Step 2 : At Node <math>V_3</math> :</p> $\frac{V_3 - V_1}{10} + \frac{V_3}{4} + \frac{V_3}{(2+2)} = 0$ $V_3 (0.6) - 0.1 V_1 = 0$ $V_1 = 6 V_3 \quad \text{Eqn 2}$			$V_3 (0.433 \times 6 - 0.1) = 1.25$ $V_3 = 0.5004 \text{ V}$ $V_3 \approx 0.5 \text{ V}$ $V_1 = 3 \text{ V}$																							
<table border="1"> <thead> <tr> <th>Across Resistor</th> <th>Voltage (volt)</th> <th>Current</th> </tr> </thead> <tbody> <tr> <td><math>R_1</math></td> <td><math>5 - V_1 = 2 \text{ V}</math></td> <td><math>I_{R_1} = \frac{V_{R_1}}{(R_1)} = \frac{2 \text{ V}}{4 \Omega} = 0.5 \text{ A} = 500 \text{ mA}</math></td> </tr> <tr> <td><math>R_2</math></td> <td><math>V_{R_2} = I_{R_2} \times R_2 = (0.25 \text{ A}) \times 4 = 1 \text{ V}</math></td> <td><math>I_{R_2} = \frac{V_{R_2}}{(R_2 + R_3)} = \frac{1 \text{ V}}{12} = 0.25 \text{ A} = 250 \text{ mA}</math></td> </tr> <tr> <td><math>R_3</math></td> <td><math>V_{R_3} = I_{R_3} \times R_3 = 0.25 \times 8 = 2 \text{ V}</math></td> <td><math>I_{R_3} = I_{R_2} = 0.25 \text{ A} = 250 \text{ mA}</math></td> </tr> <tr> <td><math>R_4</math></td> <td><math>V_{R_4} = V_1 - V_3 = 2.5 \text{ V}</math></td> <td><math>I_{R_4} = \frac{V_{R_4}}{R_4} = \frac{2.5 \text{ V}}{10 \Omega} = 250 \text{ mA} = 0.25 \text{ A}</math></td> </tr> <tr> <td><math>R_5</math></td> <td><math>V_{R_5} = V_3 = 0.5 \text{ V}</math></td> <td><math>I_{R_5} = \frac{V_{R_5}}{R_5} = \frac{0.5}{4} = 0.125 \text{ A} = 125 \text{ mA}</math></td> </tr> <tr> <td><math>R_6</math></td> <td><math>V_{R_6} = I_{R_6} \times R_6 = 0.25 \text{ V}</math></td> <td><math>I_{R_6} = \frac{V_{R_6}}{(R_6 + R_7)} = \frac{0.25}{4} = 125 \text{ mA}</math></td> </tr> <tr> <td><math>R_7</math></td> <td><math>V_{R_7} = I_{R_7} \times R_7 = 0.25 \text{ V}</math></td> <td><math>I_{R_7} = I_{R_6} = 125 \text{ mA}</math></td> </tr> </tbody> </table>			Across Resistor	Voltage (volt)	Current	$R_1$	$5 - V_1 = 2 \text{ V}$	$I_{R_1} = \frac{V_{R_1}}{(R_1)} = \frac{2 \text{ V}}{4 \Omega} = 0.5 \text{ A} = 500 \text{ mA}$	$R_2$	$V_{R_2} = I_{R_2} \times R_2 = (0.25 \text{ A}) \times 4 = 1 \text{ V}$	$I_{R_2} = \frac{V_{R_2}}{(R_2 + R_3)} = \frac{1 \text{ V}}{12} = 0.25 \text{ A} = 250 \text{ mA}$	$R_3$	$V_{R_3} = I_{R_3} \times R_3 = 0.25 \times 8 = 2 \text{ V}$	$I_{R_3} = I_{R_2} = 0.25 \text{ A} = 250 \text{ mA}$	$R_4$	$V_{R_4} = V_1 - V_3 = 2.5 \text{ V}$	$I_{R_4} = \frac{V_{R_4}}{R_4} = \frac{2.5 \text{ V}}{10 \Omega} = 250 \text{ mA} = 0.25 \text{ A}$	$R_5$	$V_{R_5} = V_3 = 0.5 \text{ V}$	$I_{R_5} = \frac{V_{R_5}}{R_5} = \frac{0.5}{4} = 0.125 \text{ A} = 125 \text{ mA}$	$R_6$	$V_{R_6} = I_{R_6} \times R_6 = 0.25 \text{ V}$	$I_{R_6} = \frac{V_{R_6}}{(R_6 + R_7)} = \frac{0.25}{4} = 125 \text{ mA}$	$R_7$	$V_{R_7} = I_{R_7} \times R_7 = 0.25 \text{ V}$	$I_{R_7} = I_{R_6} = 125 \text{ mA}$
Across Resistor	Voltage (volt)	Current																								
$R_1$	$5 - V_1 = 2 \text{ V}$	$I_{R_1} = \frac{V_{R_1}}{(R_1)} = \frac{2 \text{ V}}{4 \Omega} = 0.5 \text{ A} = 500 \text{ mA}$																								
$R_2$	$V_{R_2} = I_{R_2} \times R_2 = (0.25 \text{ A}) \times 4 = 1 \text{ V}$	$I_{R_2} = \frac{V_{R_2}}{(R_2 + R_3)} = \frac{1 \text{ V}}{12} = 0.25 \text{ A} = 250 \text{ mA}$																								
$R_3$	$V_{R_3} = I_{R_3} \times R_3 = 0.25 \times 8 = 2 \text{ V}$	$I_{R_3} = I_{R_2} = 0.25 \text{ A} = 250 \text{ mA}$																								
$R_4$	$V_{R_4} = V_1 - V_3 = 2.5 \text{ V}$	$I_{R_4} = \frac{V_{R_4}}{R_4} = \frac{2.5 \text{ V}}{10 \Omega} = 250 \text{ mA} = 0.25 \text{ A}$																								
$R_5$	$V_{R_5} = V_3 = 0.5 \text{ V}$	$I_{R_5} = \frac{V_{R_5}}{R_5} = \frac{0.5}{4} = 0.125 \text{ A} = 125 \text{ mA}$																								
$R_6$	$V_{R_6} = I_{R_6} \times R_6 = 0.25 \text{ V}$	$I_{R_6} = \frac{V_{R_6}}{(R_6 + R_7)} = \frac{0.25}{4} = 125 \text{ mA}$																								
$R_7$	$V_{R_7} = I_{R_7} \times R_7 = 0.25 \text{ V}$	$I_{R_7} = I_{R_6} = 125 \text{ mA}$																								



## d.) Final Result and Conclusion

Resistor	Voltage (V)		Current (mA)	
	Multism	Theoretical	Multism	Theoretical
R1	2	2	500	500
R2	1	1	250	250
R3	2	2	250	250
R4	2.5	2.5	250	250
R5	0.5	0.5	125	125
R6	0.25	0.25	125	125
R7	0.25	0.25	125	125

*Conclusion:*

We can observe from Above Table, Both the *Theoretical* and *Multisim* Values of Current and Voltage are **Equal**.

Hence, Experiment is Performed Successfully (without any Error).



**Expt. No:**

2

**Date:**

20-08-2020

## 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.

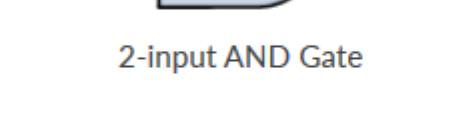
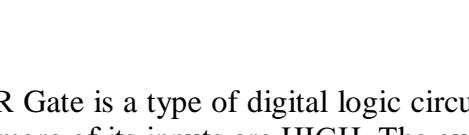
The **Digital Logic Gate** is the basic building block from which all digital electronic circuits and microprocessor based systems are constructed. Basic digital logic gates perform logical operations of AND, OR and NOT on binary numbers. In digital logic design only two voltage levels or states are 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

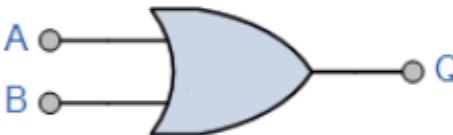
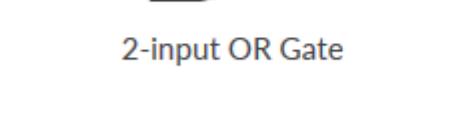
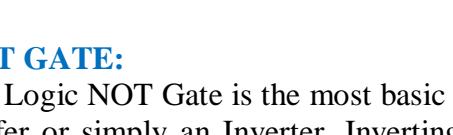


returns “LOW” again when **ANY** of its inputs are at a logic level “0”. In other words for a logic AND gate, any LOW input will give a LOW output.

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	
 Inverter or NOT Gate	A	Q
	0	1
	1	0

### NAND GATE:

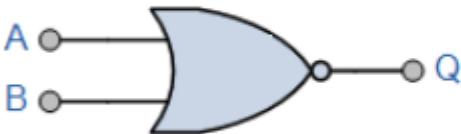
The Logic NAND Gate is a combination of a digital logic AND gate and a NOT gate connected together in series. The NAND (Not – AND) gate has an output that is normally at logic level “1” and only goes “LOW” to logic level “0” when **ALL** of its inputs are at logic level “1”.

Symbol	Truth Table		
 2-input NAND Gate	A	B	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0

### NOR GATE:

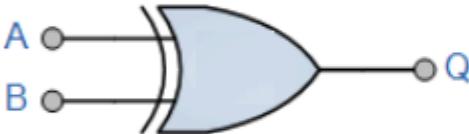
The Logic NOR Gate gate is a combination of the digital logic OR gate and an inverter or NOT gate connected together in series. The inclusive NOR (Not-OR) gate has an output that is normally at logic level “1” and only goes “LOW” to logic level “0” when **ANY** of its inputs are at logic level “1”.



Symbol	Truth Table		
	A	B	Q
 2-input NOR Gate	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
 2-input Ex-OR Gate	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**.

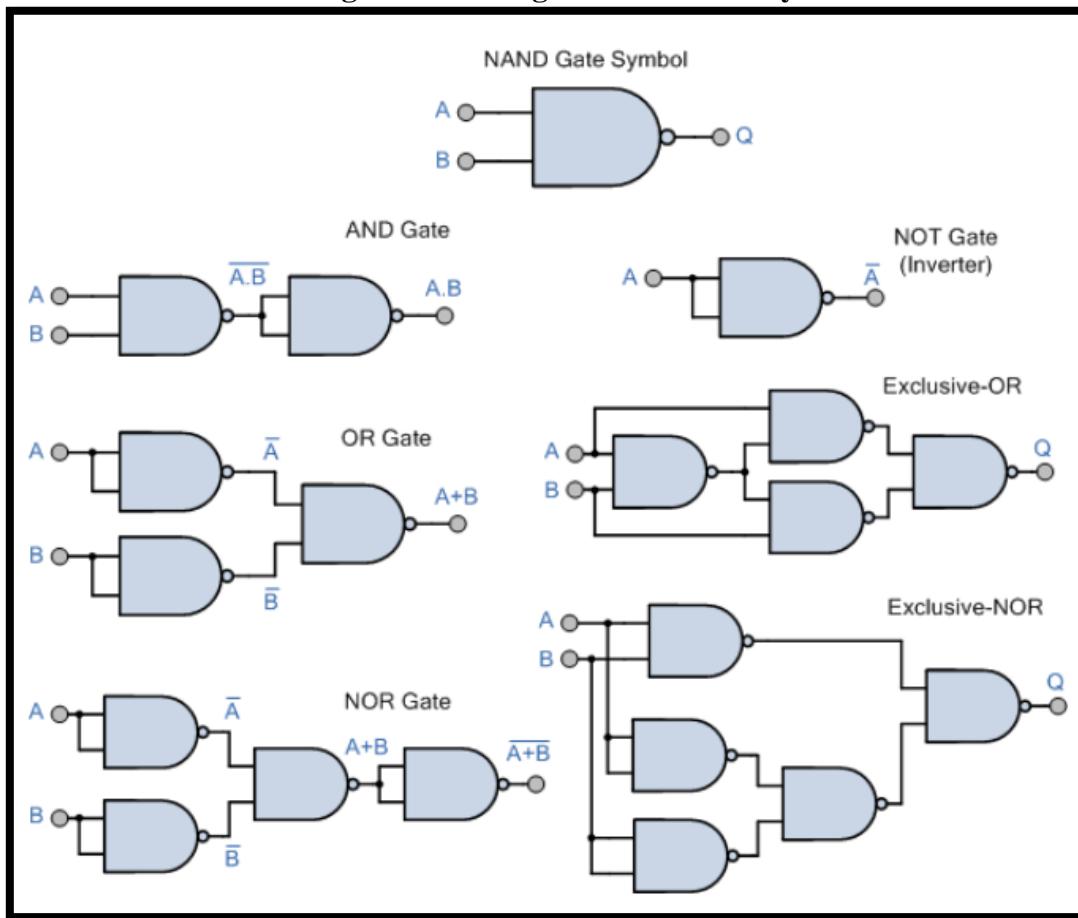


Symbol	Truth Table		
	A	B	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1

### UNIVERSAL GATES:

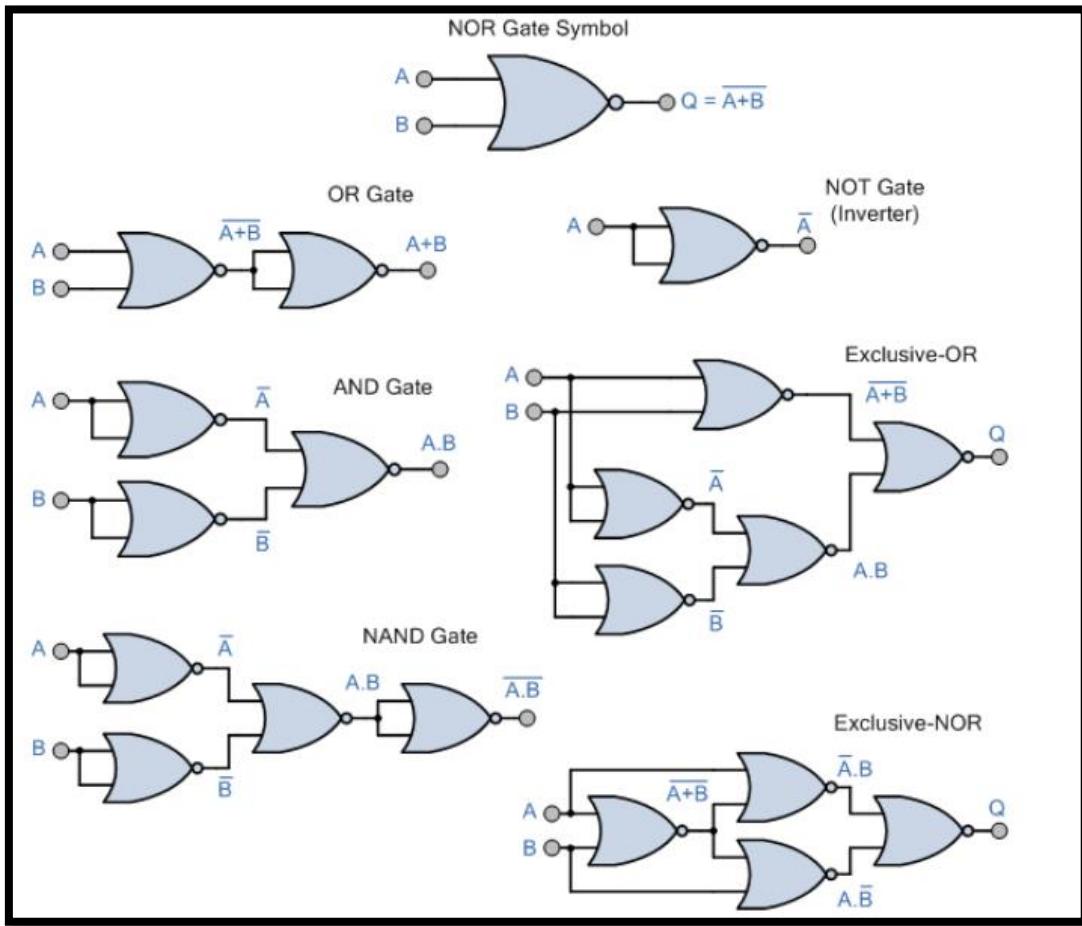
Universal Logic gates can be used to produce any other logic or Boolean function with the NAND and NOR gates. Thus ALL other logic gate functions can be created using only NAND/NOR gates making them universal logic gates.

#### Logic Gates using NAND Gate Only:





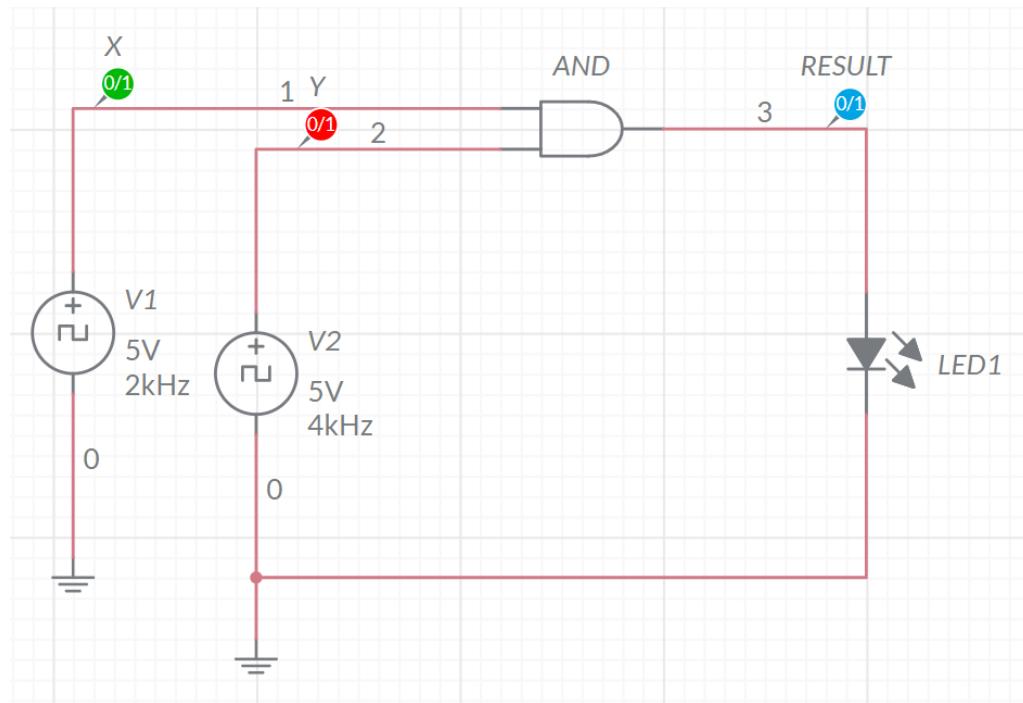
**Logic Gates using NOR Gate Only:**



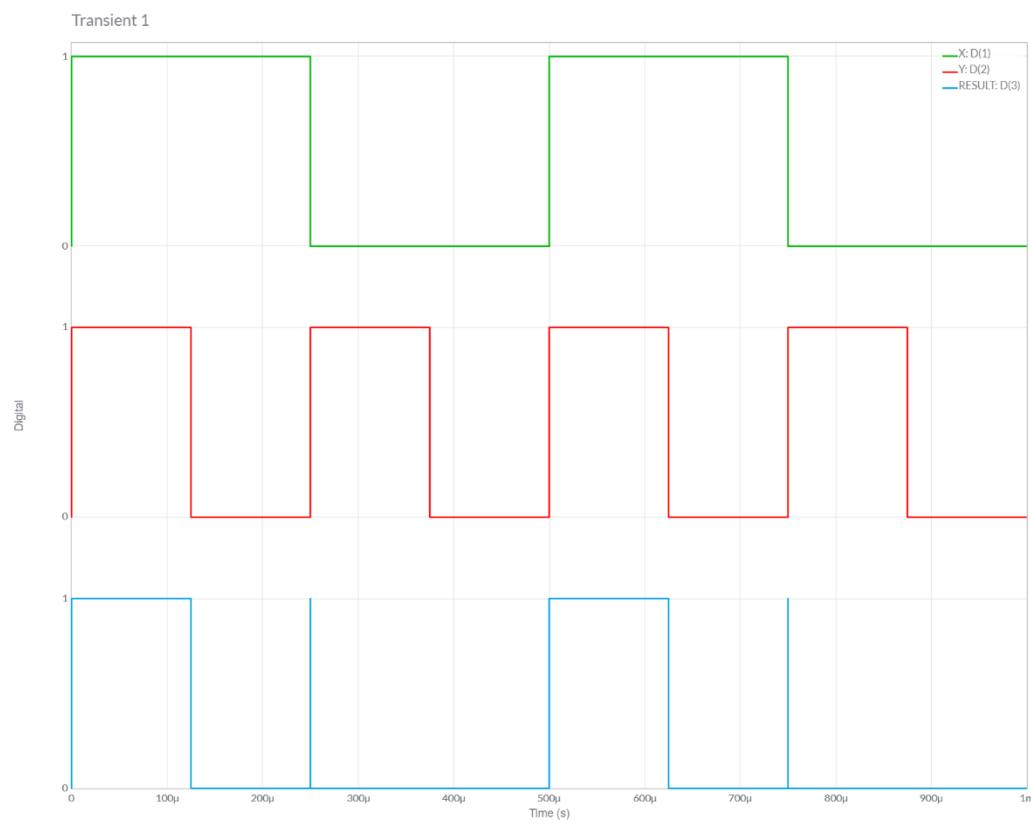


### 1.) 2-INPUT AND

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



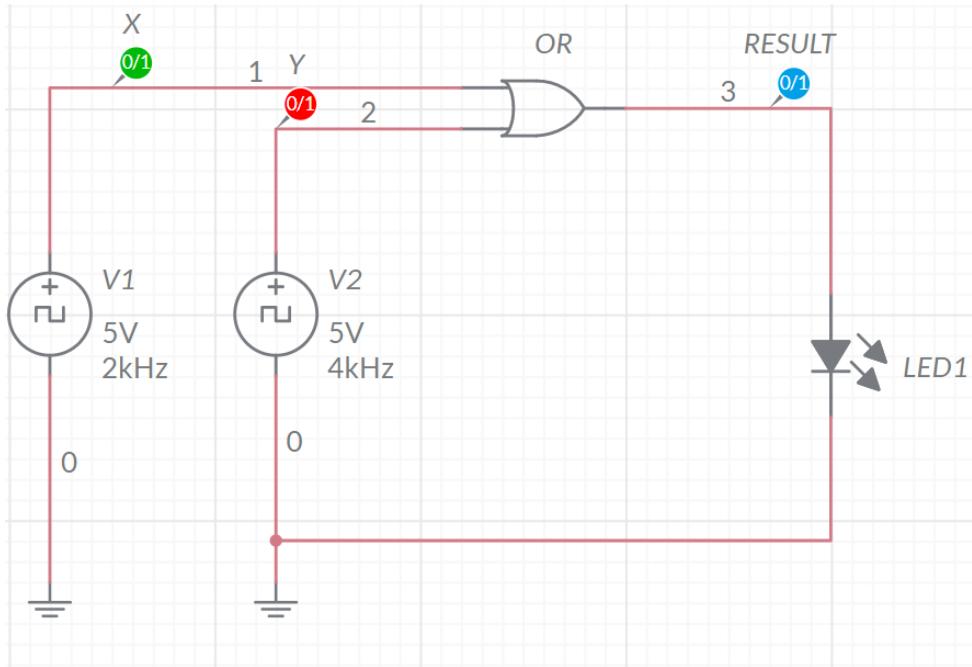
#### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



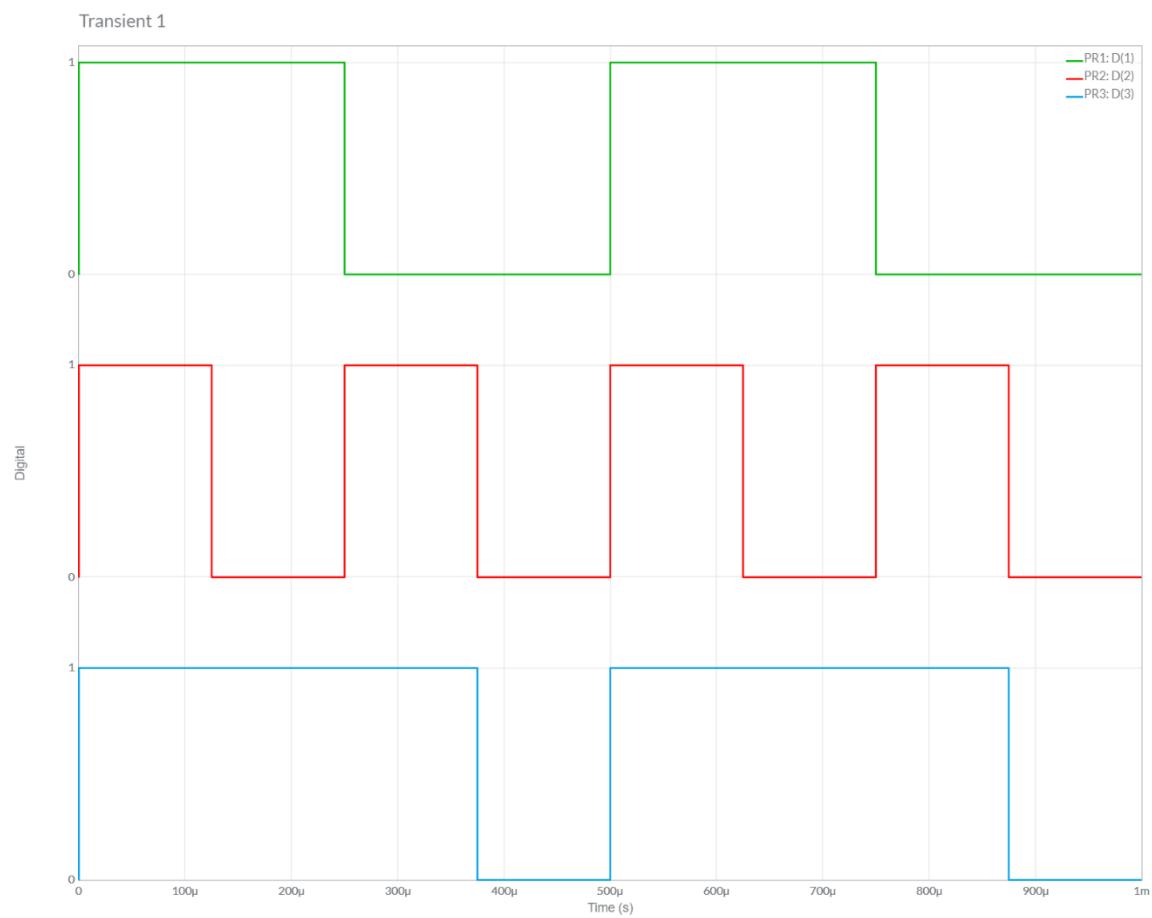


## 2.) 2-INPUT OR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



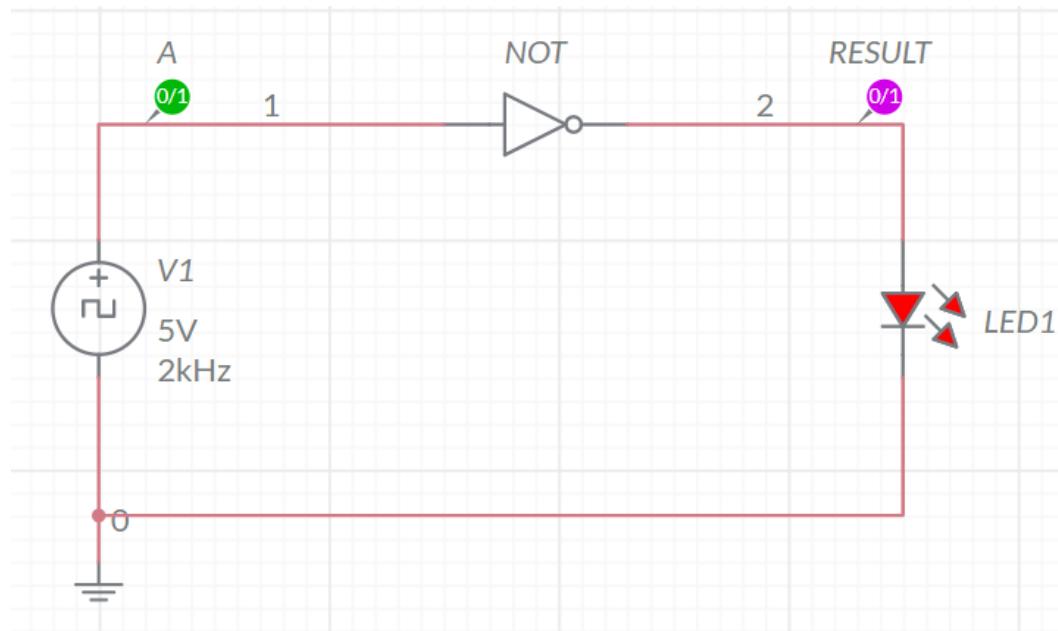
## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



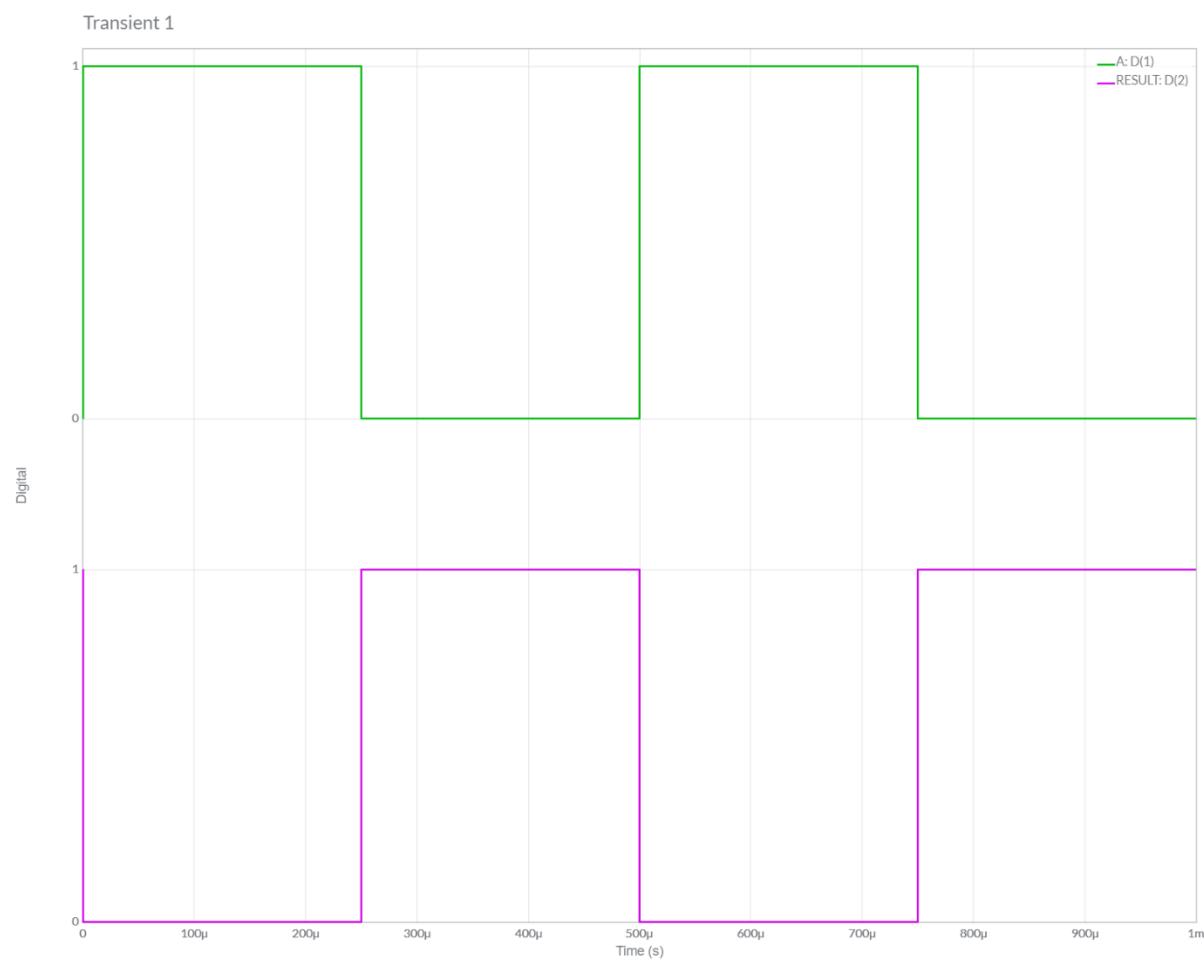


### 3.) NOT

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



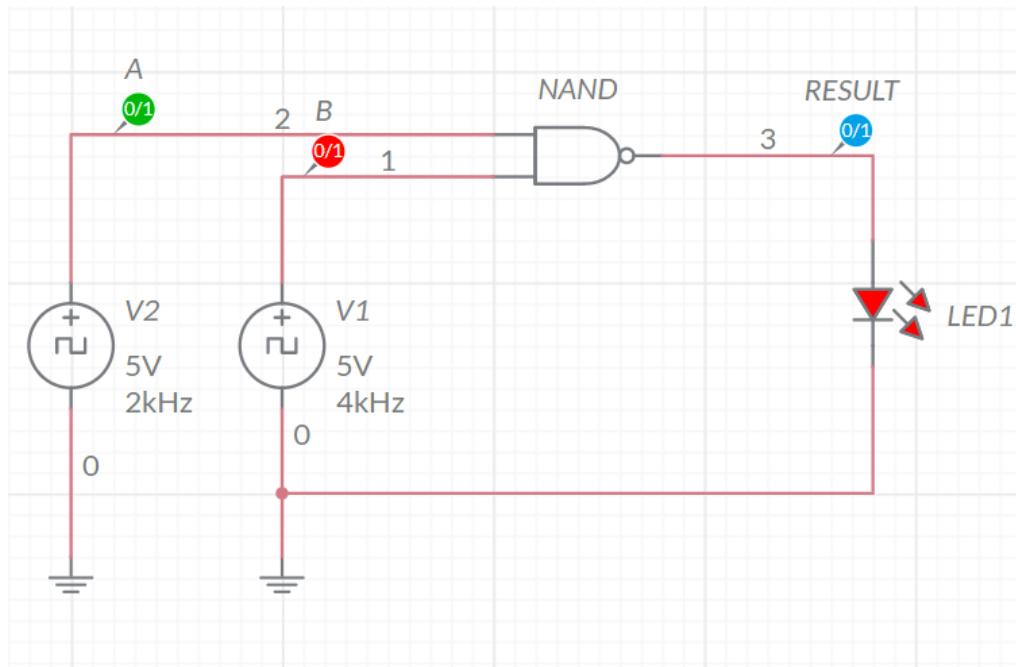
#### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



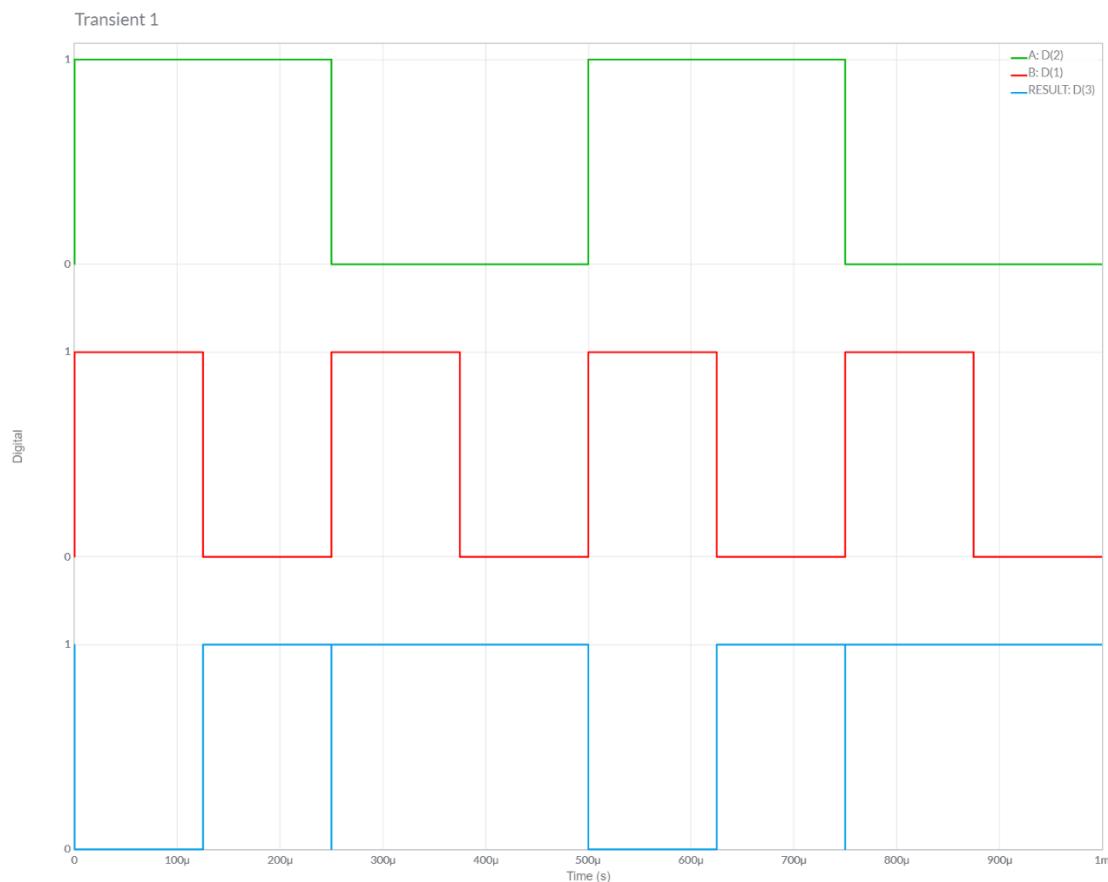


#### 4.) 2-INPUT NAND

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



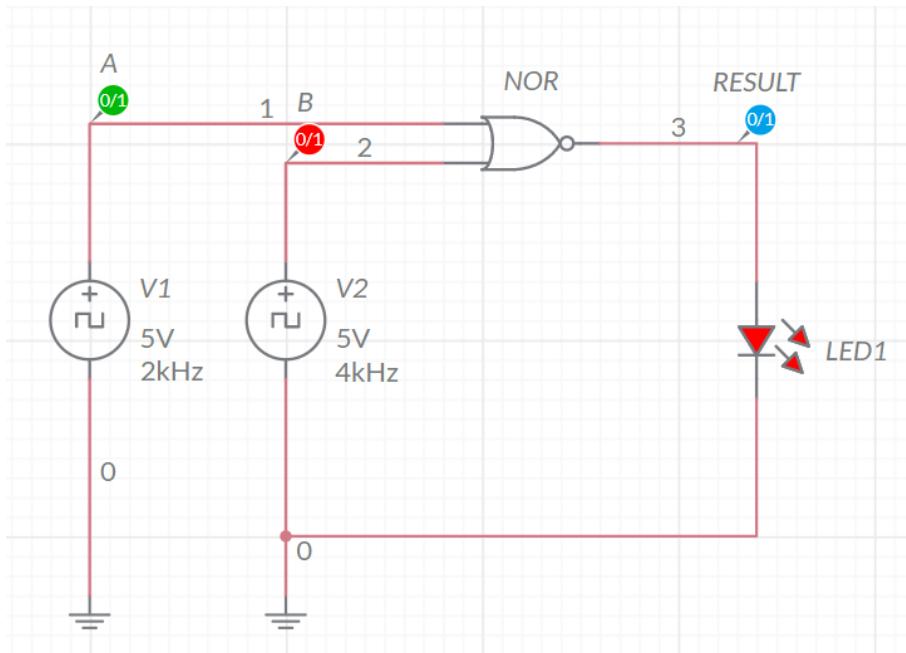
#### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



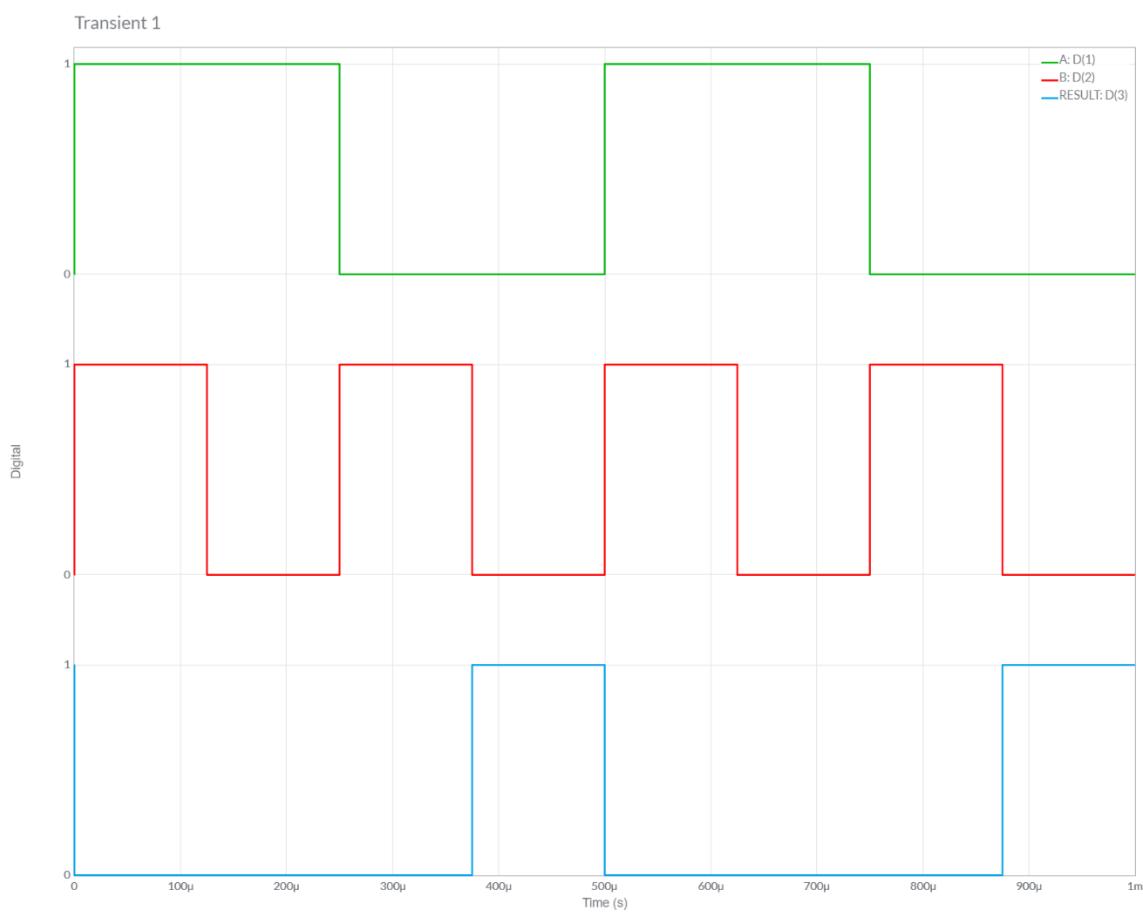


### 5.) 2-INPUT NOR

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



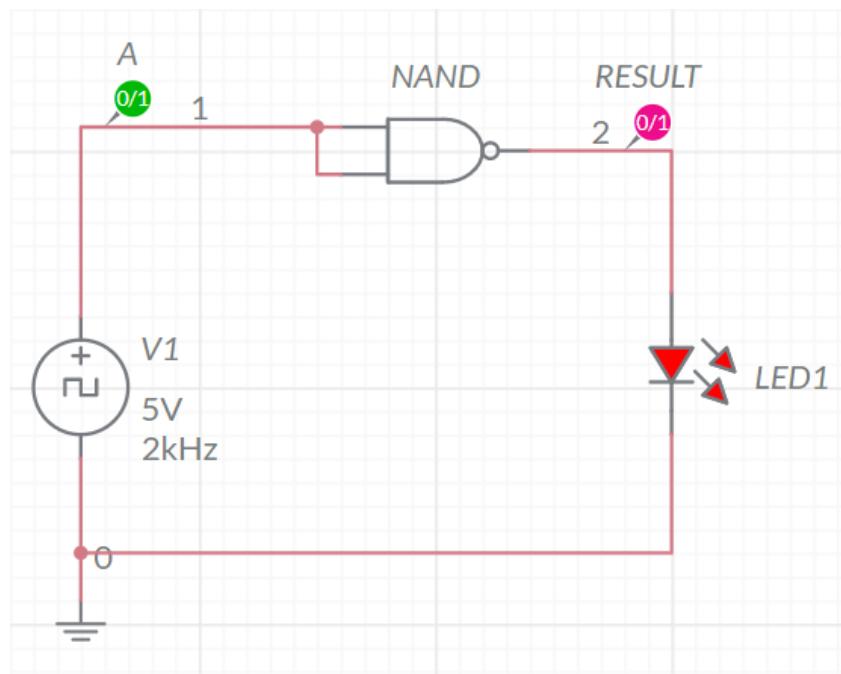
#### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



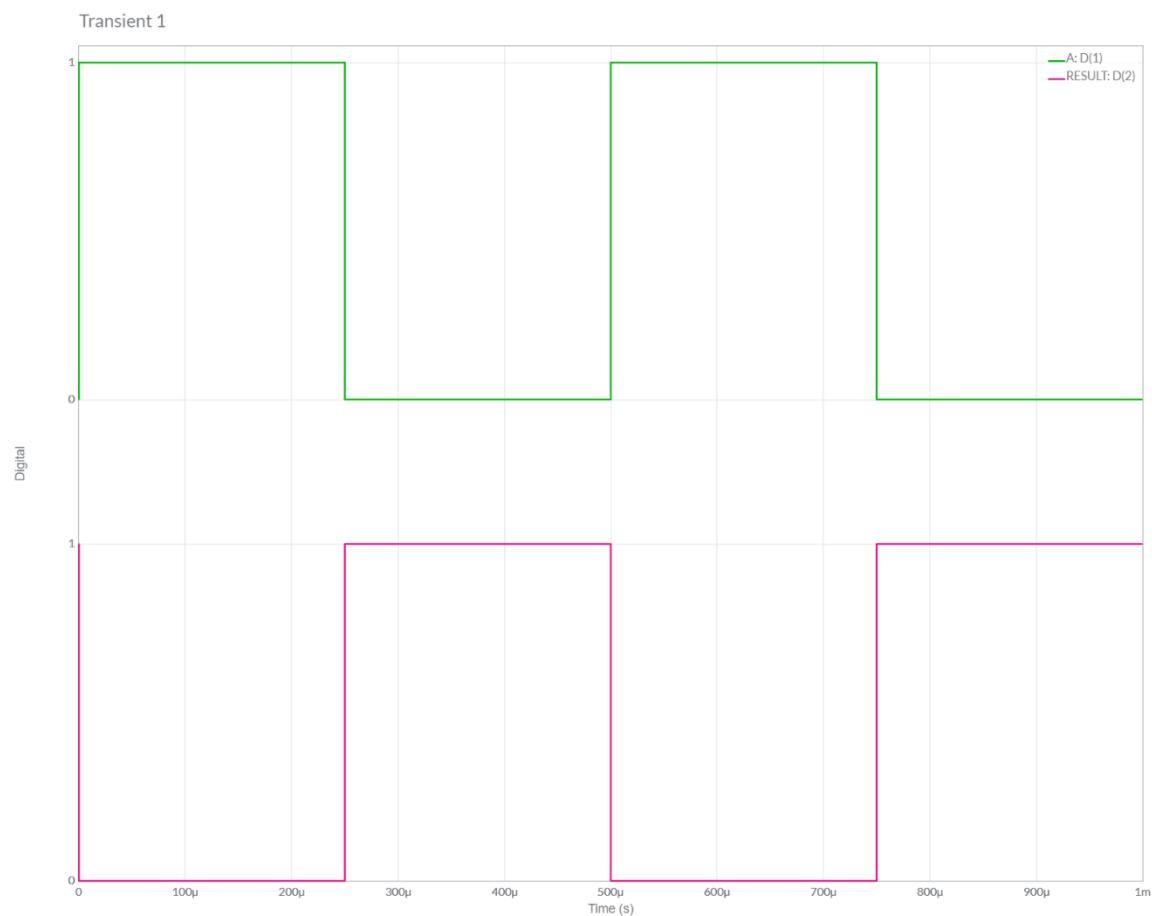


### 6.) CREATE INVERTOR USING NAND

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



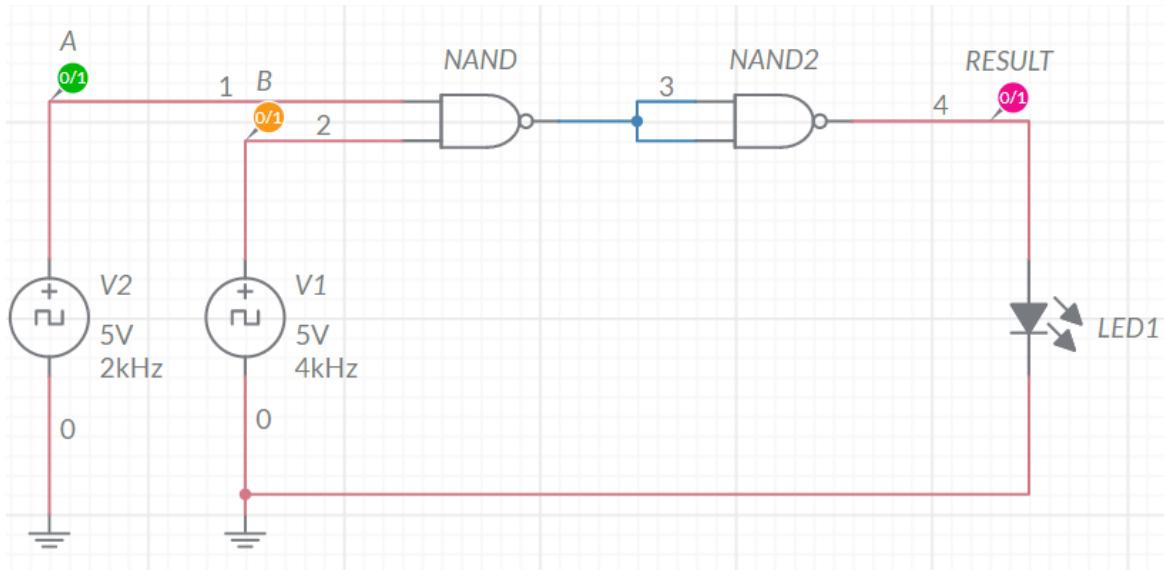
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



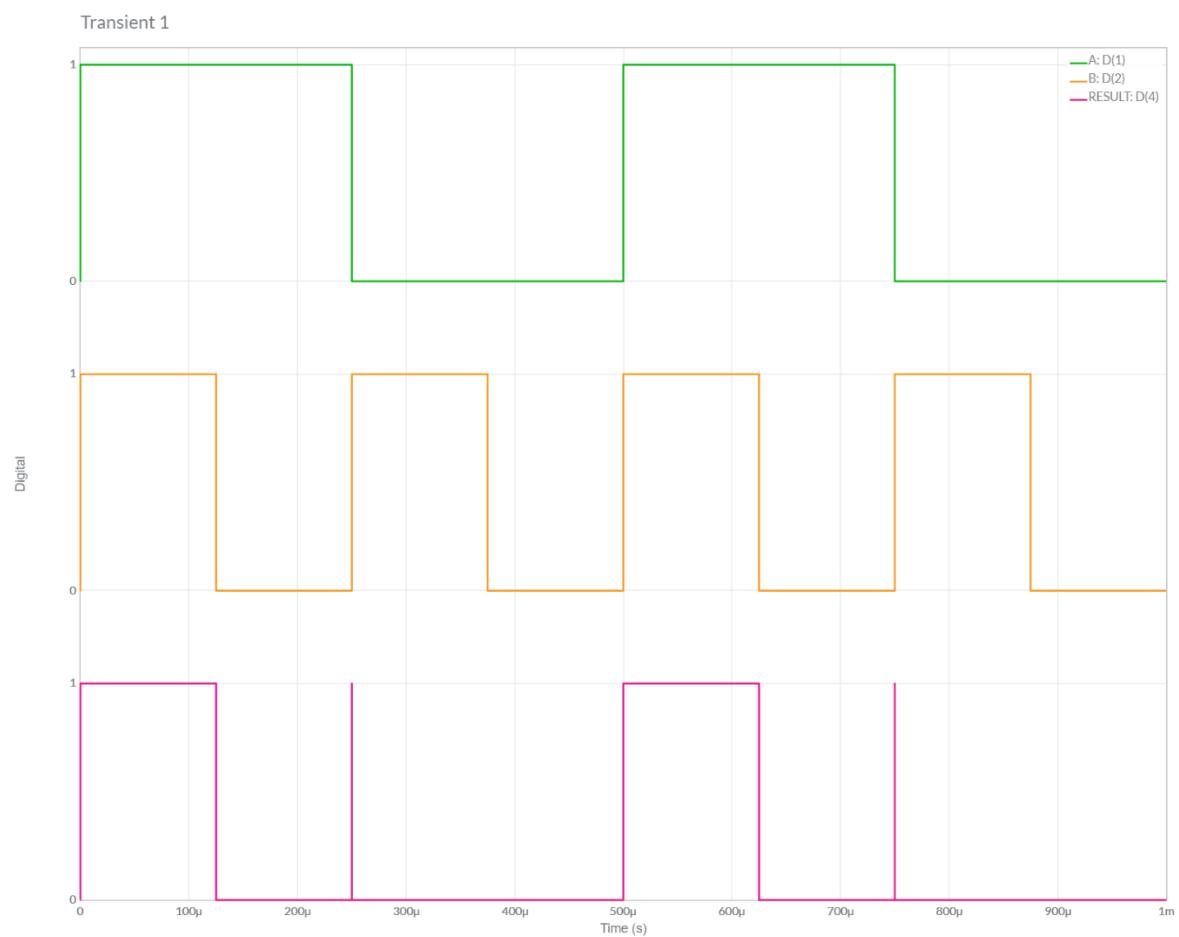


## 7.) CREATE AND USING NAND

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



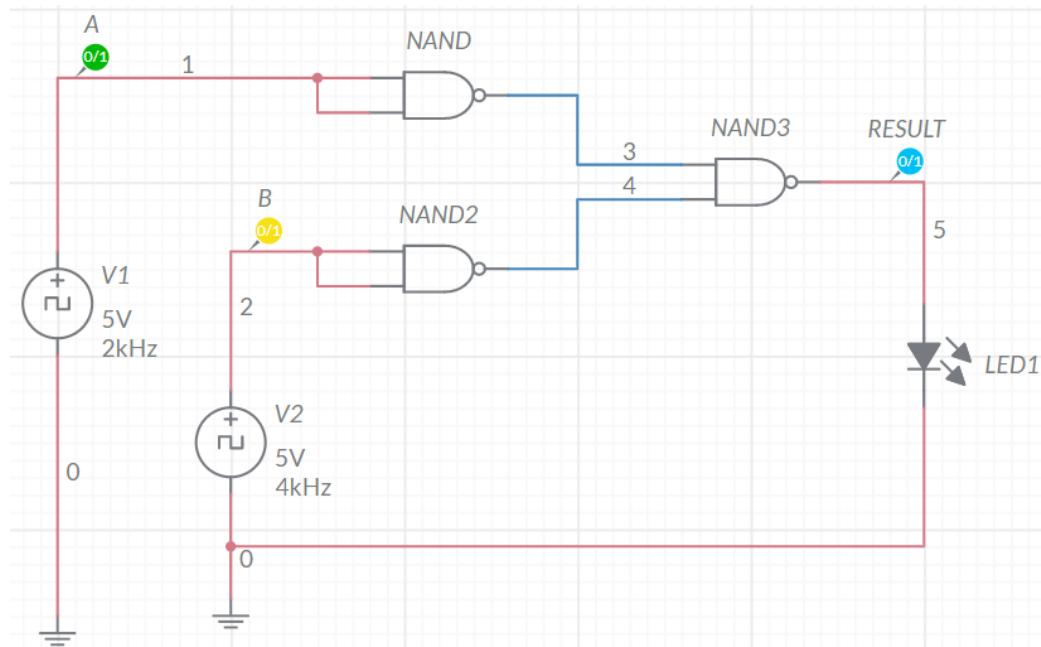
### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



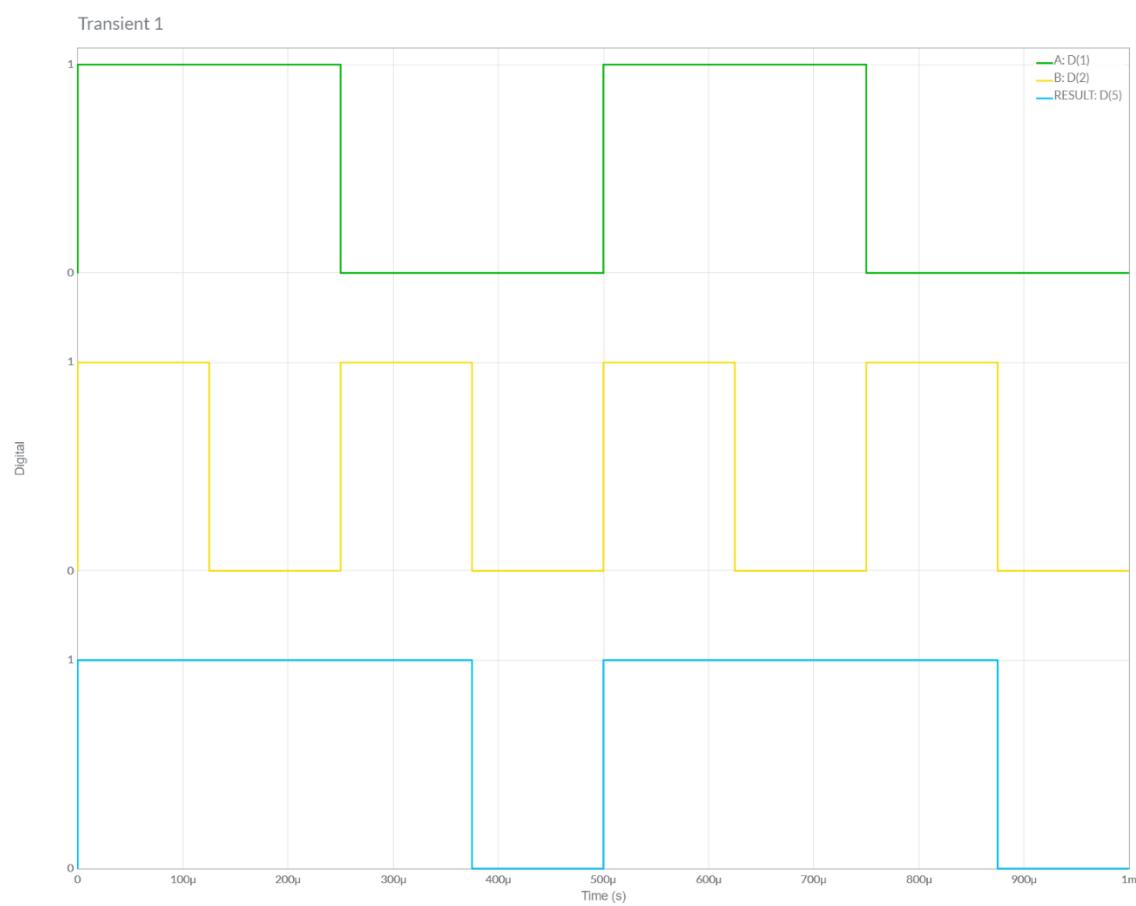


## 8.) CREATE OR USING NAND

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



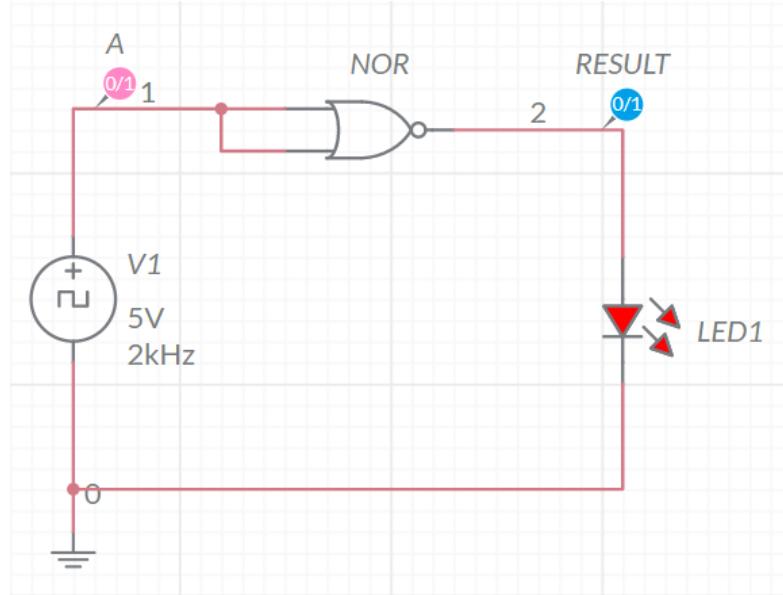
### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



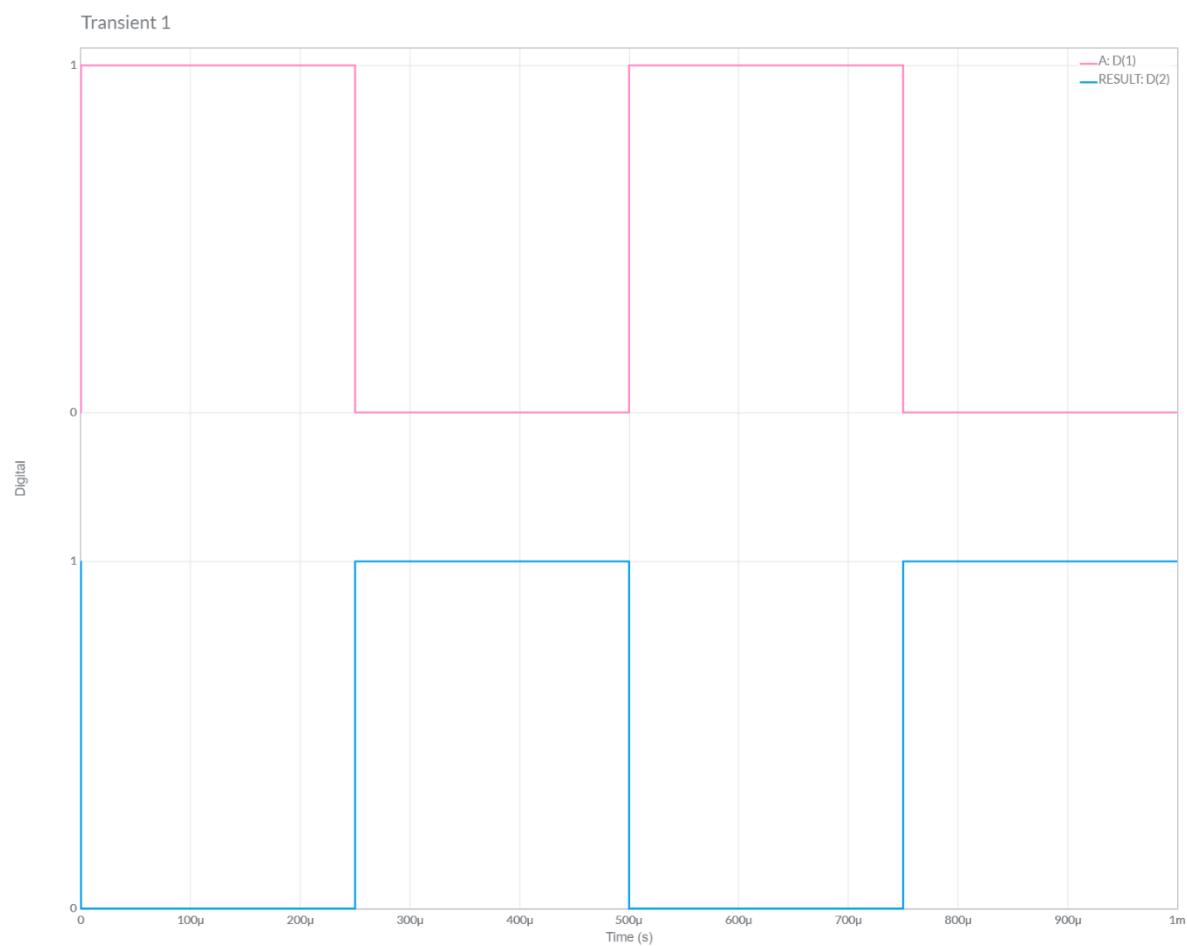


**9.) CREATE INVERTOR USING NOR**

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



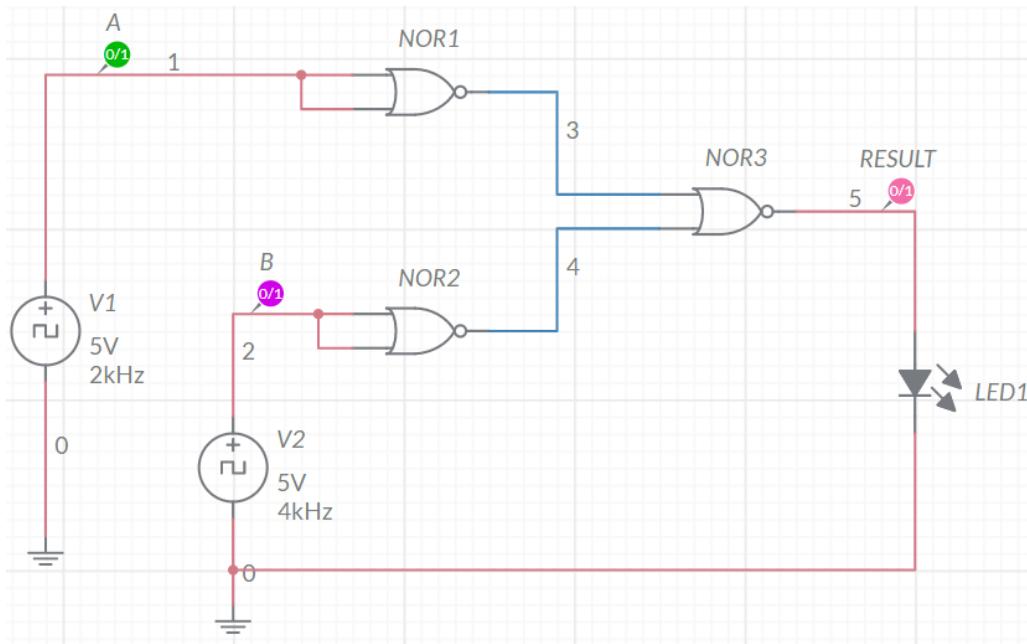
**OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):**



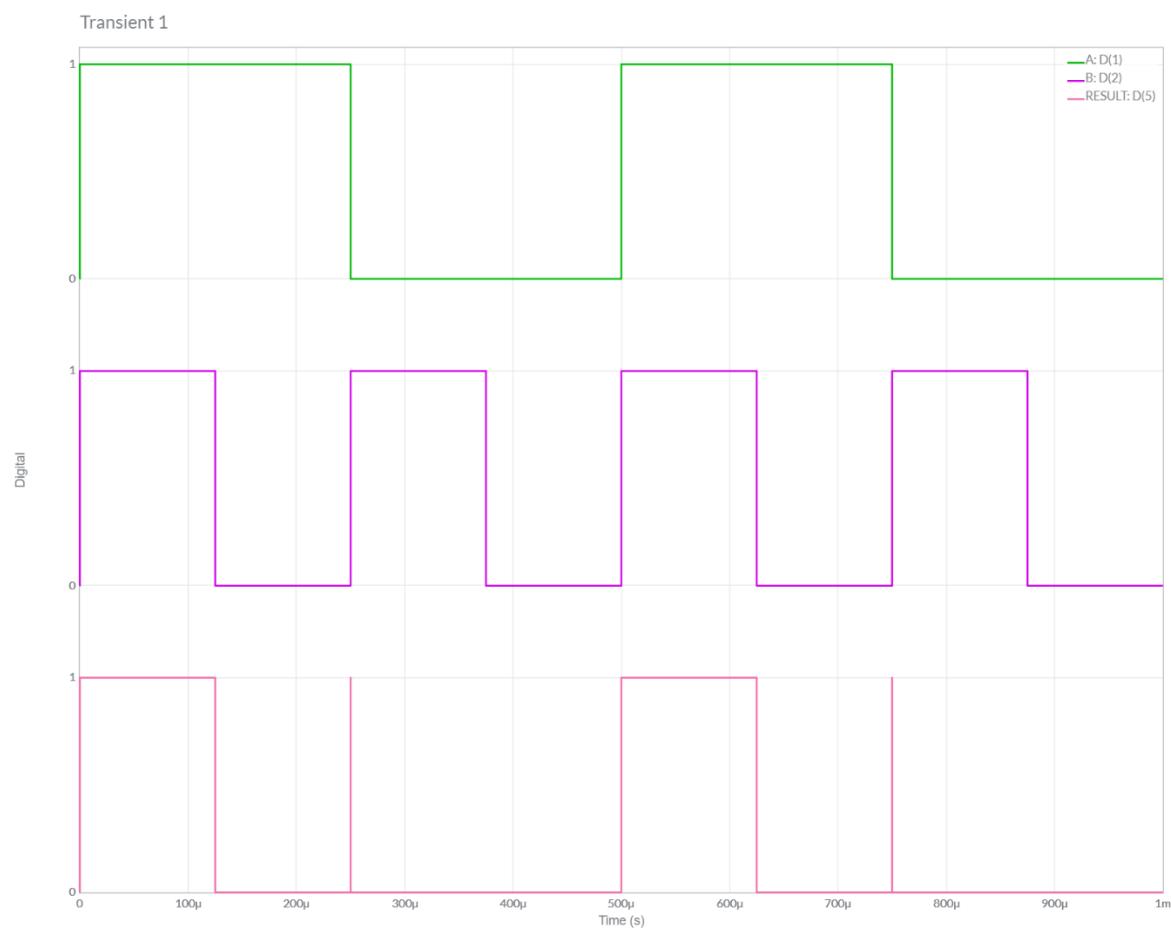


## 10.) CREATE AND USING NOR

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



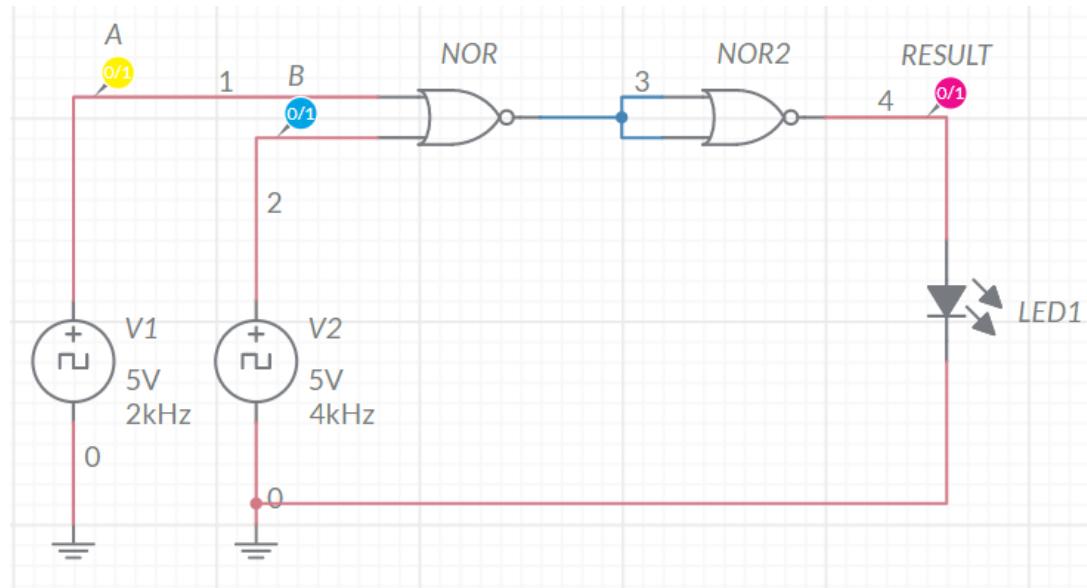
### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



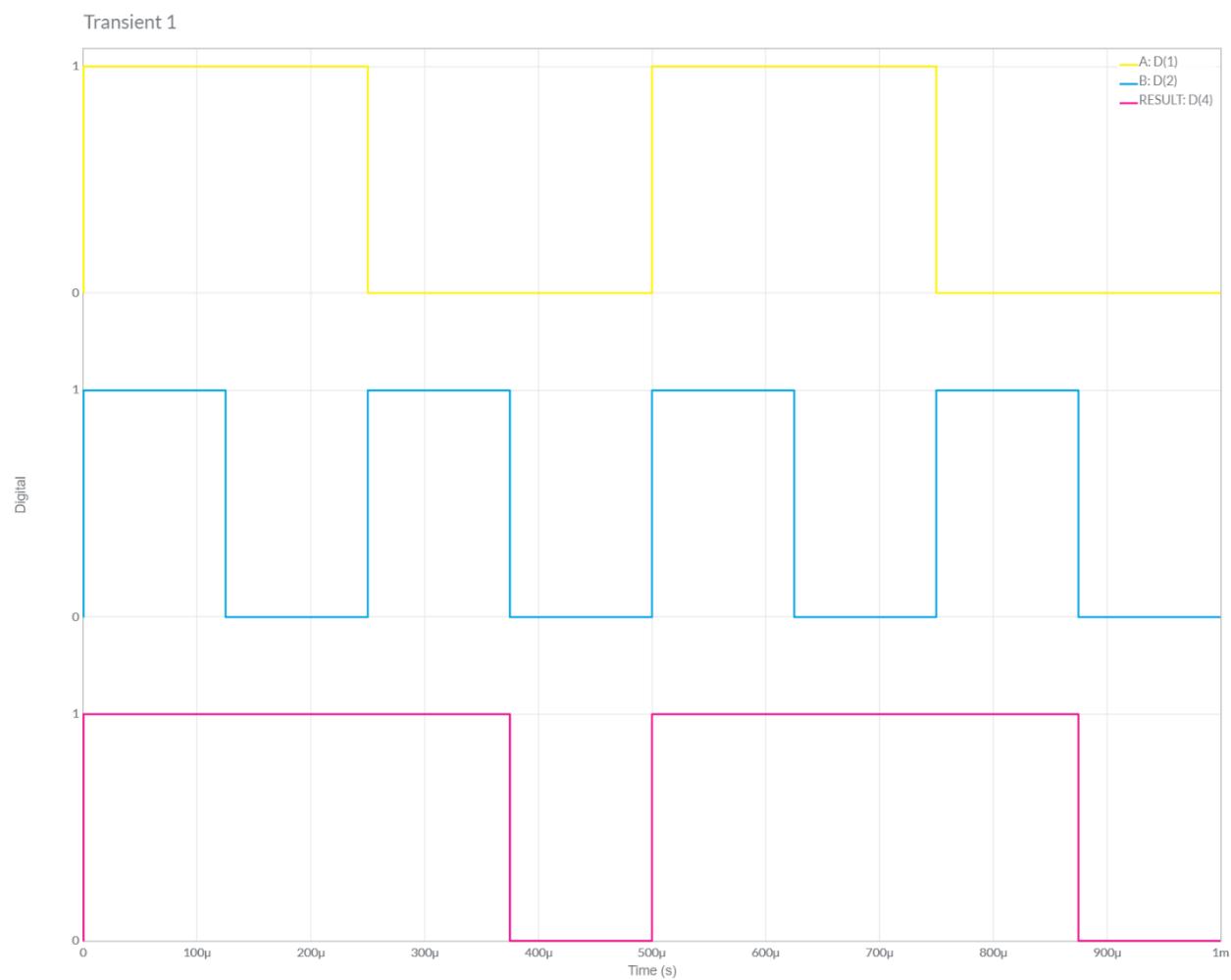


## 11.) CREATE OR USING NOR

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



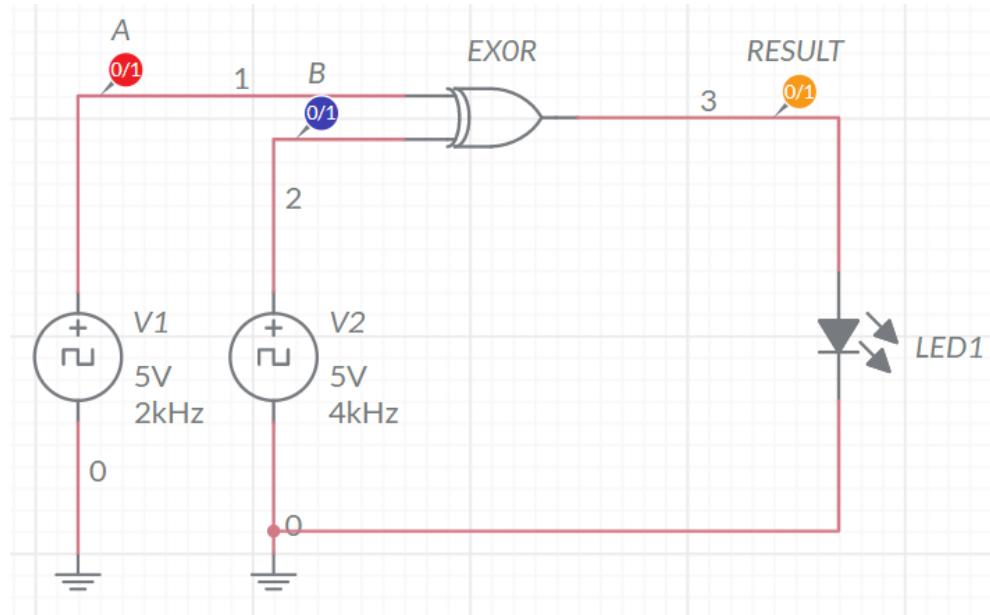
### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



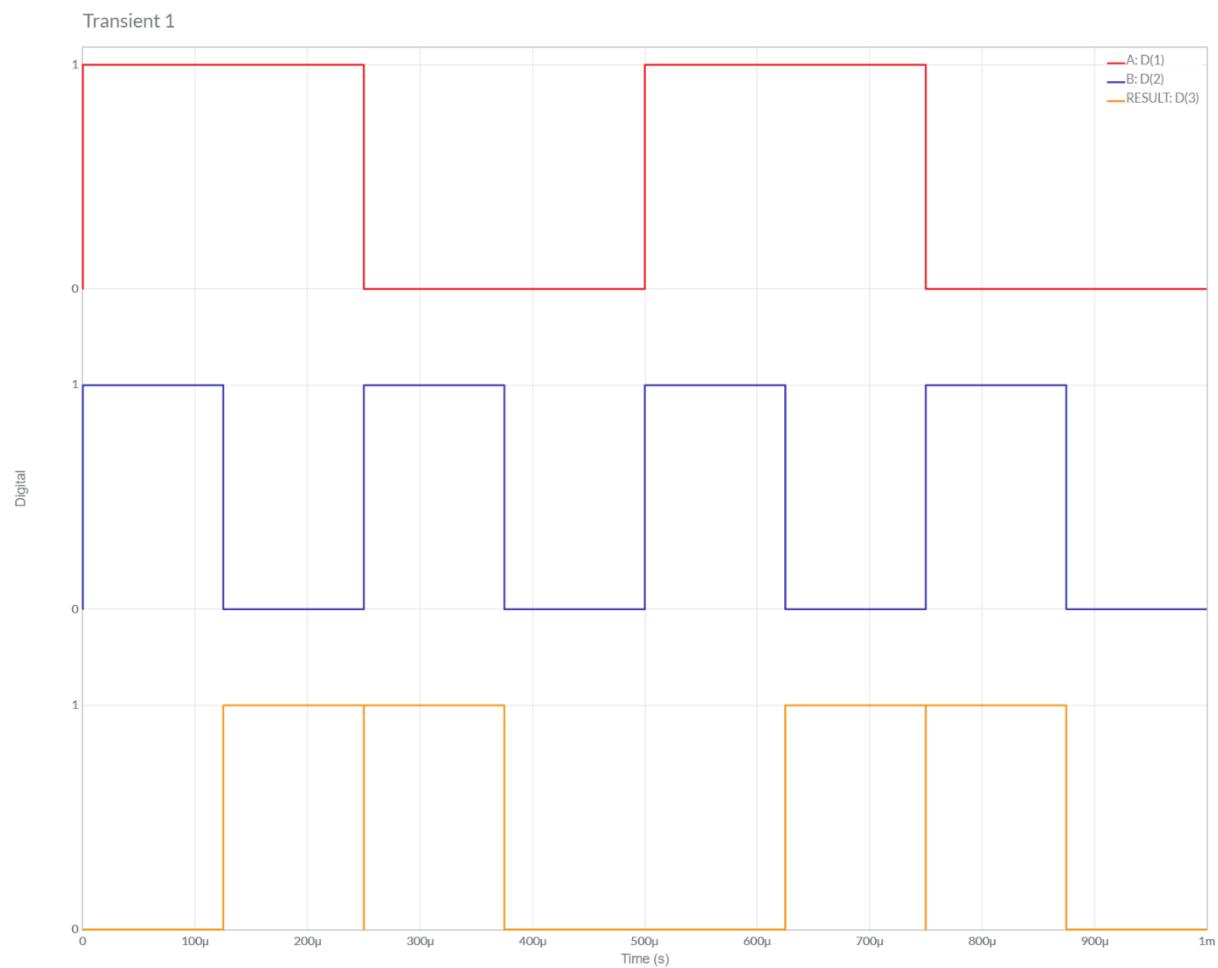


## 12.) 2-INPUT EX-OR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



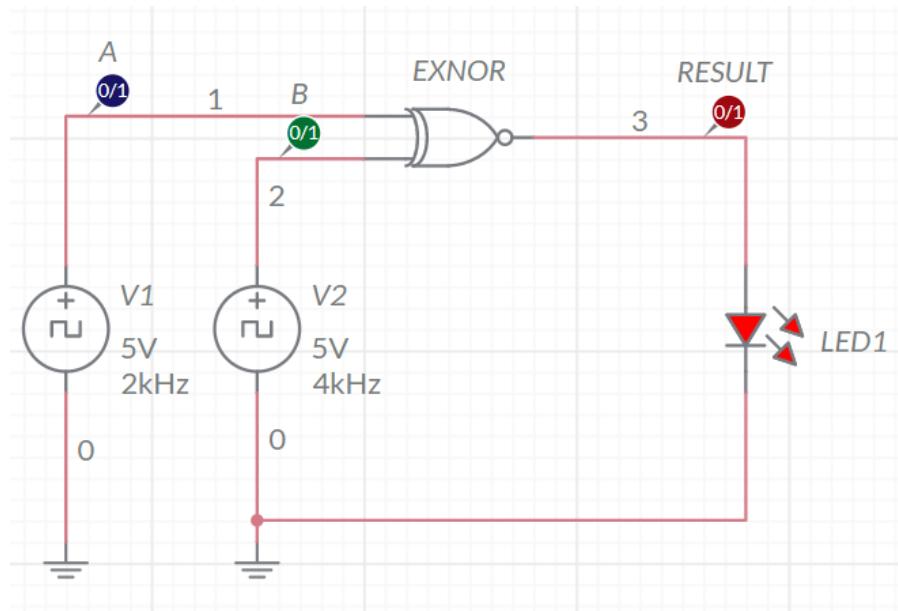
## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



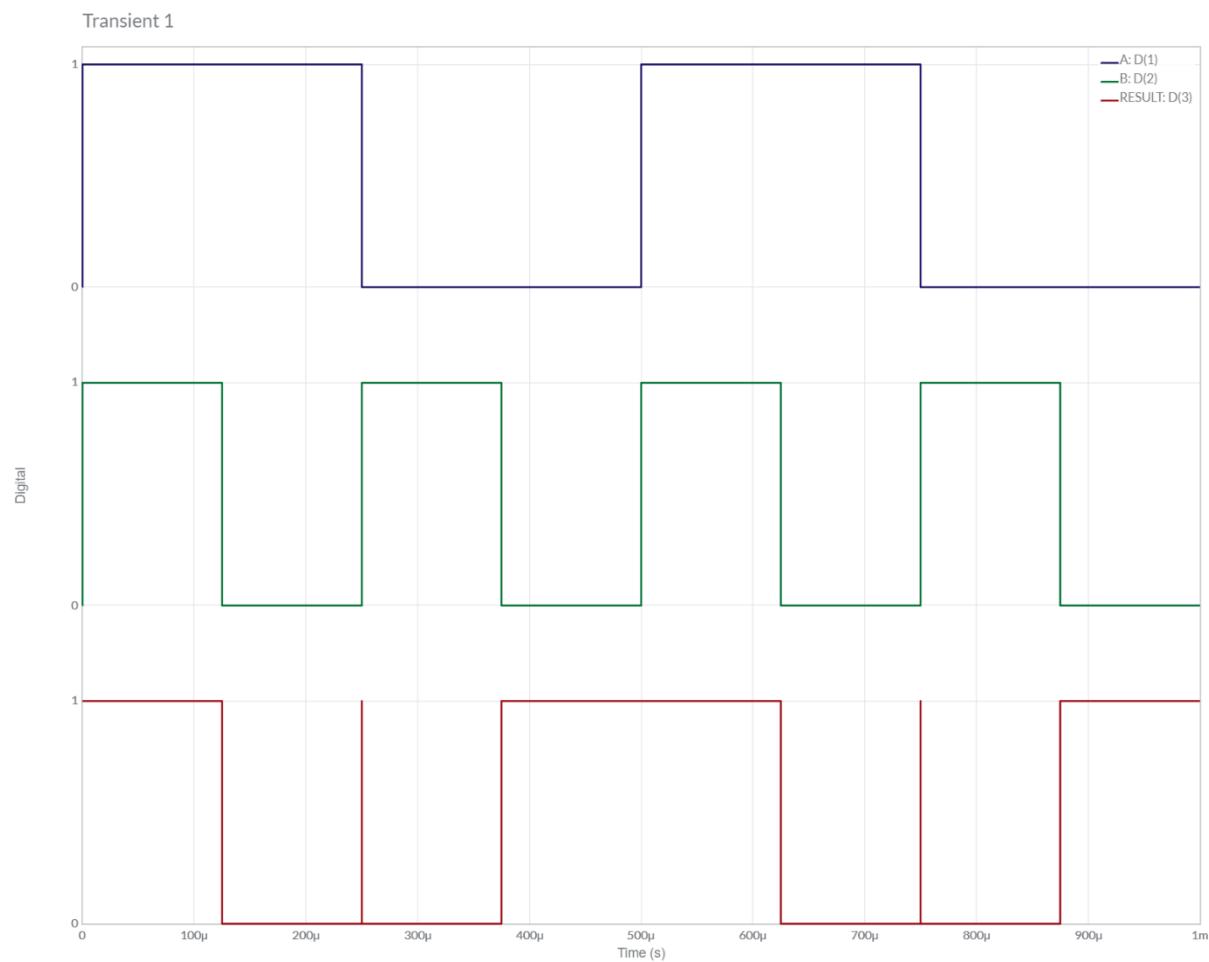


### 13.) 2-INPUT EX-NOR

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



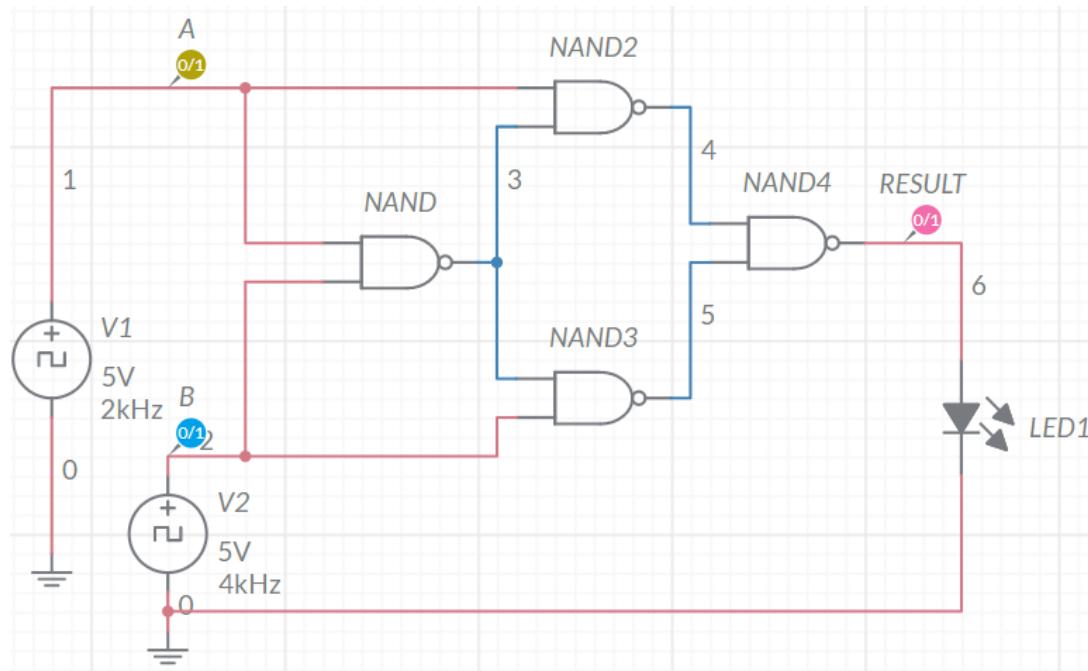
#### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



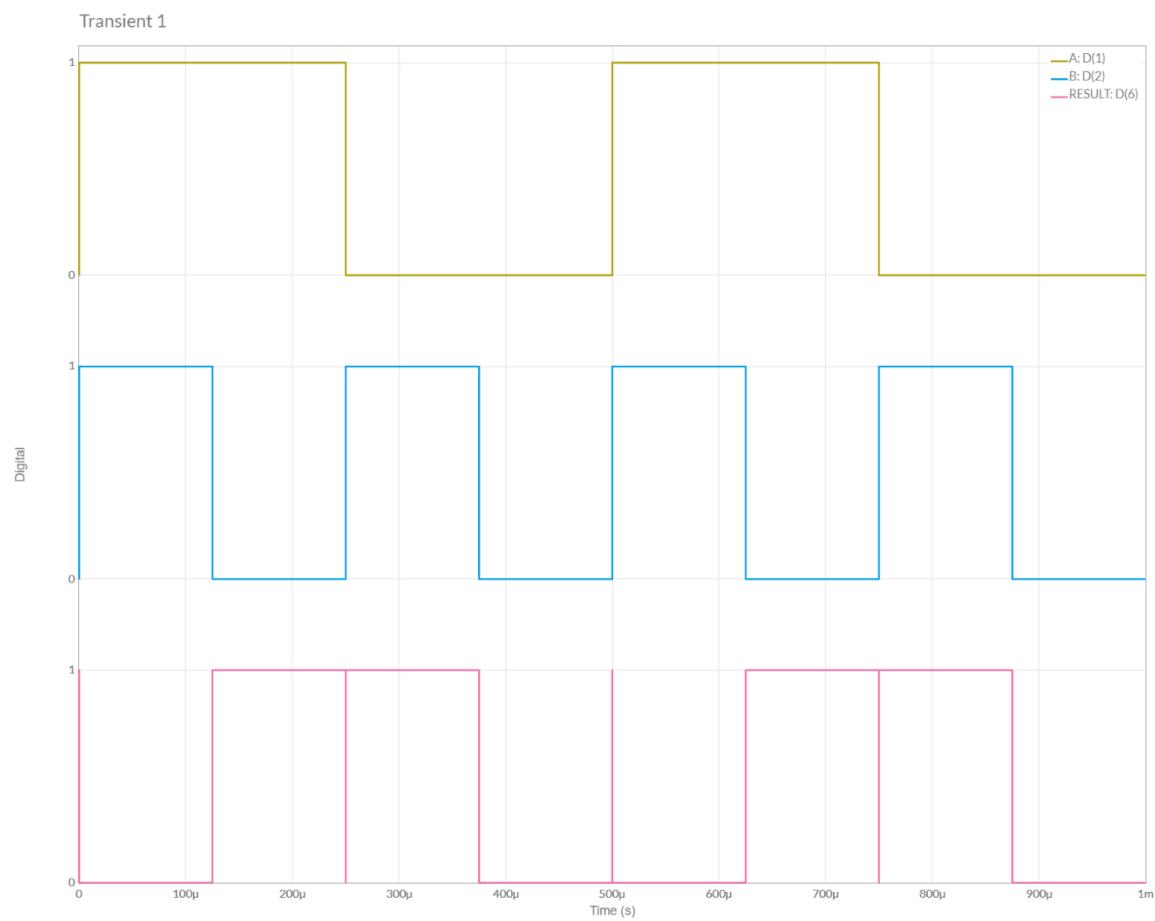


#### 14.) 2-INPUT EX-OR USING NAND

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



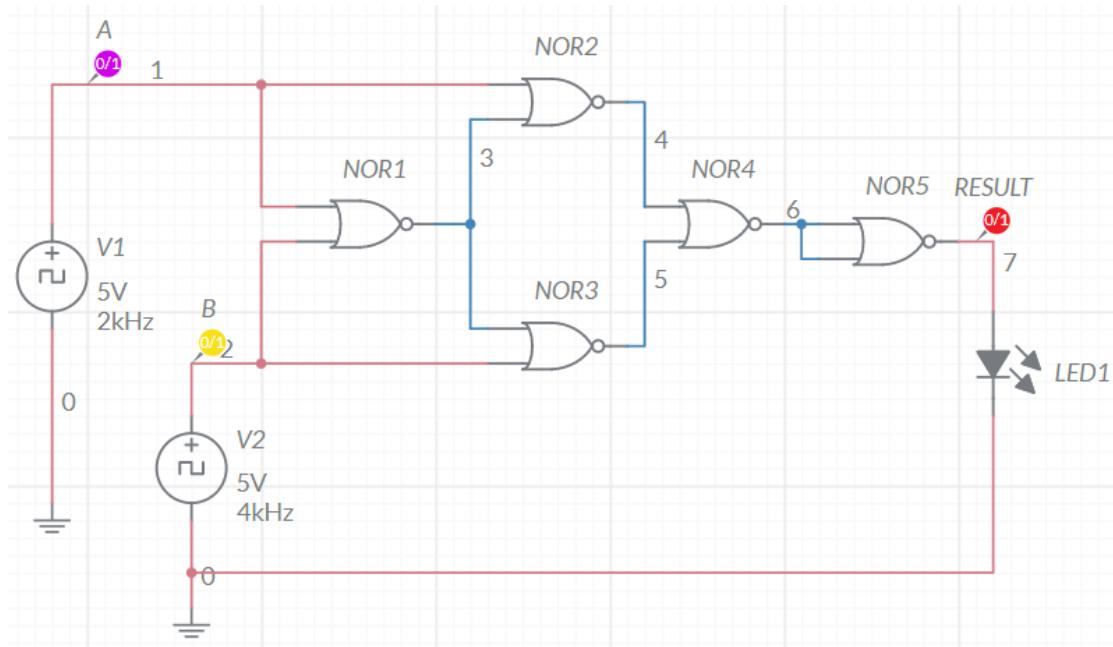
#### OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



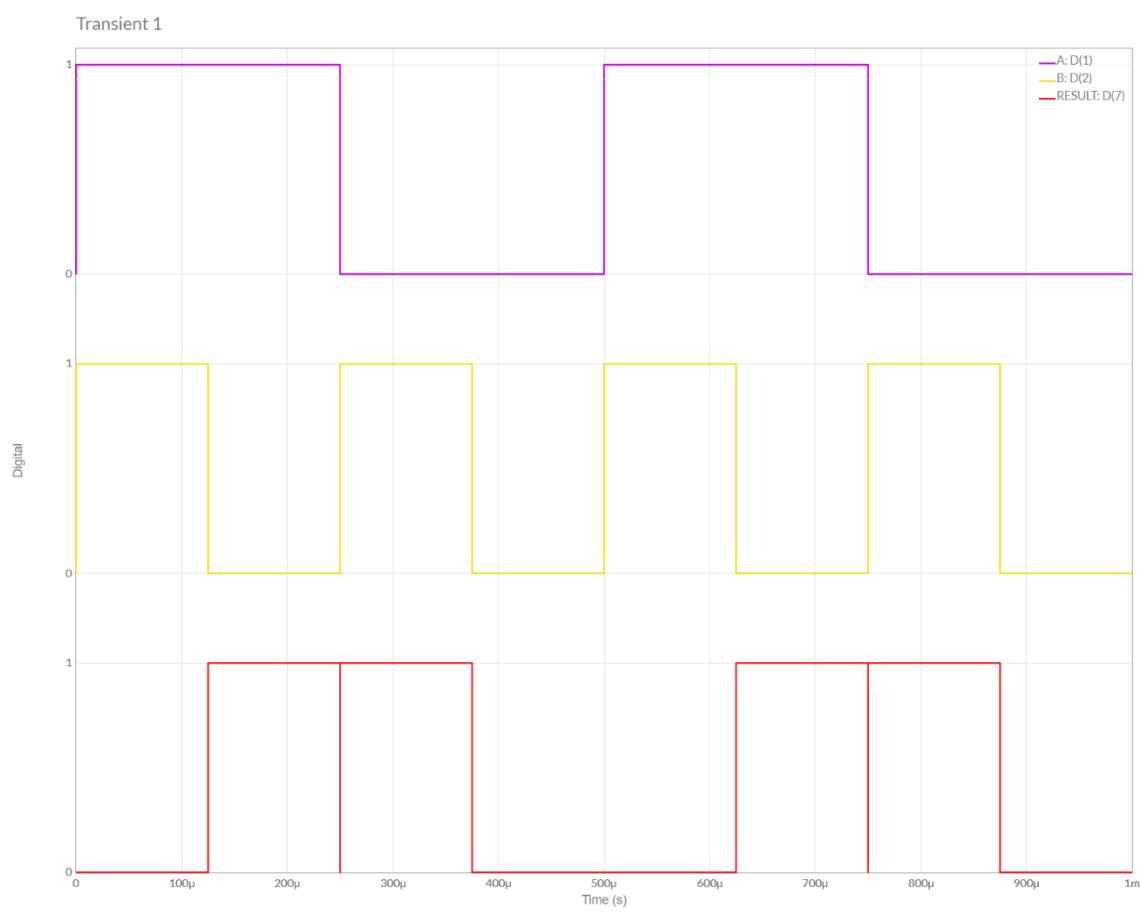


15.) 2-INPUT EX-OR USING NOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



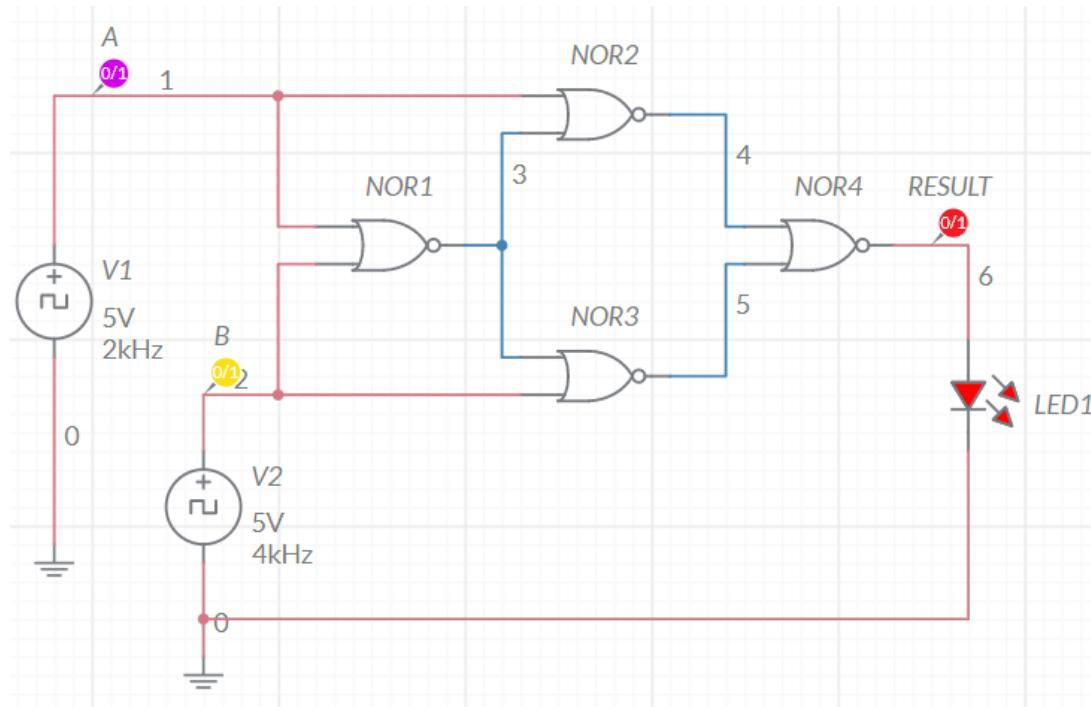
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



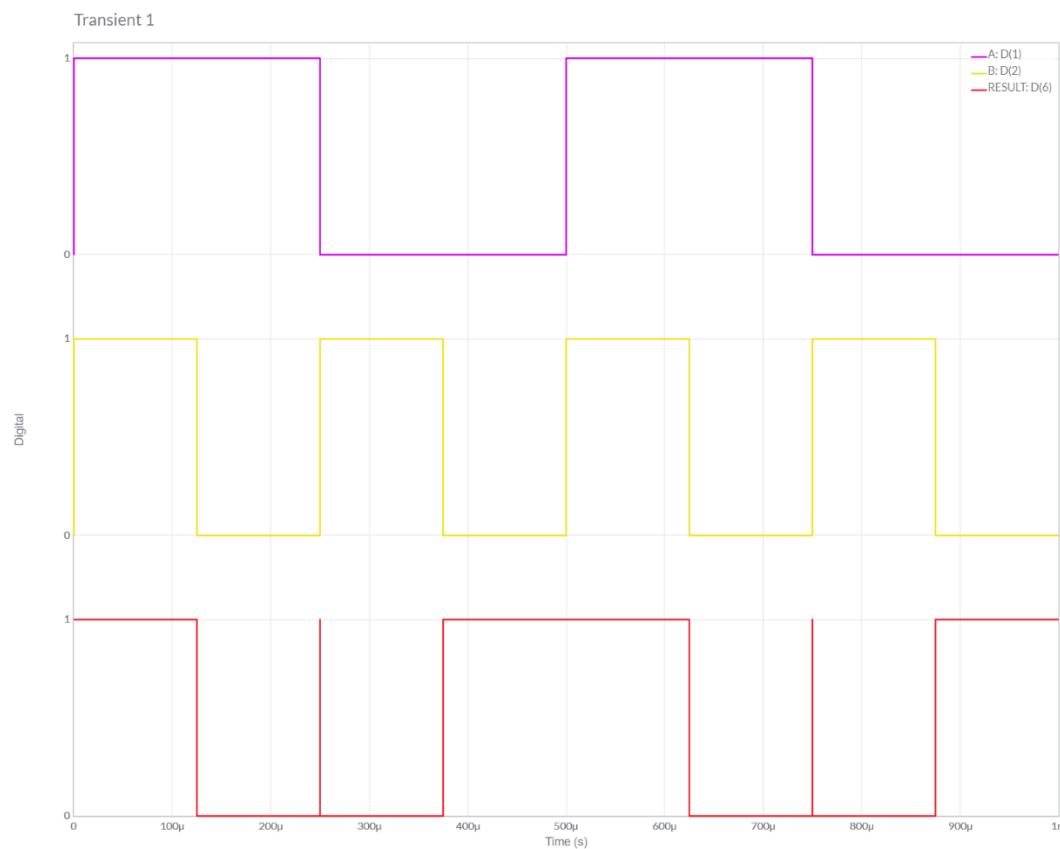


### 16.) 2-INPUT EX-NOR USING NOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



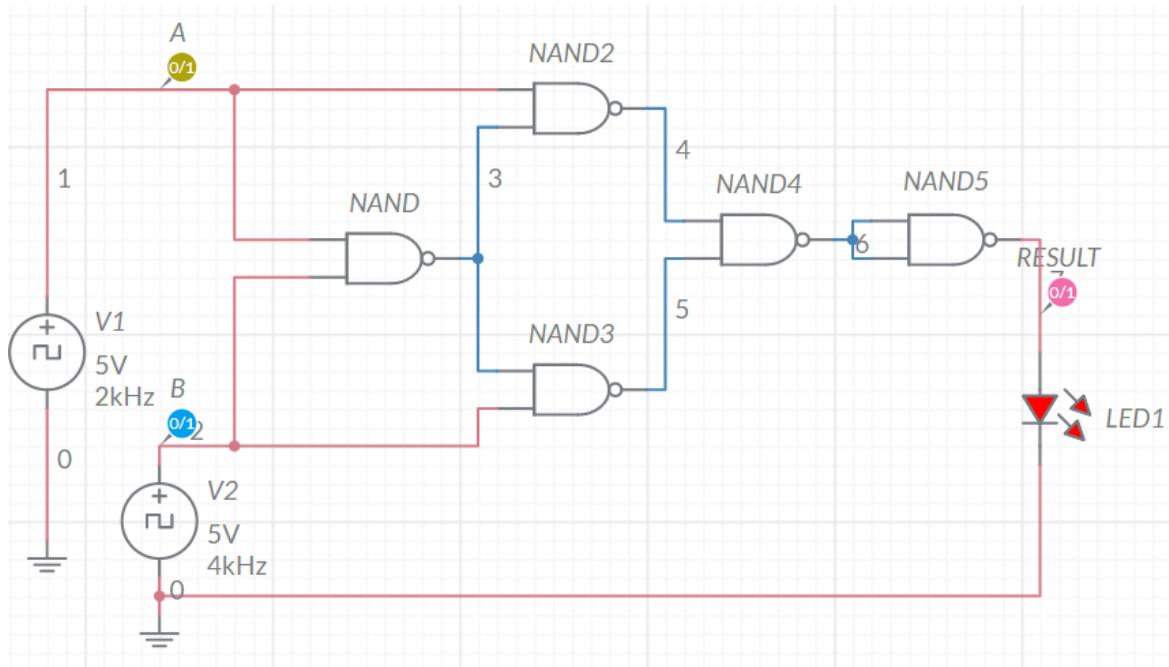
OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):



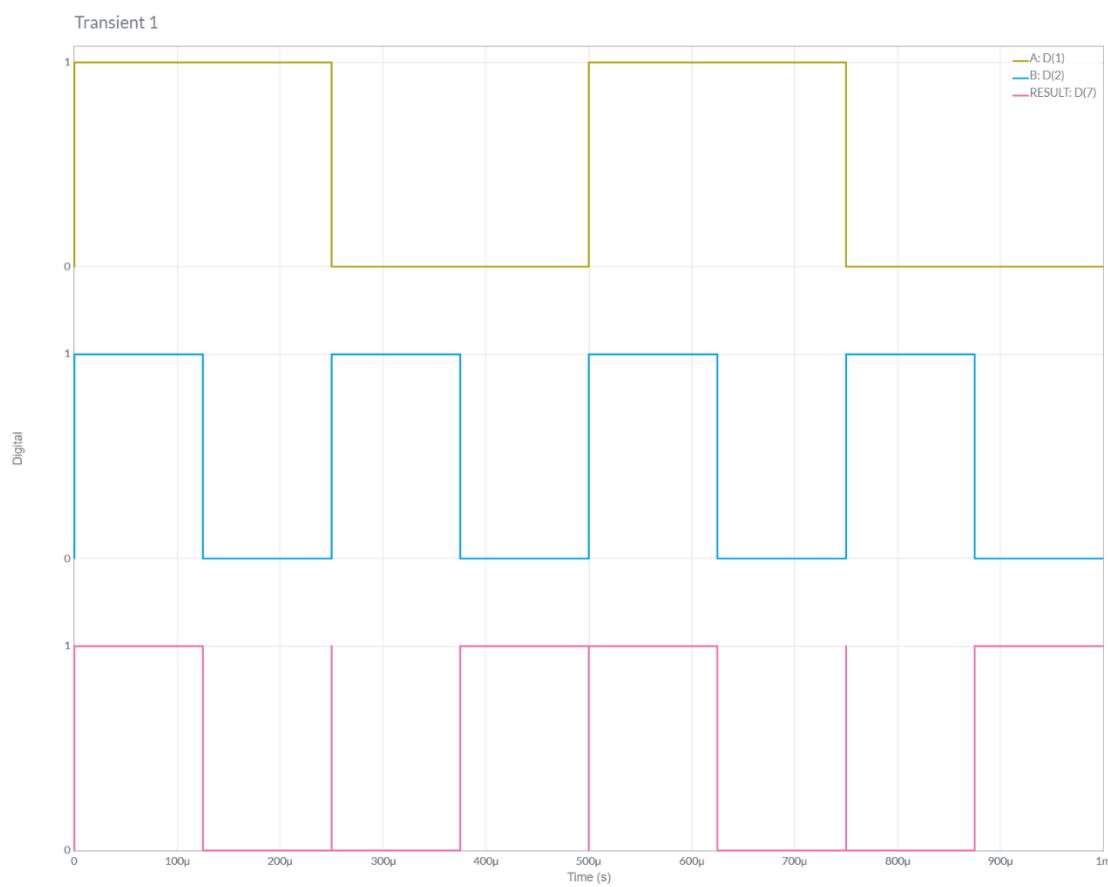


## 17.) 2-INPUT EX-NOR USING NAND

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):





## CONCLUSIONS

1.) We observe that Truth Table of all Logic Gates like AND, OR, NOT, NAND, NOR, XOR and XNOR are **Verified** with Waveforms simulated using **MULTISIM**.

Thus, we have practically verified the Working of Various Logic Gates by using **Multisim for Graphical Analysis** and **Theoretical using Truth Table**.

2.) We can also Conclude that All Logic Gates can be created using either of NAND or NOR Gates Combination. Therefore, *NAND* and *NOR* are also Called **Universal Gates**.



# ASSIGNMENT SECTION

## Question -1

a.) Calculate the Logic Gates Circuit's Output [Theoretical]

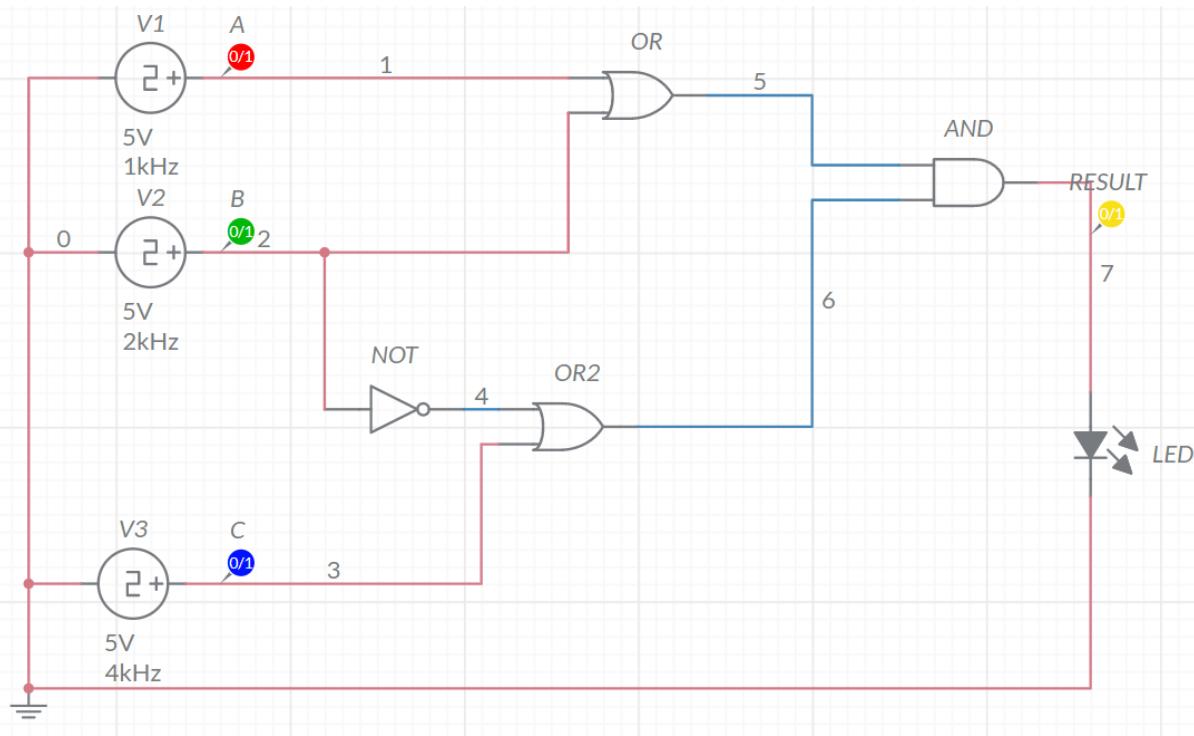
(A) Circuit 1

Final Expression =  $(A+B) \cdot (\bar{B}+C)$

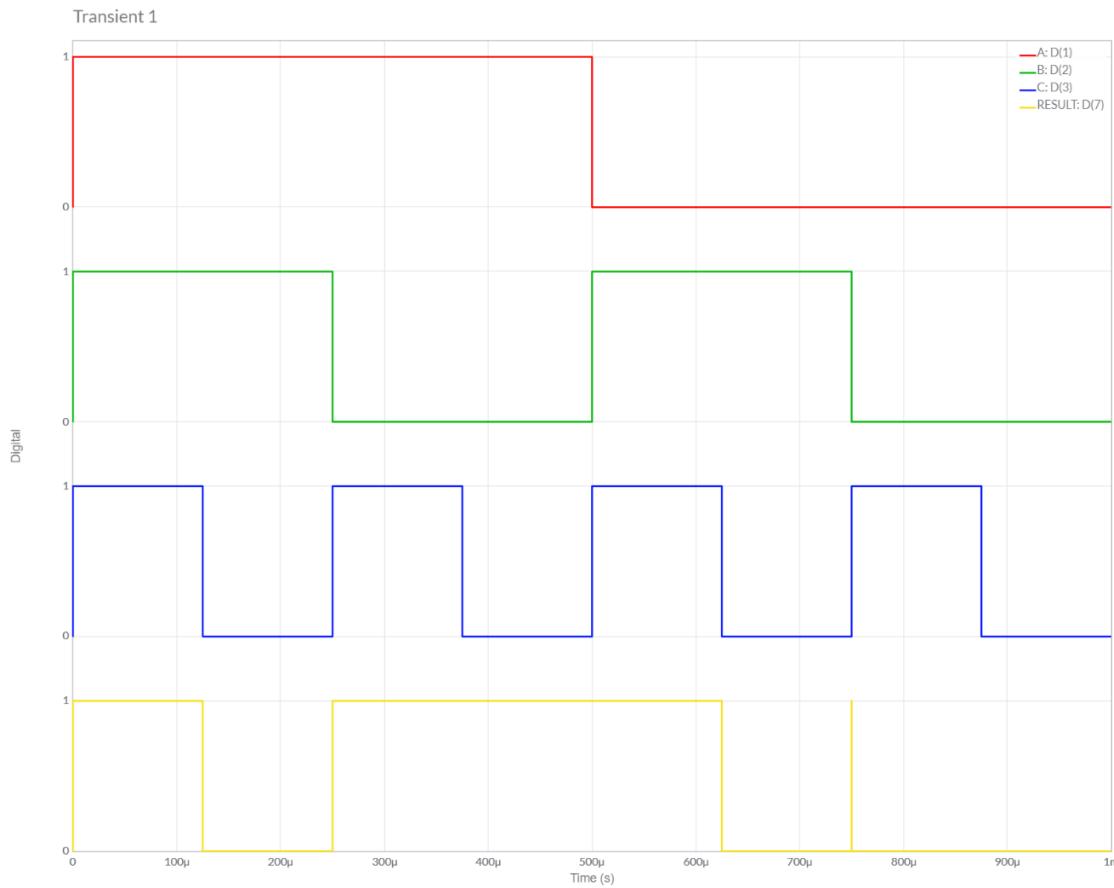
SrNo	A	B	C	$\bar{B}$	$A+B$	$\bar{B}+C$	Result : $(A+B) \cdot (\bar{B}+C)$
1.)	0	0	0	1	0	1	0
2.)	0	0	1	1	0	1	0
3.)	0	1	0	0	1	0	0
4.)	0	1	1	0	1	1	1
5.)	1	0	0	1	1	1	1
6.)	1	0	1	1	1	1	1
7.)	1	1	0	0	1	0	0
8.)	1	1	1	0	1	1	1



b.) Implement the circuit as shown in Figure in Multisim online.



c.) Time Graph





## d.) Final Result and Conclusion

	A	B	C	Graph O/I	Theoretical O/I
1	0	0	0	0	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	1	1	1	1
5	1	0	0	1	1
6	1	0	1	1	1
7	1	1	0	0	0
8	1	1	1	1	1

**Conclusion:**

We can observe from Above Table, Both the *Theoretical* and *Multisim Values of Given Circuit are Equal.*

Hence, Experiment is Performed Successfully (without any Error).

Question -2

a.) Calculate the Logic Gates Circuit's Output [Theoretical]

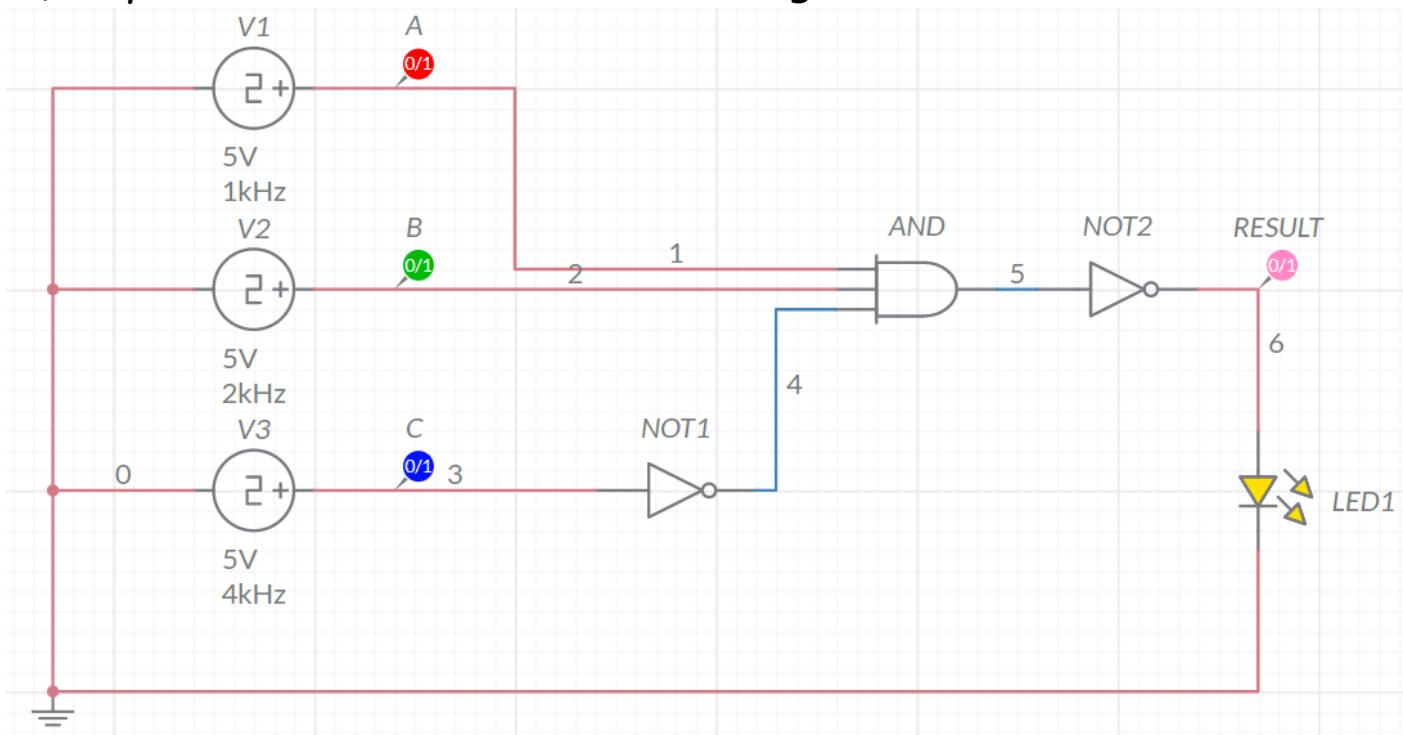
Circuit 2 :

Final Expression =  $\overline{A \cdot B \cdot \bar{C}}$

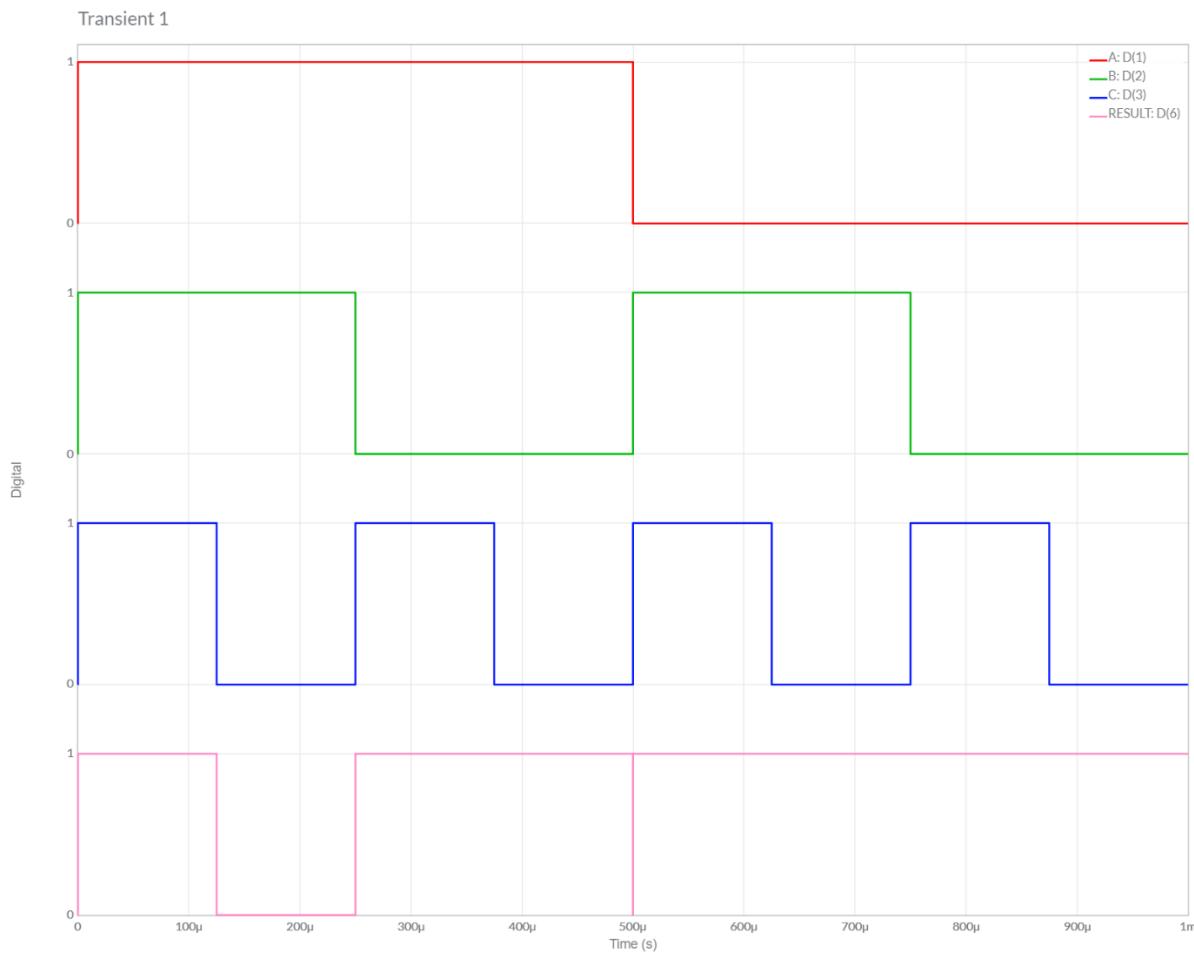
Sr.No	A	B	C	$\bar{C}$	$A \cdot B \cdot \bar{C}$	Result : $\overline{A \cdot B \cdot \bar{C}}$
1.)	0	0	0	1	0	1
2.)	0	0	1	0	0	1
3.)	0	1	0	1	0	1
4.)	0	1	1	0	0	1
5.)	1	0	0	1	0	1
6.)	1	0	1	0	0	1
7.)	1	1	0	1	1	0
8.)	1	1	1	0	0	1



b.) Implement the circuit as shown in Figure in Multisim online.



c.) Time Graph



*d.) Final Result and Conclusion*

	A	B	C	Graph O/I	Theoretical O/I
1	0	0	0	1	1
2	0	0	1	1	1
3	0	1	0	1	1
4	0	1	1	1	1
5	1	0	0	1	1
6	1	0	1	1	1
7	1	1	0	0	0
8	1	1	1	1	1

*Conclusion:*

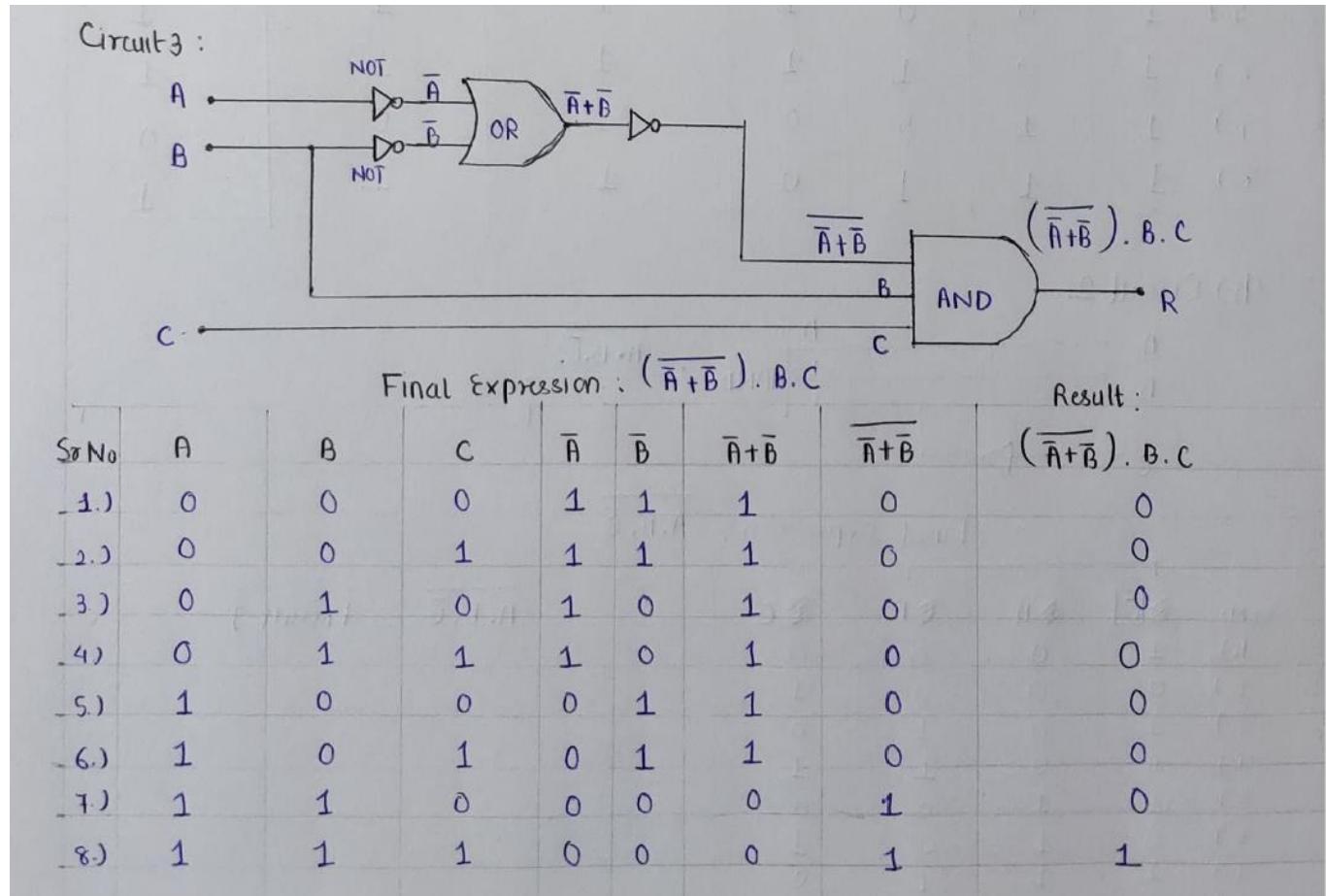
We can observe from Above Graph, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

Hence, Experiment is Performed Successfully (without any Error).



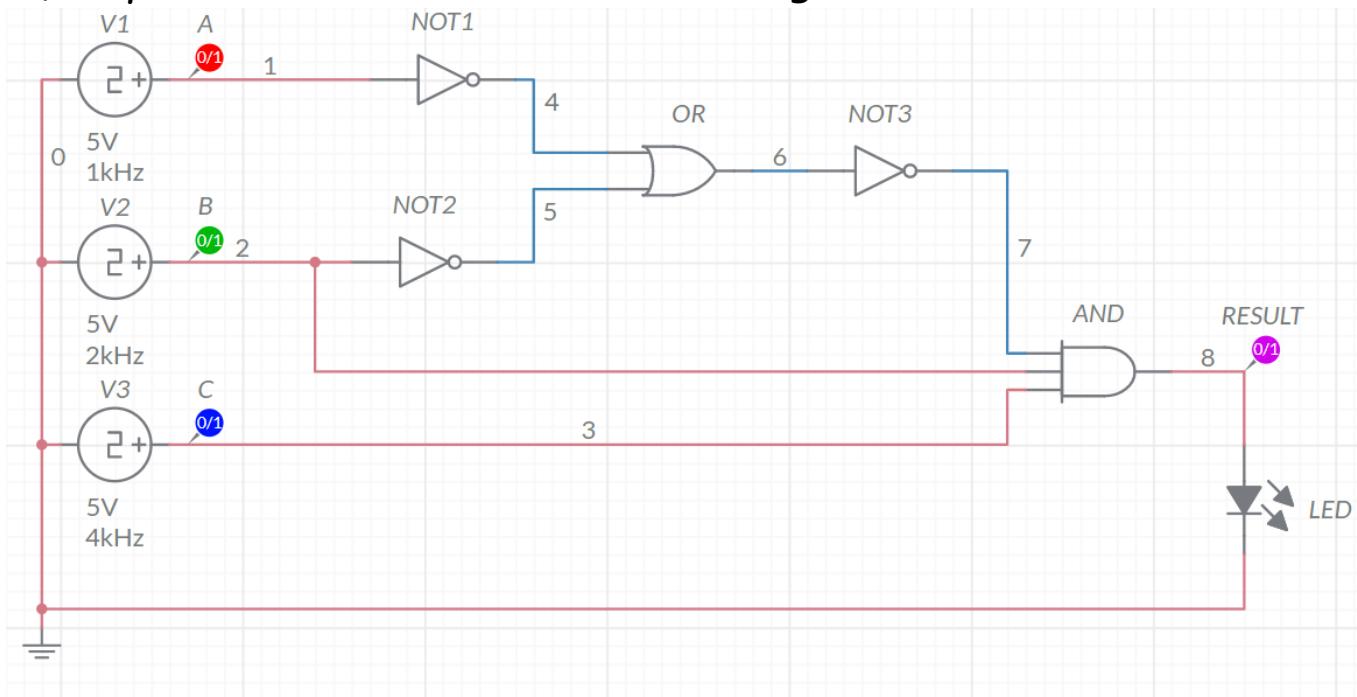
### Question -3

a.) Calculate the Logic Gates Circuit's Output [Theoretical]

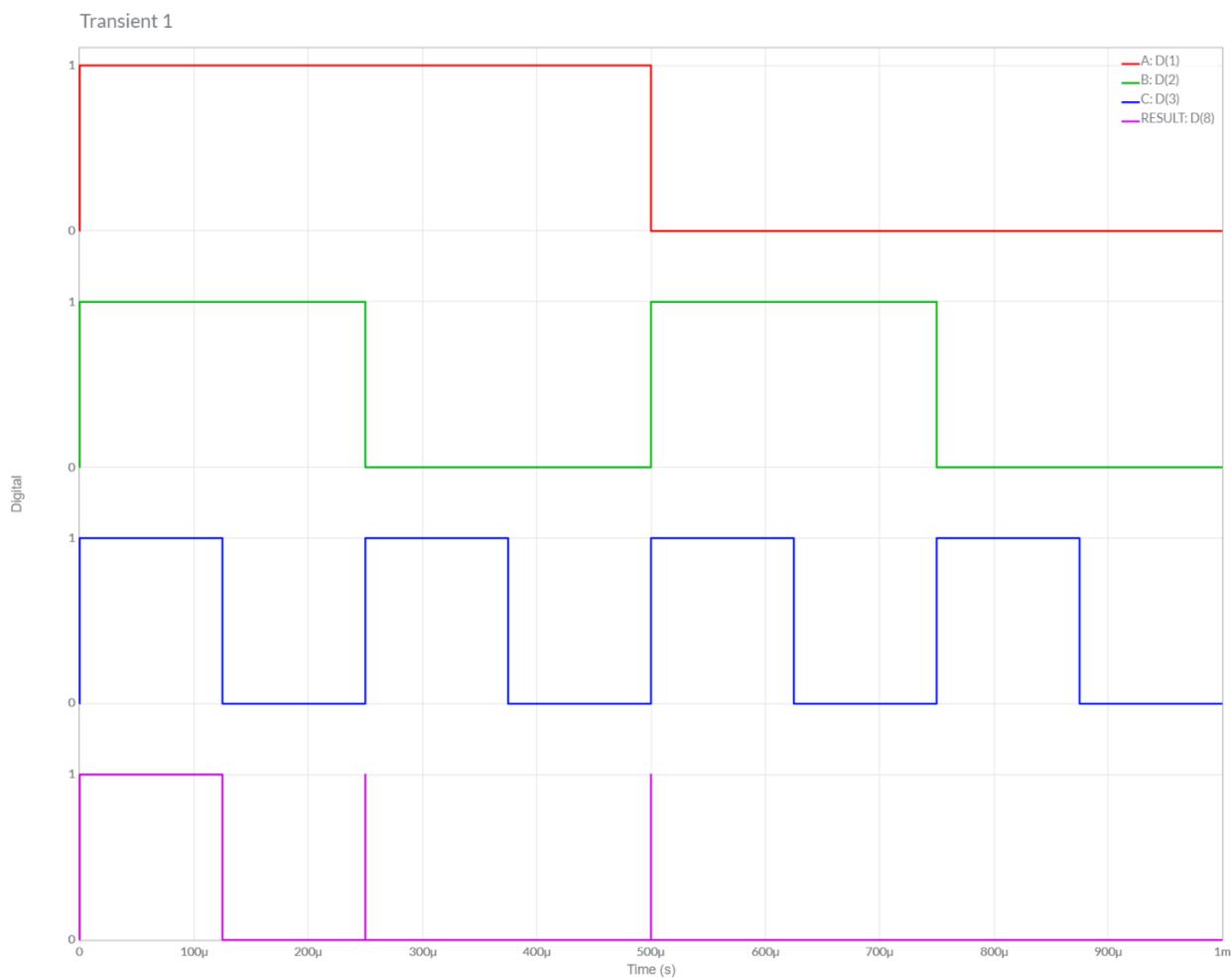




b.) Implement the circuit as shown in Figure in Multisim online.



c.) Time Graph





### d.) Final Result and Conclusion

	A	B	C	Graph O/I	Theoretical O/I
1	0	0	0	0	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	1	1	0	0
5	1	0	0	0	0
6	1	0	1	0	0
7	1	1	0	0	0
8	1	1	1	1	1

#### Conclusion:

We can observe from Above Table, Both the *Theoretical* and *Multisim Values of Given Circuit* are **Equal**.

Hence, Experiment is Performed Successfully (without any Error).



Expt. No:

3

Date:

27/08/2020

**Half Adder and Half Subtractor**

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

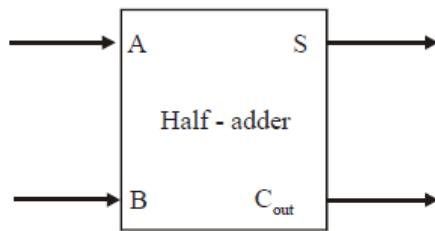
**SOFTWARE TOOLS / OTHER REQUIREMENTS:**

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.

$$S = AB + \bar{A}B$$

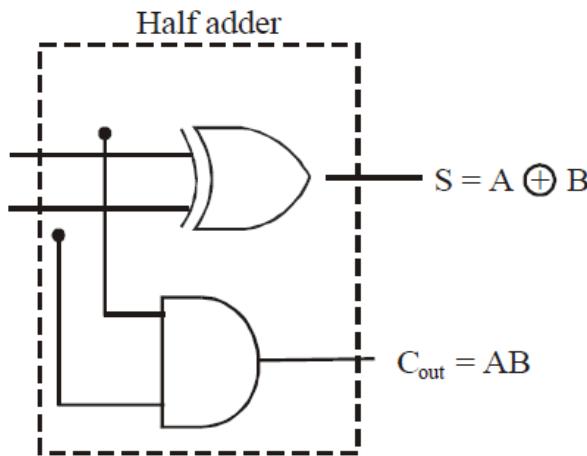
$$= A \oplus B$$

and the Boolean expression for the carry output by,

$$C_{out} = AB$$

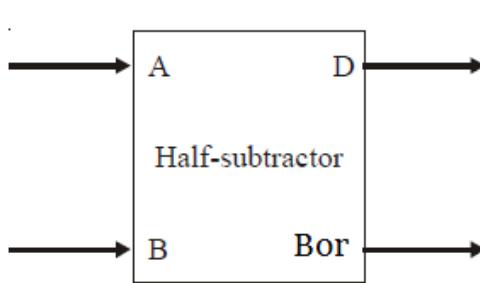


The above two equations can be implemented by a logic circuit shown in Fig below. As seen from this figure, the sum output (S) is obtained from an exclusive-OR gate while the carry output (Cout) is the output of a two-input AND gate.



### 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

Inputs		Outputs	
A	B	D	Bor
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:

$$0 - 0 = 0$$

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

$$1 - 0 = 1$$

$$1 - 1 = 0$$

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

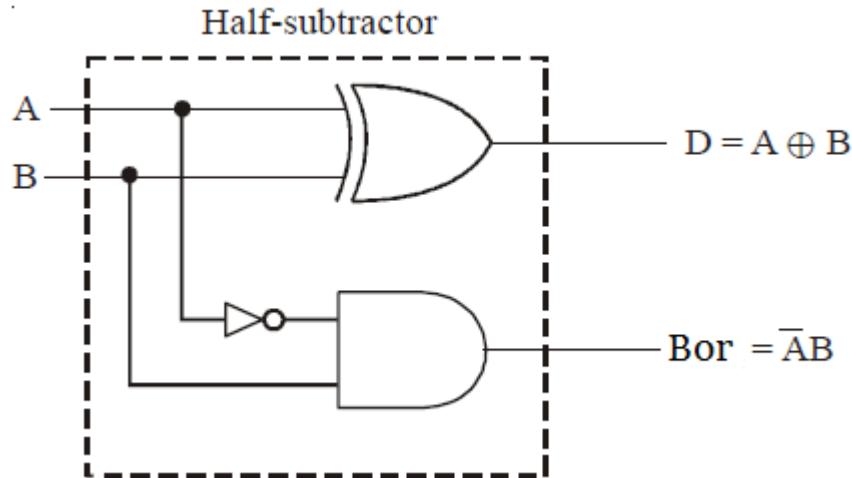
$$\begin{aligned} D &= A\bar{B} + \bar{A}B \\ &= A \oplus B \end{aligned}$$

and the Boolean expression for the borrow bit,

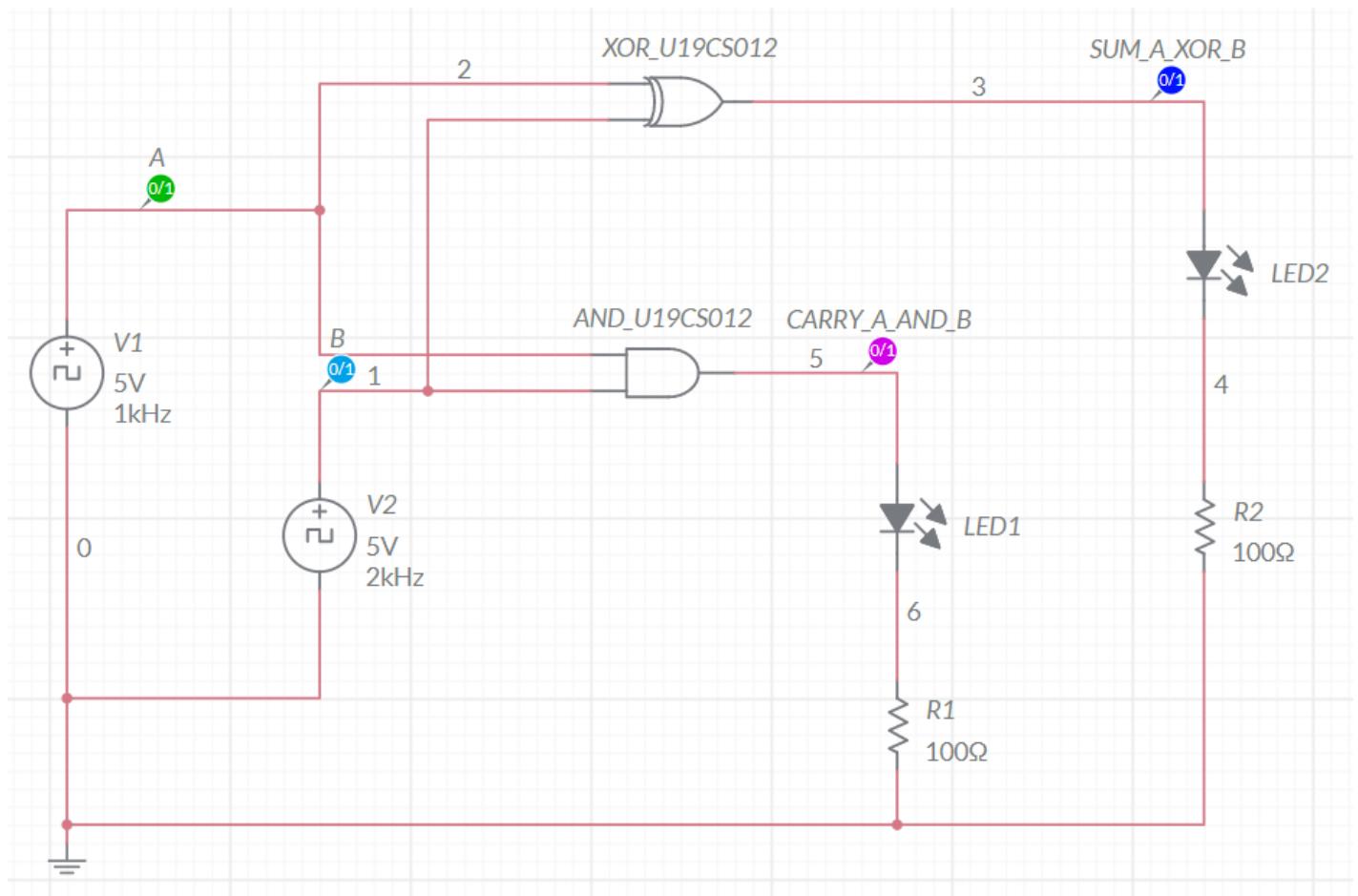
$$\text{Bor} = \bar{A}B$$



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.

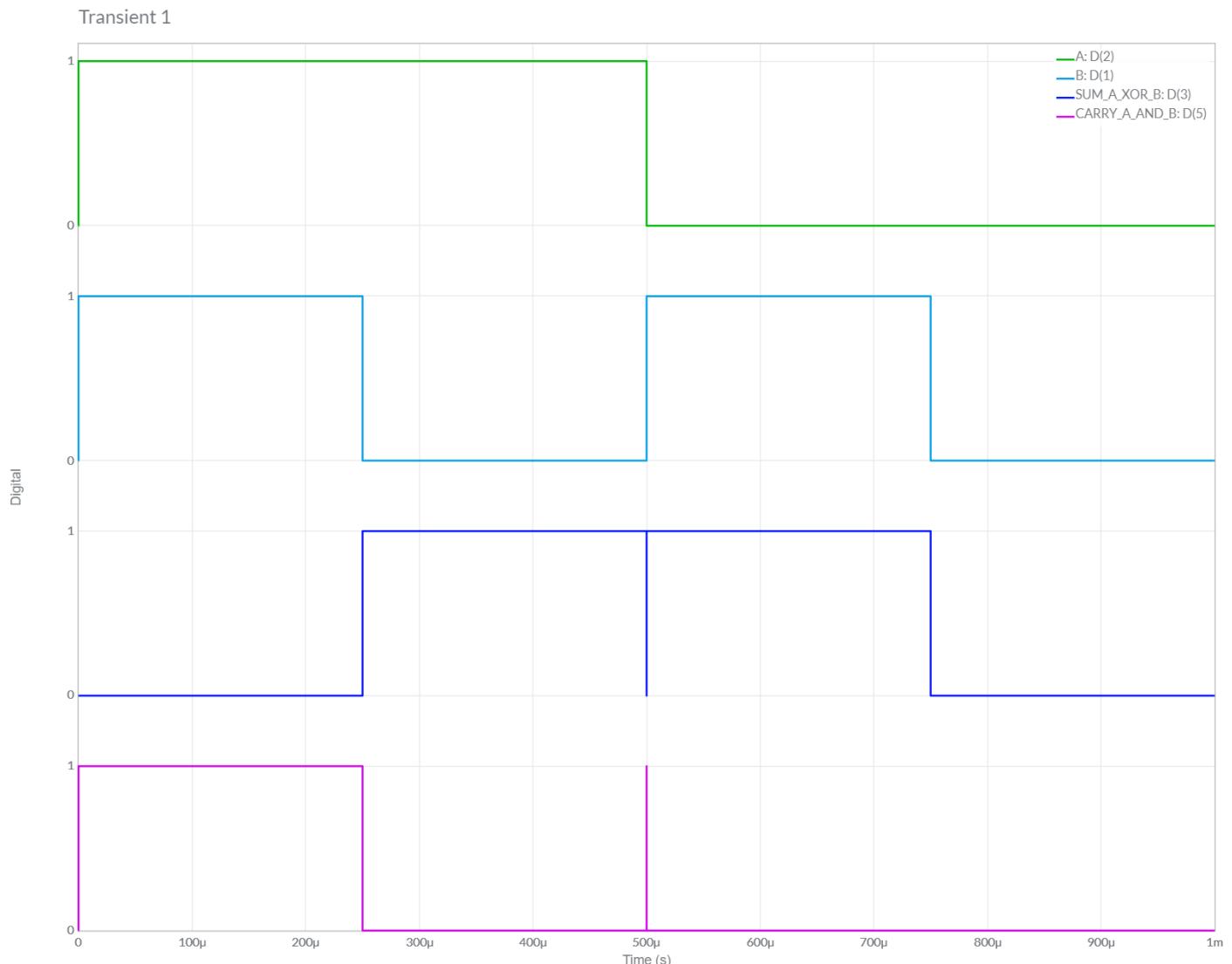


### HALF ADDER: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



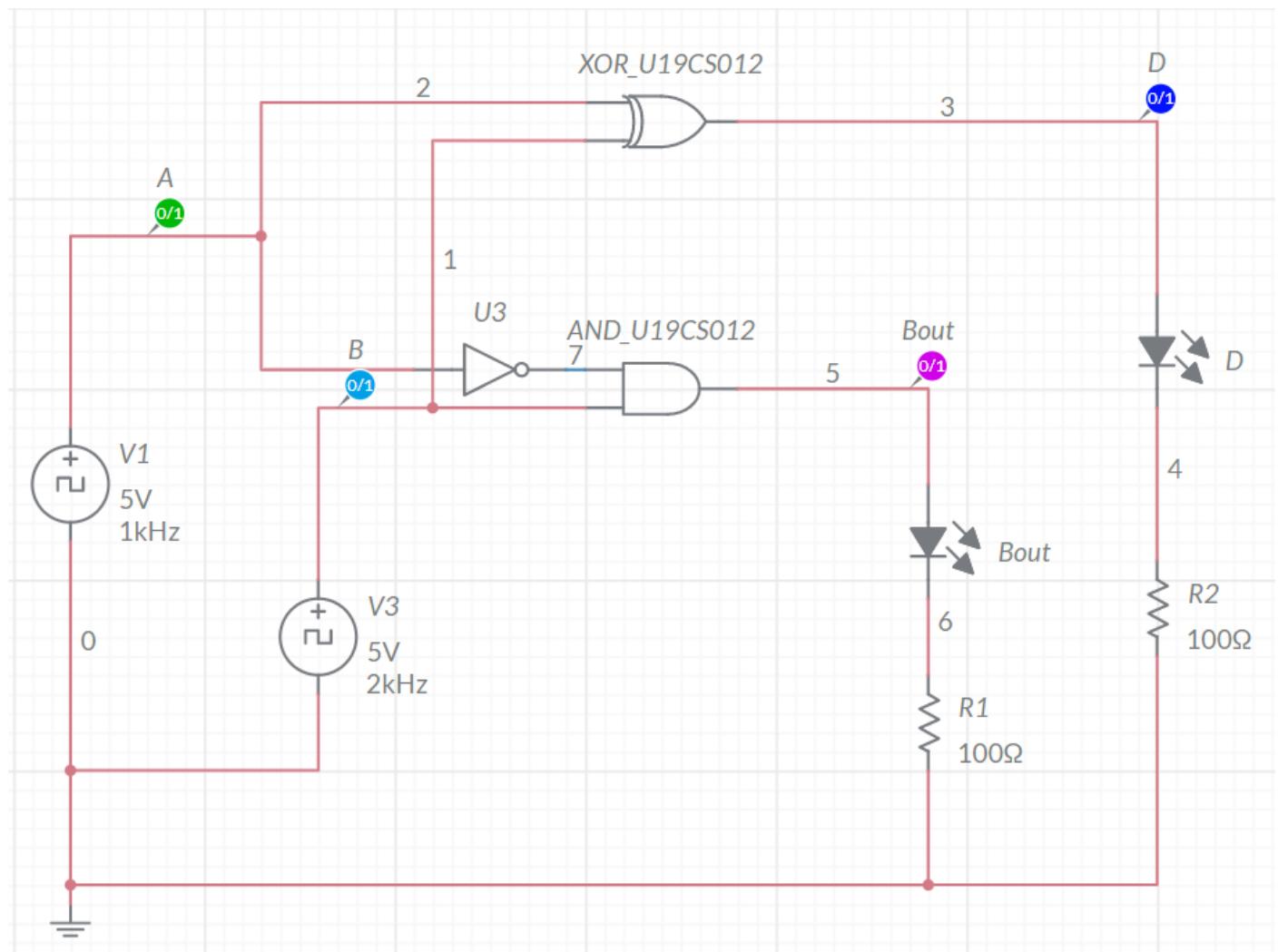


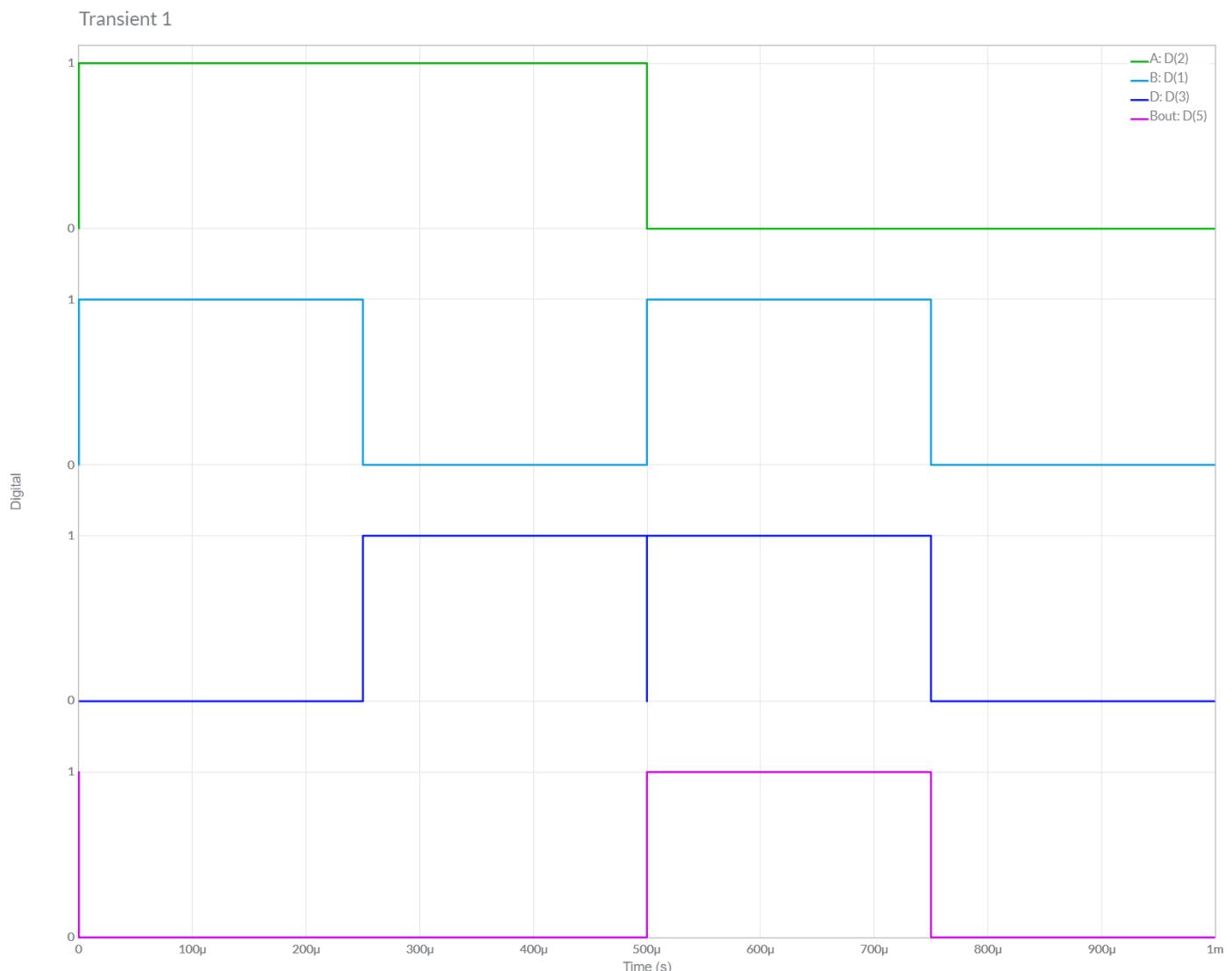
**HALF ADDER: OUTPUT PLOTS/WAVEFORMS (FROM MULTISIM):**





**HALF SUBTRACTOR: CIRCUIT/CONNECTION DIAGRAMS (MULTISIM)**



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

- 1.) The Results obtained from the Truth Table from Half-Adder Circuit and Grapher Image [Waveform Simulation] are *Equal*, Hence the Circuit is verified to be Half-Adder.
- 2.) The Results obtained from the Truth Table from Half-Subtractor Circuit and Grapher Image [Waveform Simulation] are Equal, Hence the Circuit is verified to be Half-Subtractor.
- 3.) Hence, Half Adder and Half Subtractor Circuit have been Implemented Successfully in Multisim.



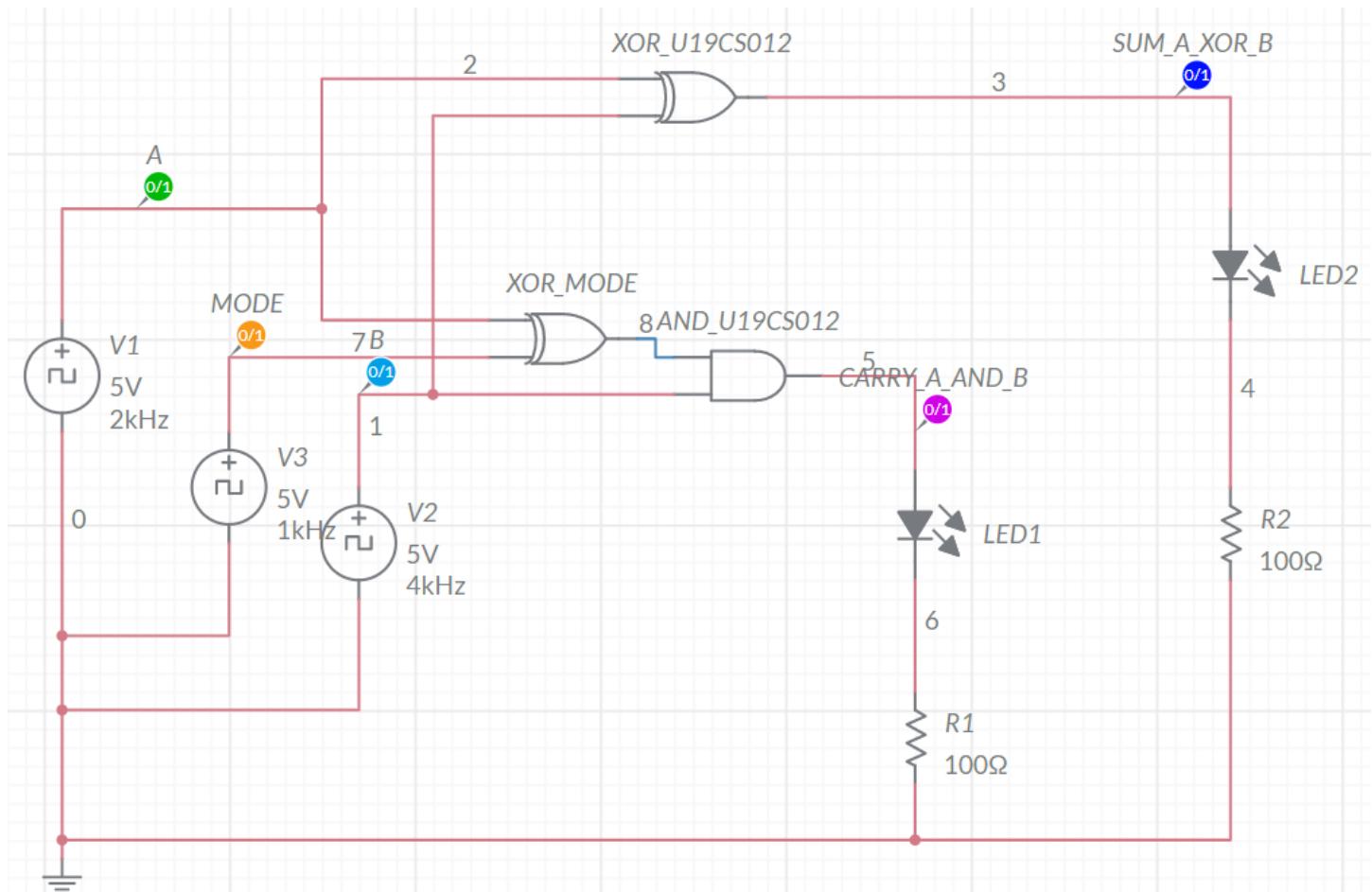
# **ASSIGNMENT - 3**

U19CS012

**Design and verify their functionality of below circuits with the help of Multisim.**

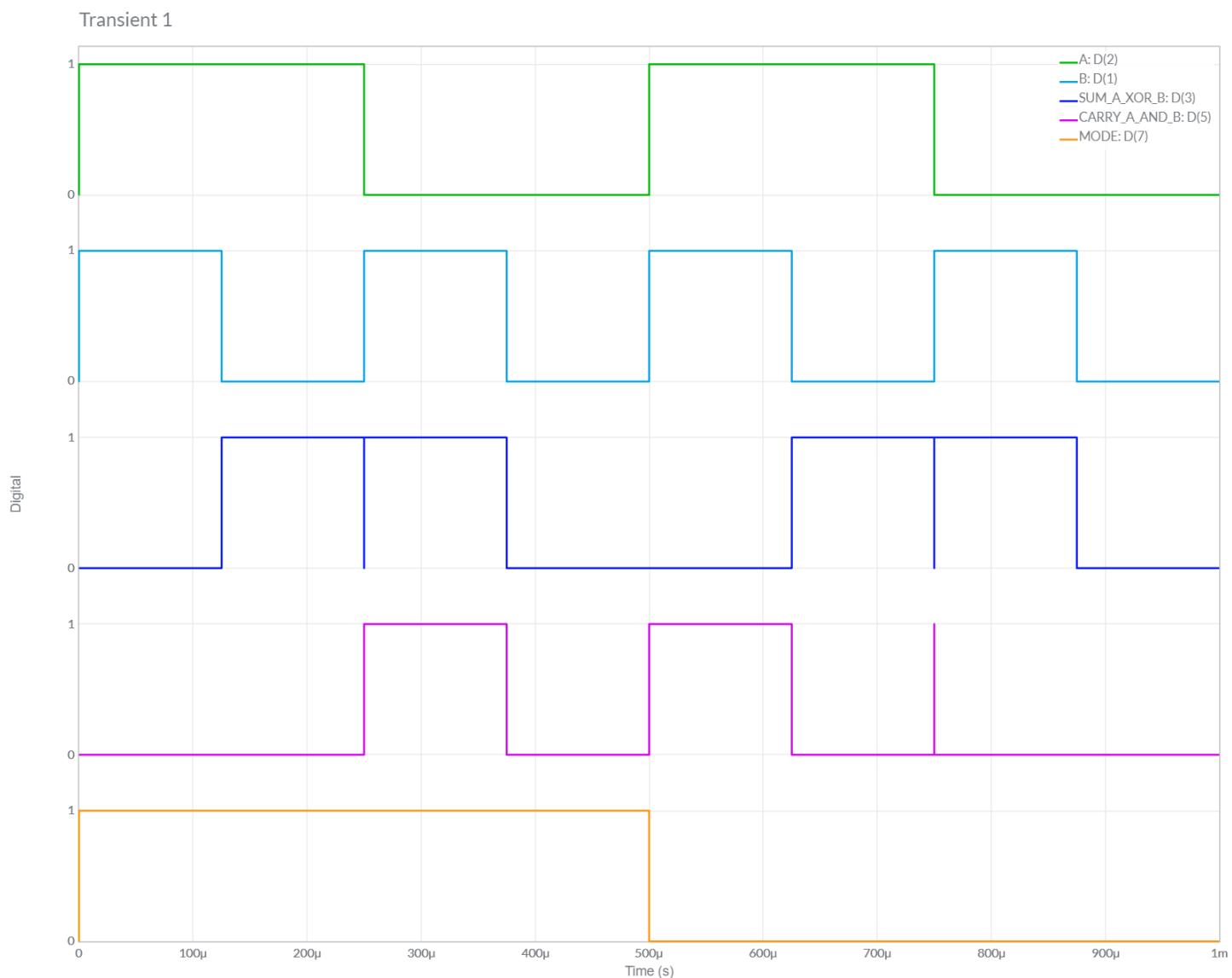
**1. Design and implement Half Adder and Half Subtractor (Single Circuit) using Mode Control 'M'.**

*a.) Implement the circuit in Multisim online*





### b.) Timing Graph



### c.) Truth Table:

When Mode = 1, Circuit Behaves as Half Subtractor Circuit

Borrow Out, Bout=  $A' \cdot B$

Difference, D =  $A \oplus B$



A	B	$B_{out}$	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

When Mode = 0, Circuit Behaves as Half Adder Circuit

$$\text{Sum, } S = A \oplus B$$

$$\text{Carry, } C = A \cdot B$$

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The Same Truth Table is Followed for Next 2 Questions as well.

### Conclusion:

We can observe from Above Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

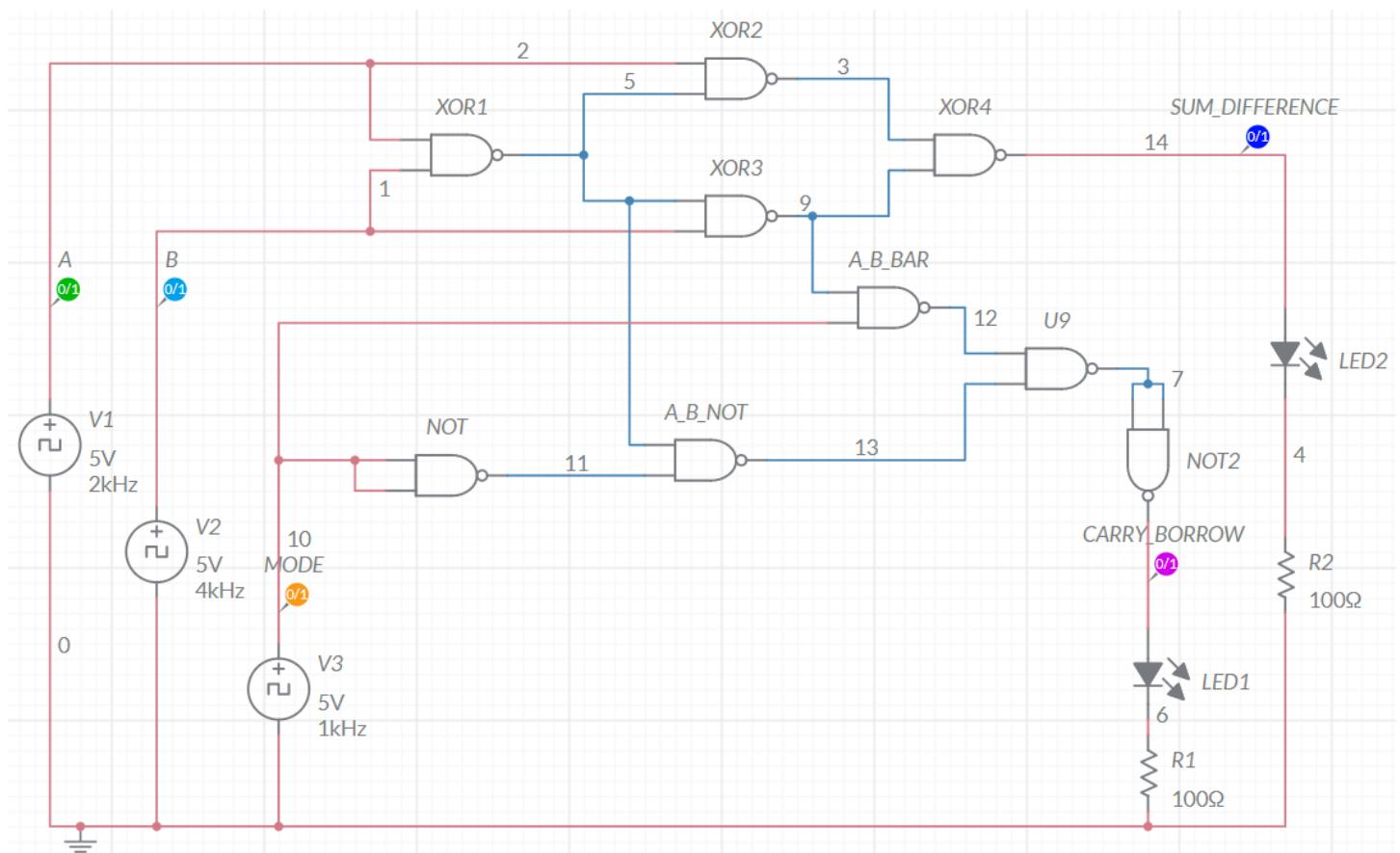
Hence, Experiment is Performed Successfully (without any Error).



**2. Design and implement the circuit in question '1' by using least number of NAND gates only.**

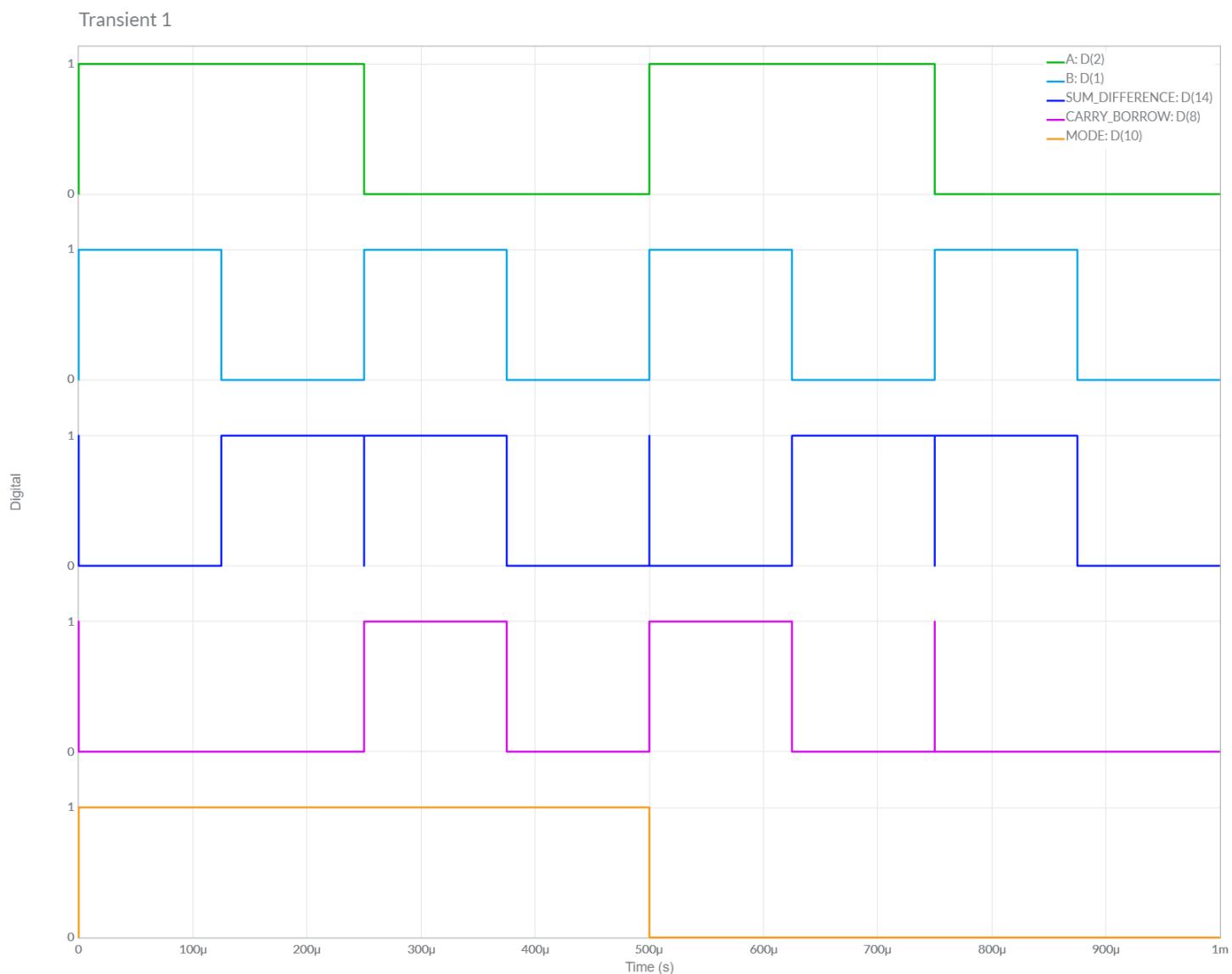
**Minimum NAND Gates Required = 9**

*a.) Implement the circuit in Multisim online*





## b.) Timing Graph

***Conclusion:***

We can observe from Above Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

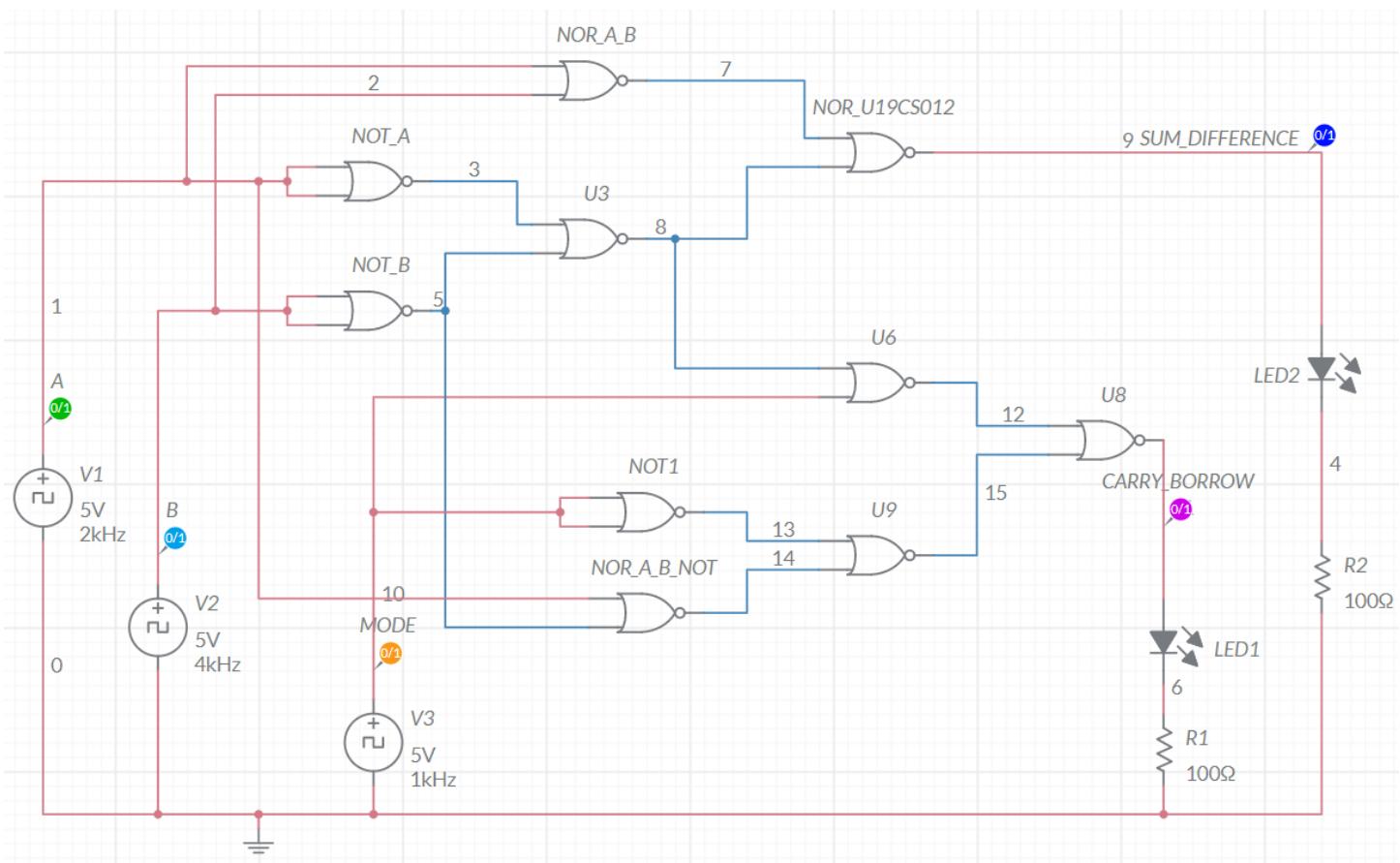
Hence, Experiment is Performed Successfully (without any Error).



**3. Design and implement the circuit in question '1' by using least number of NOR gates only.**

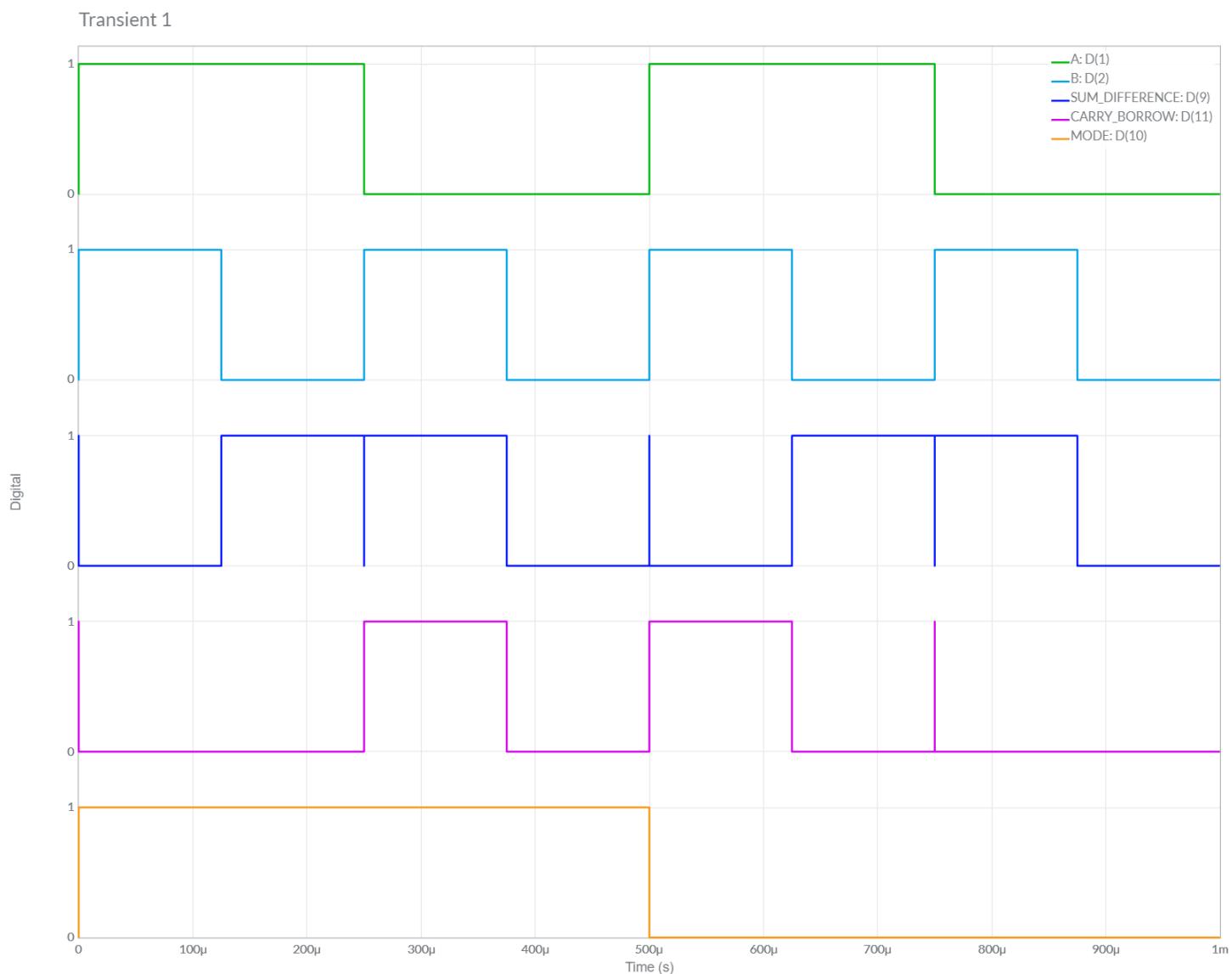
**Minimum NOR Gates Required = 10**

**a.) Implement the circuit in Multisim online**





## b.) Timing Graph

**Conclusion:**

We can observe from Above Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

Hence, Experiment is Performed Successfully (without any Error).



Expt. No:

**4**

Date:

03-09-2020

## **Full Adder and Full Subtractor**

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

### **SOFTWARE TOOLS / OTHER REQUIREMENTS:**

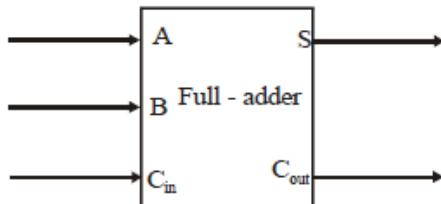
1. Multisim Simulator
2. Logic Gates (AND, NOT and EX-OR)

### **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 (Cout). Fig (b) shows the truth table for the full-adder.



(a) logic symbol

Inputs			Outputs	
A	B	C <sub>in</sub>	S	C <sub>out</sub>
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:

$$\begin{array}{rcl}
 0 + 0 + 0 & = & 0 \text{ with carry } 0 \\
 0 + 0 + 1 & = & 1 \text{ with carry } 0 \\
 0 + 1 + 0 & = & 1 \text{ with carry } 0 \\
 0 + 1 + 1 & = & 0 \text{ with carry } 1 \\
 1 + 0 + 0 & = & 1 \text{ with carry } 0 \\
 1 + 0 + 1 & = & 0 \text{ with carry } 1 \\
 1 + 1 + 0 & = & 0 \text{ with carry } 1 \\
 1 + 1 + 1 & = & 1 \text{ with carry } 1
 \end{array}$$

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

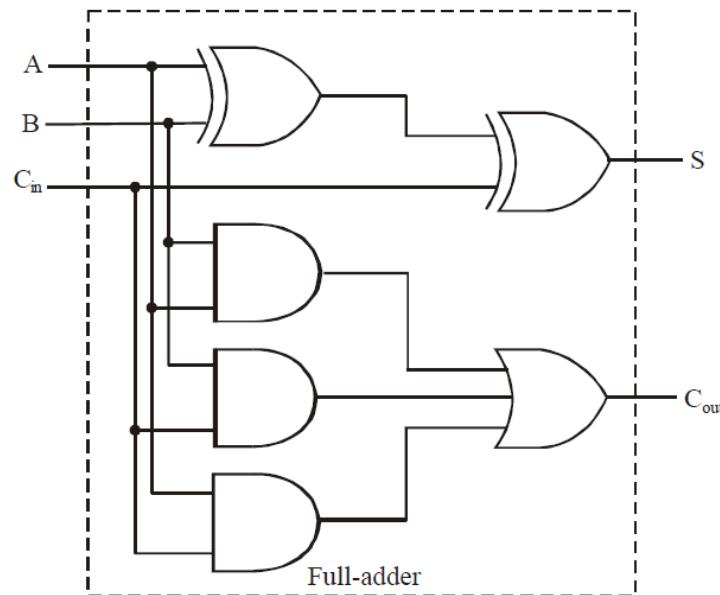


$$\begin{aligned} S &= A \oplus (B \oplus C_{in}) \\ &= A \oplus B \oplus C_{in} \end{aligned}$$

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

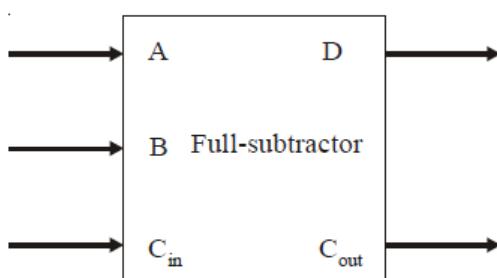
$$C_{out} = BC_{in} + AC_{in} + AB$$

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:

$$0 - 0 - 0 = 0 \text{ with borrow } 0$$

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

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

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

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

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

$$1 - 1 - 1 = 1 \text{ with borrow } 1$$

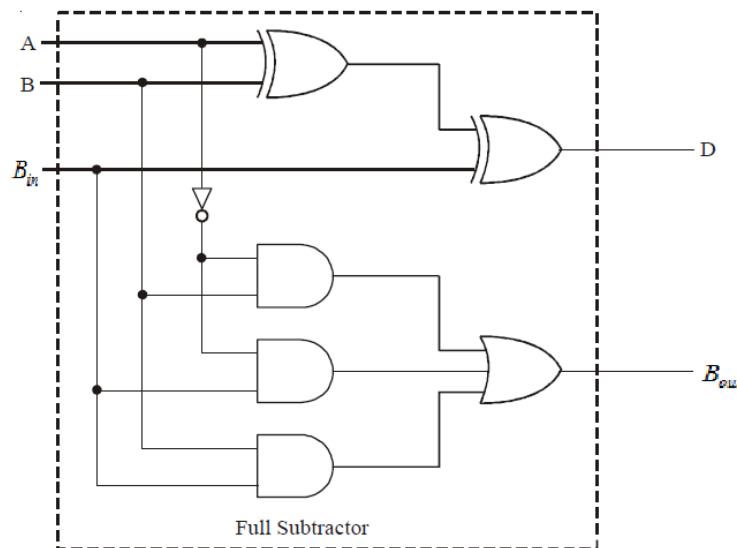
The Boolean expression for the difference bit (D) can be obtained by summing and simplifying all the input combinations from the truth table which have 1 in the corresponding difference column. The final simplified expression for difference is given by

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

and, the Boolean expression for the borrow bit,

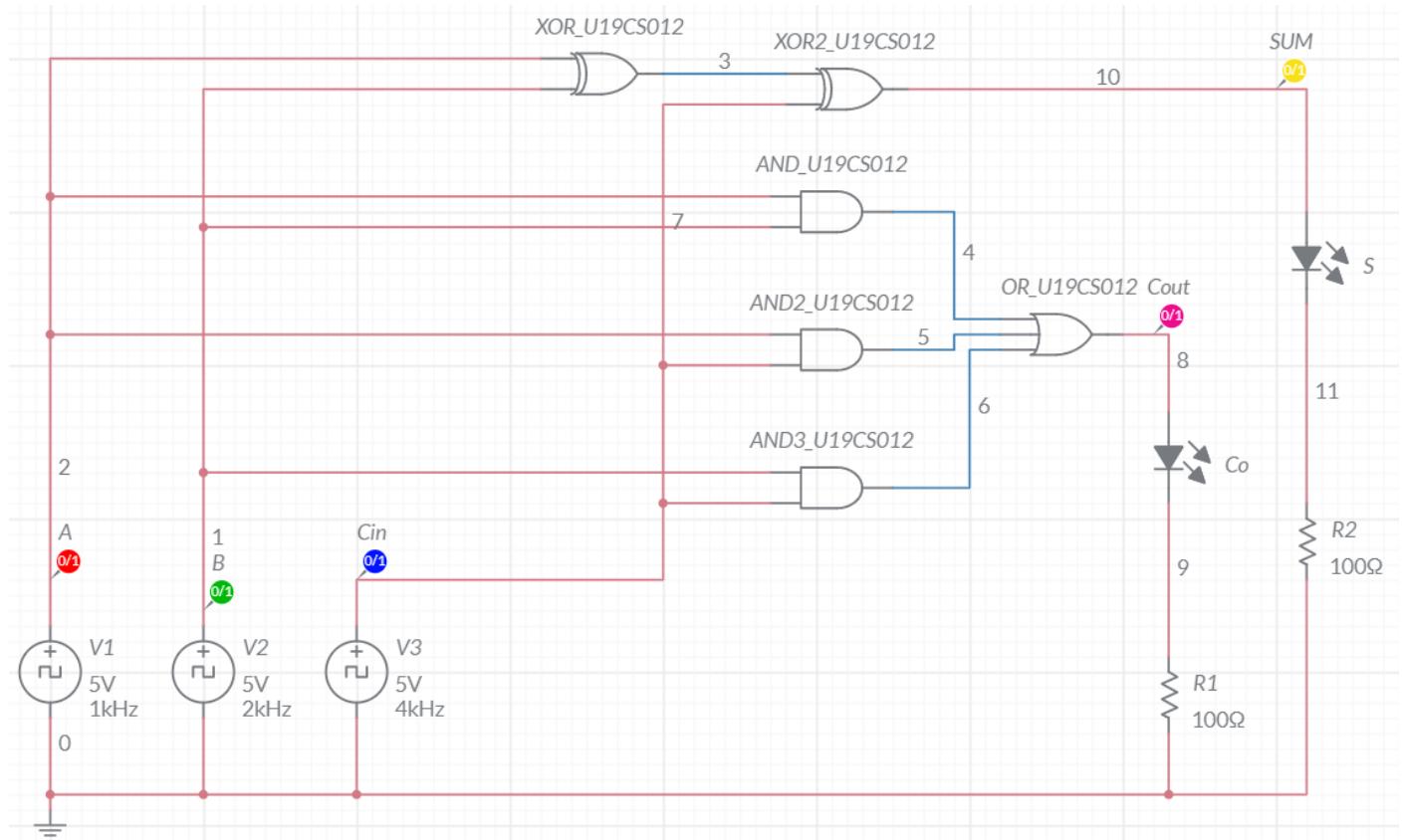
$$B_{out} = \overline{A}B + BB_{in} + \overline{A}\overline{B}_{in}$$

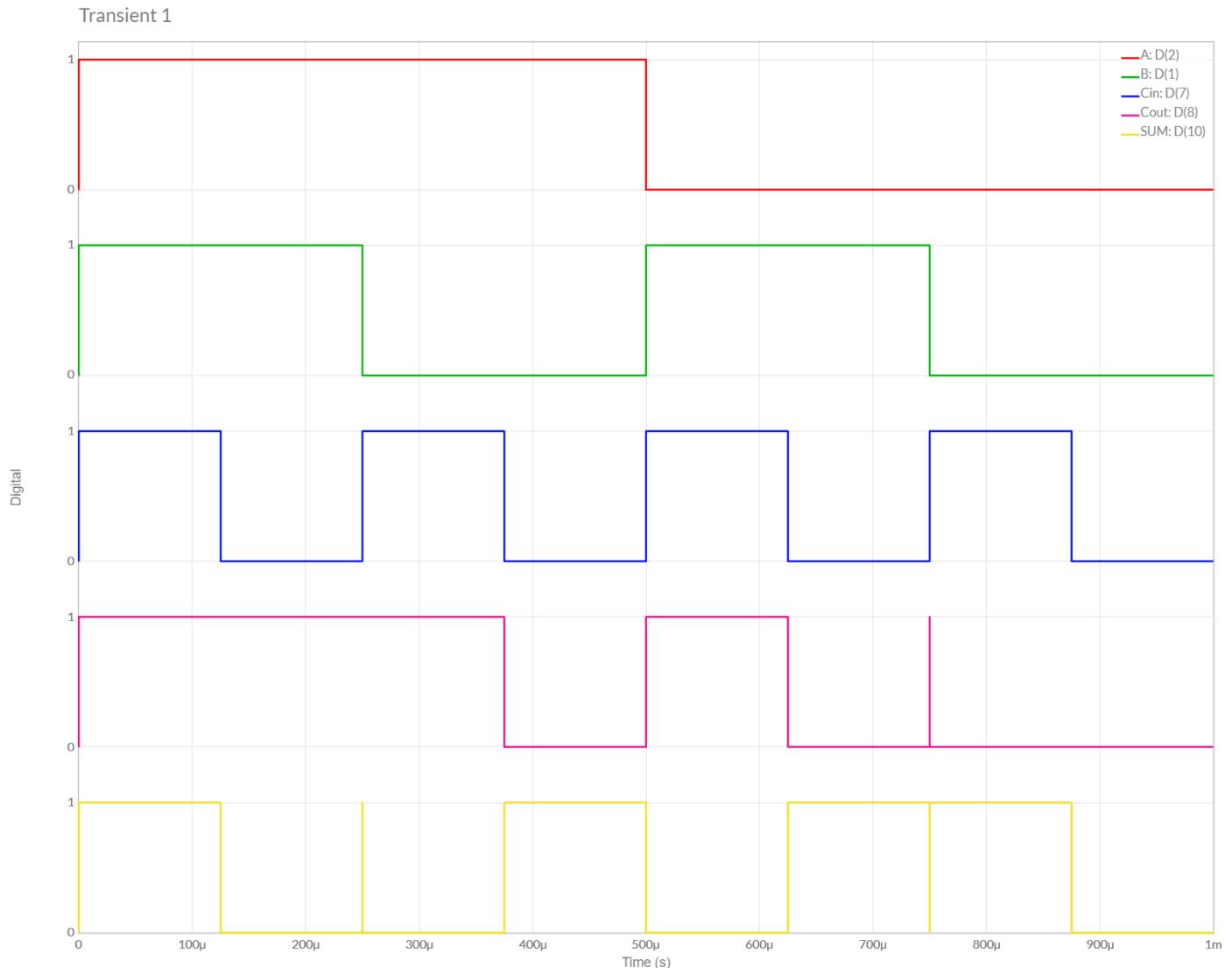
From the above two expressions we find that to implement full-subtractor's difference output function, two-2 input Exclusive-OR gates can be used. Similarly from equation of borrow we find that to implement the full-subtractor's borrow output function, a NOT gate, three 2-input AND gates followed by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





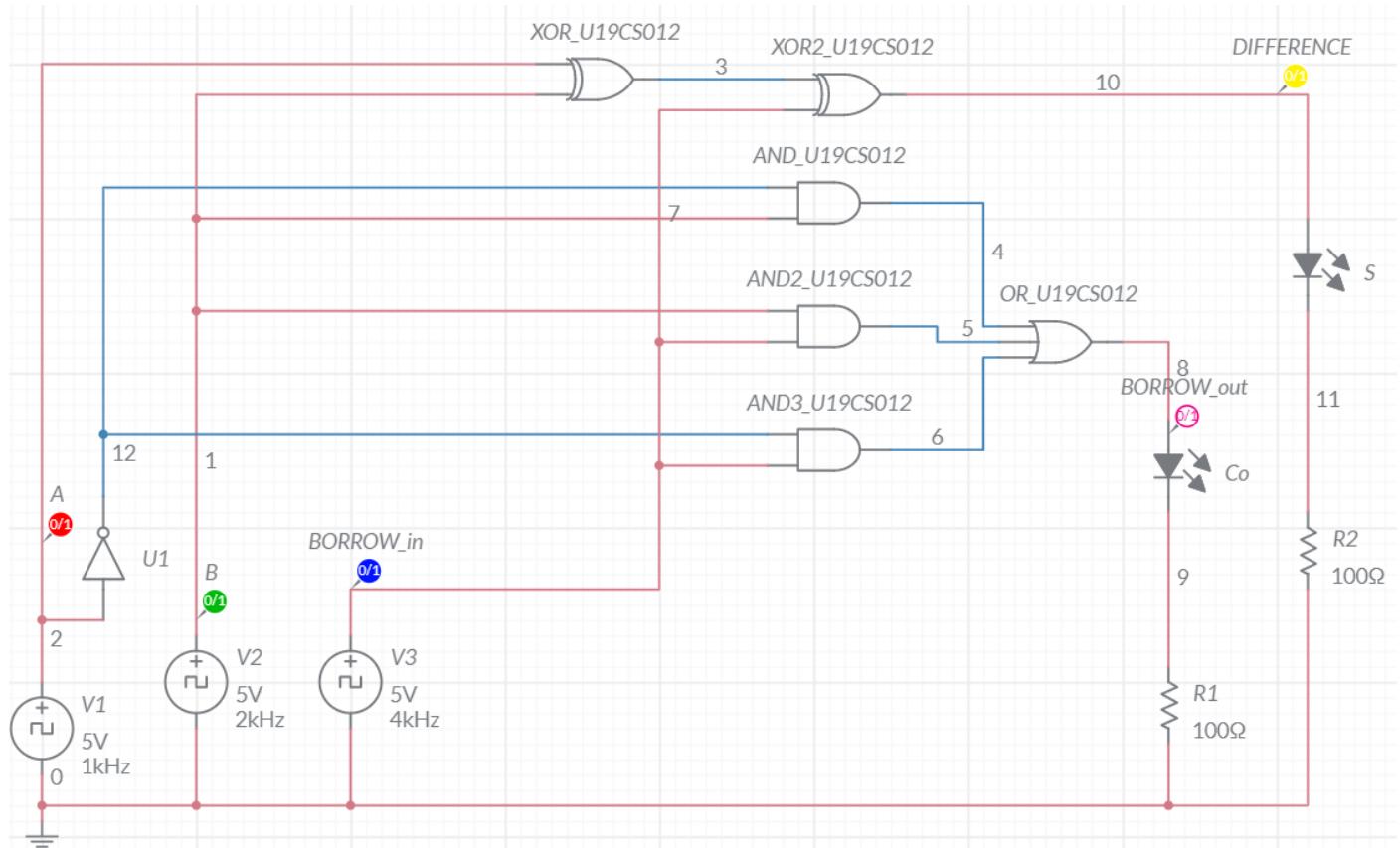
**FULL ADDER: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**

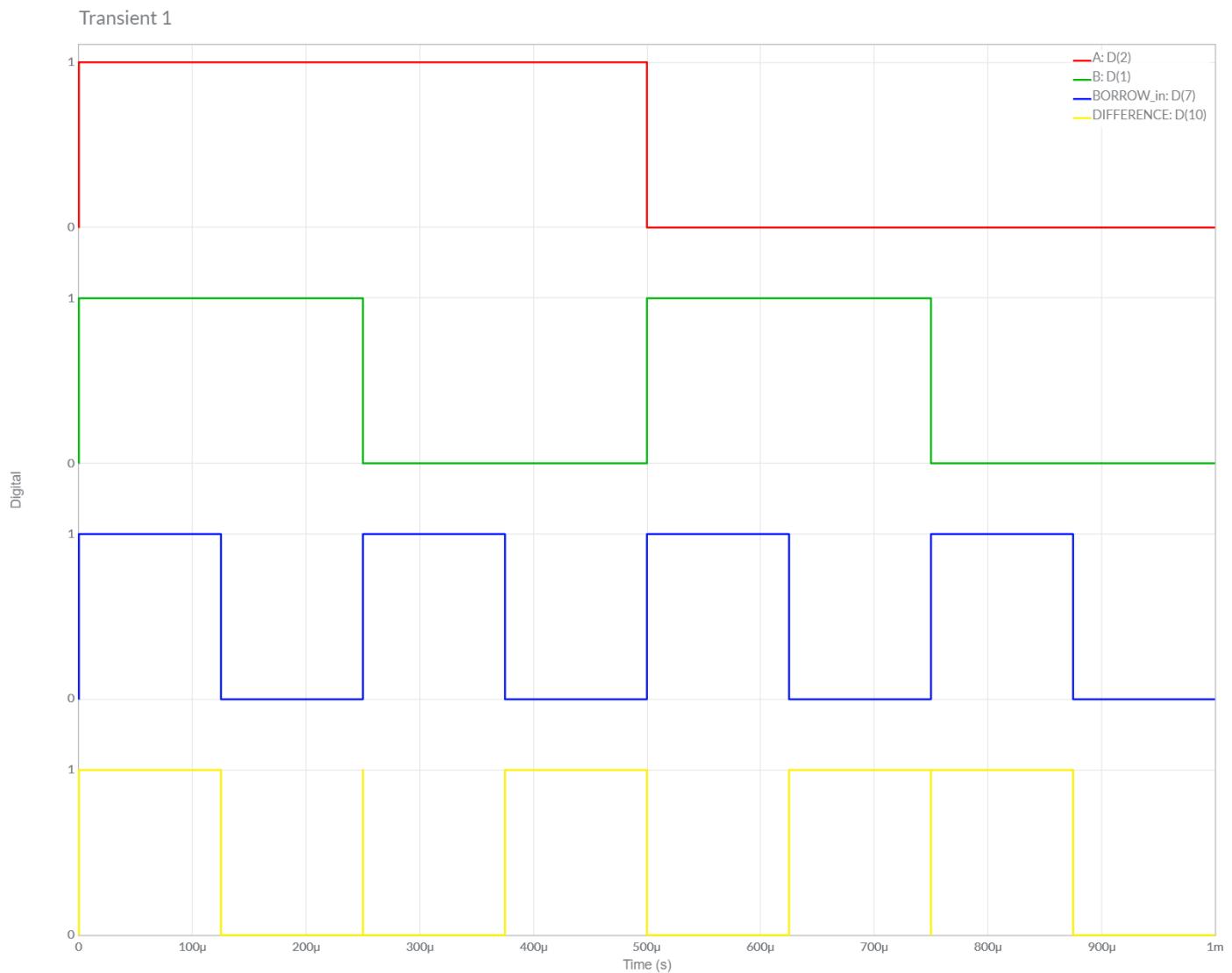


**FULL ADDER: OUTPUT PLOTS/WAVEFORMS (FROM MULTISIM):**



**FULL SUBTRACTOR: CIRCUIT/CONNECTION DIAGRAMS (MULTISIM)**



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

- 1.) Full Adder and Full Subtractor Circuit have been designed and implemented successfully.



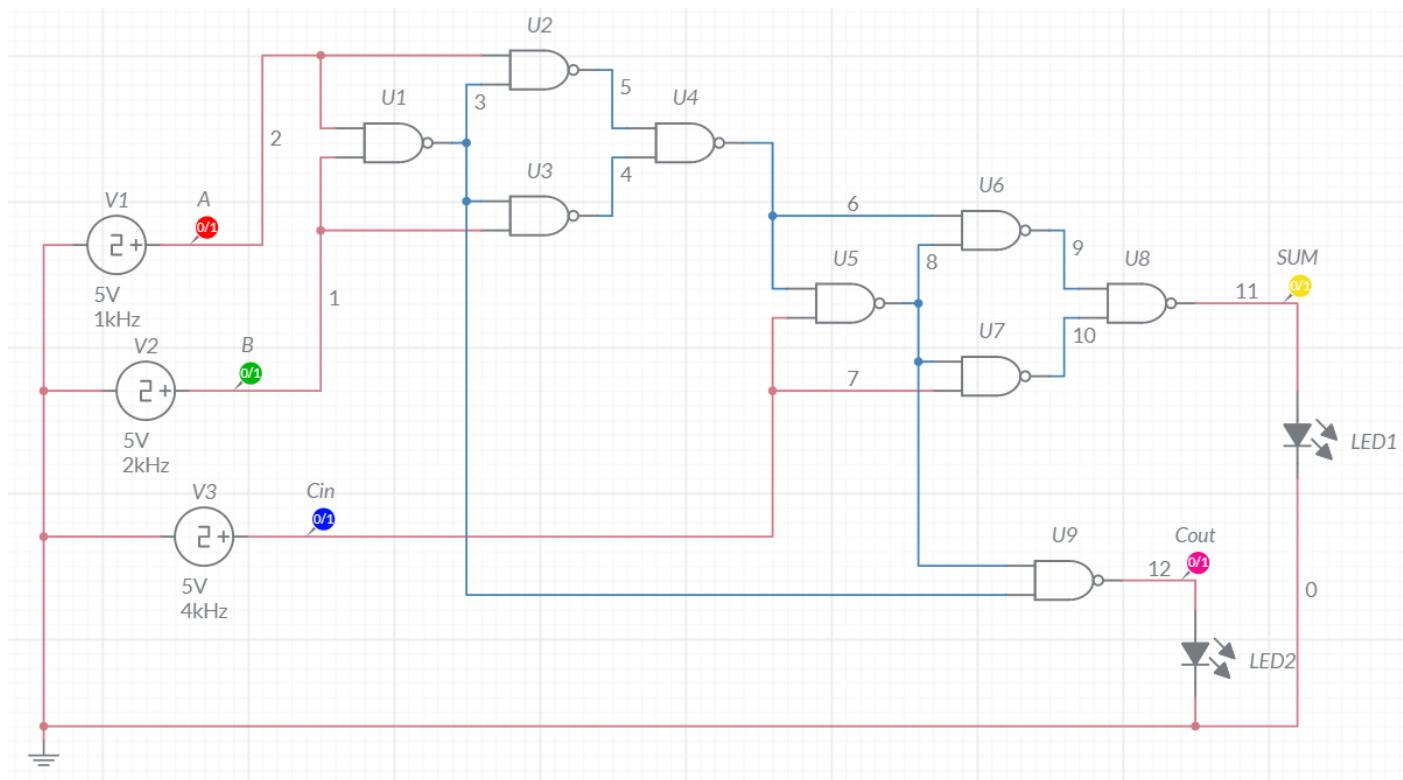
# **ASSIGNMENT-4**

U19CS012

**Design the below given circuits. Verify their Functionality with the help of Multisim.**

**1. Full Adder using least number of NAND Gates.**

*a.) Implement the circuit in Multisim online*

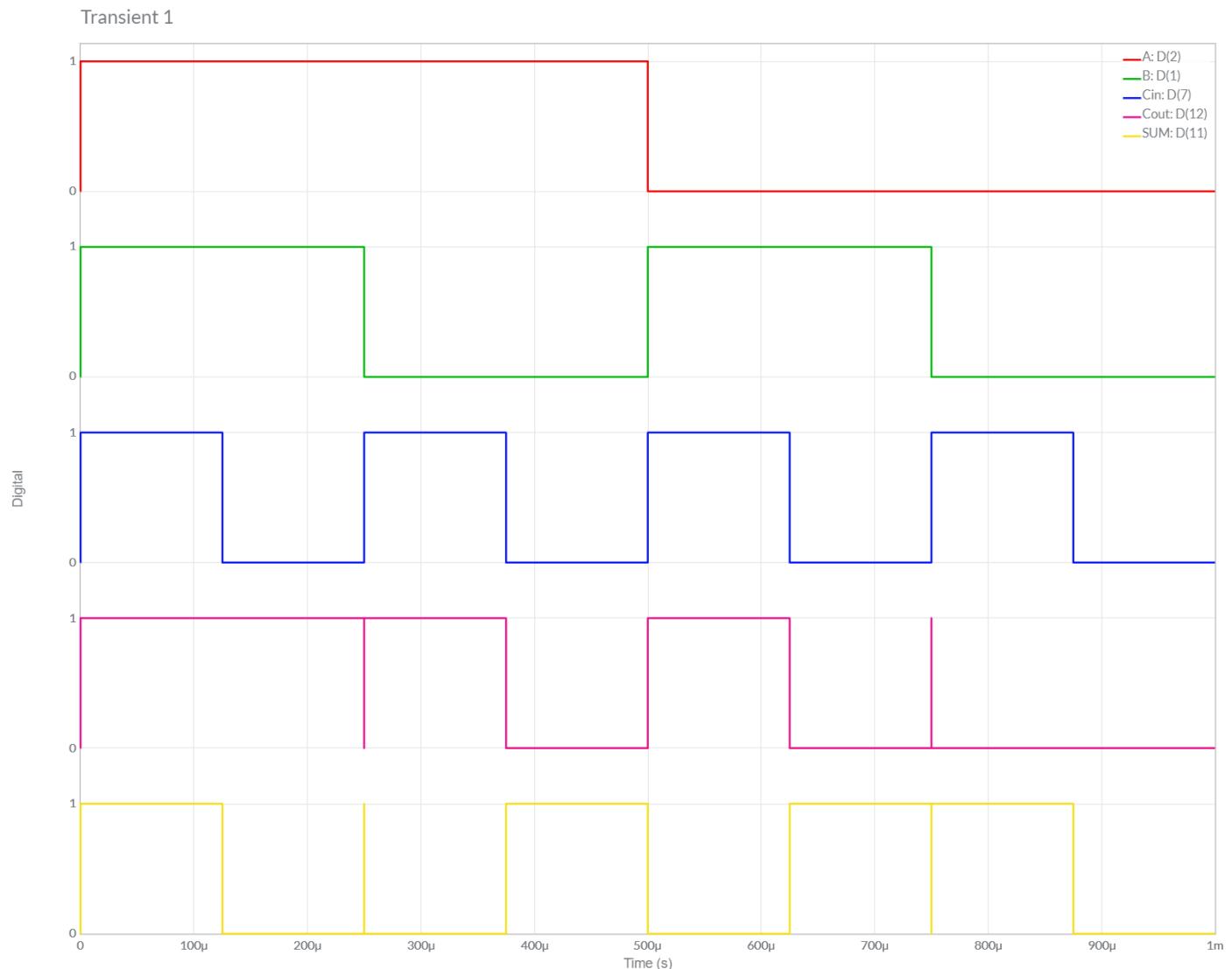


**b.) Truth Table**

	INPUTS			OUTPUTS	
	A	B	Cin	Cout	SUM
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	1	1



## c.) Timing Graph



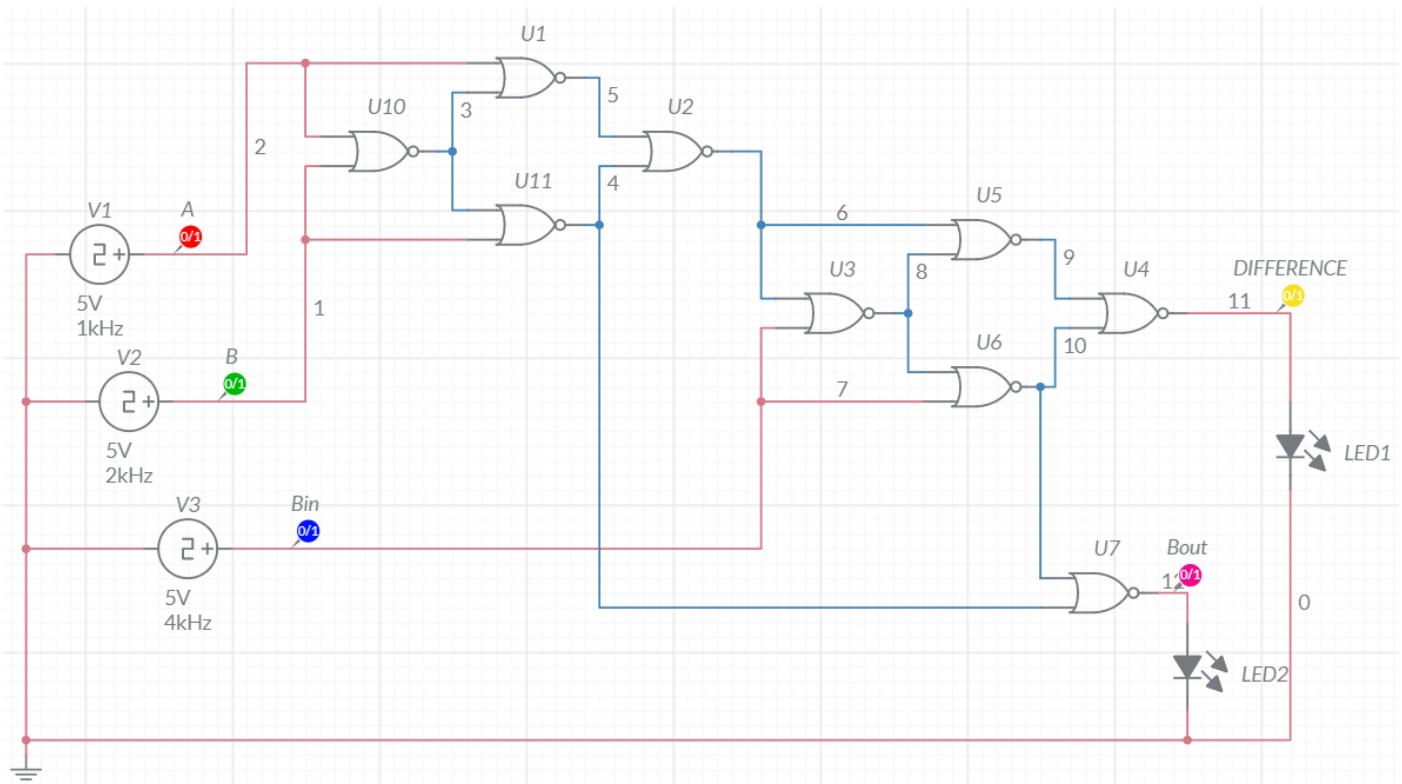
## d.) Conclusion

We can observe from Above Graph and Truth Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.  
 Hence, Experiment is Performed Successfully (without any Error) & Functionality of Circuit is verified.



## 2. Full Subtractor using least number of NOR Gates.

a.) Implement the circuit in Multisim online

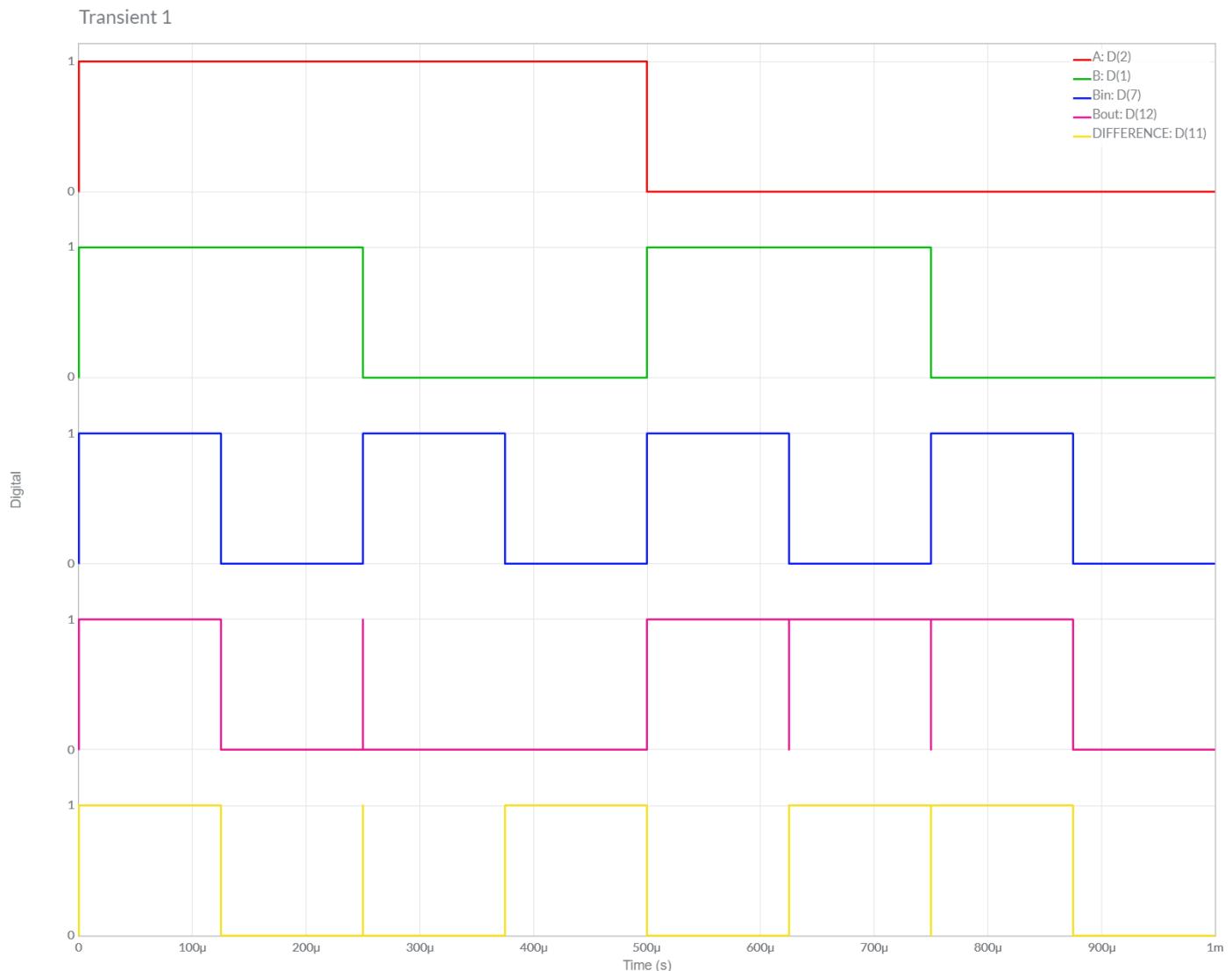


c.) Truth Table

	INPUTS			OUTPUTS	
	A	B	Bin	Bout	DIFF
1	0	0	0	0	0
2	0	0	1	1	1
3	0	1	0	1	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	0	0
7	1	1	0	0	0
8	1	1	1	1	1



## b.) Timing Graph



## d.) Conclusion

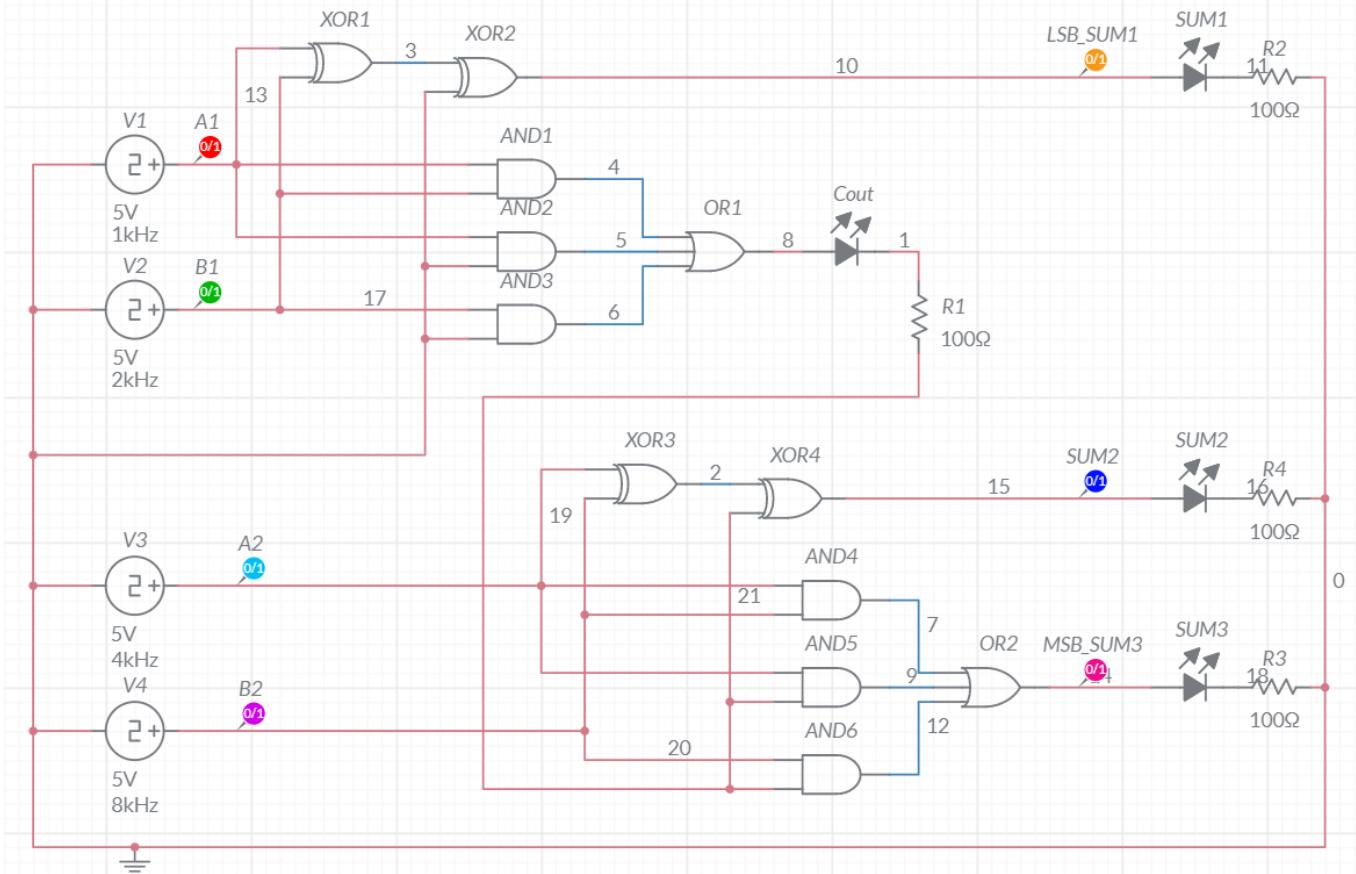
We can observe from Above Graph and Truth Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

Hence, Experiment is Performed Successfully (without any Error) & Functionality of Circuit is verified.

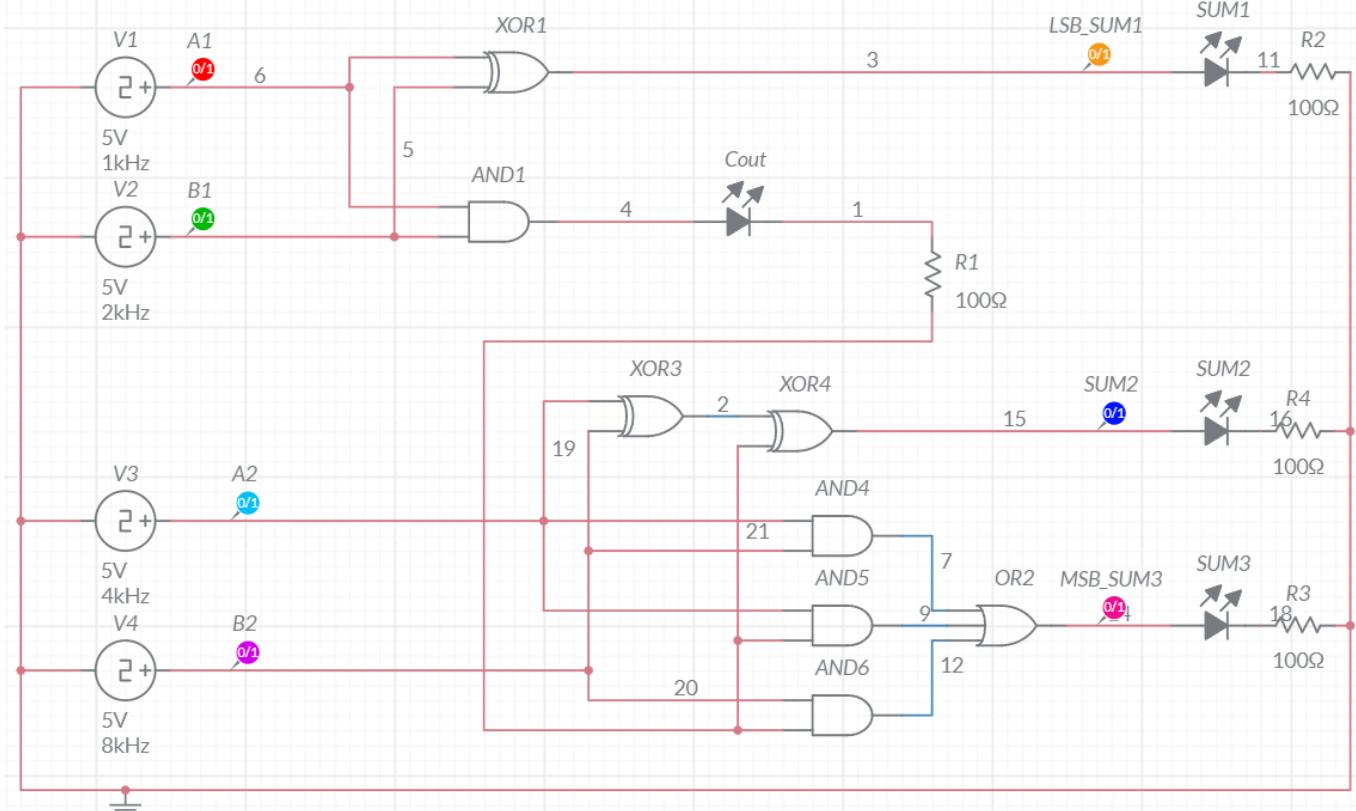


### 3. Two bit Adder circuit using Full Adders. Attach screenshots for any four input combinations.

a.) Implement the circuit in Multisim online [Using 2 Full Adders]

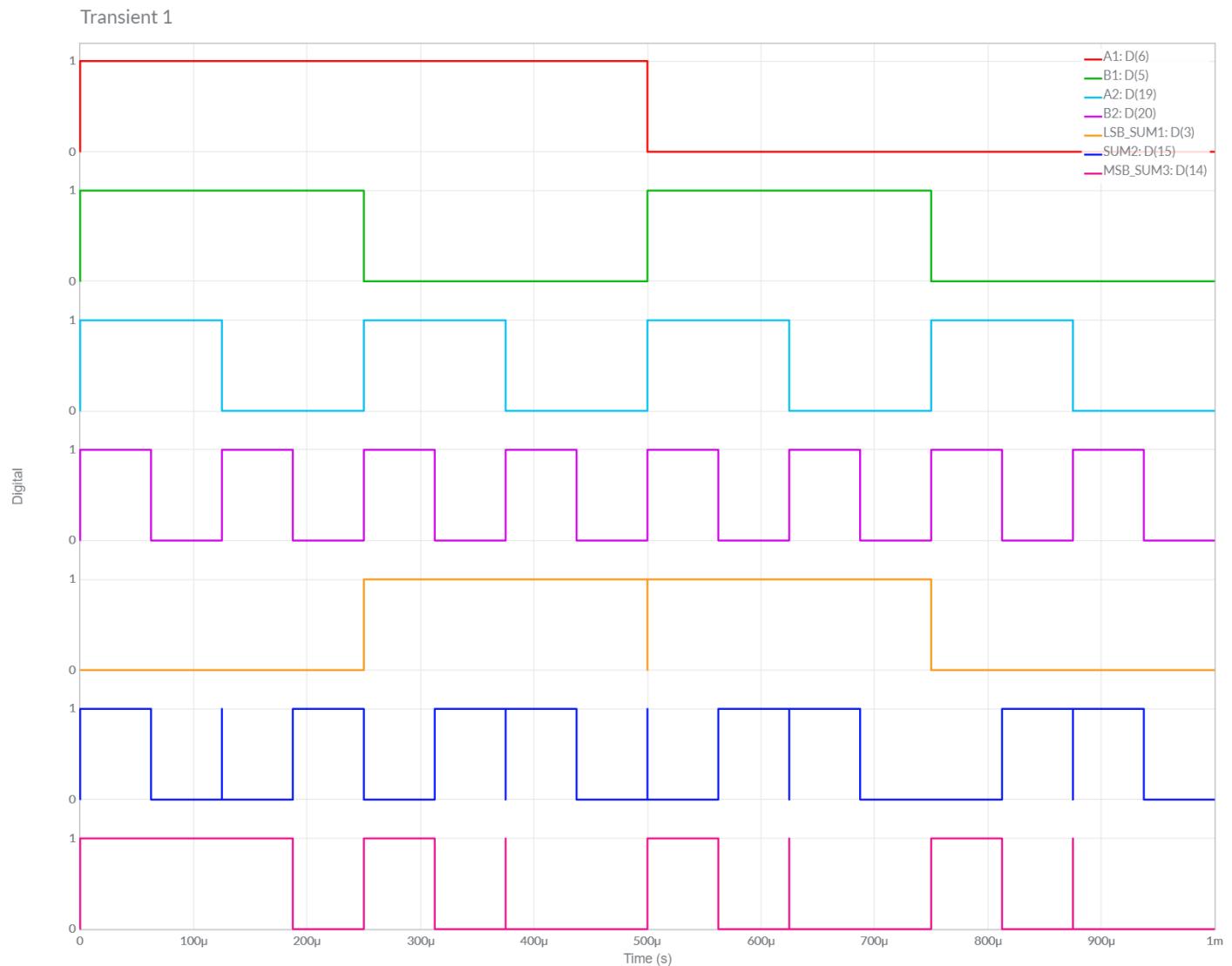


b.) Implement the circuit in Multisim online [Using 1 Full Adder + 1 Half Adder] [Optimized]





*c.) Timing Graph [Same for Both Circuit]*





## c.) Truth Table

	INPUTS				OUTPUTS		
	A1	B1	A2	B2	LSB(SUM1)	SUM2	MSB(SUM3)
1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	0
3	0	0	1	0	0	1	0
4	0	0	1	1	0	0	1
5	0	1	0	0	1	0	0
6	0	1	0	1	1	1	0
7	0	1	1	0	1	1	0
8	0	1	1	1	1	0	1
9	1	0	0	0	1	0	0
10	1	0	0	1	1	1	0
11	1	0	1	0	1	1	0
12	1	0	1	1	1	0	1
13	1	1	0	0	0	1	0
14	1	1	0	1	0	0	1
15	1	1	1	0	0	0	1
16	1	1	1	1	0	1	1

## d.) Conclusion

We can observe from Above Graph and Truth Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

Hence, Experiment is Performed Successfully (without any Error) & Functionality of Circuit is verified.



Expt. No:

5

Date:

10/09/2020

**V-I Characteristics of PN – Junction Diode**

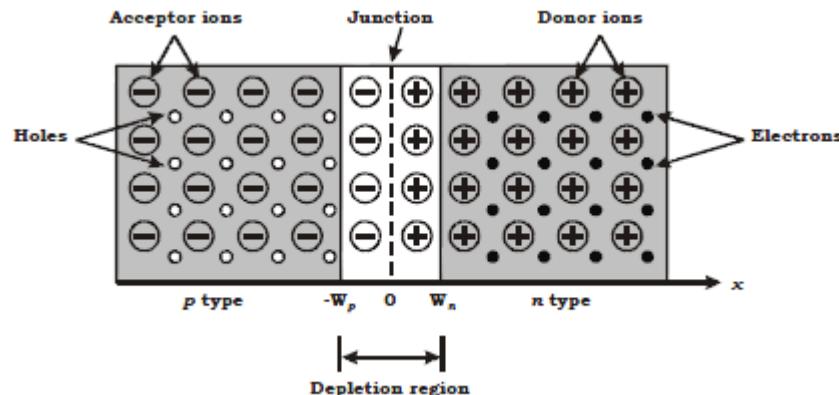
**AIM:** To obtain and plot both forward and reverse characteristics of PN – Junction Diode.

**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.

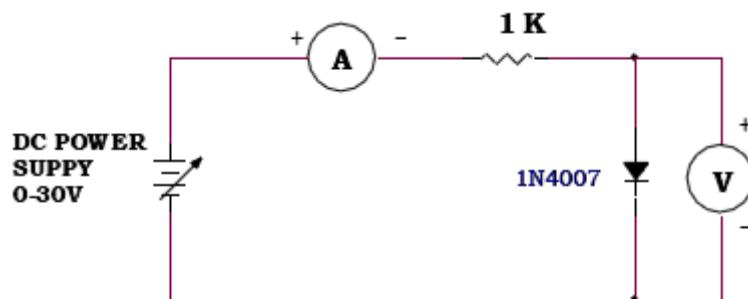
## 2) Reverse-biased:

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.

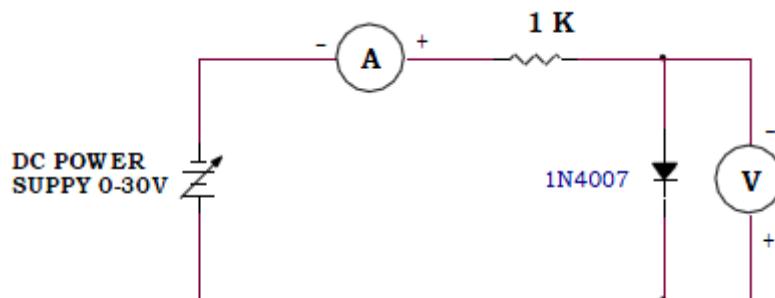
In short, current flows through diode in forward bias and does not flow through diode in reverse bias. Diode can pass current only in one direction.

### CIRCUIT DIAGRAM:

Forward bias:



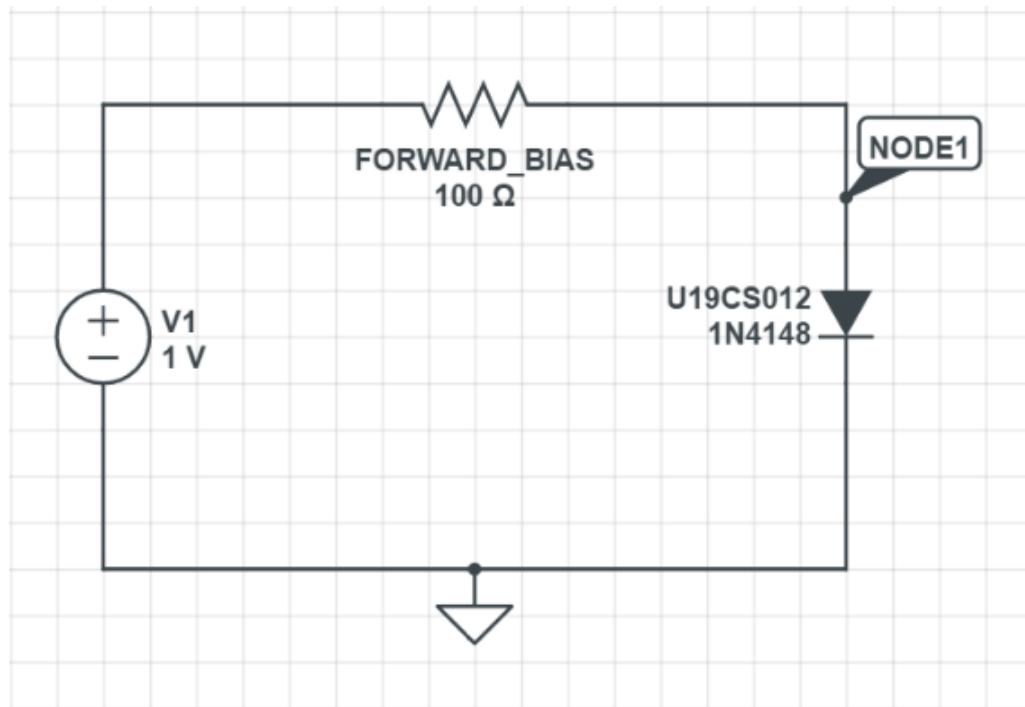
Reverse bias:



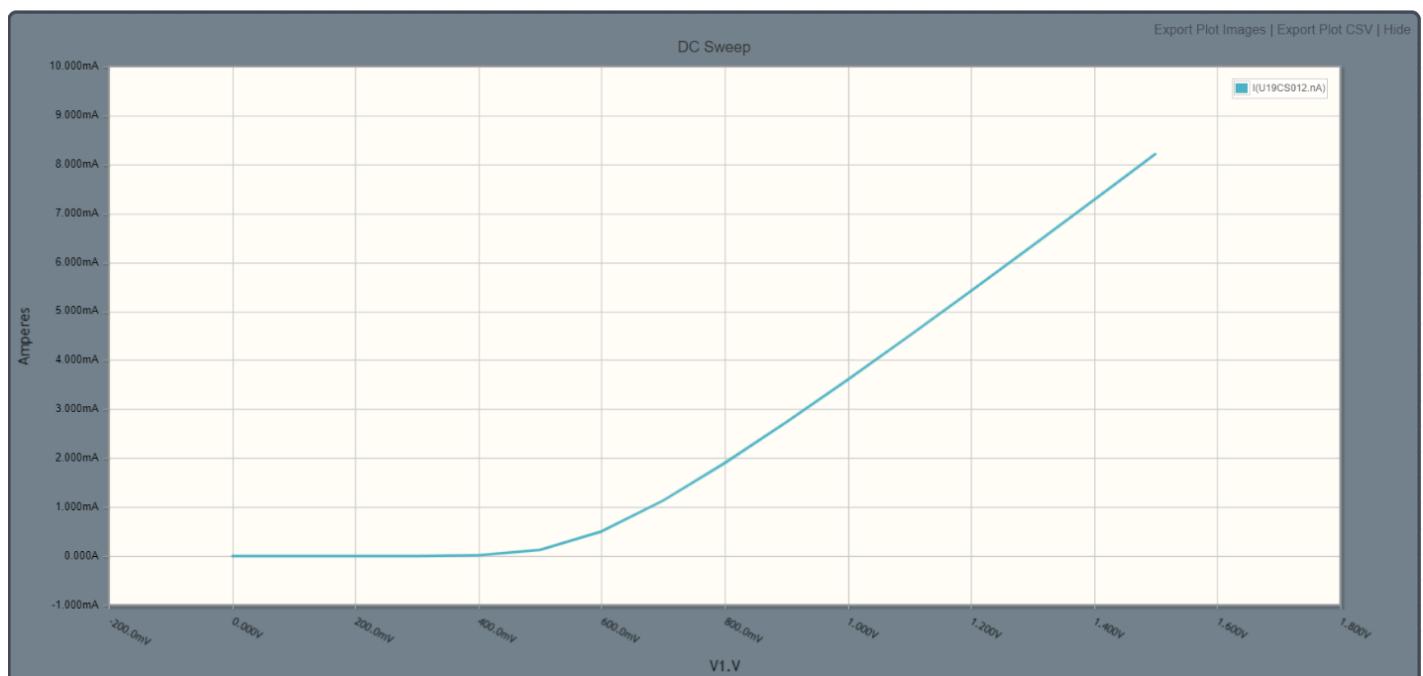


**FORWARD CHARACTERISTICS:**

**CIRCUIT/CONNECTION DIAGRAMS (FROM SIMULATOR)**



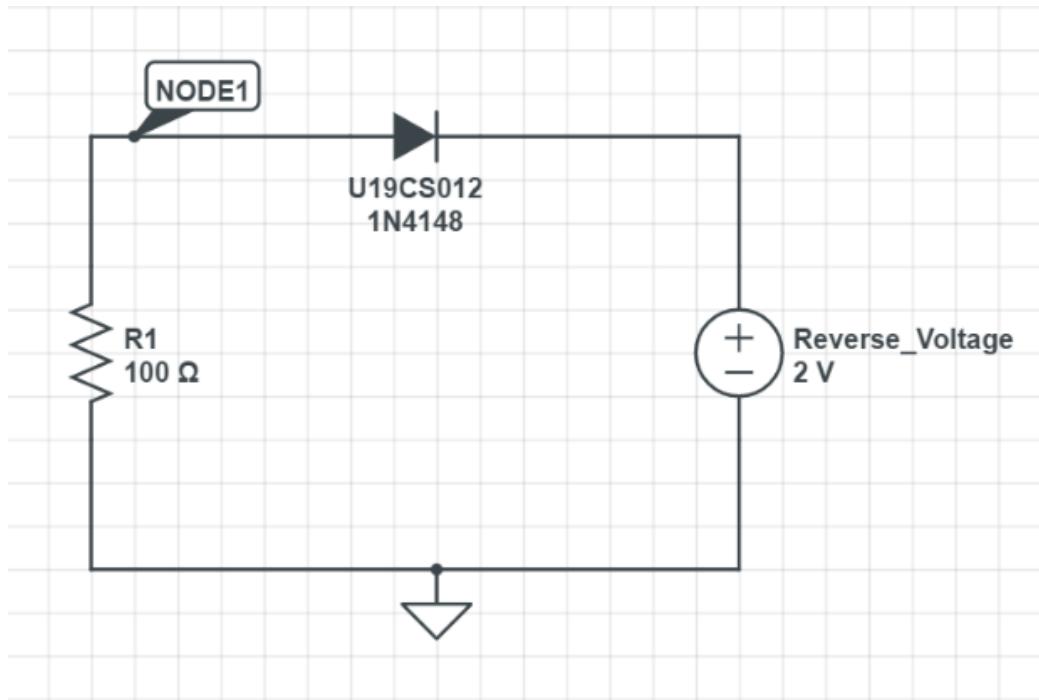
**VI-PLOT (FROM SIMULATOR/GRAFH)**



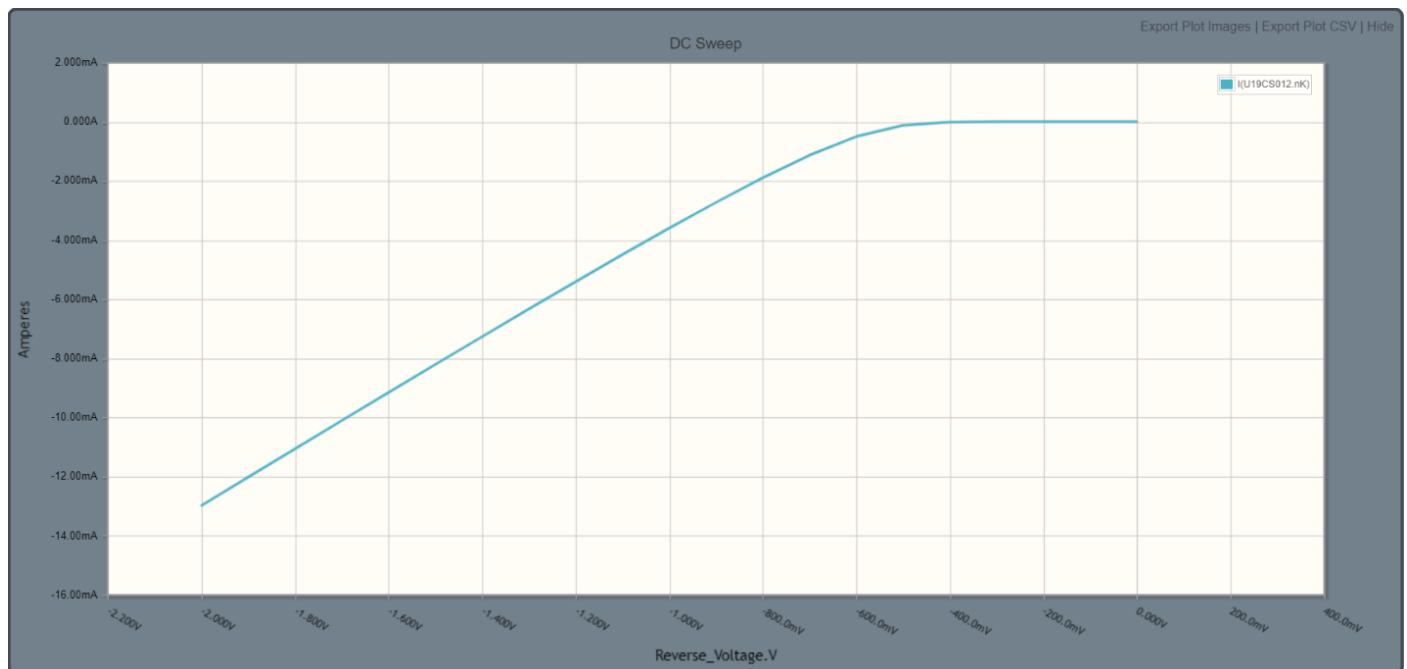


**REVERSE CHARACTERISTICS:**

**CIRCUIT/CONNECTION DIAGRAMS (FROM SIMULATOR)**



**VI-PLOT (FROM SIMULATOR/GRAFH)**



**CONCLUSIONS**

- 1.) The I-V Characteristics of Simple Diode in Forward and Reverse Bias have been Verified Practically with above Graphs with Theoretical Knowledge.



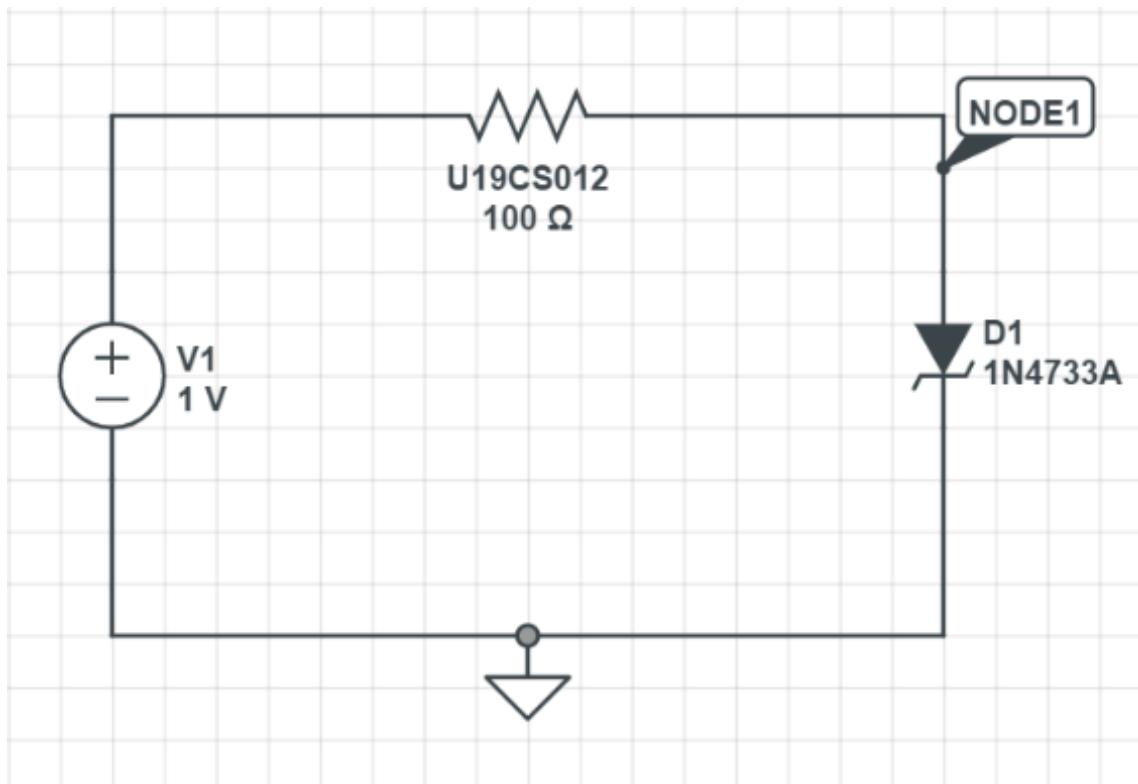
# ASSIGNMENT-5

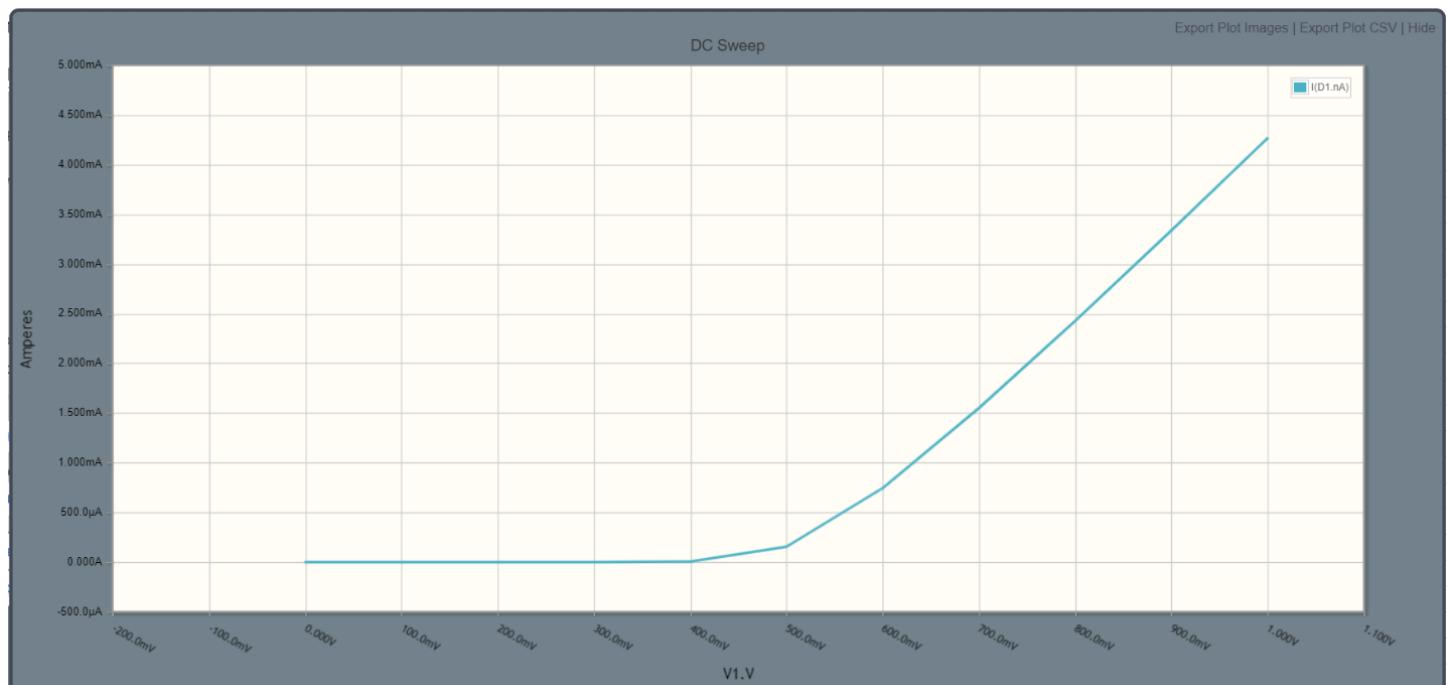
U19CS012

1. Obtain and Plot the forward and reverse characteristics of ZENER Diode in *LabOnline* Simulation Environment.

A.) Forward Characteristics of ZENER Diode

1.) Circuit Image:



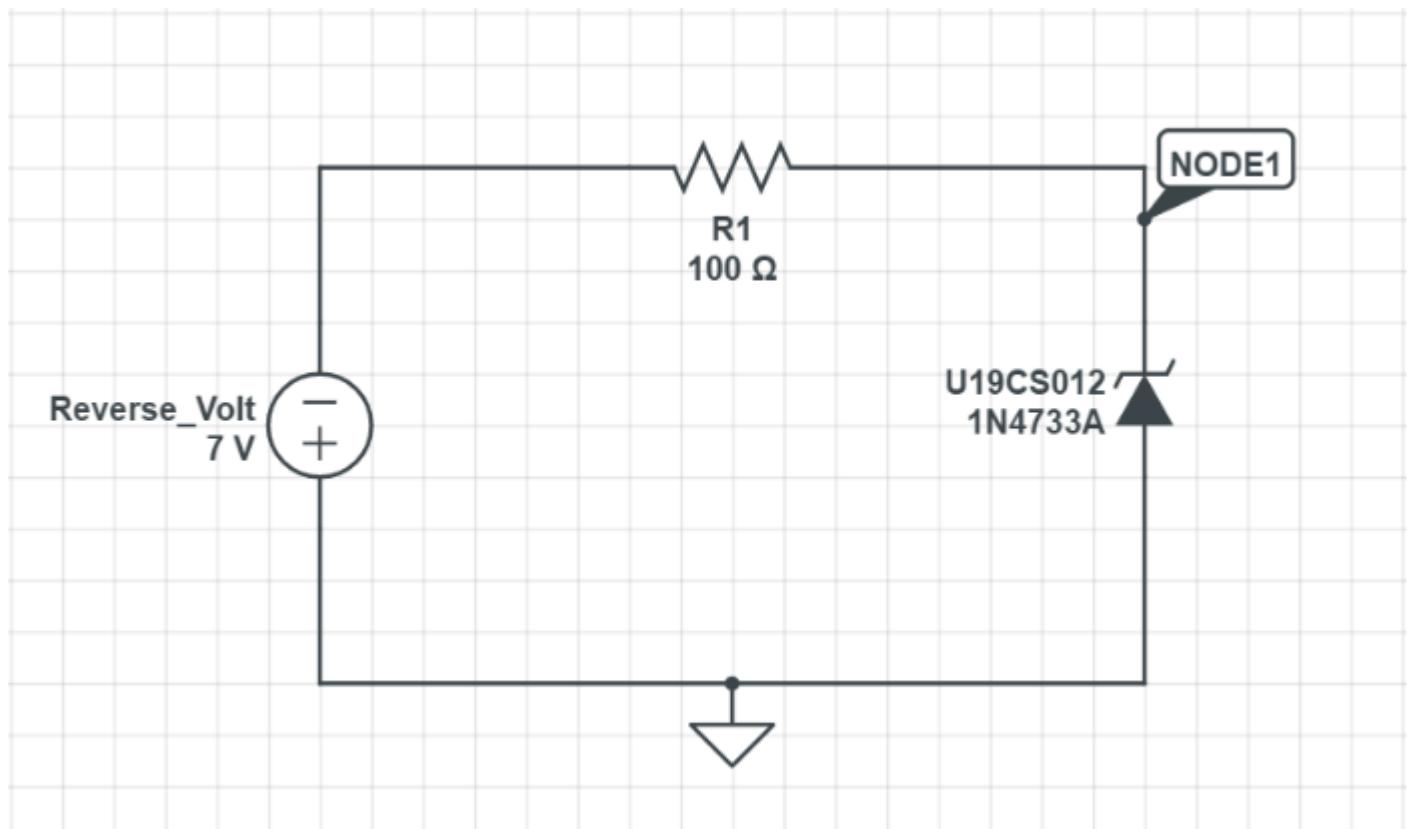
**2.) Graph:****3.) Explanation:**

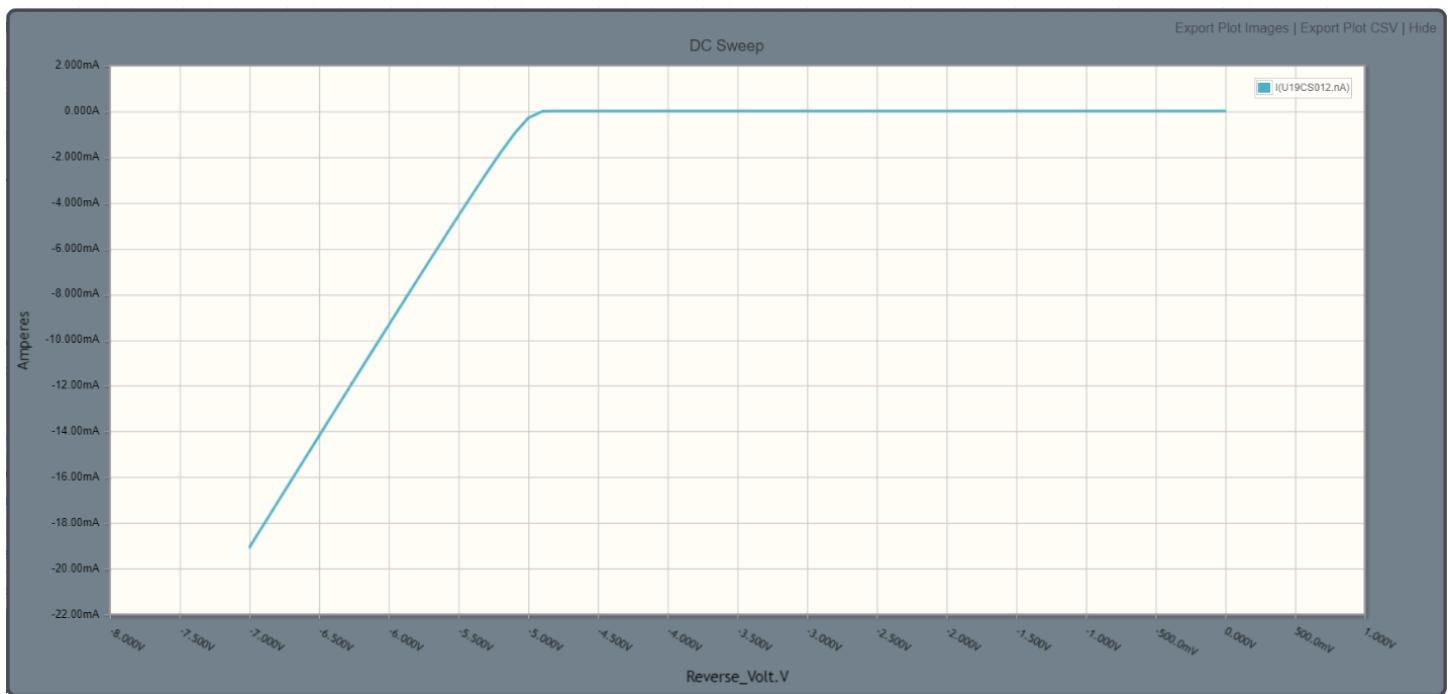
*In Forward Bias, ZENER Diode Follows the Same I-V Characteristic Curve as Simple Diode in Forward Bias. Forward Voltage = 0.5 Volts*



## B.) Reverse Characteristics of ZENER Diode

### 1.) Circuit Image:



**2.) Graph:****3.) Explanation:**

In Reverse Bias, ZENER Diode Does not follows the Same I-V Characteristic Curve as Simple Diode in Reverse Bias, Instead a Sharp Drop in Current is observed. This is Called ZENER Breakdown Voltage.

Breakdown Voltage = -5 Volts

**C.) Conclusion:**

ZENER Diode I-V Characteristics have been Practically Verified through this Simulation with the Theoretical Knowledge. Hence Experiment was Performed Successfully. [Without Error]



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:**

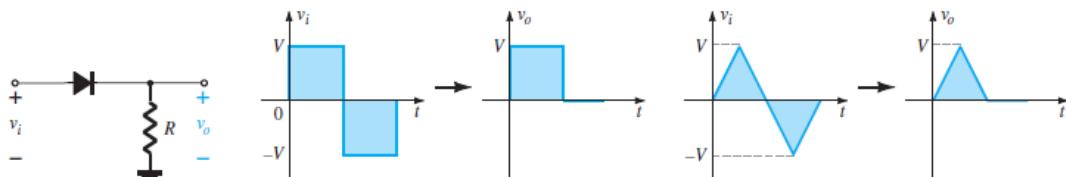
1. Multisim Simulator/Circuit Simulator

**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**

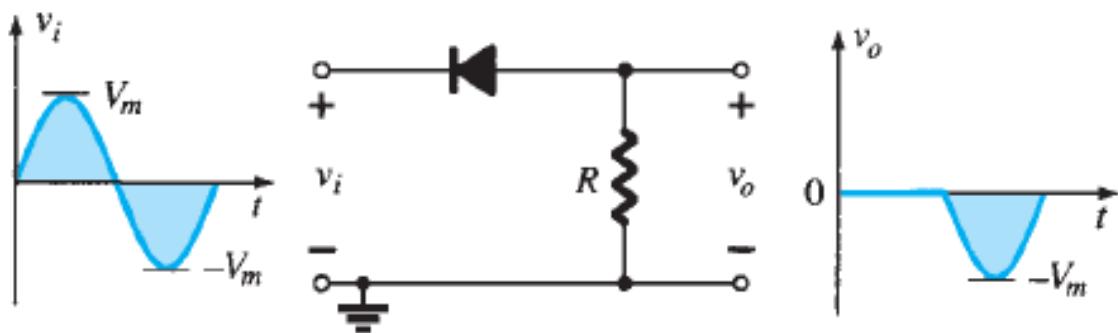
As shown above, when the positive half cycle appears, the diode being forward biased, acts as short circuit and allows the input voltage to appear across the load resistor. During the negative half cycle, the diode is reverse biased, acts as open circuit and hence we see that there is no connection between 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.

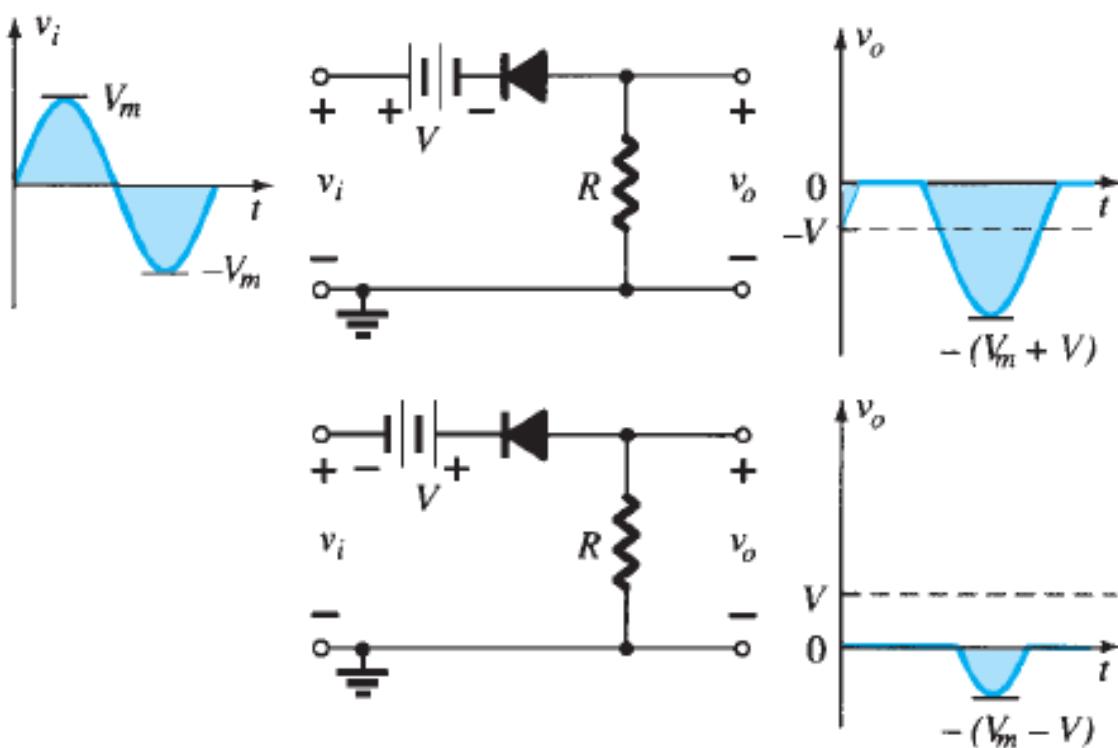
**FEW SERIES DIODE CLIPPER CONFIGURATIONS**



### POSITIVE

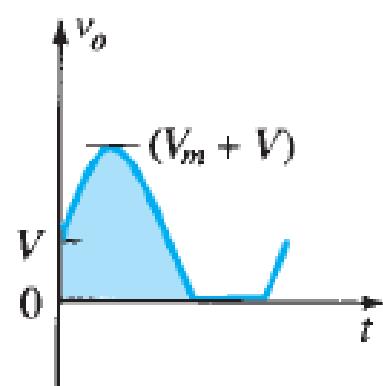
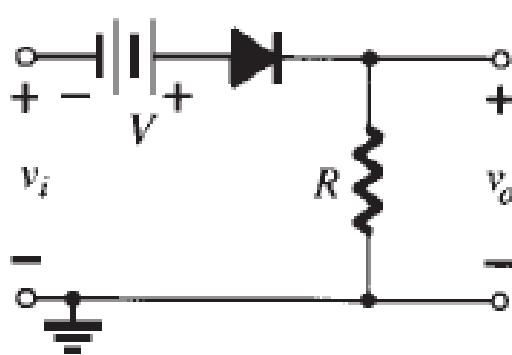
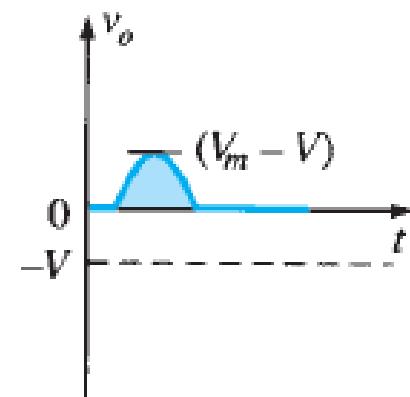
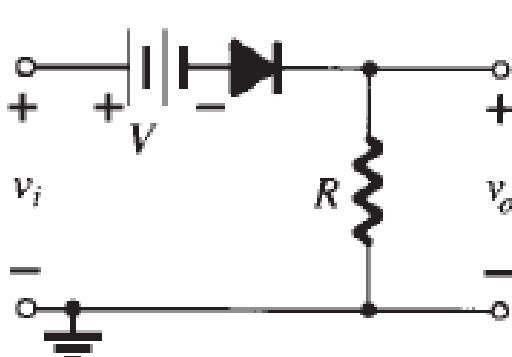
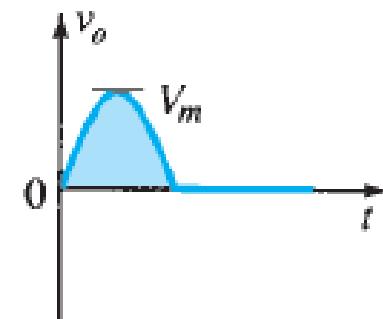
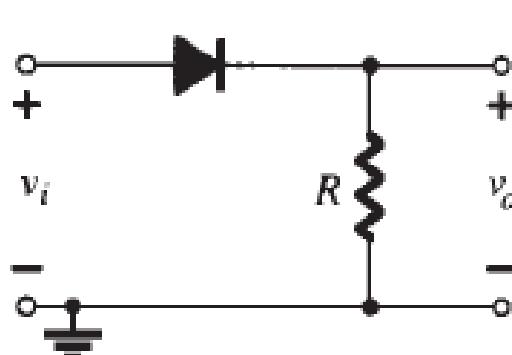


### Biased Series Clippers (Ideal Diodes)





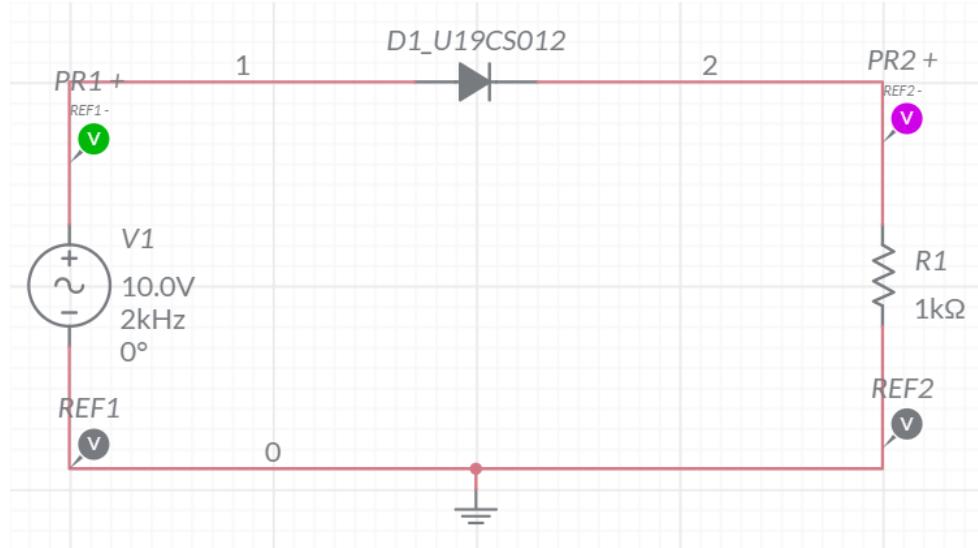
**NEGATIVE**



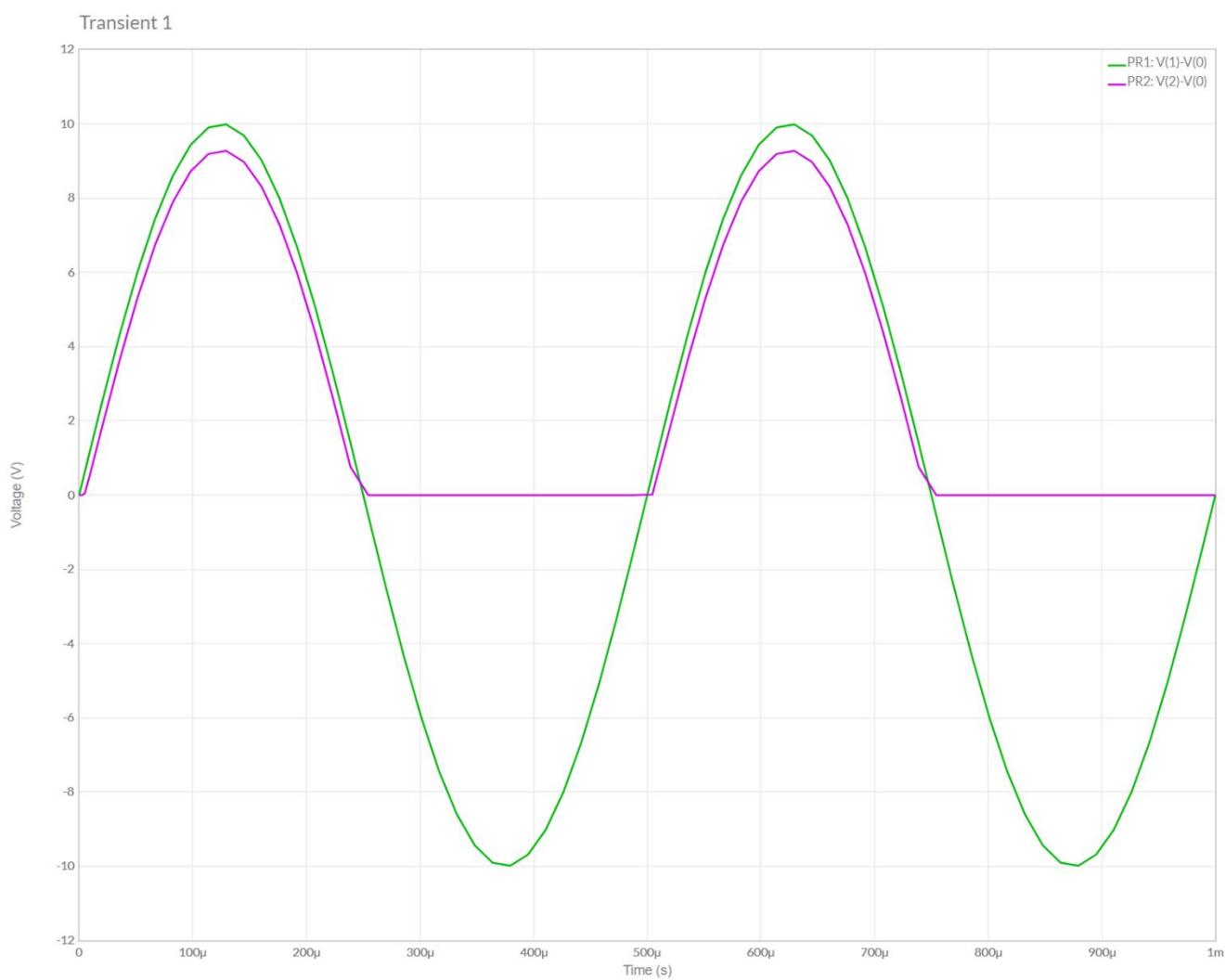


### 1.) NEGATIVE CLIPPER

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



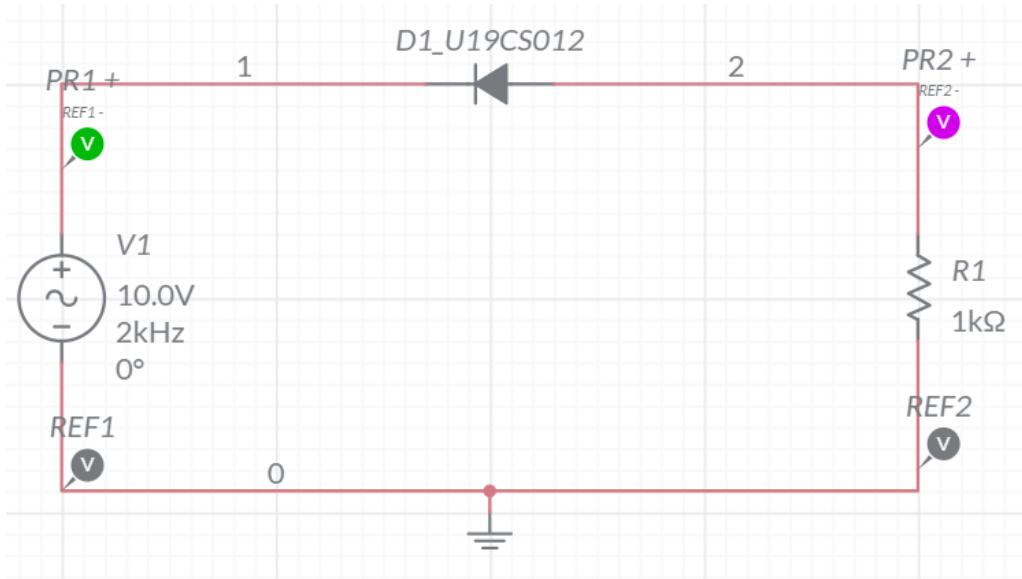
**WAVEFORMS (FROM MULTISIM)**



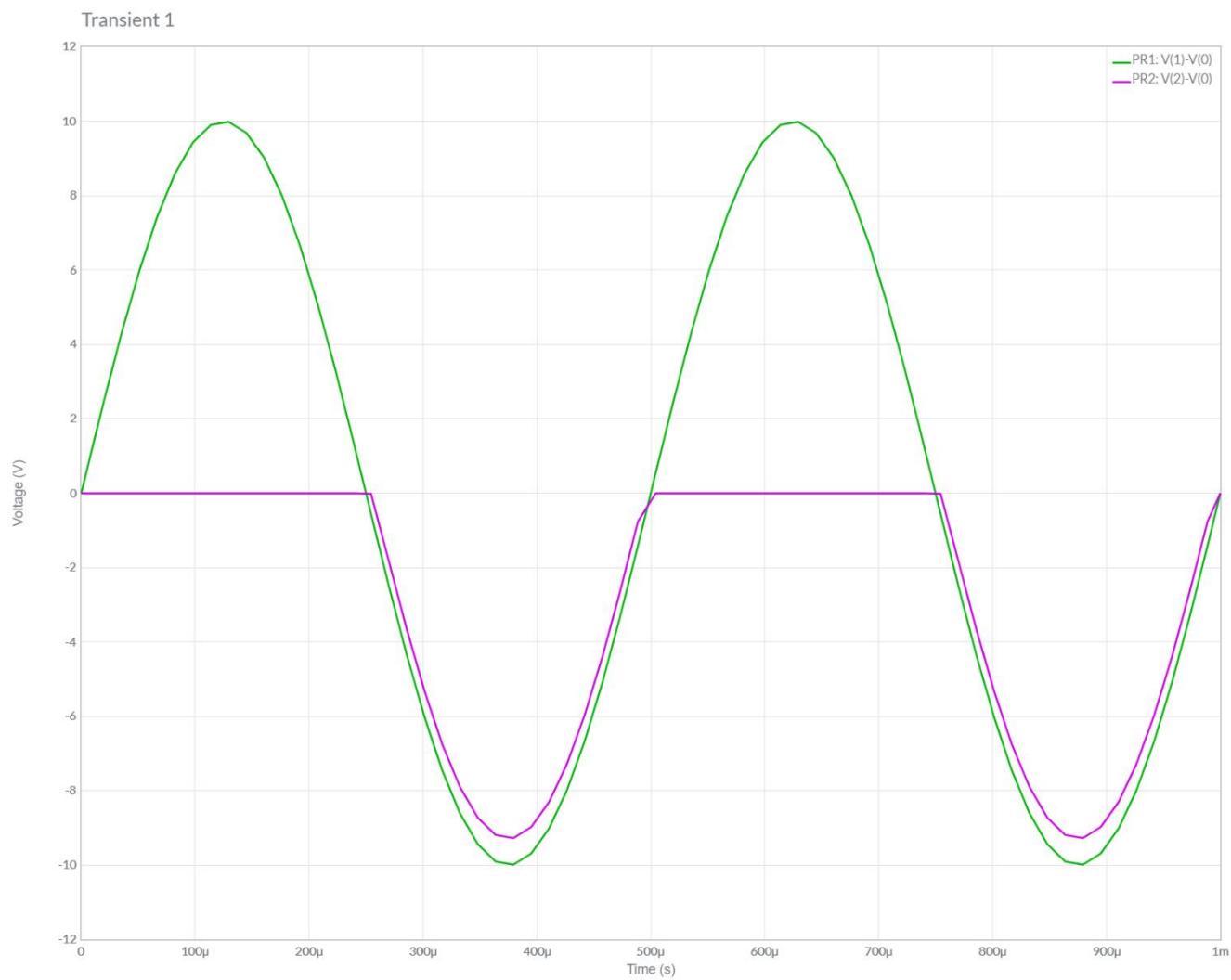


## 2.) POSITIVE CLIPPER

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



**WAVEFORMS (FROM MULTISIM)**



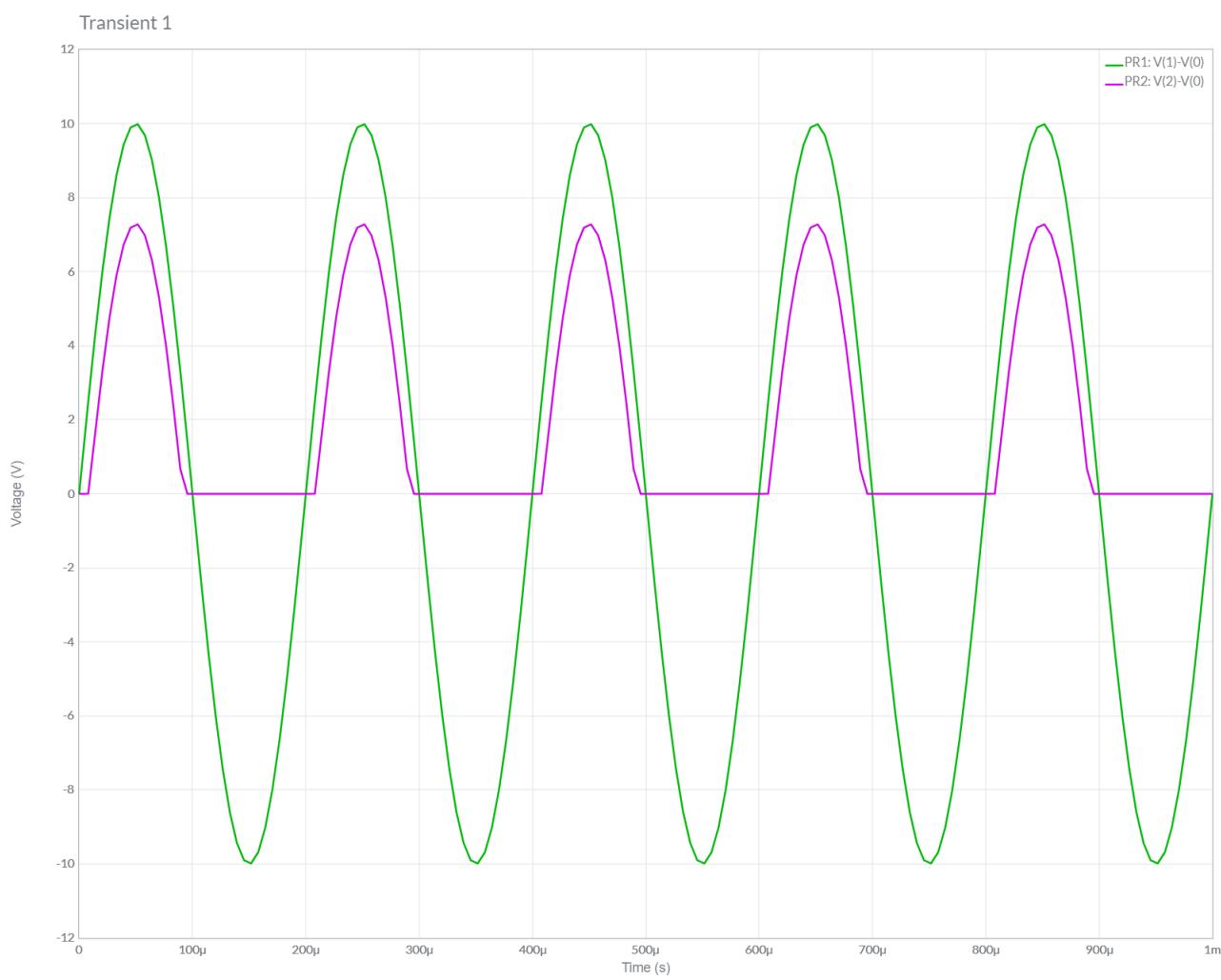


### 3.) NEGATIVE CLIPPER WITH BIAS-I

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



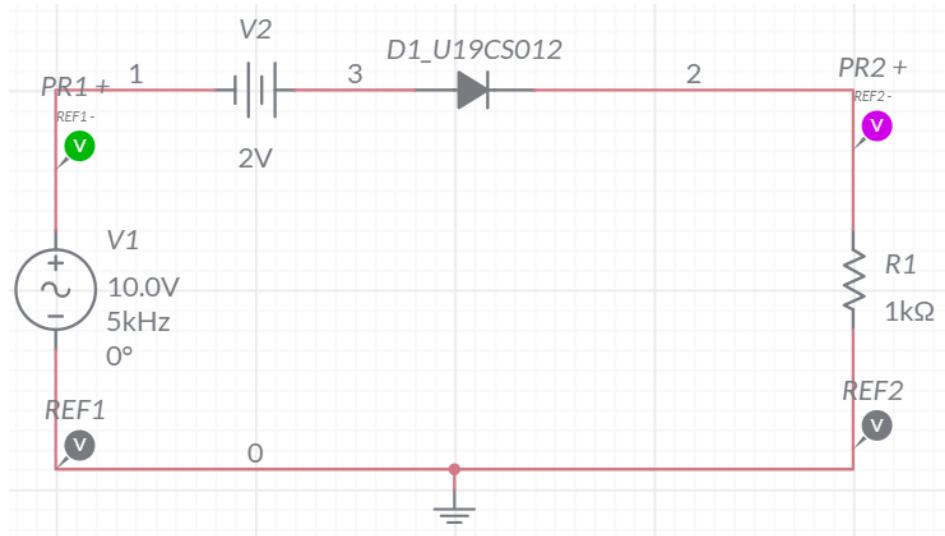
**WAVEFORMS (FROM MULTISIM)**



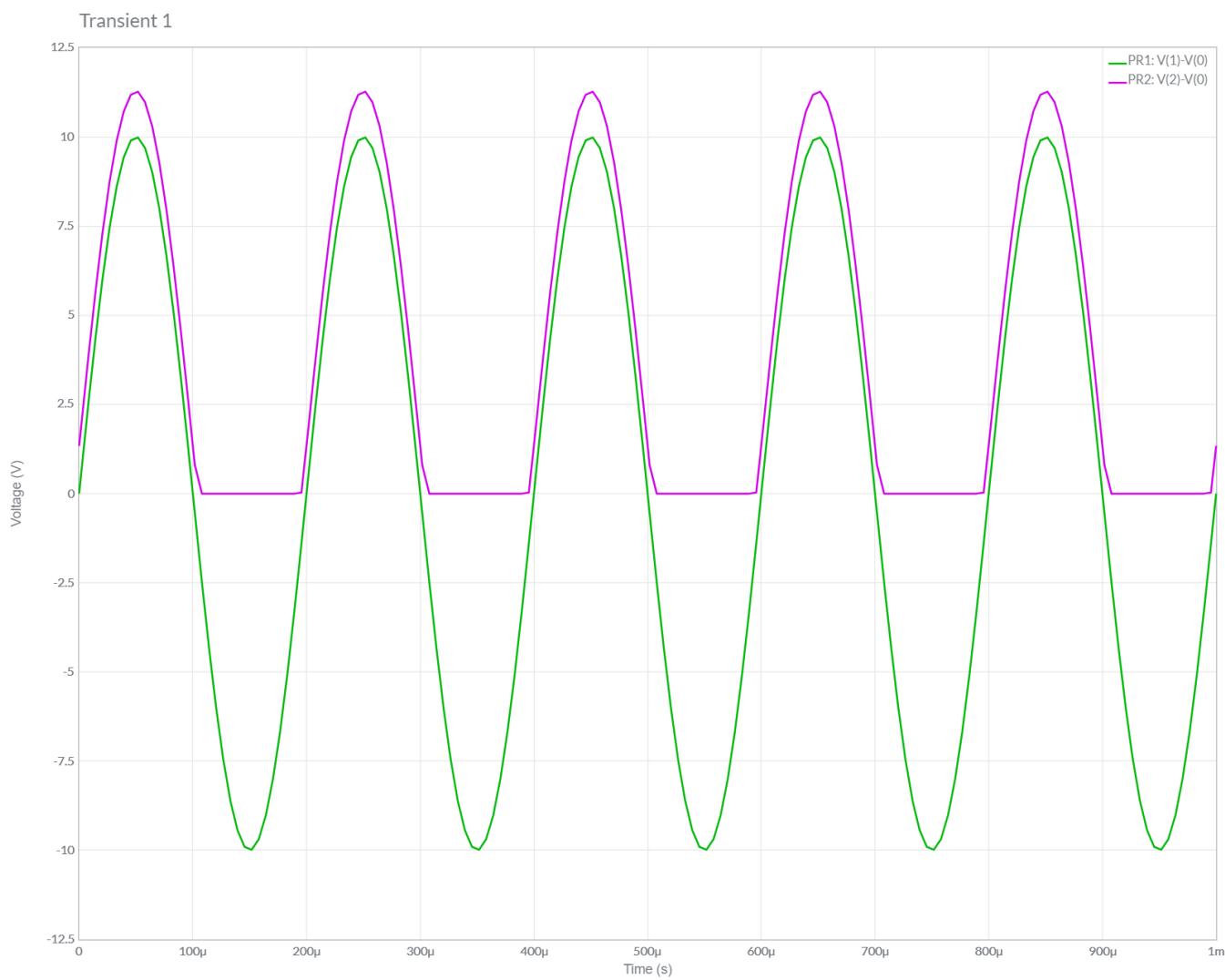


#### 4.) NEGATIVE CLIPPER WITH BIAS-II

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



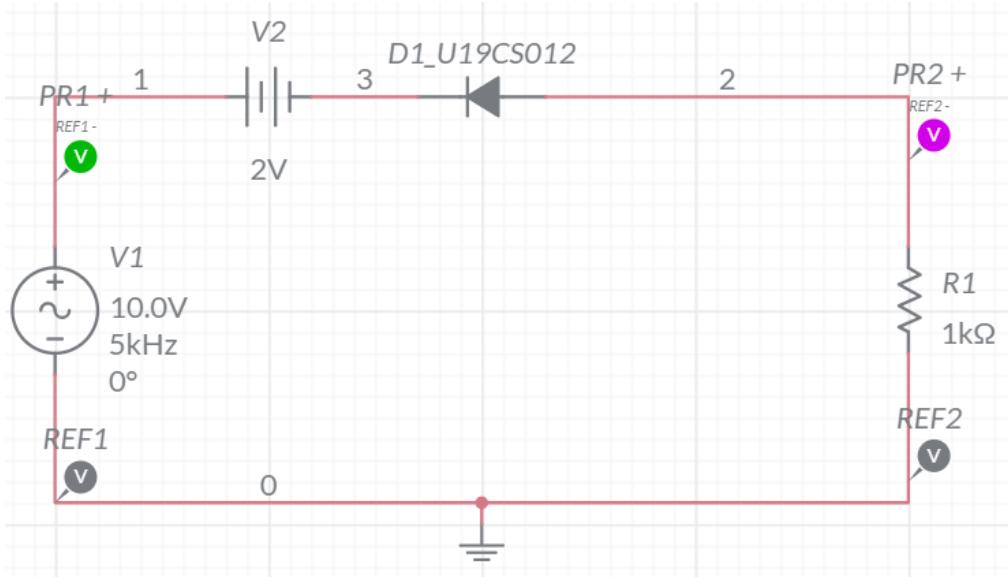
**WAVEFORMS (FROM MULTISIM)**



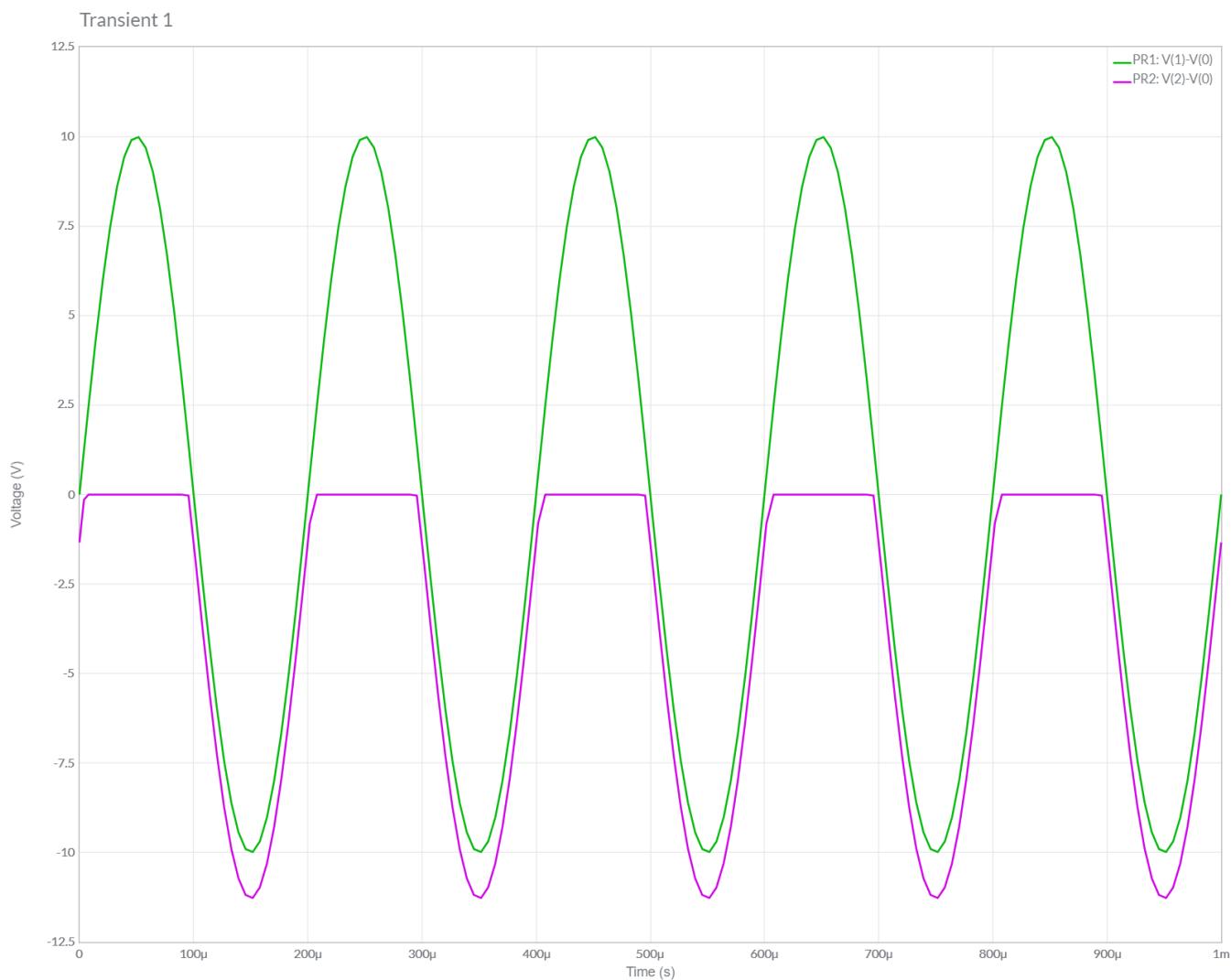


### 5.) POSITIVE CLIPPER WITH BIAS-I

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



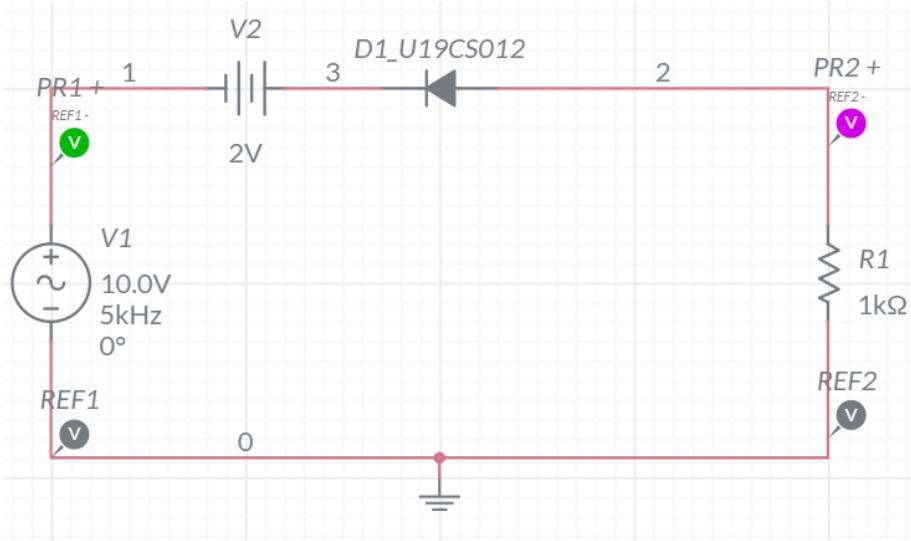
**WAVEFORMS (FROM MULTISIM)**



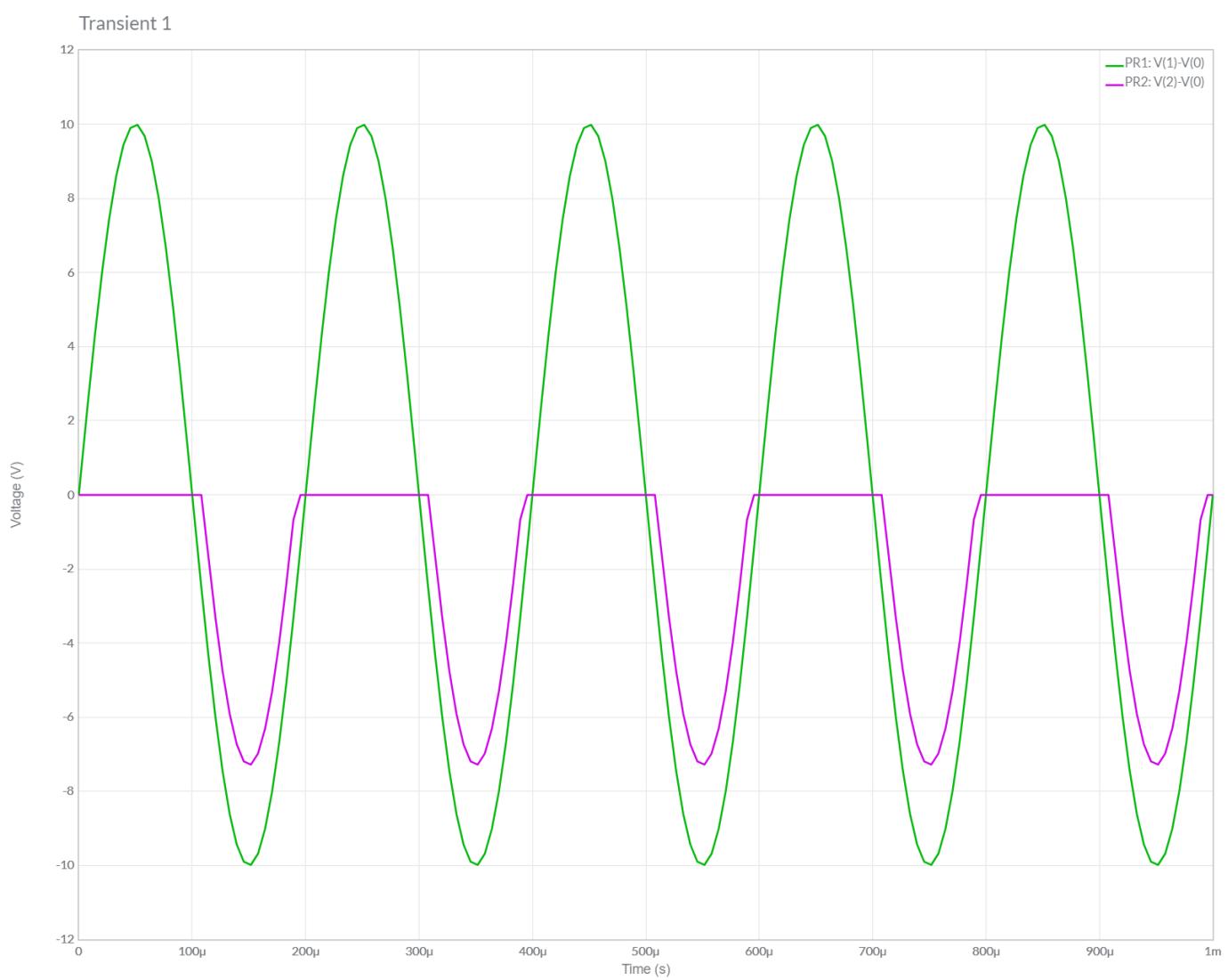


### 6.) POSITIVE CLIPPER WITH BIAS-II

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



**WAVEFORMS (FROM MULTISIM)**





## **CONCLUSIONS**

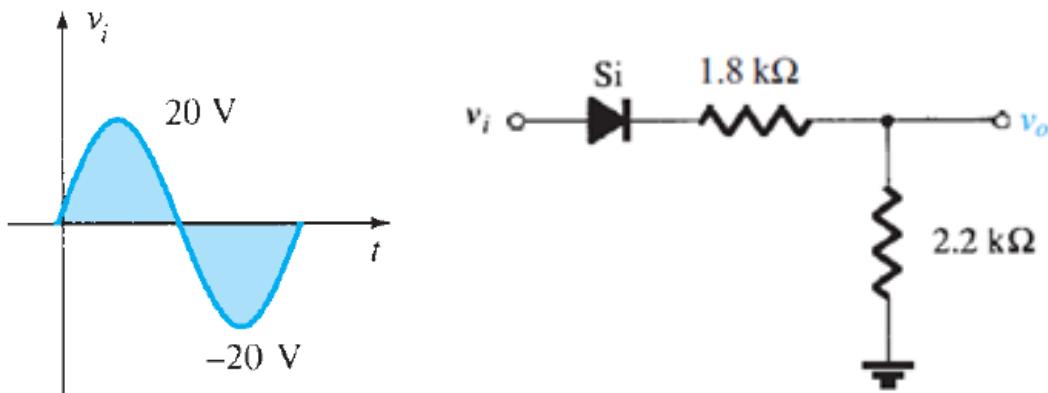
- 1.) In this Experiment, We have studied about Clipper Circuits [Both Positive and Negative] along with Different Biasing Applied.
- 2.) We Verified the Theoretical Knowledge of Series Clippers by Performing Simulations of 6 Cases of Clippers in Series in Multisim.
- 3.) Hence, we have Successfully Designed, Plotted and Verified Various Series Diode Clipper Circuits.



# **ASSIGNMENT-6**

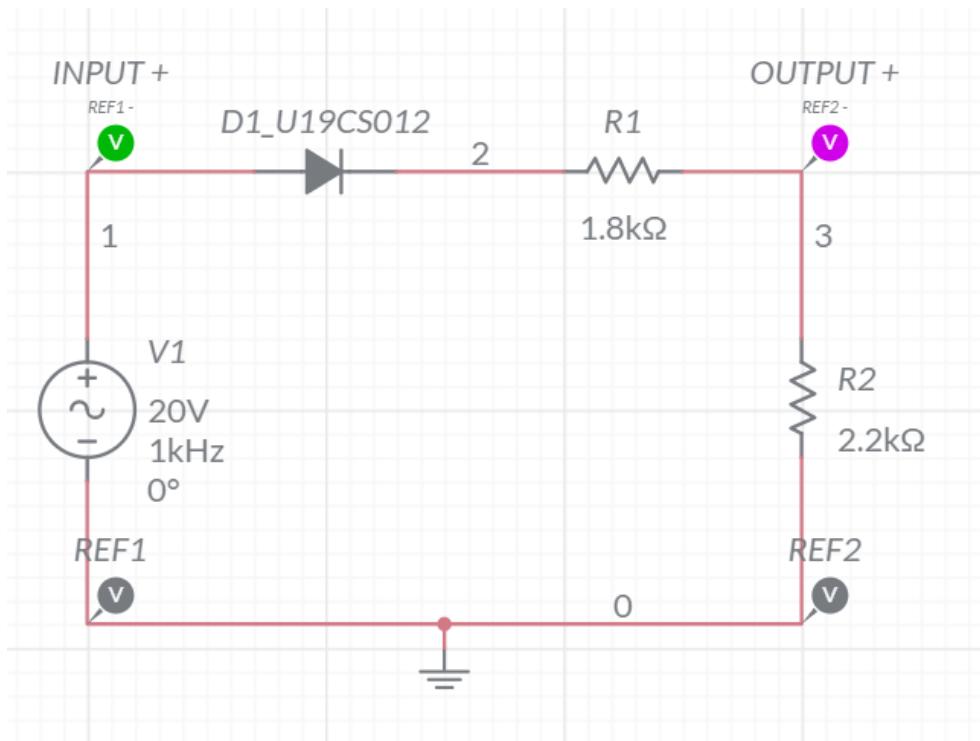
U19CS012

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



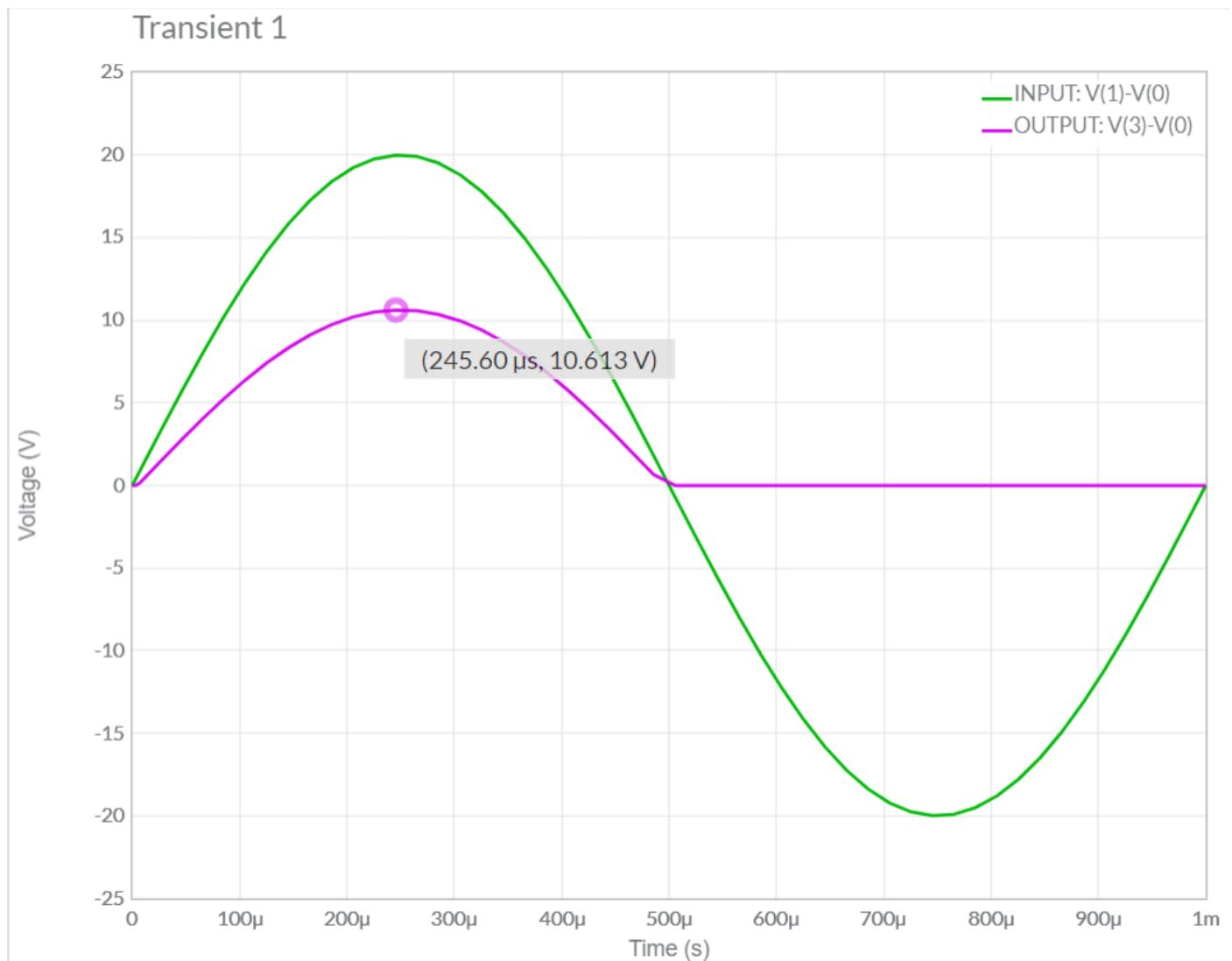
**(I) Multisim Calculations:**

**1.) Circuit Image:**





**2.) Grapher Image:**

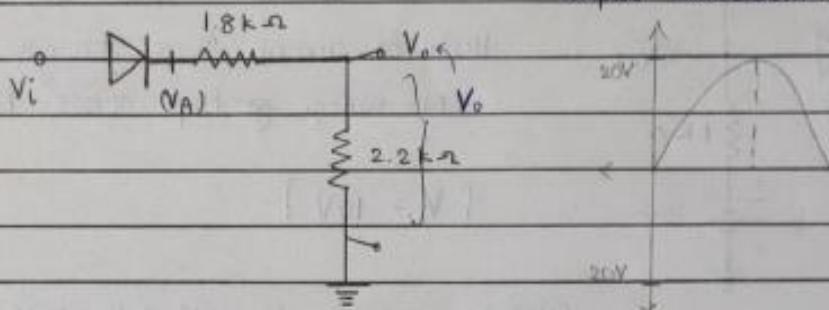




## (II) Theoretical Calculations:

DELD ASSIGNMENT 6 (WNC5012)

Q1.7



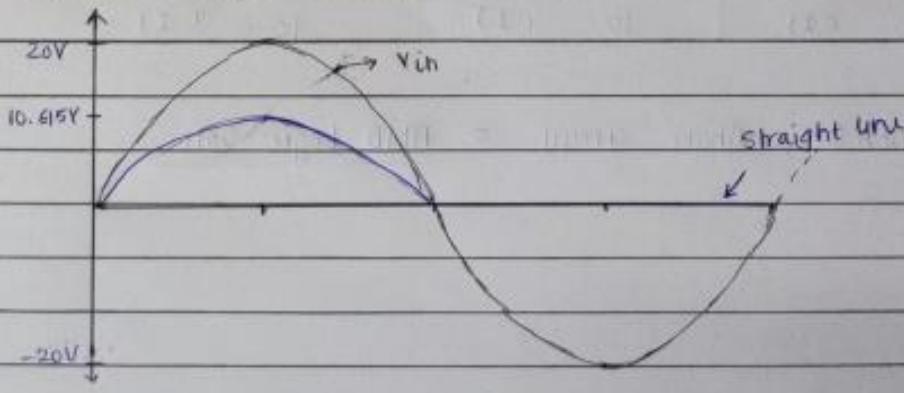
$$\begin{aligned} V_A &= V_{\text{input}} - 0.7 \text{ (forward voltage of } S_1 \text{ diode)} \\ &= (20 - 0.7) \text{ V} \\ &= 19.3 \text{ V} \end{aligned}$$

Applying Voltage division Rule,

$$\begin{aligned} V_o &= V_A \times \frac{(2.2 \text{ k}\Omega)}{(2.2 + 1.8 \text{ k}\Omega)} \\ &= 19.3 \times \left( \frac{2.2}{4} \right) = 10.615 \text{ V} \end{aligned}$$

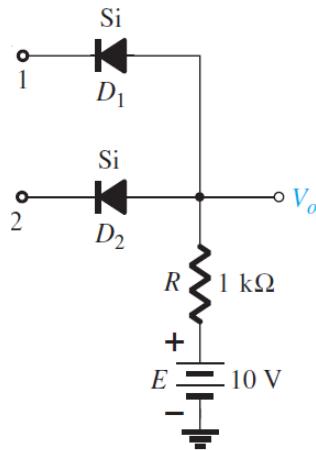
In Forward Bias  $\rightarrow$  (short circuit)Reverse Bias  $\rightarrow$  (open circuit)  $\Rightarrow i=0 \Rightarrow$  No drop across resistor 2.2 kΩ  
[follows input waveform]Positive half cycle  $\uparrow$   $V_o = 10.615 \text{ V}$  at  $V_{\text{in}} = 20 \text{ V}$ Negative half cycle  $V_o = 0 \text{ V}$  [ Negative clipper ]

Expected Output Waveform:



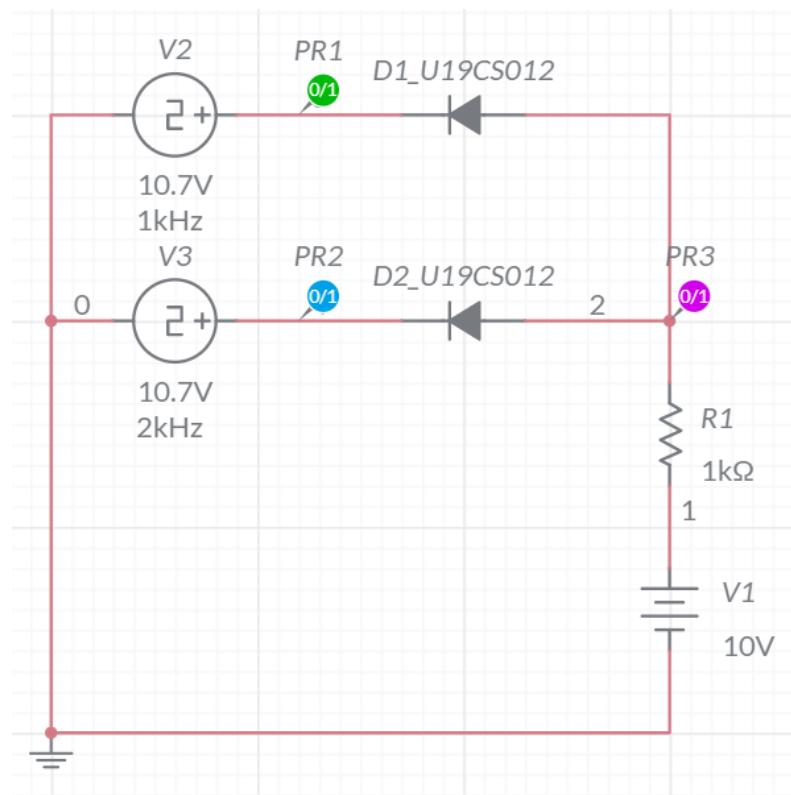


**2. Identify the type of Logic Gate implemented by the below diode configuration. Also verify it using Multisim.**



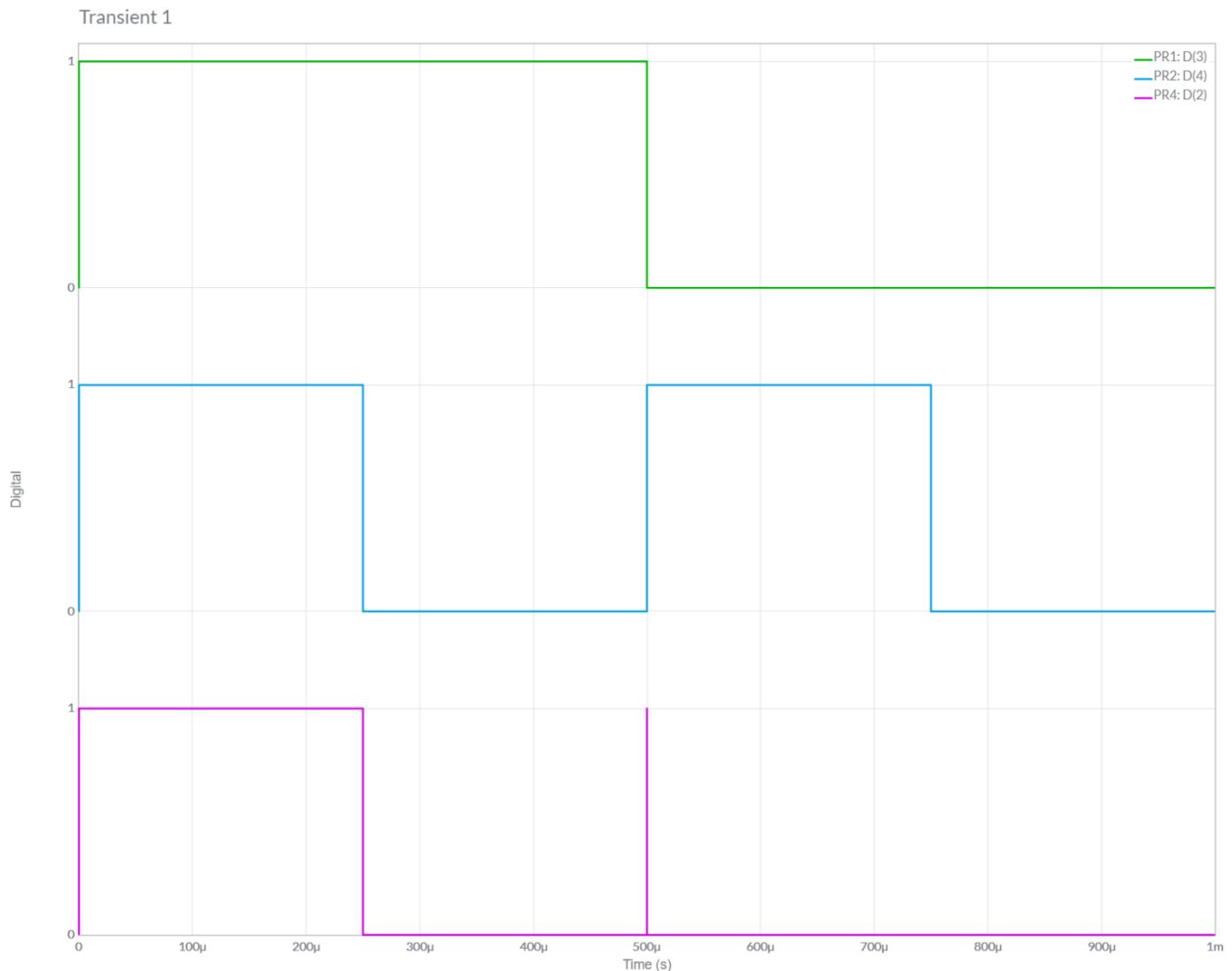
**A.) Multisim Calculations:**

**1.) Circuit Image:**





## **2.) Grapher Image:**

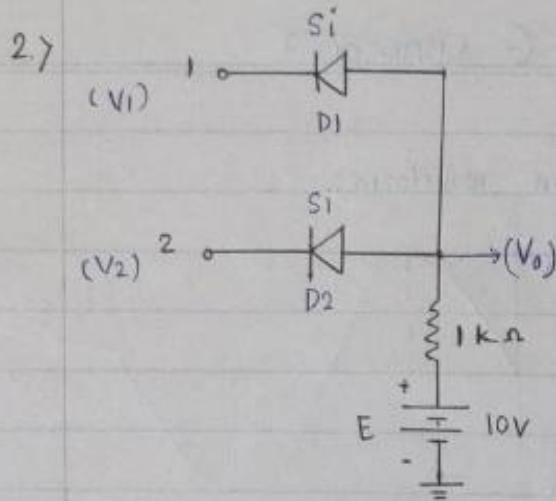


**It Follows the Truth Table of AND Gate. [From Above Graph]**

## **B.) Theoretical Calculations:**



(U19CS012)



CASE 1: Voltage at ① & ② is 10 V

$$V_1 = 10 \quad V_2 = 10$$

∴ circuit becomes open (∴ Both D1 & D2) <sup>reverse</sup>

Hence no current flows through resistor

∴ No voltage drop across 1kΩ resistor

$$[V_0 = 10 \text{ V}]$$

CASE 2: Either  $V_1 = 0 \text{ V}$  &  $V_2 = 10 \text{ V}$

CASE 3:

$$\text{or } V_1 = 10 \text{ V} \& V_2 = 0 \text{ V}$$

$$V_1 = 0 \text{ V} \quad V_2 = 0 \text{ V}$$

Due to one volt = 0 V, the

Both are grounded

Diode becomes Forward bias & acts as

∴ D1 & D2 both forward

short circuit. [ $V_1$  &  $V_2$  are connected

baised

$$\therefore [V_0 = 0 \text{ V}]$$

to ground)  
ie  $\underline{\underline{V_1 = 0 \text{ V}}}$

Truth Table (Theoretical)

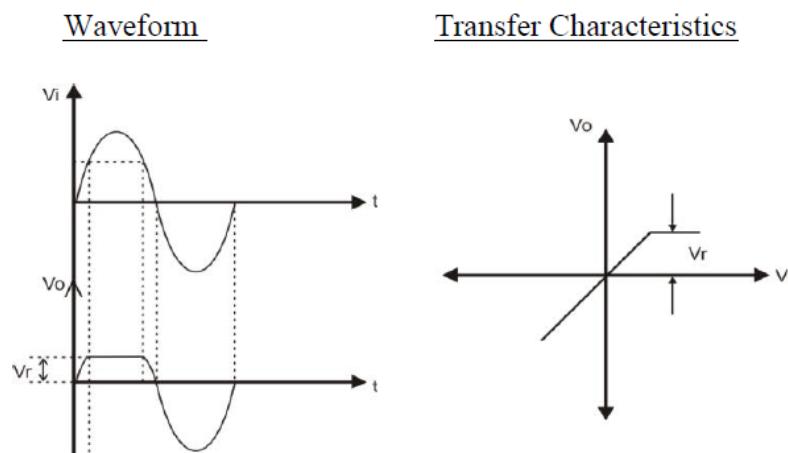
INPUTS		OUTPUT (in volt)
1 (volt)	2 (volt)	
0 (0)	0 (0)	0 (0)
10 (1)	0 (0)	0 (0)
0 (0)	10 (1)	0 (0)
10 (1)	10 (1)	10 (1)

Therefore, Above circuit = AND Logic Gate

Hence, Circuit is Verified Successfully both theoretically & practically.



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



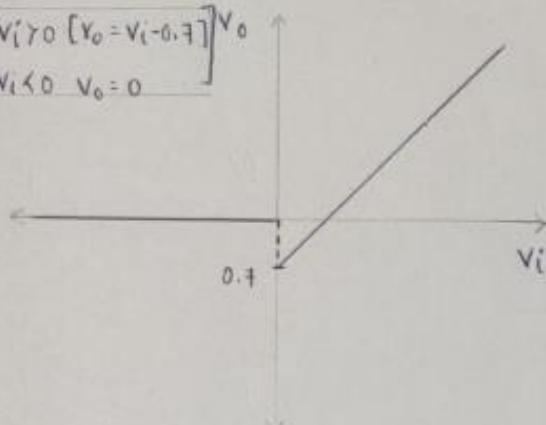


(Q3)

UI9CS012

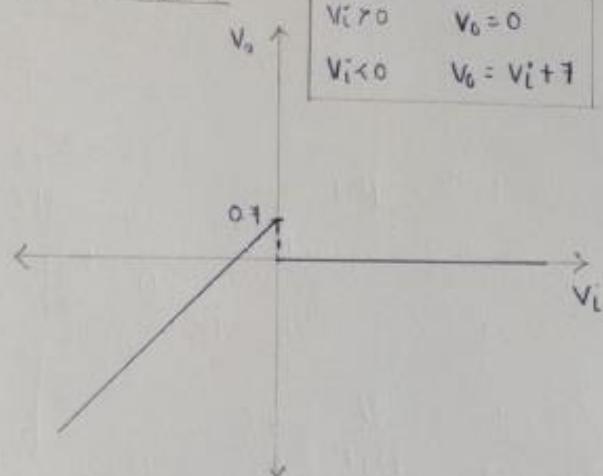
(A) Negative Clipper

$$\begin{cases} V_i > 0 & V_o = V_i - 0.7 \\ V_i < 0 & V_o = 0 \end{cases}$$



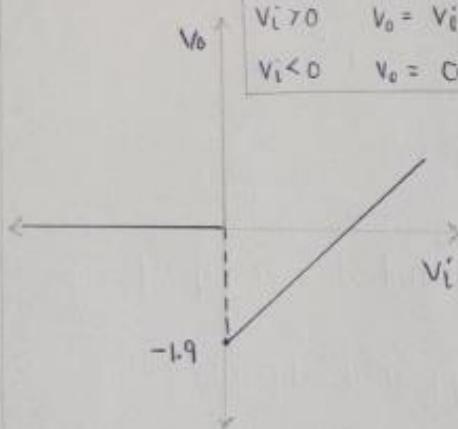
(B) Positive Clipper

$$\begin{cases} V_i > 0 & V_o = 0 \\ V_i < 0 & V_o = V_i + 0.7 \end{cases}$$



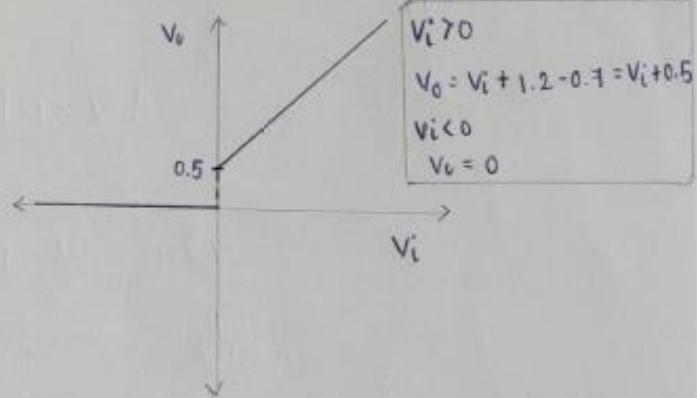
(C) Negative Clipper with Bias-I

$$\begin{cases} V_i > 0 & V_o = V_i - 1.9 \\ V_i < 0 & V_o = 0 \end{cases}$$



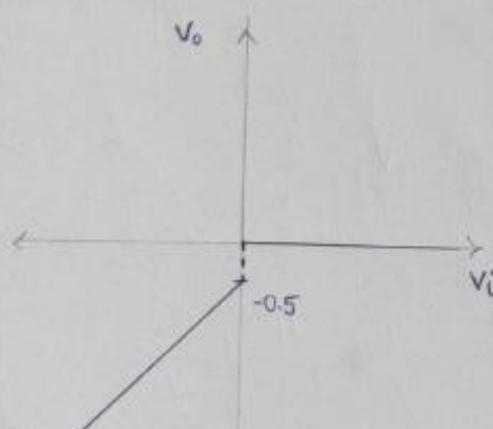
(D) Negative Clipper with Bias-II

$$\begin{cases} V_i > 0 & V_o = V_i + 1.2 - 0.7 = V_i + 0.5 \\ V_i < 0 & V_o = 0 \end{cases}$$



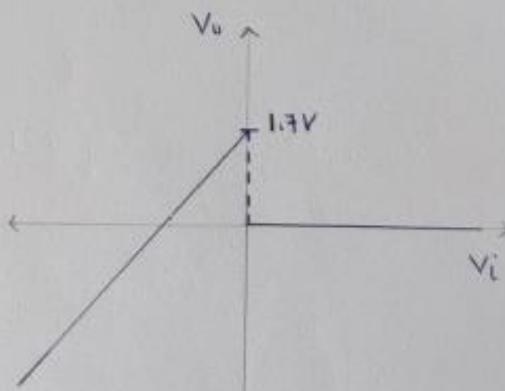
(E) Positive Clipper with Bias-I

$$\begin{cases} V_i > 0 & V_o = 0 \\ V_i < 0 & V_o = V_i + 0.5 \end{cases}$$



(F) Positive Clipper with Bias-II

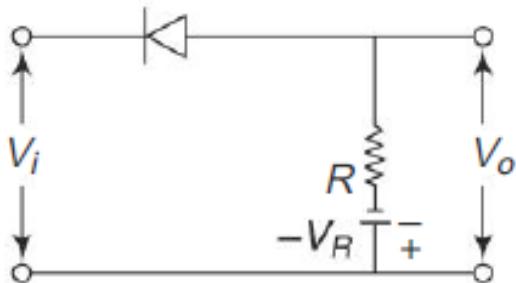
$$\begin{cases} V_i > 0 & V_o = 0 \\ V_i < 0 & V_o = V_i + 1.9 \end{cases}$$





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.)



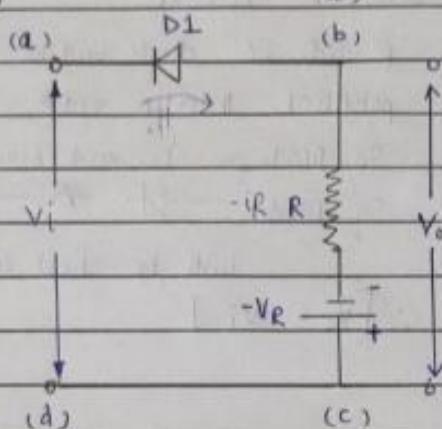
I.) Theoretical Calculations:



(619CS012)

Q4.3

(A)



(x)

(b)

(c)

(d)

$V_i$  = symmetrical sine wave  
[  $V_{peak} > V_{reverse}$  ]

(I) During Positive half cycle  $\frac{V_{peak}}{2}$  volt

∴ Diode D1 will be in

reverse bias [  $V_b = -V_R$  ]

$$V_o = +V_R$$

open circuit

therefore no current will flow

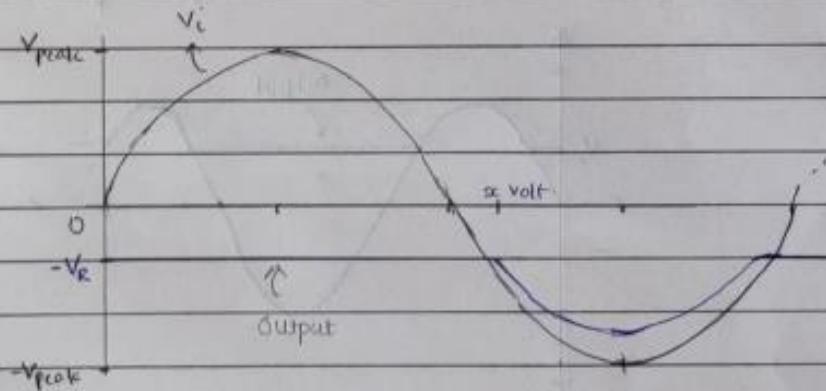
(II) During Negative half cycle  $[ V_o = -V_R + (0 \times R) = -V_R ] \quad \text{---(1)}$

half cycle after  $- \frac{V_{peak}}{2}$  volt

The Diode D1 behave as  $\Rightarrow$  ~~open circuit~~ (short circuit)

$$\therefore [ V_o = V_i ]$$

Predicted Output:



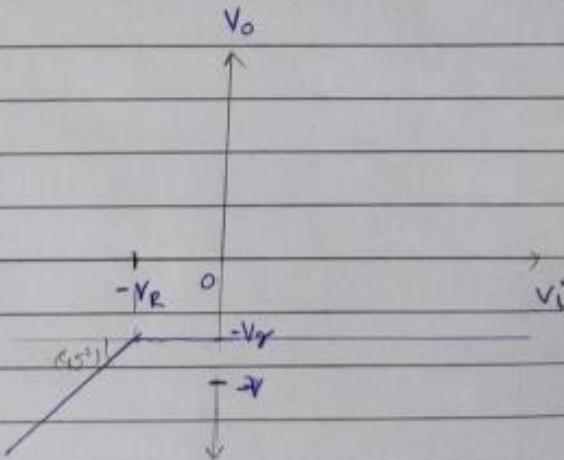
Transfer characteristics

$$\textcircled{1} \quad V_i \geq -V_R \quad V_o = -V_R$$

$$\textcircled{2} \quad V_i < -V_R \quad V_o = -V_R + \delta$$

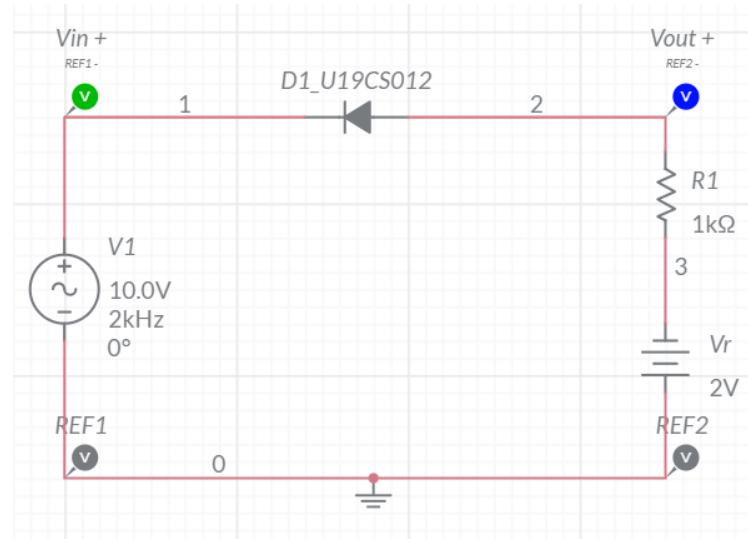
$\delta = (V_R)$

$$[ V_o = V_i ]$$

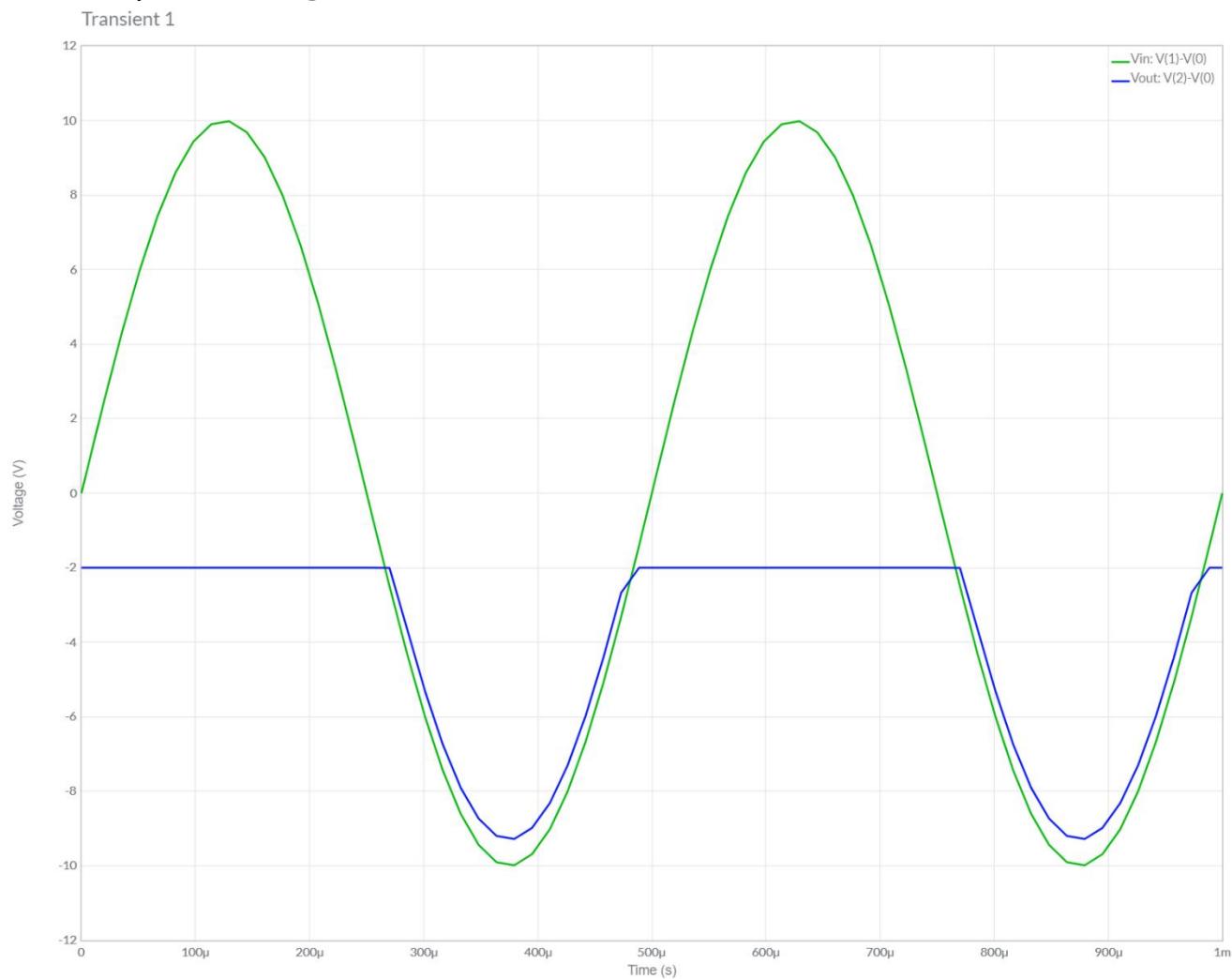




**1.) Circuit Image:**

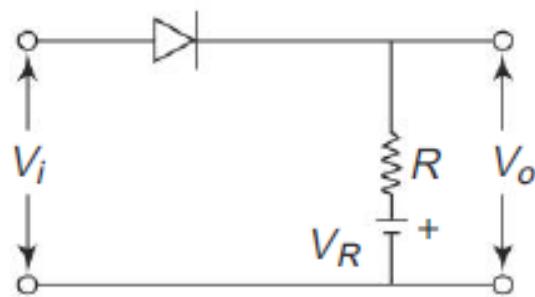


**2.) Grapher Image:**





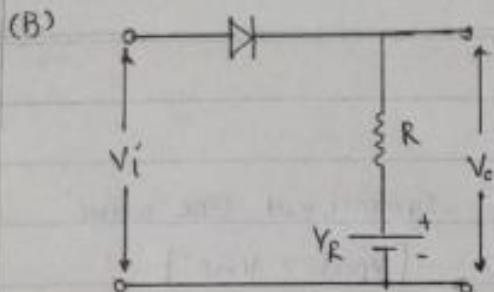
[B.)



I.) Theoretical Calculations:



U19CS012



(I) Case I:  $V_i > V_R$

p side of diode will be at higher potential than n side,

So Diode is Forward biased.

So Diode  $\rightarrow \text{D} \rightarrow \text{---}$

will be short circuited

$$\therefore [V_o = V_i]$$

(II) Case II:  $V_i < V_R$

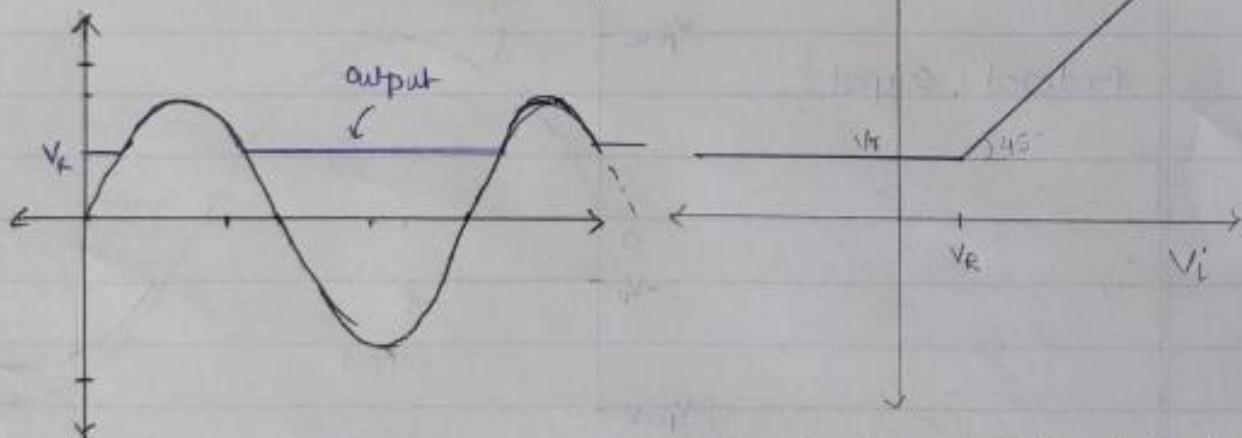
p side of Diode will be at low potential than n side

so Diode is Reverse biased

$\rightarrow \text{D} \rightarrow \text{---} \rightarrow \text{---}$

$\therefore$  Diode is Open circuit

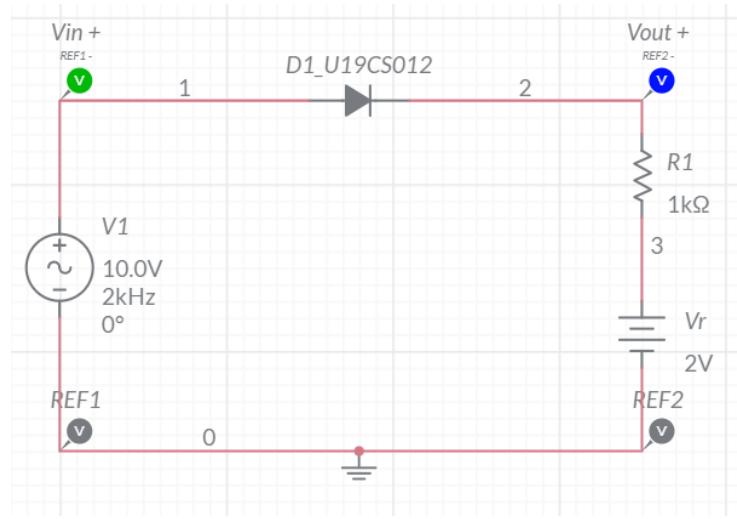
$$\therefore [V_o = V_R]$$



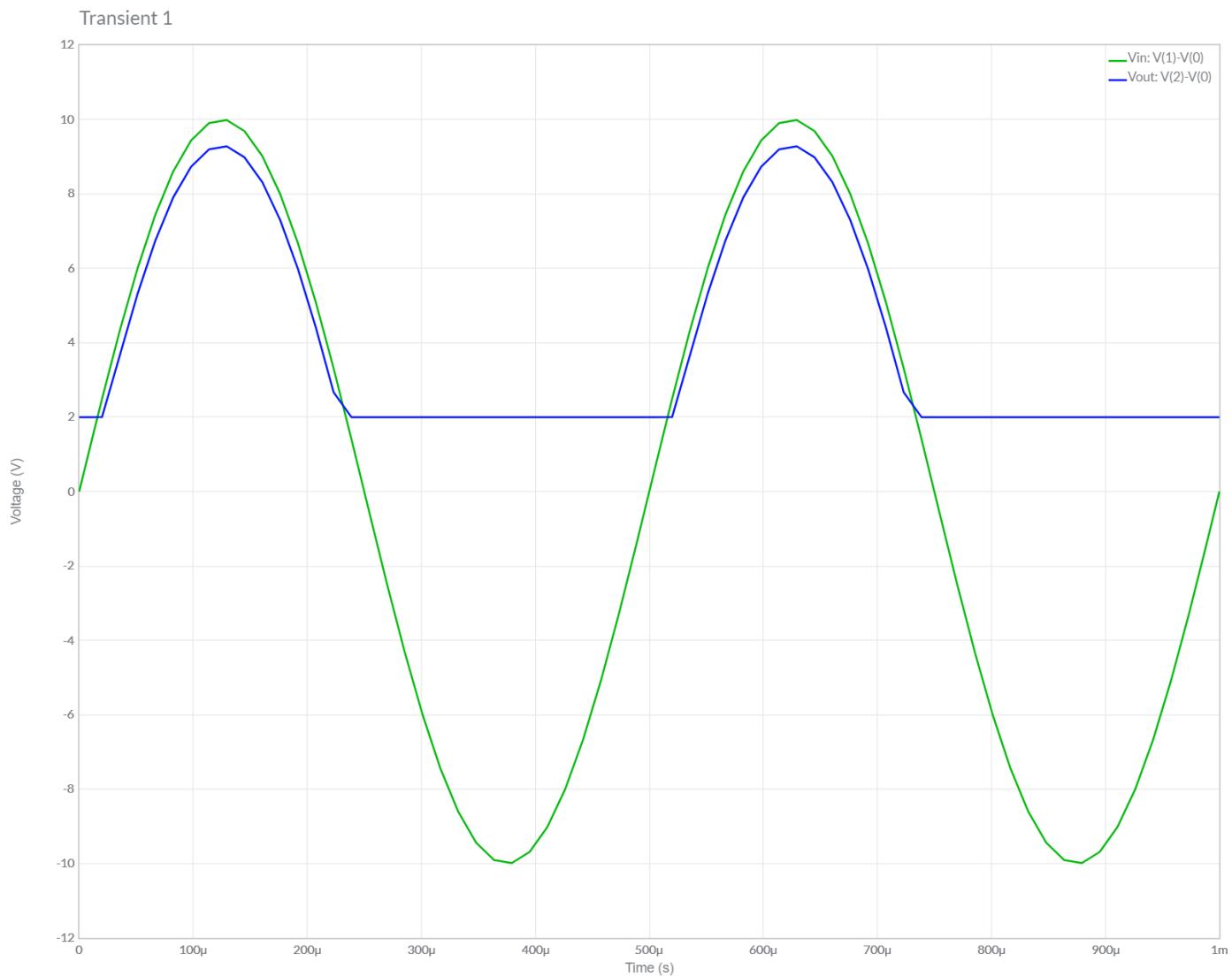
Transfer characteristics of circuit



**1.) Circuit Image:**



**2.) Grapher Image:**





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:**

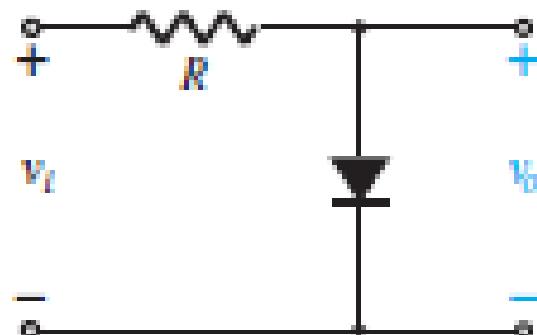
1. Multisim Simulator/Circuit Simulator

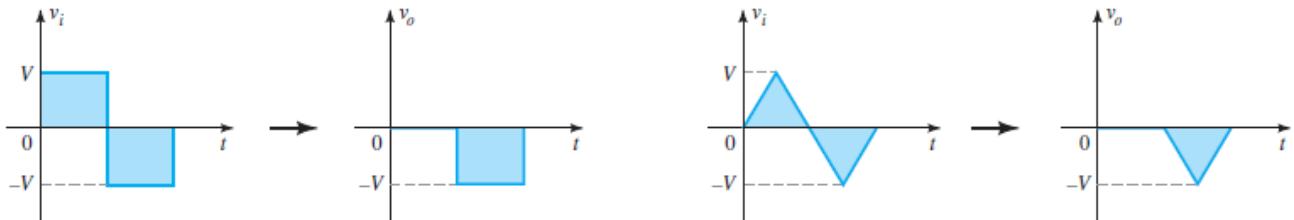
**THEORY:**

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**

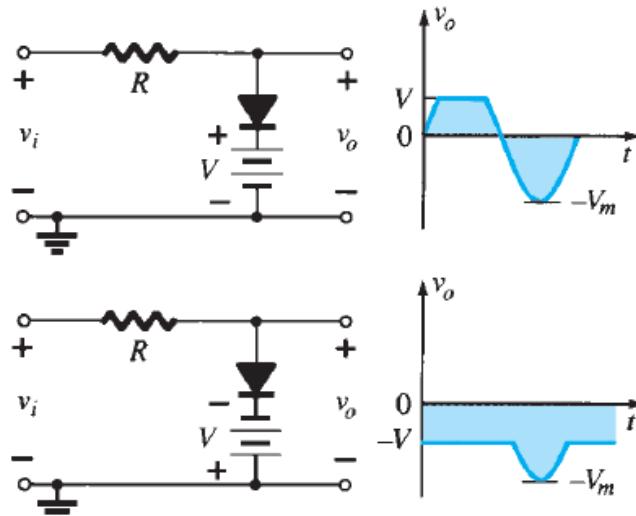


As shown above, when the positive half cycle appears, the diode being forward biased, acts as short circuit and thus the output voltage remains at zero level. During the negative half cycle, the diode is reverse biased, acts as open circuit and hence we see that the output node comes into direct contact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

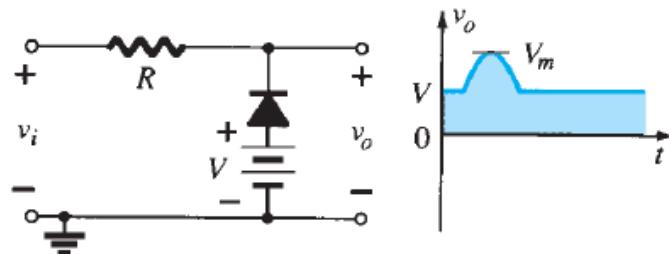
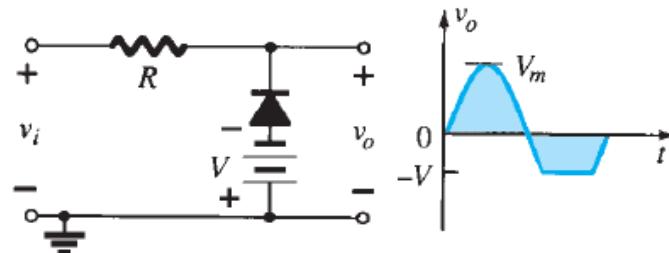
### FEW SHUNT DIODE CLIPPER CONFIGURATIONS

#### POSITIVE



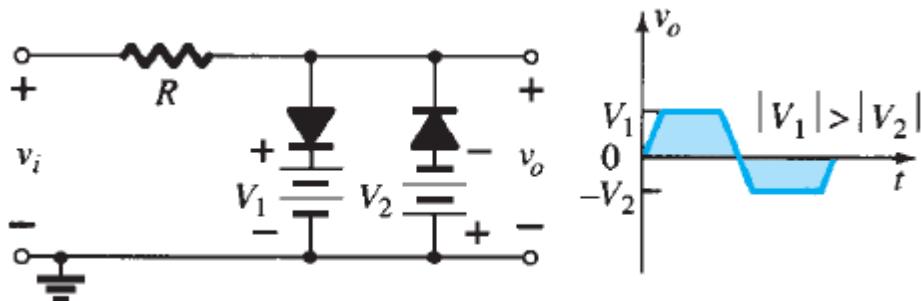


### NEGATIVE



### TWO LEVEL CLIPPERS

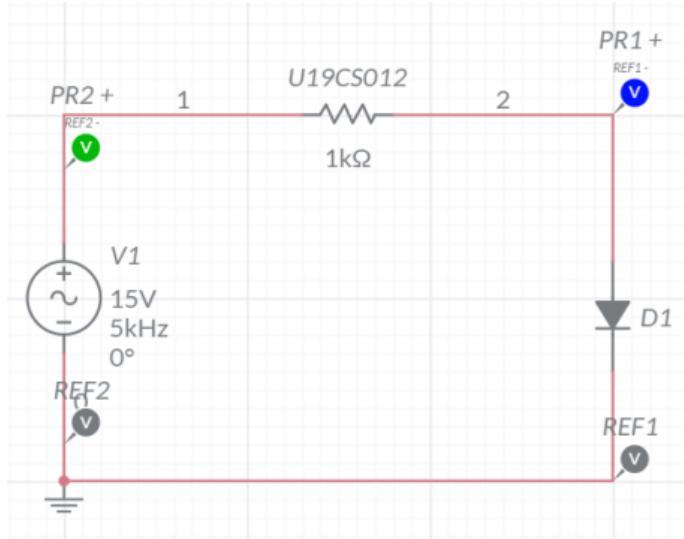
These circuits employ clipping in both the directions (Positive as well as Negative Half Cycles) as shown in figure below:



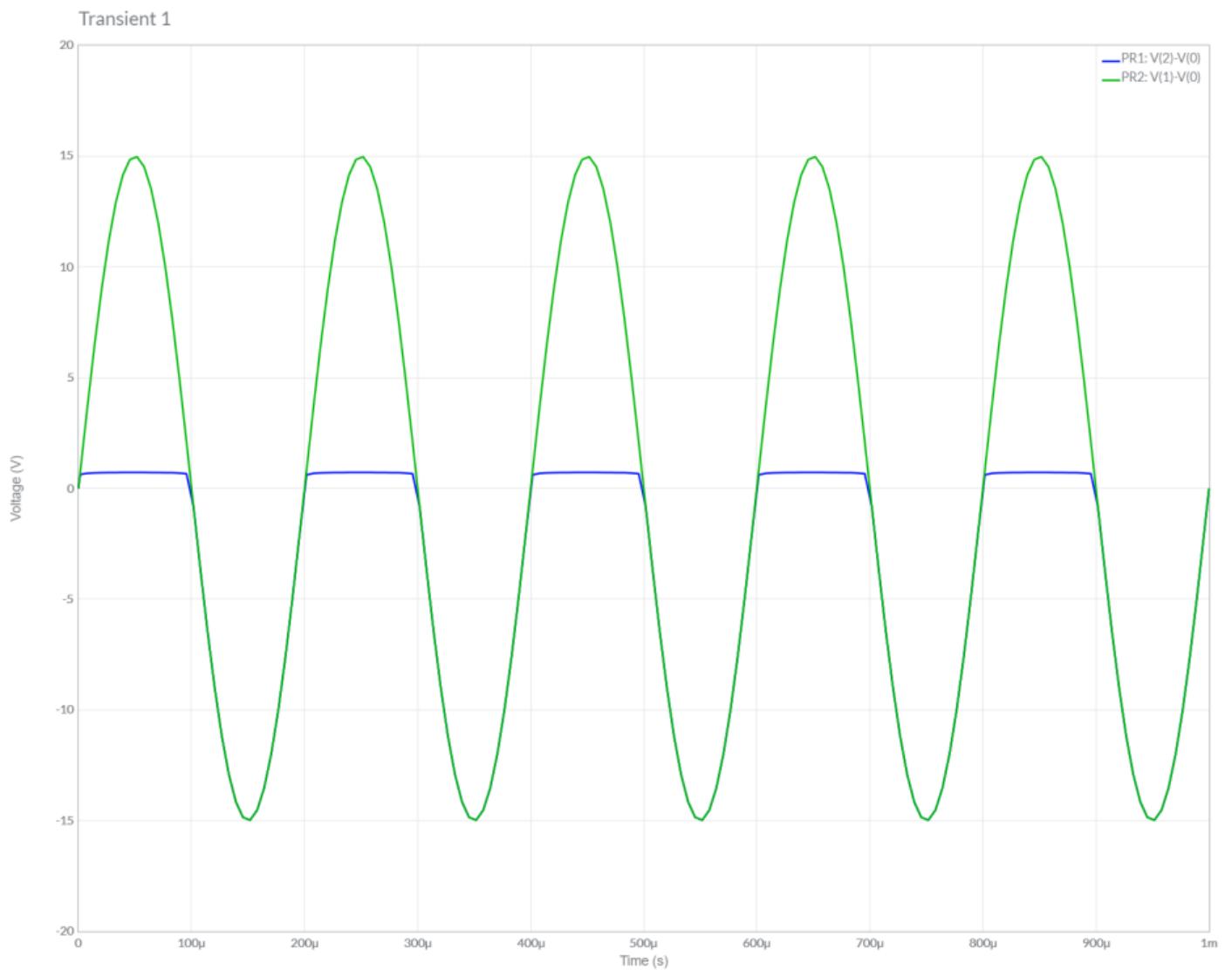


### 1.) SHUNT POSITIVE CLIPPER

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



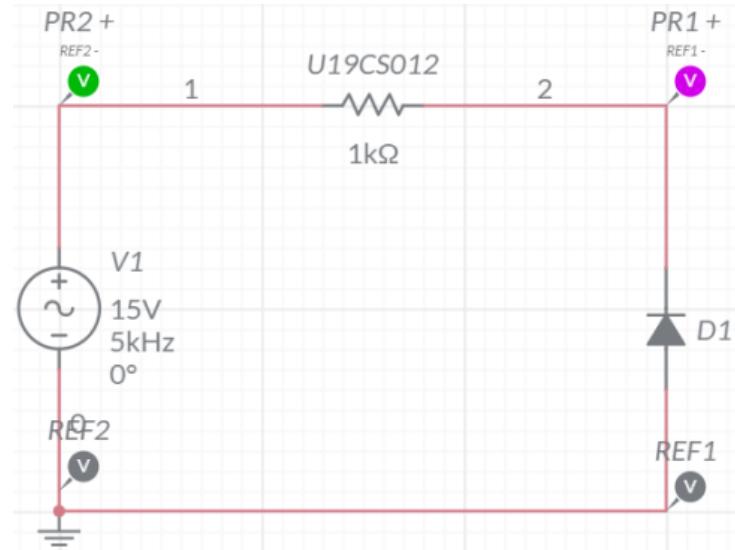
#### WAVEFORMS (FROM MULTISIM)



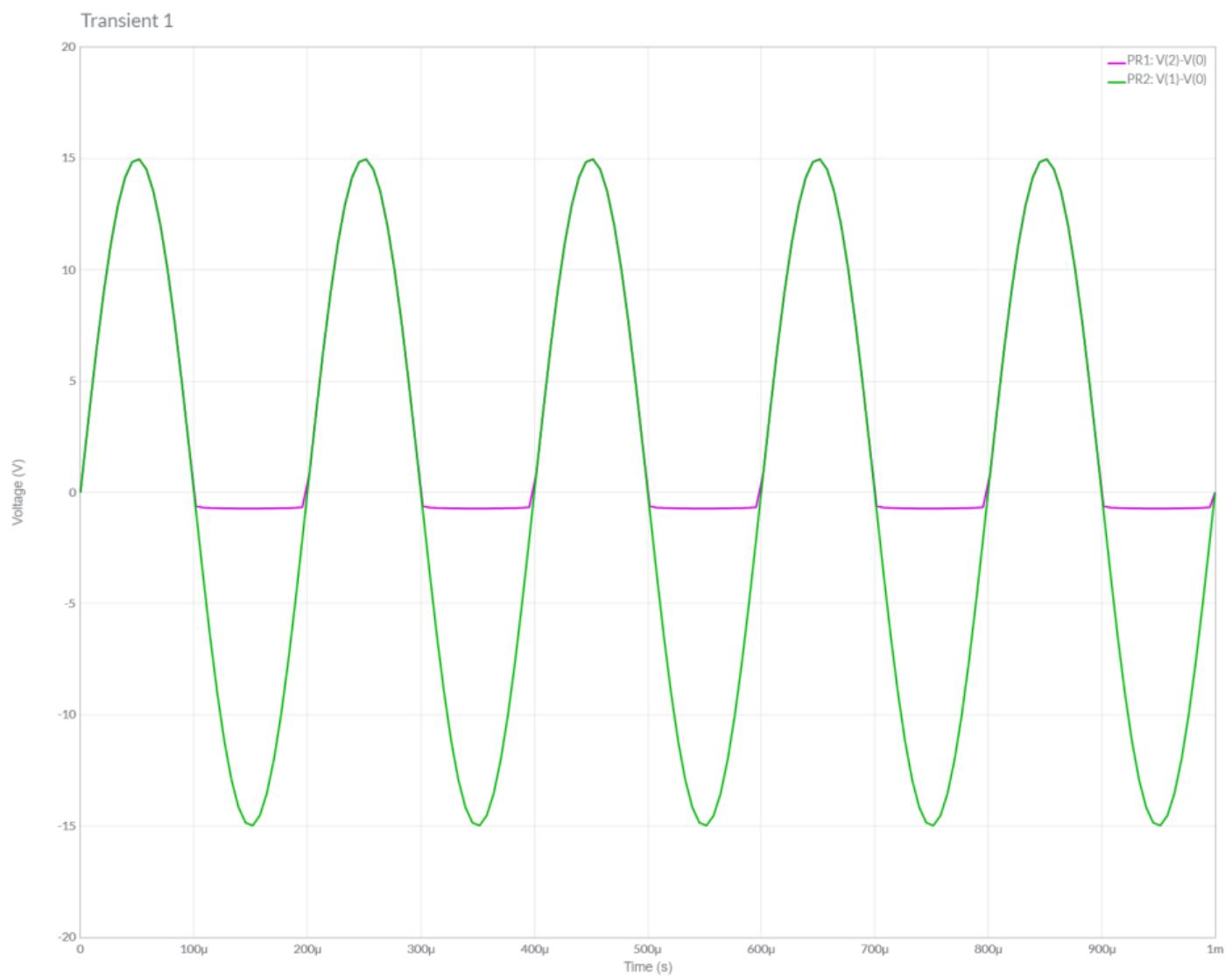


## 2.) SHUNT NEGATIVE CLIPPER

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



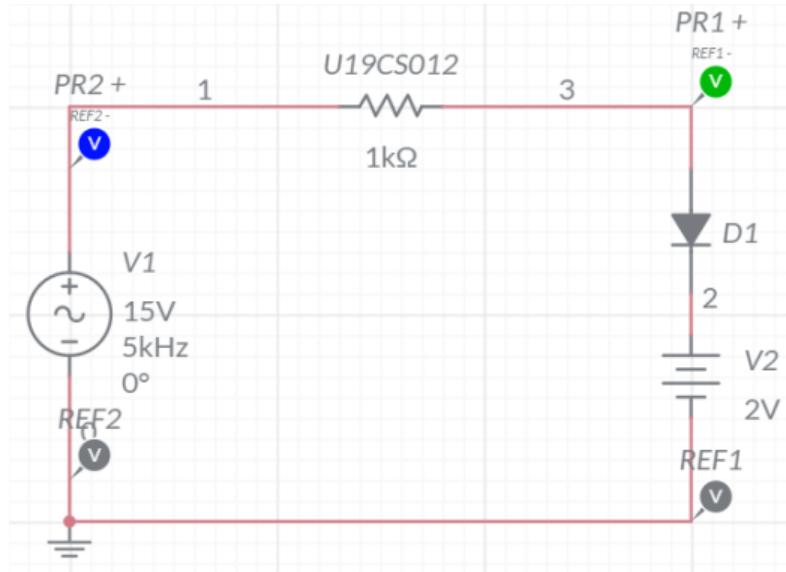
**WAVEFORMS (FROM MULTISIM)**



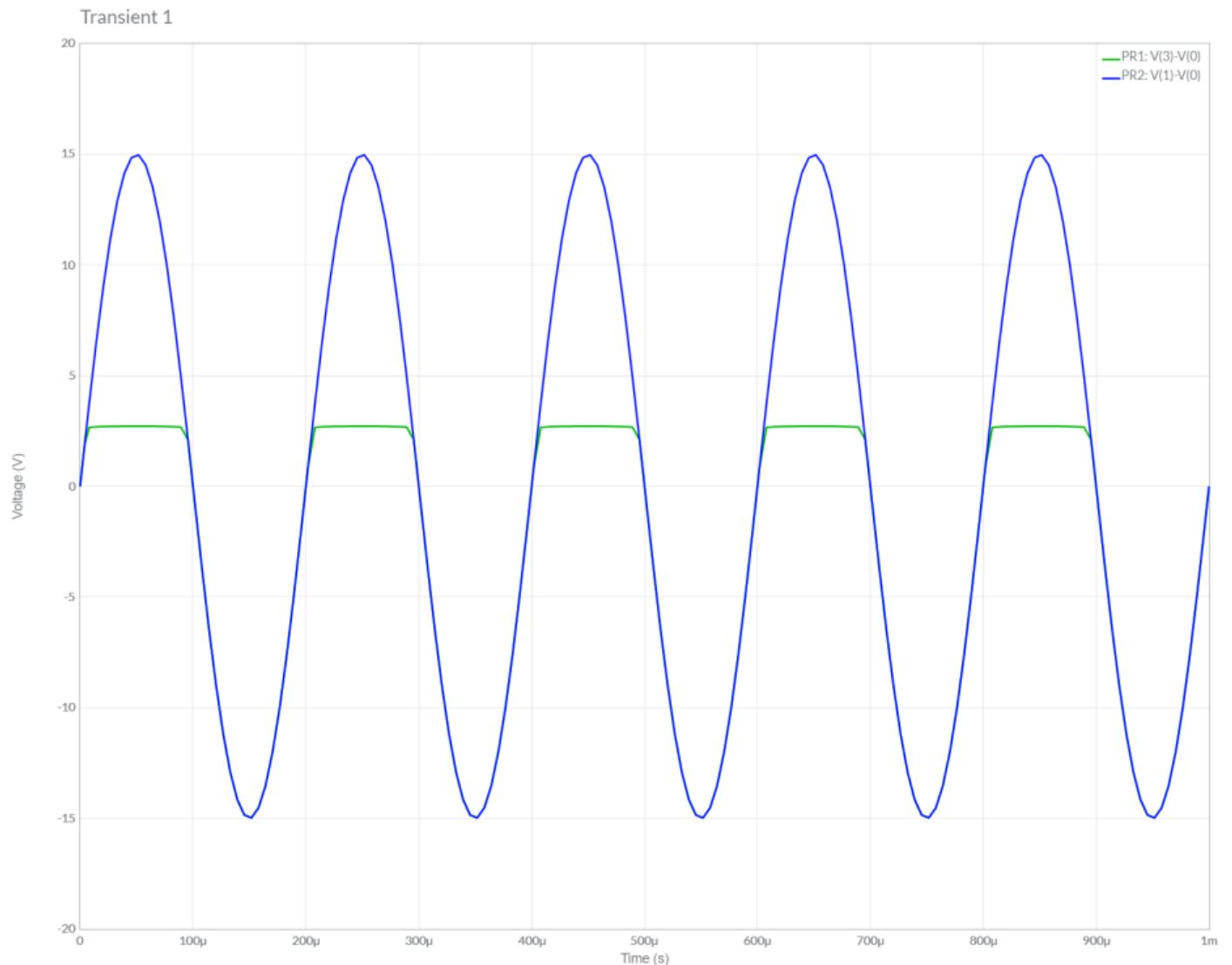


### 3.) SHUNT POSITIVE CLIPPER WITH BIAS-I

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



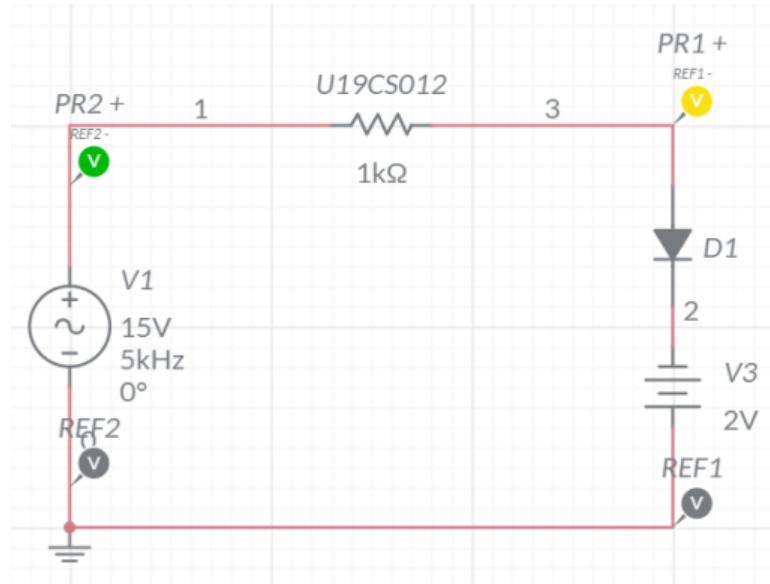
**WAVEFORMS (FROM MULTISIM)**



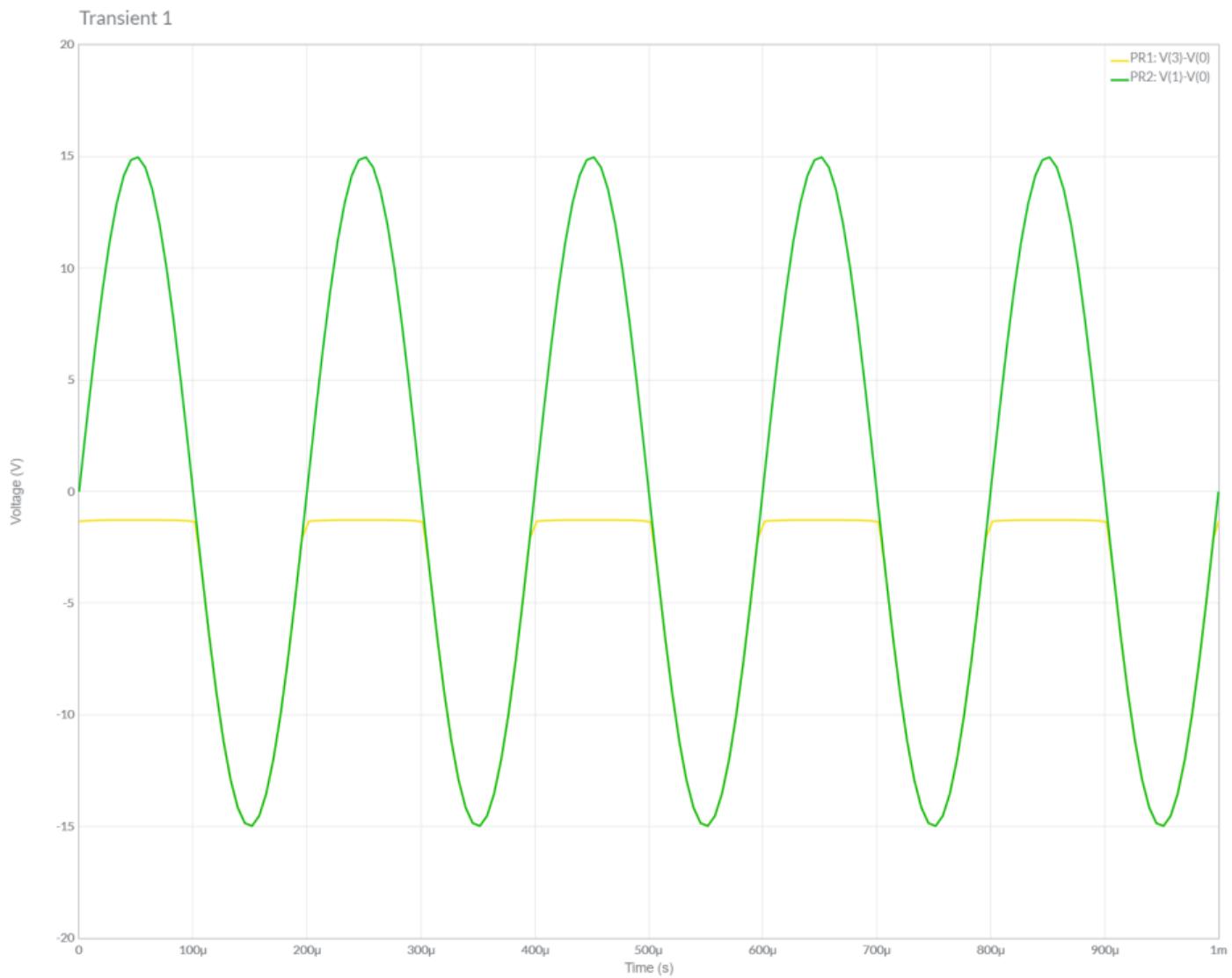


#### 4.) SHUNT POSITIVE CLIPPER WITH BIAS-II

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



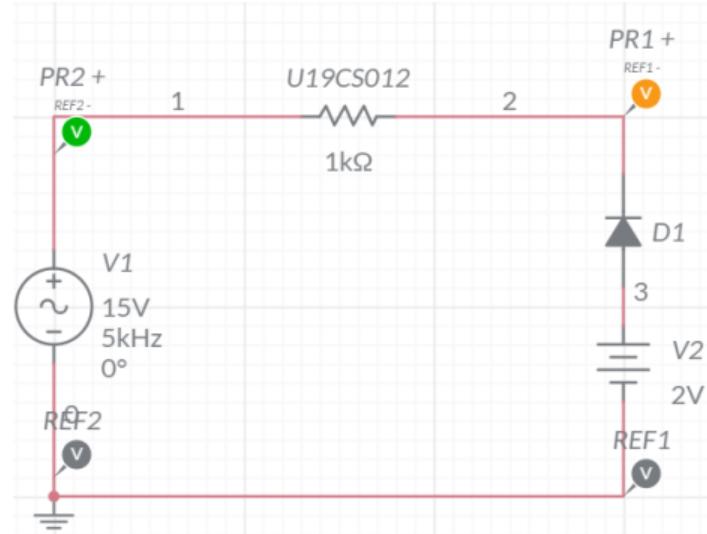
**WAVEFORMS (FROM MULTISIM)**



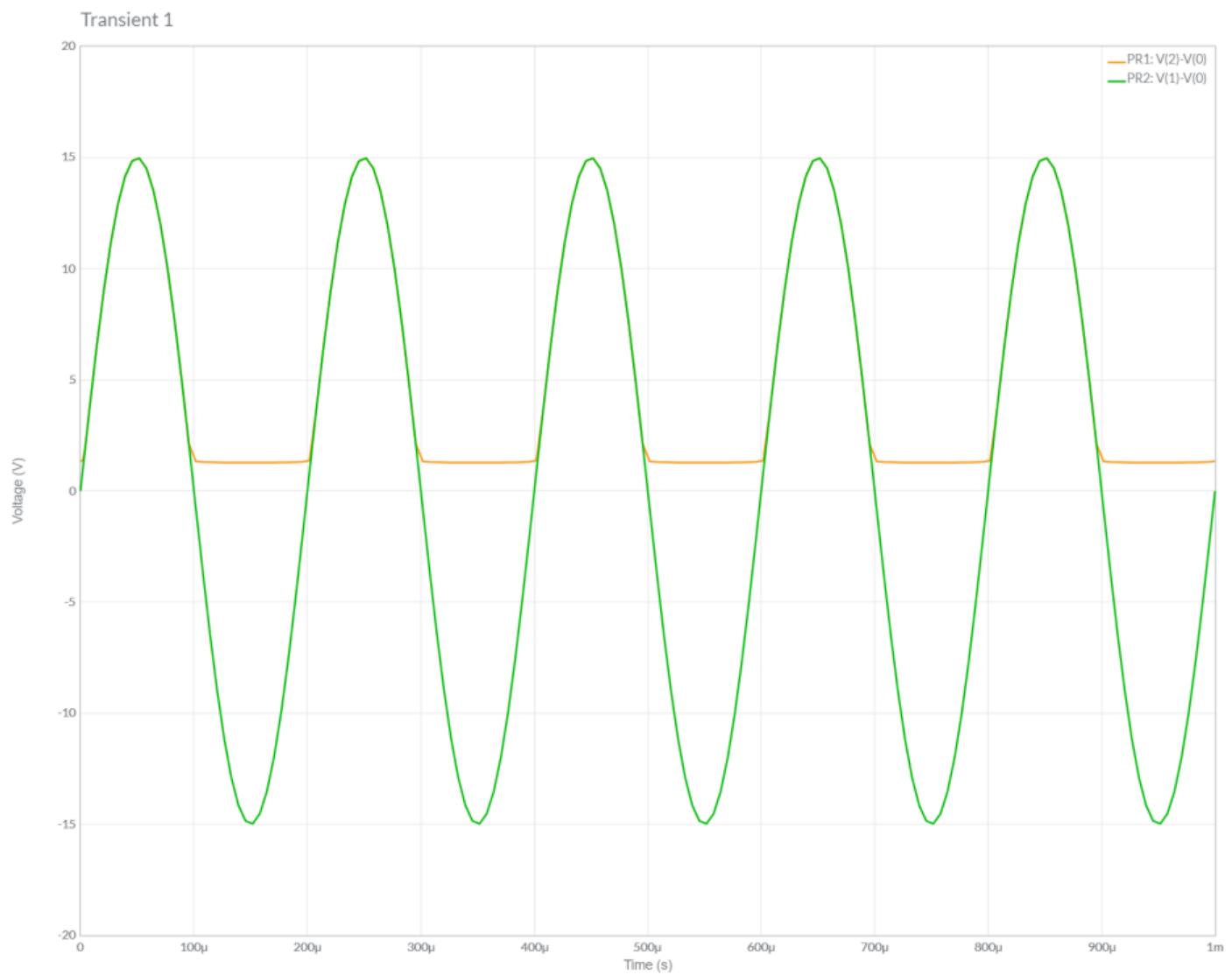


### 5.) SHUNT NEGATIVE CLIPPER WITH BIAS-I

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



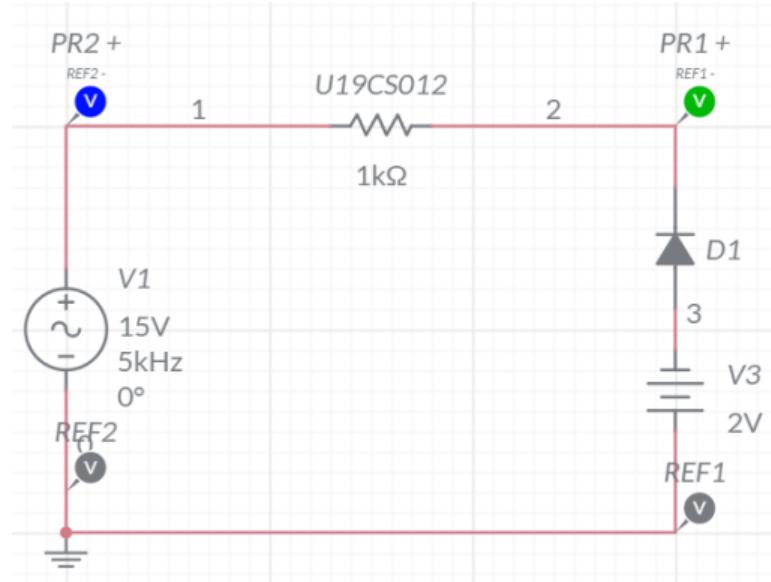
#### WAVEFORMS (FROM MULTISIM)



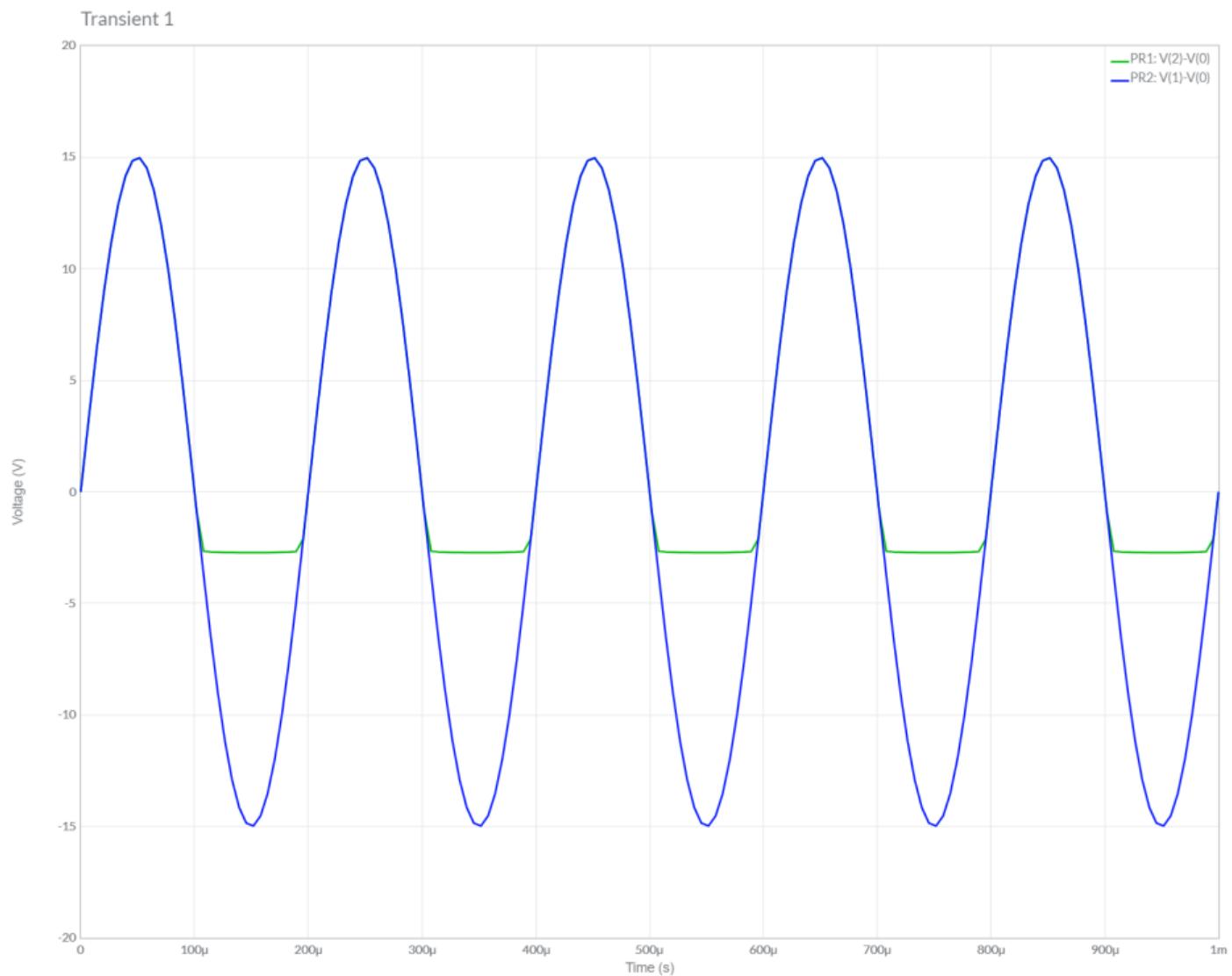


### 6.) SHUNT NEGATIVE CLIPPER WITH BIAS-II

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



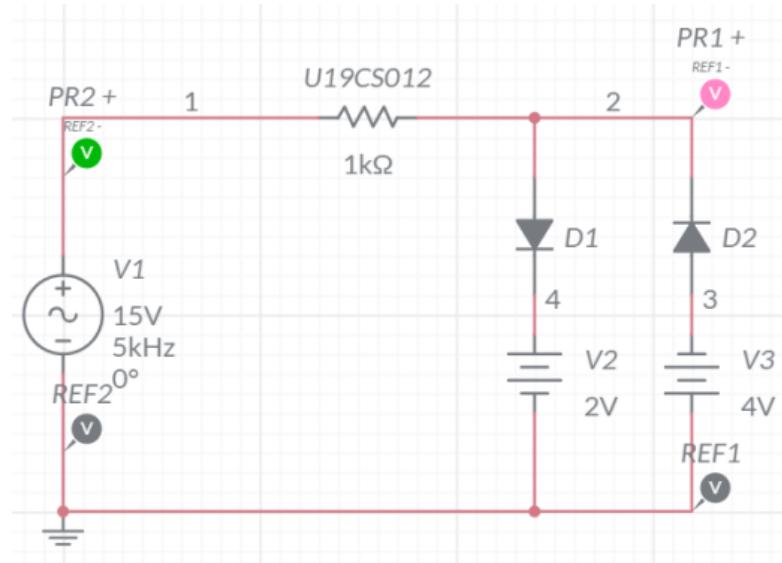
**WAVEFORMS (FROM MULTISIM)**



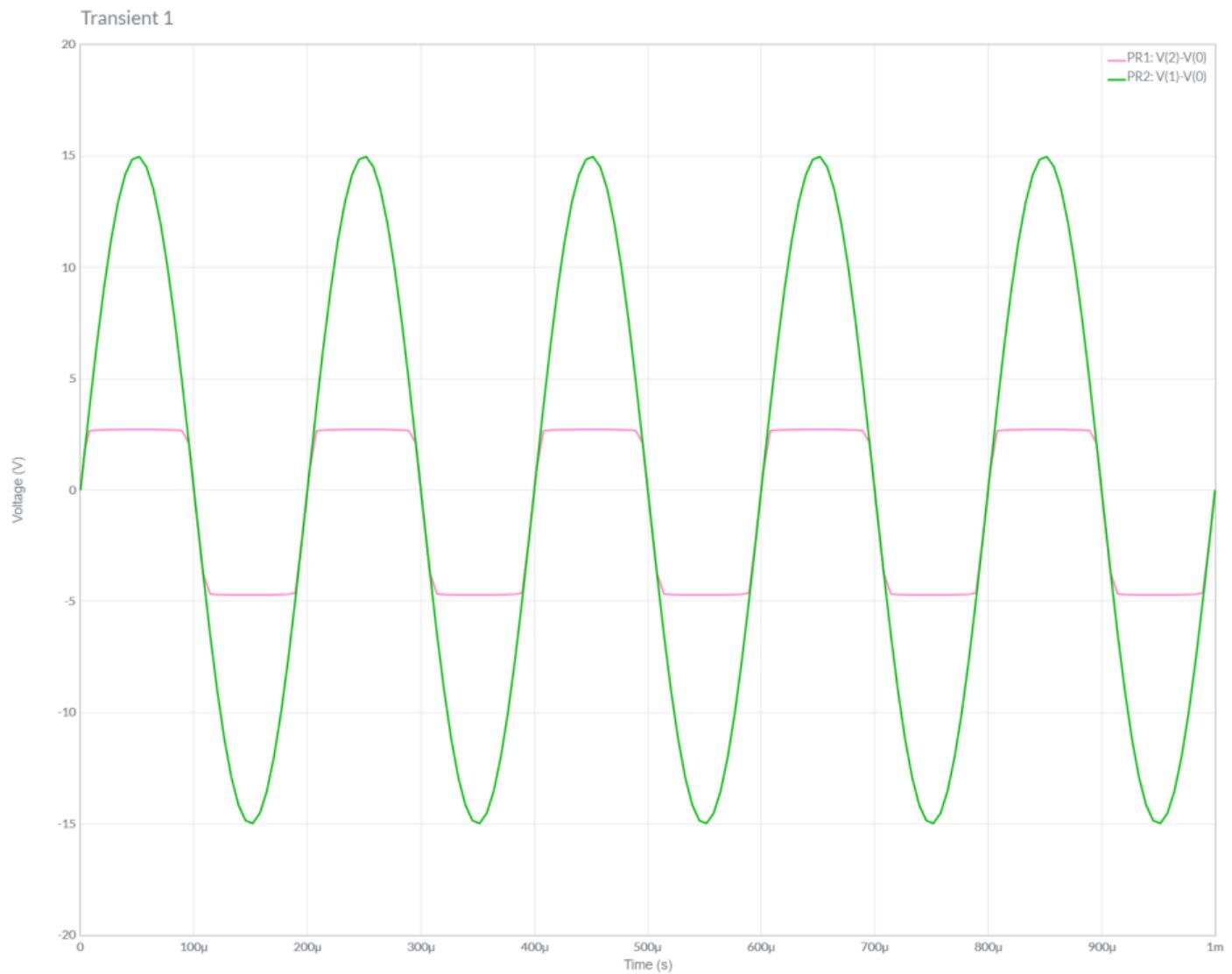


## 7.) DUAL CLIPPER

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



### WAVEFORMS (FROM MULTISIM)





## **CONCLUSIONS**

- 1.) In this Experiment, We have studied about Shunt Clipper Circuits [Both Positive and Negative] along with Different Biasing Applied.
- 2.) We Verified the Theoretical Knowledge of Shunt Clippers by Performing Simulations of 7 Cases of Shunt Clippers in Multisim.
- 3.) Hence, we have Successfully Designed, Plotted and Verified Various Shunt Diode Clipper Circuits.

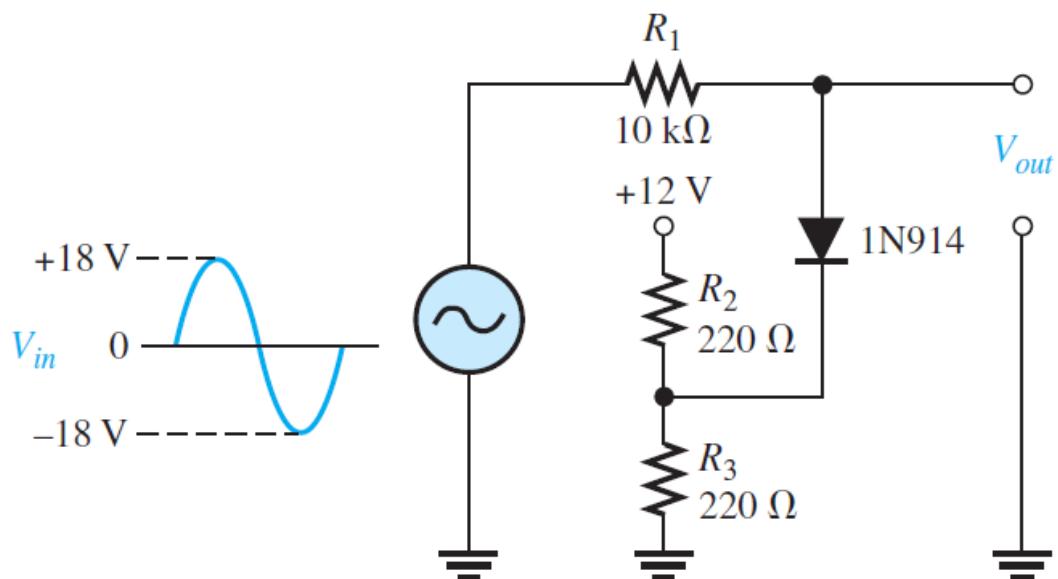


## **ASSIGNMENT - 7**

**U19CS012**

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

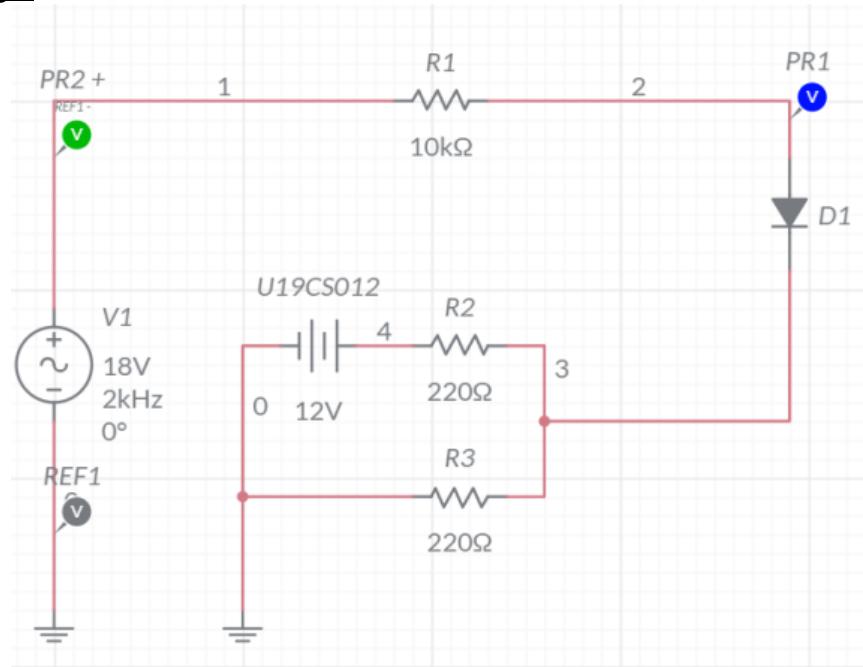
**Circuit a.)**



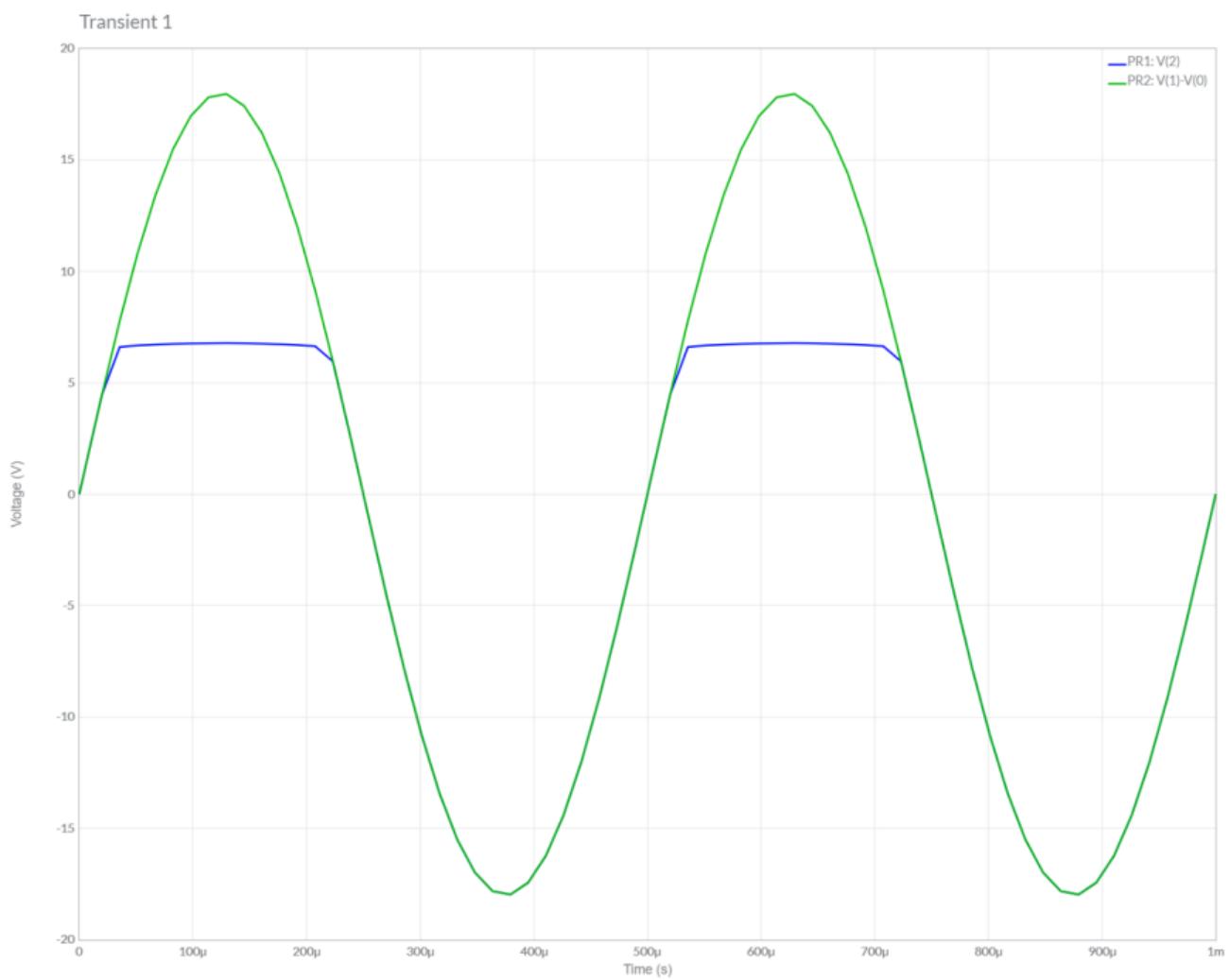


**A.) Multisim Calculations:**

**1.) Circuit Image:**



**2.) Grapher Image:**





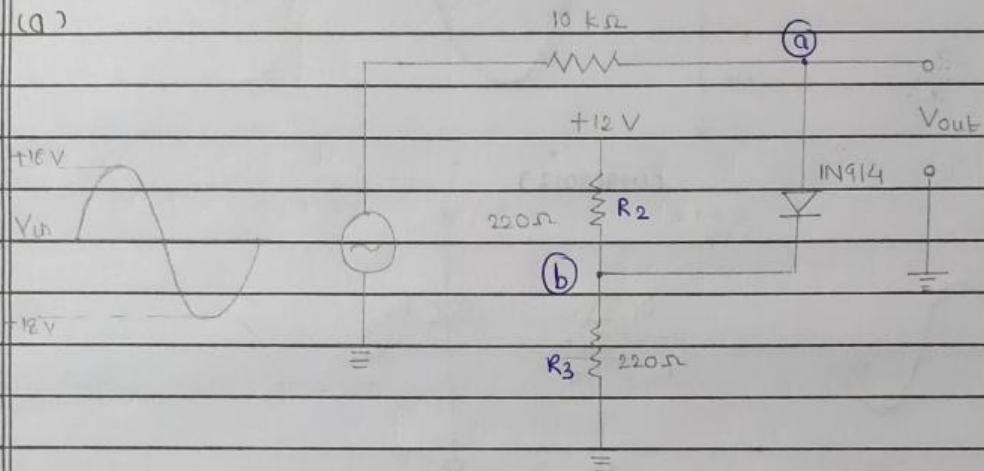
**B.) Theoretical Calculations:**

Practical - 7 Assignment  
(UI9CS012)

(1)

Q1) Determine and Plot Output Voltage waveform

(a)



→ Current through 12V battery and passing through  $R_2$  and  $R_3$ .

$$I = \frac{12}{R_2 + R_3} = \frac{12}{220 + 220} = \frac{12}{440}$$

$$\text{Voltage at } b, V_b = I \times R_3 = \frac{12}{440} \times 220 = 6V$$

∴ Voltage at Cathode of Diode = 6V

∴  $V_i(t) < (6 + 0.7) V$  diode will be reverse biased,

∴ Diode  $\xrightarrow{\text{act}} \text{Open circuit}$

$$\therefore V_o = V_i(t) \quad \text{for } [V_i(t) < (6.7) V] \quad (1)$$

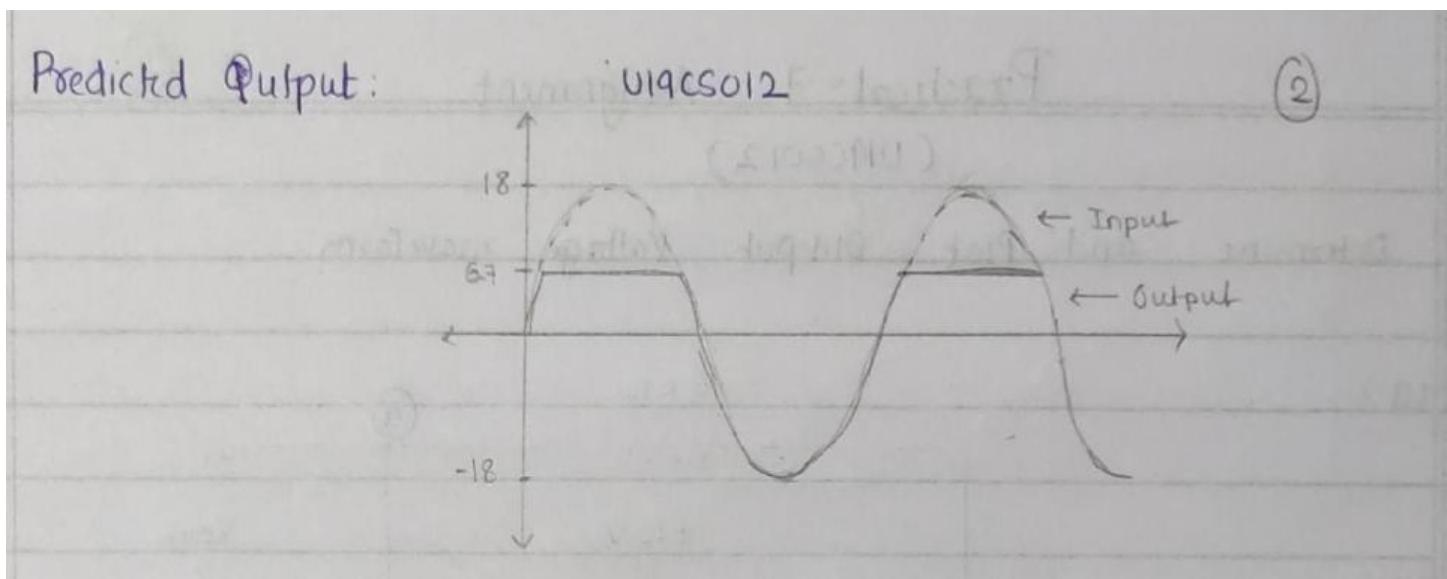
→ For  $V_i(t) > 6.7V$ , diode will be forward biased and will act as

Short circuit for voltage drop of 0.7V

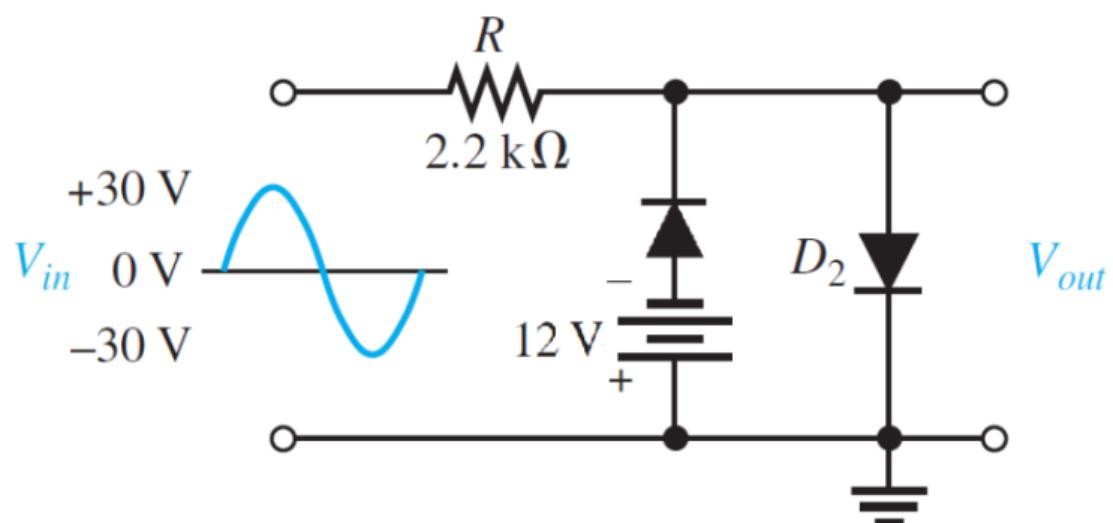
$$\therefore V_o = V_b + 0.7 = (6 + 0.7) V$$

$$\therefore V_o = 6.7V$$

$$\therefore V_o = 6.7V \quad \text{for } [V_i > 6.7V] \quad (2)$$



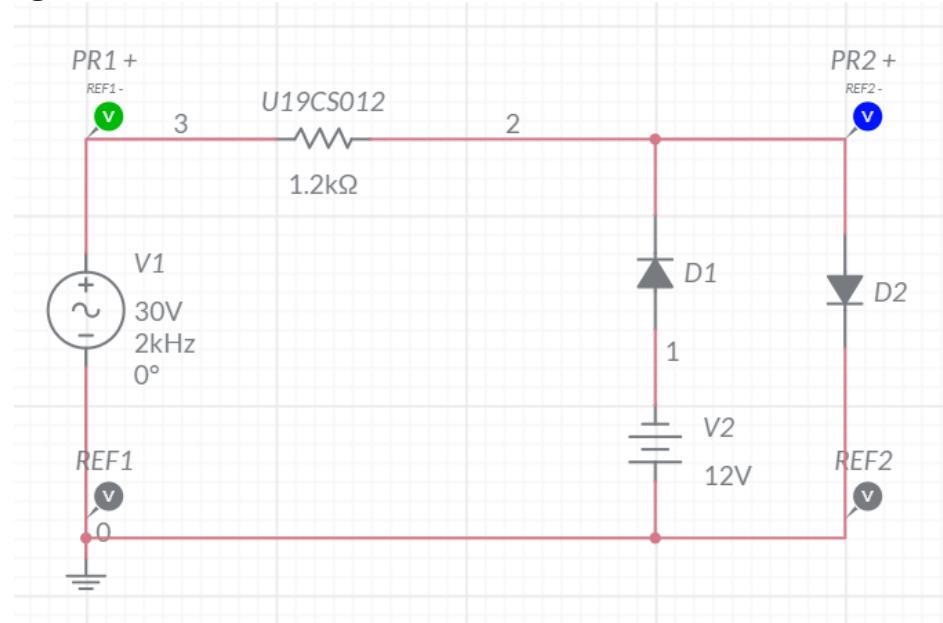
Circuit b.)



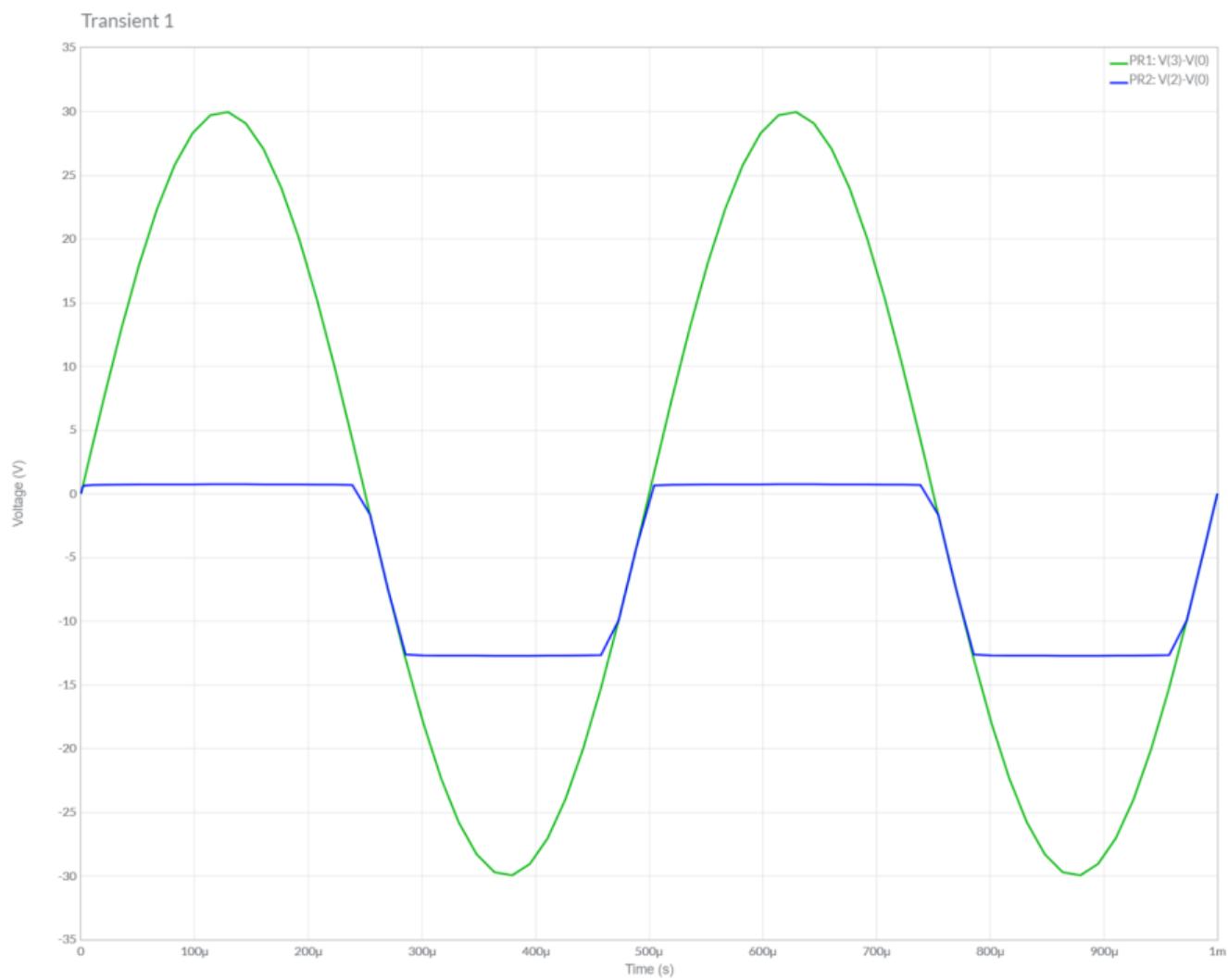


**A.) Multisim Calculations:**

**1.) Circuit Image:**



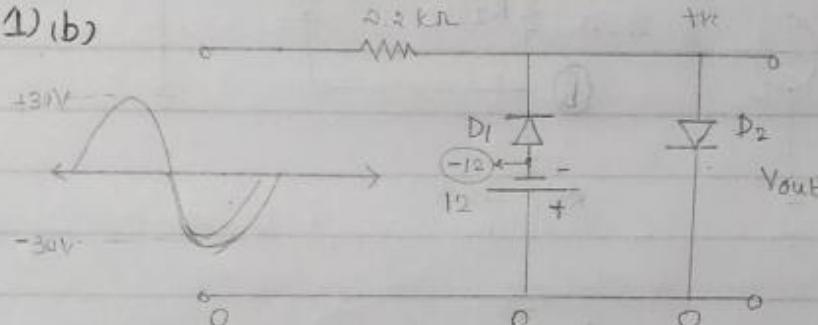
**2.) Grapher Image:**





### B.) Theoretical Calculations:

(CU19CS012)

(1)(b) 

**+ve Half cycle**

$D_1 \rightarrow$  Reverse bias  
 $D_1 \rightarrow -\infty$  (open circuit)

$D_2 \rightarrow$  Forward bias  
 $D_2 \rightarrow 0$  (short circuit)

$[V_o = 0.7V] [30 > V_{in} > 0]$  —①

$\frac{1}{R} \Leftrightarrow 0.7V$  

**-ve Half cycle**

$-12.7 < V_{in} < 0$

$-30 < V_{in} < -12.7$

$D_2 \rightarrow$  Always Reverse bias (-ve cycle)  
 $\therefore D_2 \rightarrow -\infty$  (open circuit)

**(I)  $-12.7 < V_{in} < 0$**

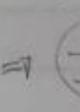
$D_1 \rightarrow$  Reverse bias  
 $D_1 \rightarrow -\infty$  (open circuit)

$[V_o = V_{in}]$  —②

**(II)  $-30 < V_{in} < -12.7V$**

$D_1 \rightarrow$  Forward bias  
 $D_1 \rightarrow 0$  (short circuit)

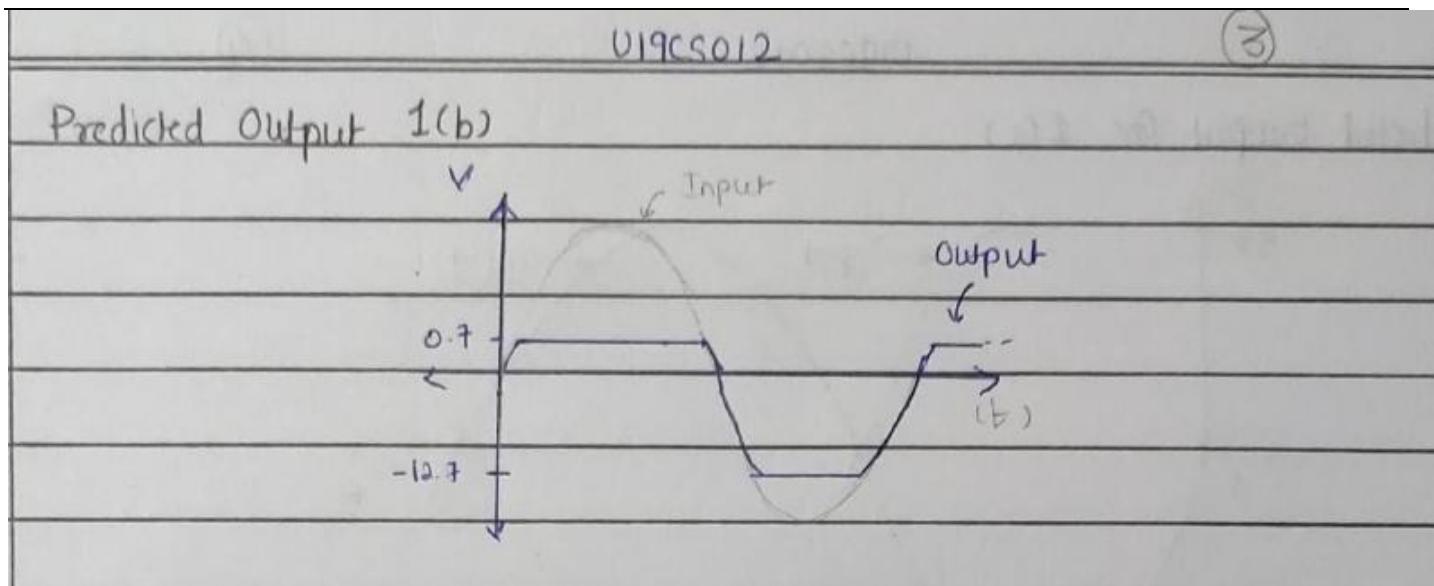
$[V_o = -12.7V] (-30 < V_{in} < -12.7)$  —③

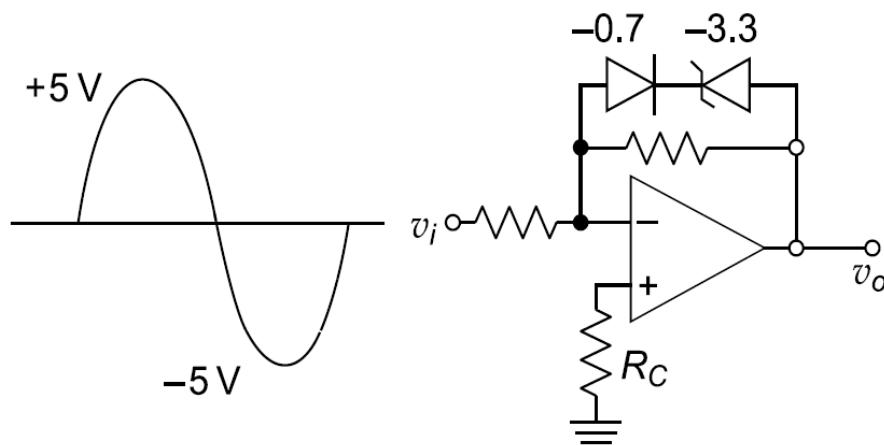
**PREDICTION**

$V_o = \begin{cases} 0.7V & [0 \leq V_{in} < 30] \\ V_{in} & [-12.7 \leq V_{in} < 0] \\ -12.7V & [-30 \leq V_{in} < -12.7] \end{cases}$
---

\*  $0 < V_{in} < 0.7$   $[V_o = V_{in}]$



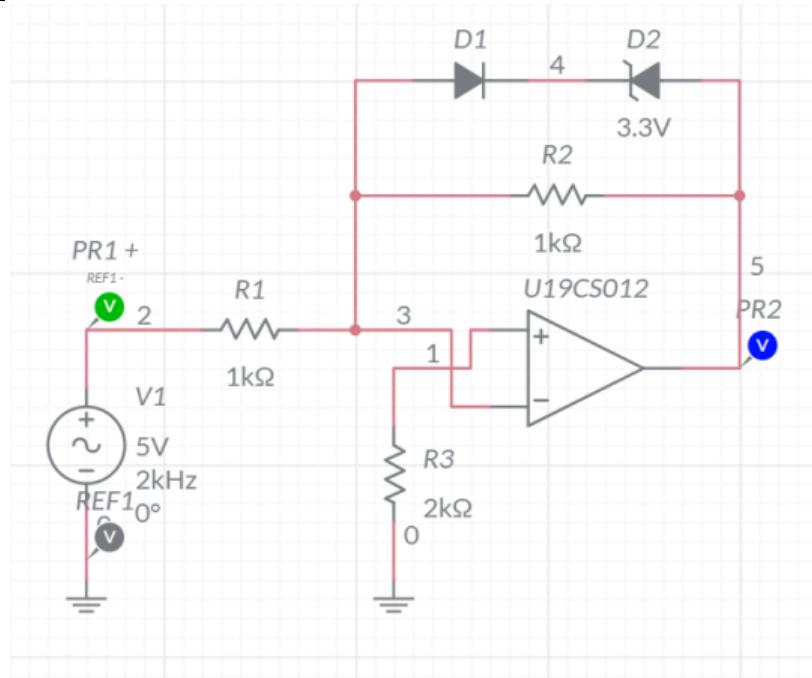
Circuit c.)



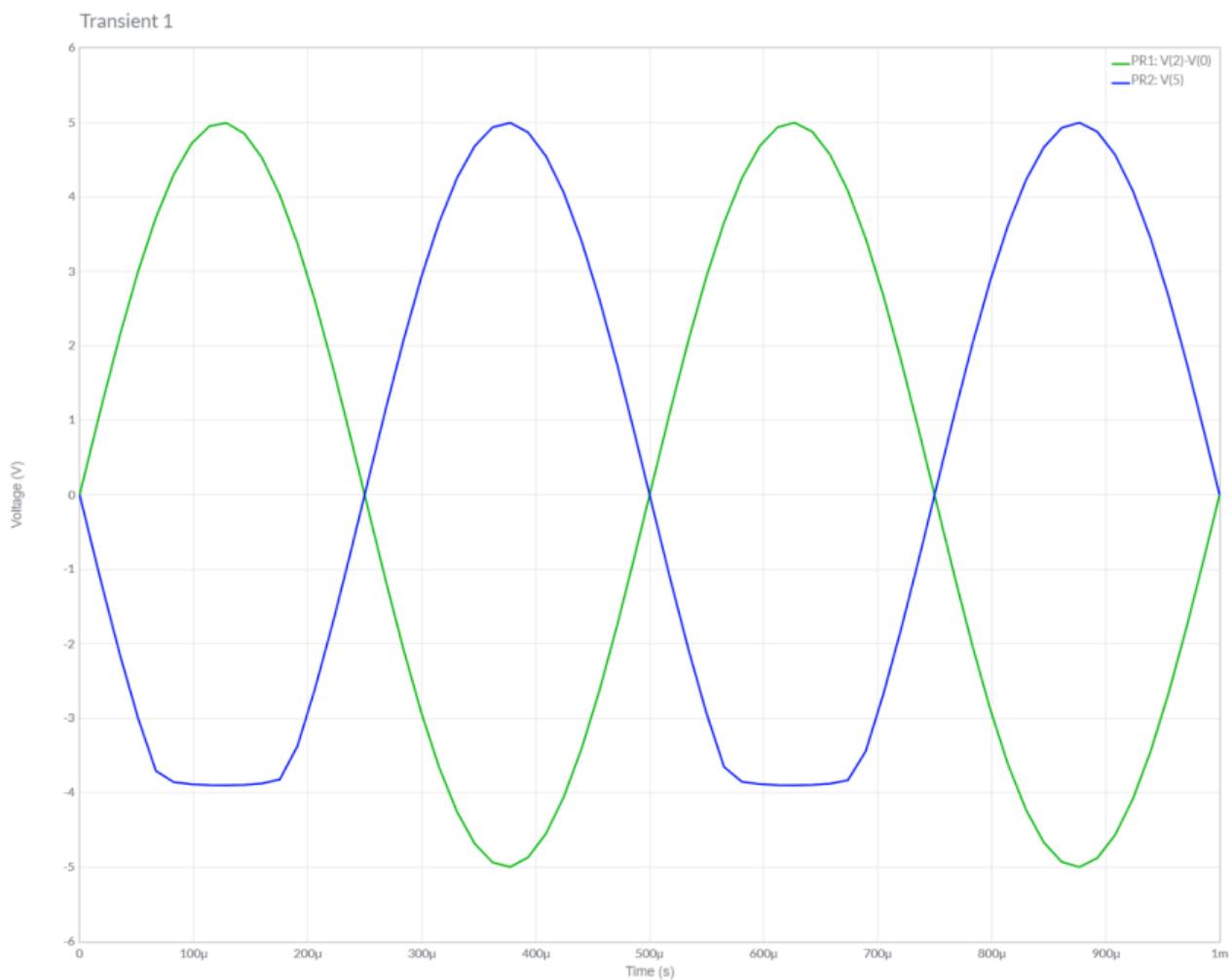


**A.) Multisim Calculations:**

**1.) Circuit Image:**



**2.) Grapher Image:**



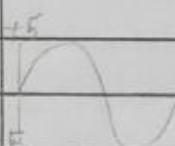


**B.) Theoretical Calculations:**

(1)(c)

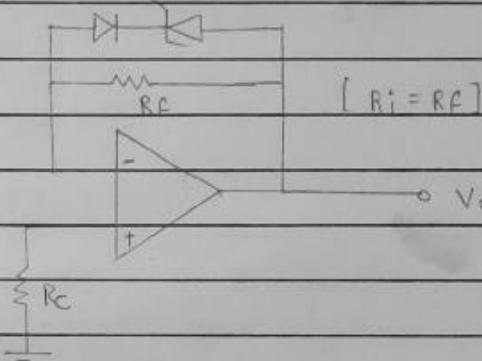
U19CS012

-0.7V -3.3V



$(V_i)$

$R_i$



$[R_i = R_f]$

$V_o$

$\leq R_c$

For  $4V > V_i > 0V$ , Zener diode will be reverse biased and no current will flow through diode [less than breakdown voltage]

$$\therefore V_o = -\frac{R_f}{R_i} V_i = -V_i \quad (\because R_f = R_i) \quad [4V > V_i > 0V] \quad -①$$

For  $V_i > 4V$ , Zener diode will reach breakdown & pn diode will be forward biased. So voltage drop will be  $-0.7 - 3.3 = -4V$

$$\text{so, } V_o = -4V \quad [V_i > 4V] \quad -②$$

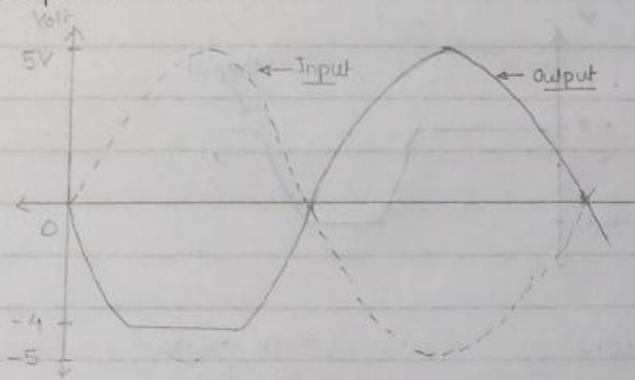
For  $V_i < 0V$  pn diode will be reverse biased, so  $\rightarrow \circ$  open circuit. No current will flow through diode.

$$[V_o = -V_i] \quad [V_i < 0]$$

U19CS012

④

Predicted output for 1(c)





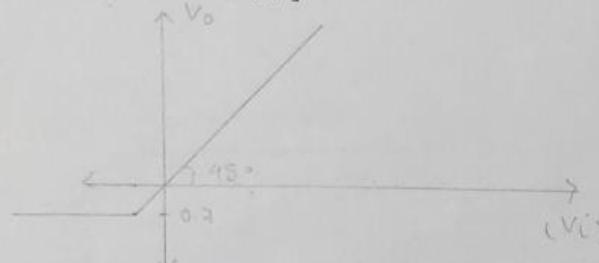
2. Draw the transfer characteristics for all the clipper configurations which are part of your today's practical (Practical - 7).

Q2.7

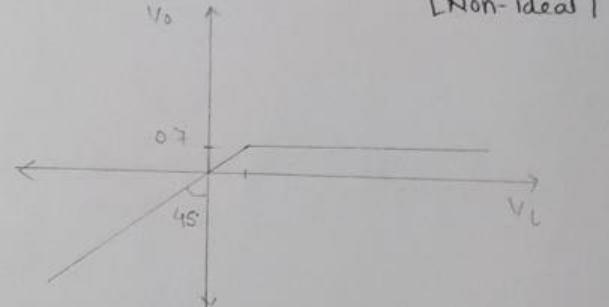
U19CS012

(5)

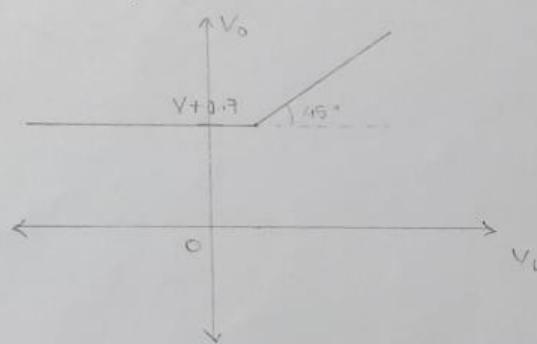
(1) Shunt Negative Clipper  
[Non-Ideal]



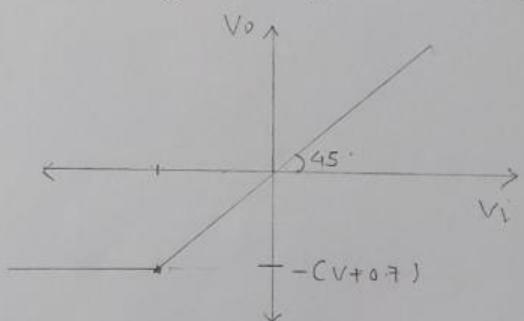
(2) Shunt Positive Clipper  
[Non-ideal]



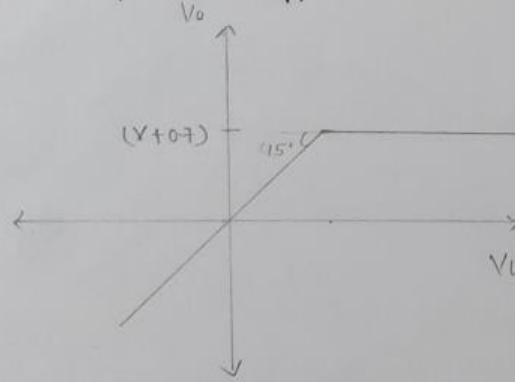
(3) Shunt Negative Clipper with Bias-I



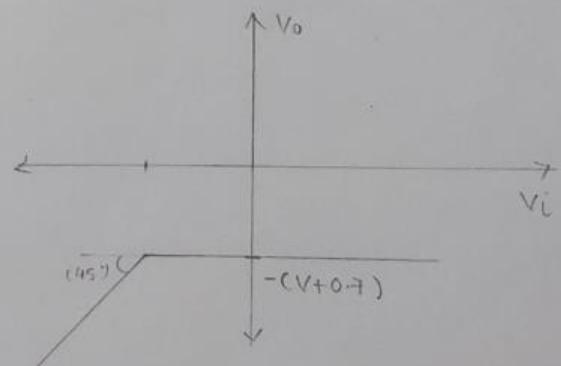
(4) Shunt Negative Clipper with Bias II



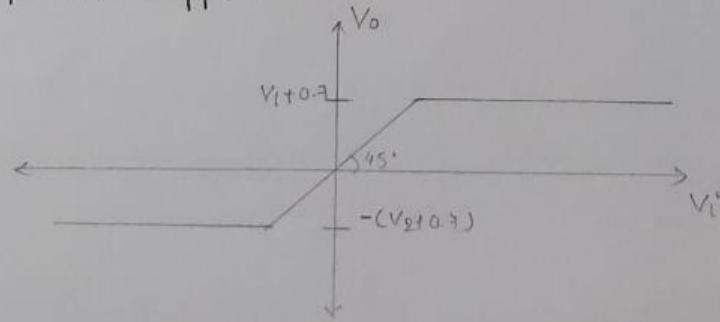
(5) Shunt positive clipper with Bias-I



(6) Shunt positive clipper with bias II

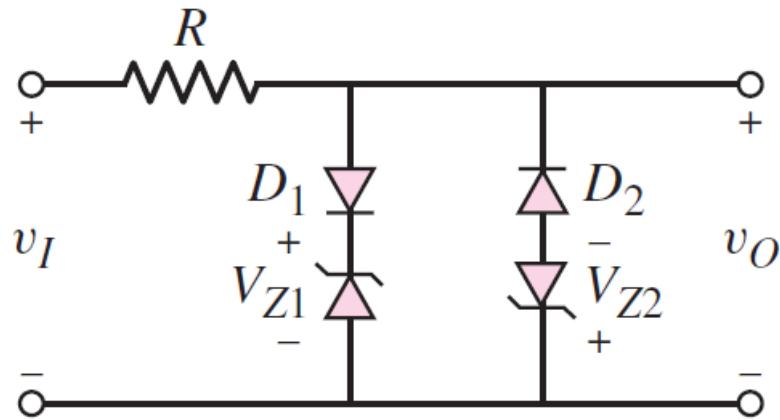


(7) Dual Clipper



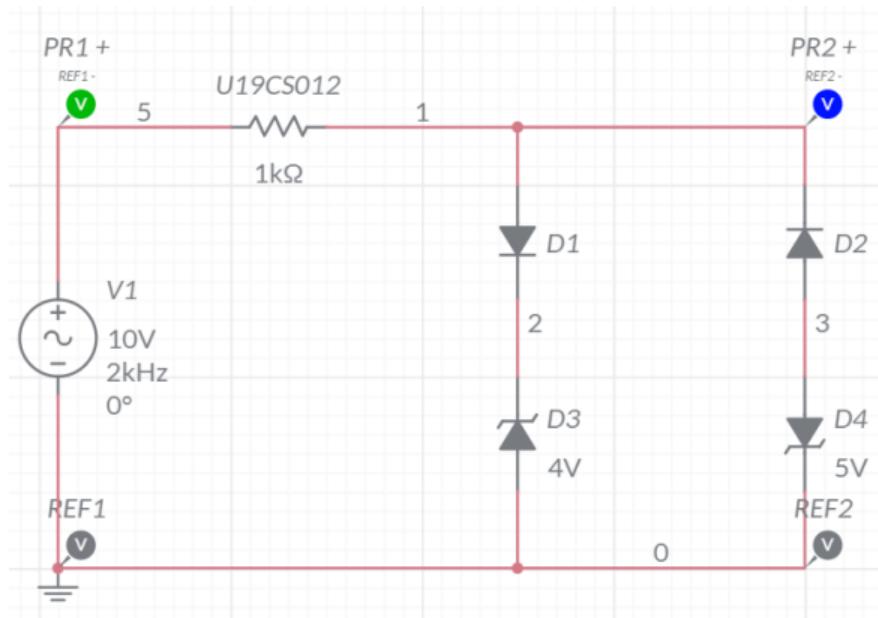


3. Assuming Symmetrical Sine wave input with peak value greater than the Zener reference voltage, predict the output and plot the Transfer Characteristics for the following Clipper Circuits:



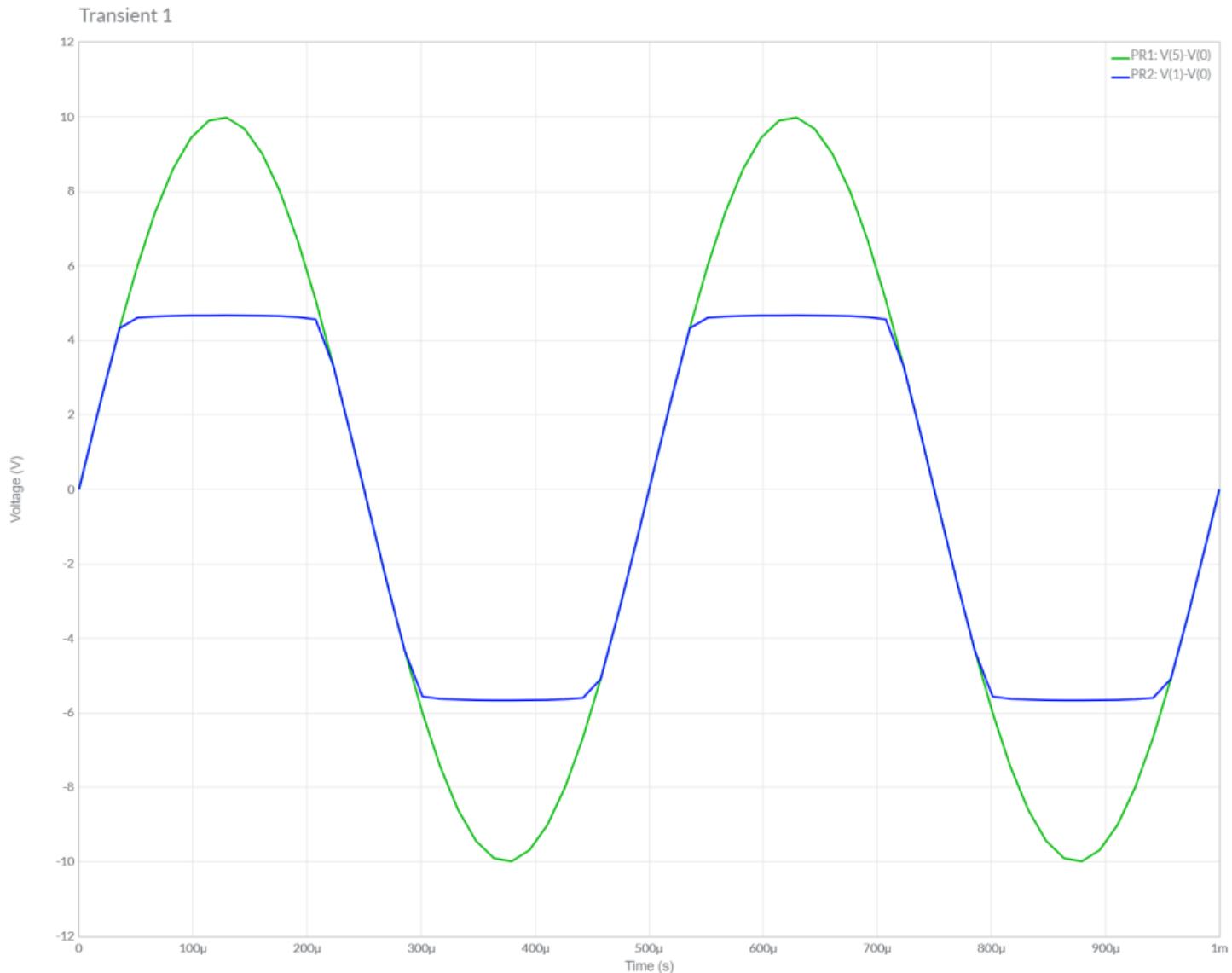
A.) Multisim Calculations:

1.) Circuit Image:



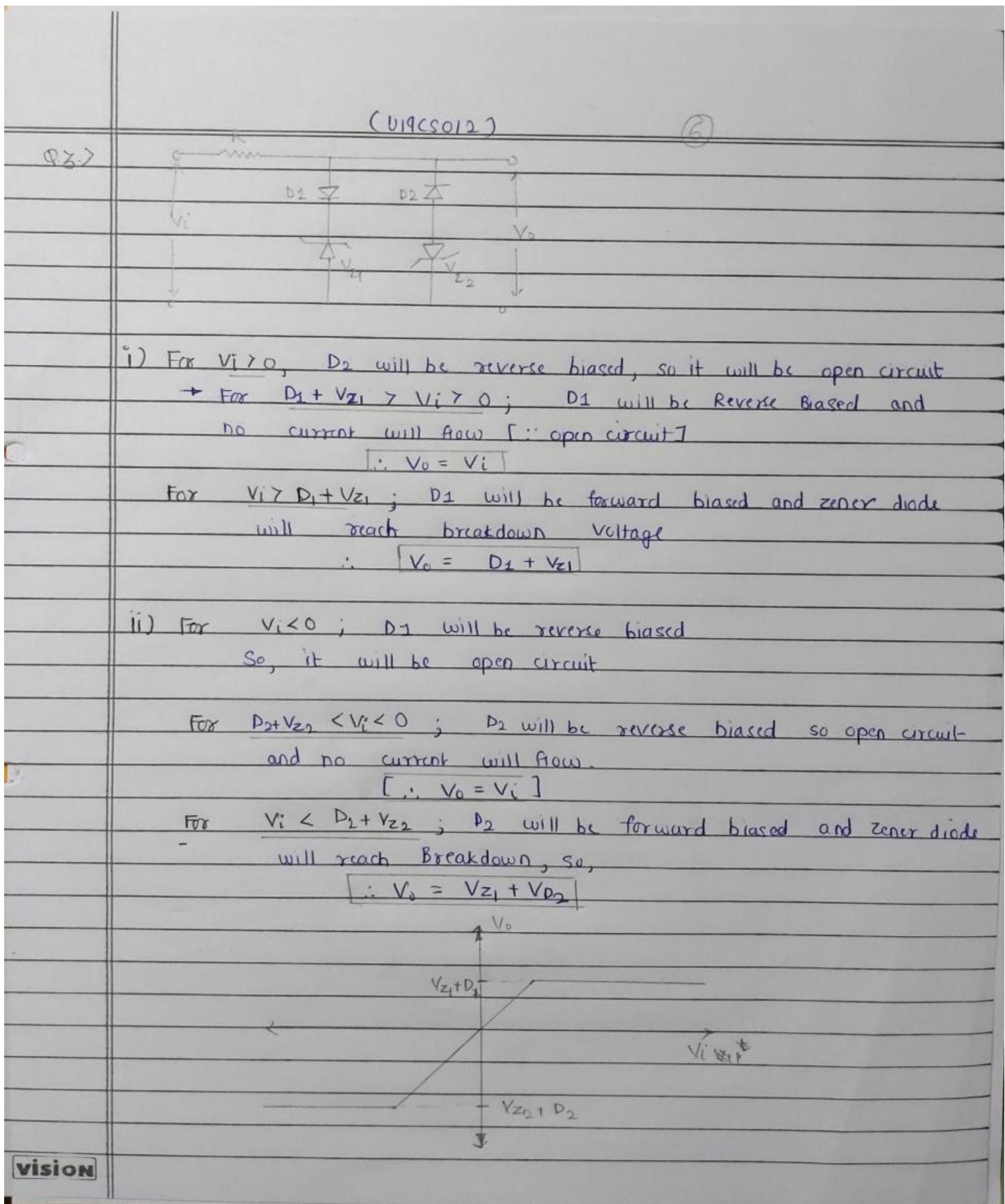


**2.) Grapher Image:**





## B.) Theoretical Calculations:



VISION



Expt. No:

8

Date:

22/10/2020

**Diode Clamper Circuits**

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

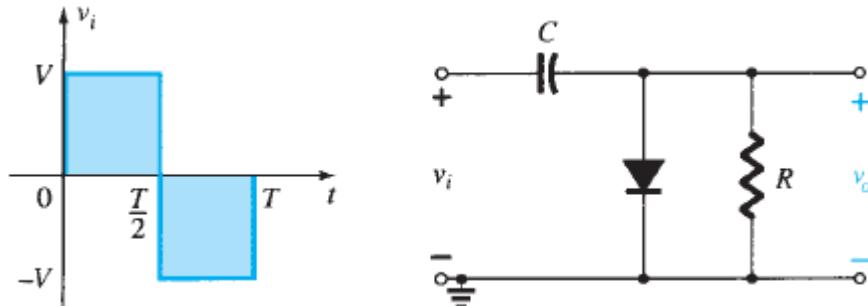
**SOFTWARE TOOLS / OTHER REQUIREMENTS:**

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.

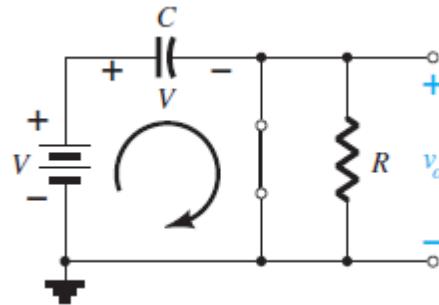
**Analysis**

**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.

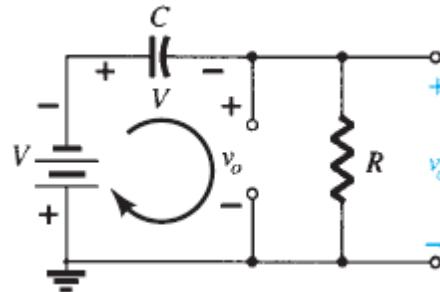


**Fig. a**

**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

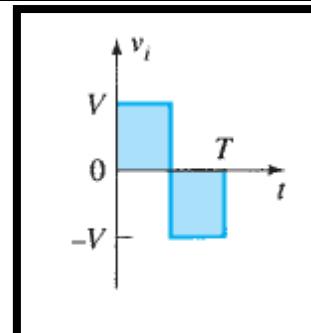
$$\begin{aligned} -V - V - v_o &= 0 \\ v_o &= -2V \end{aligned}$$

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

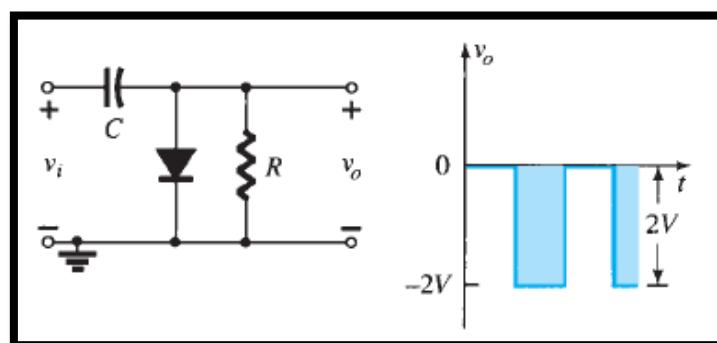
**Step 5: Check that the total swing of the output matches that of the input.**

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

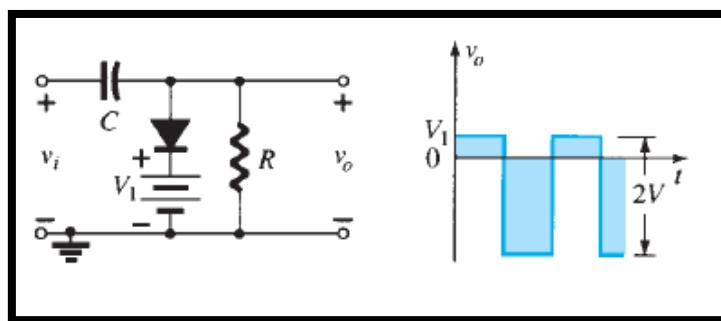
#### Few Clamper Configurations



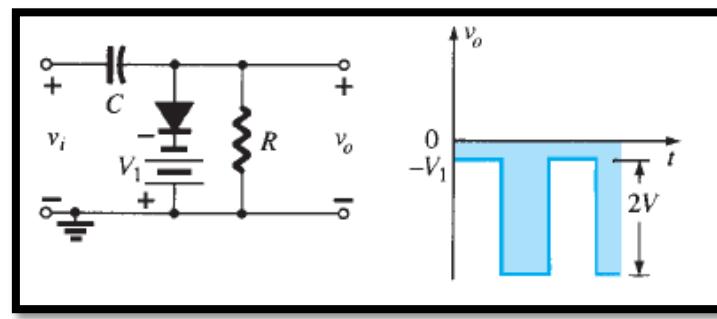
Common Input for all below circuits



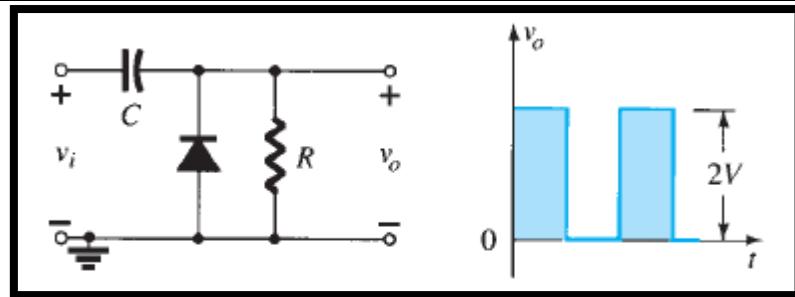
Negative Clamper with Zero Bias



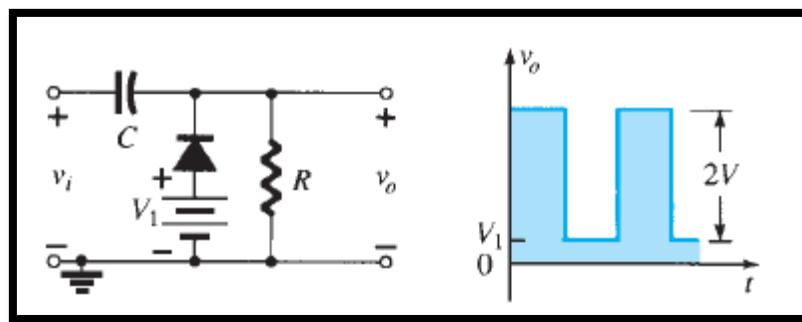
Negative Clamper with Positive Bias



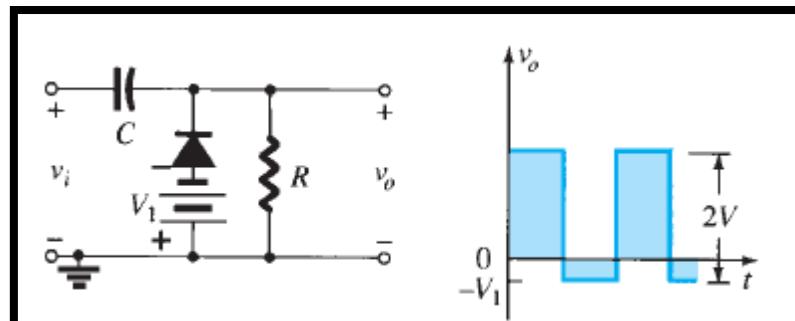
Negative Clamper with Negative Bias



Positive Clamper with Zero Bias



Positive Clamper with Positive Bias

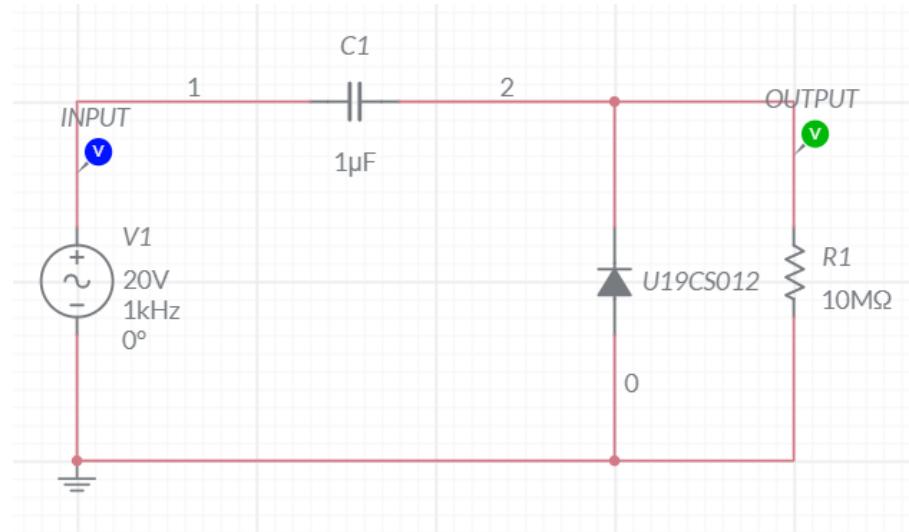


Positive Clamper with Negative Bias

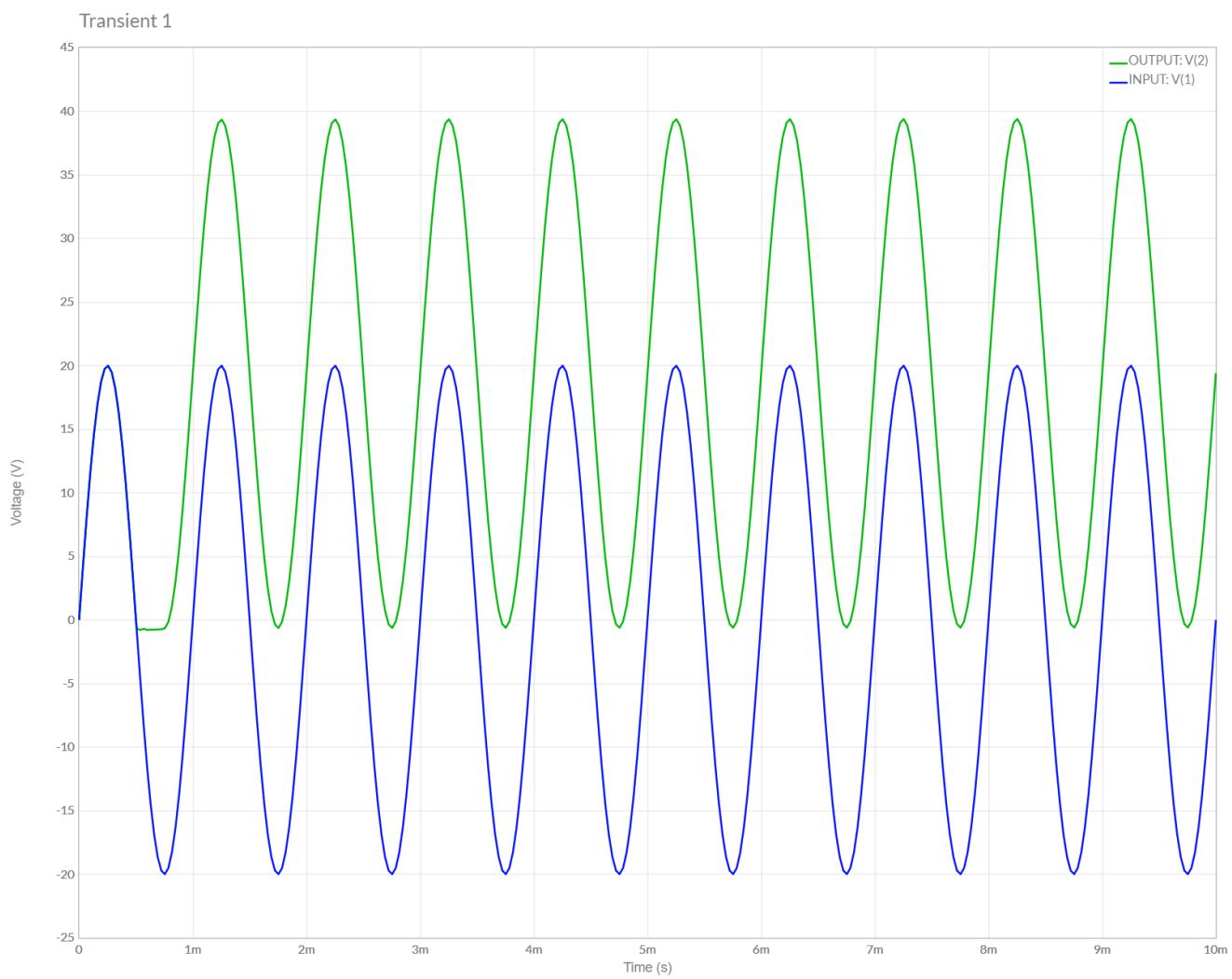


### A.) POSITIVE CLAMPER WITH NO BIAS

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



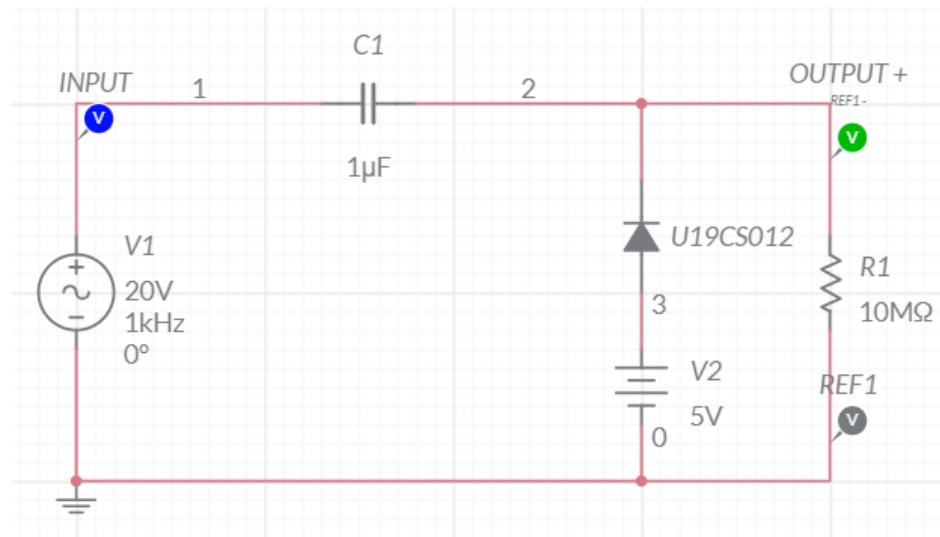
**WAVEFORMS (FROM MULTISIM)**



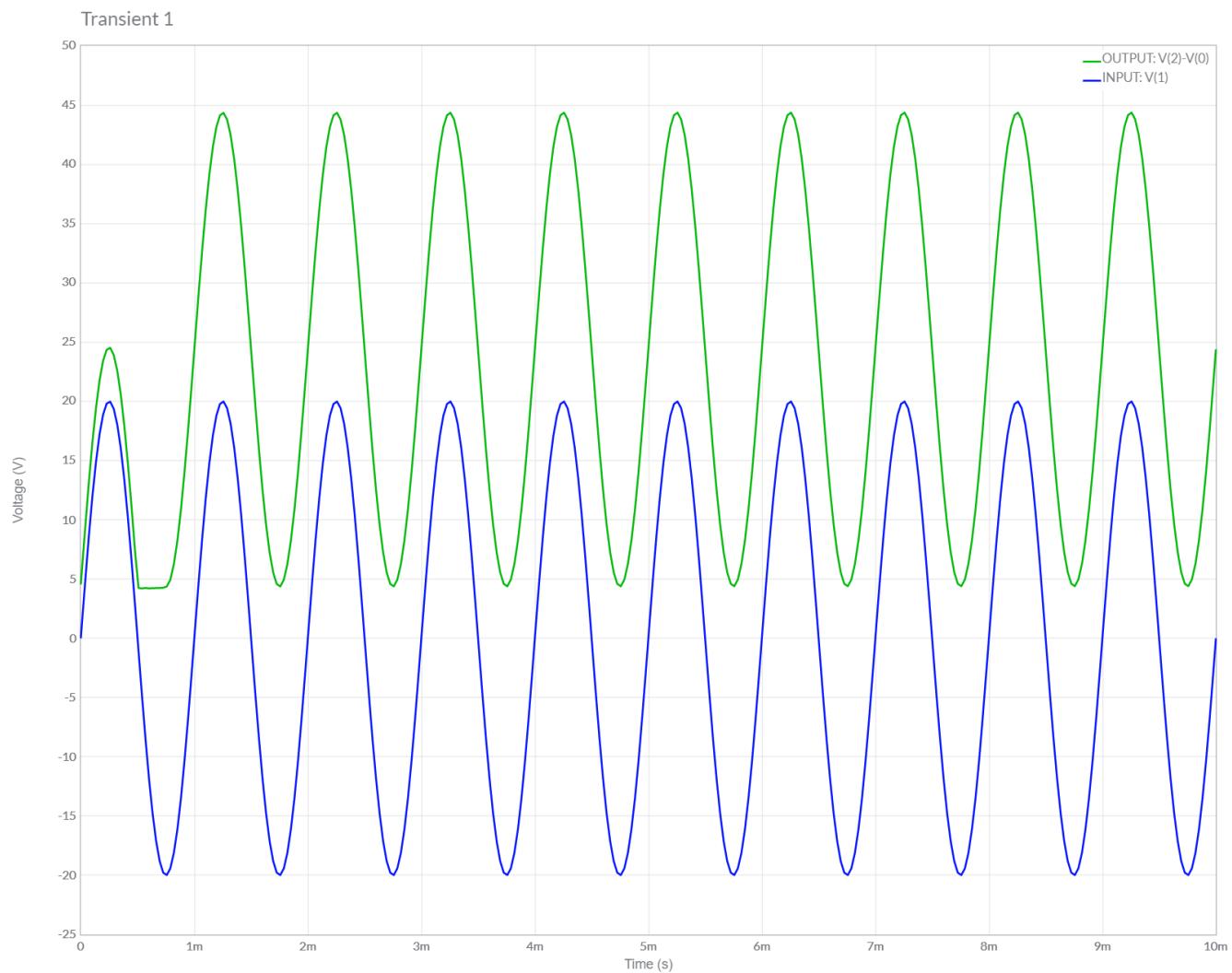


### B.) POSITIVE CLAMPER WITH POSITIVE DC BIAS

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



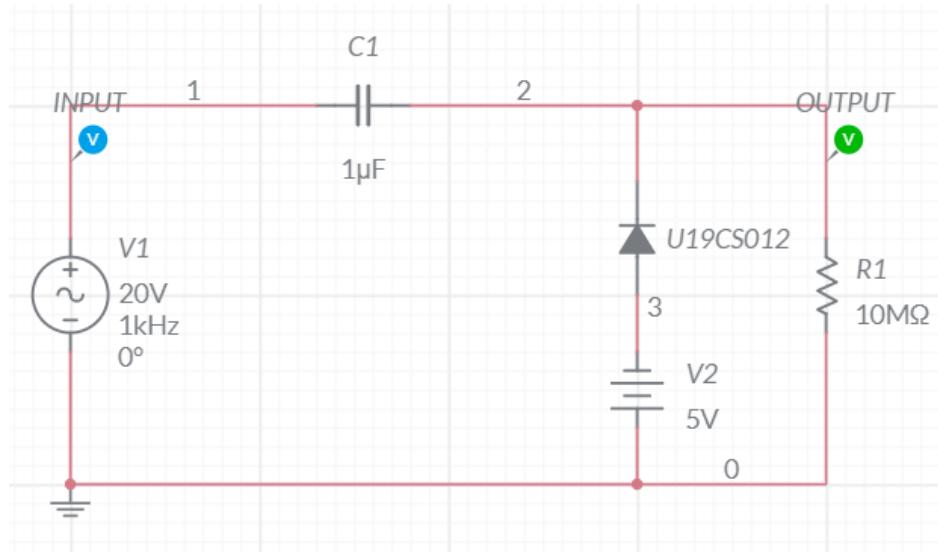
**WAVEFORMS (FROM MULTISIM)**



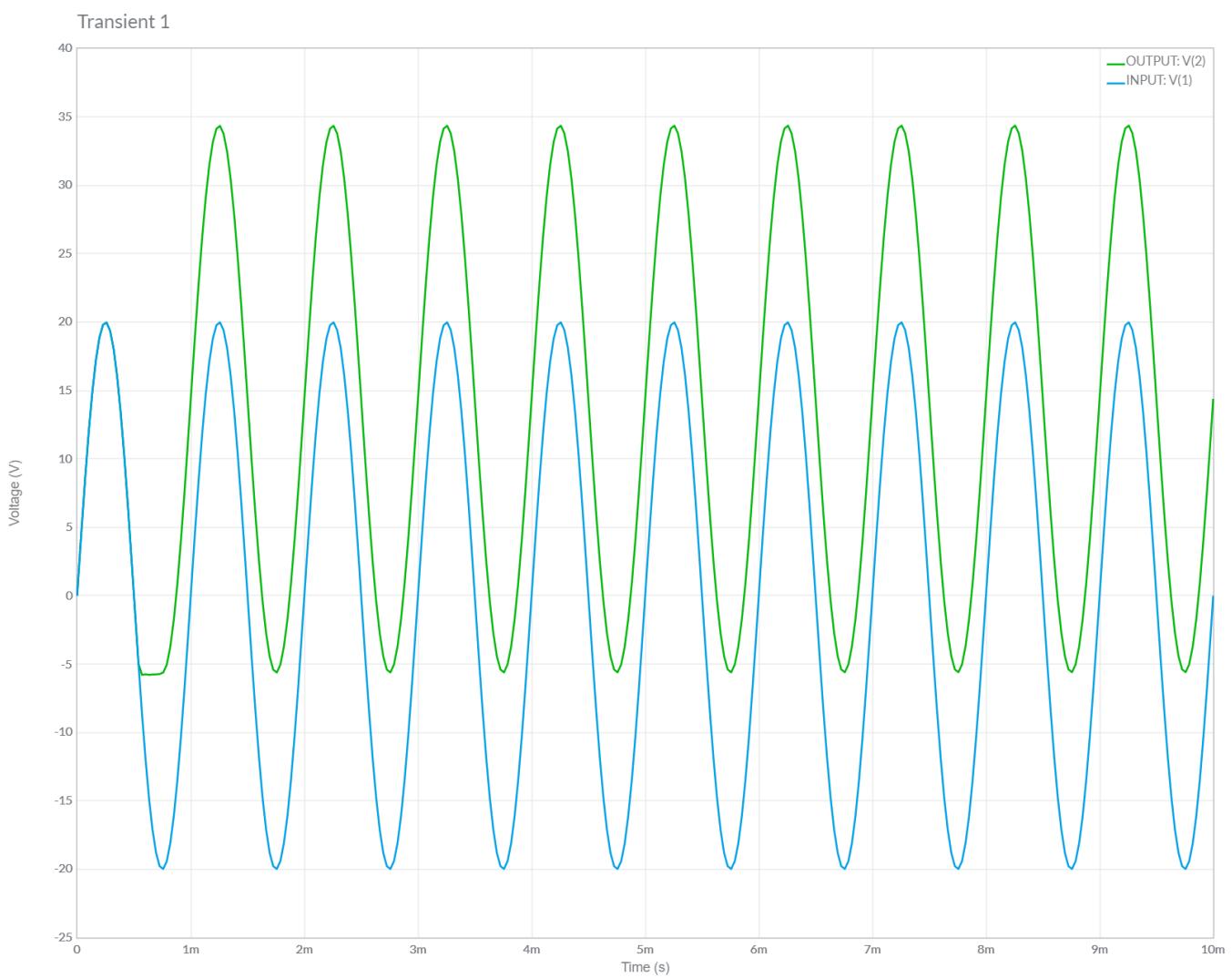


### C.) POSITIVE CLAMPER WITH NEGATIVE DC BIAS

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



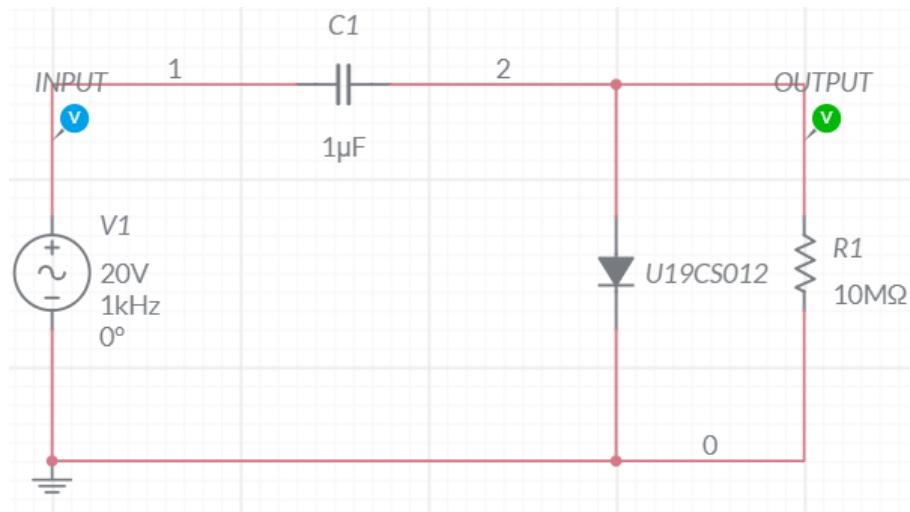
**WAVEFORMS (FROM MULTISIM)**



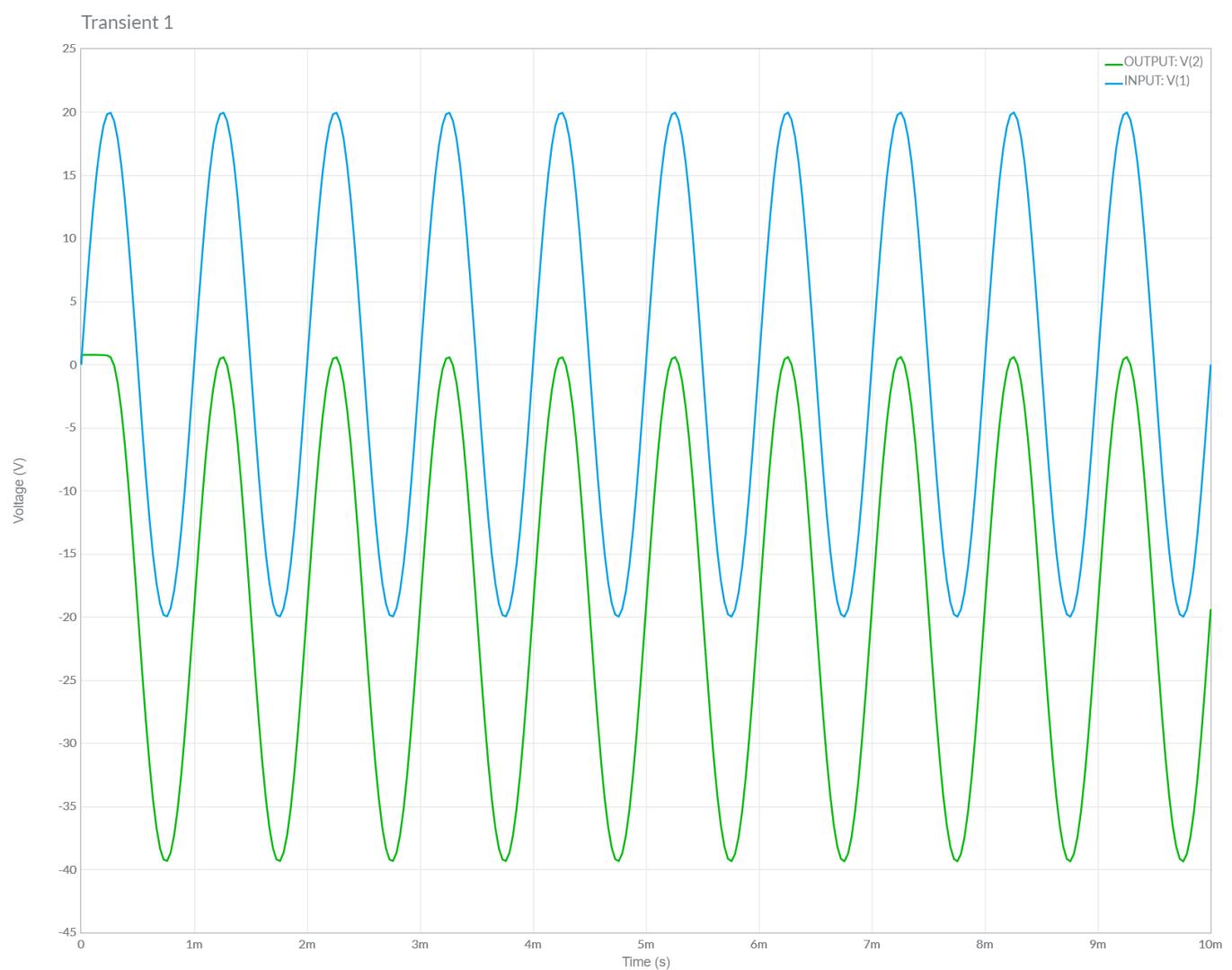


### D.) NEGATIVE CLAMPER WITH NO BIAS

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



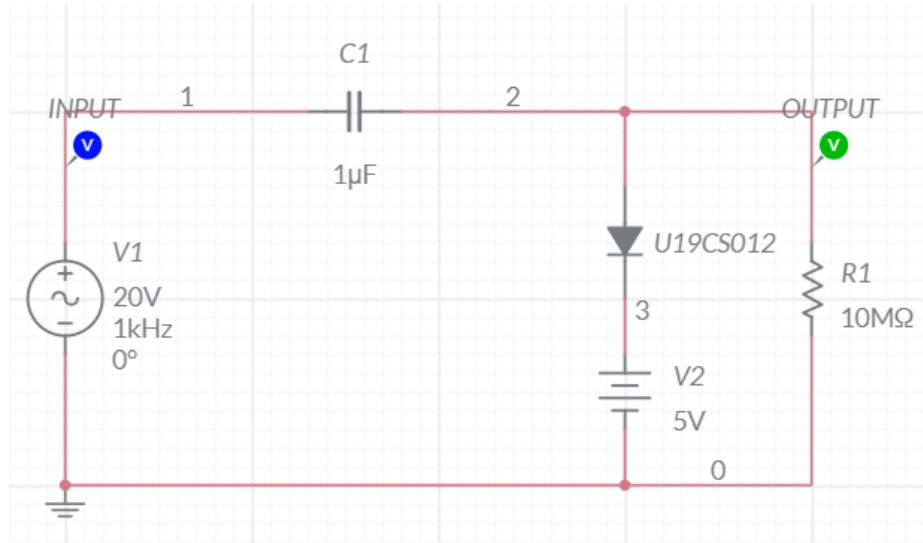
**WAVEFORMS (FROM MULTISIM)**



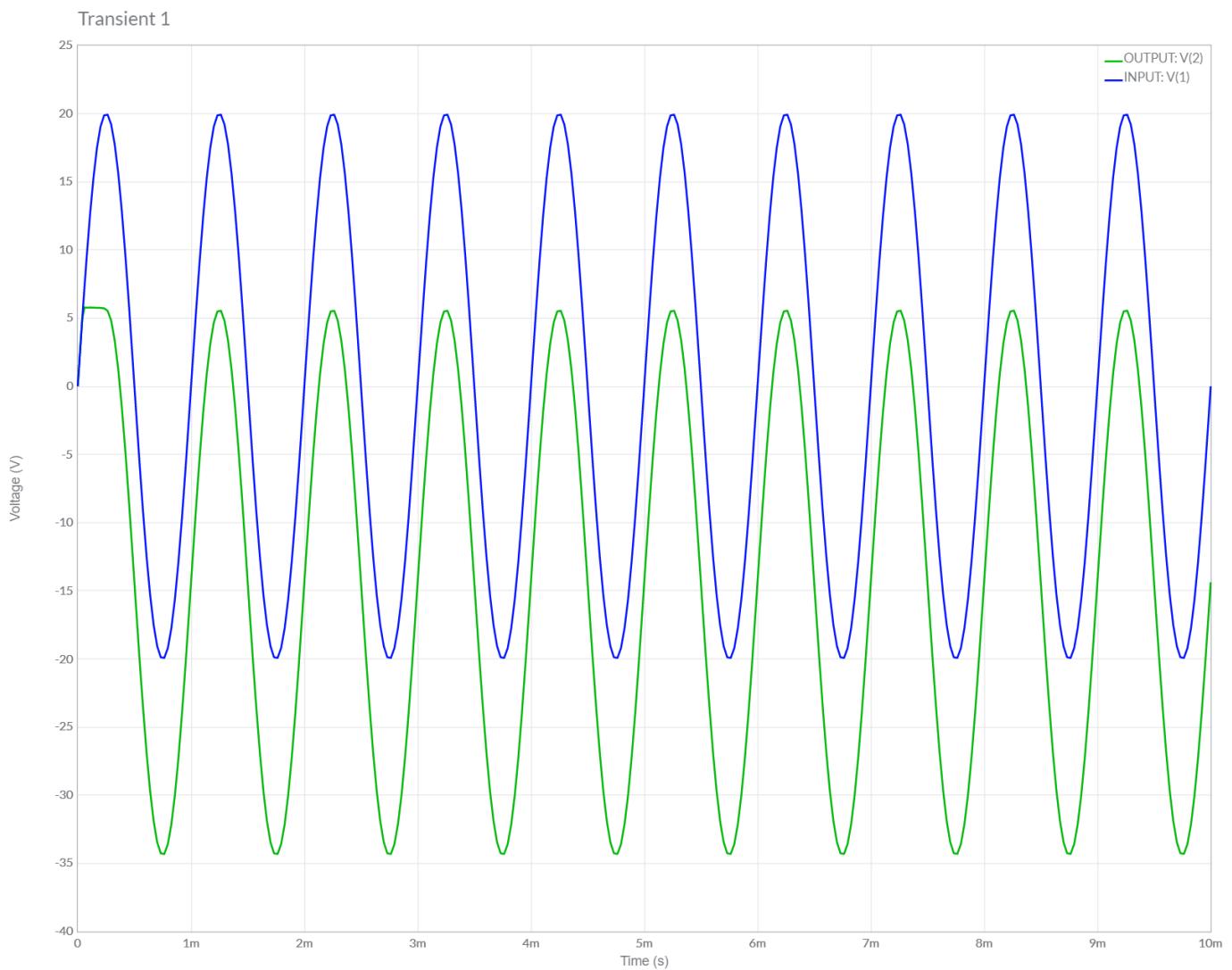


### E.) NEGATIVE CLAMPER WITH POSITIVE DC BIAS

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



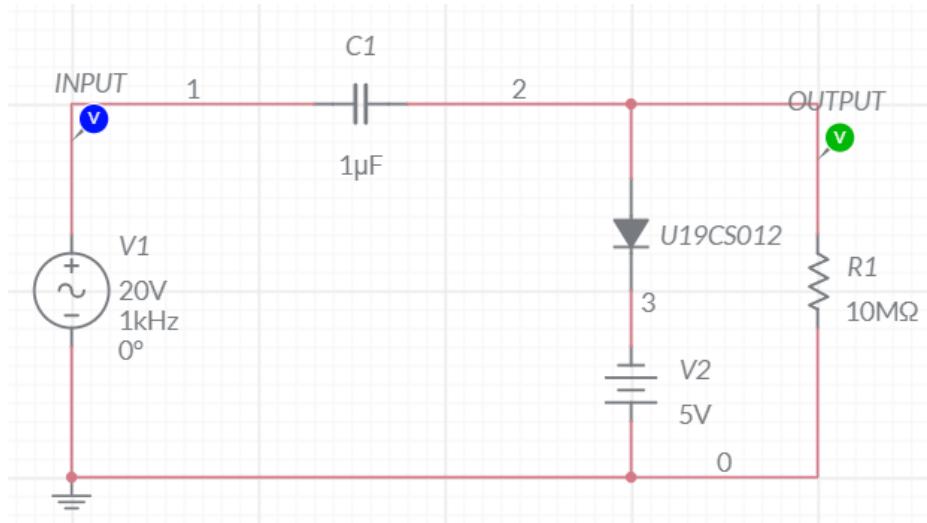
**WAVEFORMS (FROM MULTISIM)**



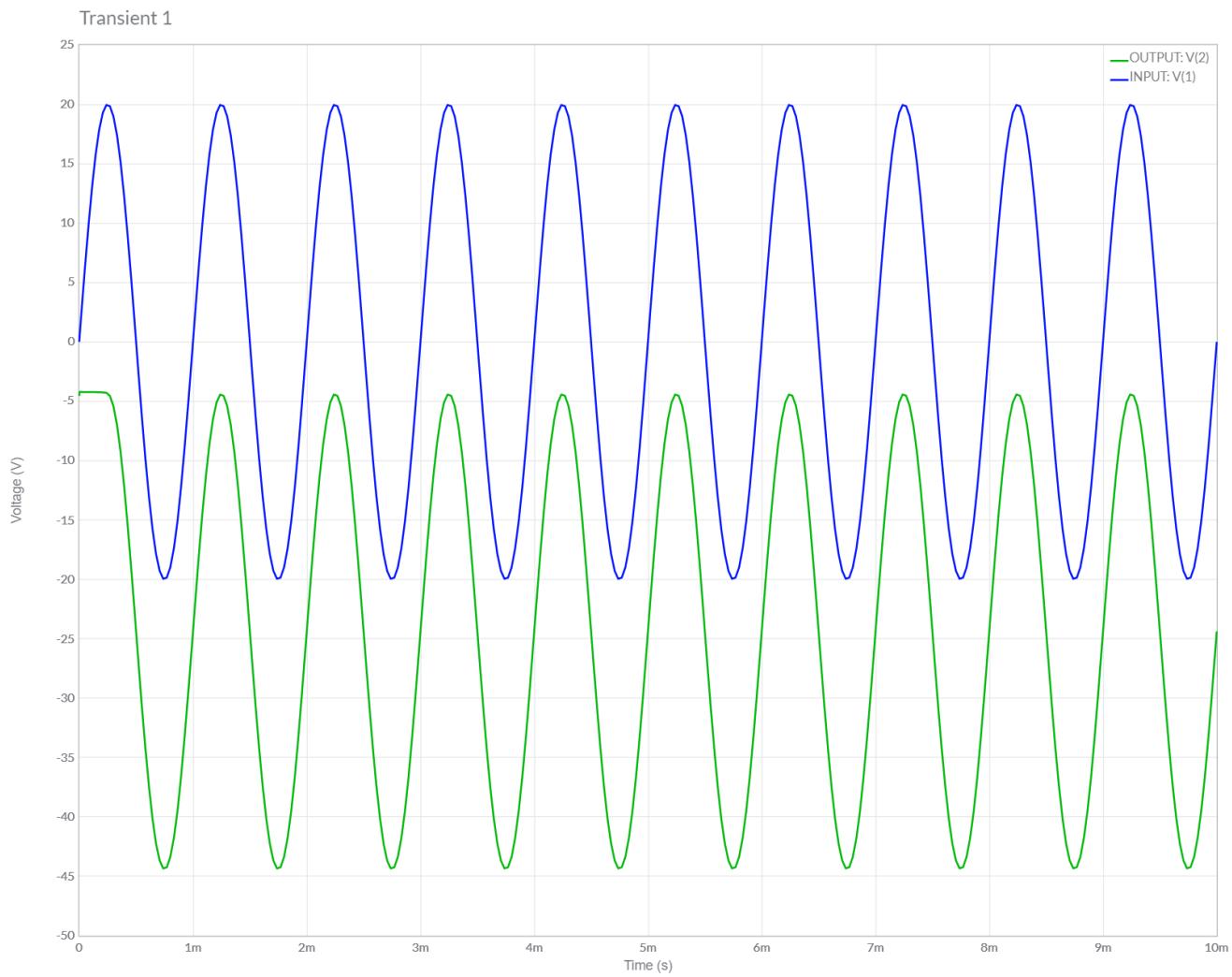


### F.) NEGATIVE CLAMPER WITH NEGATIVE DC BIAS

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



**WAVEFORMS (FROM MULTISIM)**





## **CONCLUSIONS**

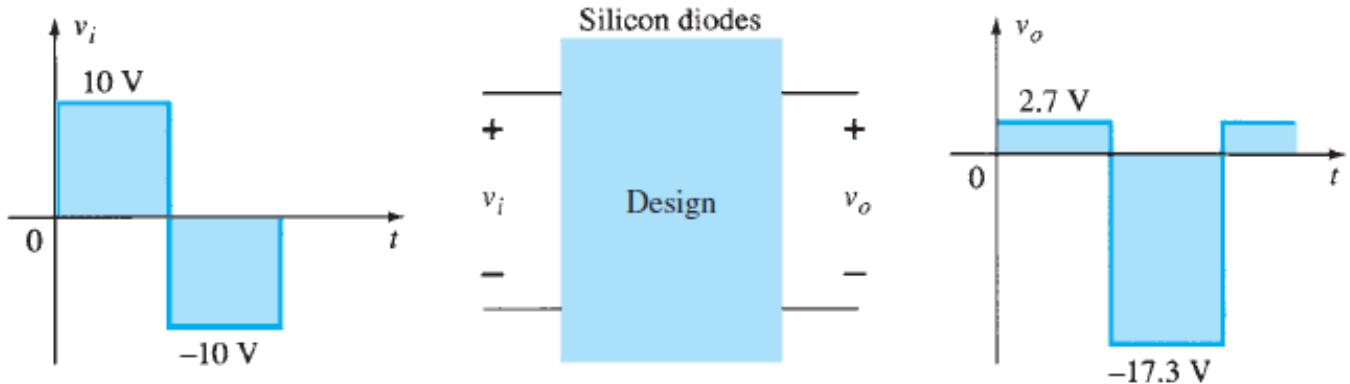
- 1.) In this Experiment, We have studied about Clamper Circuits [Both Positive and Negative] along with Different Biasing Applied [No Biasing, Positive Biasing and Negative Biasing].
- 2.) We Verified the Theoretical Knowledge of Clampers by Performing Simulations of 6 Cases of Clamper Circuits in Multisim.
- 3.) Hence, we have Successfully Designed, Plotted and Verified Various Clamper Circuits.



# **ASSIGNMENT - 8**

U19CS012

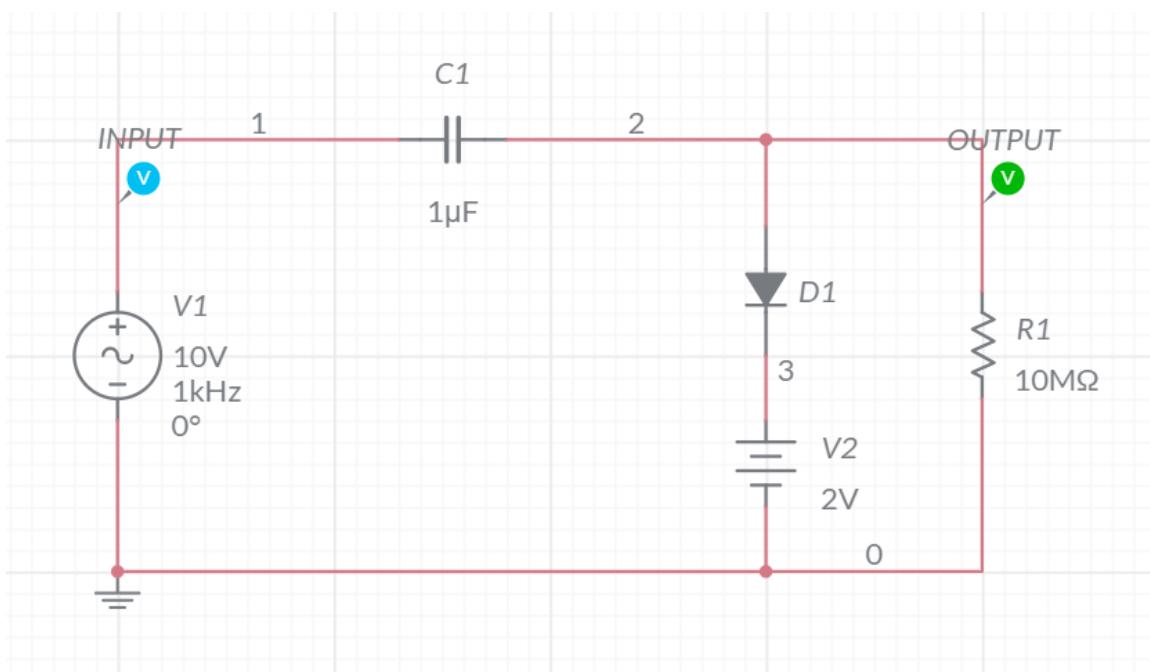
1. Design a Clamper to perform the following operation.



Answer :

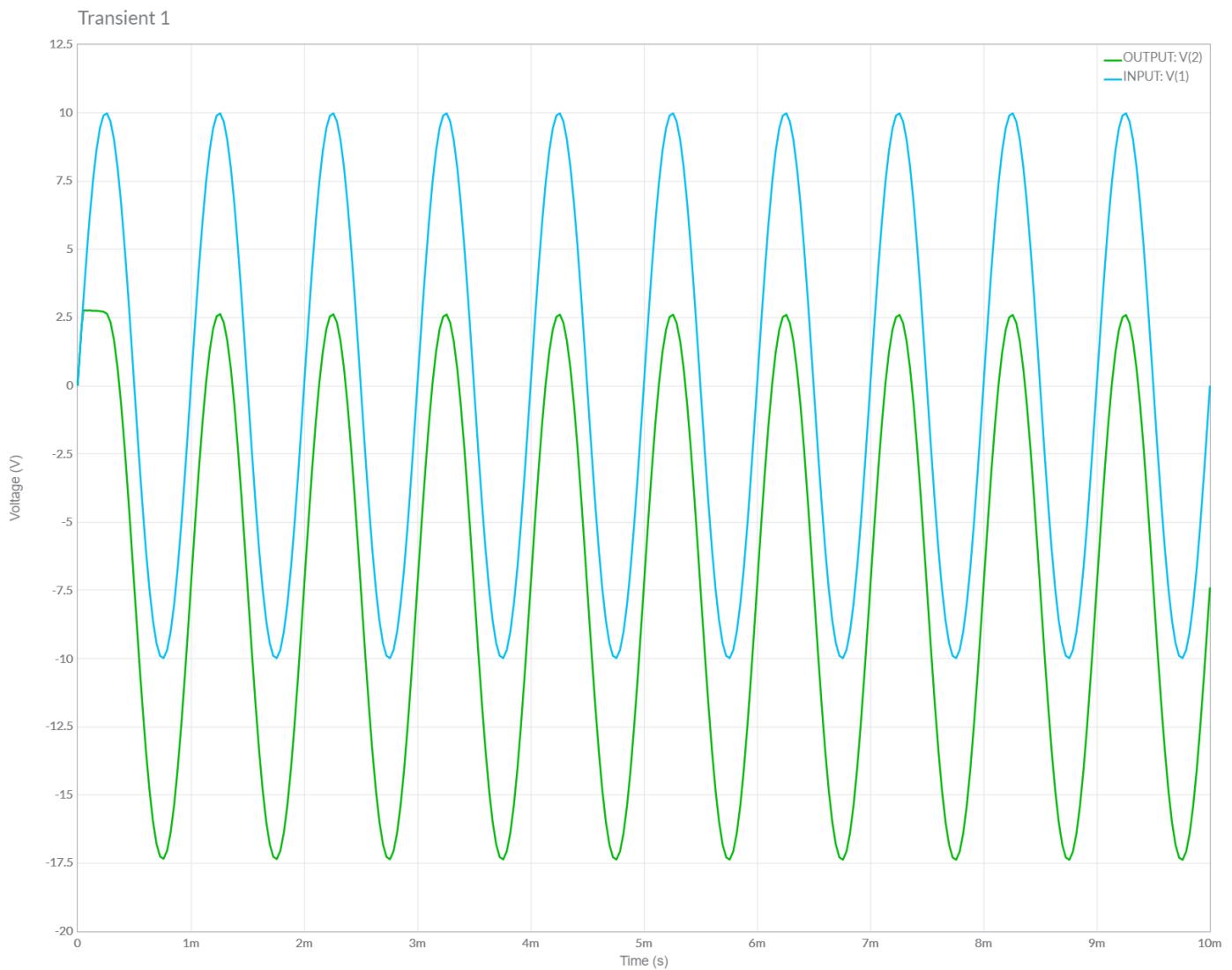
Since the Waveform is Shifted Downwards, **Negative Clipper** Must be used.  
But, the Waveform needs to be Shifted Slightly Upward by 2.7 V  
[Forward Bias = 2V + 0.7V [Diode]]

### 1.) Circuit Image:





**2.) Grapher Image:**





Expt. No:

**9**

Date:

**22/10/2020**

## Full Wave Rectifier

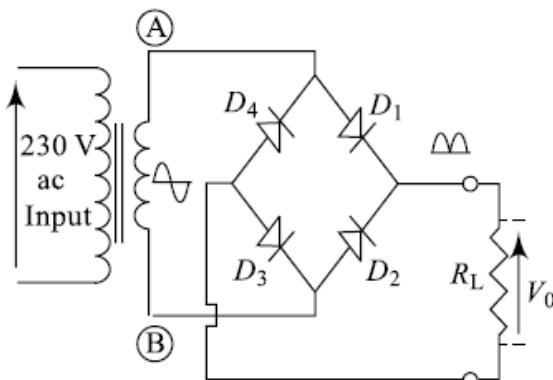
**AIM:** To study, design and implement Full Wave Rectifier Circuit.

### **SOFTWARE TOOLS / OTHER REQUIREMENTS:**

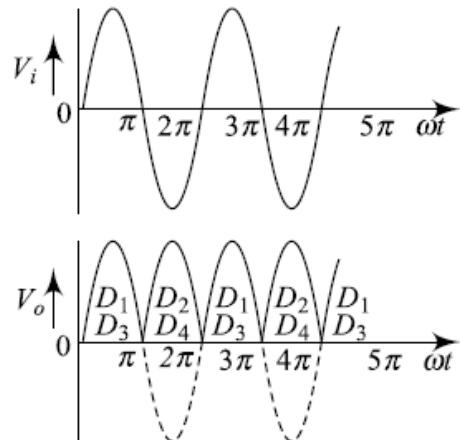
1. Multisim Simulator/Circuit Simulator

### **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.



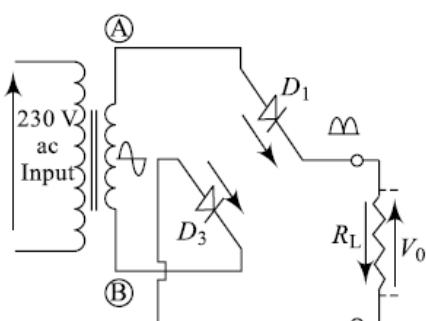
(a) Bridge rectifier circuit



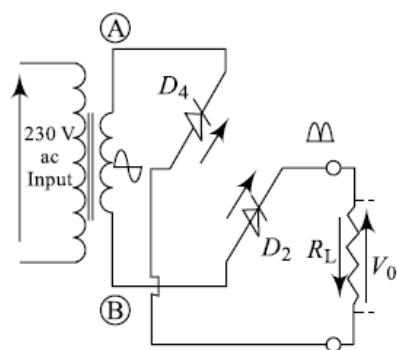
(b) Input/output waveforms

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  $\pi$  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.

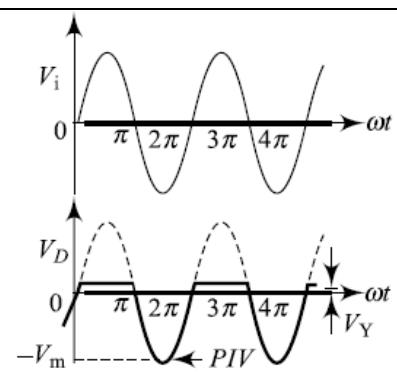
from  $\omega t = \pi$  to  $2\pi$  and the diodes D1 and D3 will be in OFF condition. Thus, between  $\omega t = 0$  to  $\pi$ , D1 and D3 conduct and result in an output, between  $\omega t = \pi$  to  $2\pi$ , D2 and D4 conduct and result in an output  $V_o$  as indicated in Fig. (b). It can therefore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



(a) Positive half-cycle

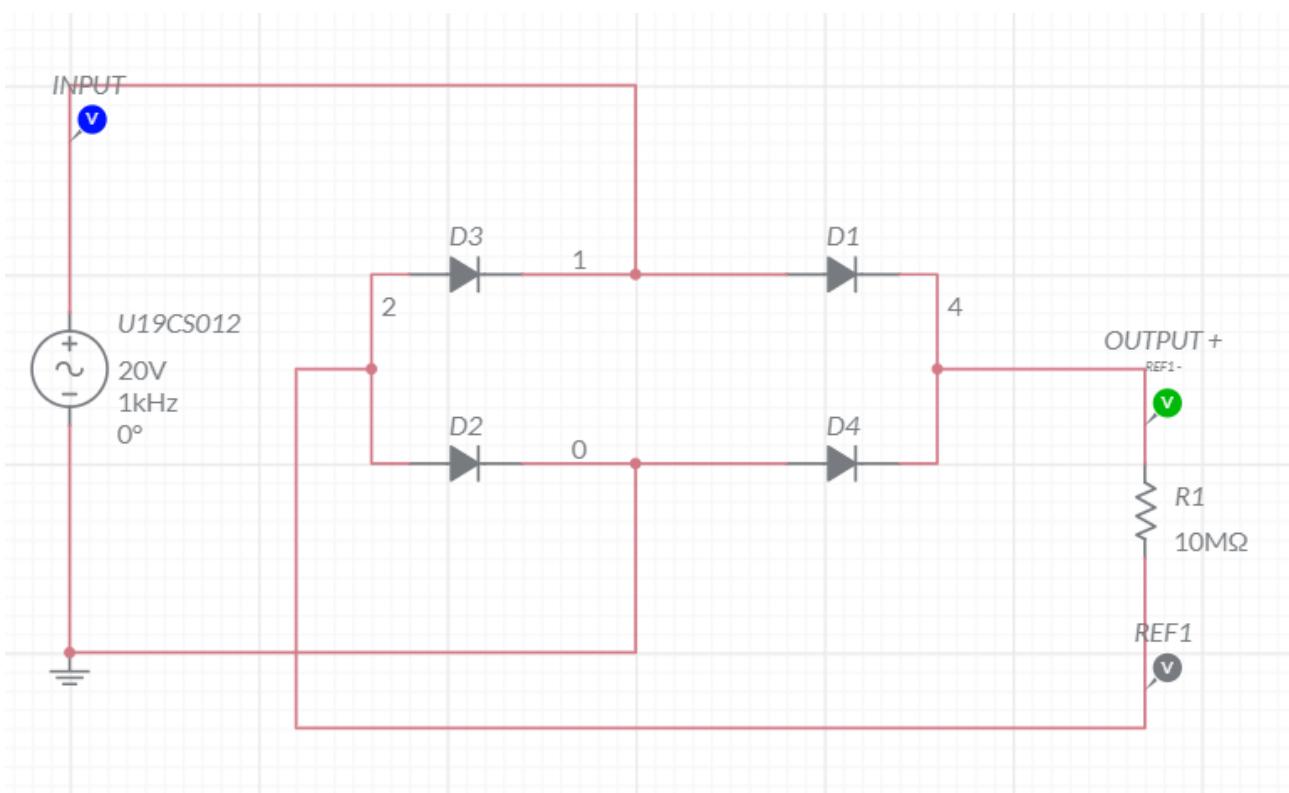


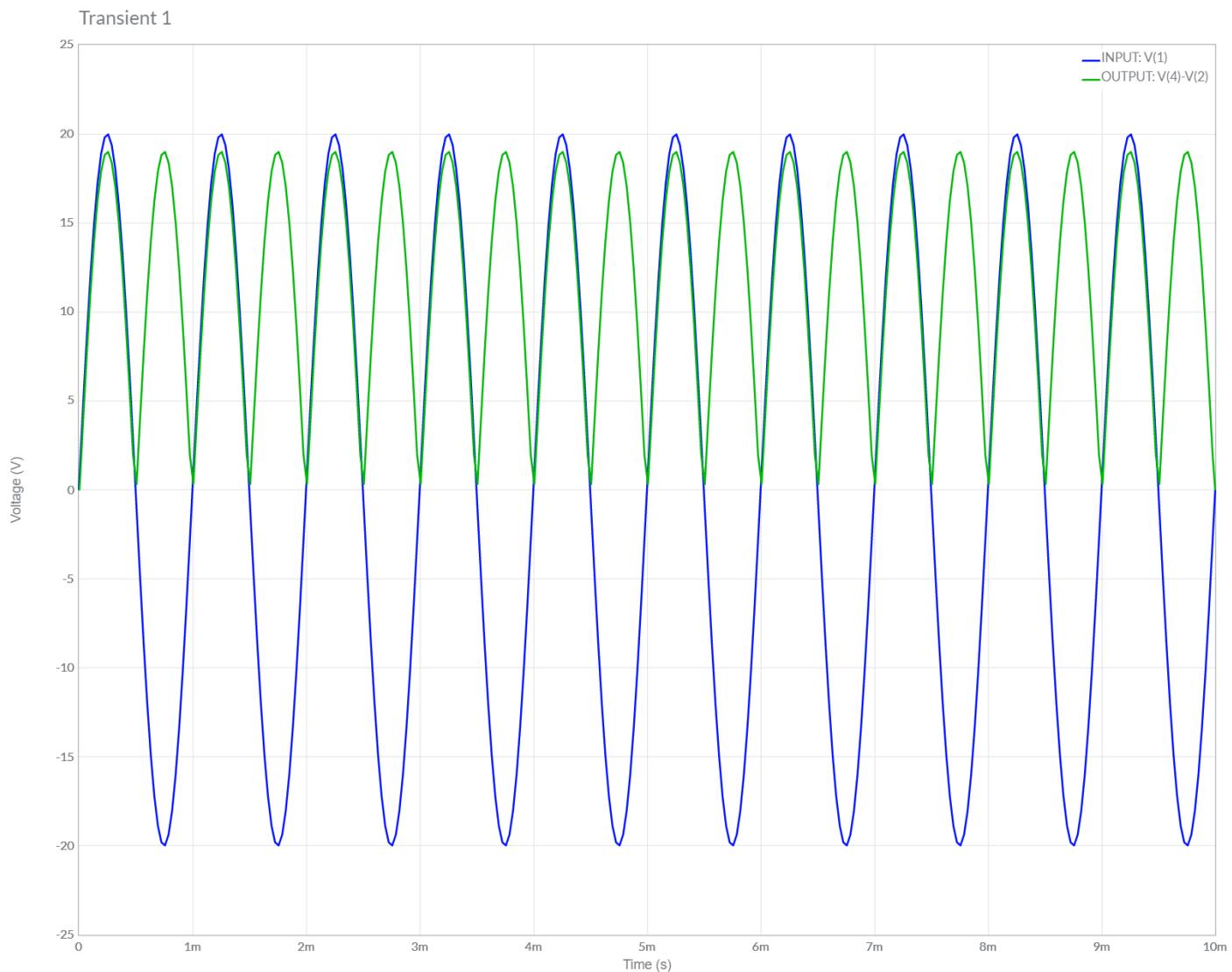
(b) Negative half-cycle



(c) Diode waveforms

### CIRCUIT DIAGRAM (FROM MULTISIM)



**WAVEFORMS (FROM MULTISIM)****CONCLUSIONS**

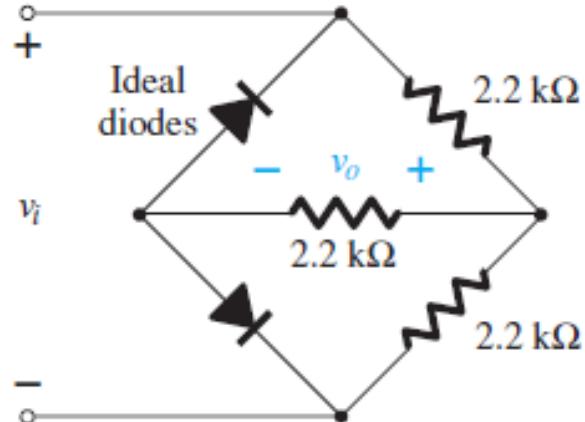
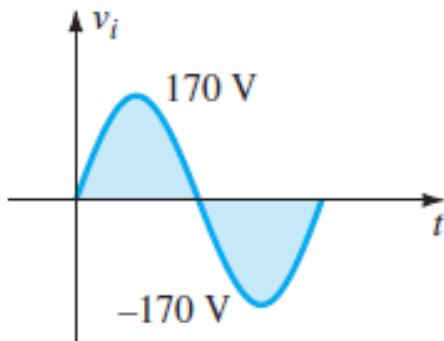
- 1.) In this Experiment, We have studied about Full Wave Rectifier Circuit and its Working.
- 2.) We Verified the Theoretical Knowledge of Full Wave Rectifier Circuit by Performing Simulations of Full Wave Rectifier Circuit in Multisim.
- 3.) Hence, we have Successfully Designed, Plotted and Verified Full Wave Rectifier Circuit.



# **ASSIGNMENT-9**

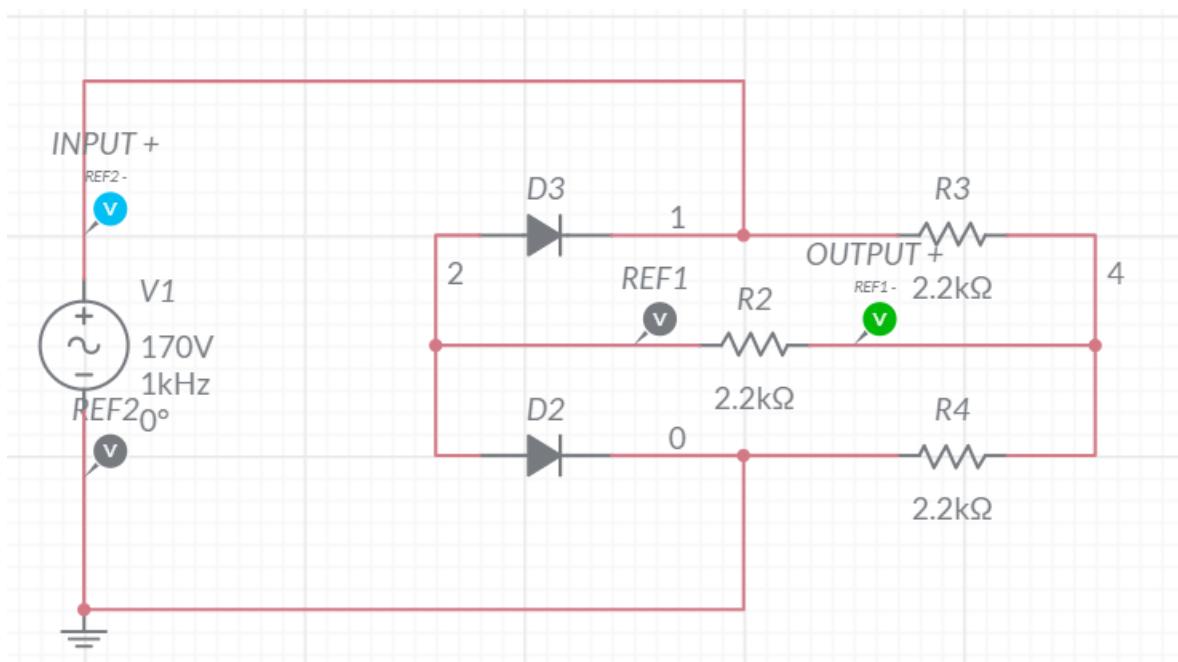
U19CS012

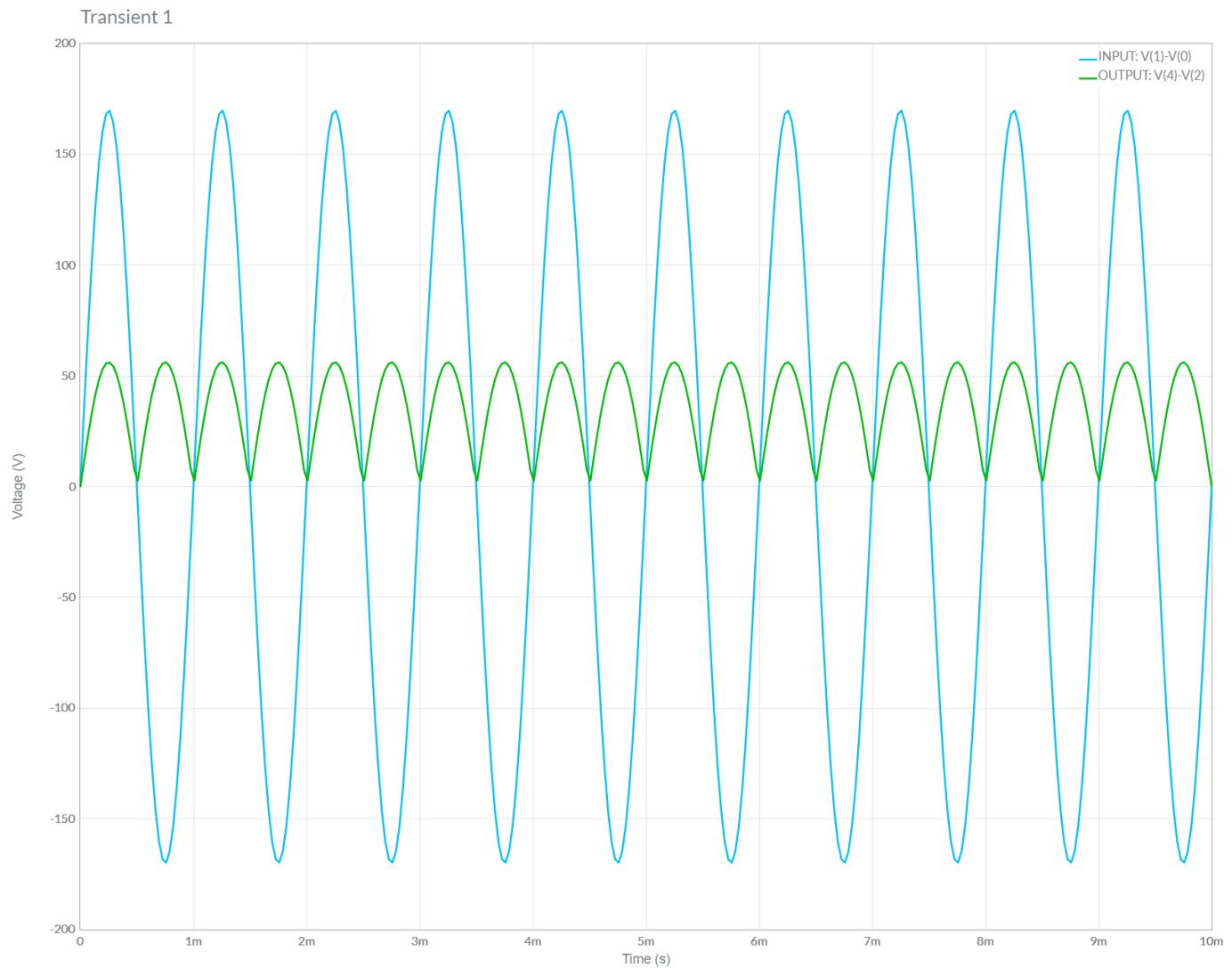
1. Calculate and Plot  $V_o$  for the following circuit.



A.) Multisim Calculations:

1.) Circuit Image:

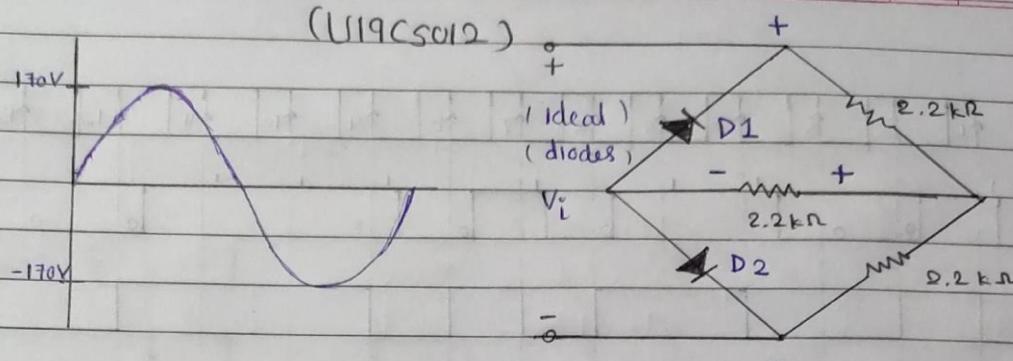


**2.) Grapher Image:****B.) Calculation Part**



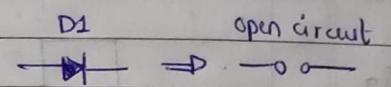
Page No.	
Date	

Q2)

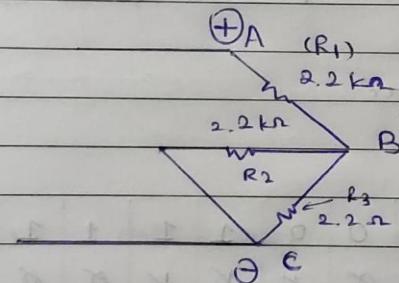
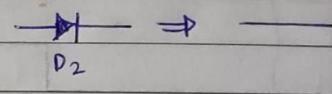


(A) During Positive half cycle

D1 will be Reverse biased



D2 will be short circuit  
(forward Biased)



$$\text{Equivalent} = R_1 + (R_2 \parallel R_3)$$

$$= 2.2 \text{ k}\Omega + \frac{1}{\left(\frac{1}{2.2} + \frac{1}{2.2}\right)}$$

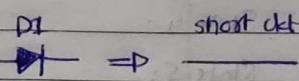
$$= [3.3 \text{ k}\Omega]$$

$$i = \frac{V}{R_{\text{req}}} = \frac{170}{(3.3 \text{ k}\Omega)}$$

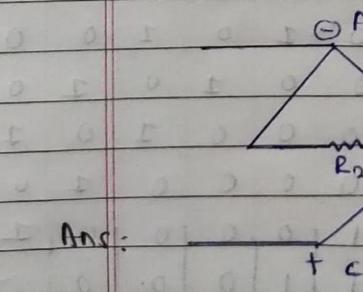
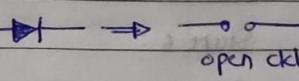
Ans:  $V_{R_2} = (V) \times \frac{(1.1)}{(1.1 + 2.2)} = 170 \times \frac{1}{3} = 170 \text{ V}$

(B) During Negative half cycle

D1 will be forward biased



D2 will be reverse biased



$$\text{Req} = R_3 + (R_1 \parallel R_2) = (2.2 + 1.1) \text{ k}\Omega = 3.3 \text{ k}\Omega$$

$$V_{R_2} = (V) \times \frac{(R_1 \parallel R_2)}{(R_2 + (R_1 \parallel R_2))} = 170 \times \frac{1.1}{(1.1 + 2.2)}$$

Ans:  $V_{R_2} = \frac{170}{3} \text{ V}$



Expt. No:

**10**

Date:

**29-10-2020**

## **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:**

1. 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.

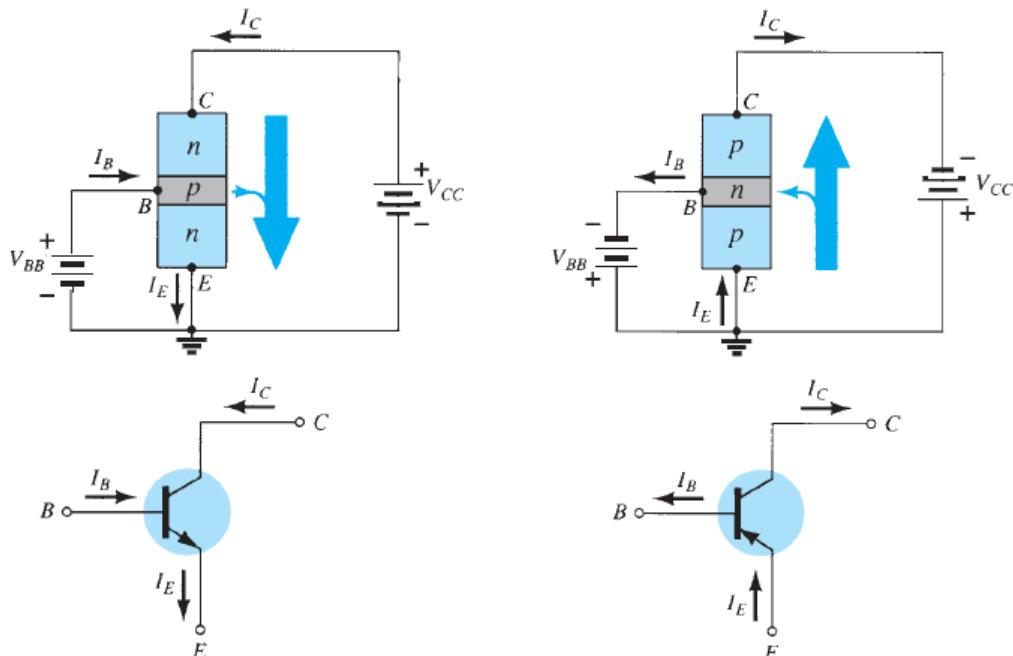


Fig. 10.1

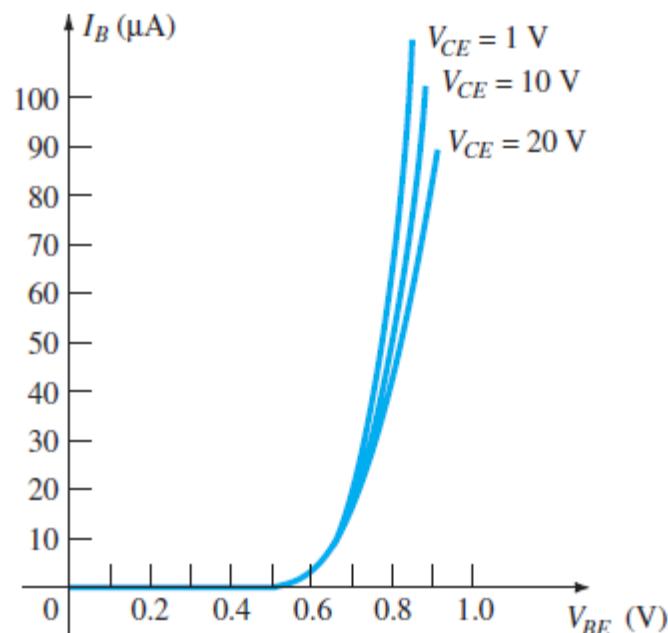


Fig. 10.2 (a) CE Input Characterisitcs

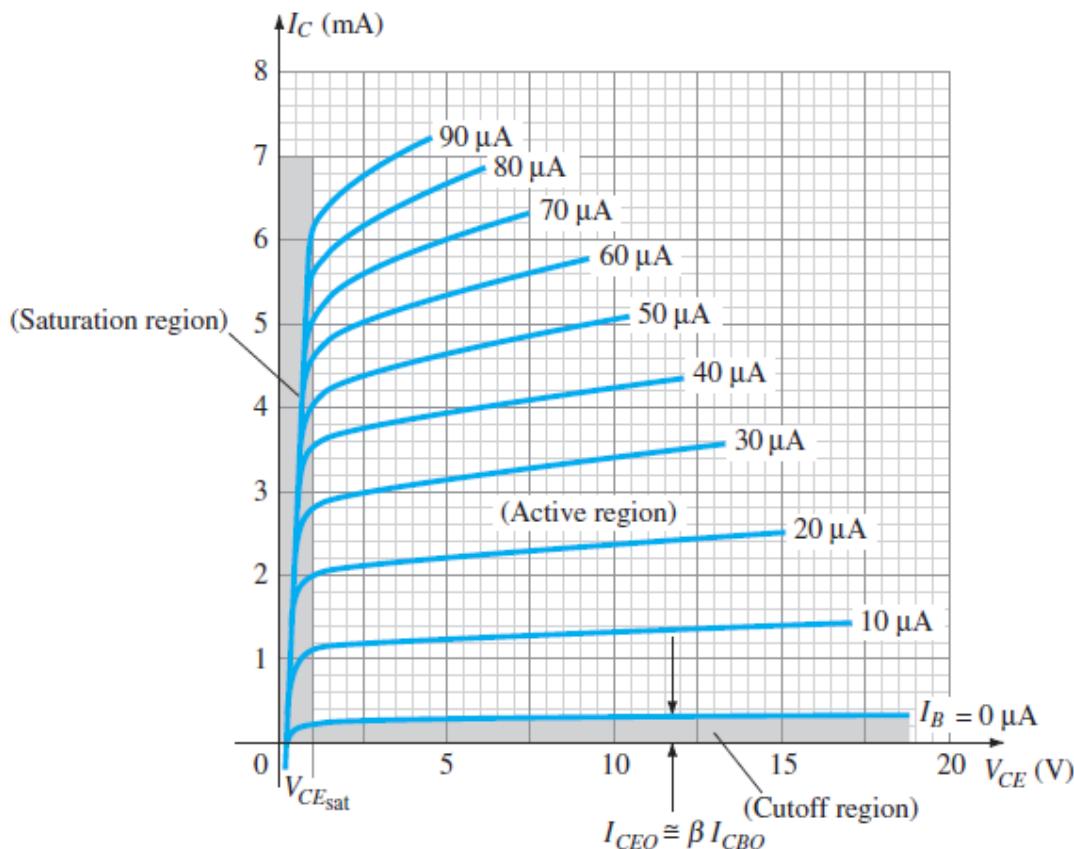


Fig. 10.2 (b) CE Output Characterisitcs



## 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.

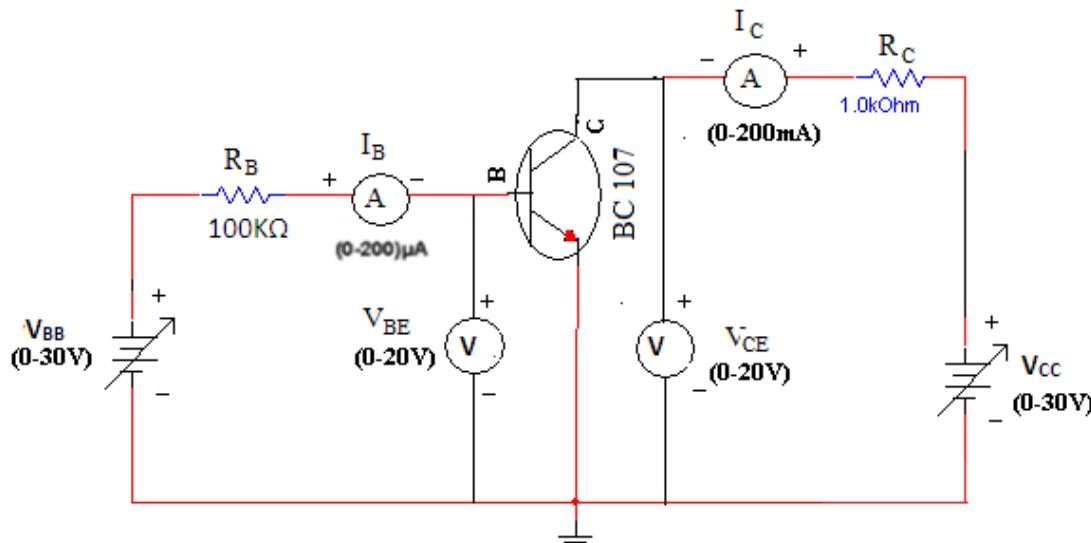


Fig. 10. 3 Circuit Diagram to obtain CE Input/Output Characteristics

## 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.

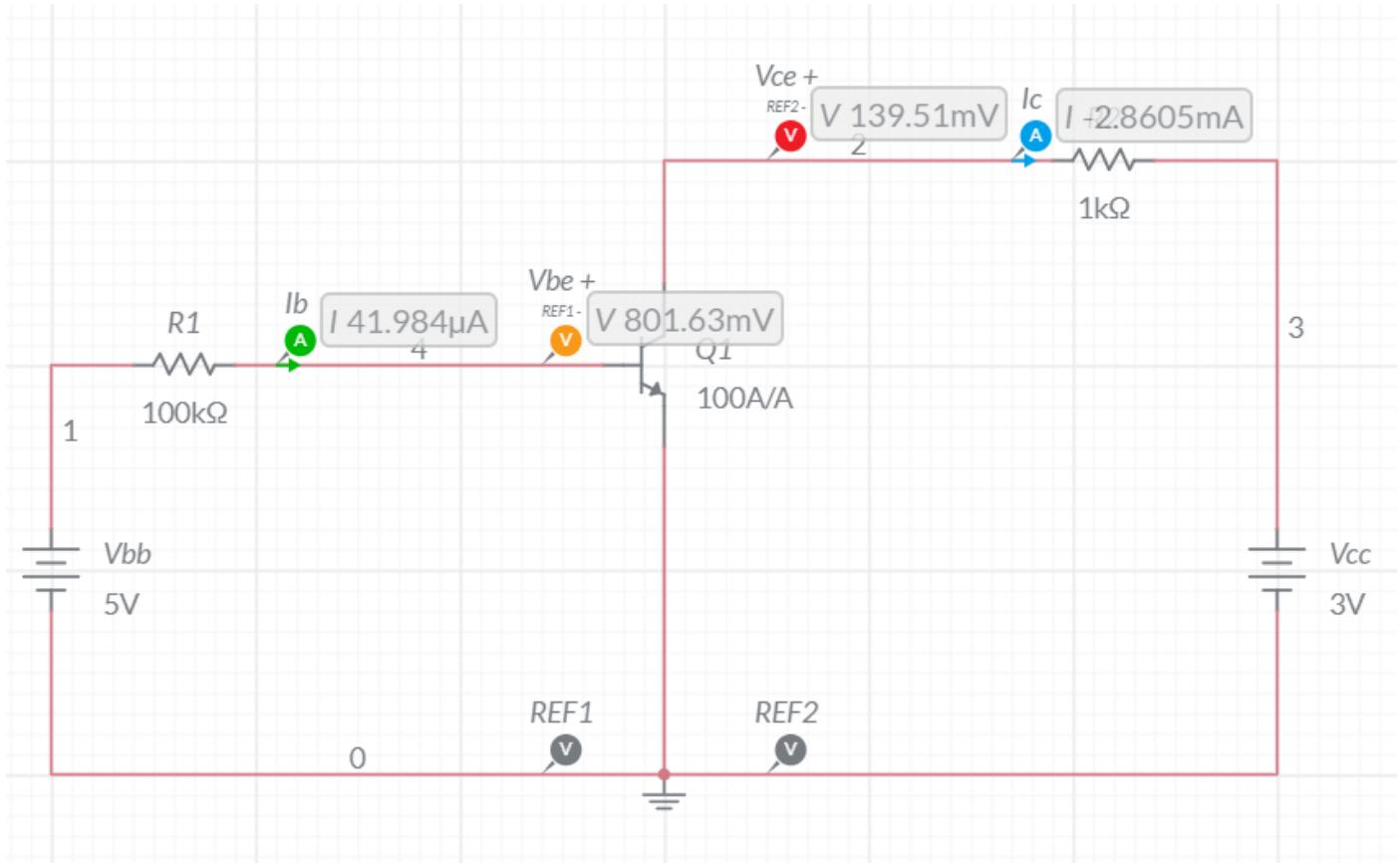
## PROCEDURE

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)



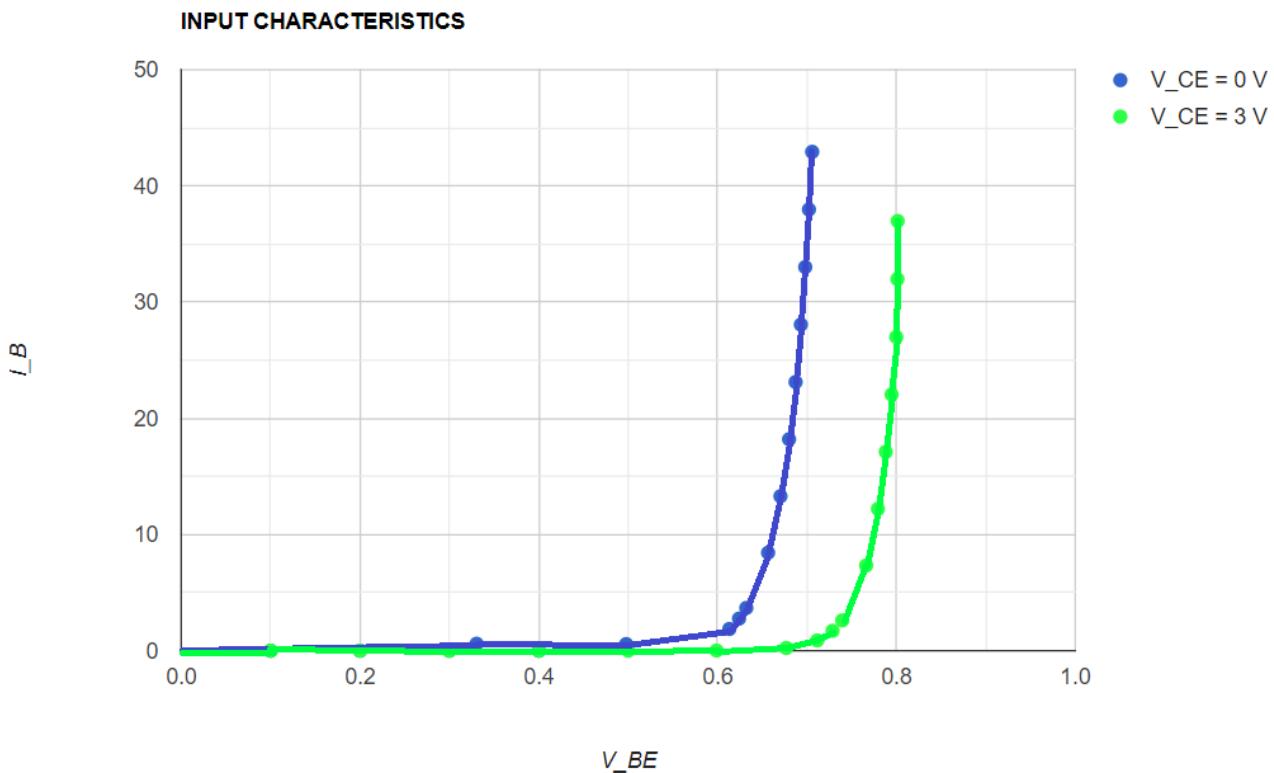


**OBSERVATION TABLE**

$V_{BB}$	$V_{CE} = 0V$		$V_{CE} = 3V$	
	$V_{BE}$ (IN VOLTS)	$I_B$ (IN $\mu A$ )	$V_{BE}$ (IN VOLTS)	$I_B$ (IN $\mu A$ )
0	0	0	0	0
0.1	0.1	0	0.1	0
0.2	0.2	0	0.2	0
0.3	0.3	0	0.3	0
0.4	0.39995	0	0.4	0
0.5	0.49770	0.22965	0.49998	0
0.6	0.56696	0.33038	0.59886	0.011367
0.7	0.597	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	0.63211	3.6789	0.73942	2.6058
1.5	0.65648	8.4352	0.76620	7.3380
2	0.67033	13.297	0.77936	12.206
2.5	0.67995	18.200	0.78811	17.119
3	0.68727	23.127	0.79466	22.053
3.5	0.69314	28.069	0.79988	27.001
4	0.69801	33.020	0.80116	31.988
4.5	0.70218	37.978	0.80144	36.986
5	0.70580	42.942	0.80163	41.984



## GRAPH



## CALCULATIONS

(A) Calculation For Input Impedance	[U19CS012]
$\text{Input Impedance} = R_i = \frac{\Delta V_{BE}}{\Delta I_B}$ $= \frac{(6.84 - 6.56)}{(23.1 - 8.4)} \times 10^5$ $= \frac{0.31}{14.7} \times 10^5$ $= 2.108 \text{ k}\Omega$	[V <sub>CC</sub> is constant]
$\therefore \text{Reverse Voltage Gain} = \frac{\Delta V_{EB}}{\Delta V_{CE}} = \frac{-(6.9 - 7.98)}{3} \times 10^{-1} = 0.36 \times 10^{-1}$ $= 3.6 \times 10^{-2}$	

Input impedance = hie =  $R_i = \Delta V_{BE} / \Delta I_B$  ( $V_{CE}$  = constant) = **2.108 kΩ**

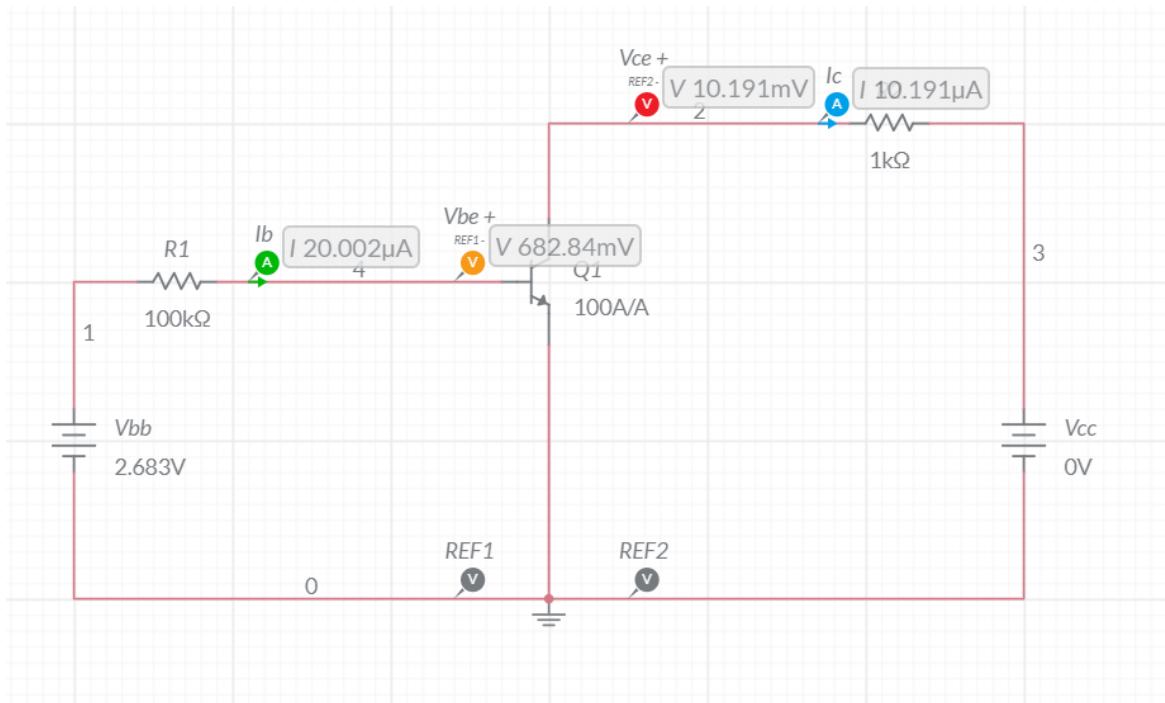
Reverse voltage gain = hre =  $\Delta V_{EB} / \Delta V_{CE}$  ( $I_B$  = constant) =  **$3.6 \times 10^{-2}$**

## OUTPUT CHARACTERISTICS



**CIRCUIT DIAGRAM (FROM MULTISIM)**

**20 UA**



**60 UA**



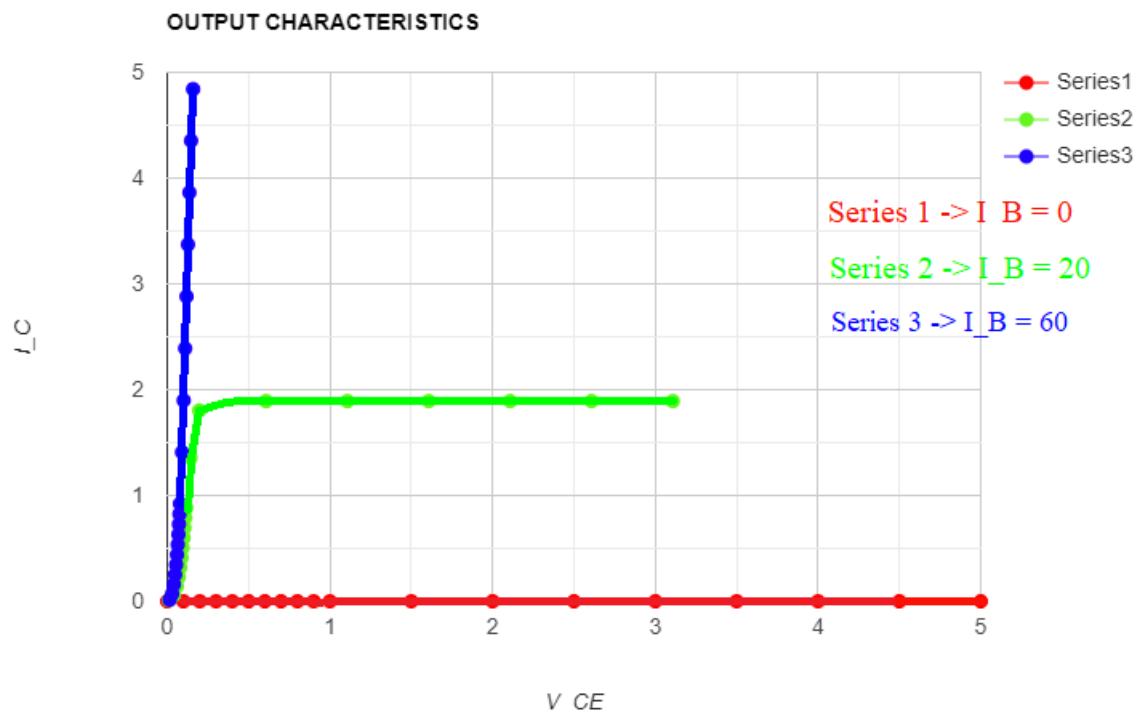
**OBSERVATION TABLE**



V <sub>CC</sub>	I <sub>B</sub> = 0 μA		I <sub>B</sub> = 20 μA		I <sub>B</sub> = 60 μA	
	V <sub>CE</sub> (IN VOLTS)	I <sub>C</sub> (IN MA)	V <sub>CE</sub> (IN VOLTS)	I <sub>C</sub> (IN MA)	V <sub>CE</sub> (IN VOLTS)	I <sub>C</sub> (IN MA)
0	0	0	0.0102	0.0102	0.014528	0.014528
0.1	0.1	0	0.0421	0.0579	0.030123	0.069877
0.2	0.2	0	0.05933	0.14067	0.040552	0.15945
0.3	0.3	0	0.07108	0.22891	0.048373	0.25163
<b>0.4</b>	<b>0.4</b>	<b>0</b>	<b>0.08016</b>	<b>0.31983</b>	<b>0.054647</b>	<b>0.34535</b>
0.5	0.5	0	0.08772	0.41228	0.059905	0.4401
0.6	0.6	0	0.09431	0.5057	0.064448	0.53555
0.7	0.7	0	0.1003	0.59971	0.068464	0.63154
0.8	0.8	0	0.10585	0.69415	0.072076	0.72792
0.9	0.9	0	0.11115	0.78885	0.075369	0.82463
1	1	0	0.1163	0.8837	0.078405	0.92159
1.5	1.5	0	0.14345	1.3566	0.091037	1.4090
2	2	0	0.19702	1.8030	0.10124	1.8988
2.5	2.5	0	0.6077	1.8923	0.11028	2.3897
3	3	0	1.1077	1.8923	0.11882	2.8812
3.5	3.5	0	1.6077	1.8923	0.12736	3.3726
4	4	0	2.1077	1.8923	0.13637	3.8636
4.5	4.5	0	2.6077	1.8923	0.14651	4.3535
5	5	0	3.1077	1.8923	0.15902	4.841



## GRAPH



## CALCULATIONS

(B)	Calculation For Output Impedance	[1119CS012]
	$\text{Output Admittance} = \frac{1}{h_{oe}} = (R_o)^{-1} = \frac{\Delta I_C}{\Delta V_{CE}} = \frac{(5.895 - 4.8)}{(1.1 - 0.15)} \times 10^{-3}$ $= \frac{1.095}{0.95} \times 10^{-3}$ $= 1.1 \times 10^{-3} \text{ siemens}$	
	$\text{Forward Current Gain} = \frac{\Delta I_C}{\Delta I_B} = \frac{(5.895 - 2.105)}{40} \times 10^3$ $= 94.78$	

Output admittance  $1/h_{oe} = (R_o)^{-1} = \Delta I_C / \Delta V_{CE}$  ( $I_B$  is constant) = **1.1 x 10<sup>-3</sup> S**  
 Forward current gain =  $h_{fe} = \Delta I_C / \Delta I_B$  ( $V_{CE}$  = constant) = **94.78**



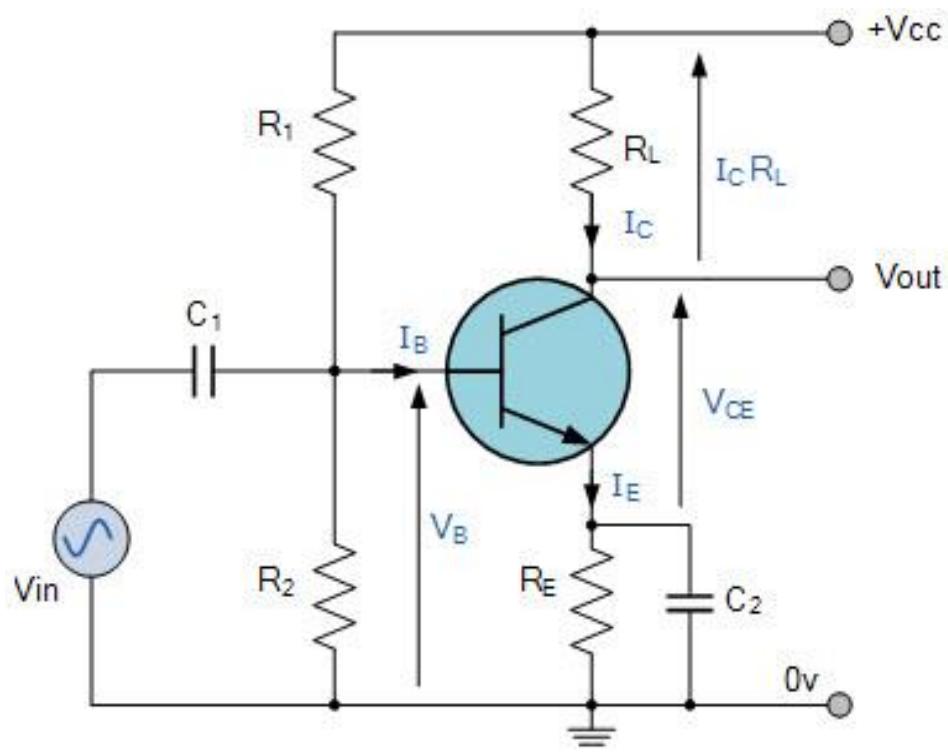
## PART – B

## 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).

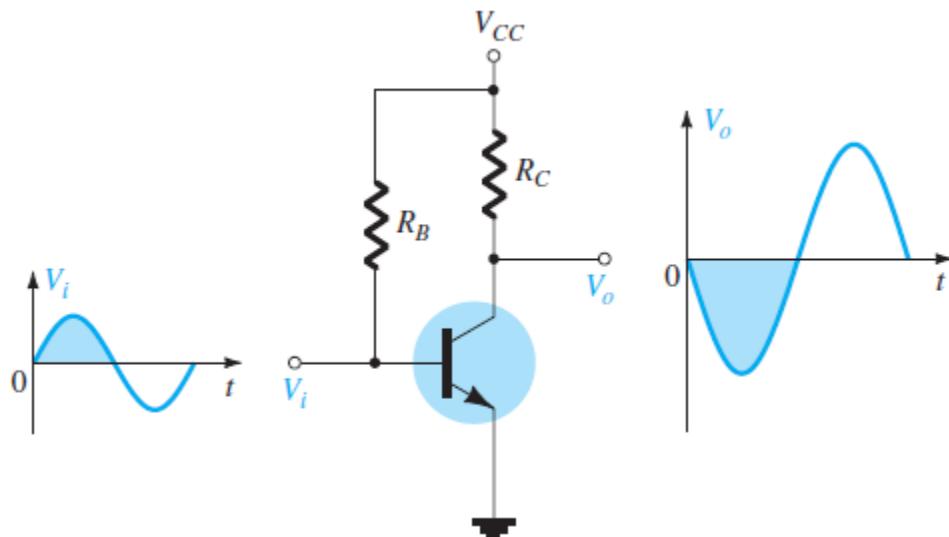
To obtain low distortion when used as an amplifier the operating quiescent point needs to be correctly selected. This is in fact the DC operating point of the amplifier and its position may be established at any point along the load line by a suitable biasing arrangement. The best possible position for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



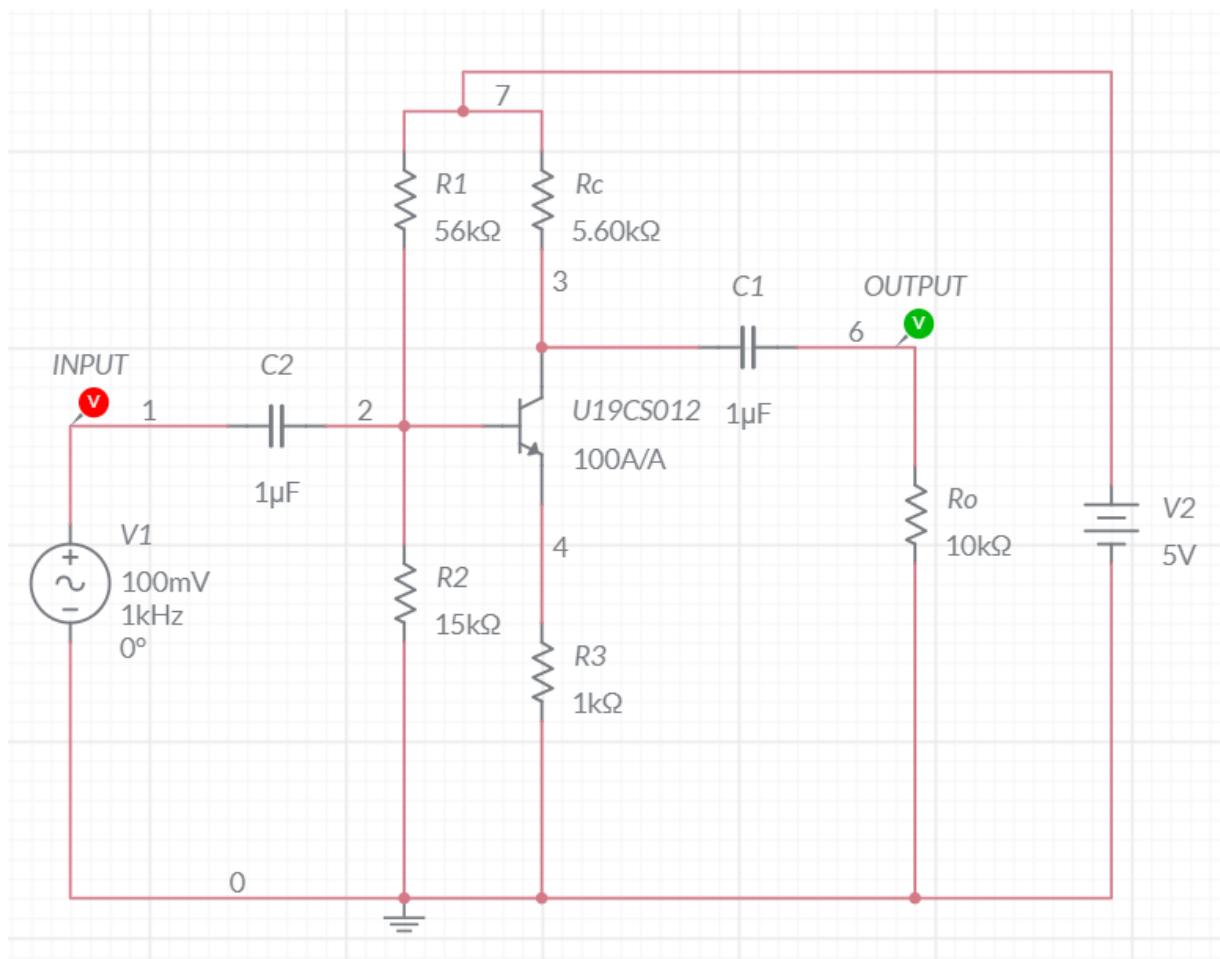
Common Emitter Amplifier Circuit

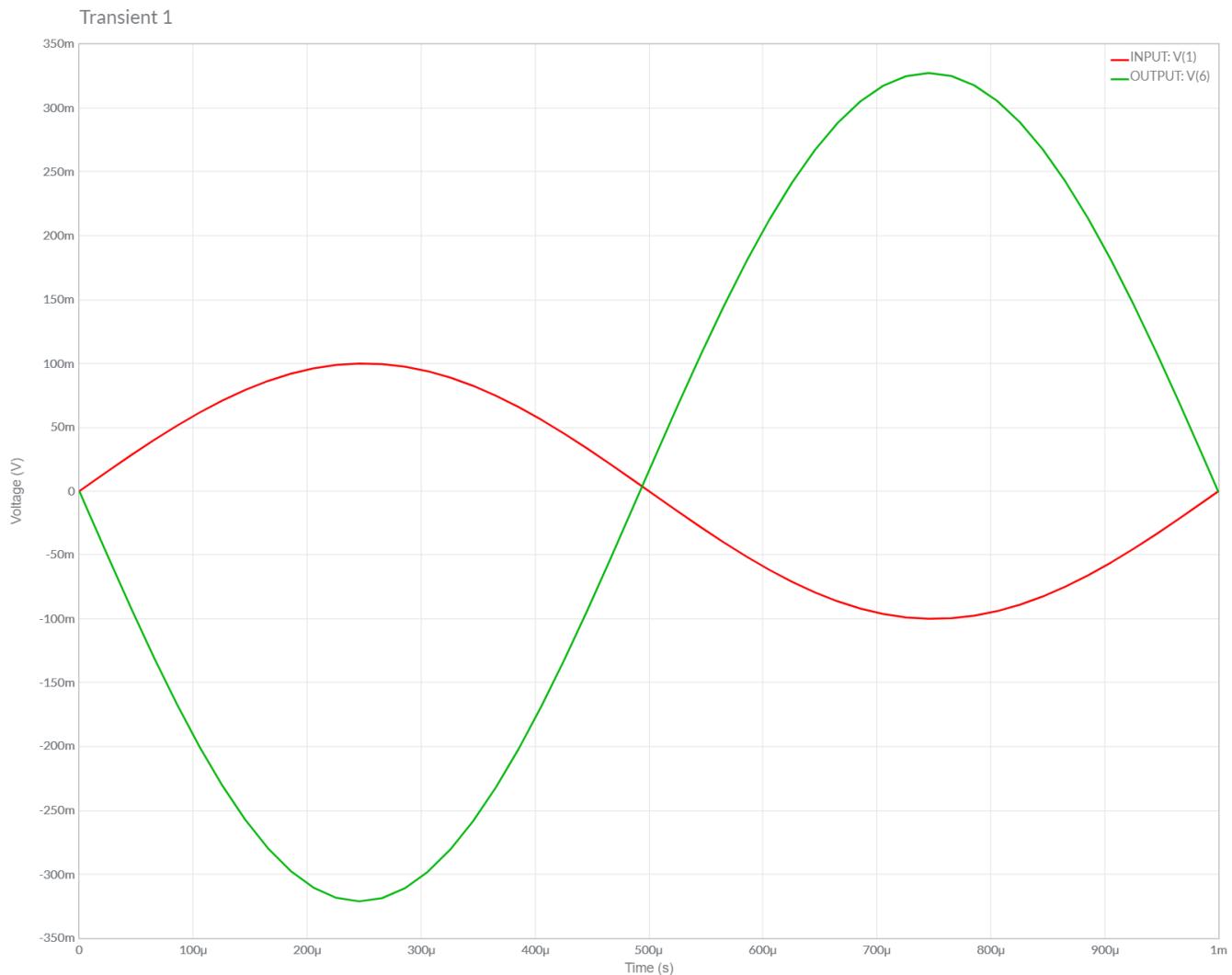
## 180 DEGREE PHASE SHIFT

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**

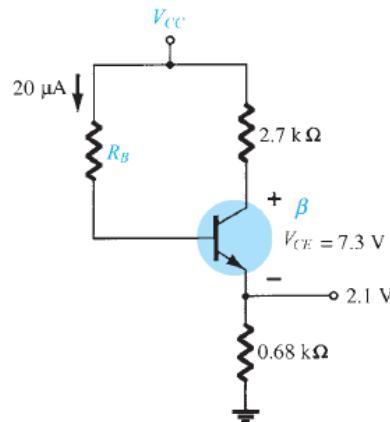
- 1.) In this Experiment, We have studied about Input and Output Characteristics of BJT in Common Emitter Configuration and also implemented it successfully on Multisim.
- 2.) We Verified the Theoretical Knowledge of Input and Output Characteristics of BJT in Common Emitter Configuration by Plotting Input & Output Characteristic Graph.
- 3.) We also Implemented Common Emitter Amplifier and Observed its Input and Output Waveforms.



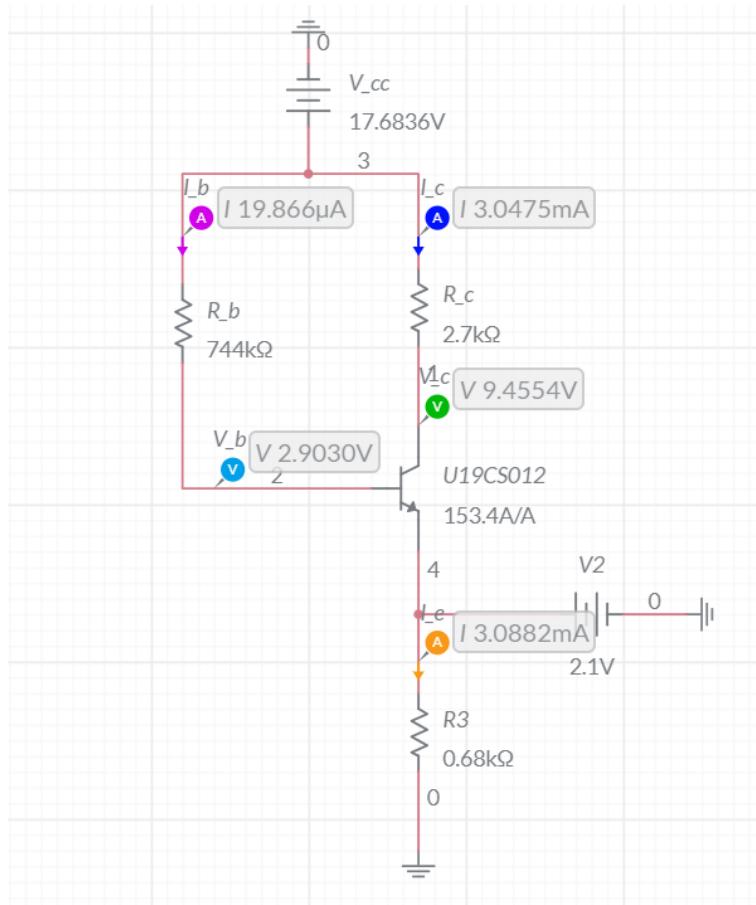
# ASSIGNMENT-10

U19CS012

1.) Using the information provided in the figure below, determine the values of Beta, Vcc and Rb theoretically. Also compute and verify the values of Ic, Vb, and Vc by implementing the circuit on Multisim.

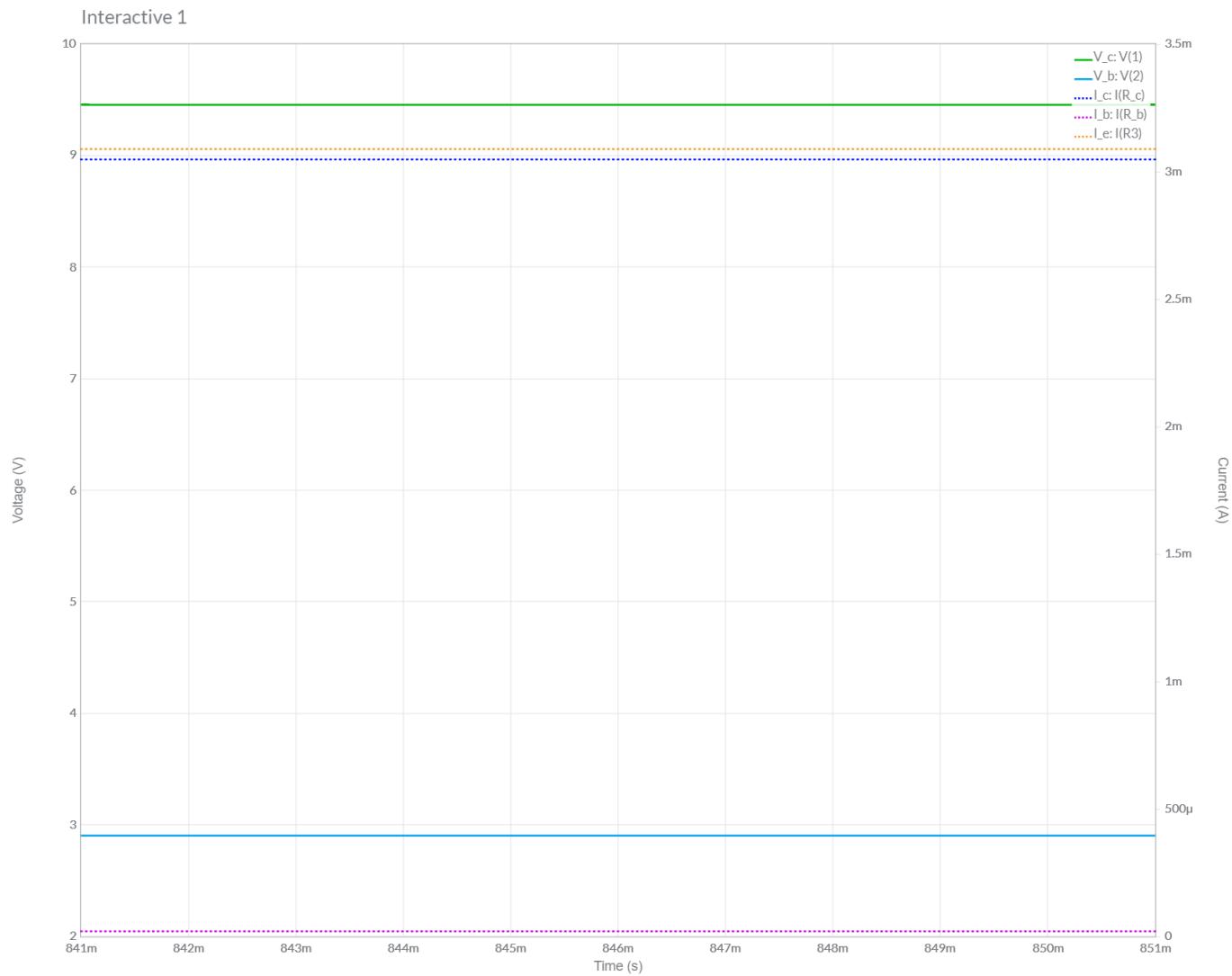


1.) Circuit Image:





## 2.) Grapher Image:

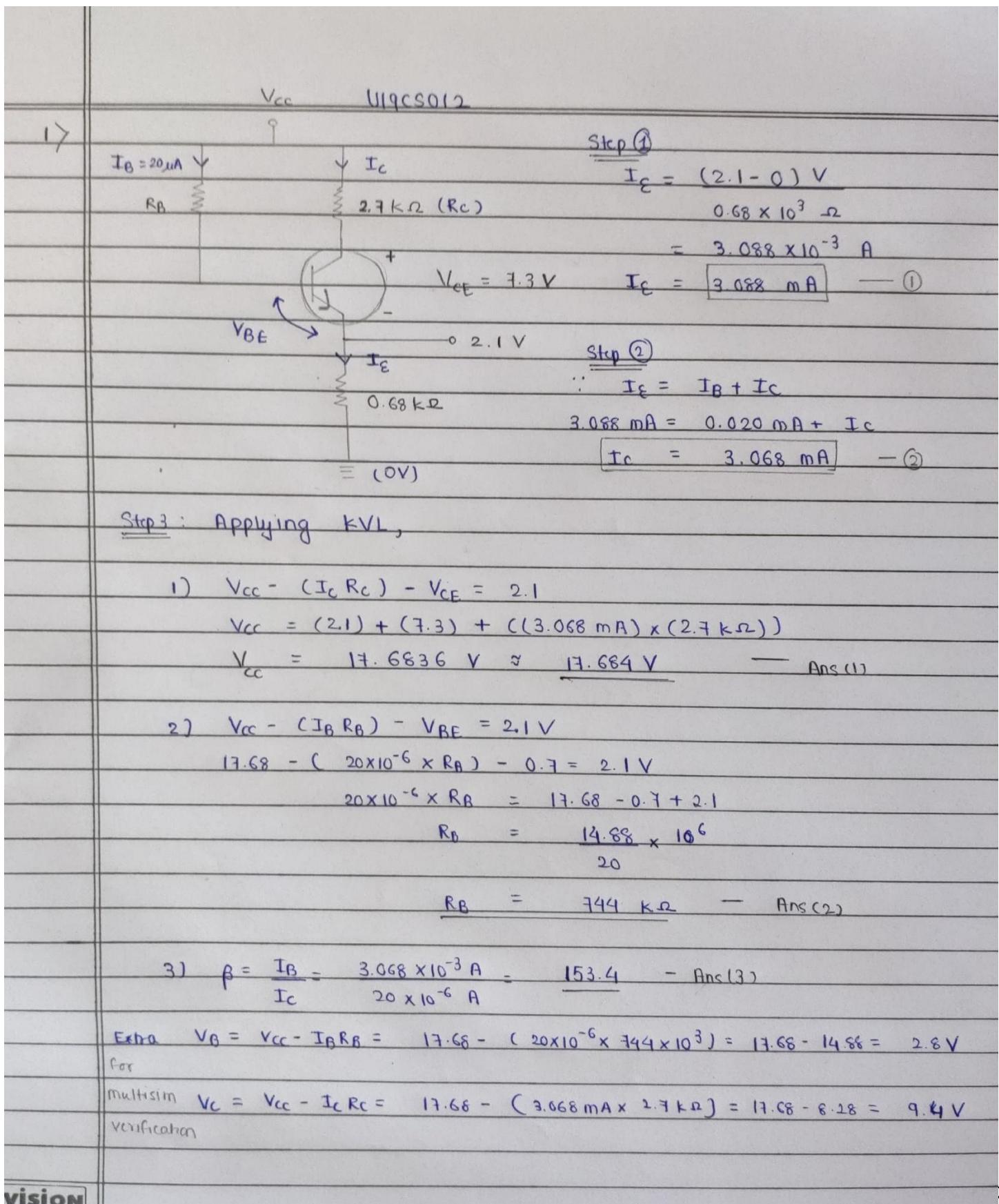


Parameter	Graph	Theoretical
$V_b$	2.9030 V	2.8 V
$V_c$	9.4554 V	9.4 V
$I_b$	0.019866 mA	0.020 mA
$I_c$	3.0475 mA	3.068 mA
$I_e$	3.0882 mA	3.088 mA

We can Clearly See Both the **Theoretical** and **Multisim Values** are *Approximately Equal*. Hence the Experiment was Performed Successfully and Circuit is verified.

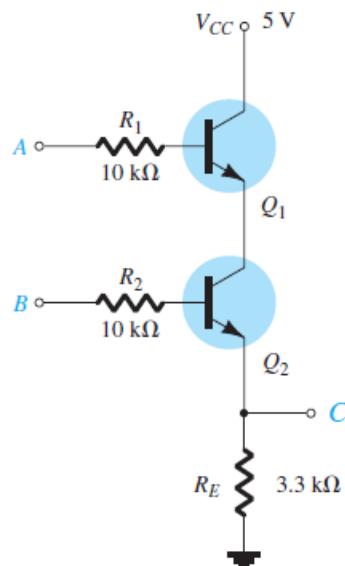


### 3.) Calculations





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 and 55).



Answer:

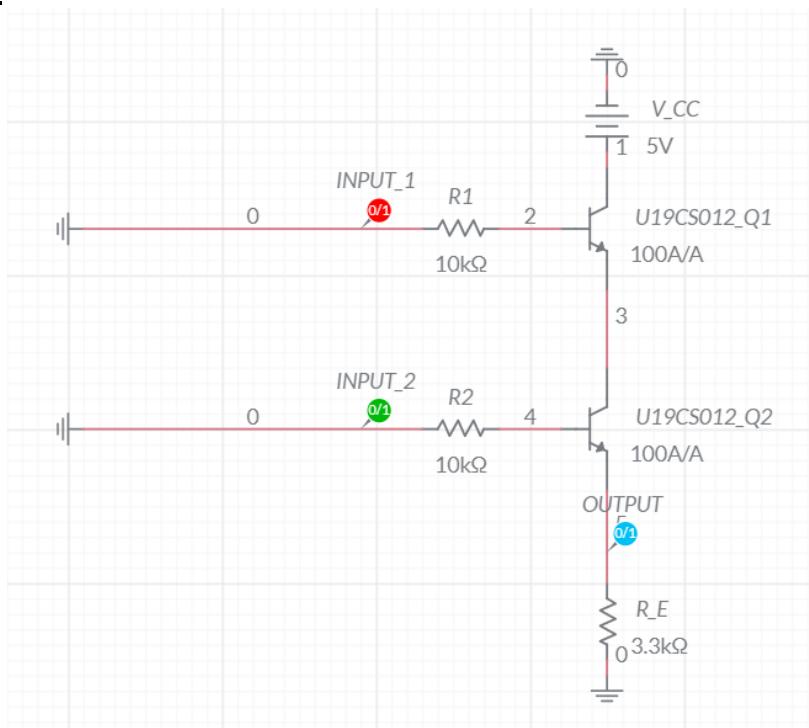
By Observing the Truth Table Obtained from Multisim Simulation, We can clearly see that the Above Circuit [Equivalent to] represents the **Logical AND** Gate.

INPUT1	INPUT2	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1



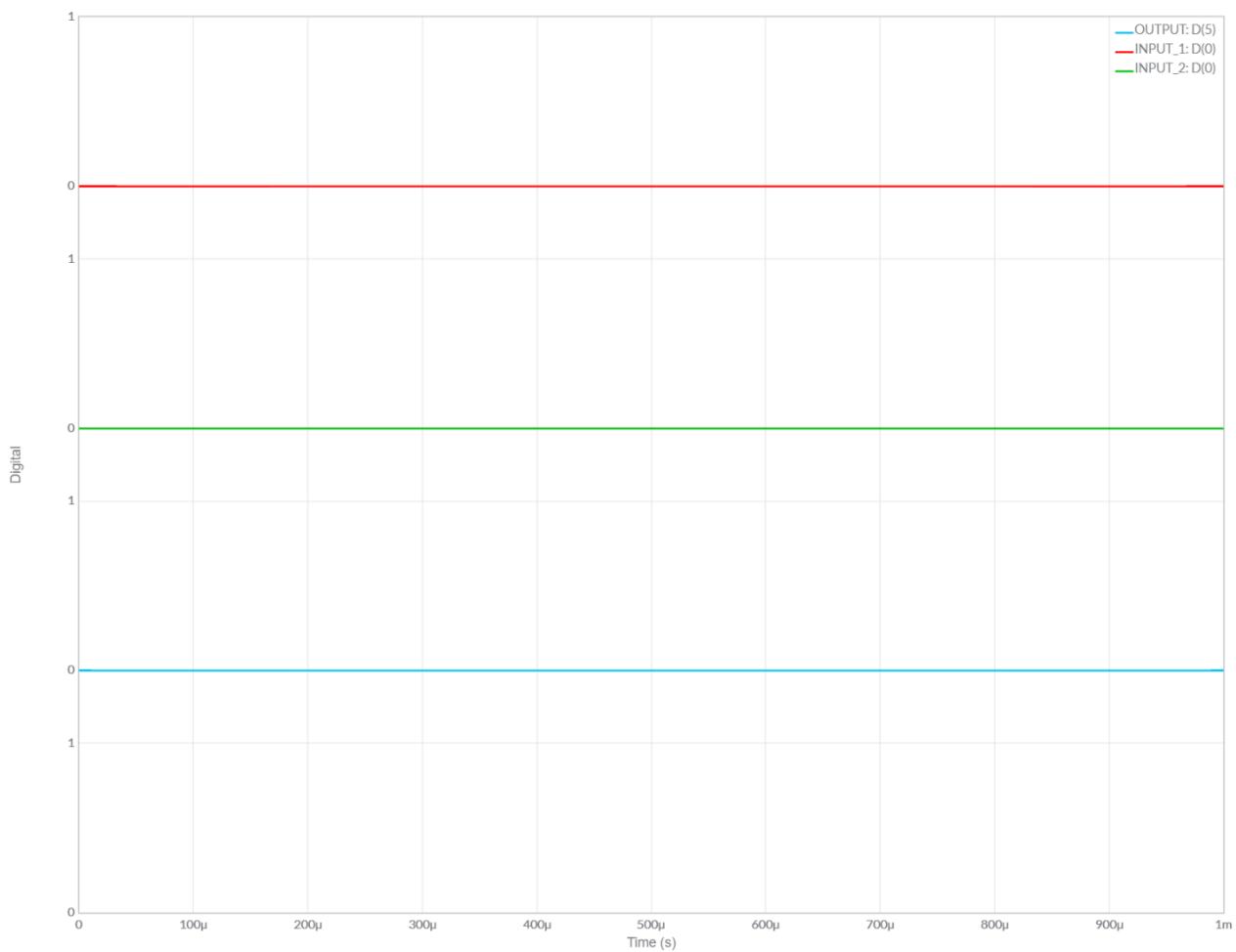
**A.) Case #1: 00**

**1.) Circuit Image:**



**2.) Grapher Image:**

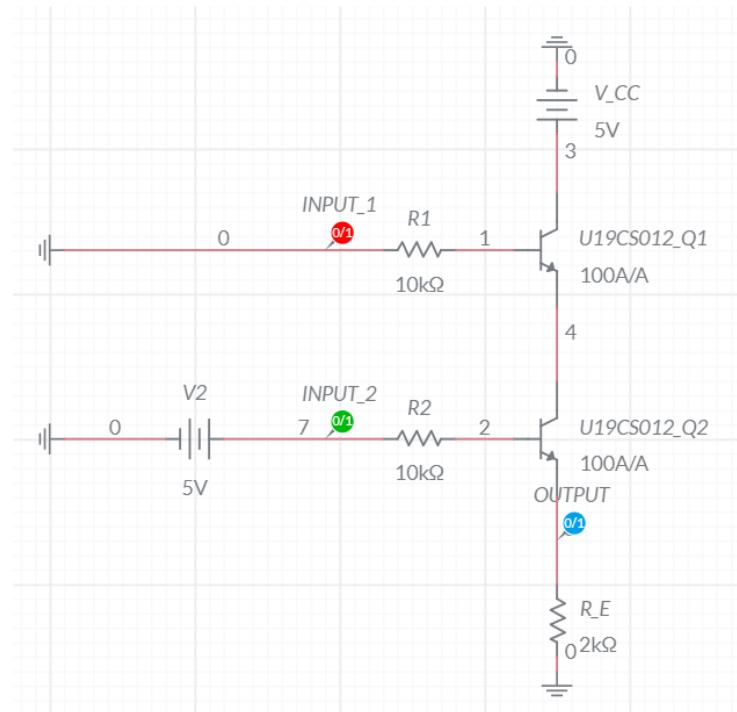
Transient 1



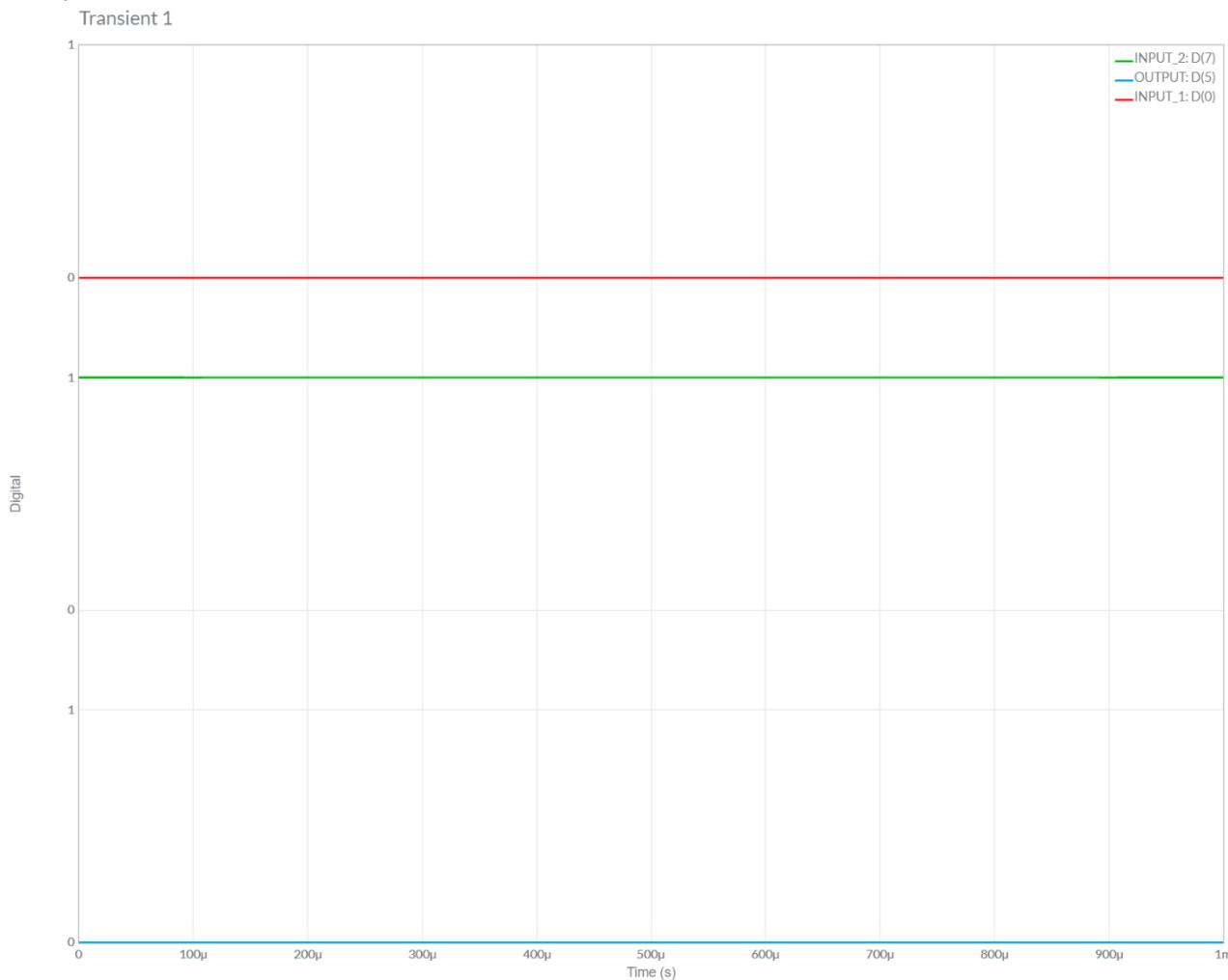


### B.) Case #2: 05

#### 1.) Circuit Image:



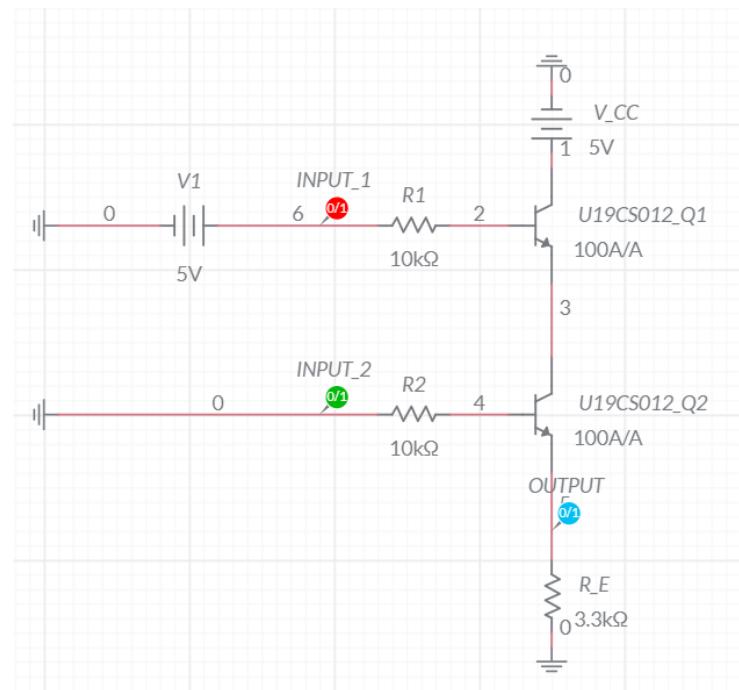
#### 2.) Grapher Image:



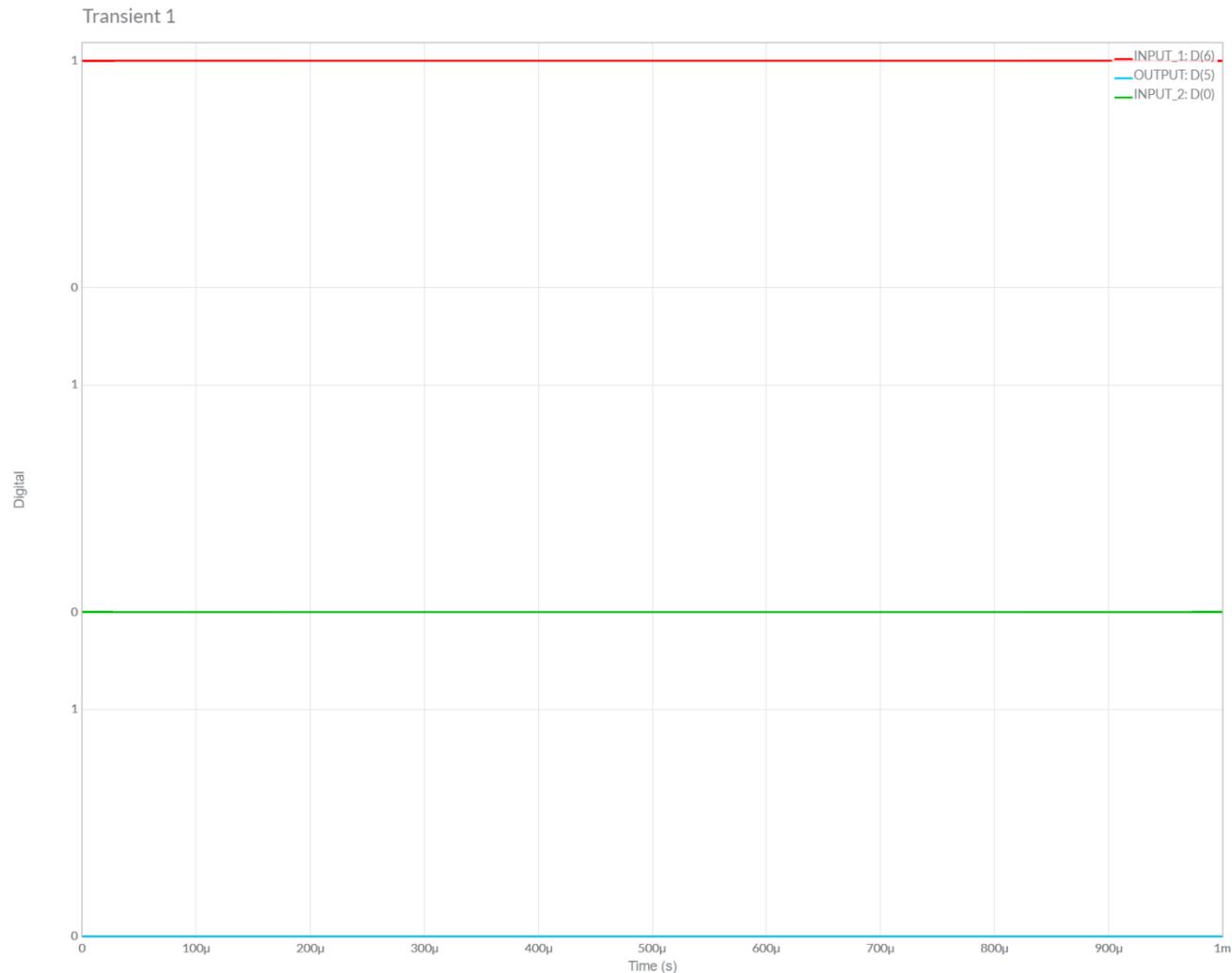


**C.) Case #3: 50**

**1.) Circuit Image:**



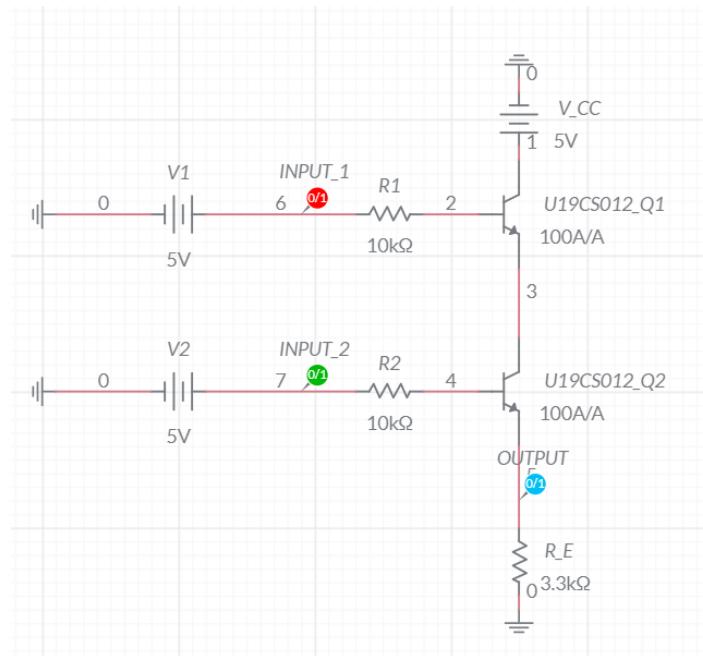
**2.) Grapher Image:**



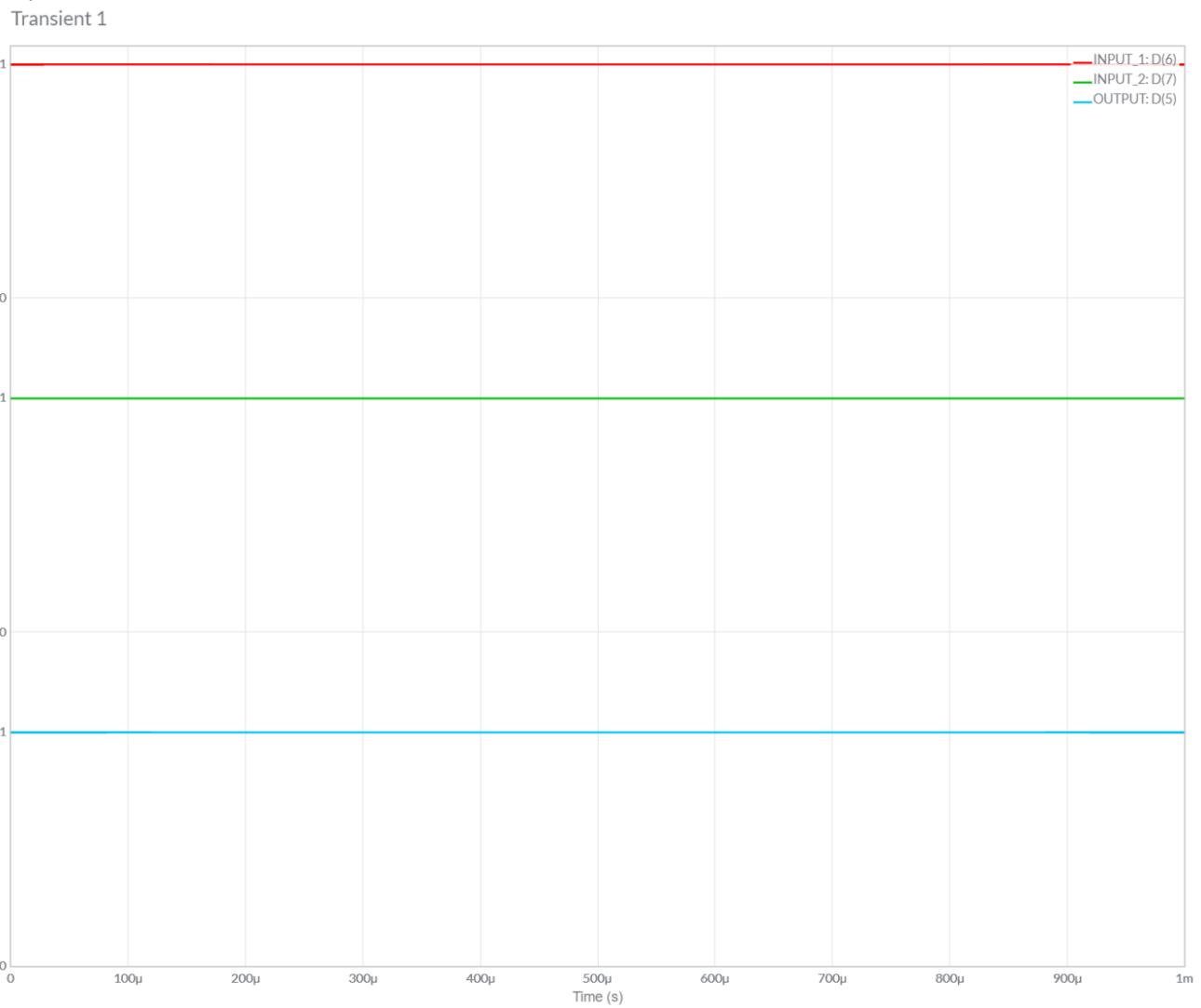


**D.) Case #4: 55**

**1.) Circuit Image:**



**2.) Grapher Image:**





Expt. No:

11

Date:

7/11/2020

## 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.

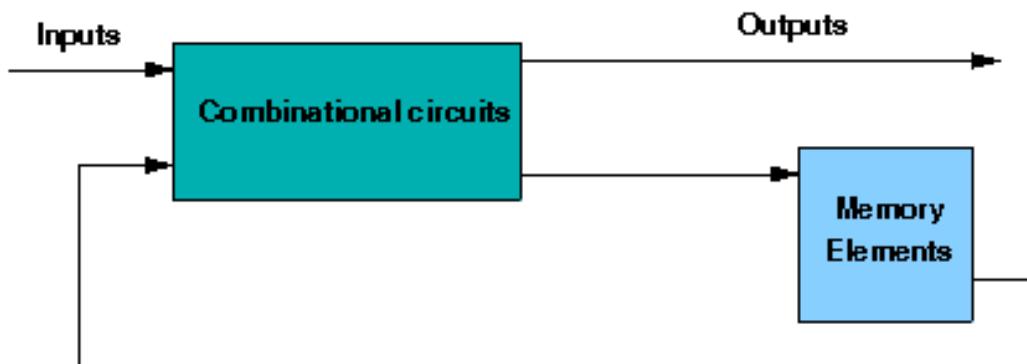
### SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator/Circuit Simulator

### **THEORY:**

In a sequential circuit the present output is determined by both the present input and the past output.

In order to receive the past output some kind of memory element can be used. The memory element commonly used in the sequential circuits are time-delay devices. The block diagram of the sequential circuit-



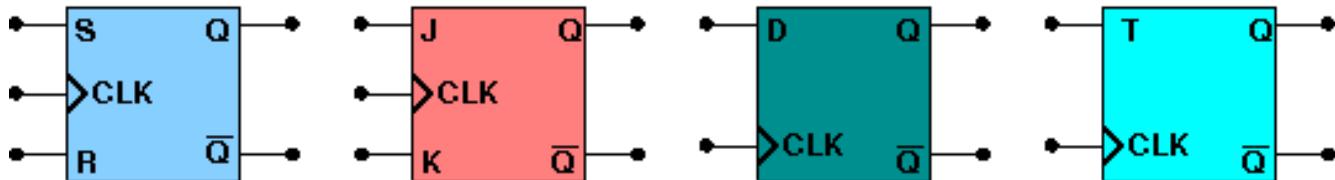
A circuit with flip-flops is considered a *sequential circuit* even in the absence of combinational logic.

Circuits that include flip-flops are usually classified by the function they perform. Two such circuits are registers and counters:

1. **Register** is a group of flip-flops. Its basic function is to hold information within a digital system so as to make it available to the logic units during the computing process.
2. **Counter** is essentially a register that goes through a predetermined sequence of states. There are various different kind of flip-flops. Some of the common flip-flops are: R-S flip-flop, D flip-flop, J-K flip-flop, T flip-flop.



The block diagram of different flip-flops are shown here –



- RS flip-flop if r is high then reset state occurs and when s=1 set state. The both cannot be high simultaneously. This input combination is avoided.
- JK flip-flop if J and K are both low then no change occurs. If J and K are both high at the clock edge then the output will toggle from one state to the other.
- D flip-flop the d flip-flop tracks the input, making transitions with match those of the input d. It is used as data store.
- T flip-flop or "Toggle" flip-flop changes its output on each clock edge,

### I.) REGISTERS:

Flip flops can be used to store *a single bit of binary data* (1 or 0). However, in order to store multiple bits of data, we need multiple flip flops. N flip flops are to be connected in an order to store N bits of data.

**A Register** is a device which is used to store such information.  
It is a group of flip flops connected in series used to store multiple bits of data.

The information stored within these registers can be transferred with the help of **shift registers**.

Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses.

An **n-bit** shift register can be formed by connecting **n** flip-flops where each flip flop stores a single bit of data.

The registers which will shift the bits to left are called "Shift left registers".  
The registers which will shift the bits to right are called "Shift right registers".

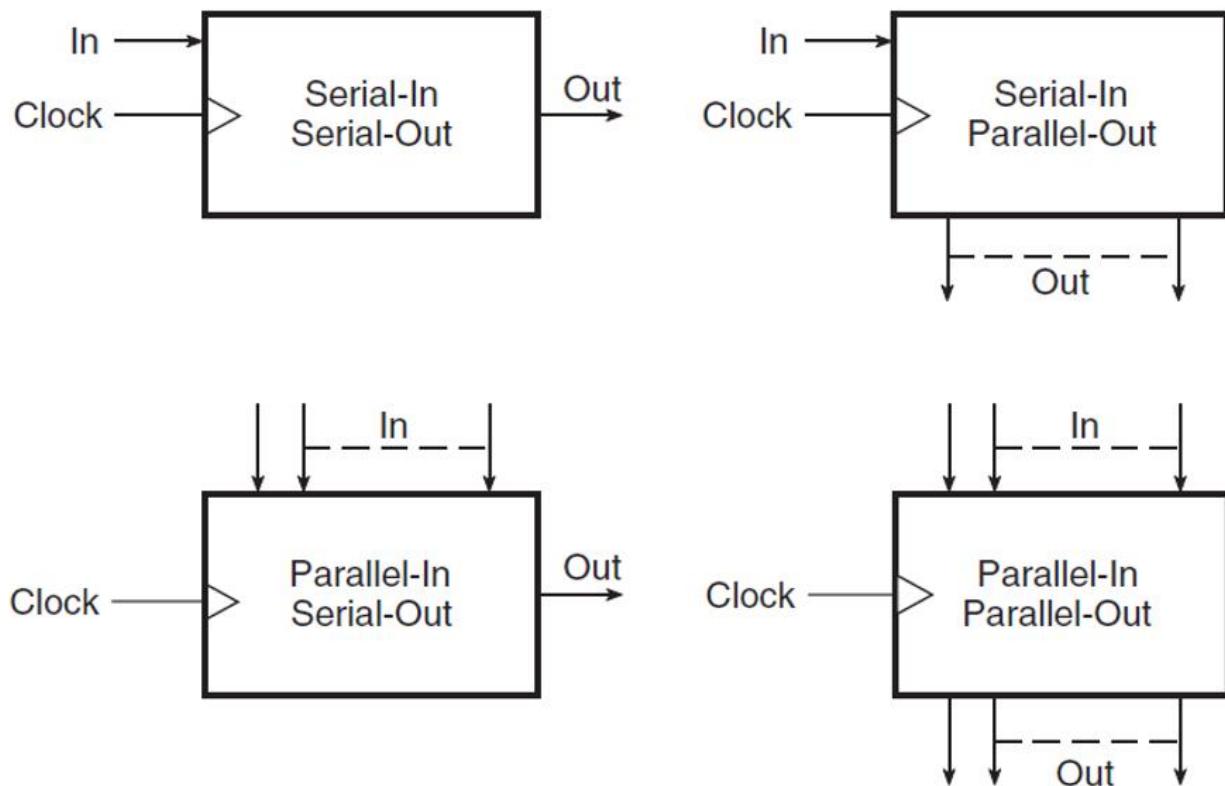


Shift registers are basically of **4** types.

These are:

1. Serial In Serial Out shift register [SISO]
2. Serial In parallel Out shift register [SIPO]
3. Parallel In Serial Out shift register [PISO]
4. Parallel In parallel Out shift register [PIPO]

## Types



### A.) Serial-In Serial-Out Shift Register (SISO)

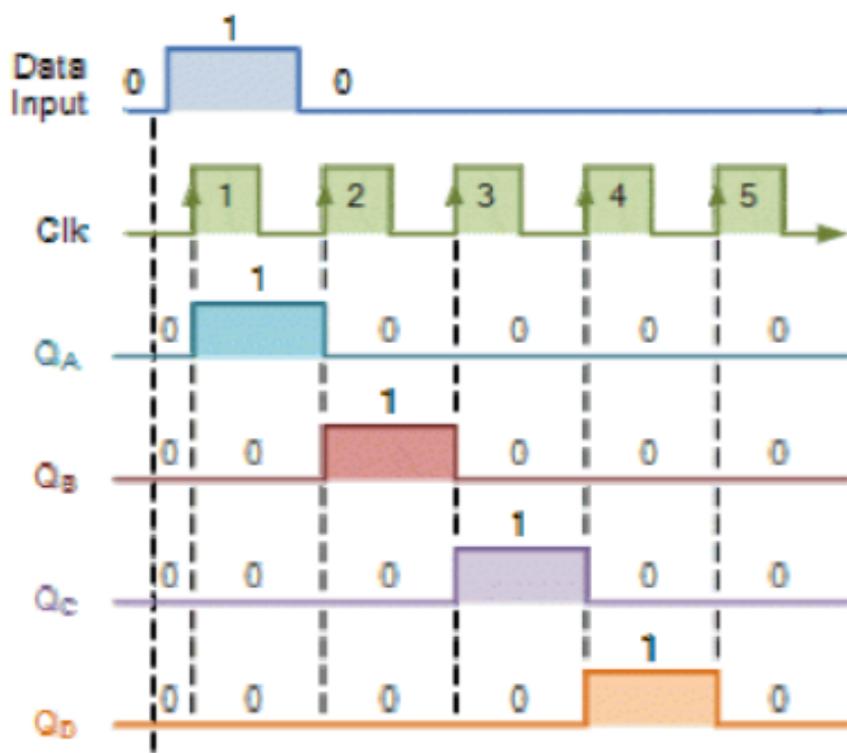
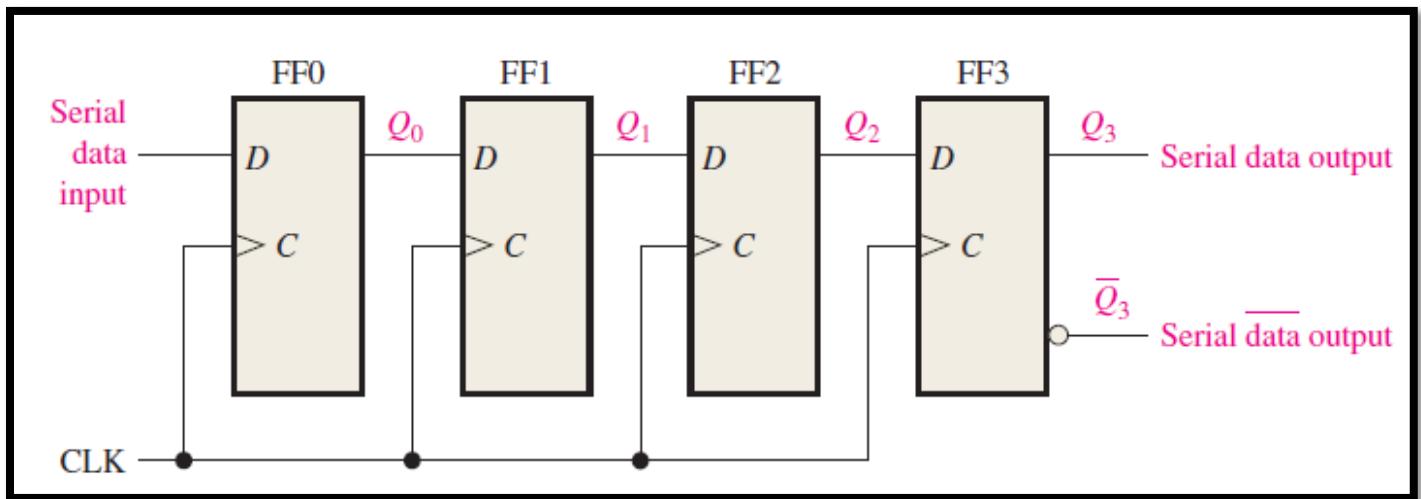
The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as *Serial-In Serial-Out shift register*.

Since there is only **one** output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift Register.

The logic circuit given below shows a serial-in serial-out shift register.

The circuit consists of *four D flip-flops* which are connected in a serial manner.

All these flip-flops are synchronous with each other since the same clock signal is applied to each flip flop.



The timing diagram of data shift through a 4-bit SISO shift register



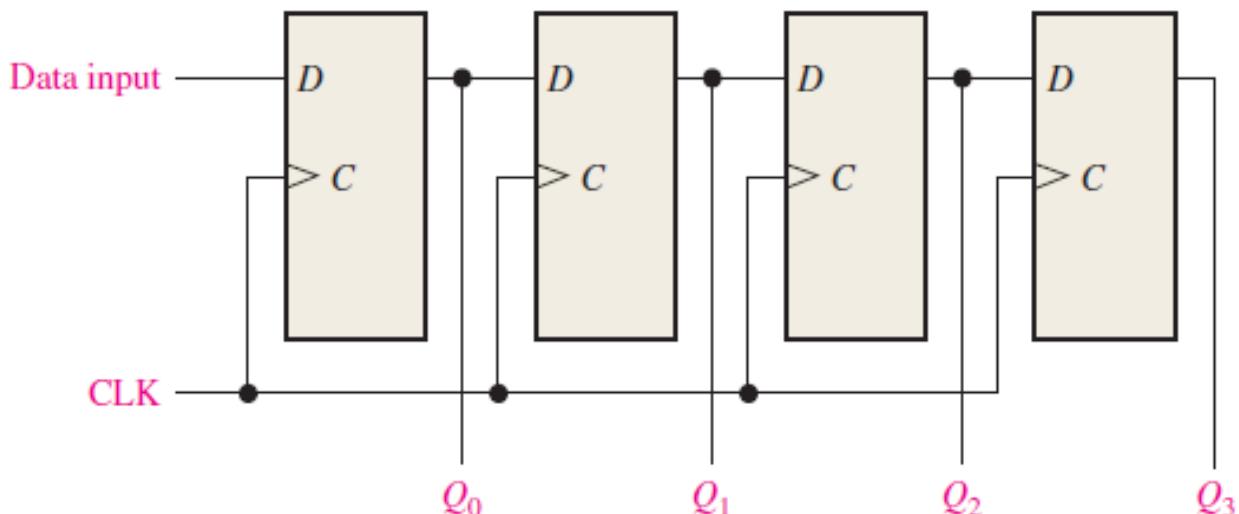
### B.) Serial-In Parallel-Out shift Register (SIPO)

The shift register, which allows serial input (one bit after the other through a single data line) and produces a parallel output is known as *Serial-In Parallel-Out* shift register.

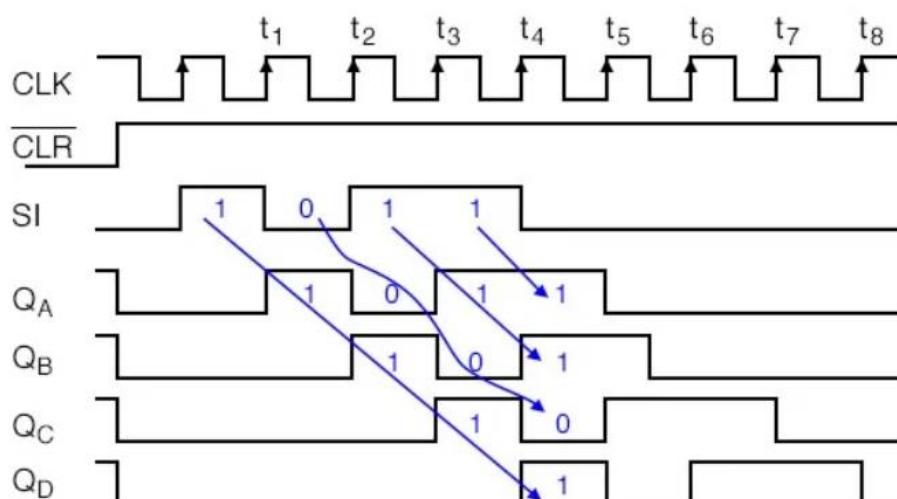
The logic circuit given below shows a serial-in-parallel-out shift register. The circuit consists of four D flip-flops which are connected.

The clear (CLR) signal is connected in addition to the clock signal to all the 4 flip flops in order to RESET them.

The output of the first flip flop is connected to the input of the next flip flop and so on. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip flop.



**SERIAL IN PARALLEL OUT [S.I.P.O.]**



Serial-in/ parallel-out shift register waveforms



### C.) Parallel-In Serial-Out Shift Register (PISO)

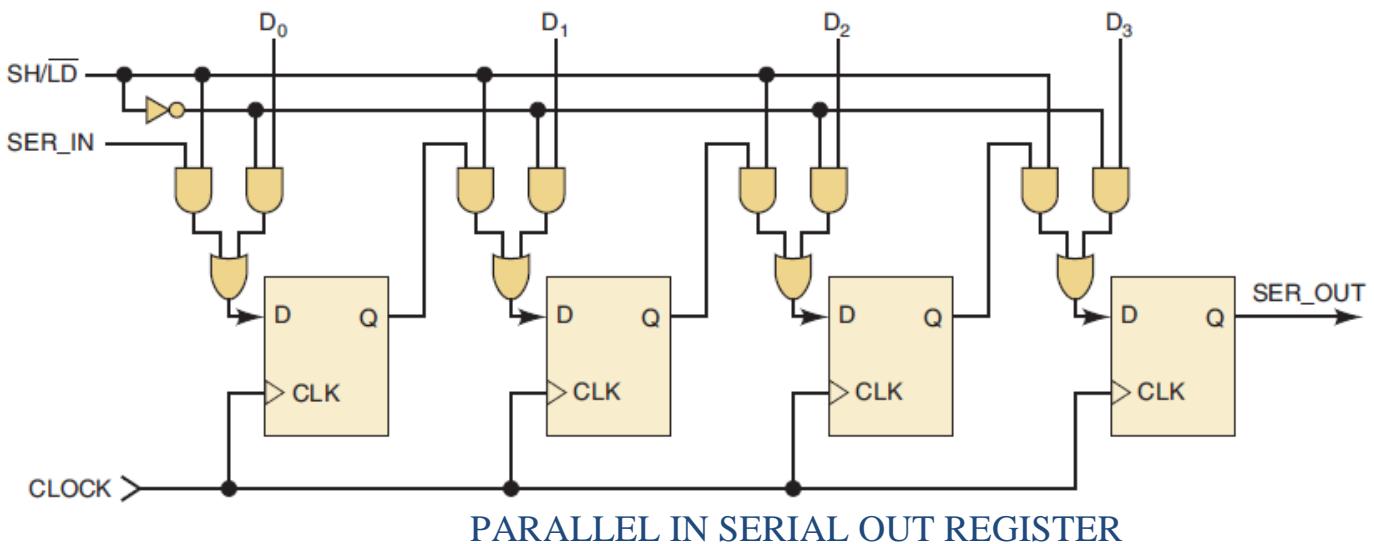
The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and produces a serial output is known as *Parallel-In Serial-Out* shift register.

The logic circuit given below shows a parallel-in-serial-out shift register.

The circuit consists of four D flip-flops which are connected.

The clock input is *directly* connected to all the flip flops but the input data is connected individually to each flip flop through a multiplexer at the input of every flip flop.

The output of the previous flip flop and parallel data input are connected to the input of the MUX and the output of MUX is connected to the next flip flop. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip flop.



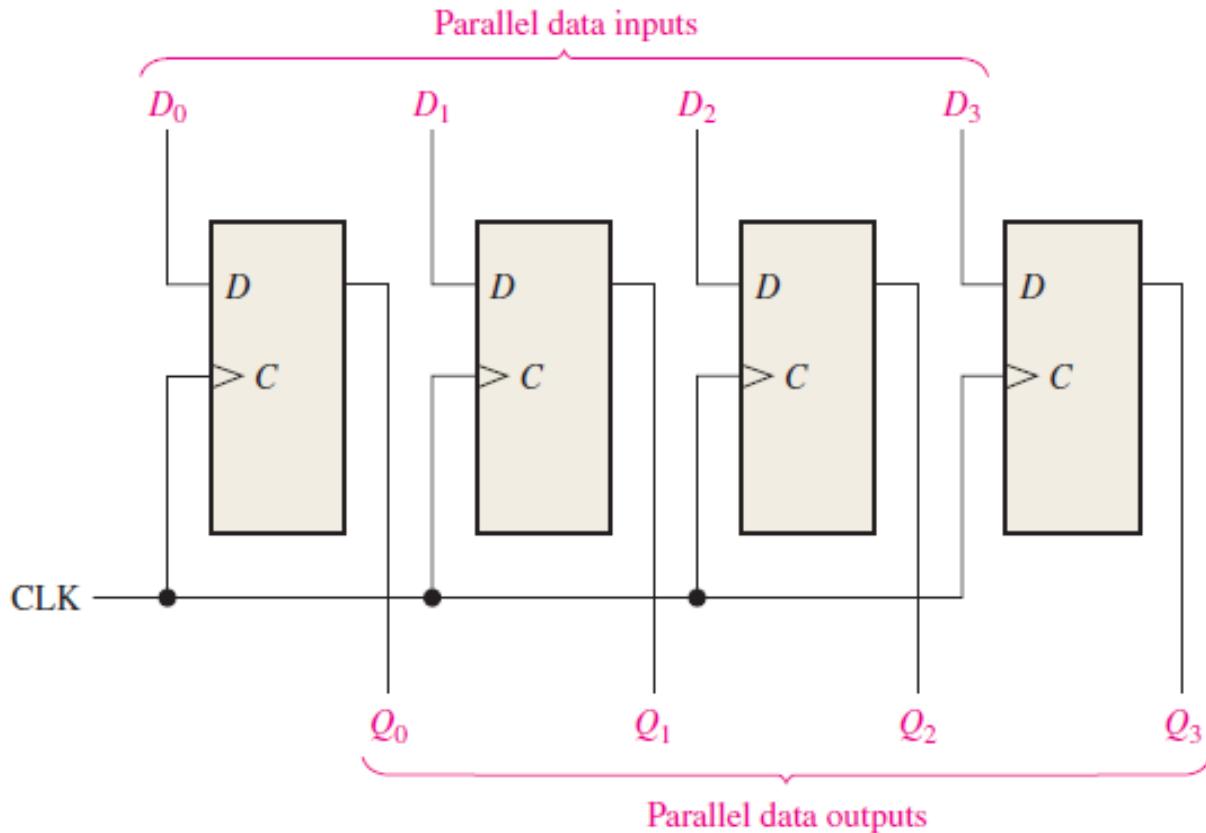
### D.) Parallel-In Parallel-Out Shift Register (PIPO)

The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and also produces a parallel output is known as *Parallel-In parallel-Out* shift register.

The logic circuit given below shows a parallel-in-parallel-out shift register.

The circuit consists of four D flip-flops which are connected. The clear (CLR) signal and clock signals are connected to all the 4 flip flops.

In this type of register, there are no interconnections between the individual flip-flops since no serial shifting of the data is required. Data is given as input separately for each flip flop and in the same way, output also collected individually from each flip flop.



### COUNTER:

**Counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal.

Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit. For example, in *UP counter* a counter increases count for every rising edge of clock. Not only counting, a counter can follow the certain sequence based on our design like any random sequence 0, 1, 3, 2....They can also be designed with the help of flip flops.

### Modulus:

To determine the number of flip-flops 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  $16 = 2^4$ .

$$\text{Modulus} \geq 2^N$$

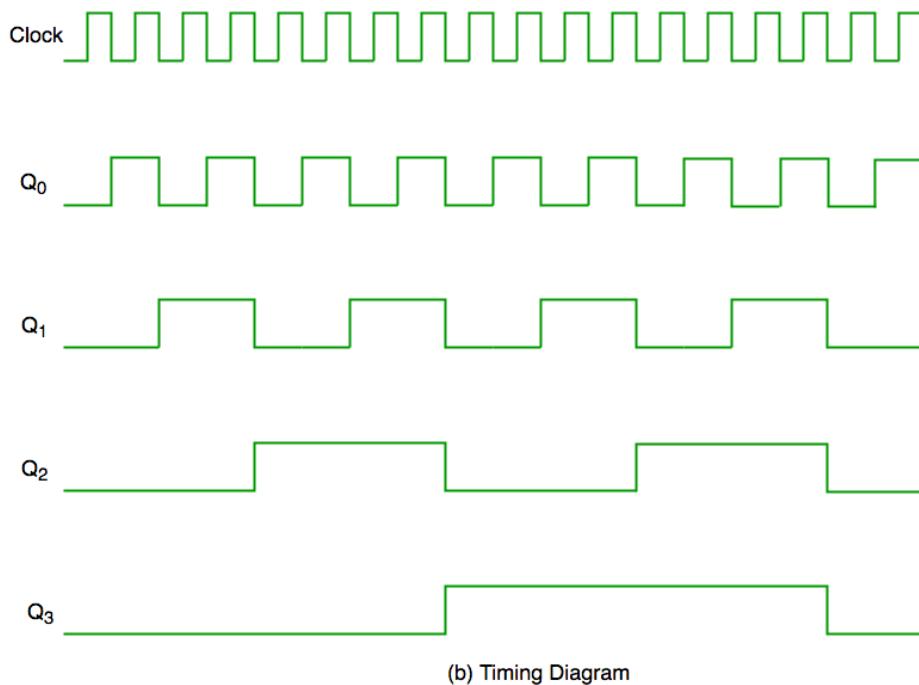
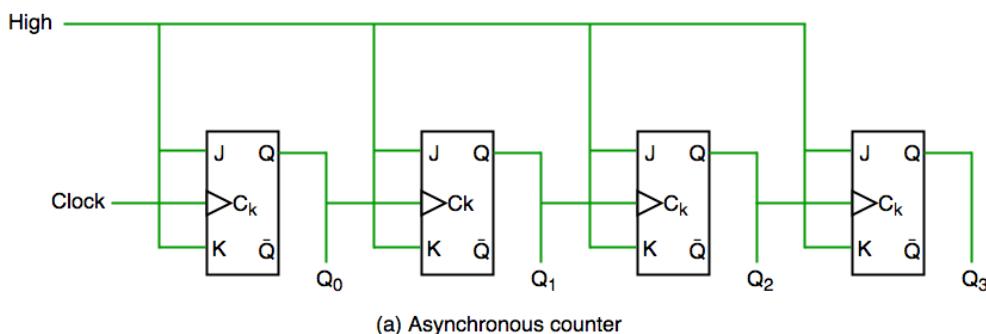


Counters are broadly divided into two categories

1. Asynchronous counter
2. Synchronous counter

### 1. Asynchronous Counter

In asynchronous counter we don't use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following flip flop is driven by output of previous flip flops. We can understand it by following diagram-



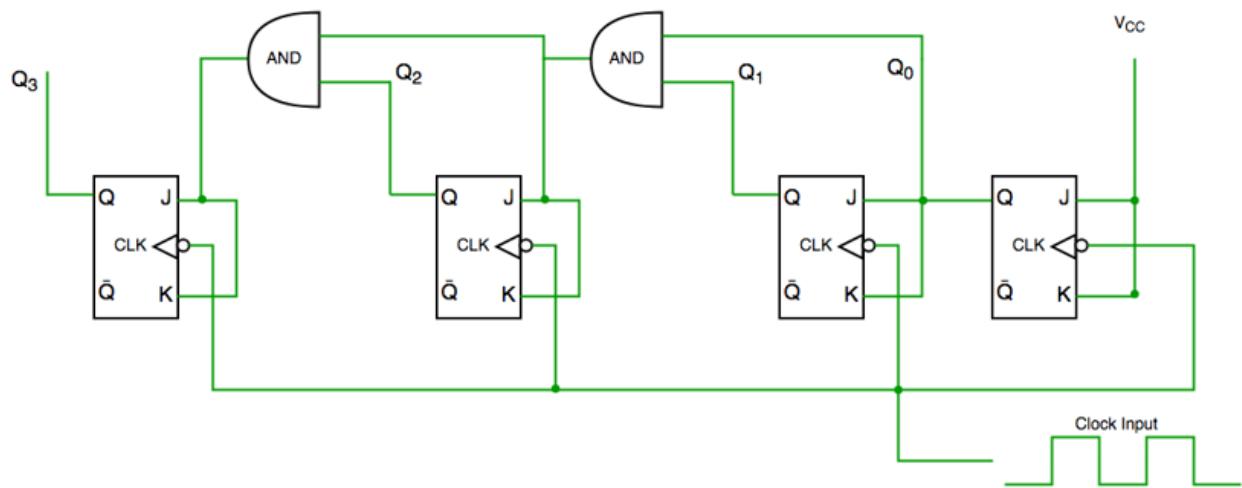
### ASYNCHRONOUS COUNTER AND ITS TIMING DIAGRAM

It is evident from timing diagram that  $Q_0$  is changing as soon as the rising edge of clock pulse is encountered,  $Q_1$  is changing when rising edge of  $Q_0$  is encountered (because  $Q_0$  is like clock pulse for second flip flop) and so on. In this way ripples are generated through  $Q_0$ ,  $Q_1$ ,  $Q_2$ ,  $Q_3$  hence it is also called **Ripple counter**.

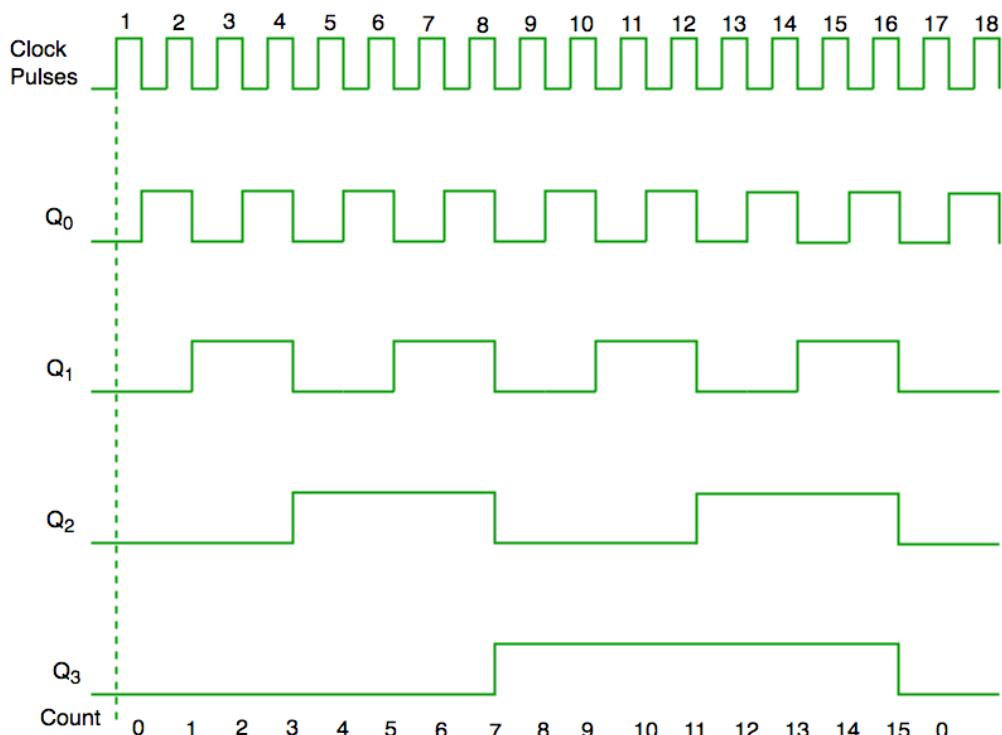


## 2. Synchronous Counter

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop.



**SYNCHRONOUS COUNTER**



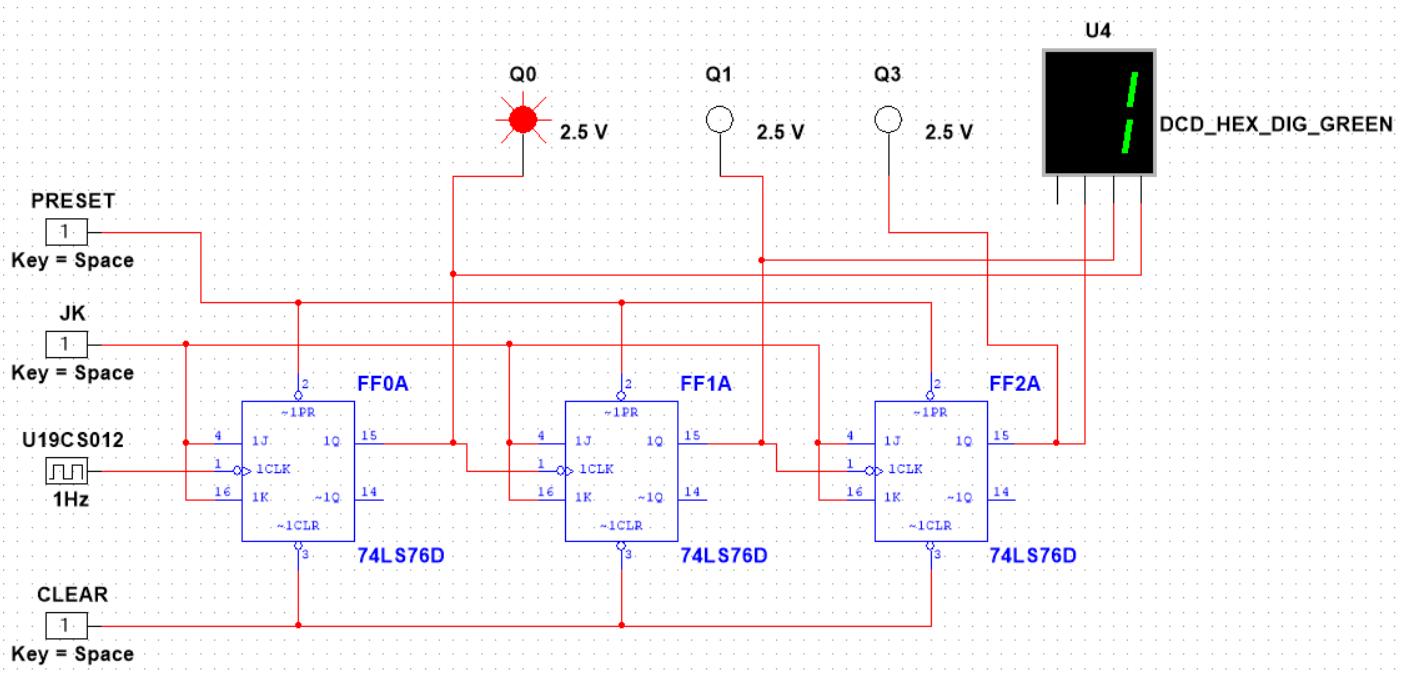
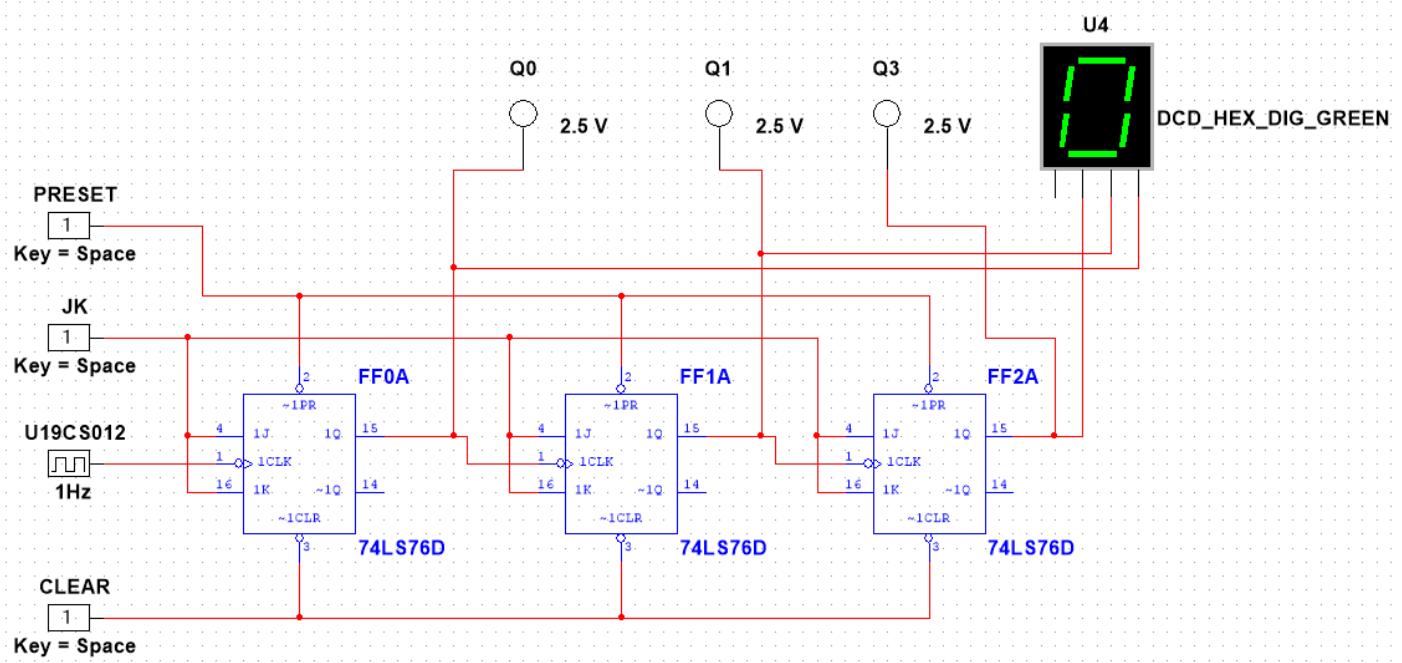
**TIMING DIAGRAM**

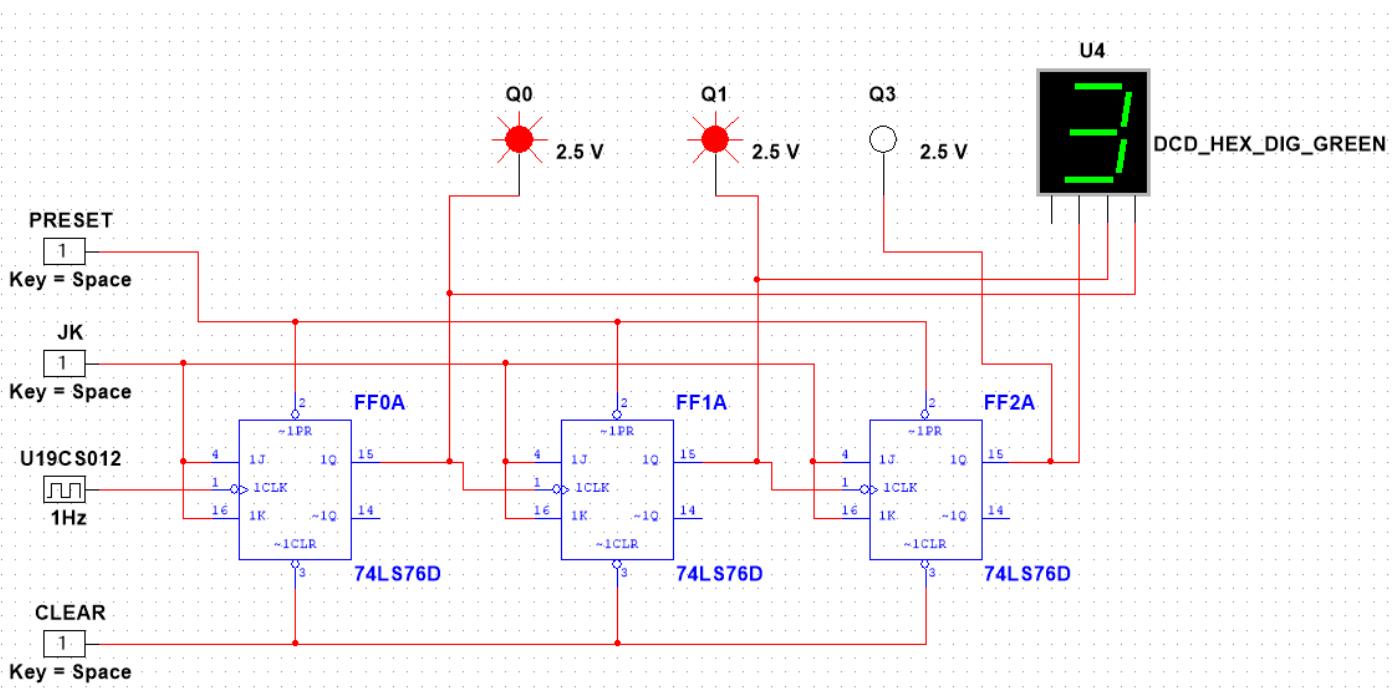
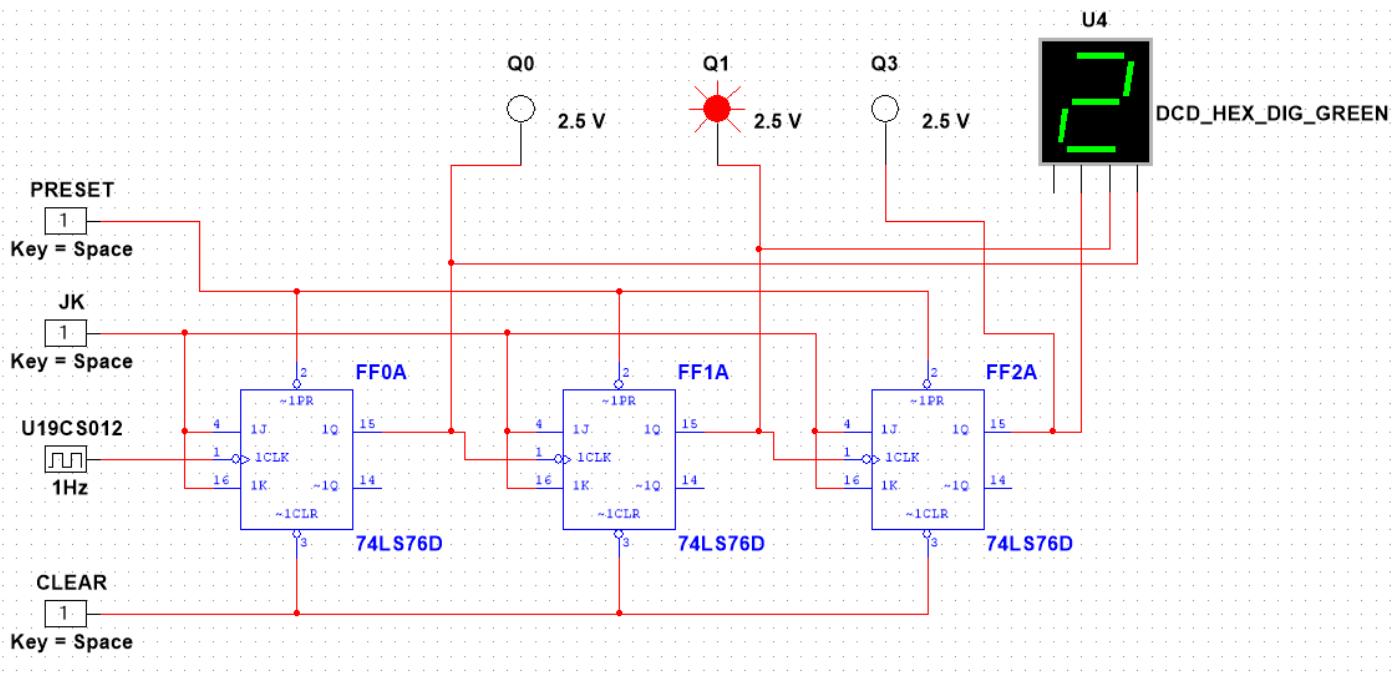
From circuit diagram we see that Q0 bit gives response to each falling edge of clock while Q1 is dependent on Q0, Q2 is dependent on Q1 and Q0, Q3 is dependent on Q2, Q1 and Q0.

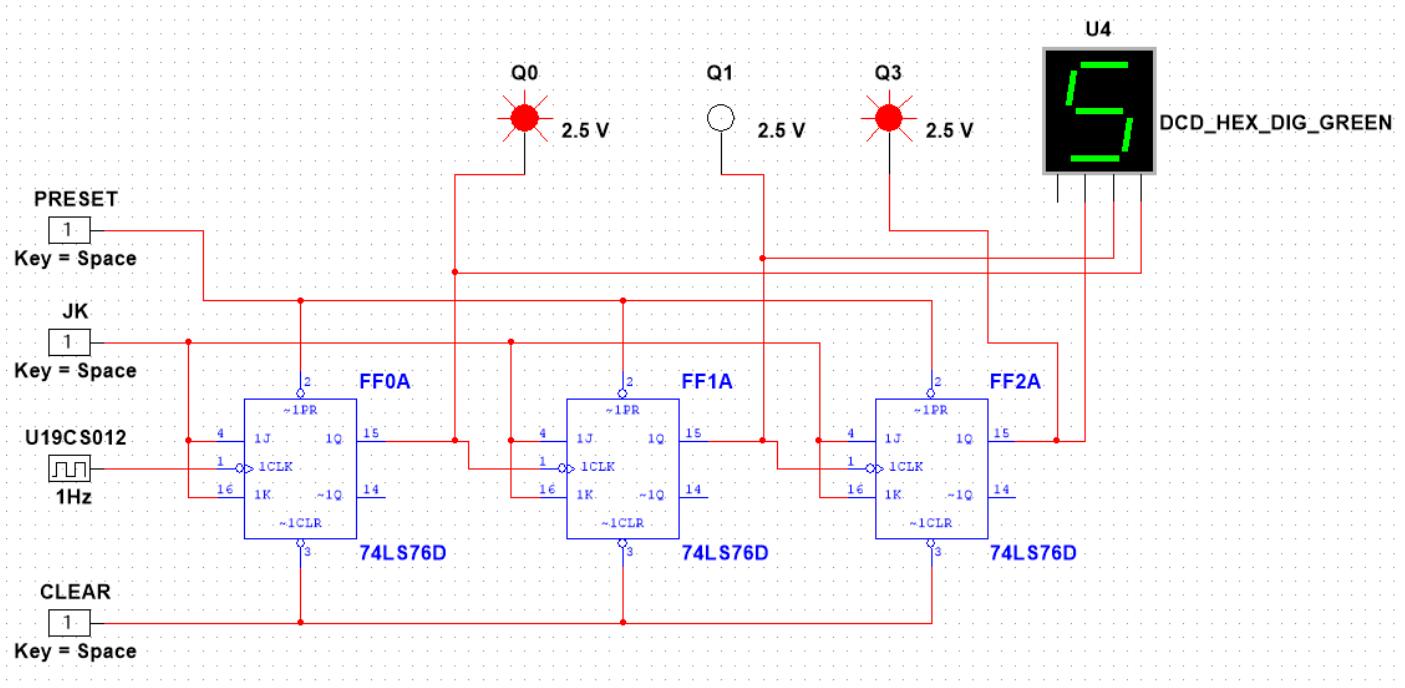
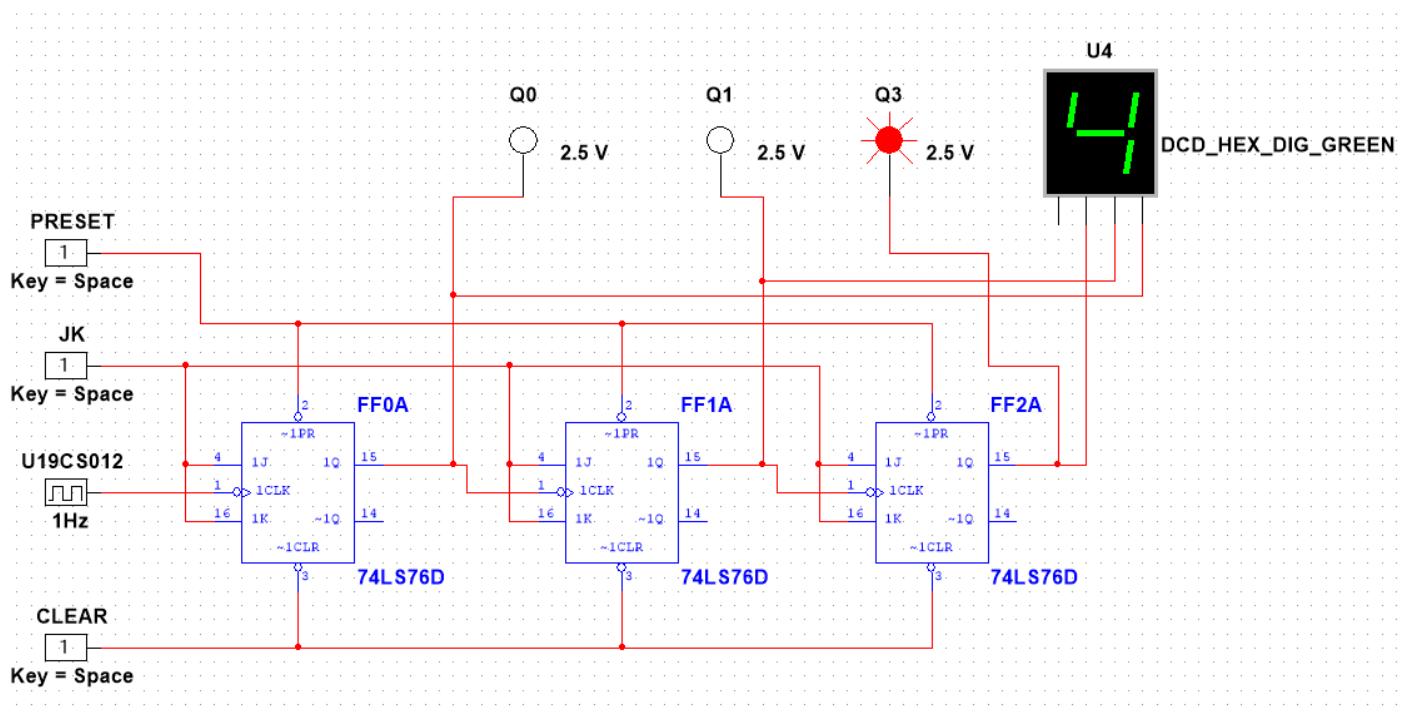


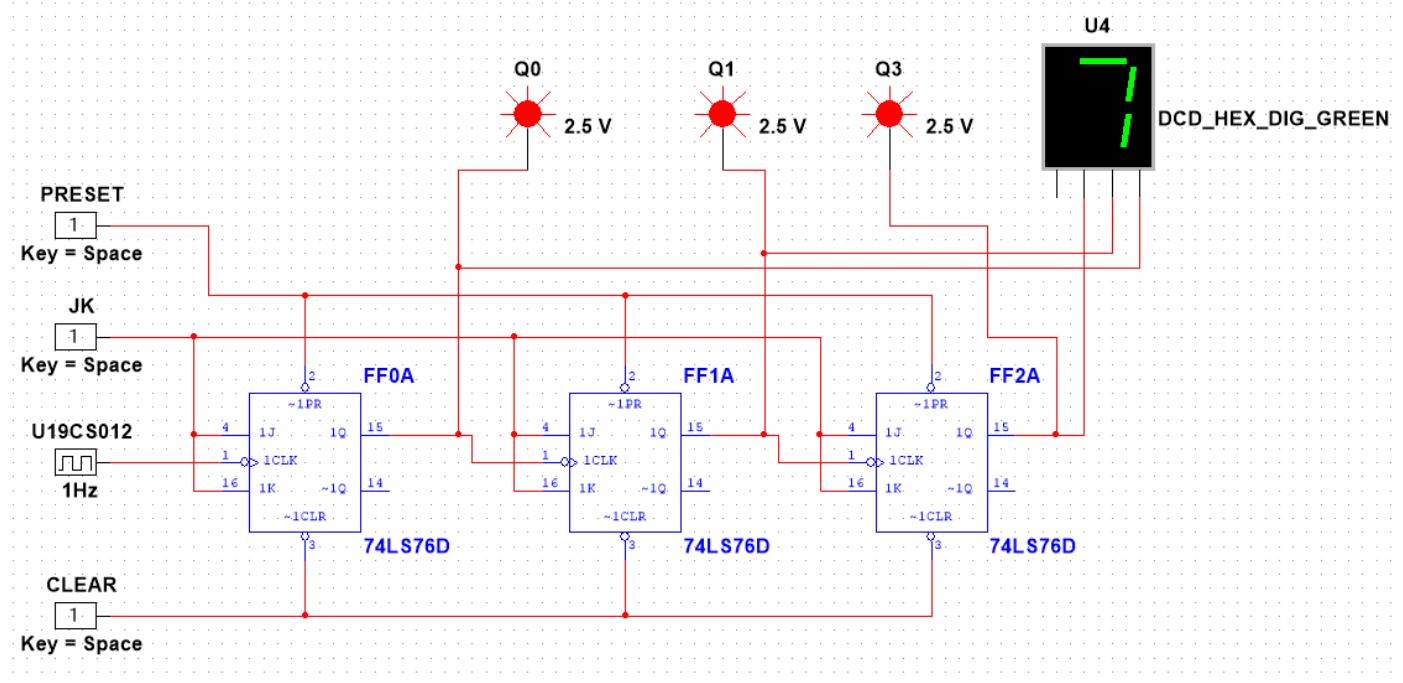
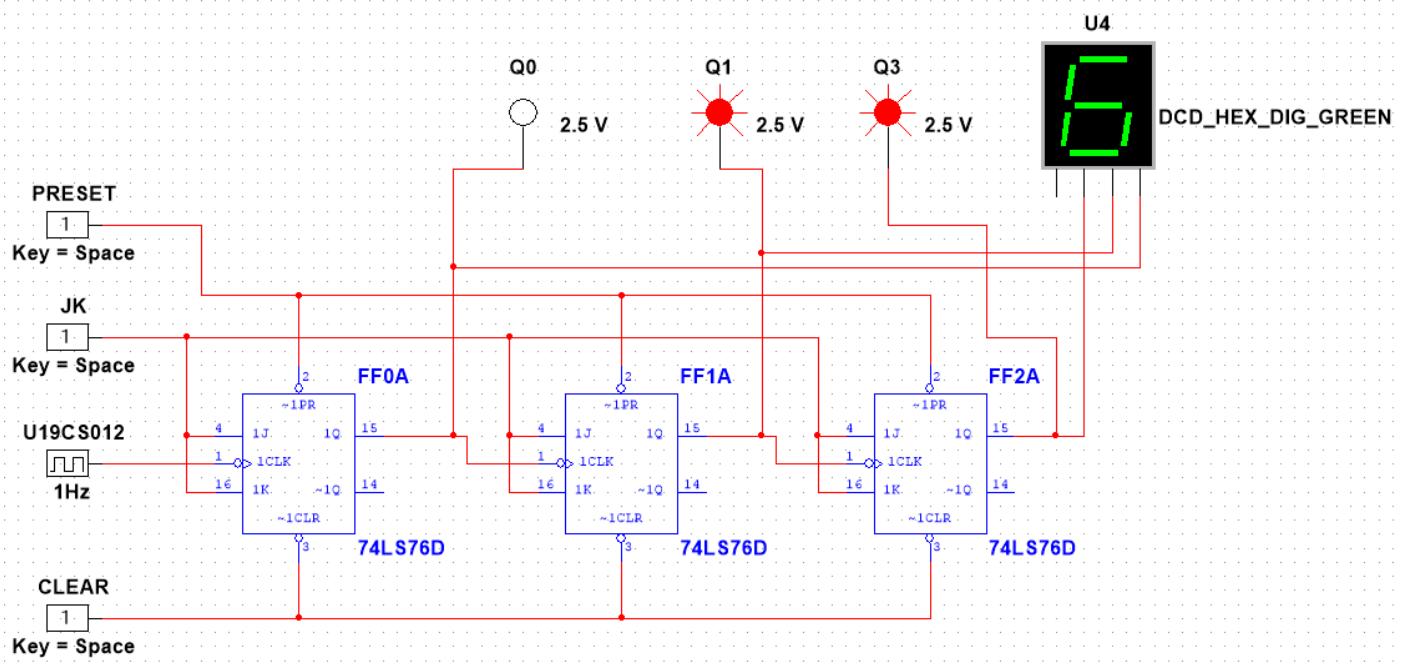
**SIMULATION SCREENSHOTS**

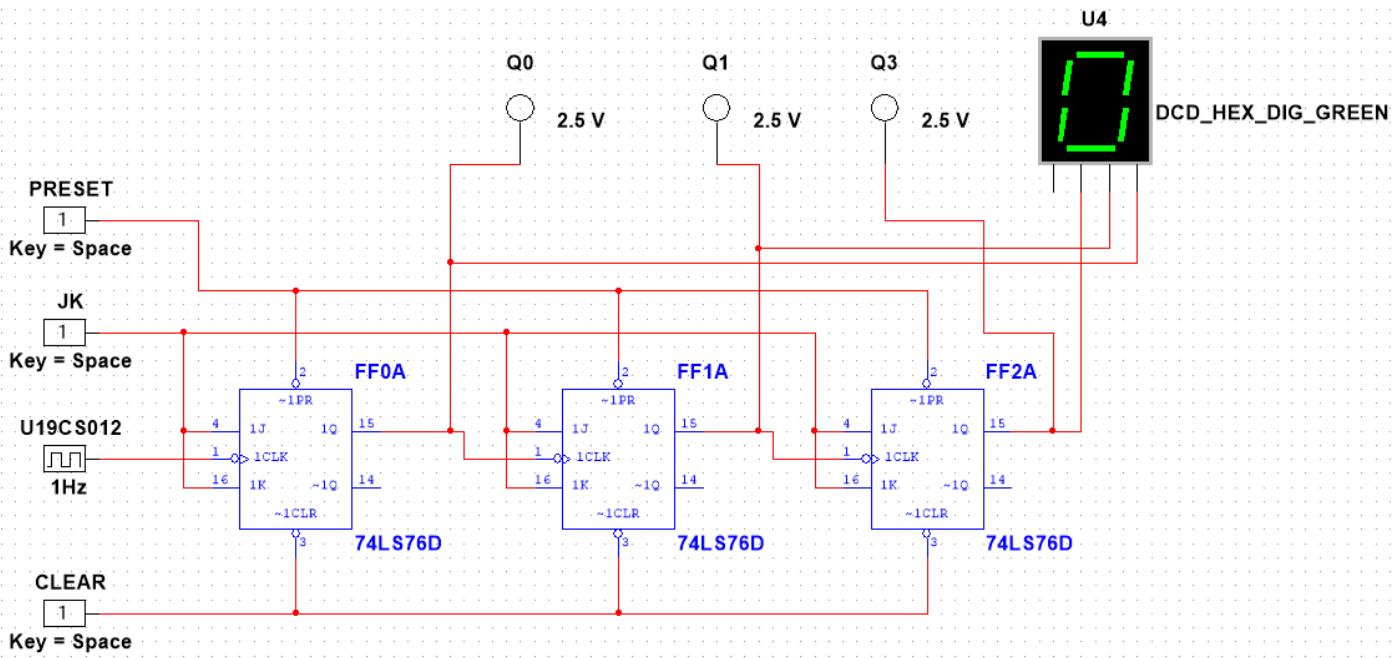
**3-BIT UP FULL MODULUS COUNTER**





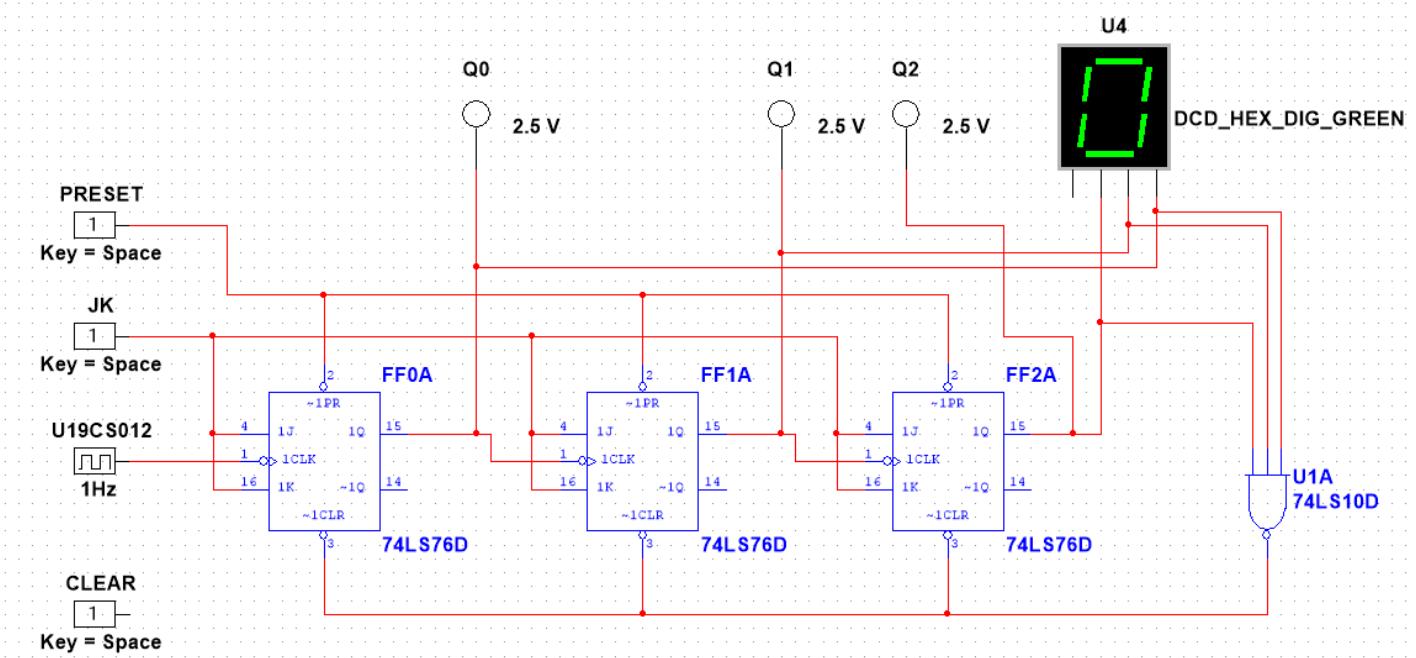


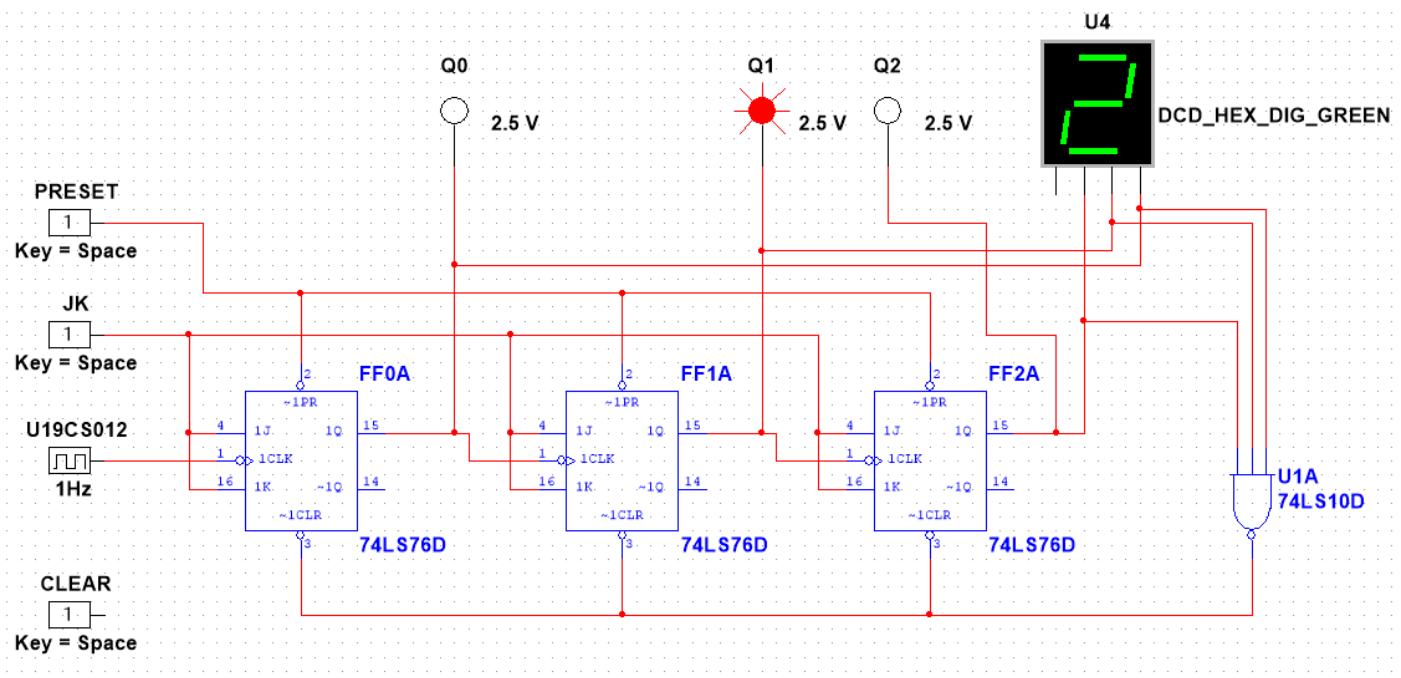
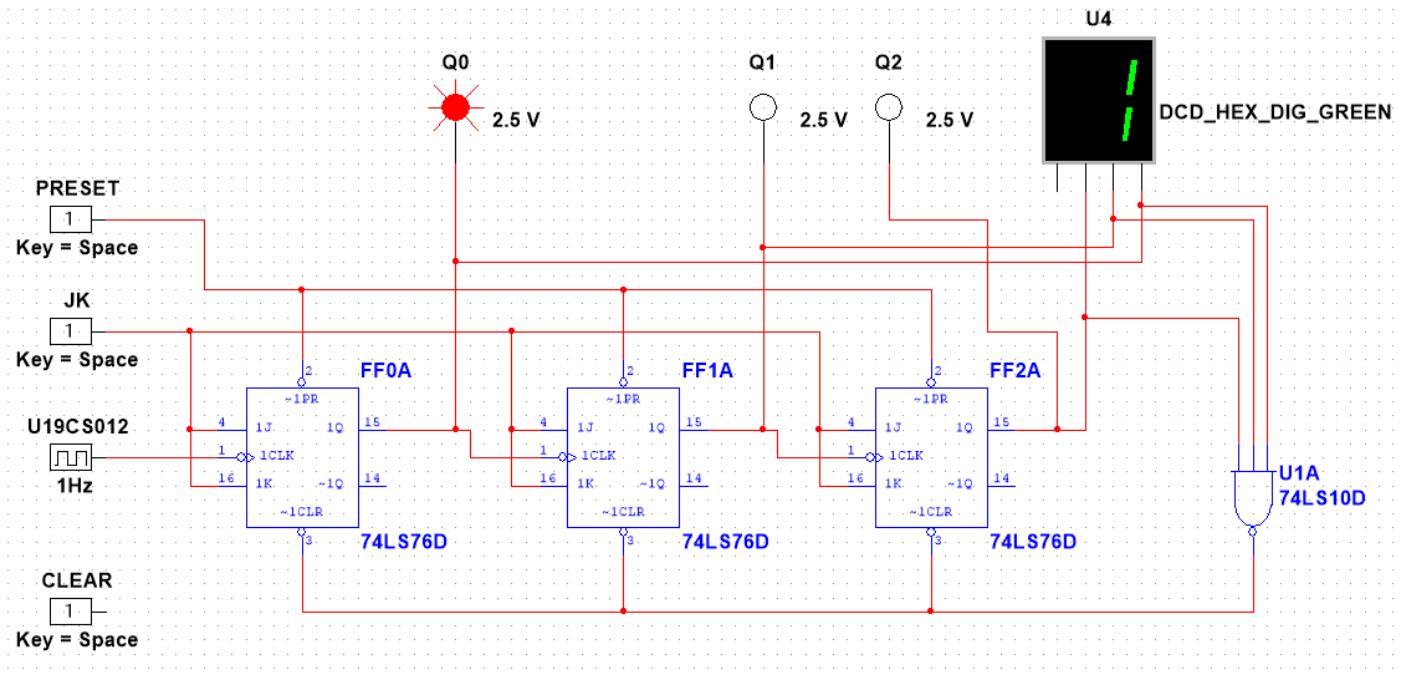


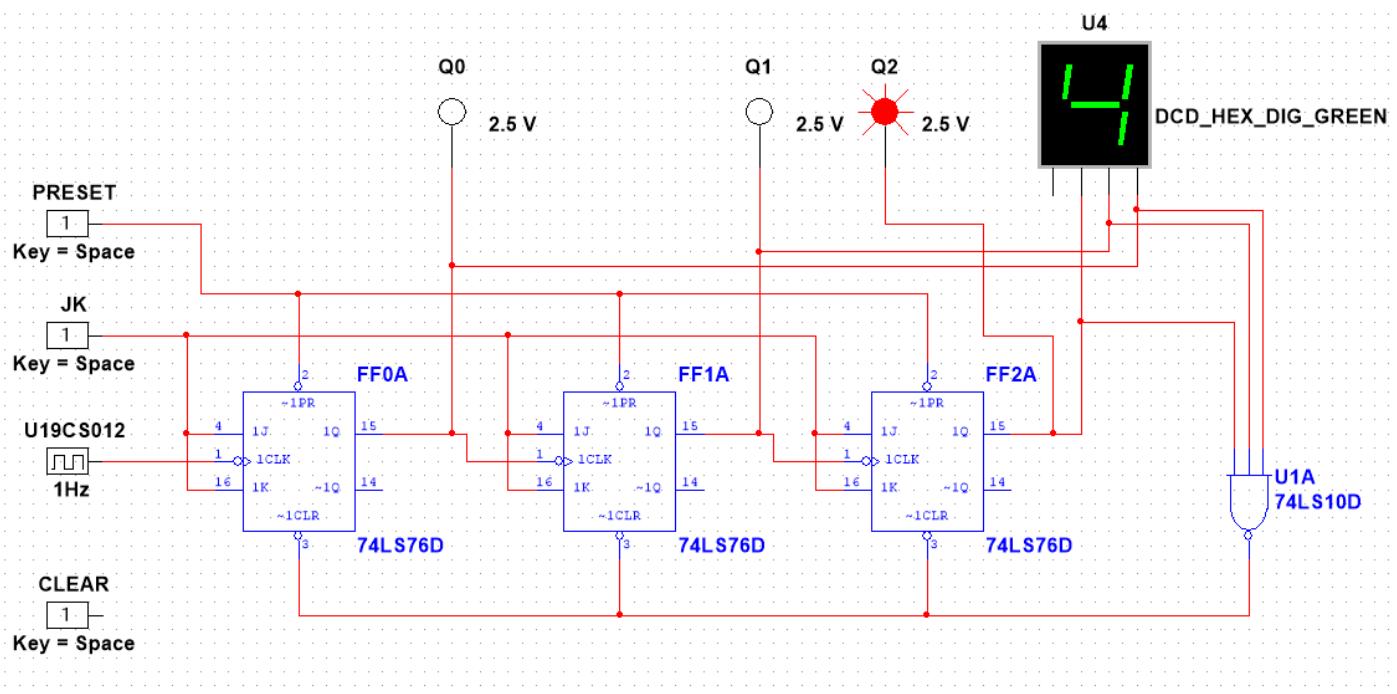
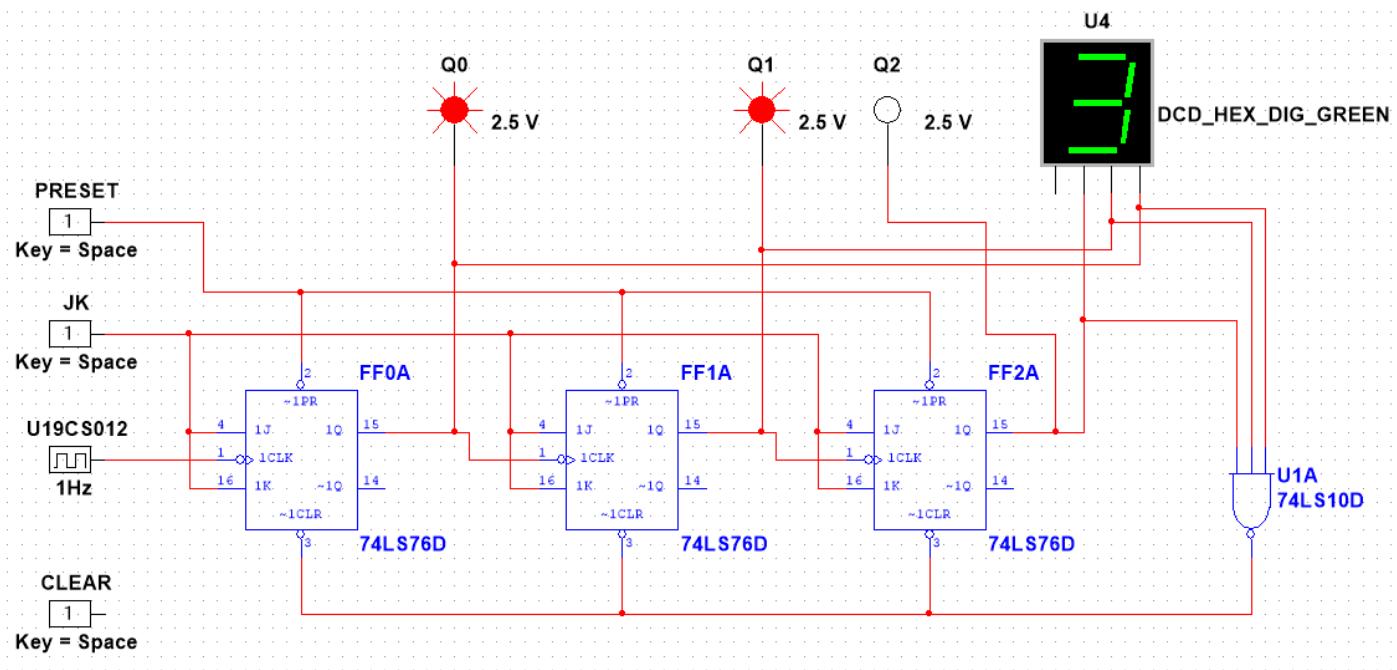


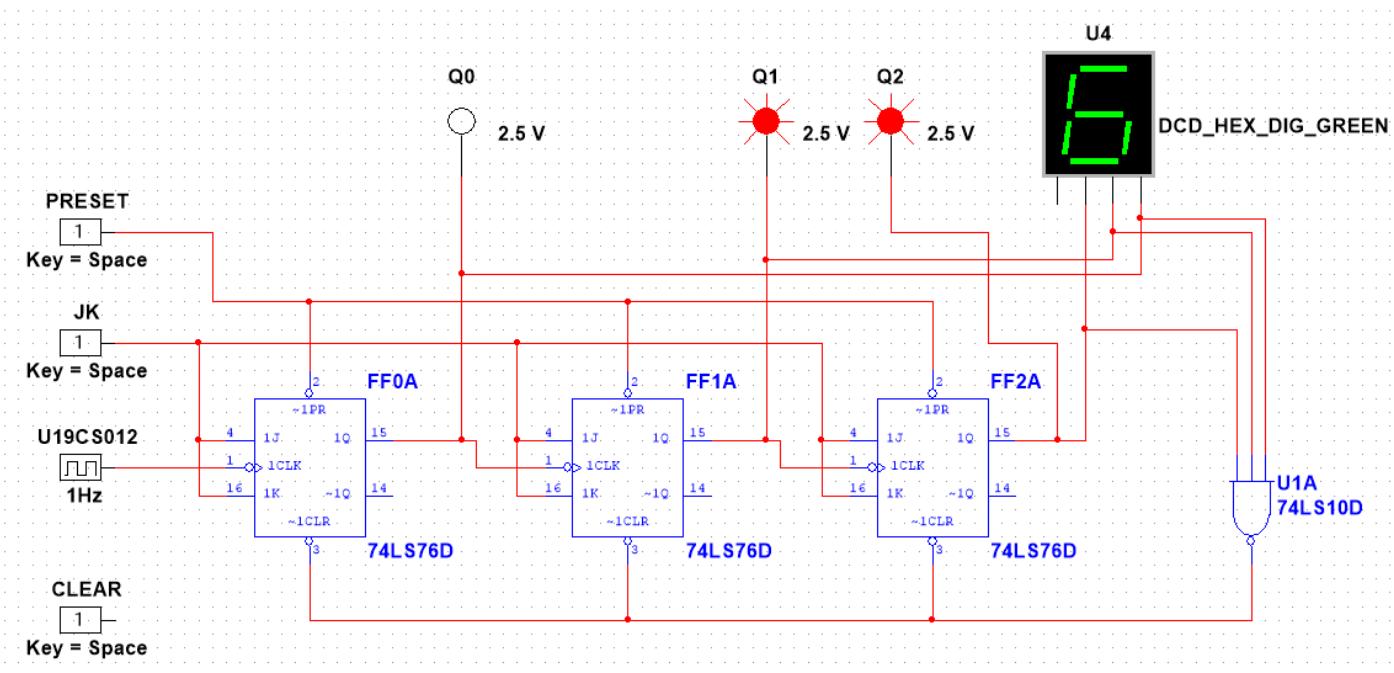
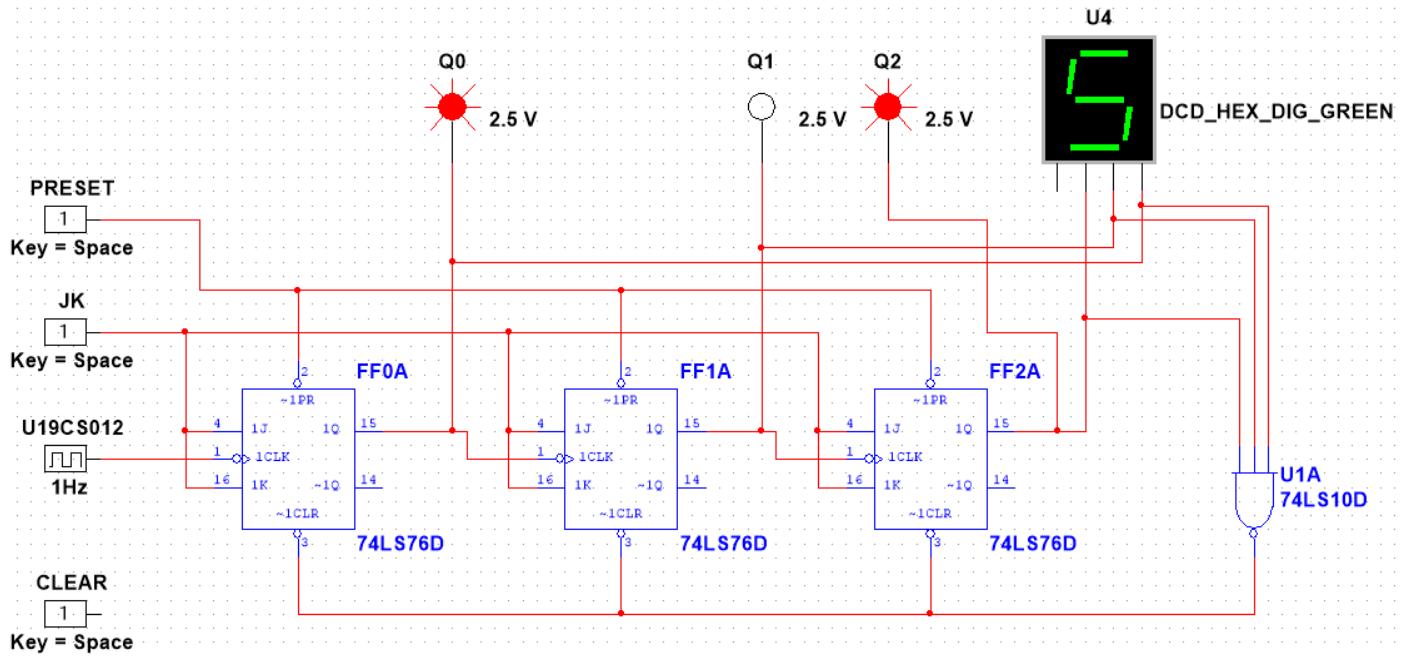
After this the Cycle Repeats Itself [0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 0]

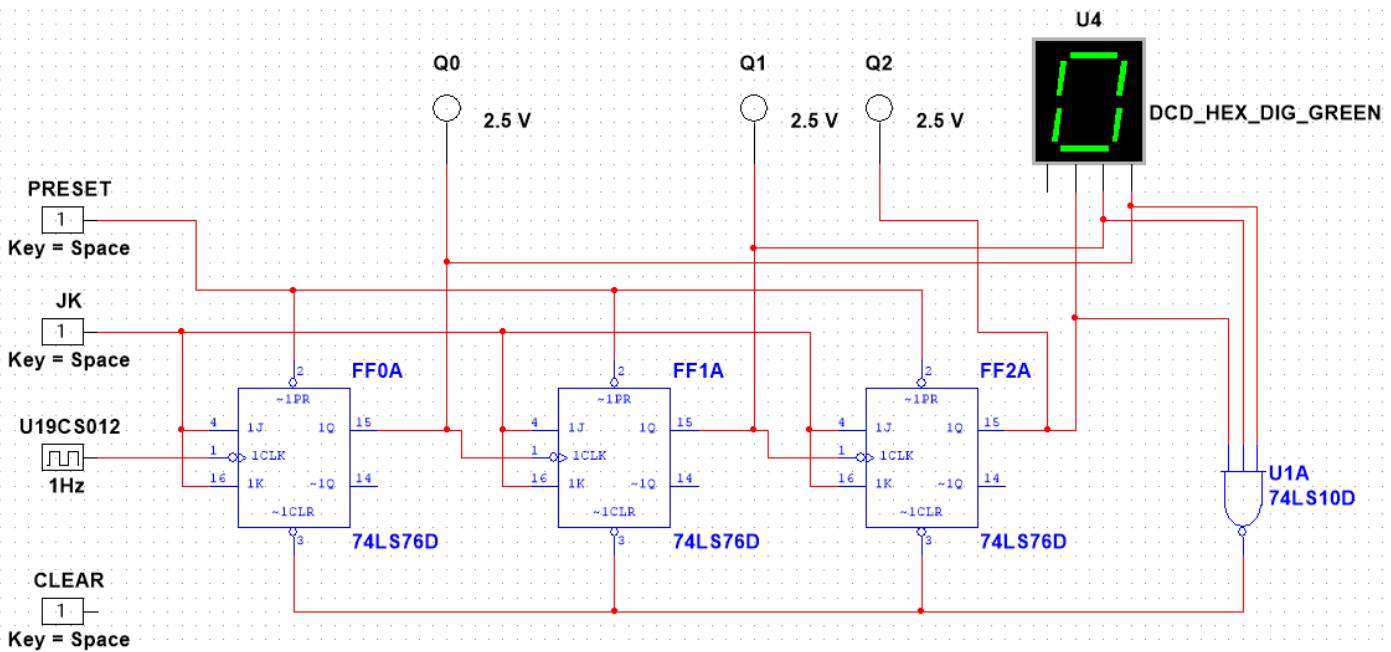
### MOD-7 COUNTER





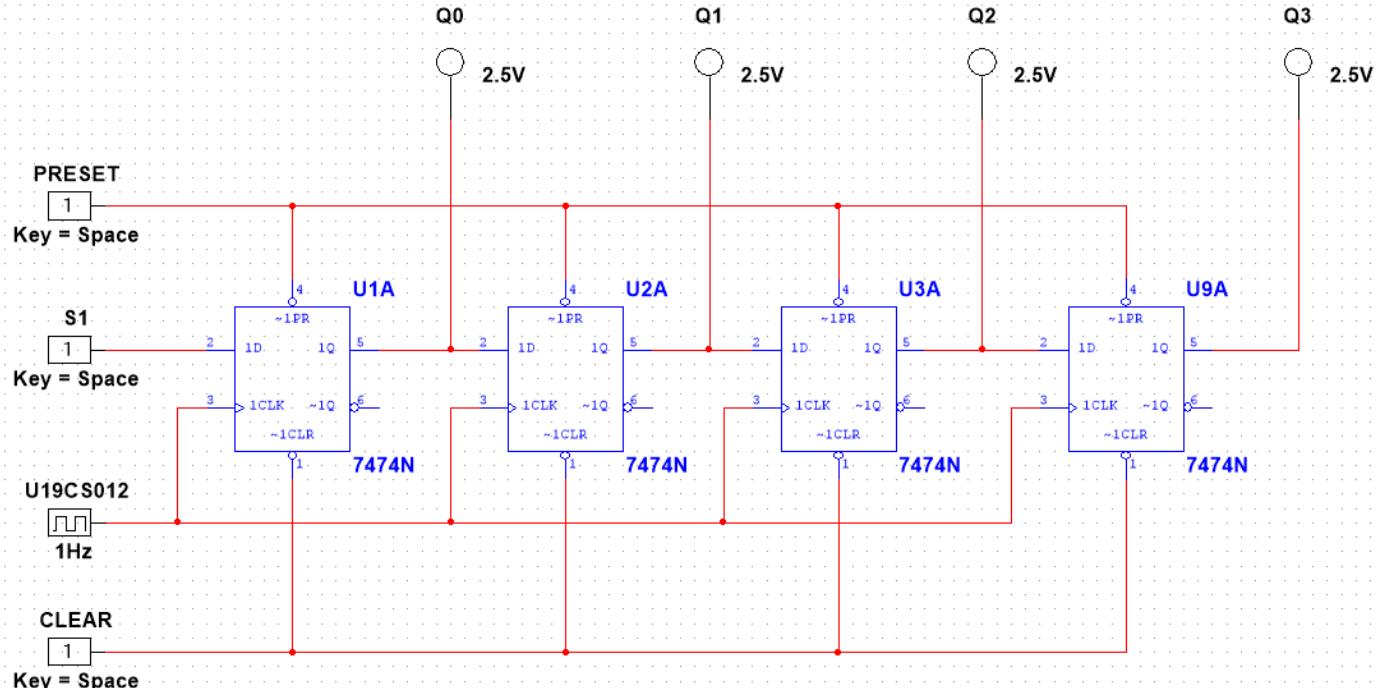


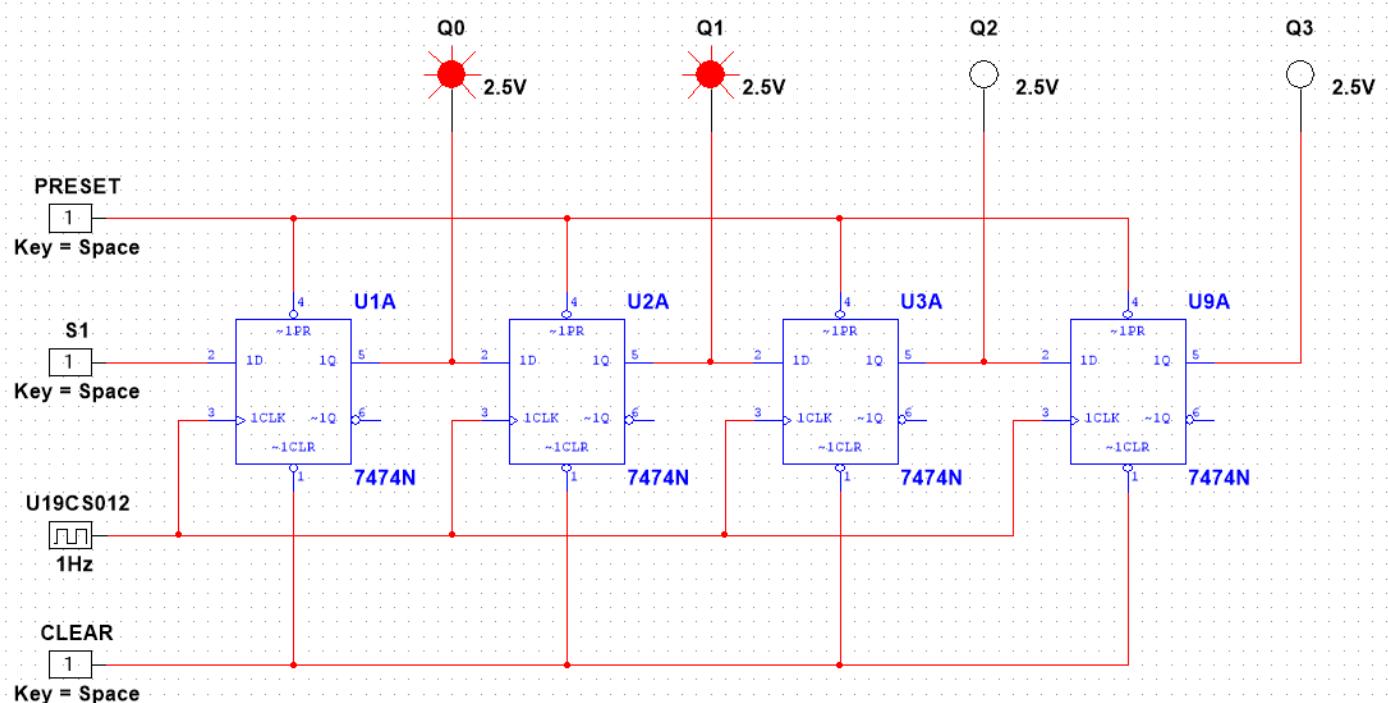
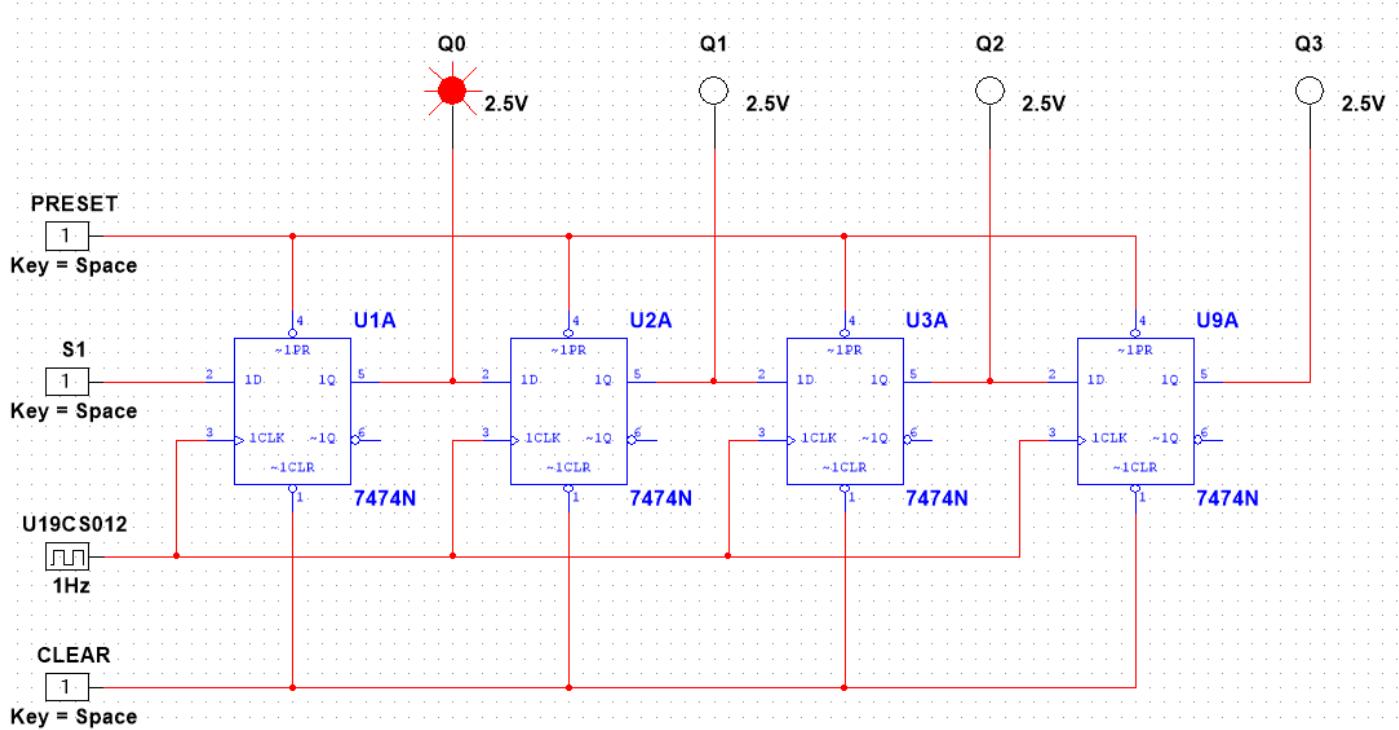


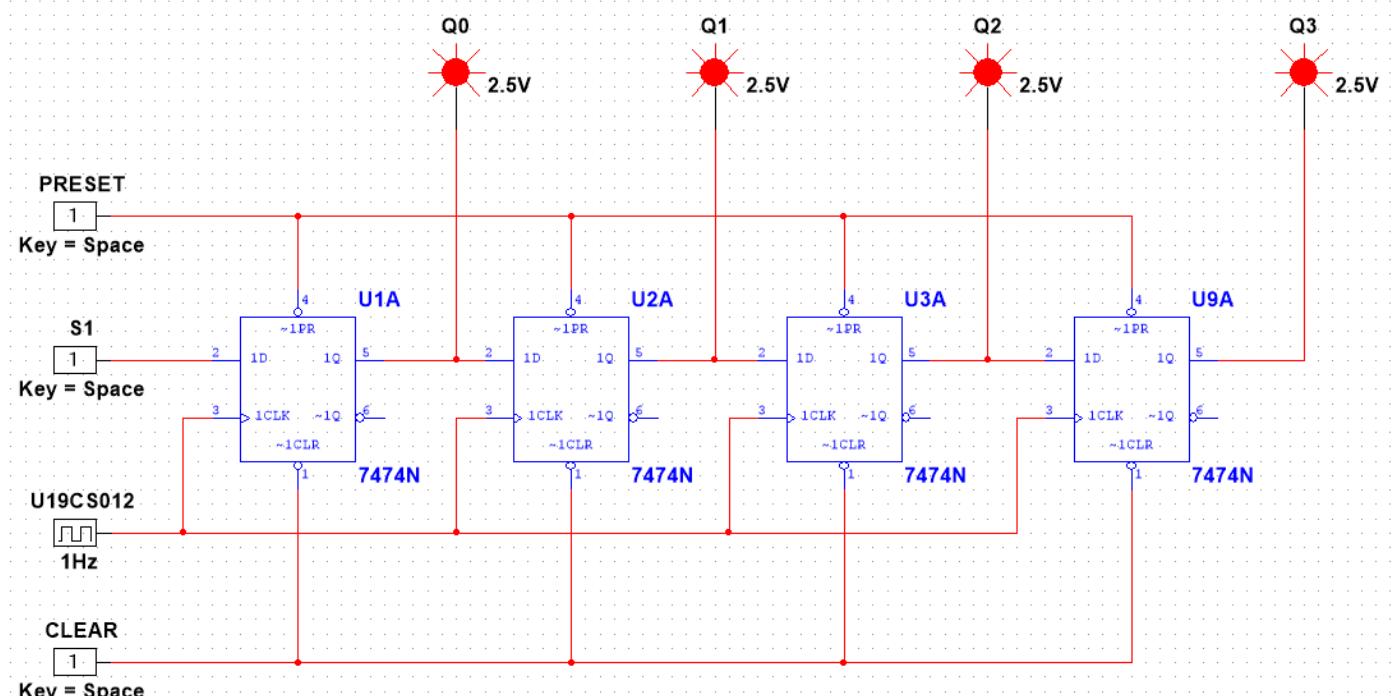
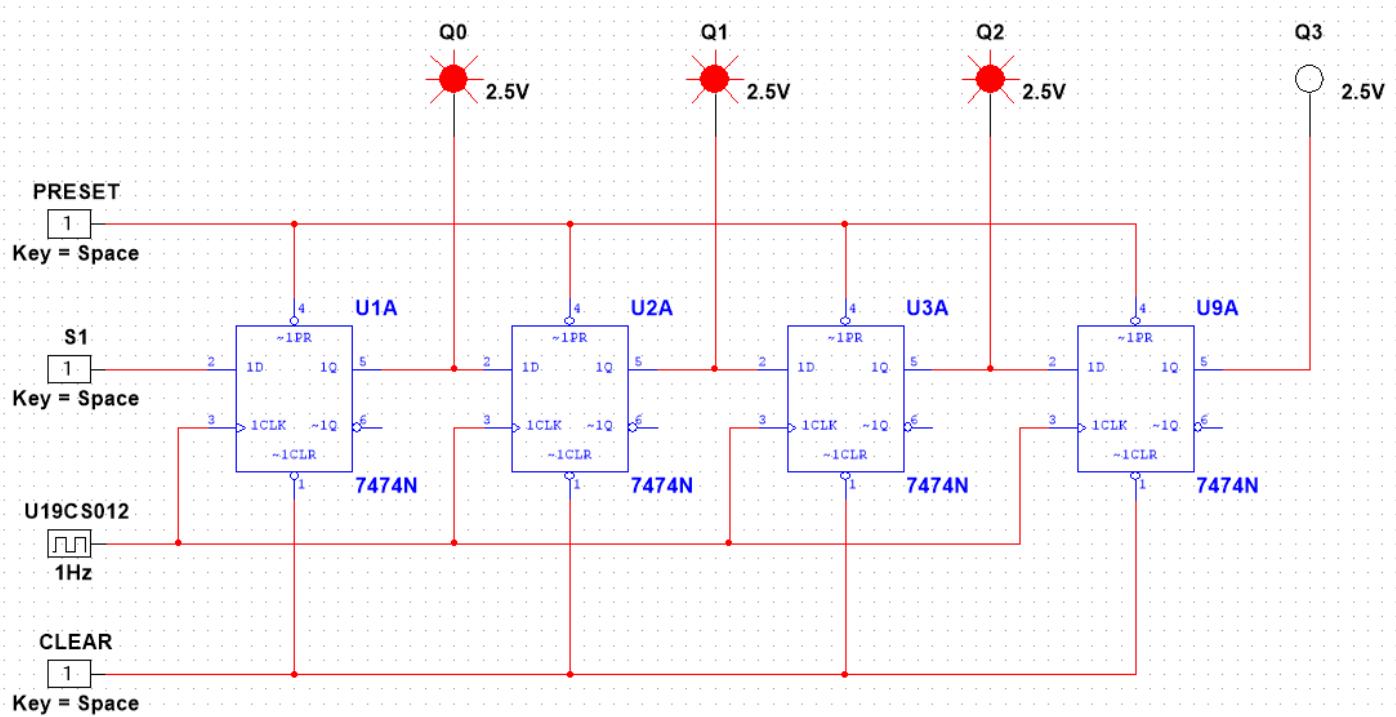


After this the Cycle Repeats Itself [0 → 1 → 2 → 3 → 4 → 5 → 6 → 0]

### SHIFT RIGHT REGISTER

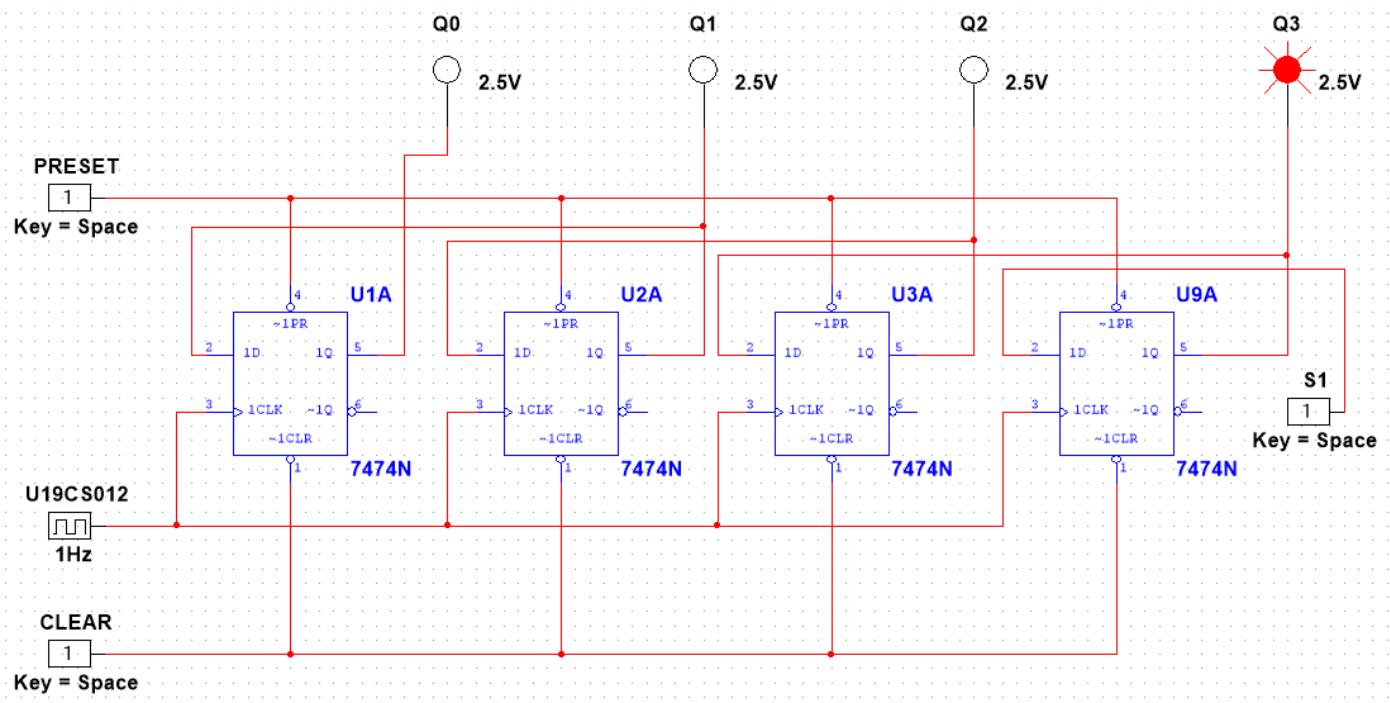
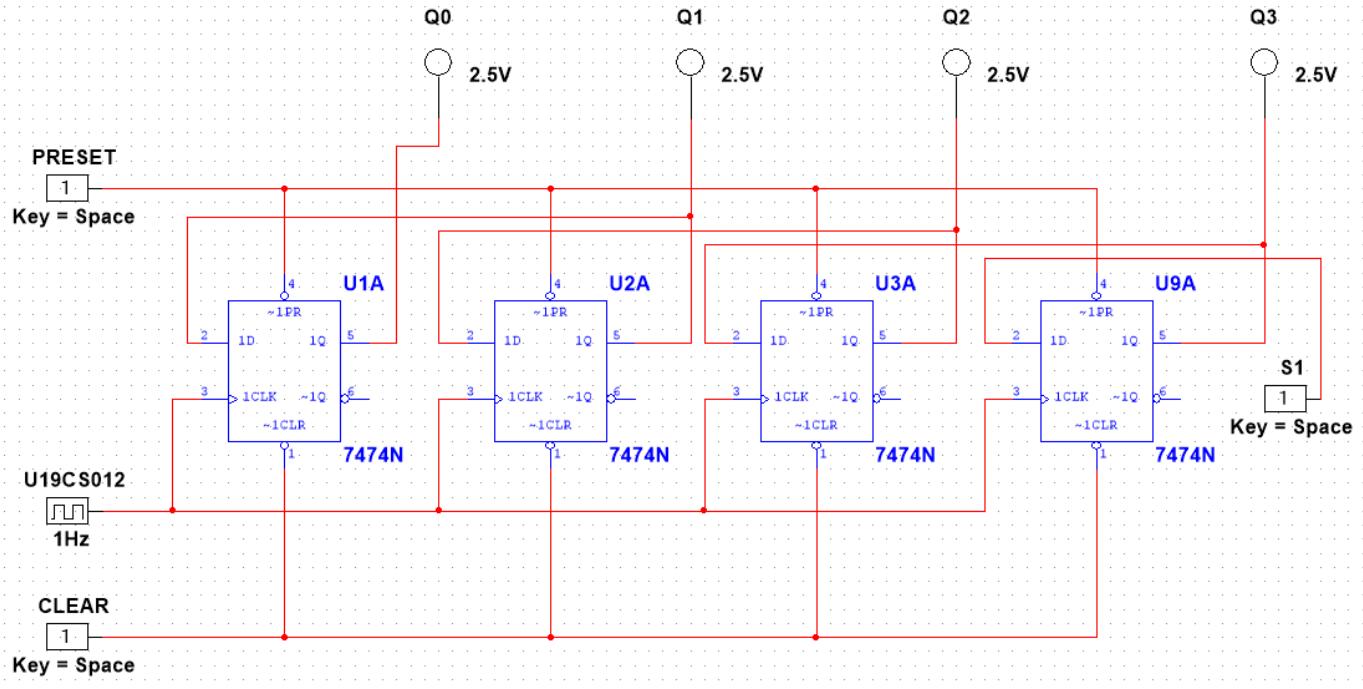


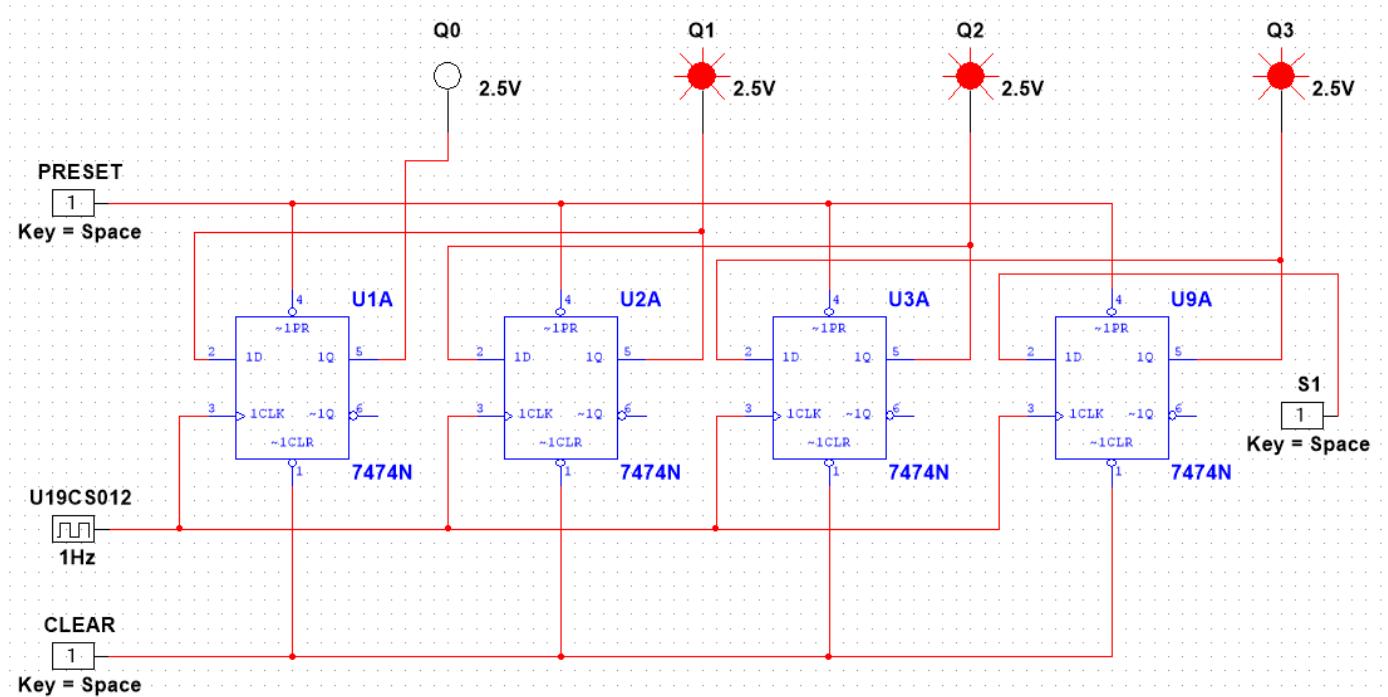
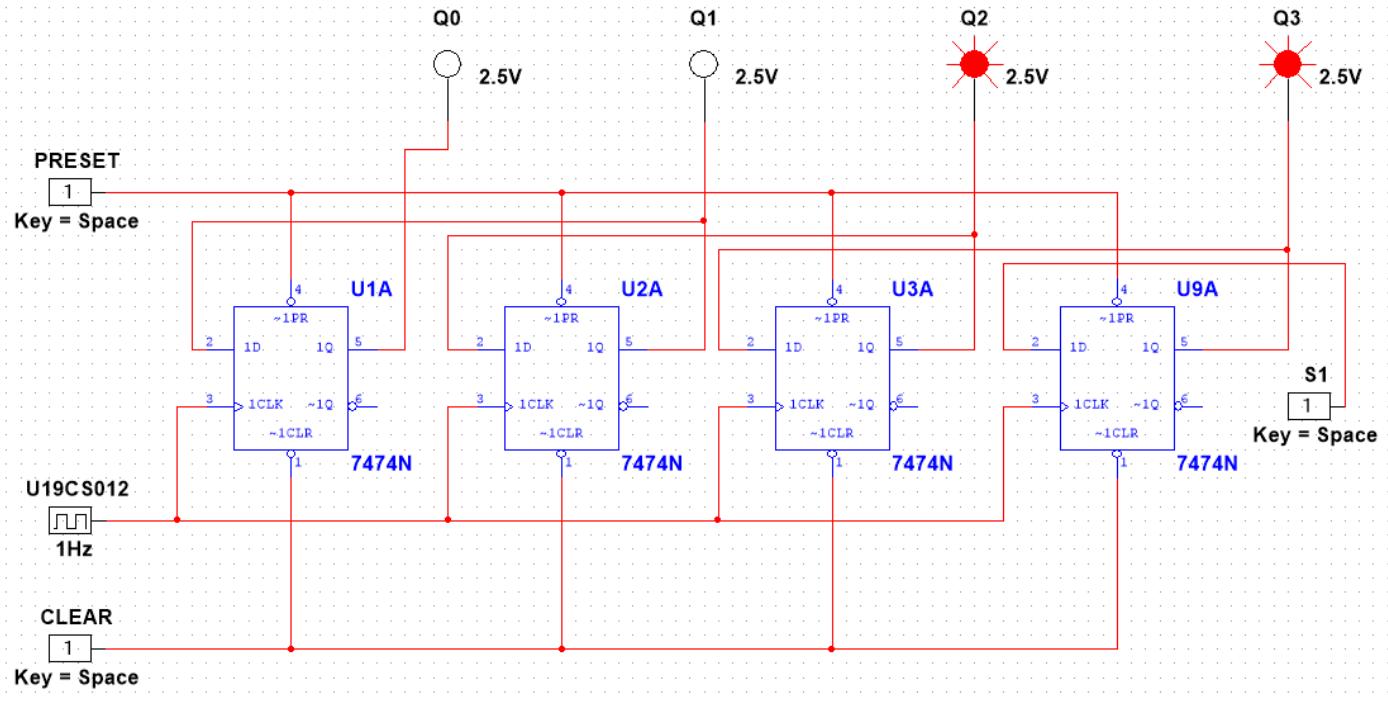


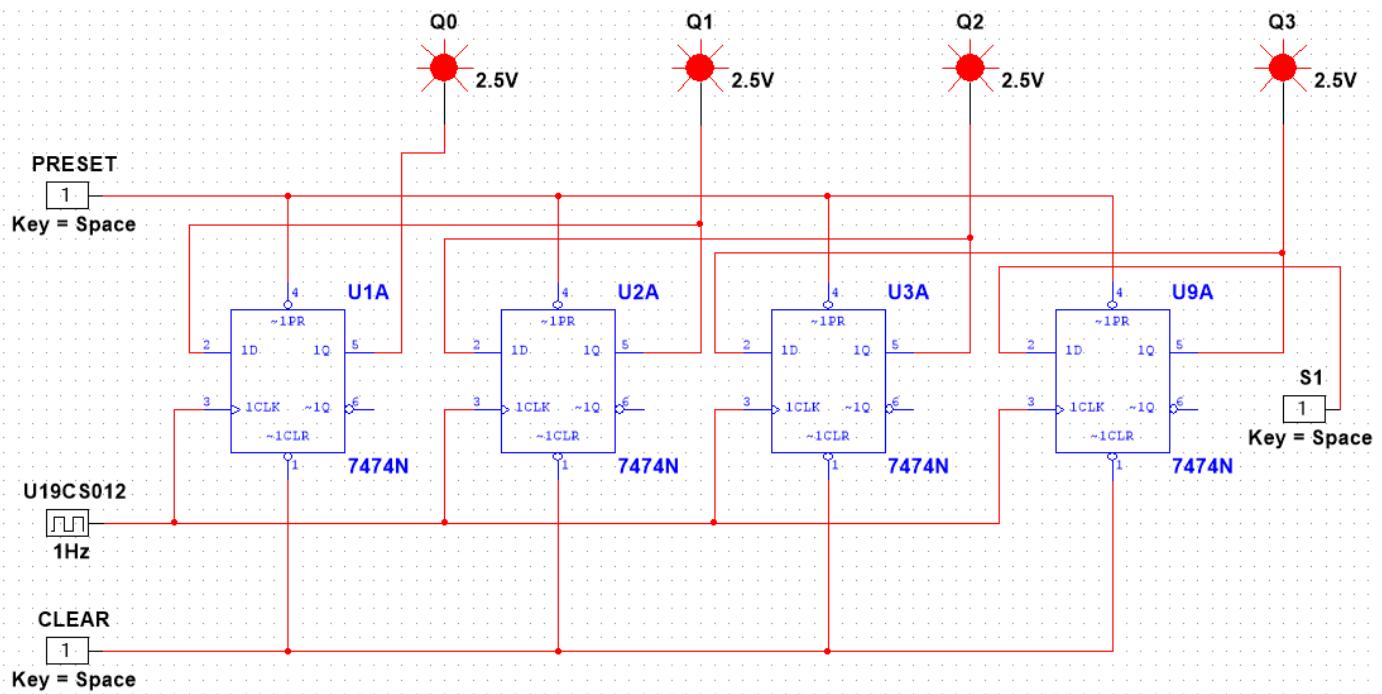




**SHIFT LEFT REGISTER**







### CONCLUSIONS

- 1.) In this Experiment, We have studied about Registers and Counters like 3-Bit Full Modulus and 7-Modulus Circuit and also implemented it successfully on Multisim.
- 2.) We Verified the Theoretical Knowledge of Shift Registers [Left and Right Shift] by Observing the Bit Movement in Shift Registers.
- 3.) We also Implemented 3-Bit Up Counter, Mod-7 Counter, 4-Bit Shift Right Register and 4-Bit Shift Left Register using Multisim and Observed its Working.



# ASSIGNMENT-11

U19CS012

1.) Design and Implement MOD-12 Counter in Multisim using JK Flip-Flops.

A.) Solution:

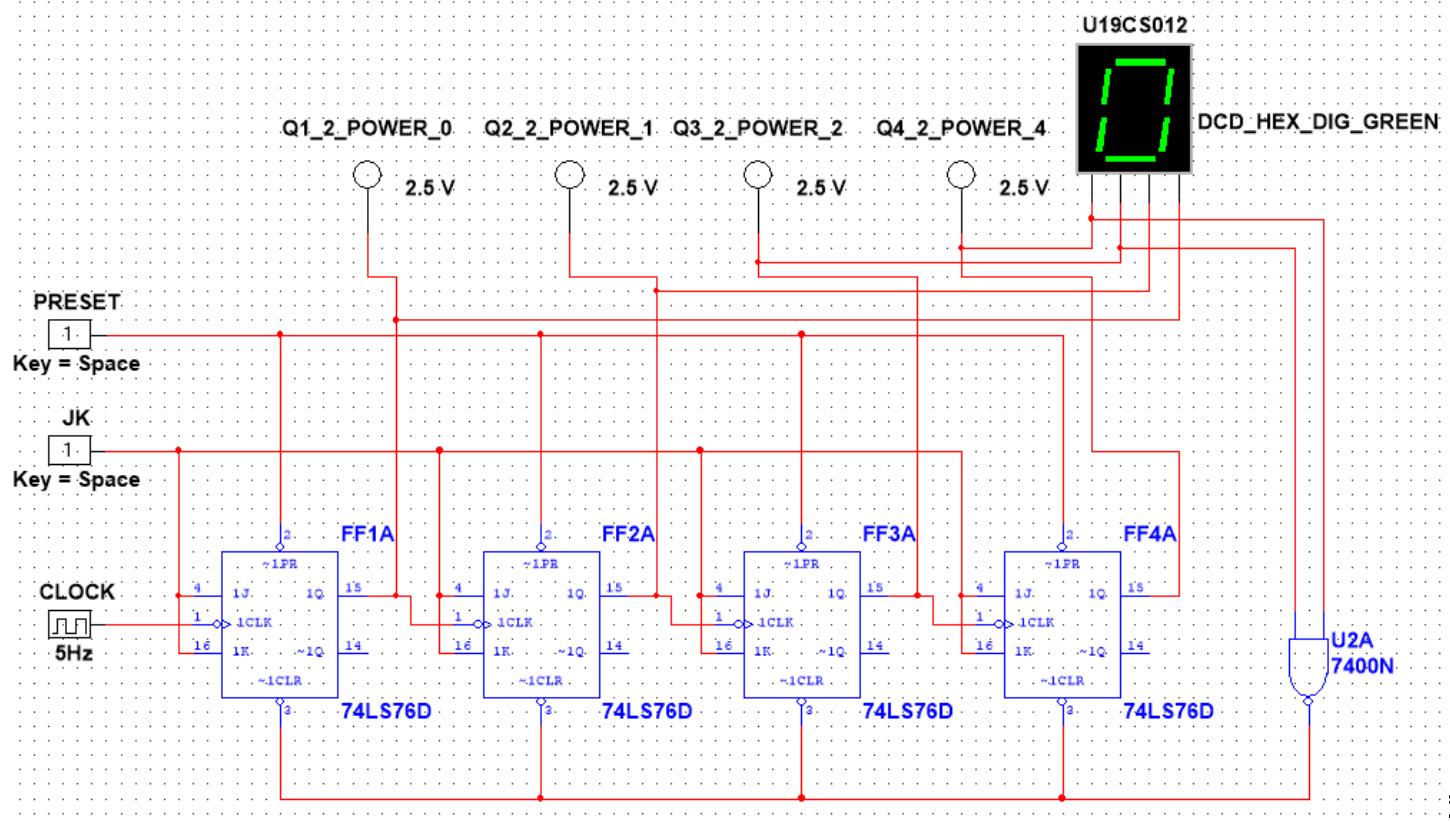
A MOD-12 Counter would be having 12 Valid States from 0 [0000] to 11 [1011].

We will make 4-Bit Full Counter [0-15] and Try to Reduce Valid States to 12[0-11] using Additional NAND Gate that will take *Clear All Bits* as the Value in Counter turns out to be 12 [1100] i.e. When *First Two Bits Become One*, we will Clear all Bits.

The Concept is Similar to MOD-7 Counter Implemented Earlier in Practical 11.

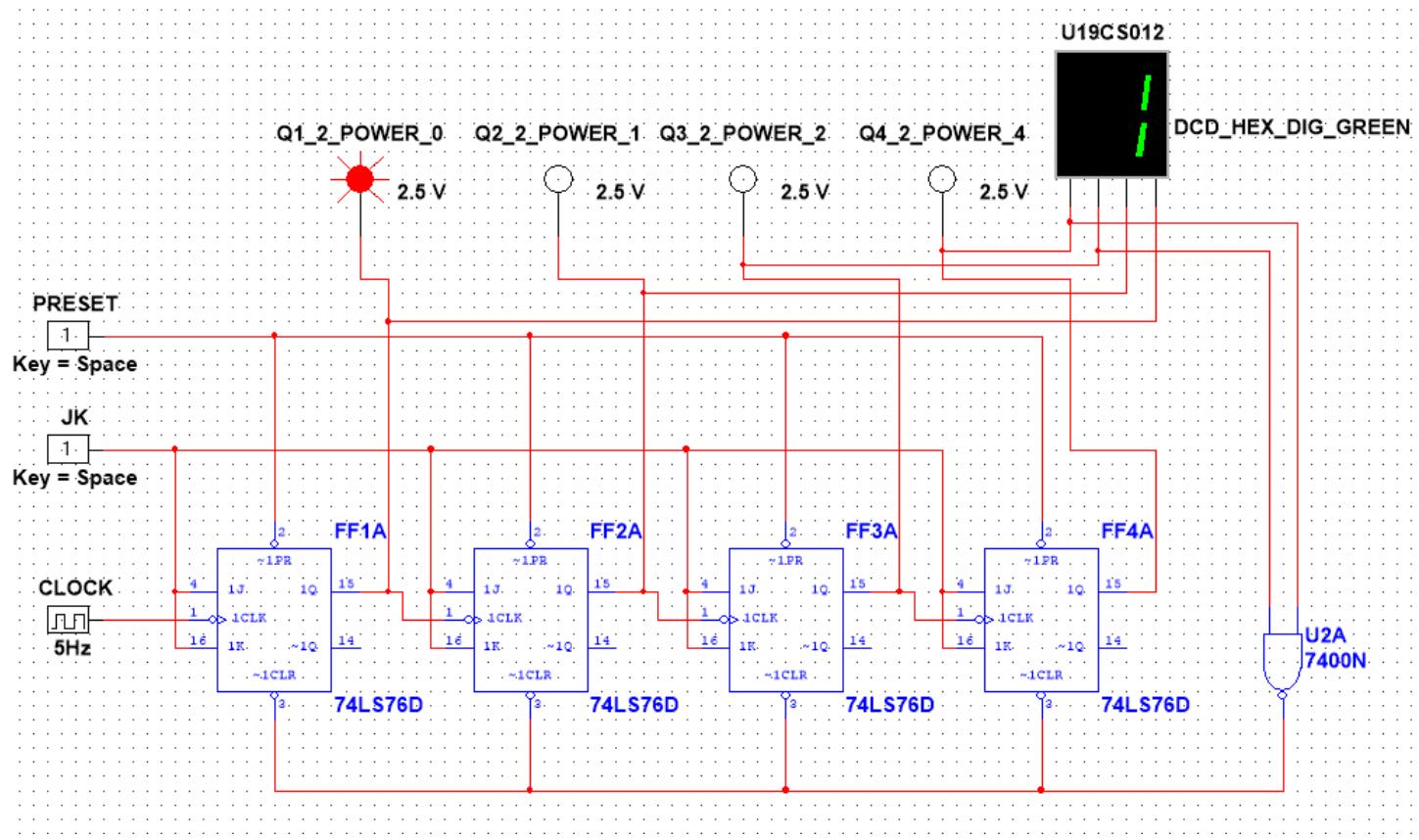
B.) Implementation:

1.) Valid State: 0 [0000]

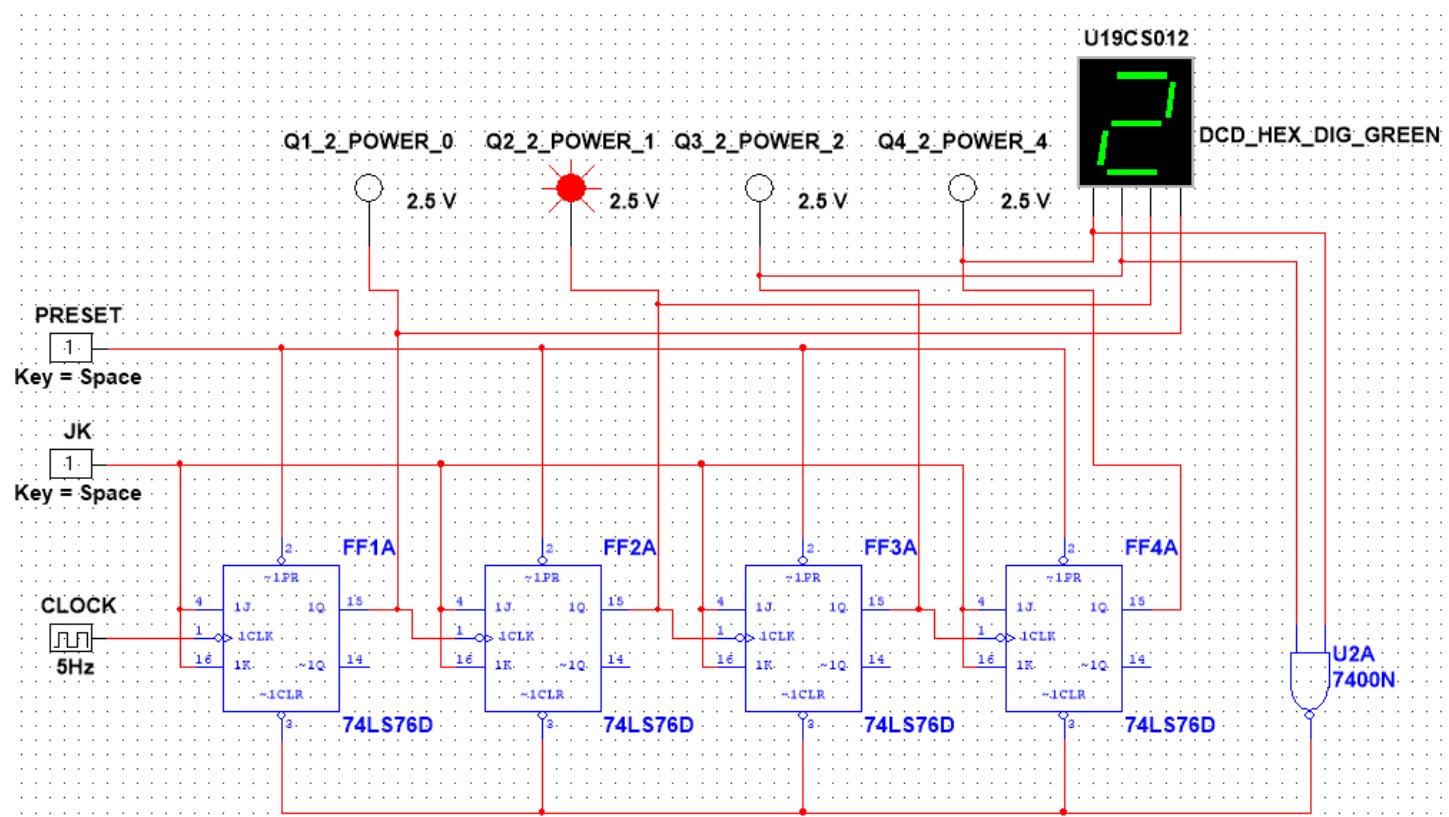




2.) Valid State: 1 [0001]

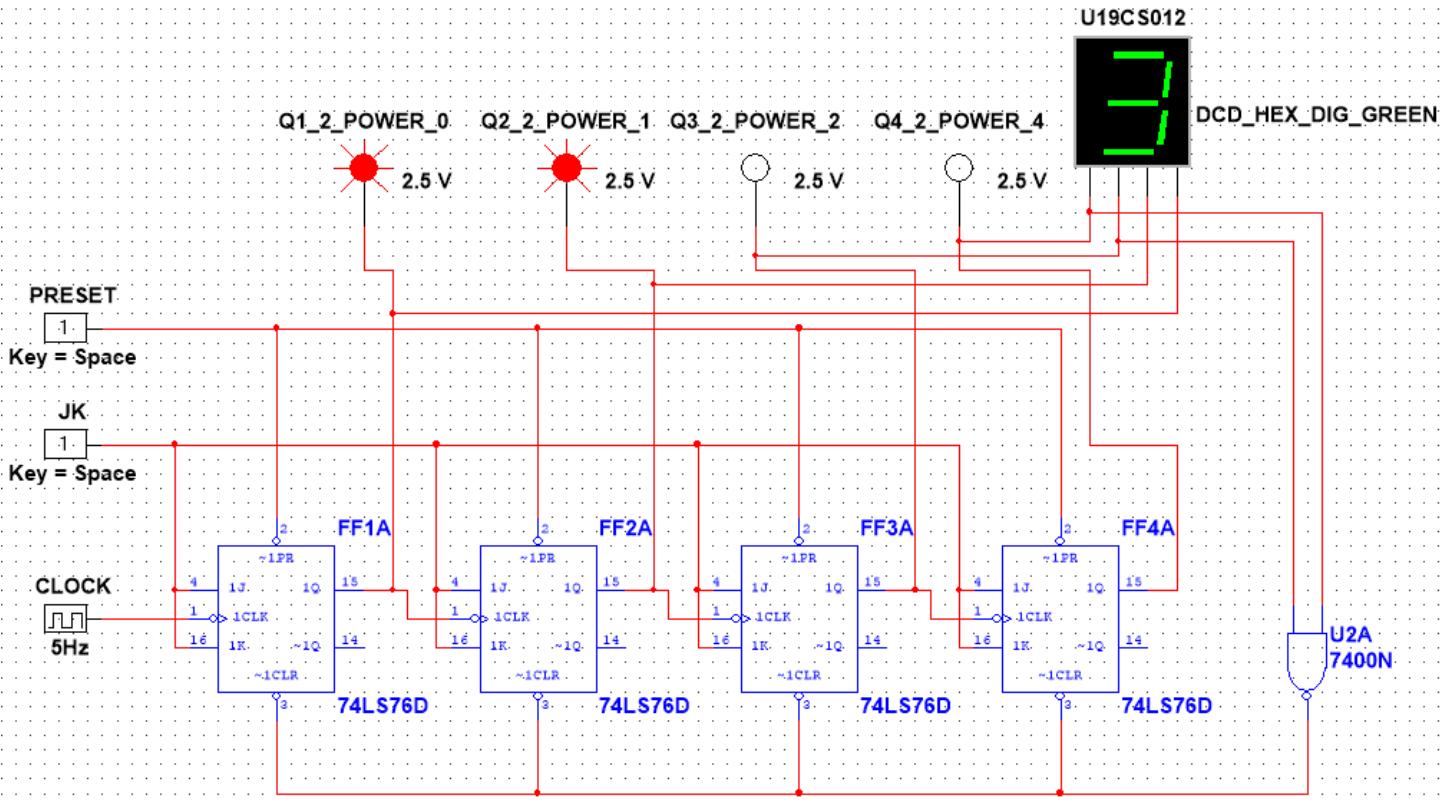


3.) Valid State: 2 [0010]

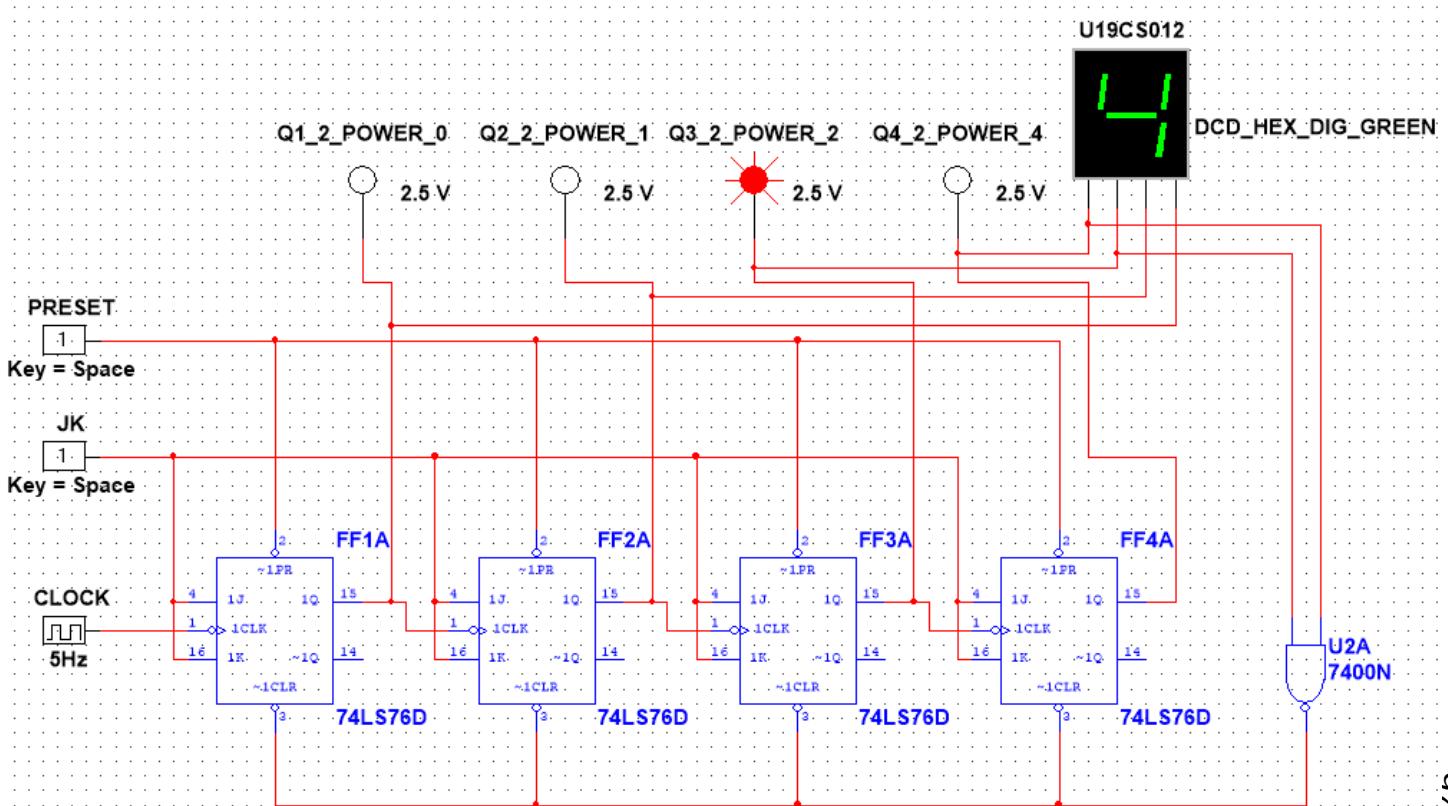




4.) Valid State: 3 [0011]

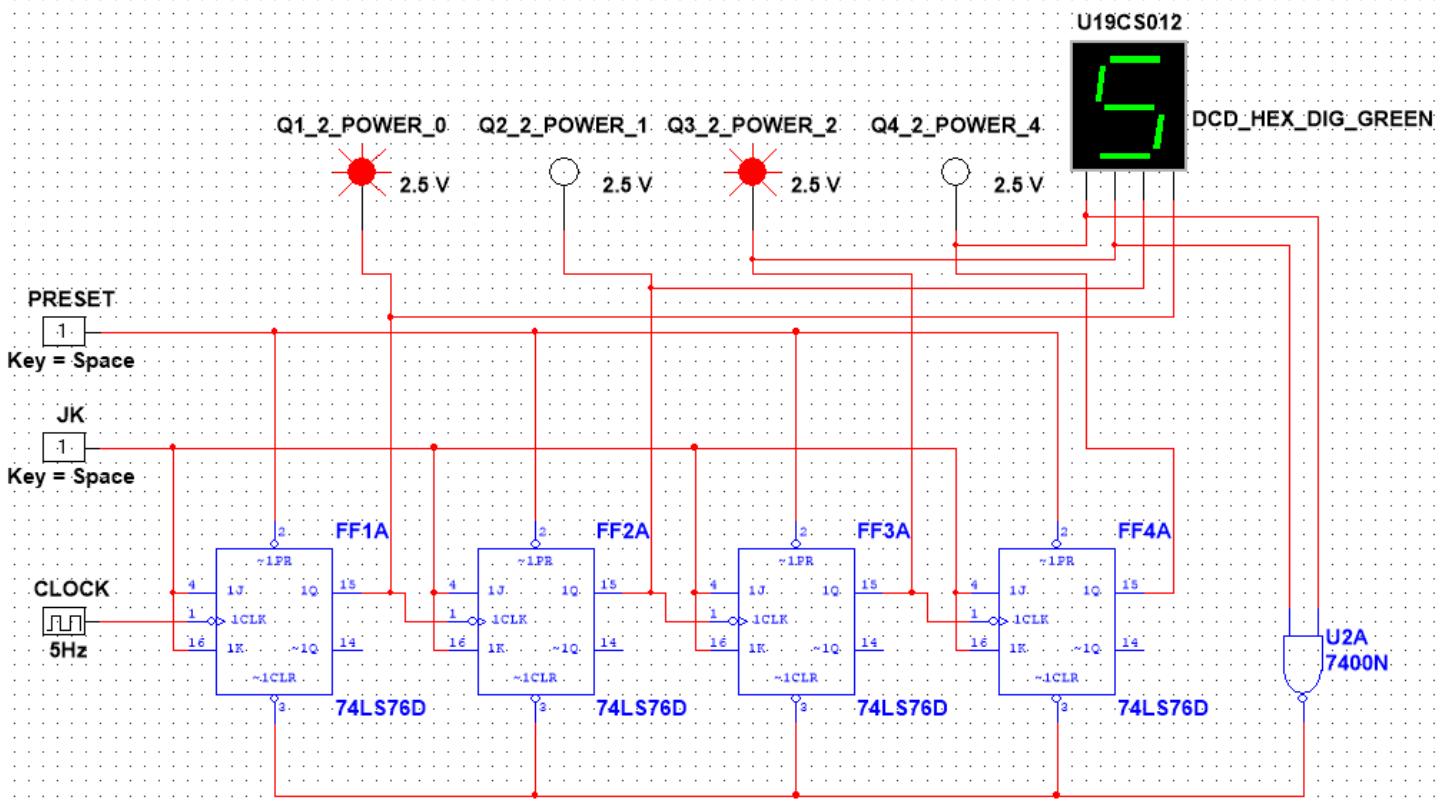


5.) Valid State: 4 [0100]

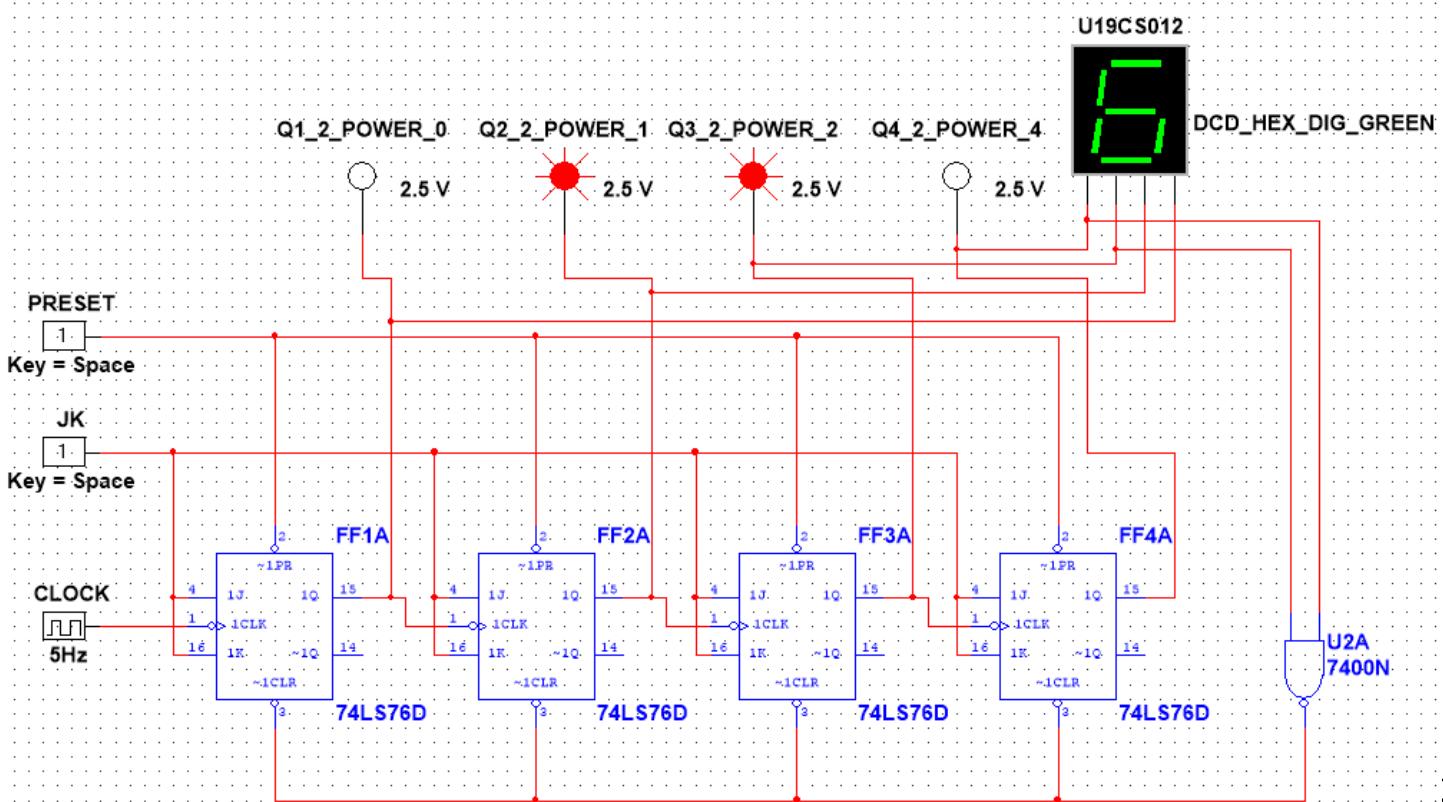




6.) Valid State: 5 [0101]

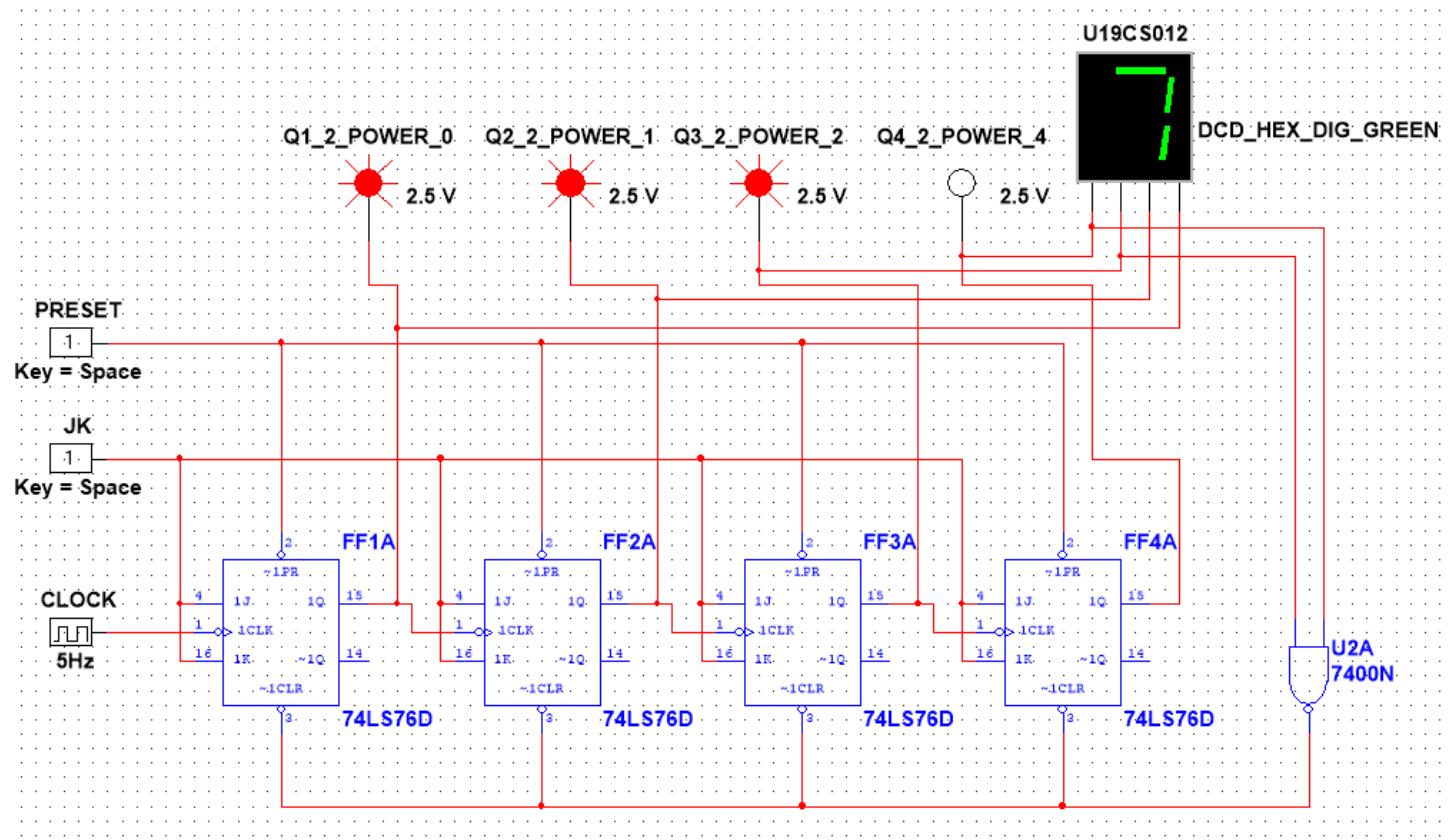


7.) Valid State: 6 [0110]

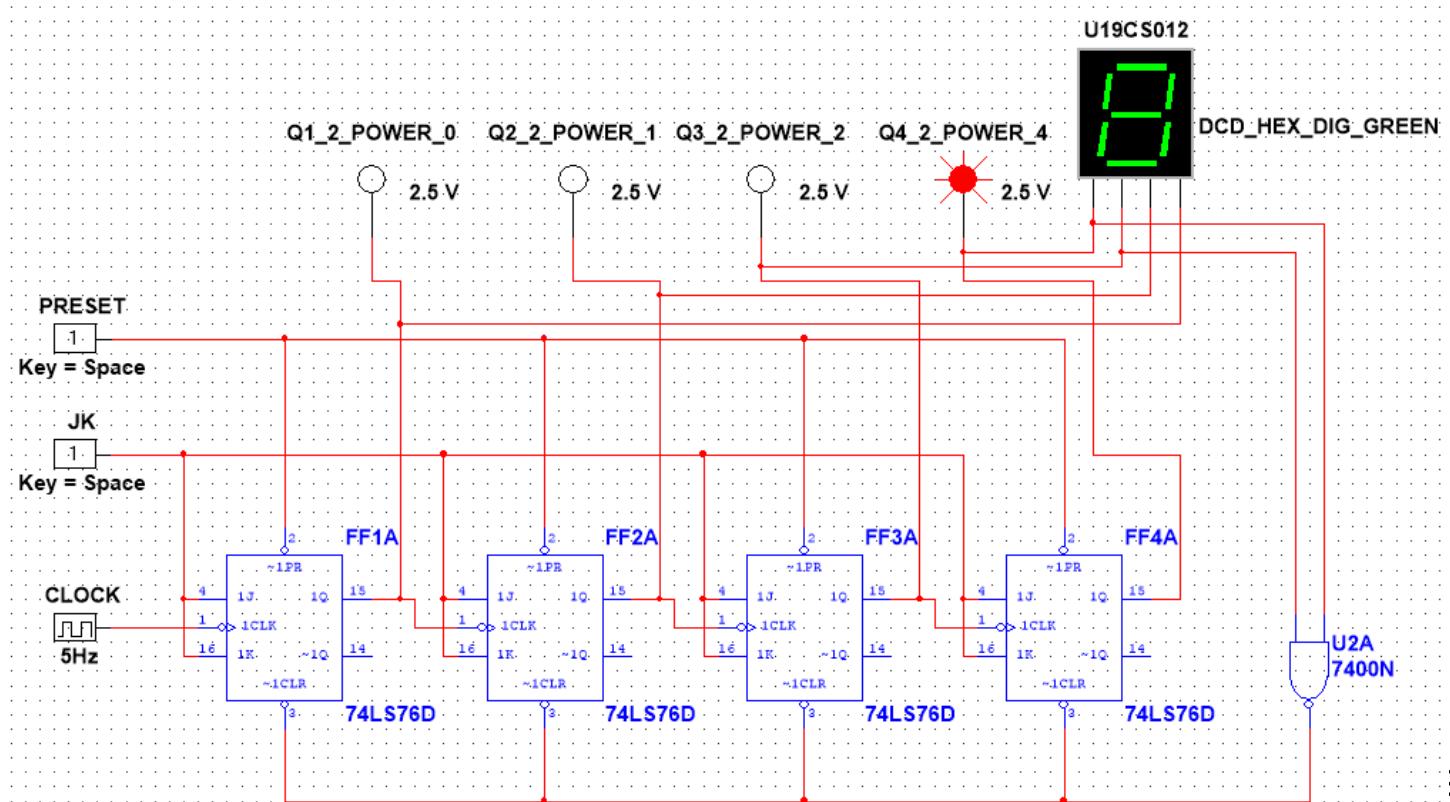




8.) Valid State: 7 [0111]

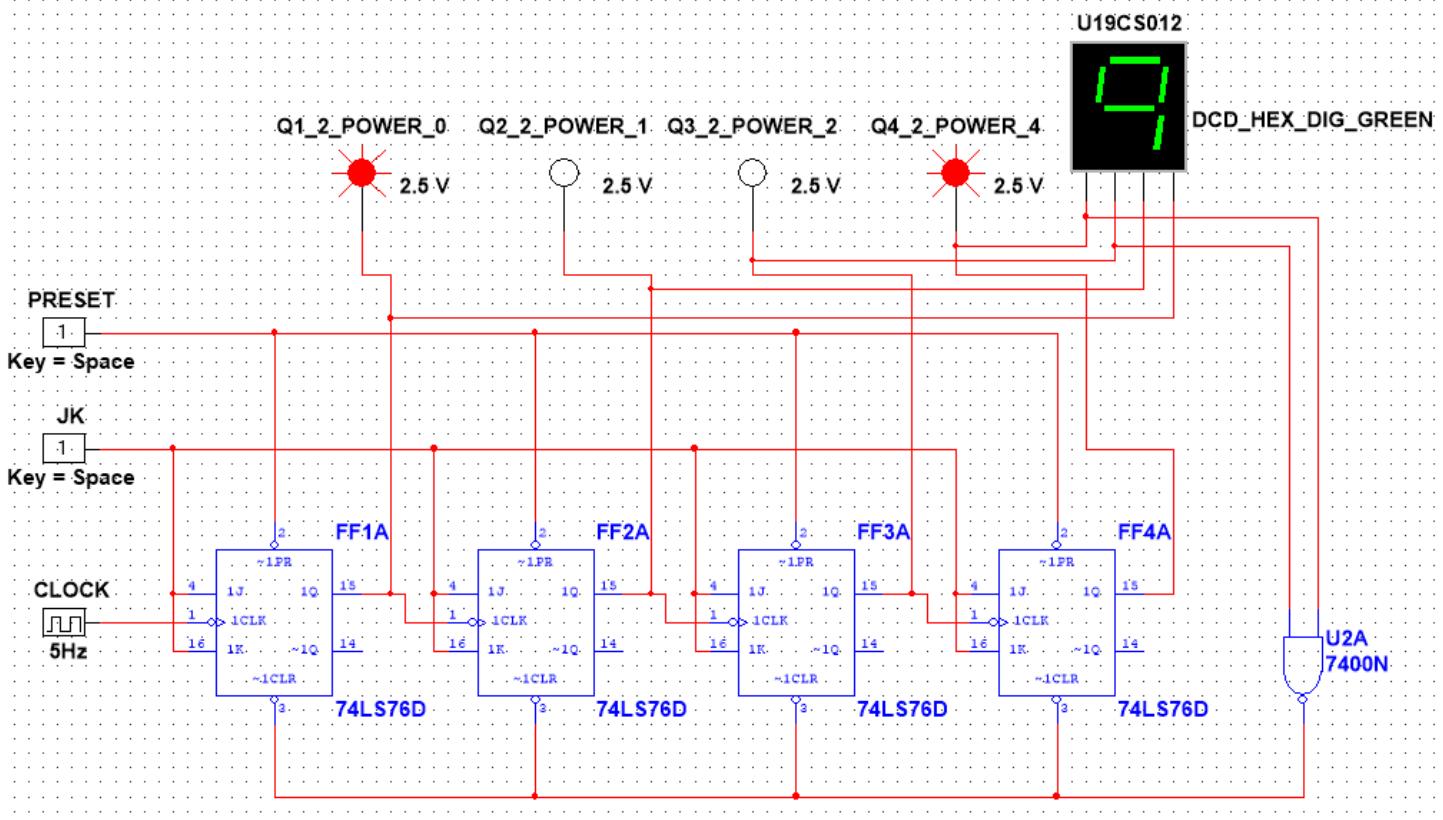


9.) Valid State: 8 [1000]

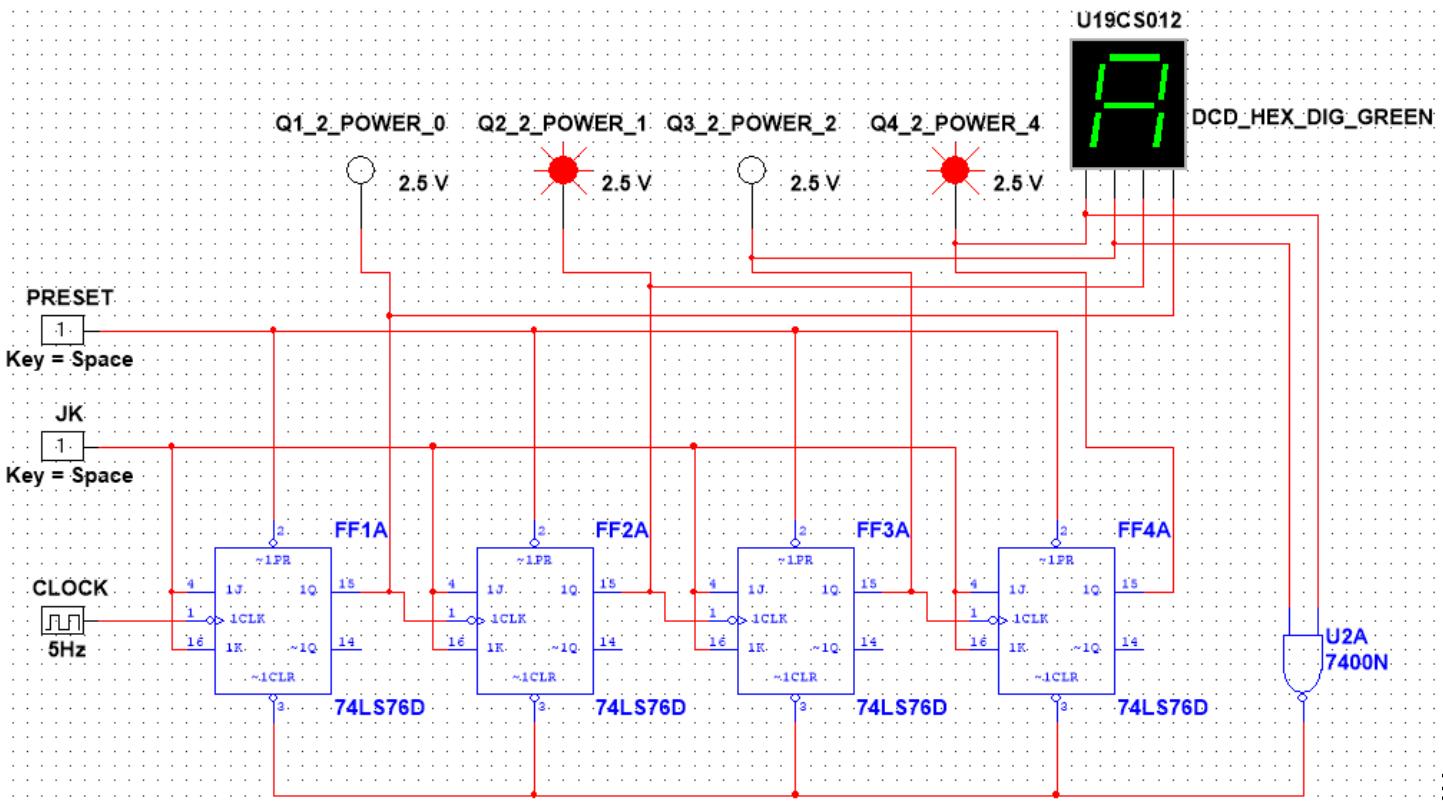




10.) Valid State: 9 [1001]

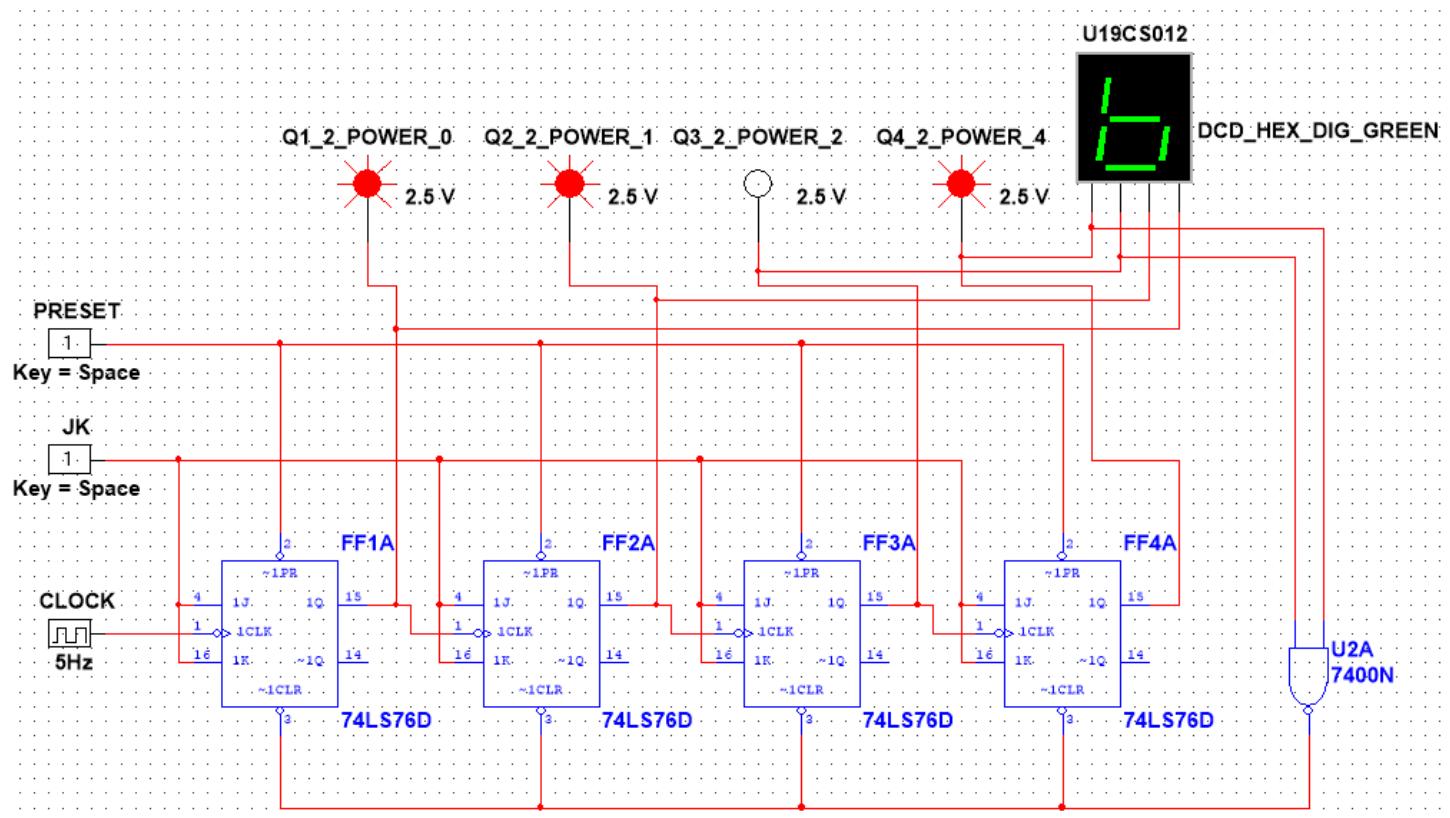


11.) Valid State: 10 [1010]



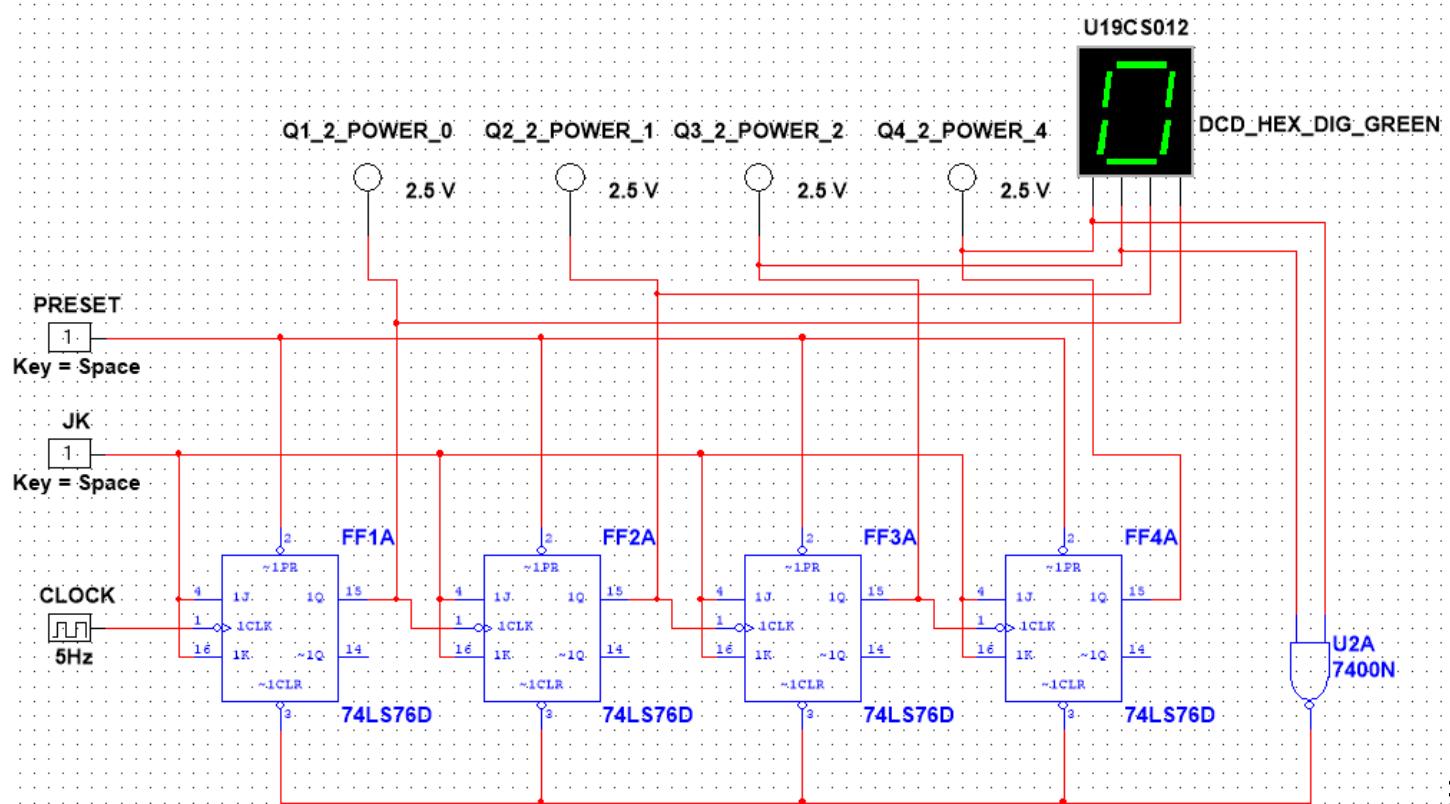


12.) Valid State: 11 [1011]



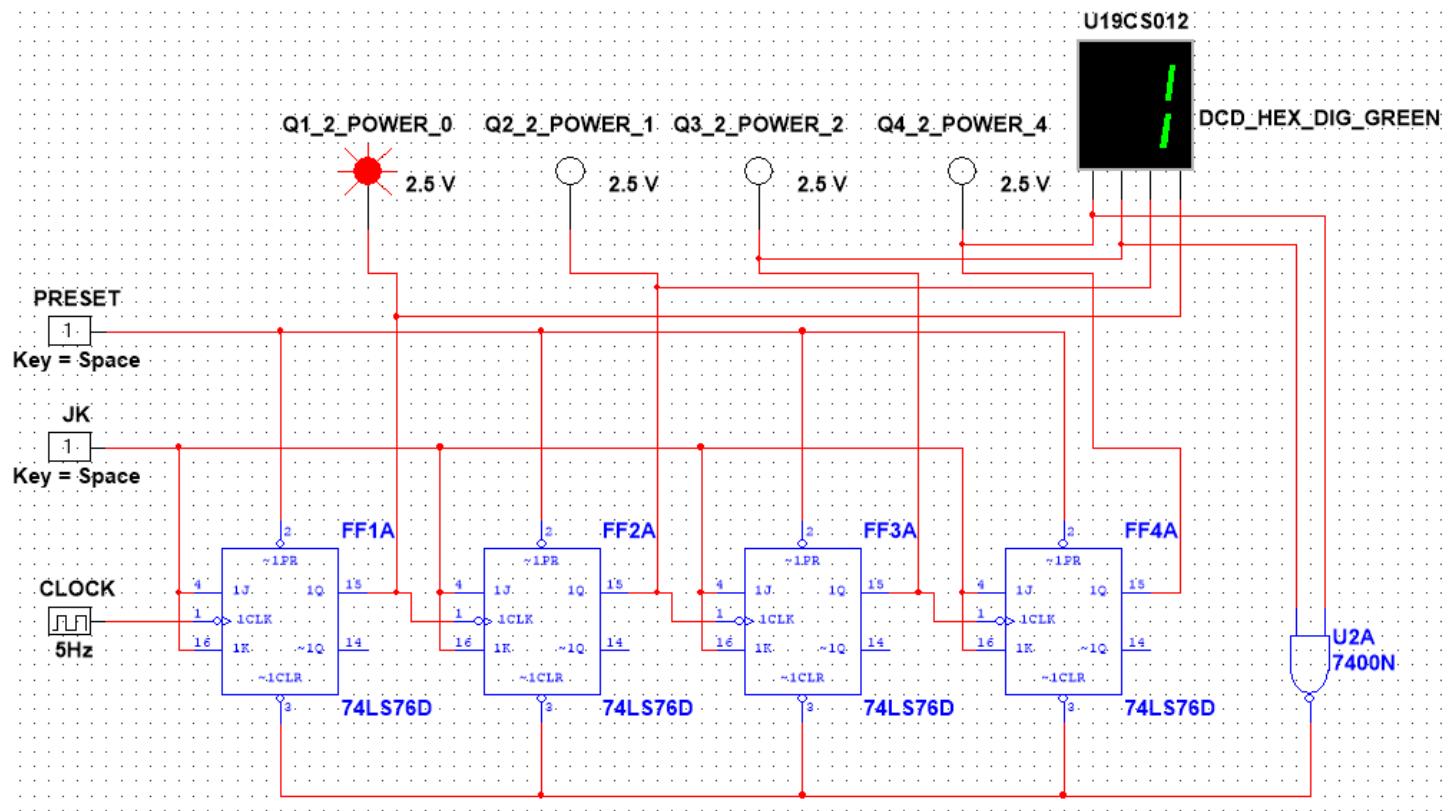
Cycle Starts Repeating Again

13.) Valid State: 12 [1100] -> 0 [0000]

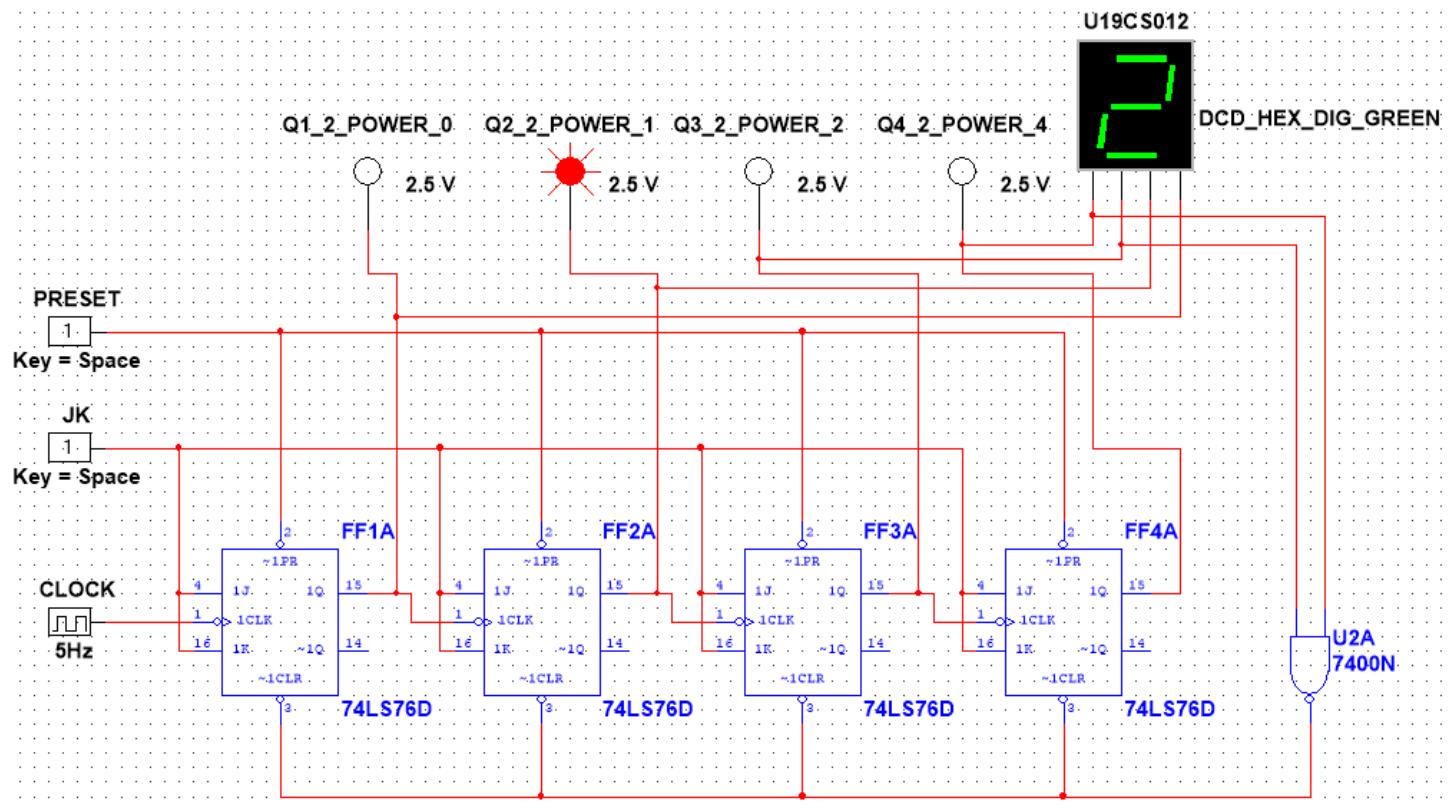




14.) Valid State: 13 [1101]  $\rightarrow$  1 [0001]

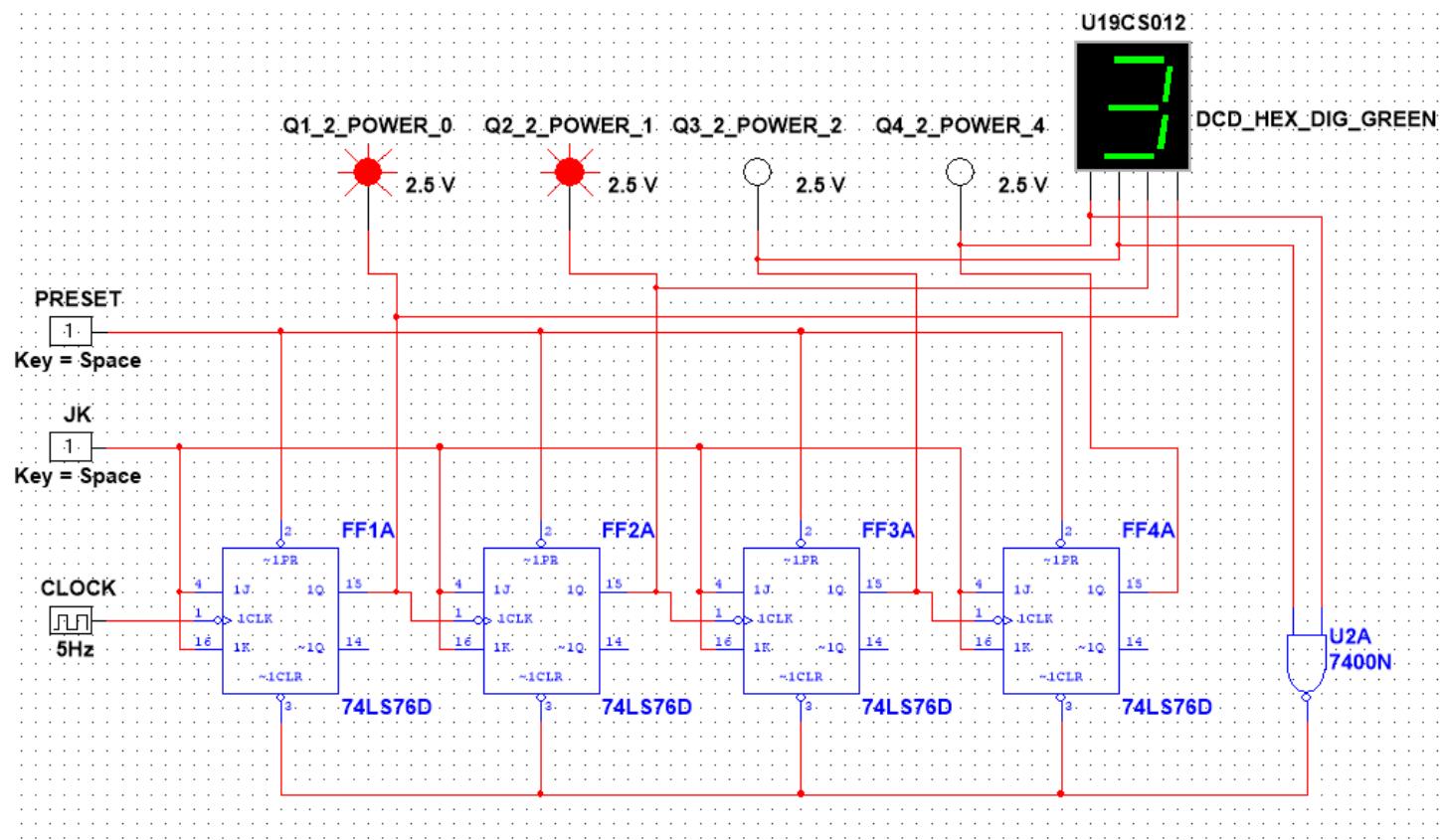


15.) Valid State: 14 [1110]  $\rightarrow$  2 [0010]



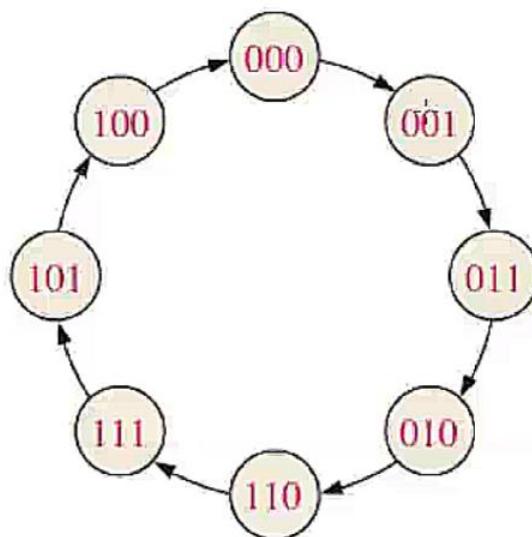


16.) Valid State: 15 [1111]  $\rightarrow$  3 [0011]



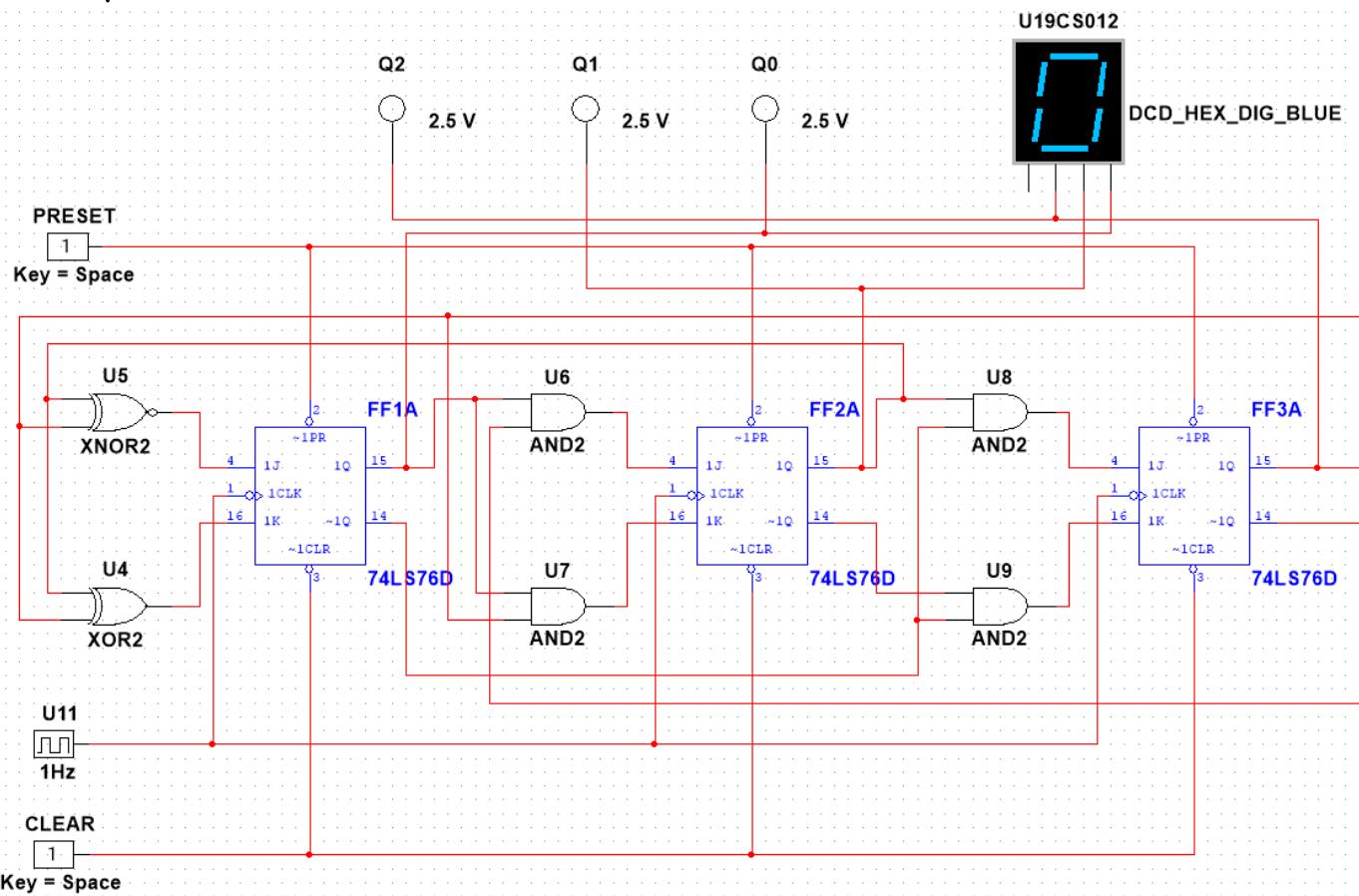
And the Loop Continues...

2.) Design and Implement 3-BIT Gray Code UP Synchronous Counter in Multisim using JK Flip-Flops and Logic Gates.

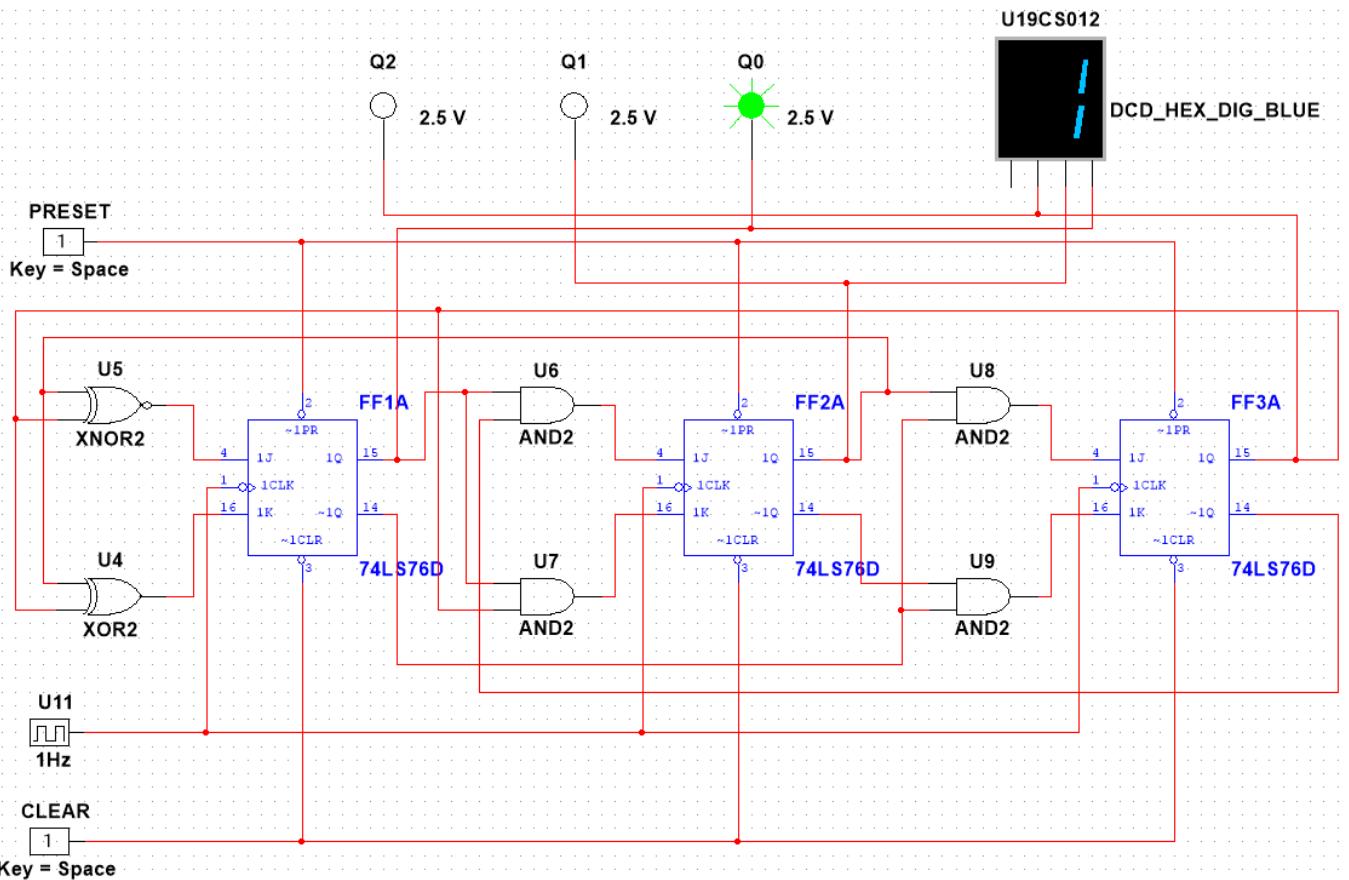




**1.) Gray Code State: 000 [0]**

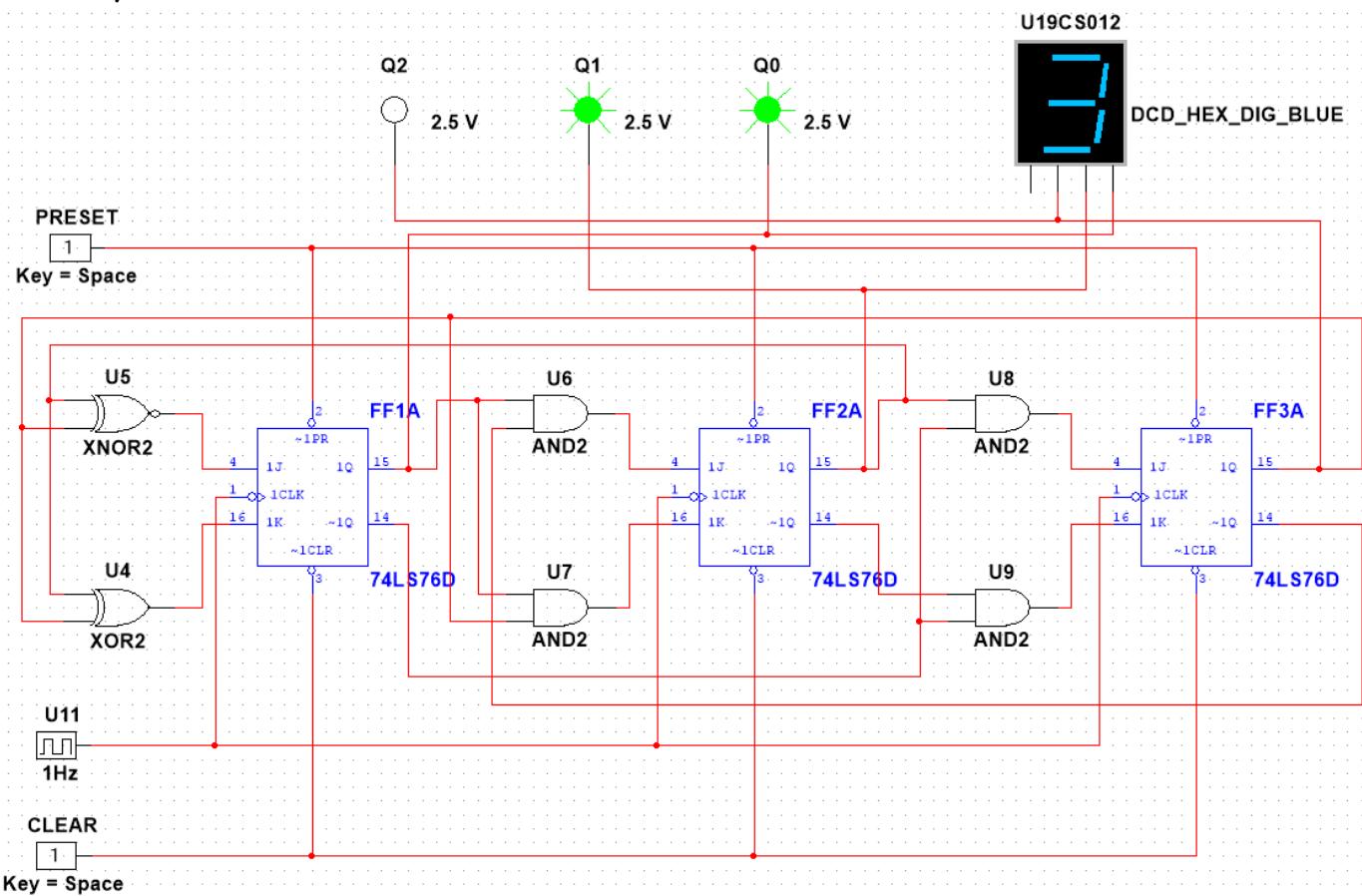


**2.) Gray Code State: 001 [1]**

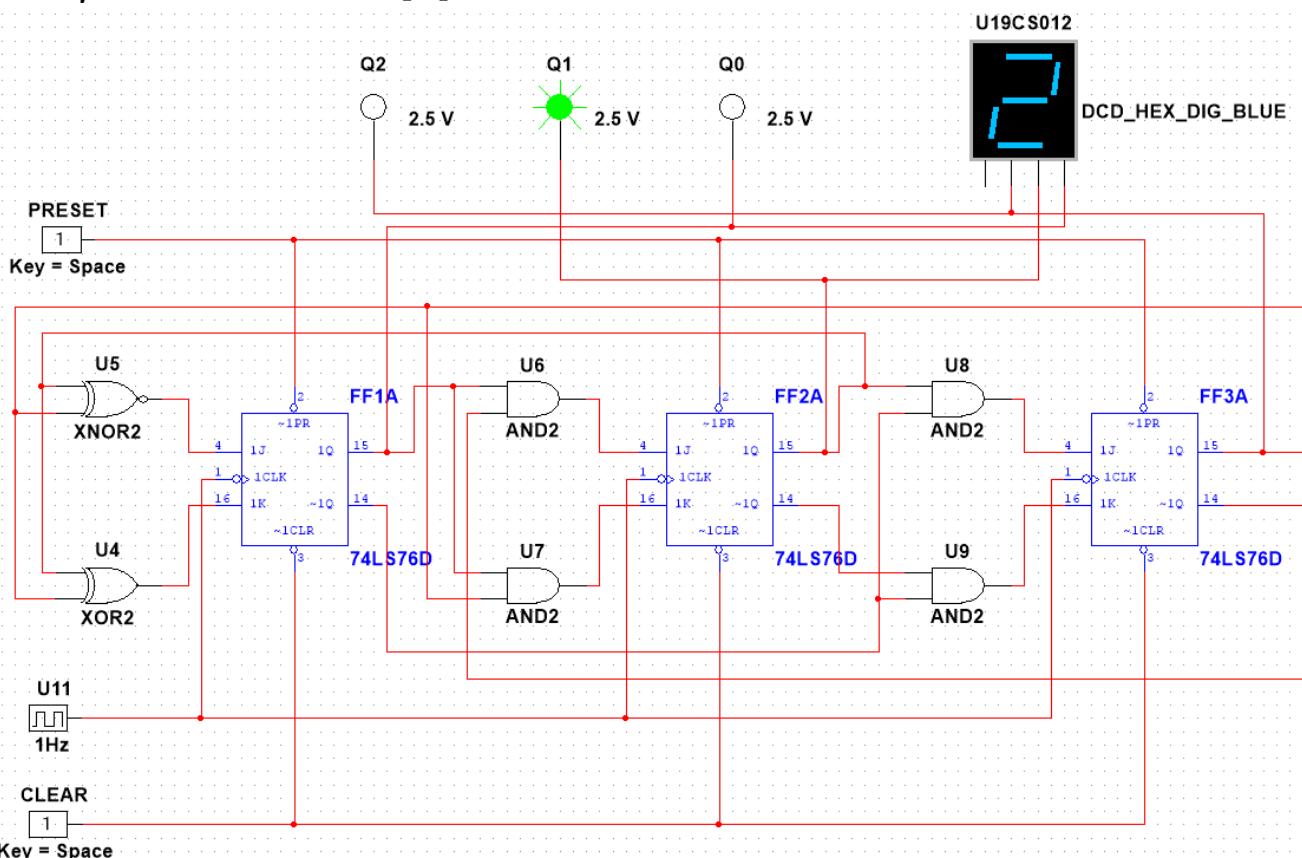




**3.) Gray Code State: 011 [3]**

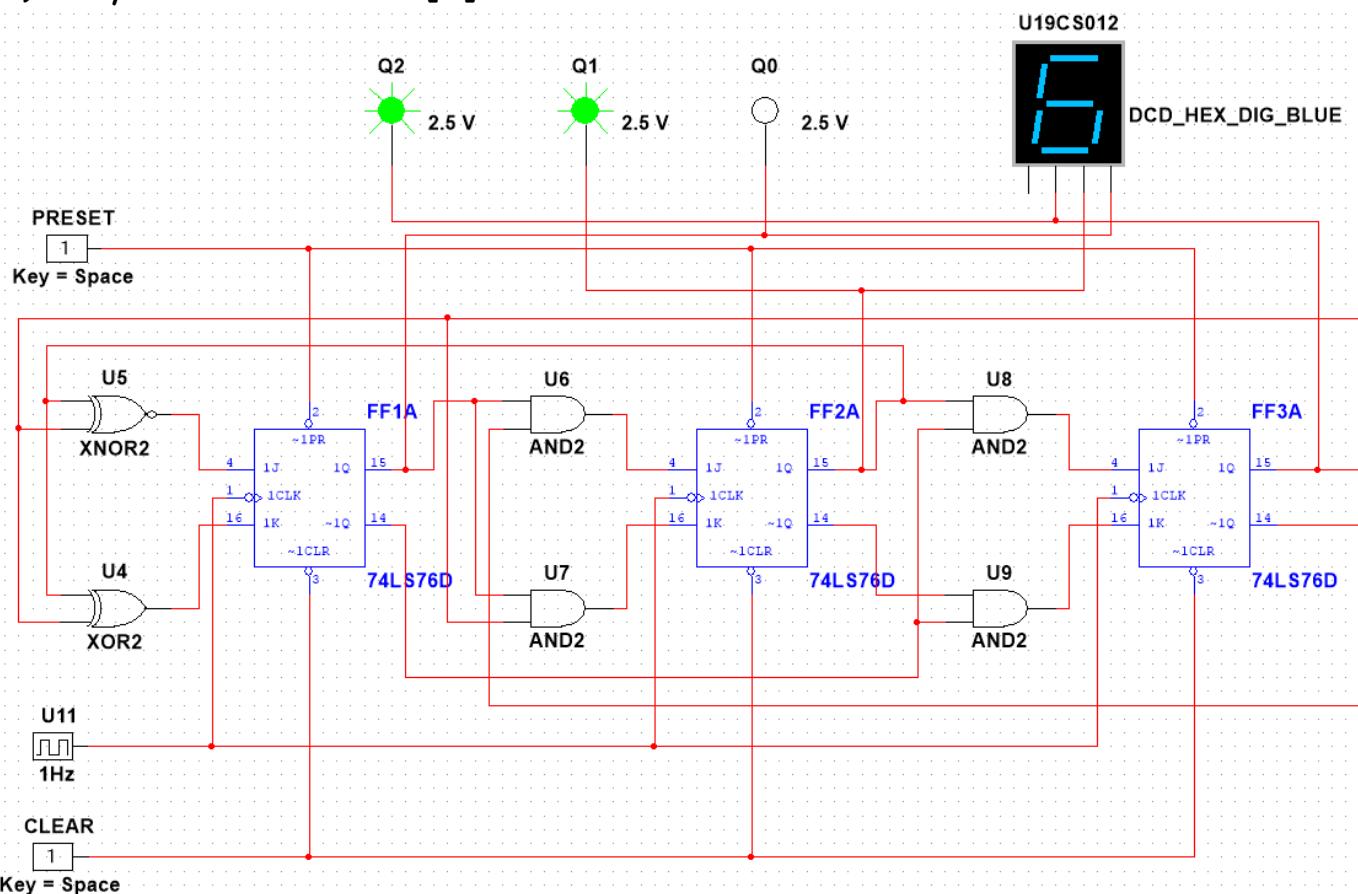


**4.) Gray Code State: 010 [2]**

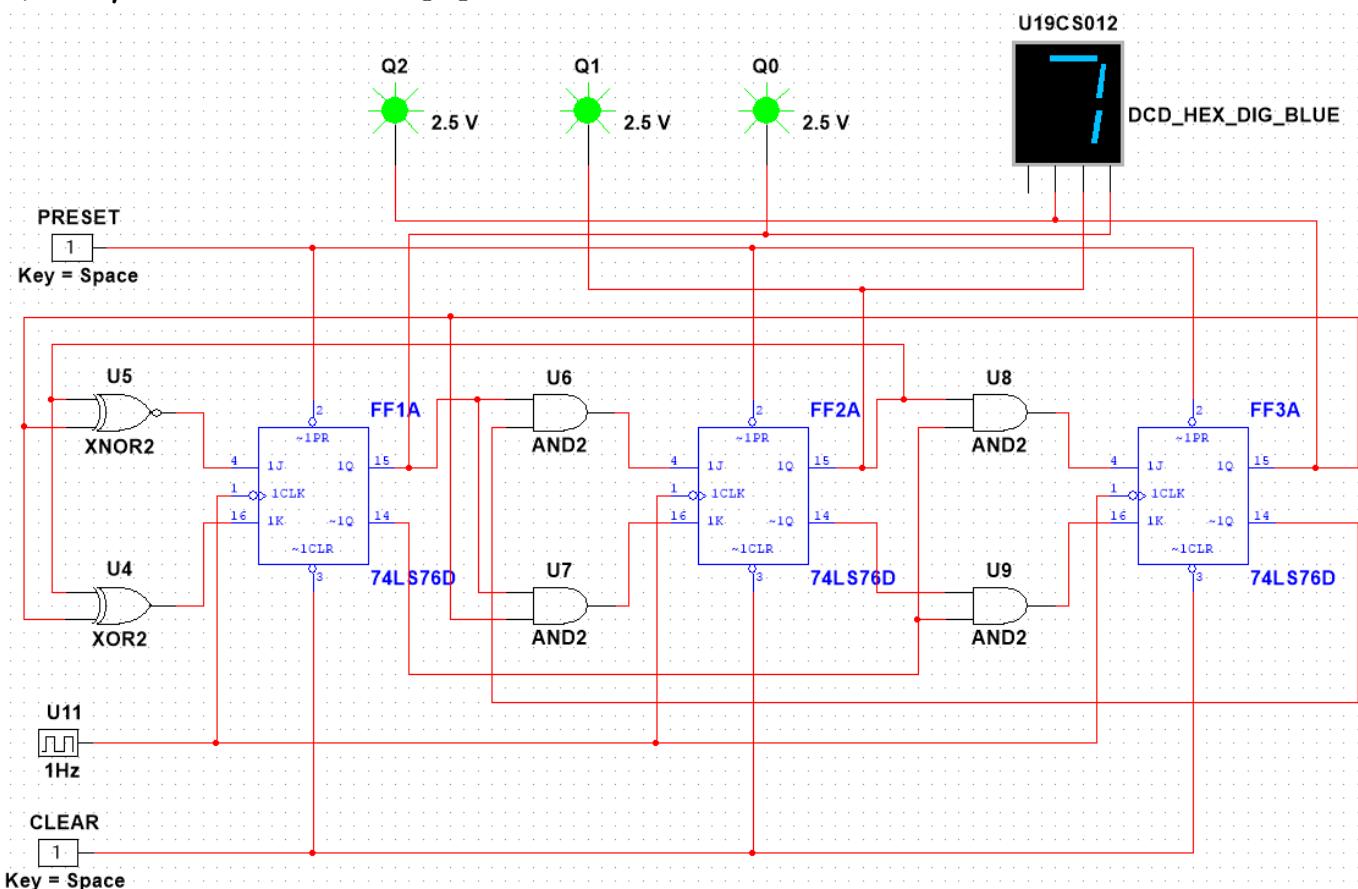




**5.) Gray Code State: 110 [6]**

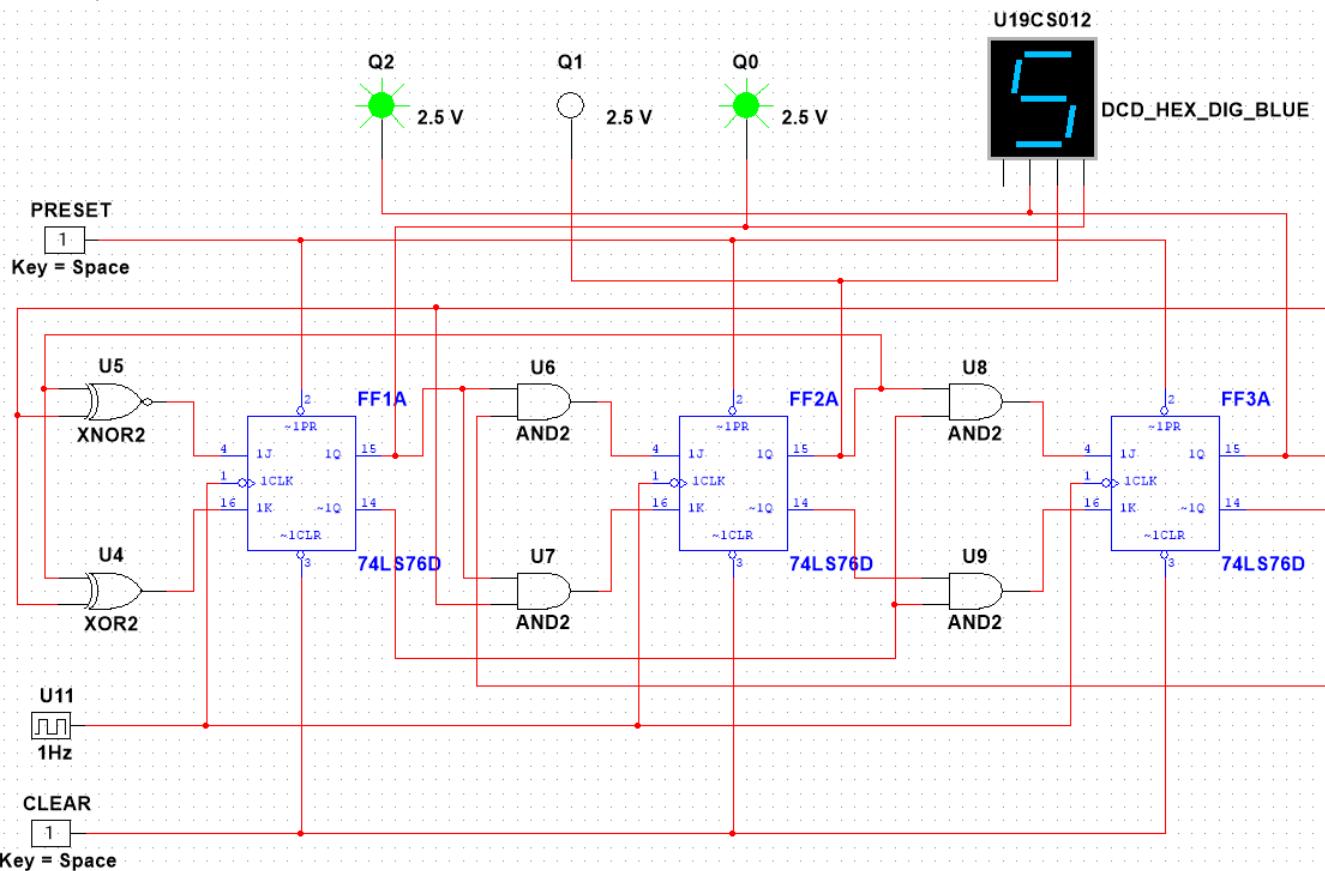


**6.) Gray Code State: 111 [7]**

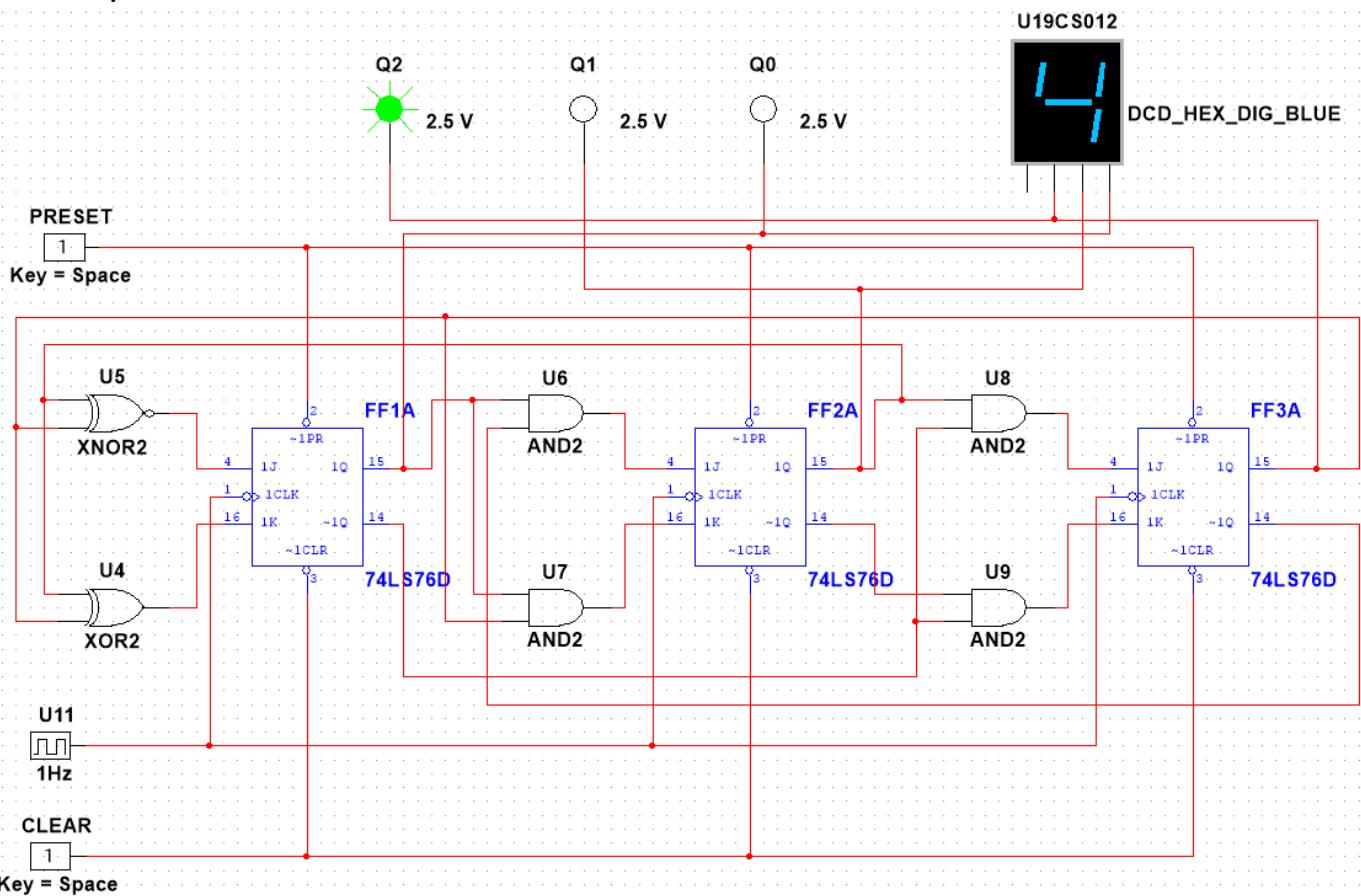




**7.) Gray Code State: 101 [5]**



**8.) Gray Code State: 100 [4]**



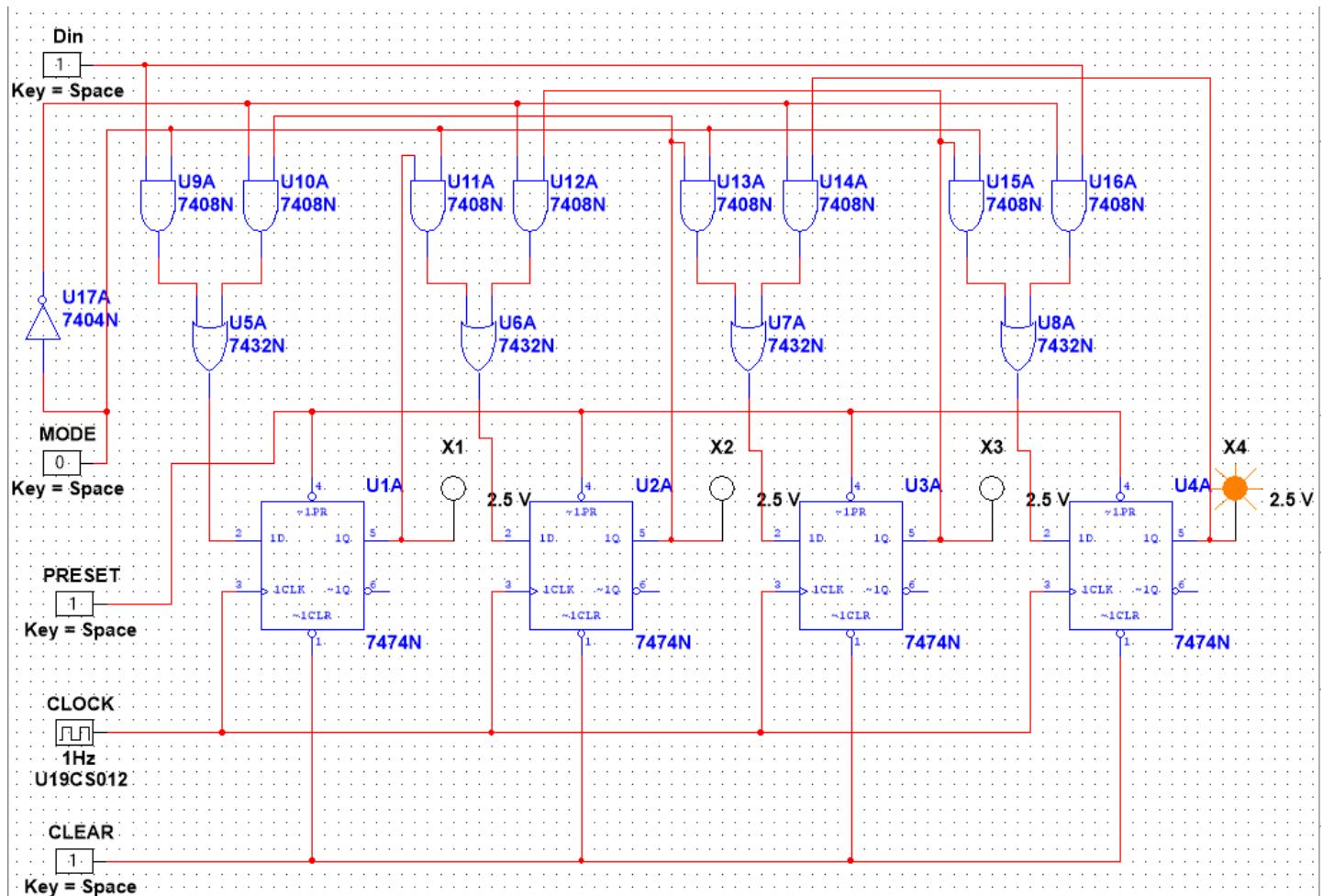


**3.) Design and Implement Bidirectional Shift Registers Using Mode Control in Multisim using JK Flip-Flops and Logic Gates.**

**A1.) Implementation:**

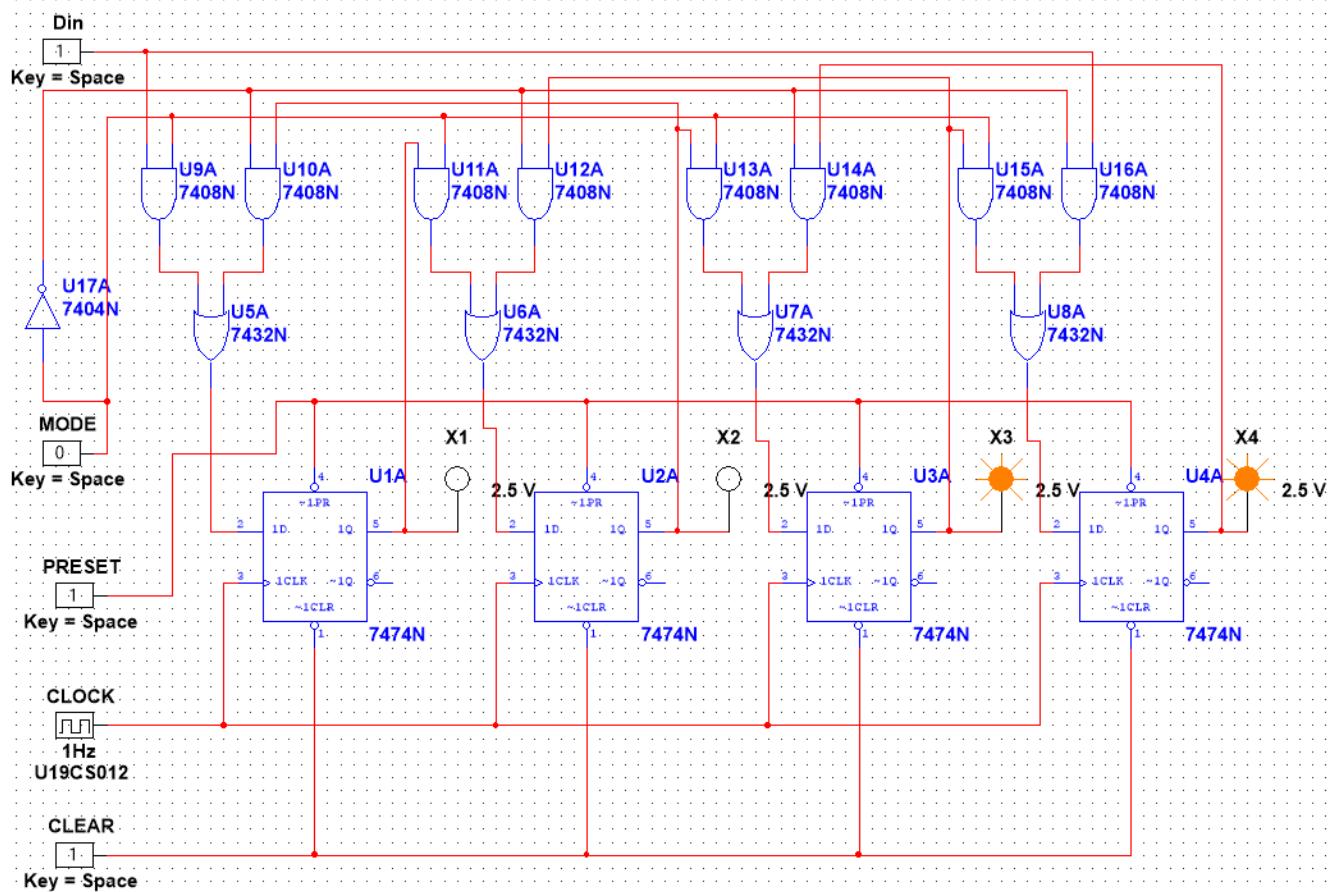
**Mode Control: 0 -> Shift LEFT Register**

**1.) State 1: 0001**

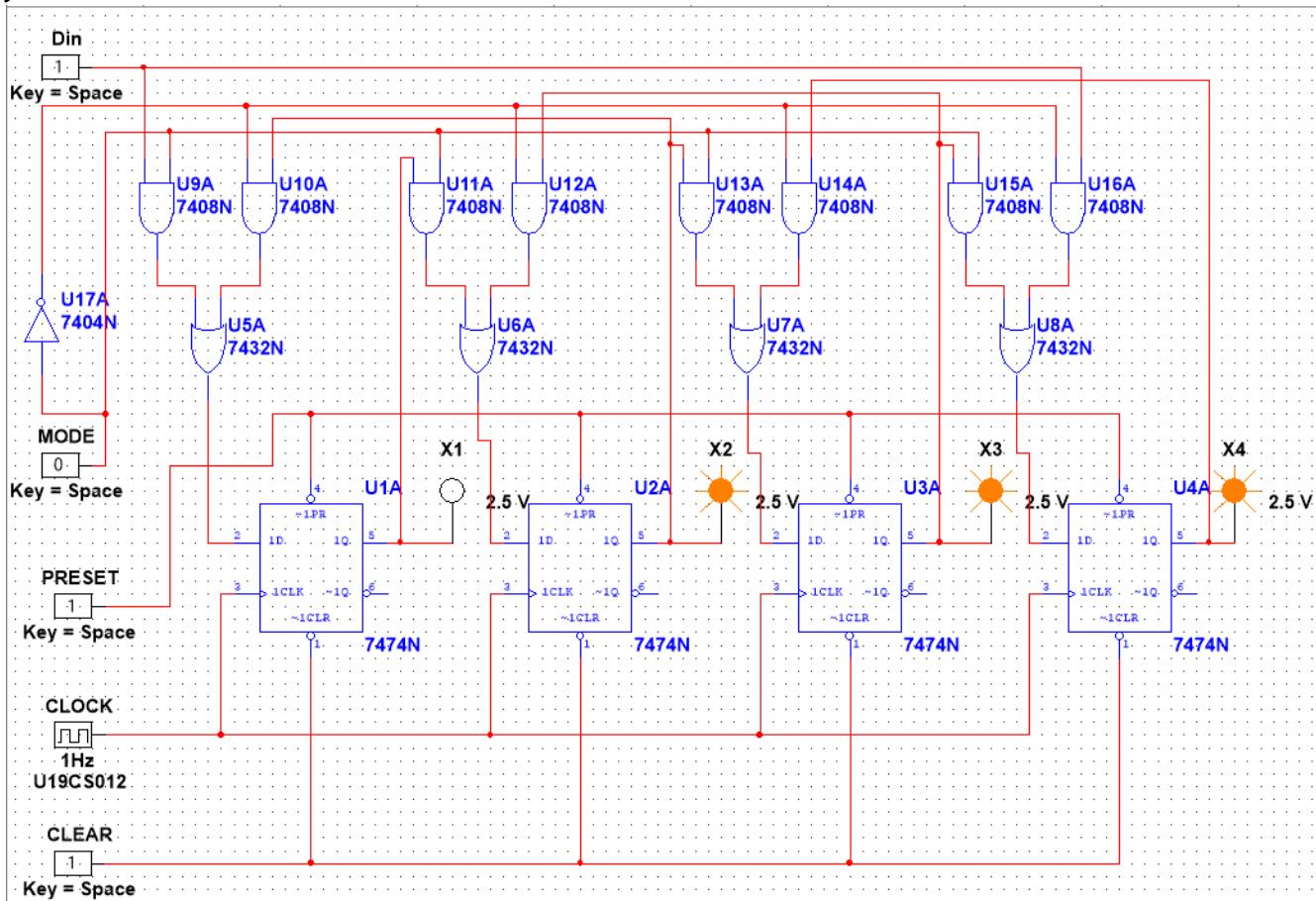




### 2.) State 2: 0011

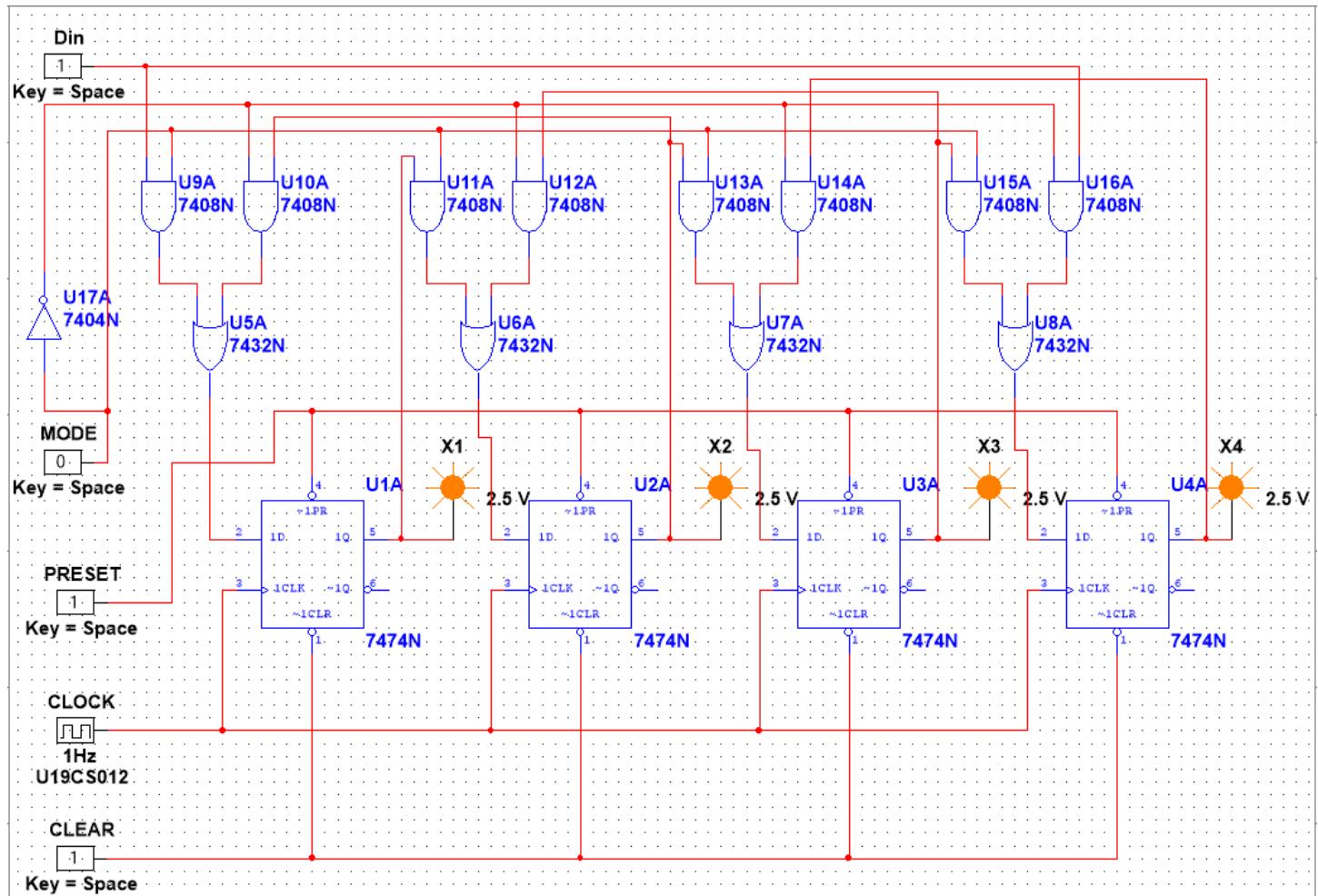


### 3.) State 3: 0111





4.) State 4: 1111

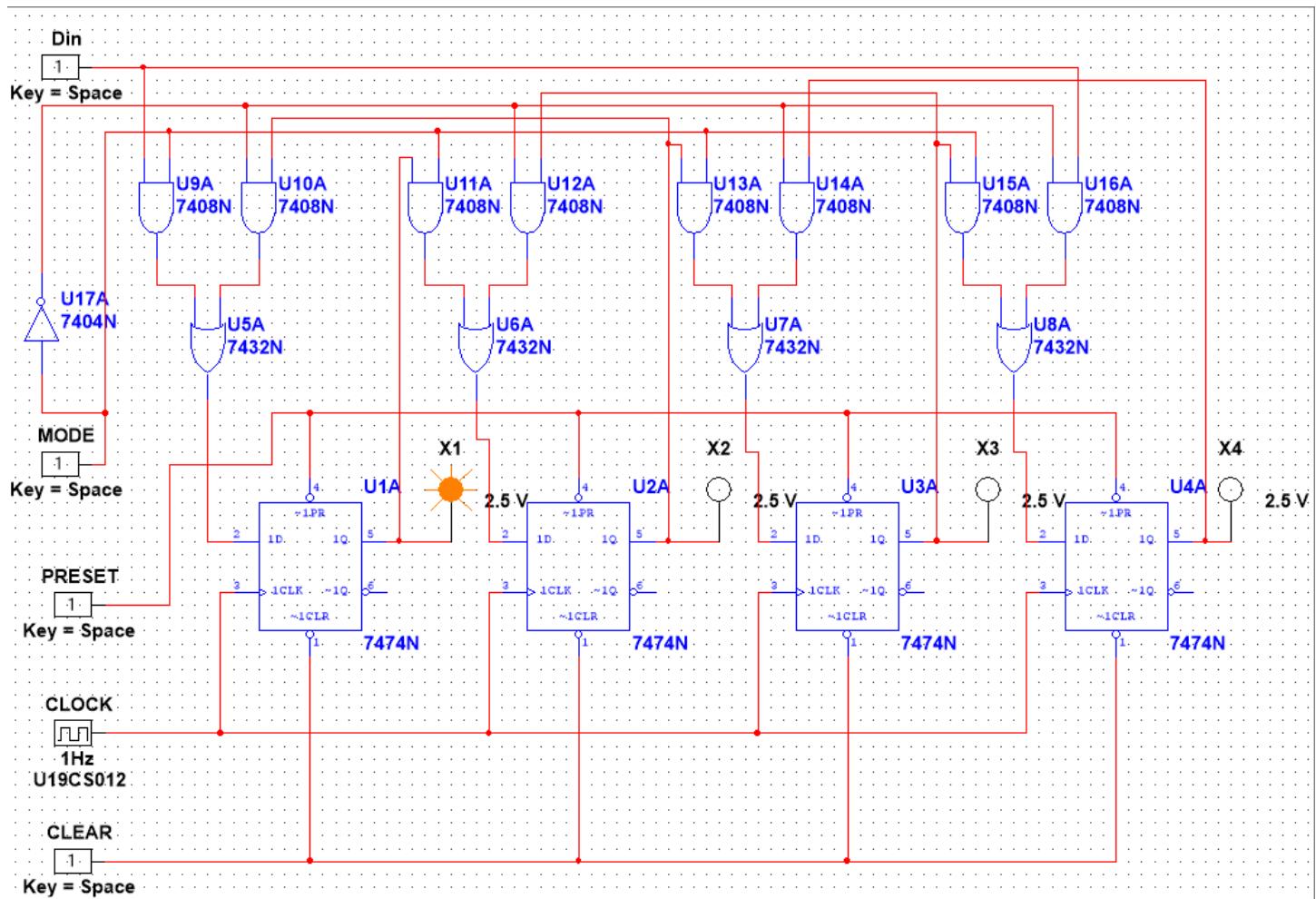




**A2.) Implementation:**

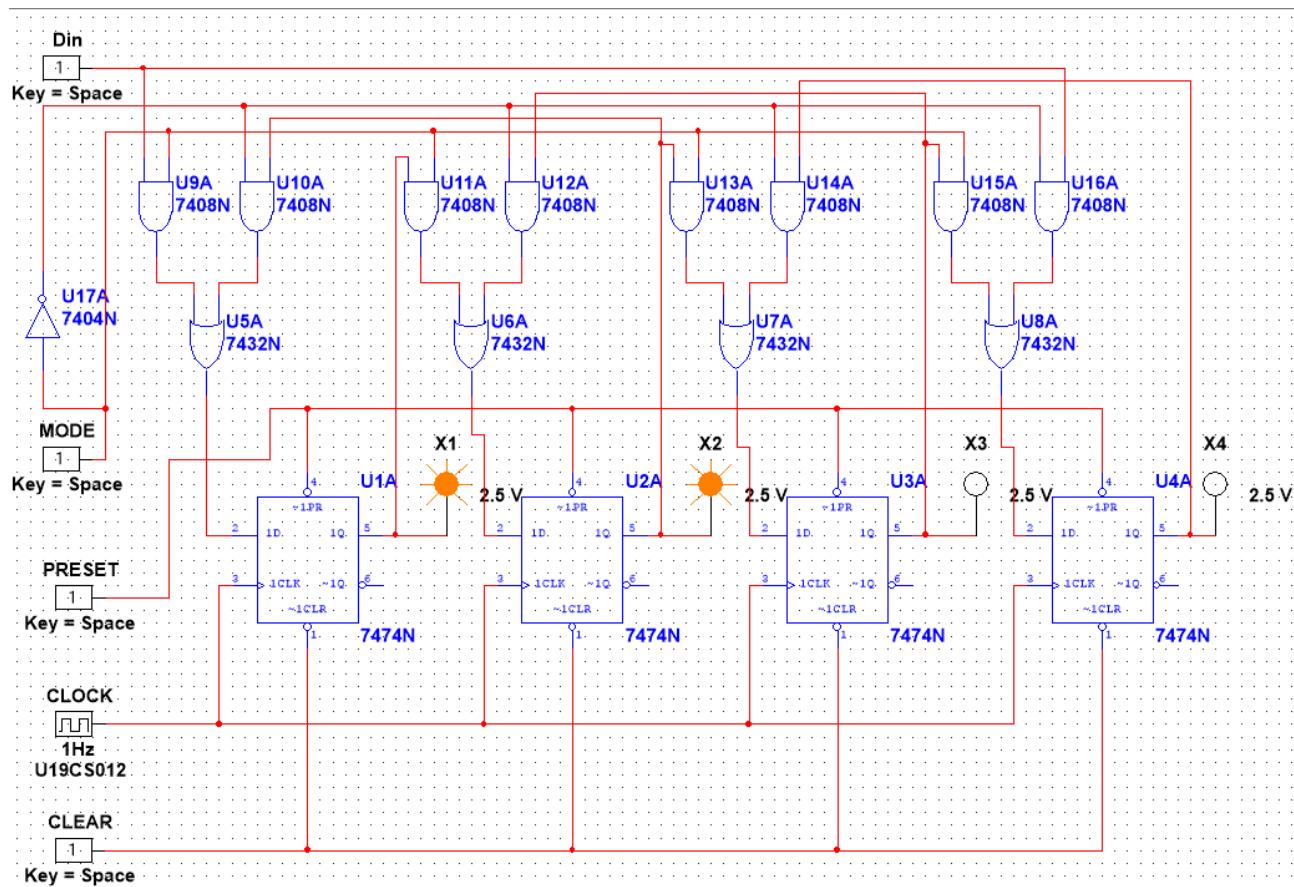
**Mode Control: 1 → Shift Right Register**

1.) State 1: 1000

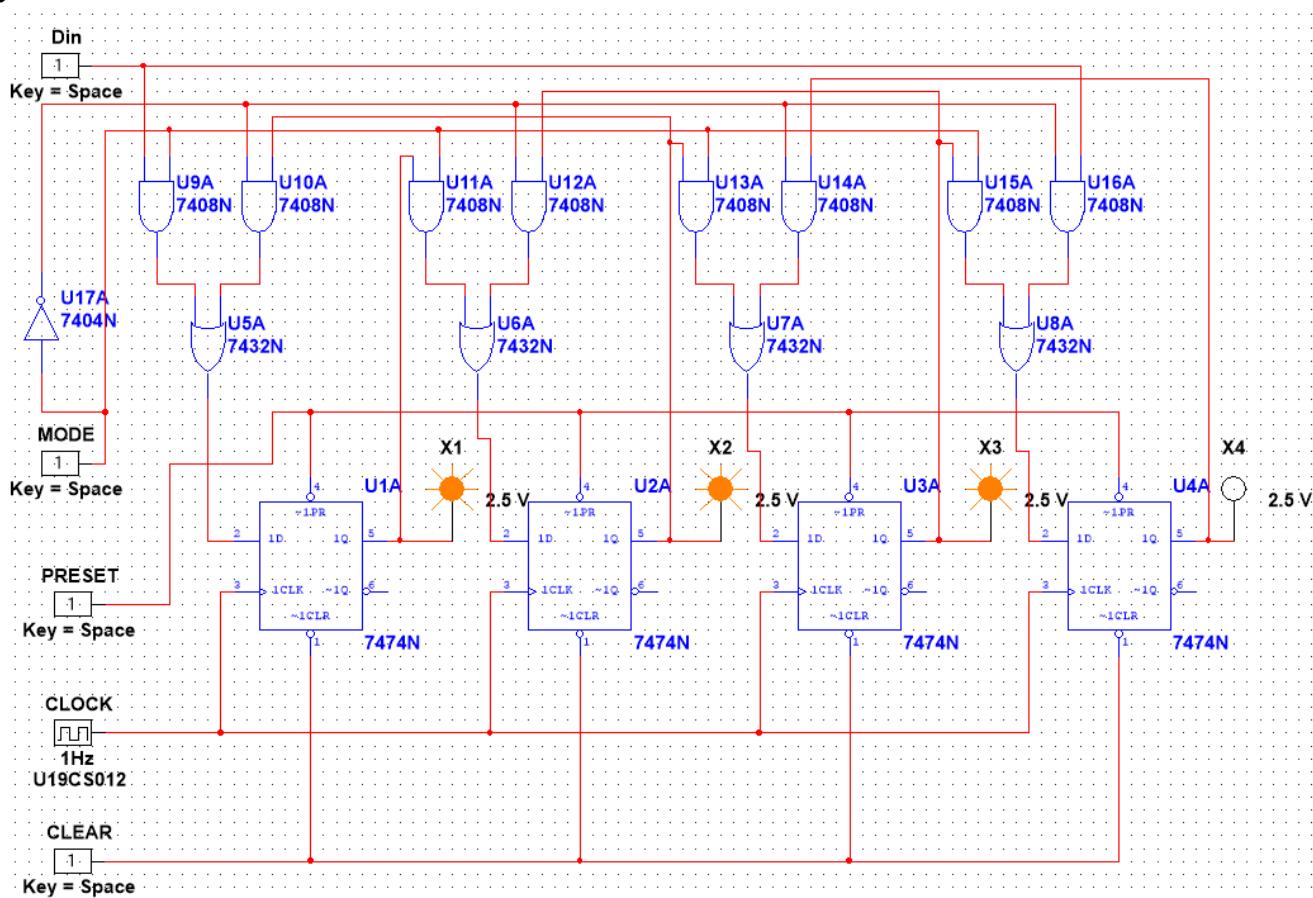




## 2.) State 2: 1100

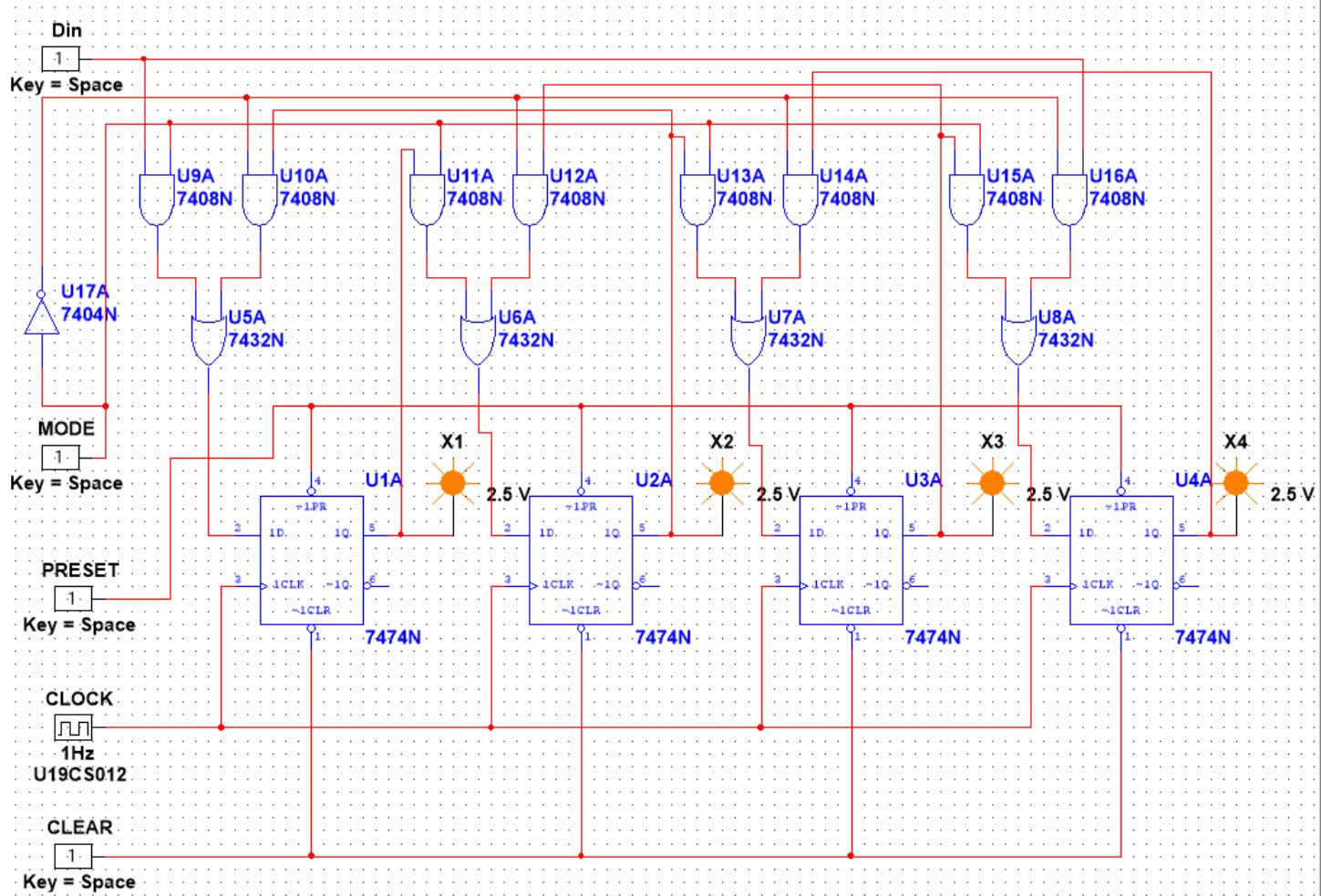


## 3.) State 3: 1110





4.) State 4: 1111



#### D.) CONCLUSION:

We have Successfully Implemented MOD-12 Counter, Synchronous Gray Counter and Bi-directional Shift Registers [Using Mode Control] with the Help of JK Flip-flop and Logic Gates and verified our MULTISIM Outputs and Results from Theoretical Knowledge of these Circuits taught in DELD Classes.



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:****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.

**Gray Code:**

**Gray code** - also known as **Cyclic Code**, **Reflected Binary Code (RBC)**, **Reflected Binary (RB)** or **Grey code** - is defined as an ordering of the binary number system such that each incremental value can only differ by one bit. In gray code, while traversing from one step to another step only one bit in the code group changes. That is to say that two adjacent code numbers differ from each other by only one bit.

**Binary to Gray Code Converter:**

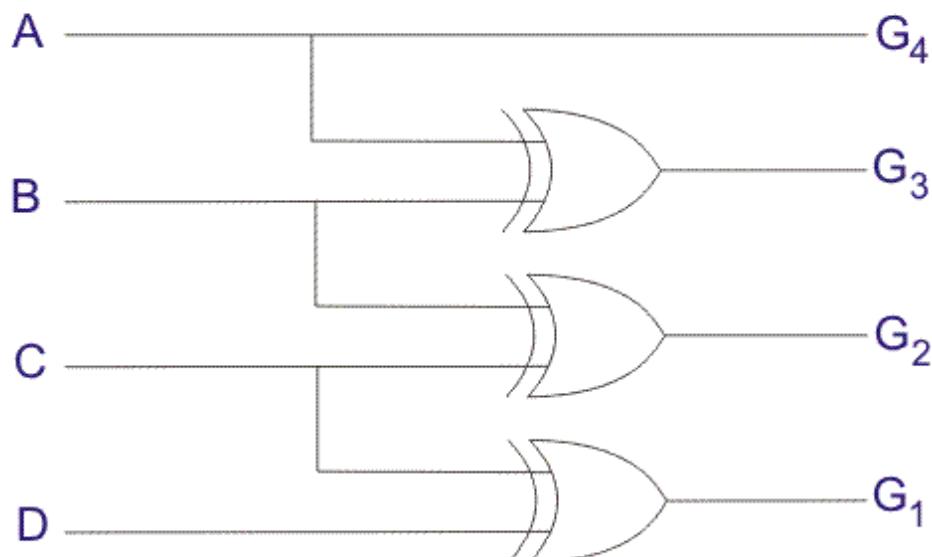
The logical circuit which converts the binary code to equivalent gray code is known as **binary to gray code converter**. An n-bit gray code can be obtained by reflecting an n-1 bit code about an axis after  $2^{n-1}$  rows and putting the MSB (Most Significant Bit) of 0 above the axis and the MSB of 1 below the axis. Reflection of Gray codes is shown below.

The 4 bit binary to gray code conversion table is given below:

Decimal Number	4 bit Binary Number <u>ABCD</u>	4 bit Gray Code <u>G<sub>1</sub>G<sub>2</sub>G<sub>3</sub>G<sub>4</sub></u>
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

**How to Convert Binary to Gray Code:**

1. The MSB (Most Significant Bit) of the gray code will be exactly equal to the first bit of the given binary number.
2. The second bit of the code will be exclusive-or (XOR) of the first and second bit of the given binary number, i.e if both the bits are same the result will be 0 and if they are different the result will be 1.
3. The third bit of gray code will be equal to the exclusive-or (XOR) of the second and third bit of the given binary number. Thus the binary to gray code conversion goes on. An example is given below to illustrate these steps.

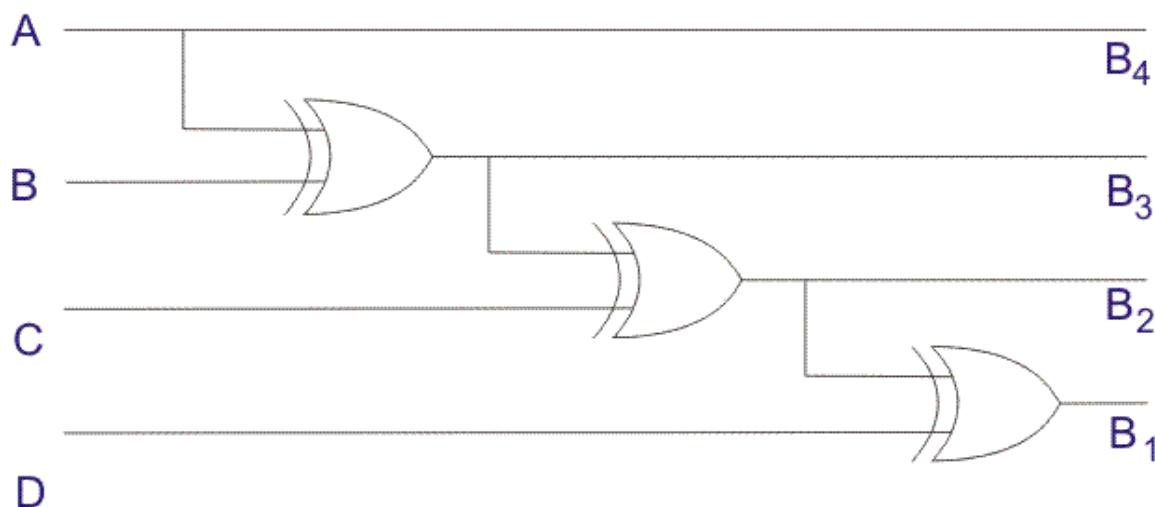
**Logic Circuit for Binary to Gray Code Converter**

**Gray to Binary Code Converter:**

In a gray to binary code converter, the input is gray code and output is its equivalent binary code.

4 bit Gray Code	4 bit Binary Code
A B C D	$B_4 \ B_3 \ B_2 \ B_1$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

The gray code to binary converter circuit is shown below:



Logic Circuit for Gray to Binary Code Converter



### Gray Code to Binary Conversion:

1. The MSB of the binary number will be equal to the MSB of the given gray code.
2. Now if the second gray bit is 0, then the second binary bit will be the same as the previous or the first bit. If the gray bit is 1 the second binary bit will alter. If it was 1 it will be 0 and if it was 0 it will be 1.
3. This step is continued for all the bits to do Gray code to binary conversion.

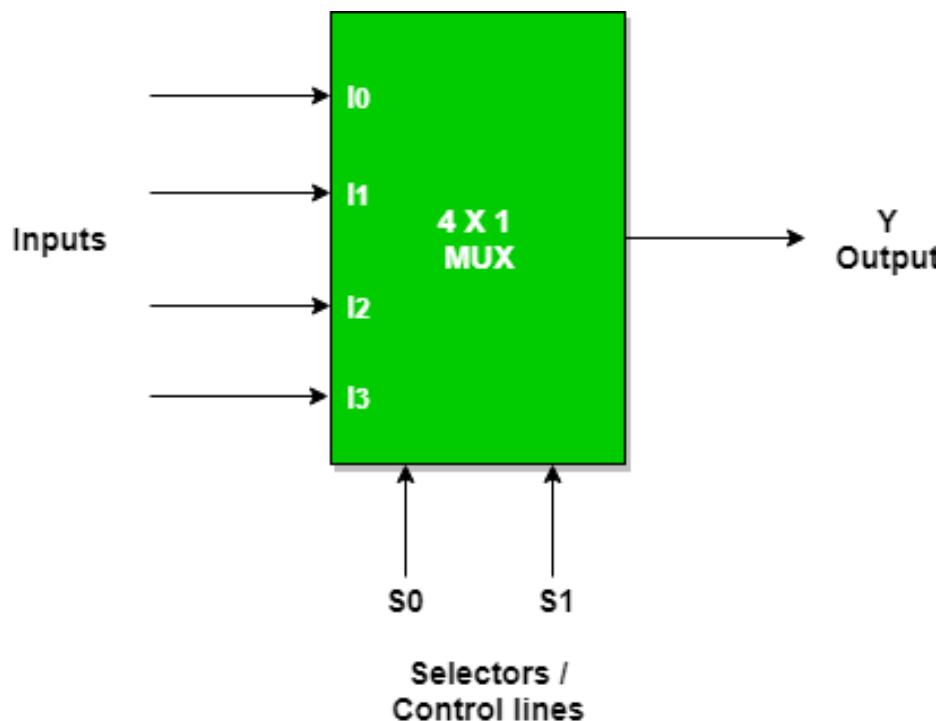
### Multiplexers in Digital Logic:

It is a combinational circuit which have many data inputs and single output depending on control or select inputs.

For  $N$  input lines,  $\log n$  (base2) selection lines, or we can say that for  $2^n$  input lines,  $n$  selection lines are required.

Multiplexers are also known as "Data n selector, parallel to serial convertor, many to one circuit, universal logic circuit".

Multiplexers are mainly used to increase amount of the data that can be sent over the network within certain amount of time and bandwidth.

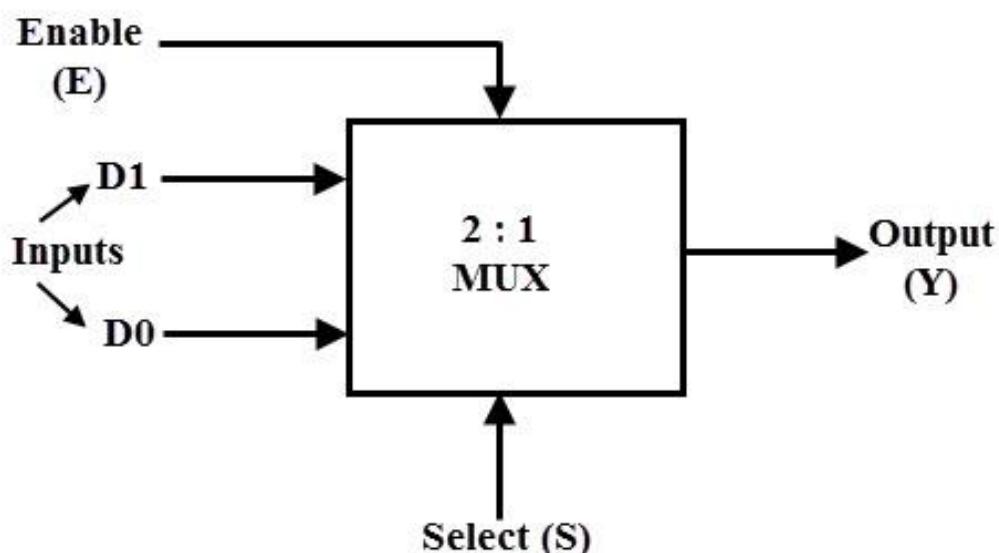


**2 to 1 Multiplexer:**

2 to 1 means that this multiplexer has 2 input channels and 1 output. 2 channels mean it has 1 control signal.

When the control signal is "0", the first channel is selected and the 2<sup>nd</sup> channel is selected when the control signal is "1".

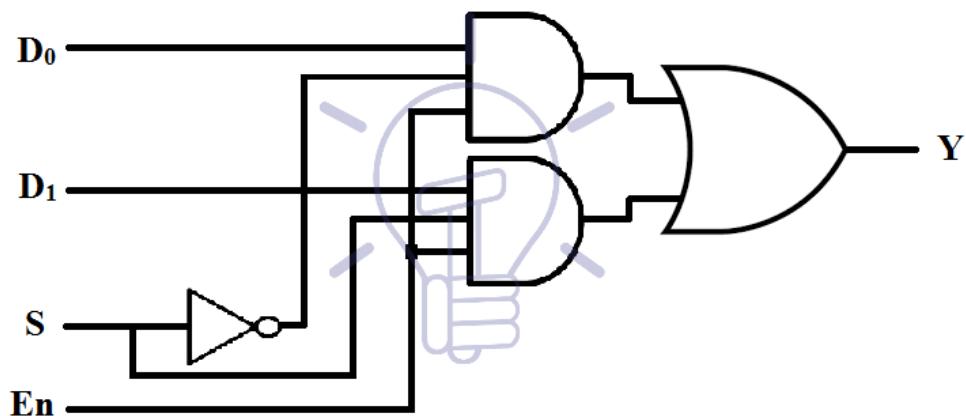
There is also an Enable bit used for enabling/disabling the circuit. When enable is high, MUX is enabled. When Enable pin is Low, MUX is disabled.



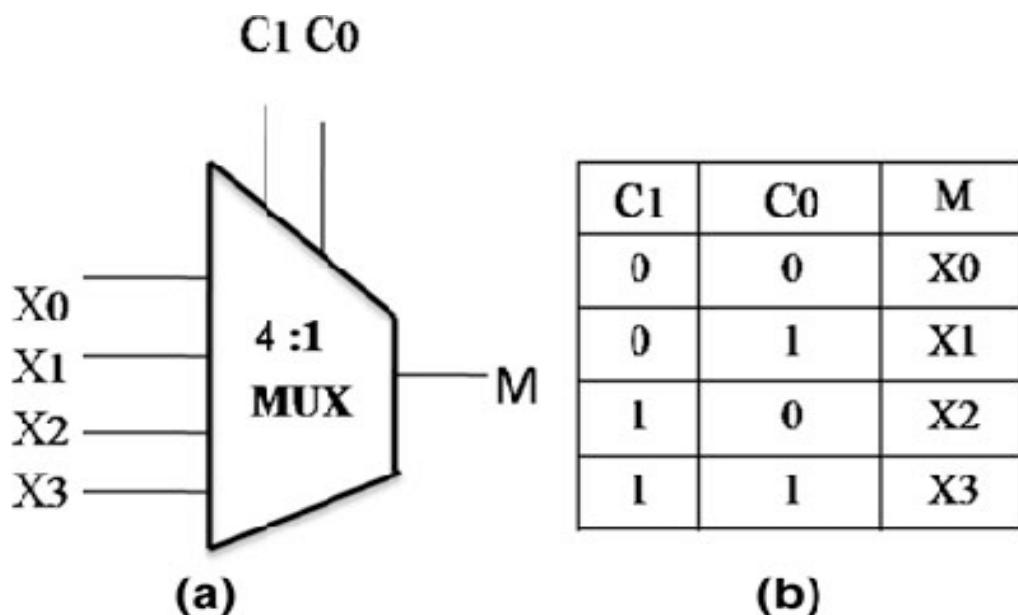
Select	Inputs		Output
0	0	0	0
0	0	1	1
1	1	0	1
1	1	1	1

A MUX need AND gates equal to the number of input channels, NOT gates equal to the number of Control signals and a single OR gate.

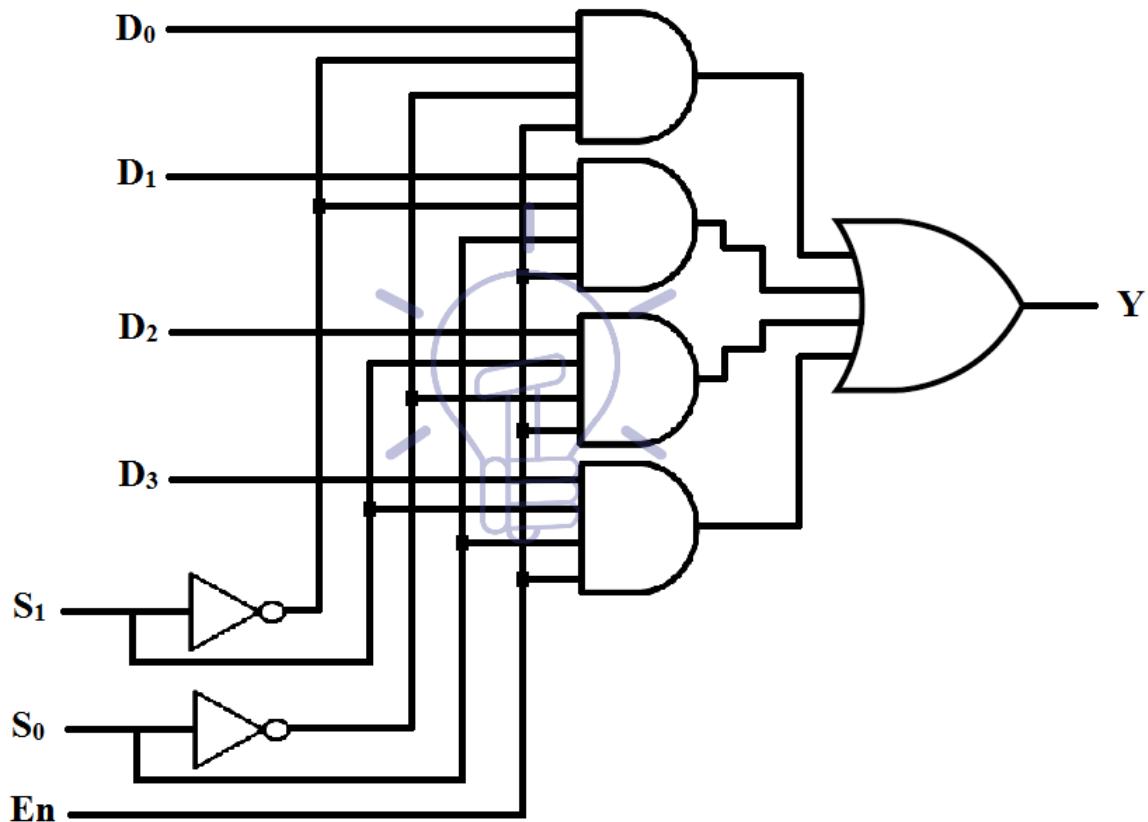
Implantation of Multiplexer using logic gates is given below.

**4 to 1 Multiplexer:**

This multiplexer has 4 input channels, 1 output, and 2 control signals. Each binary combination of control signal will select one out of four input channels.



4 to 1 multiplexer implementation using logic gates is shown in the figure given below.



### Application of Multiplexer:

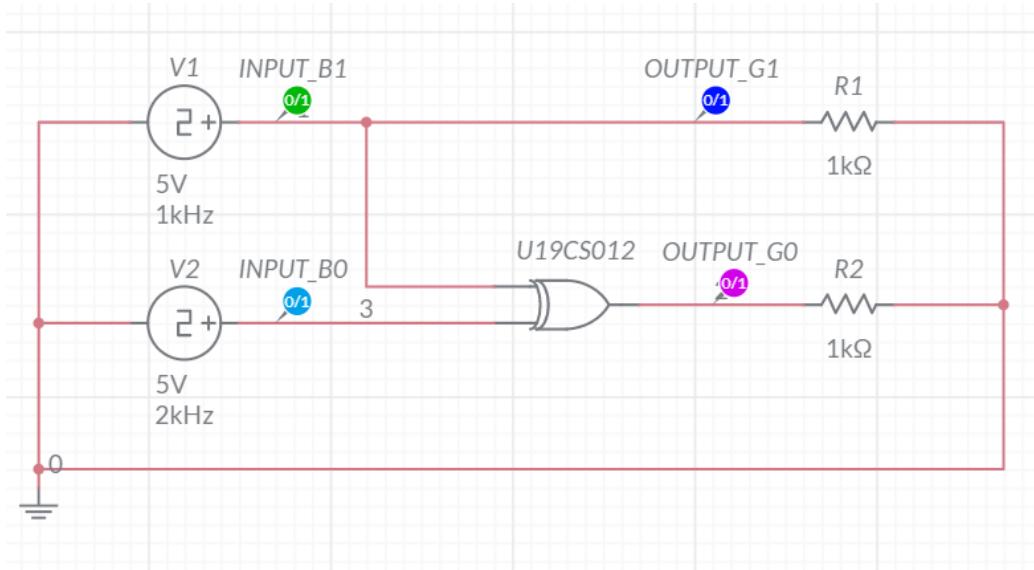
In all types of digital system applications, multiplexers find its immense usage. Since these allows multiple inputs to be connected independently to a single output, these are found in variety of applications including data routing, logic function generators, control sequencers, parallel-to-serial converters,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



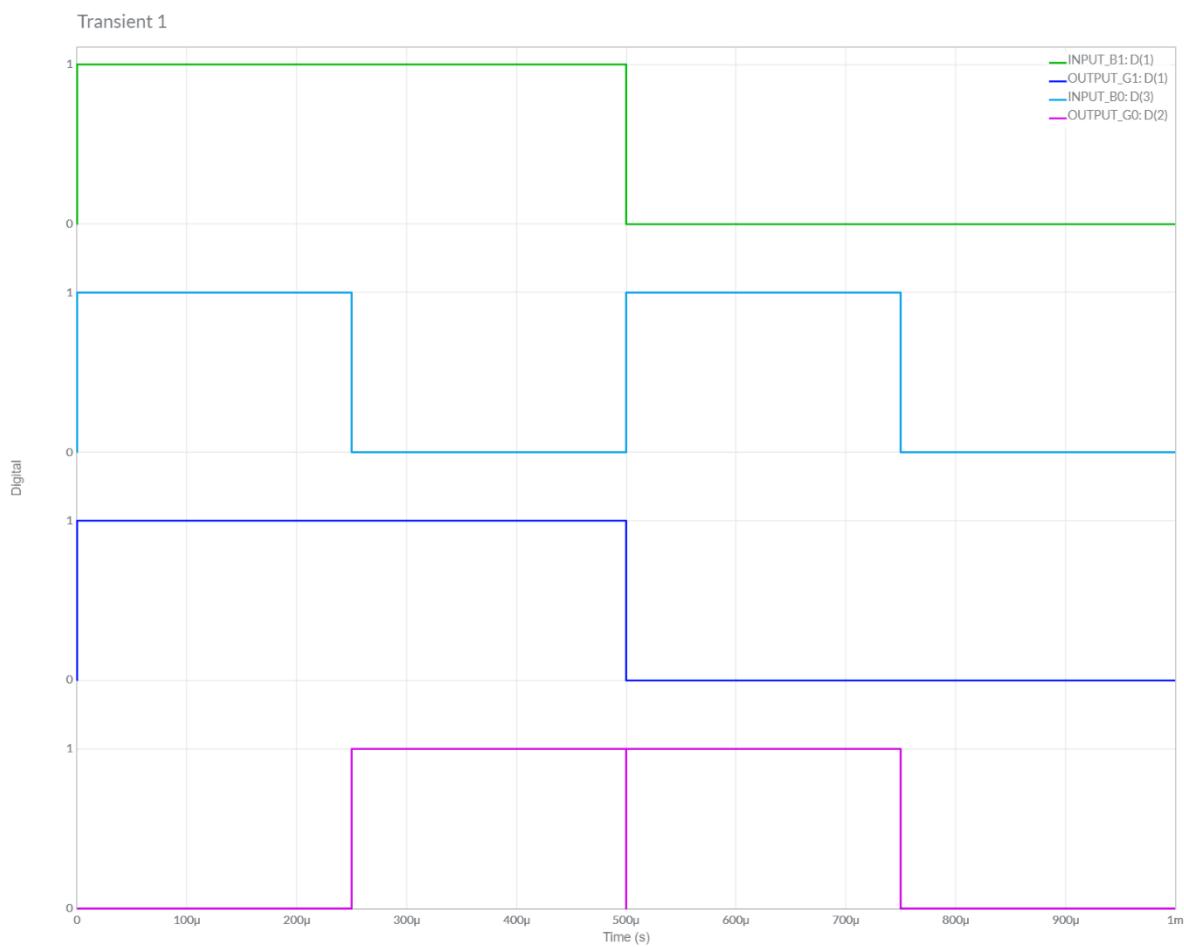
**SIMULATION SCREENSHOTS**

**2 - BIT Binary to Gray code converter**

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



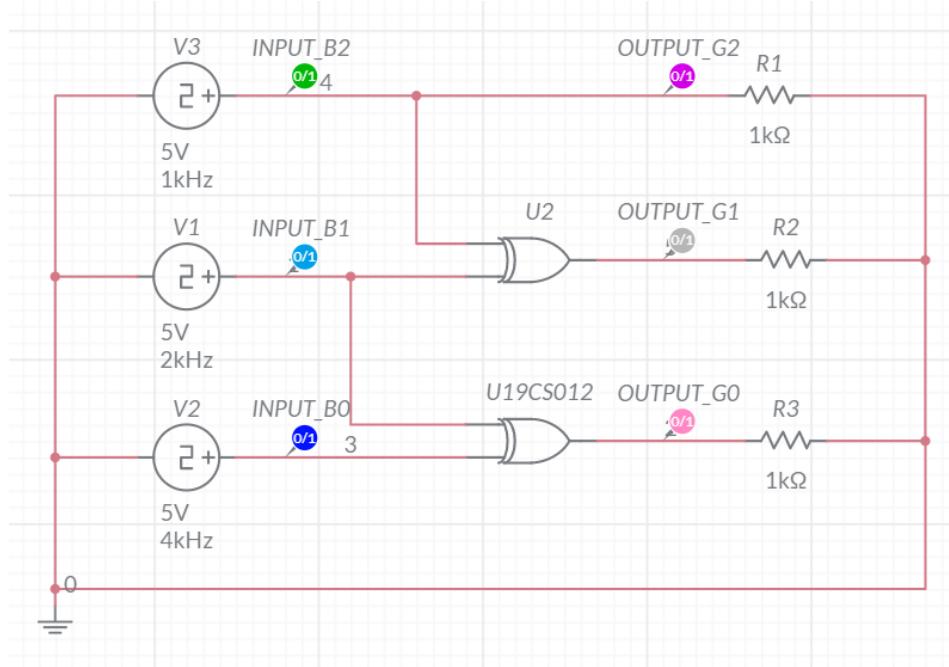
**WAVEFORMS (FROM MULTISIM)**



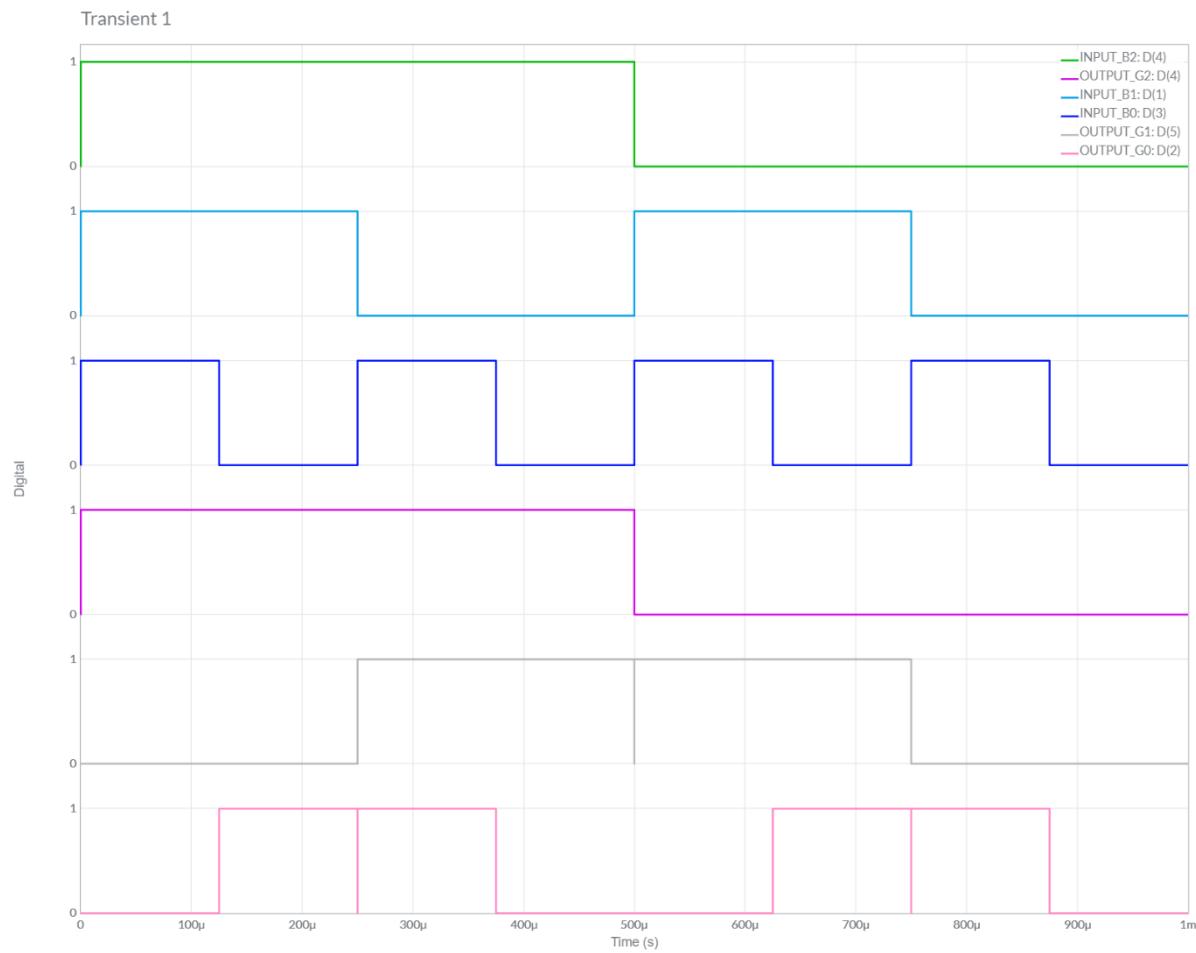


### 3 - BIT Binary to Gray code converter

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

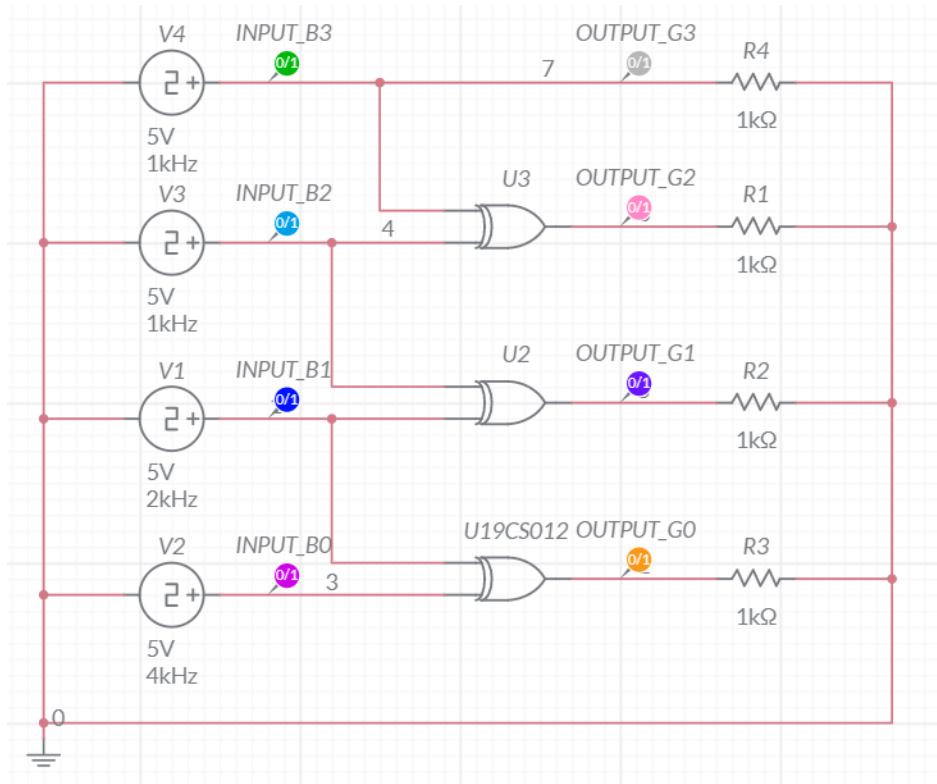


#### WAVEFORMS (FROM MULTISIM)

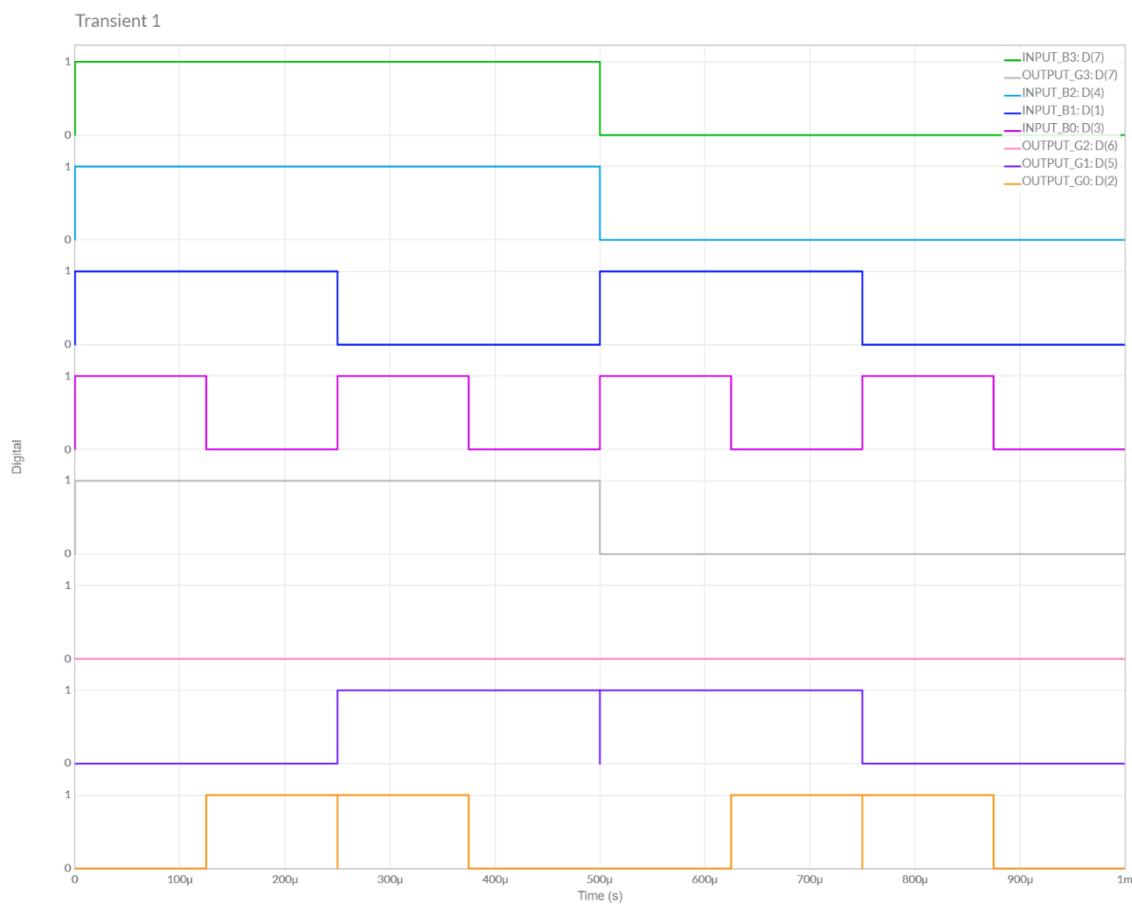




**4 - BIT Binary to Gray code converter**  
**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



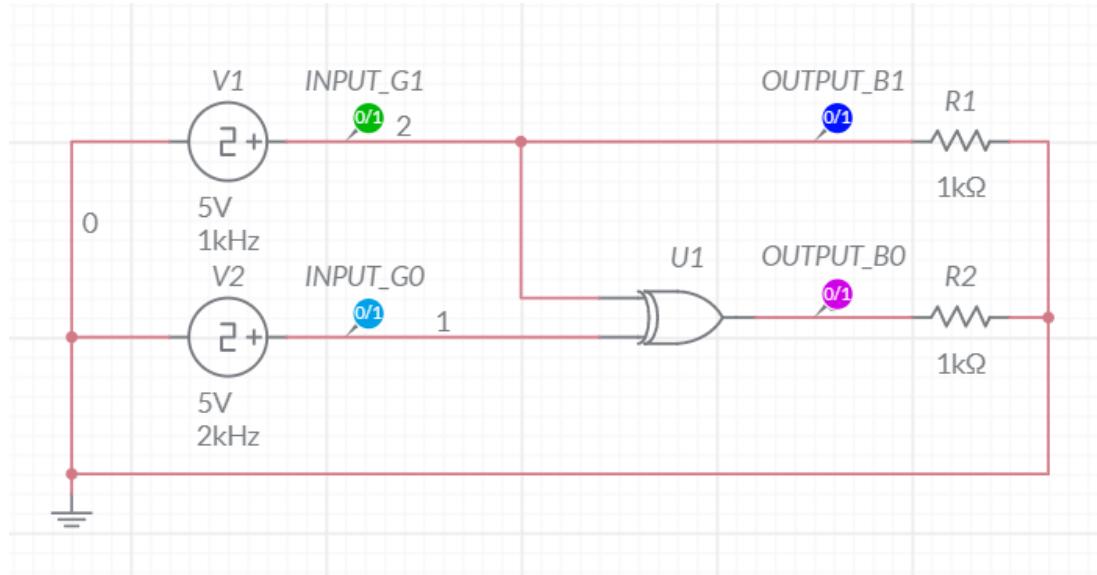
**WAVEFORMS (FROM MULTISIM)**



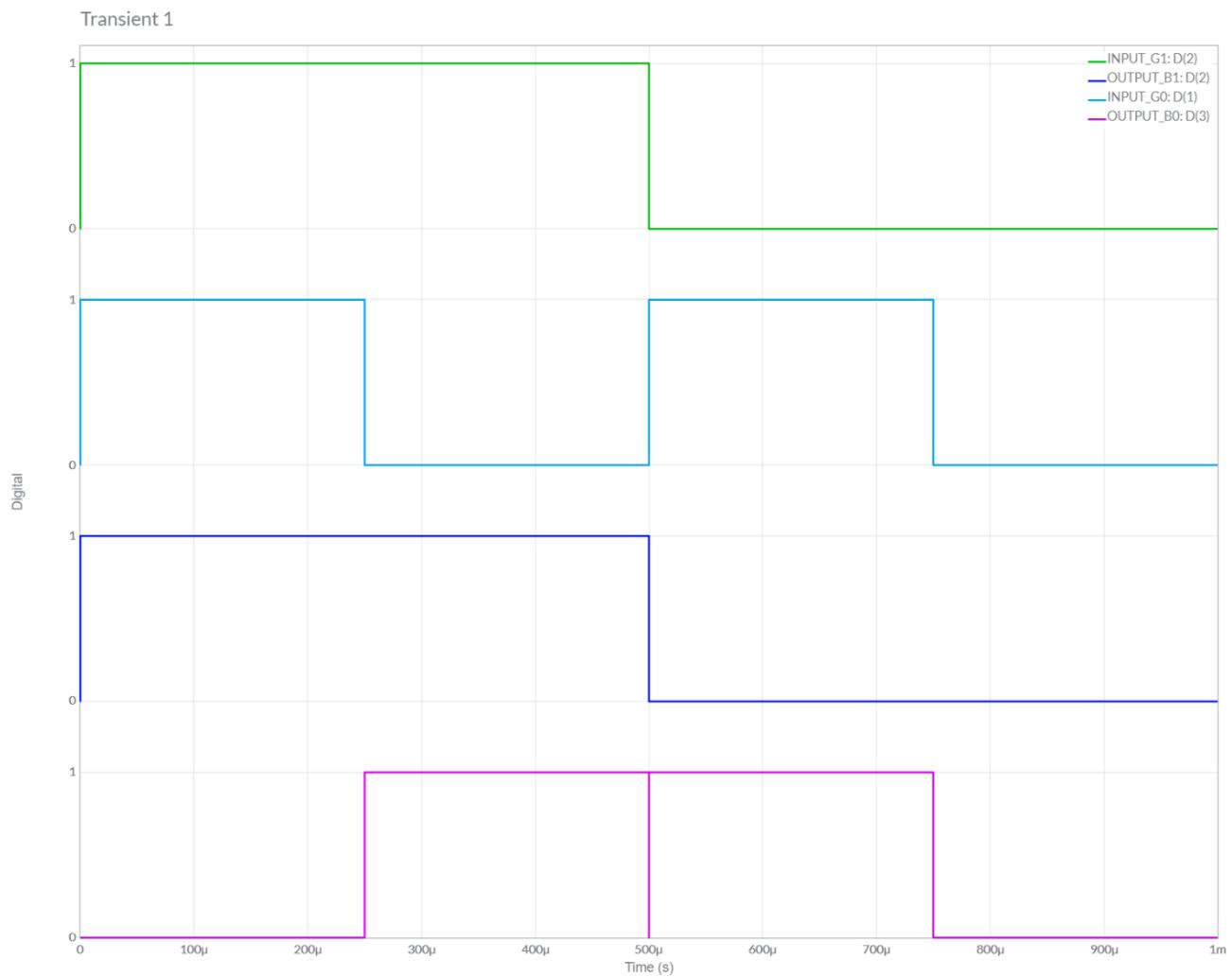


### 2 - BIT Gray to Binary code converter

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



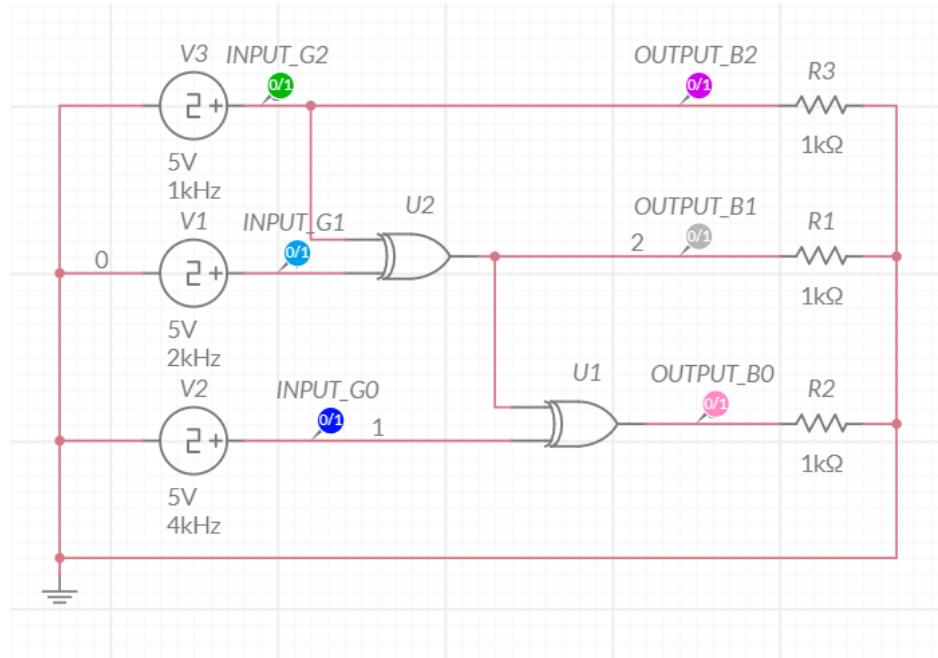
#### WAVEFORMS (FROM MULTISIM)



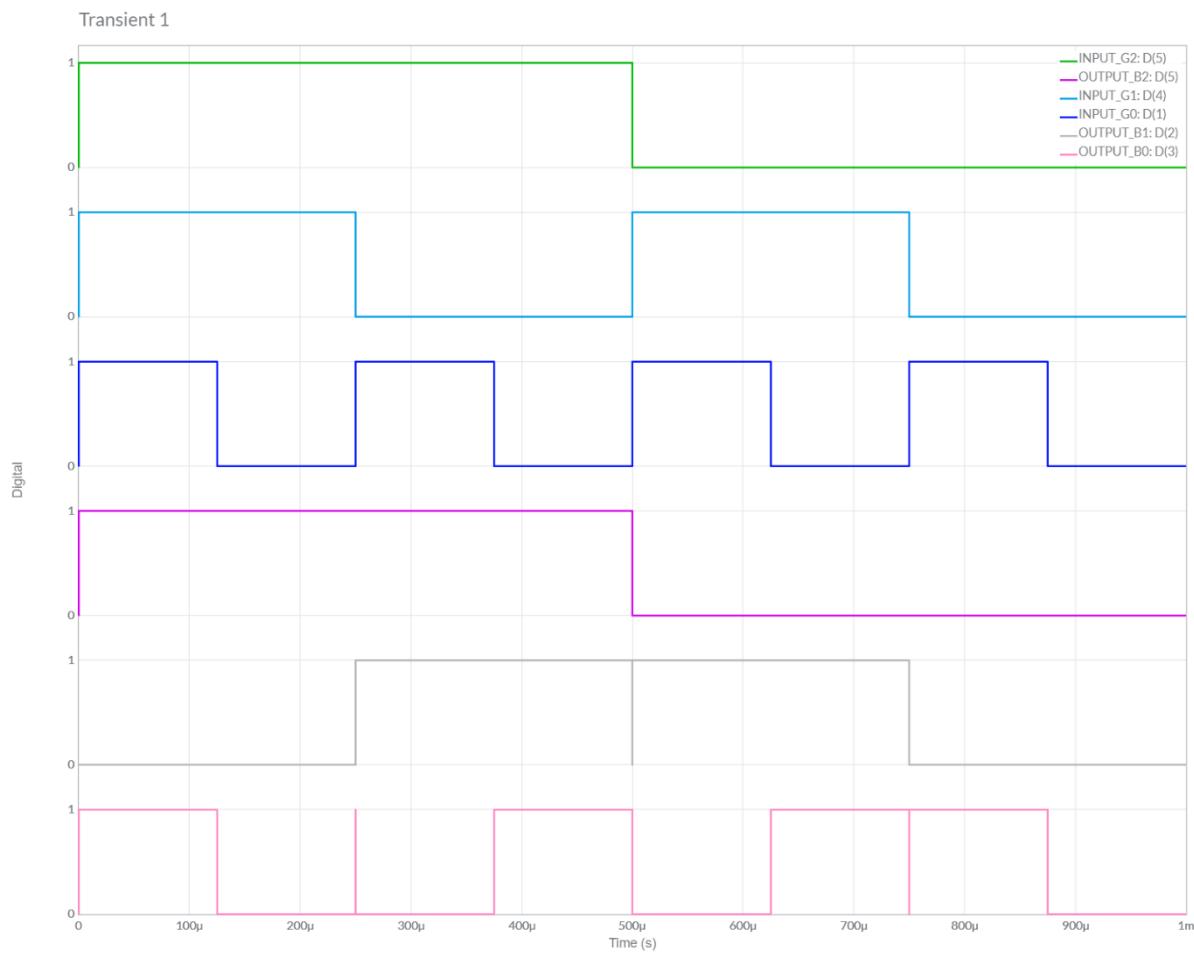


### 3 - BIT Gray to Binary code converter

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



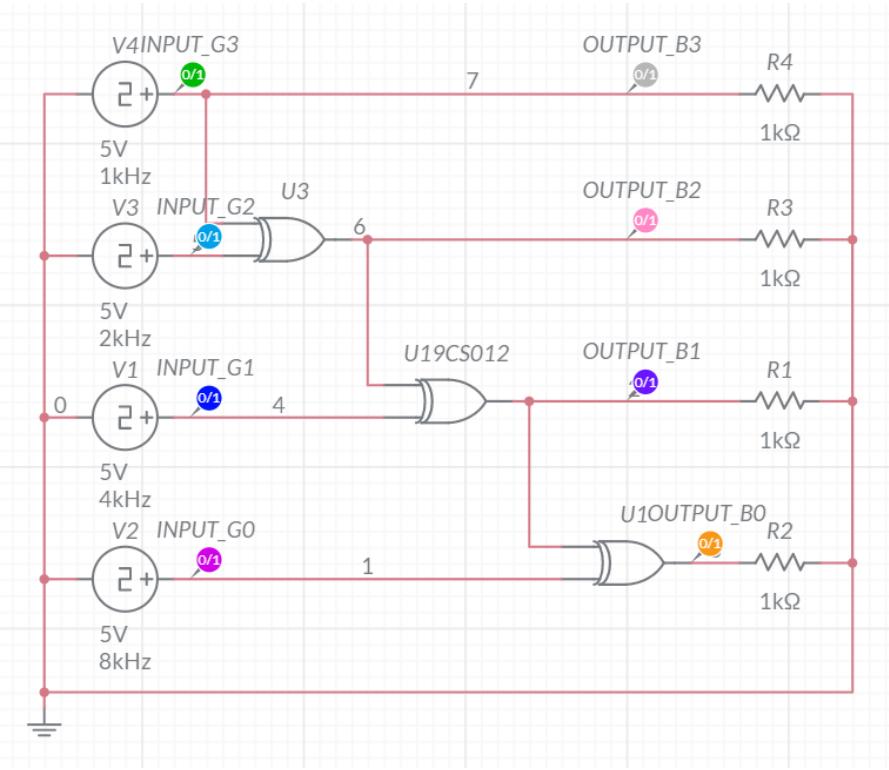
#### WAVEFORMS (FROM MULTISIM)



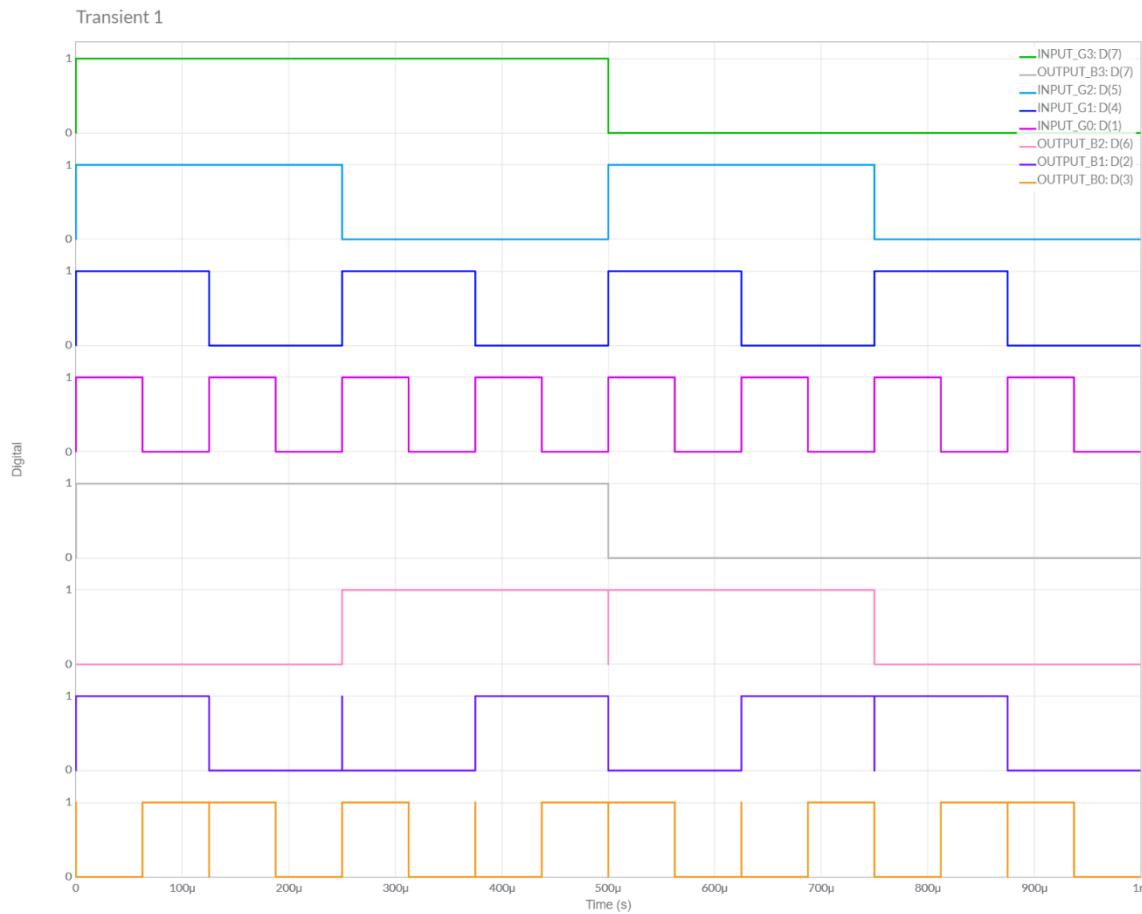


## 4 - BIT Gray to Binary code converter

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



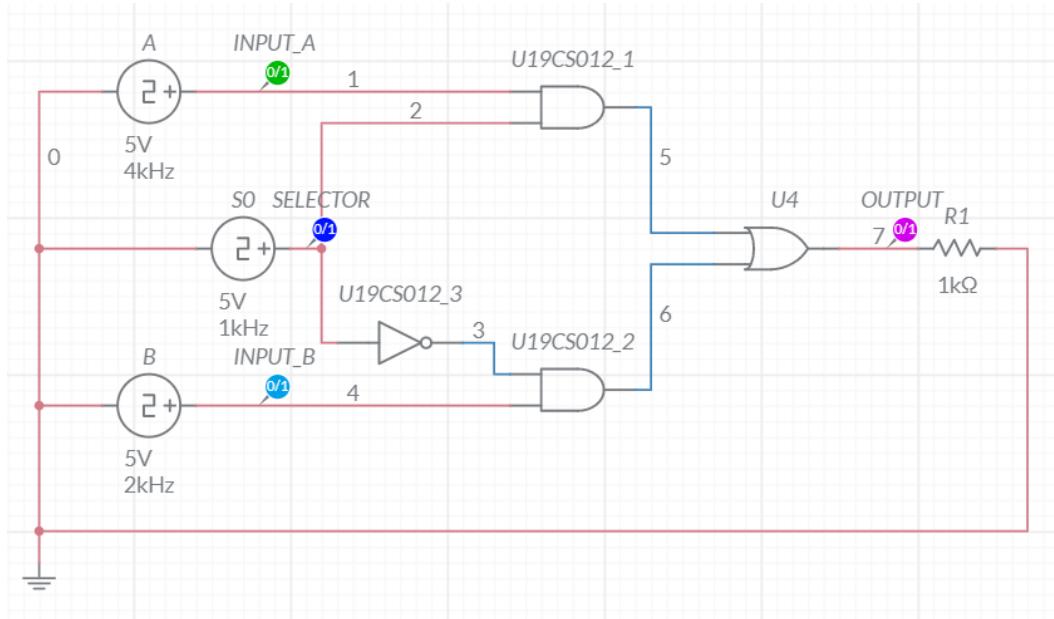
## WAVEFORMS (FROM MULTISIM)



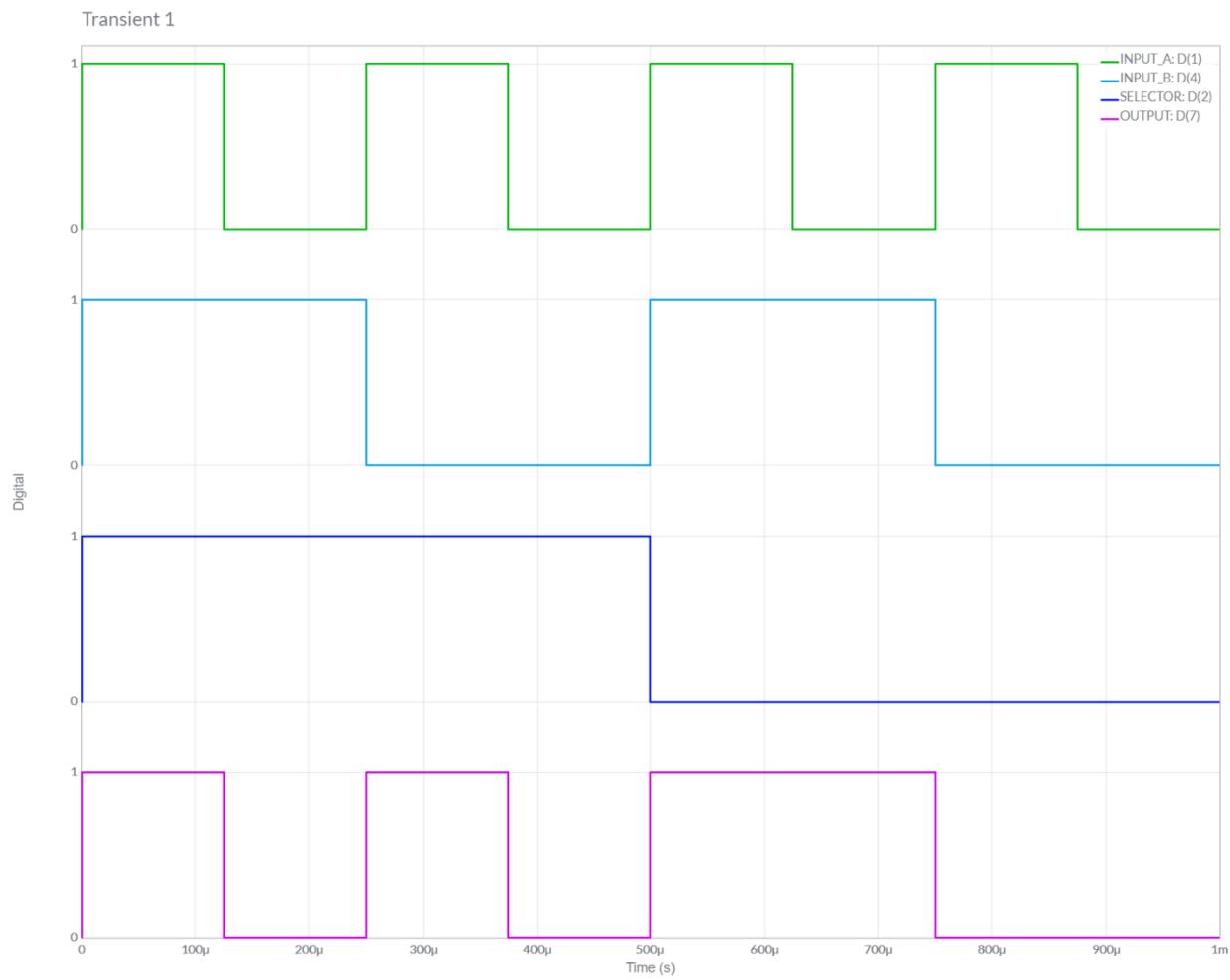


## 2 : 1 MULTIPLEXOR

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

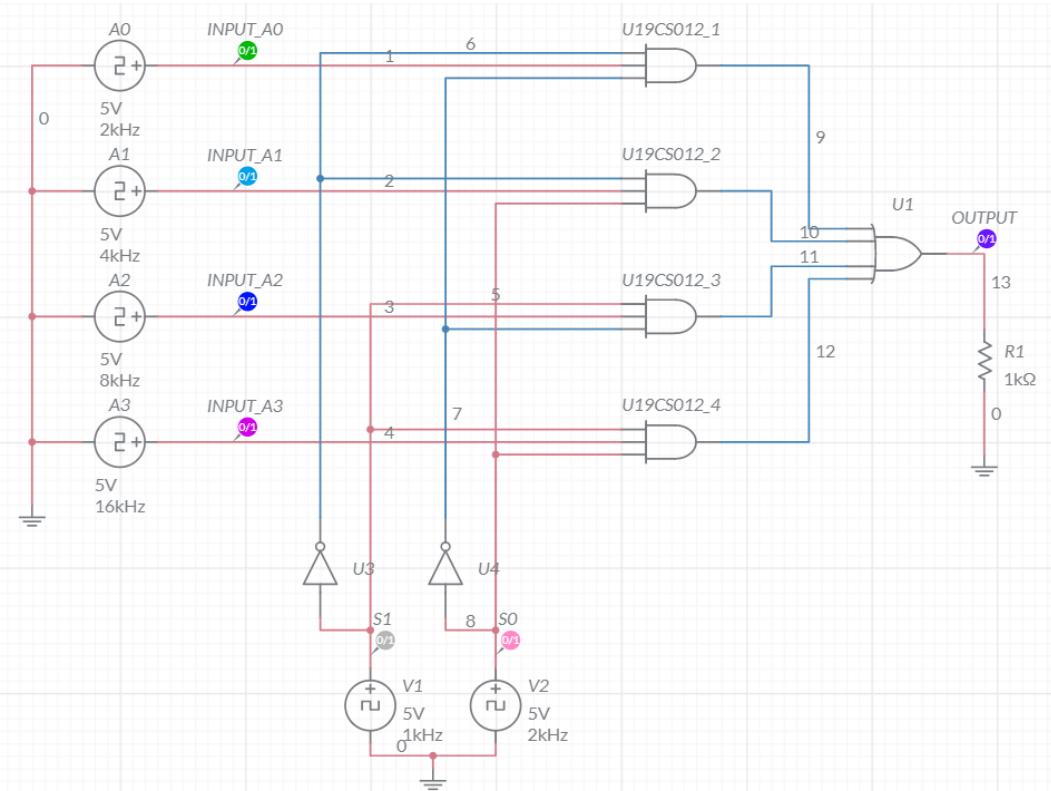


### WAVEFORMS (FROM MULTISIM)

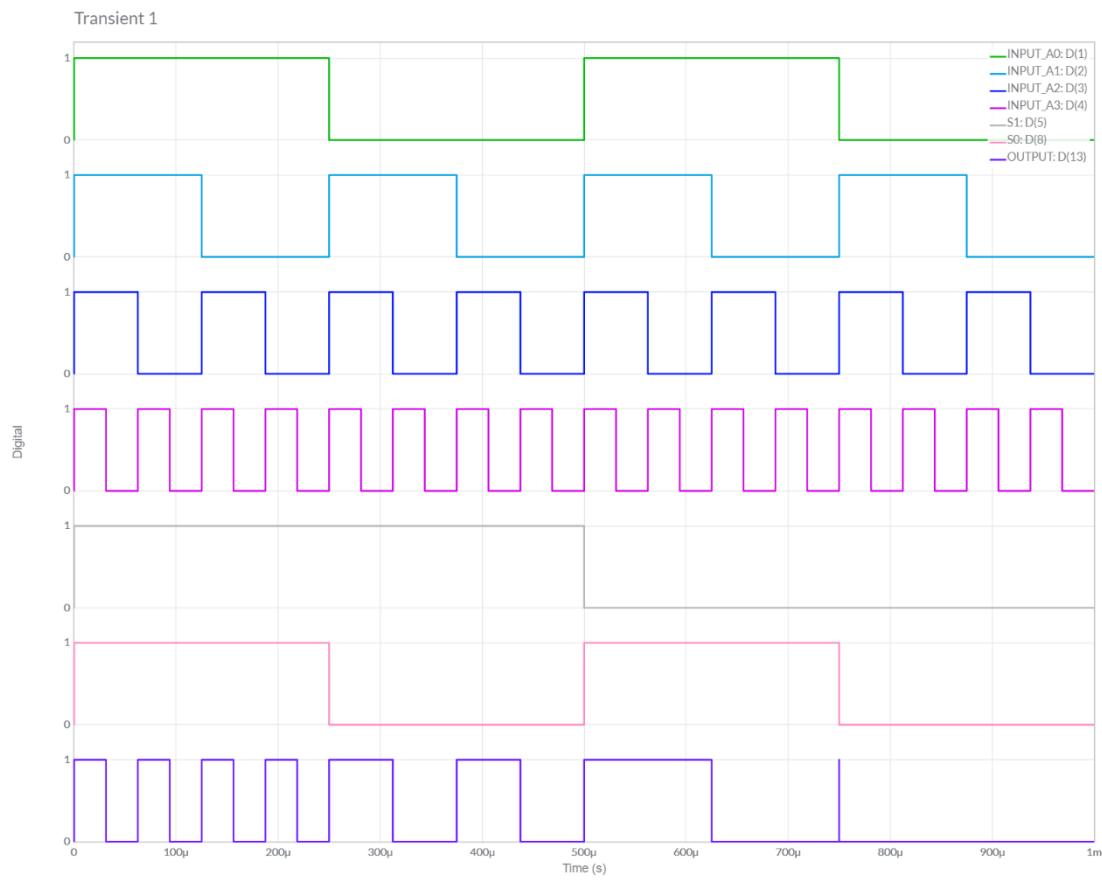




**4 : 1 MULTIPLEXOR**  
**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



**WAVEFORMS (FROM MULTISIM)**

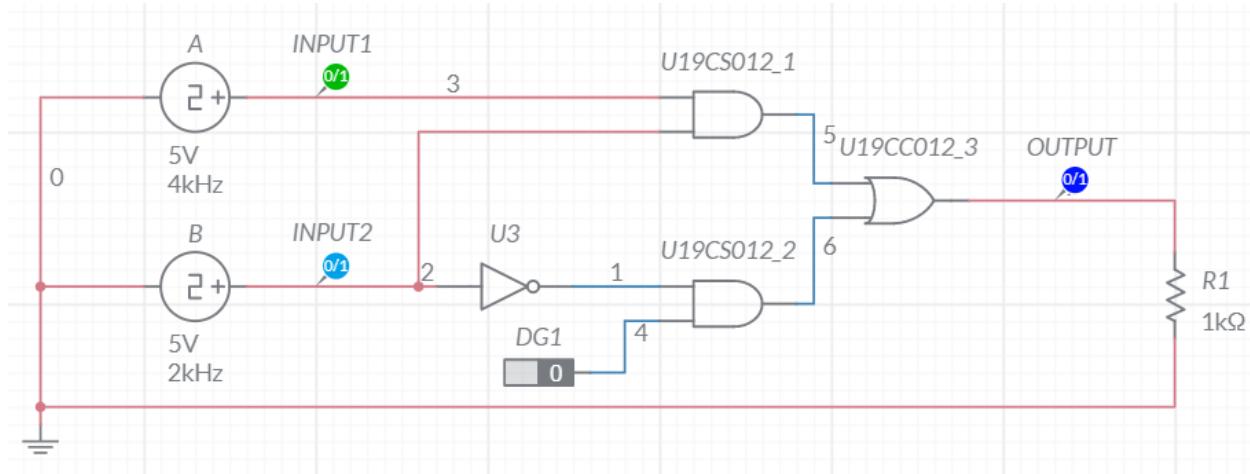




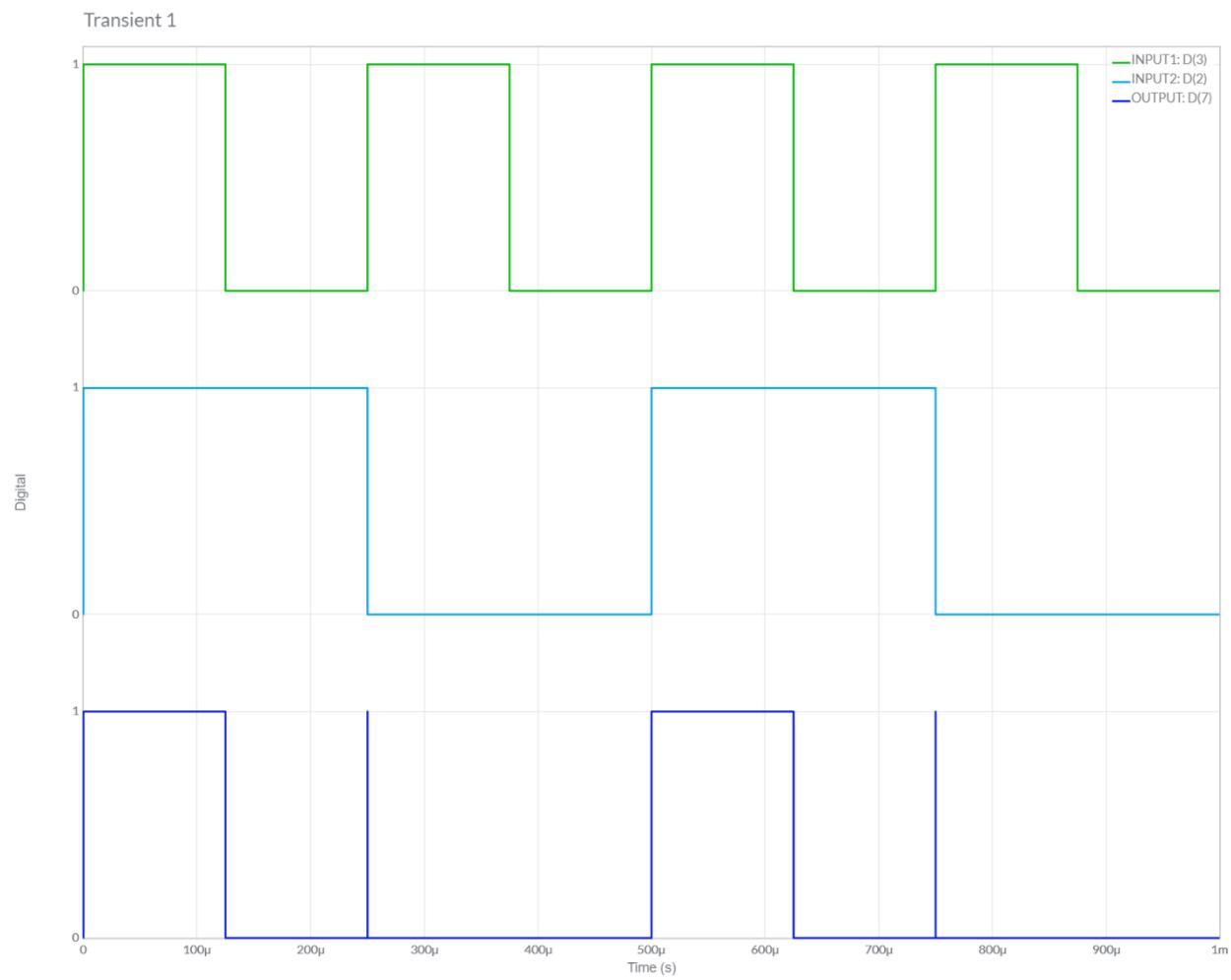
## BASIC GATES USING 2:1 MULTIPLEXER

### AND GATE

#### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



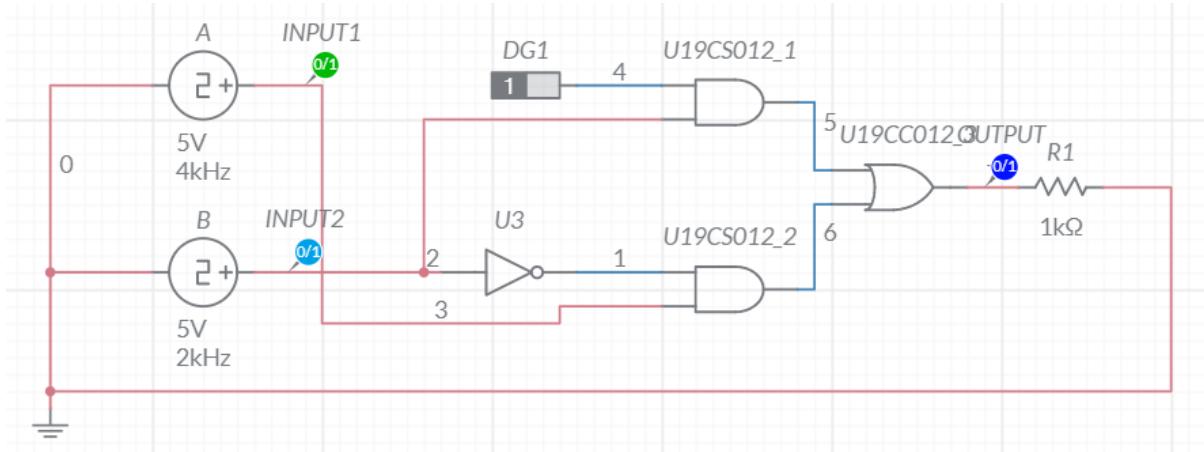
#### WAVEFORMS (FROM MULTISIM)



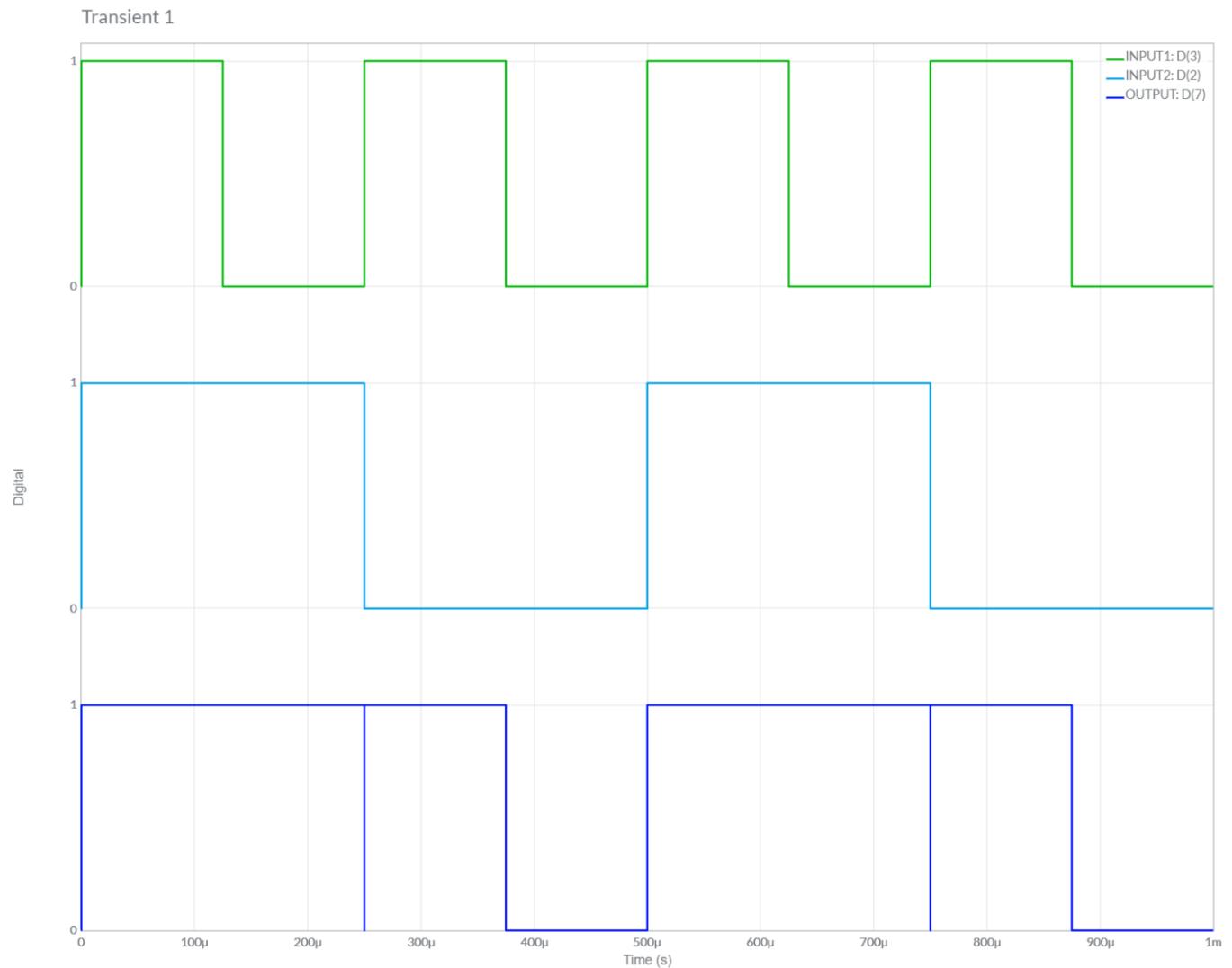


## OR GATE

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



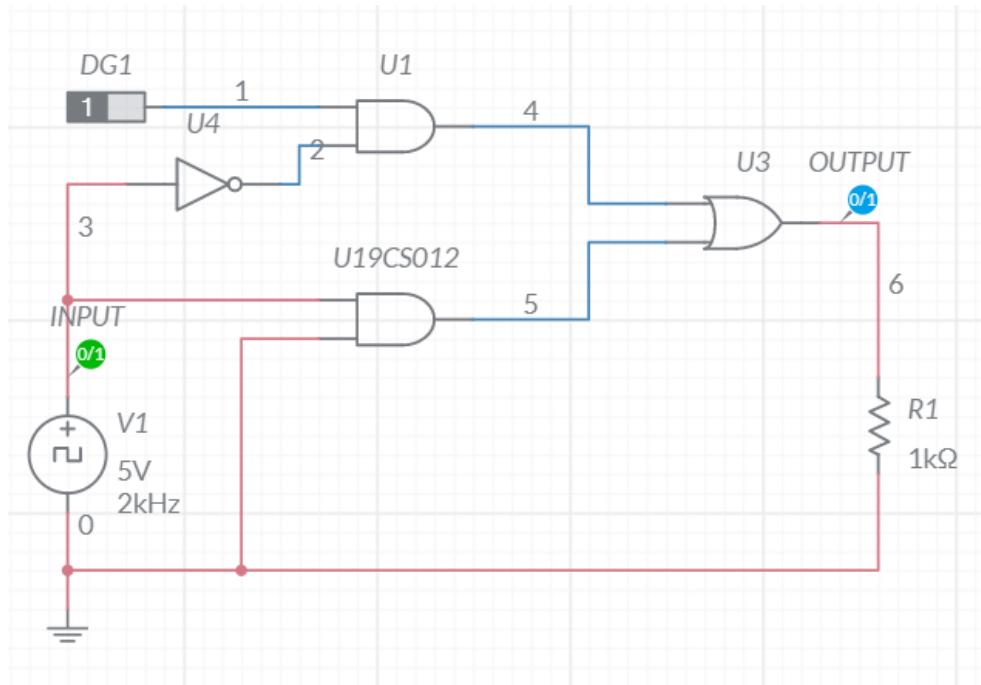
### WAVEFORMS (FROM MULTISIM)



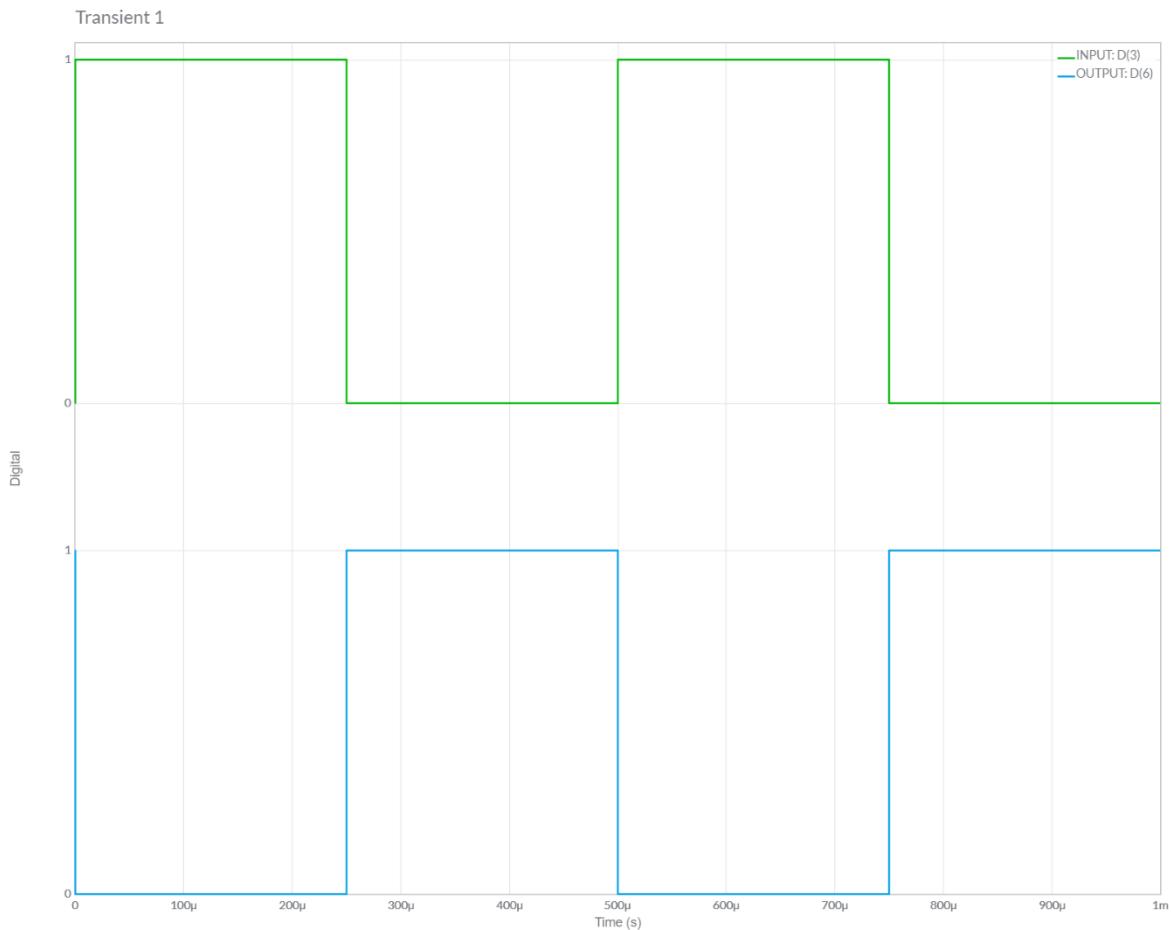


## NOT GATE

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



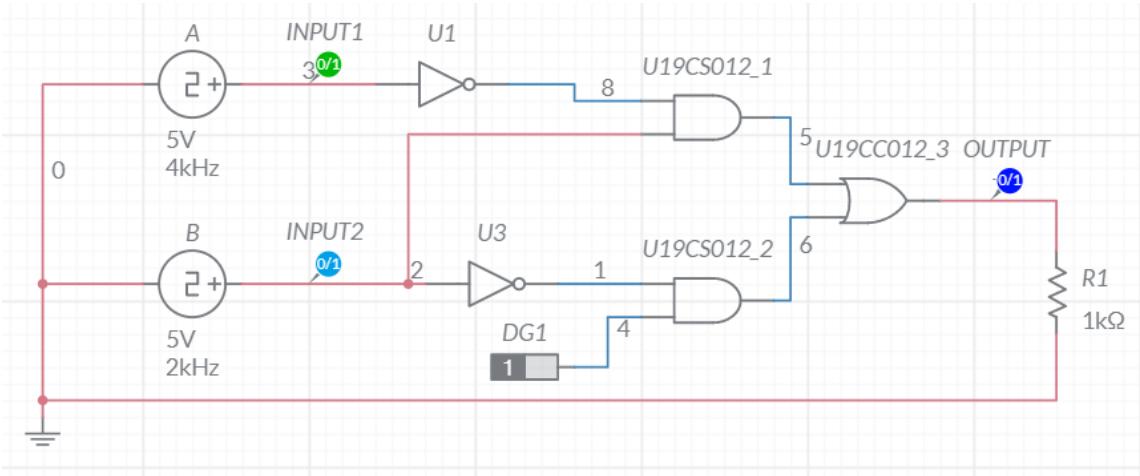
### WAVEFORMS (FROM MULTISIM)





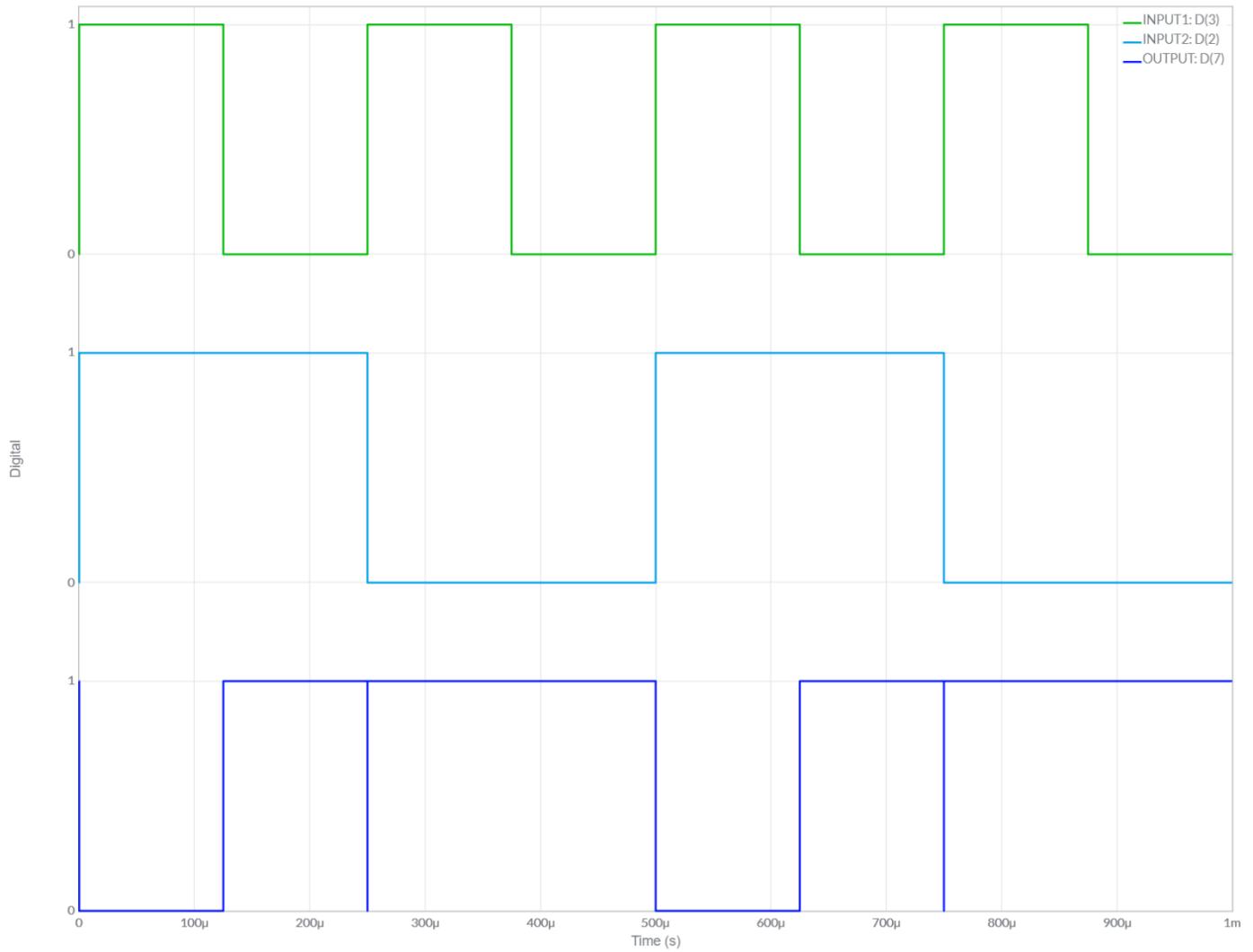
## NAND GATE

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



### WAVEFORMS (FROM MULTISIM)

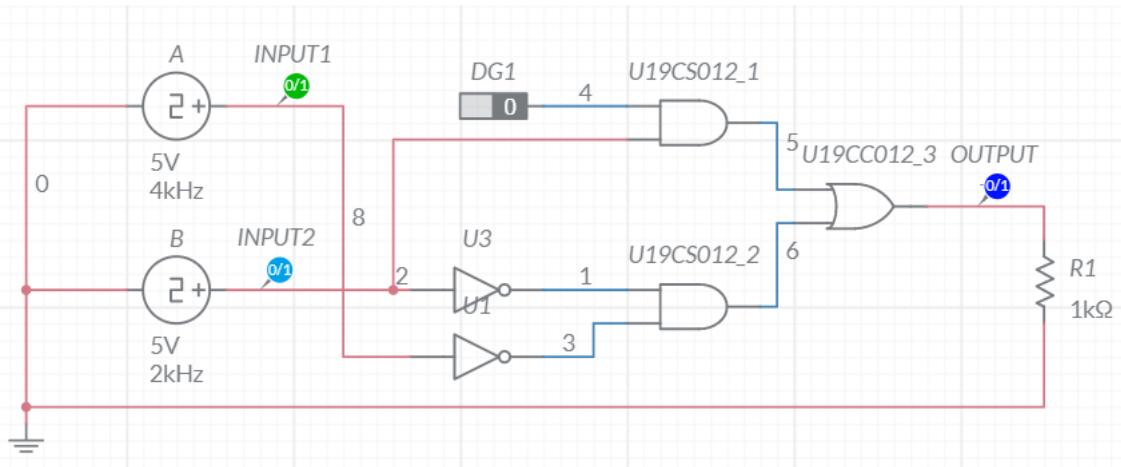
Transient 1





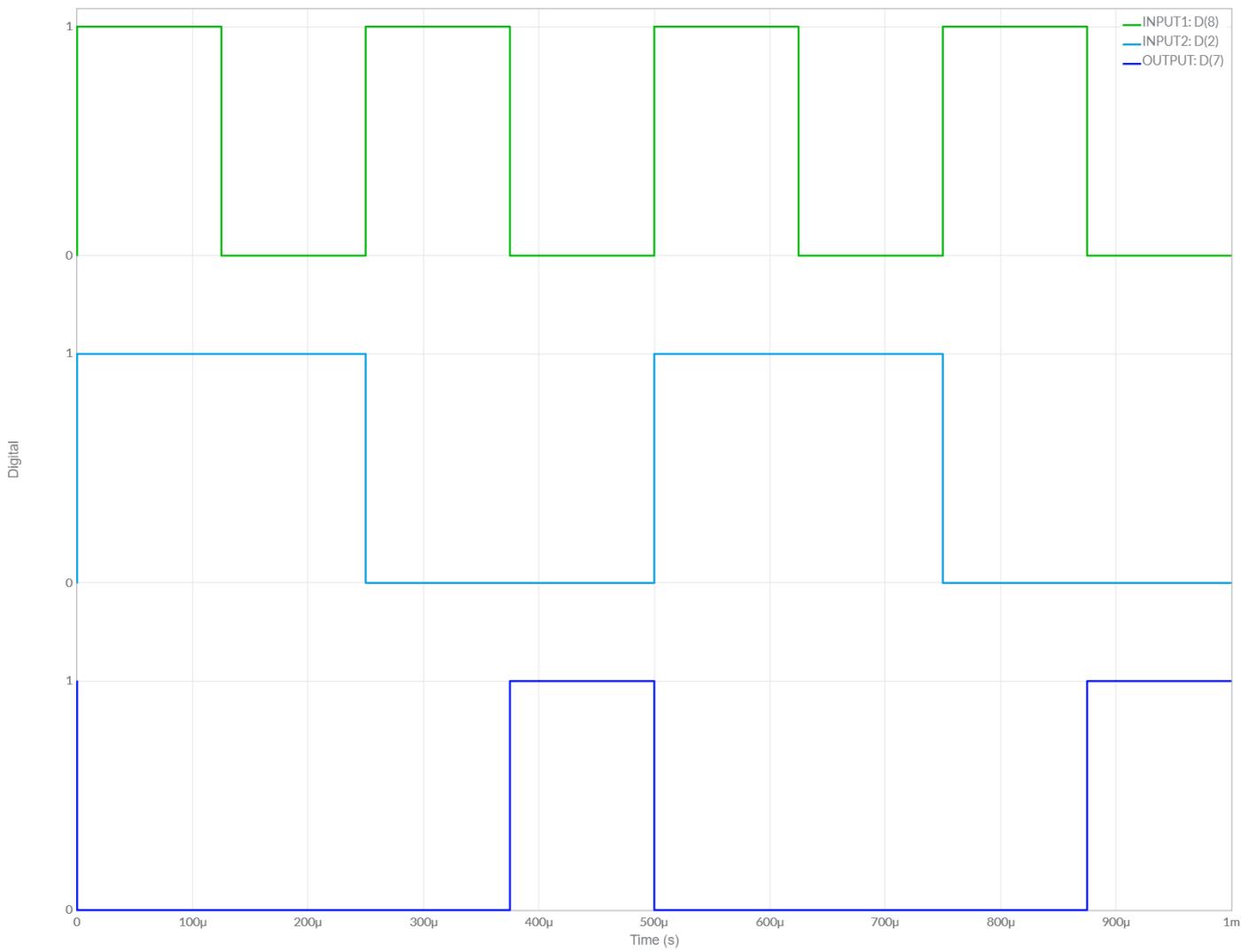
## NOR GATE

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



### WAVEFORMS (FROM MULTISIM)

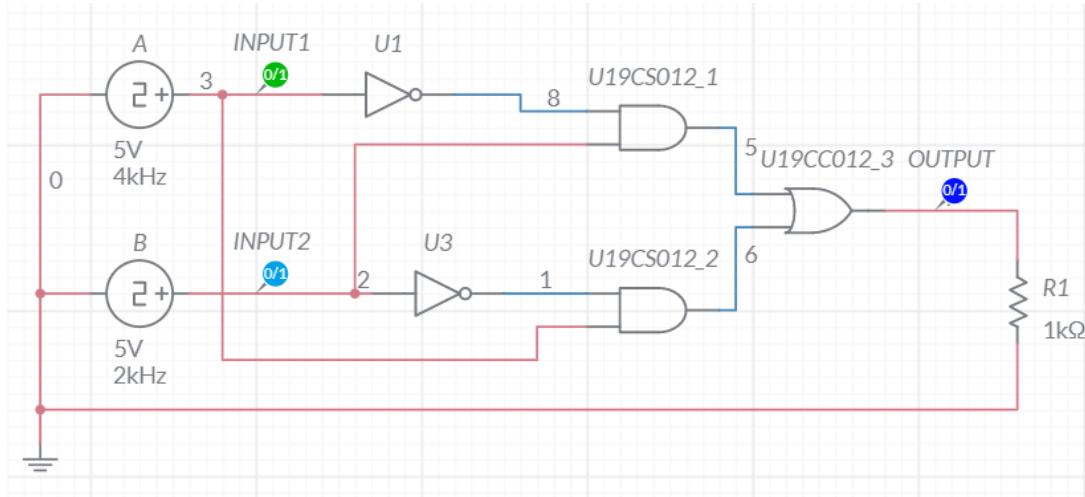
Transient 1



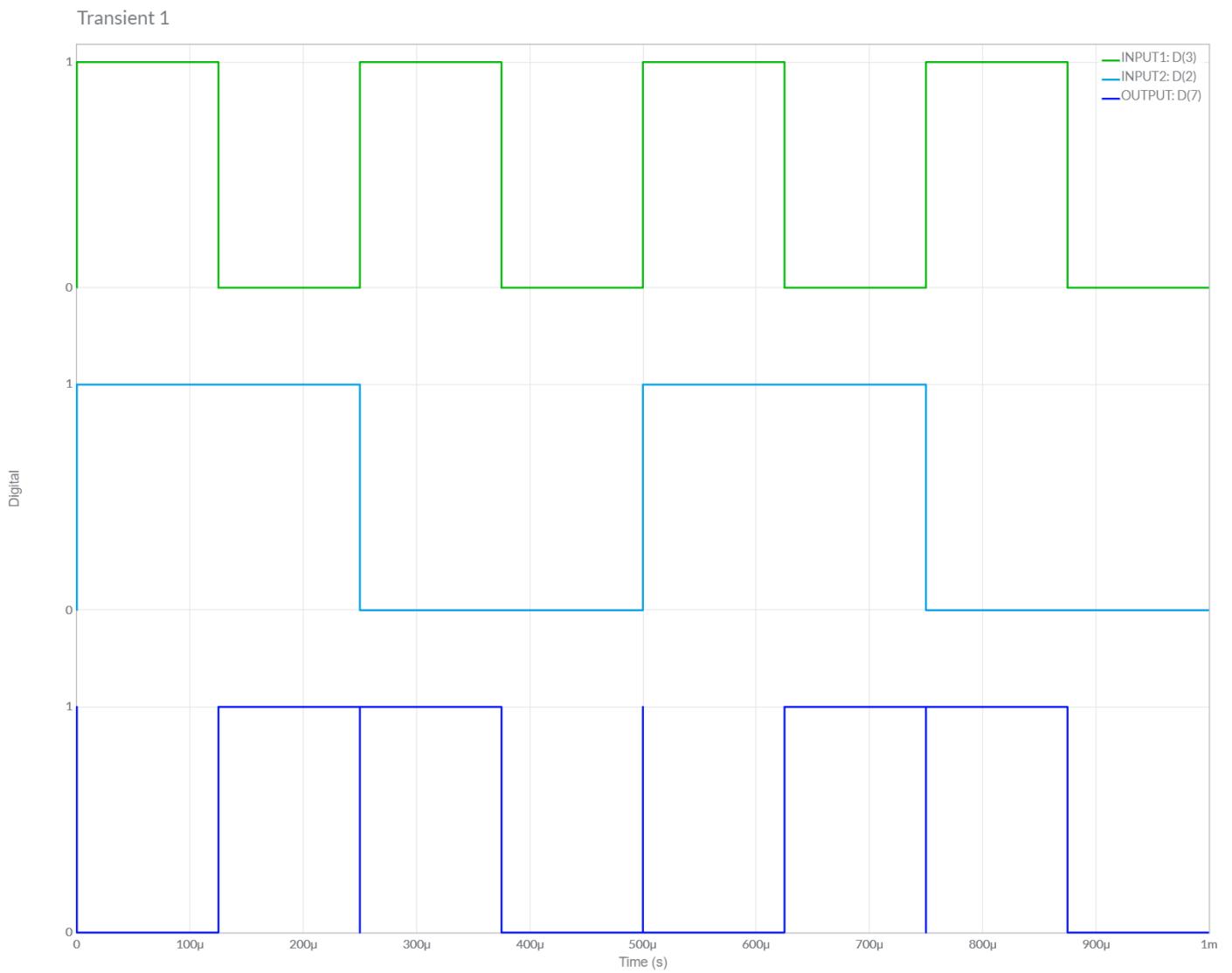


## XOR GATE

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



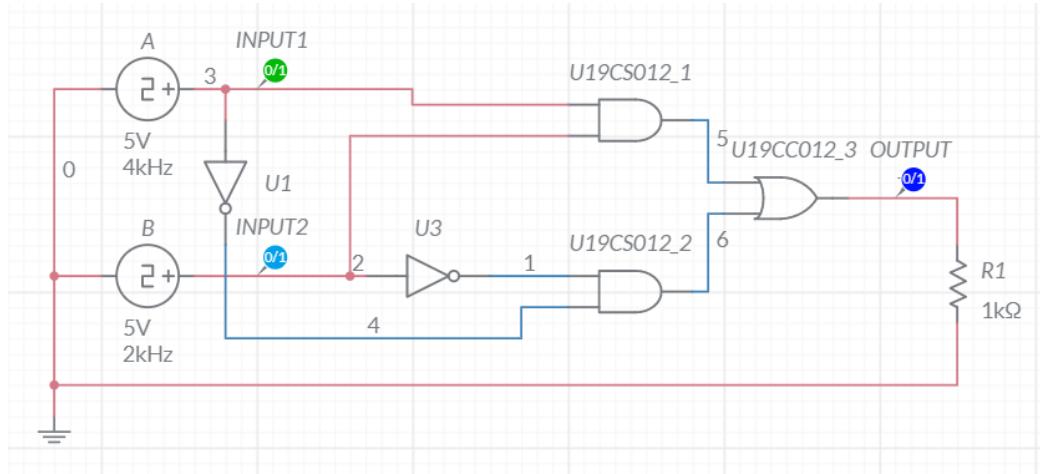
### WAVEFORMS (FROM MULTISIM)



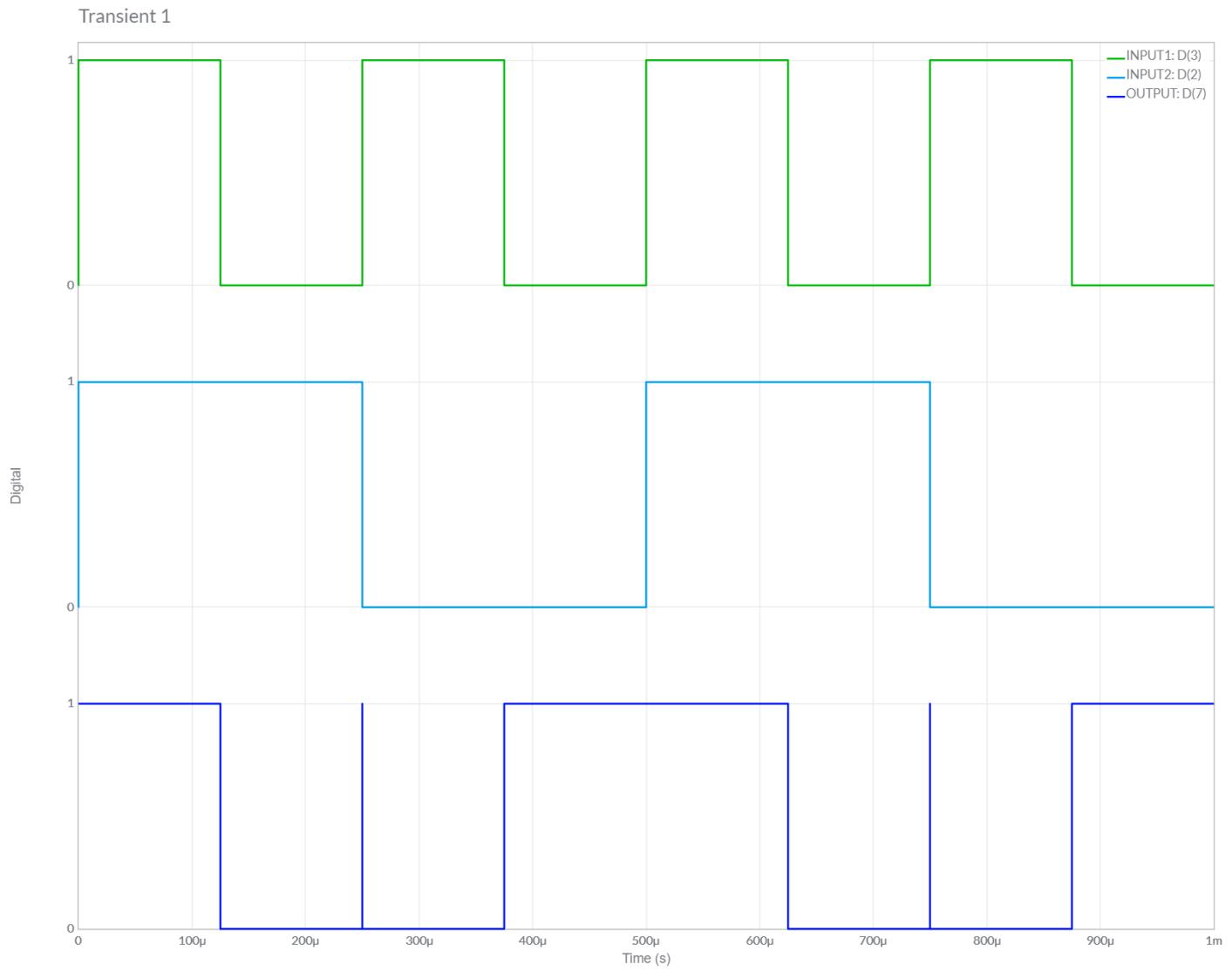


## XNOR GATE

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



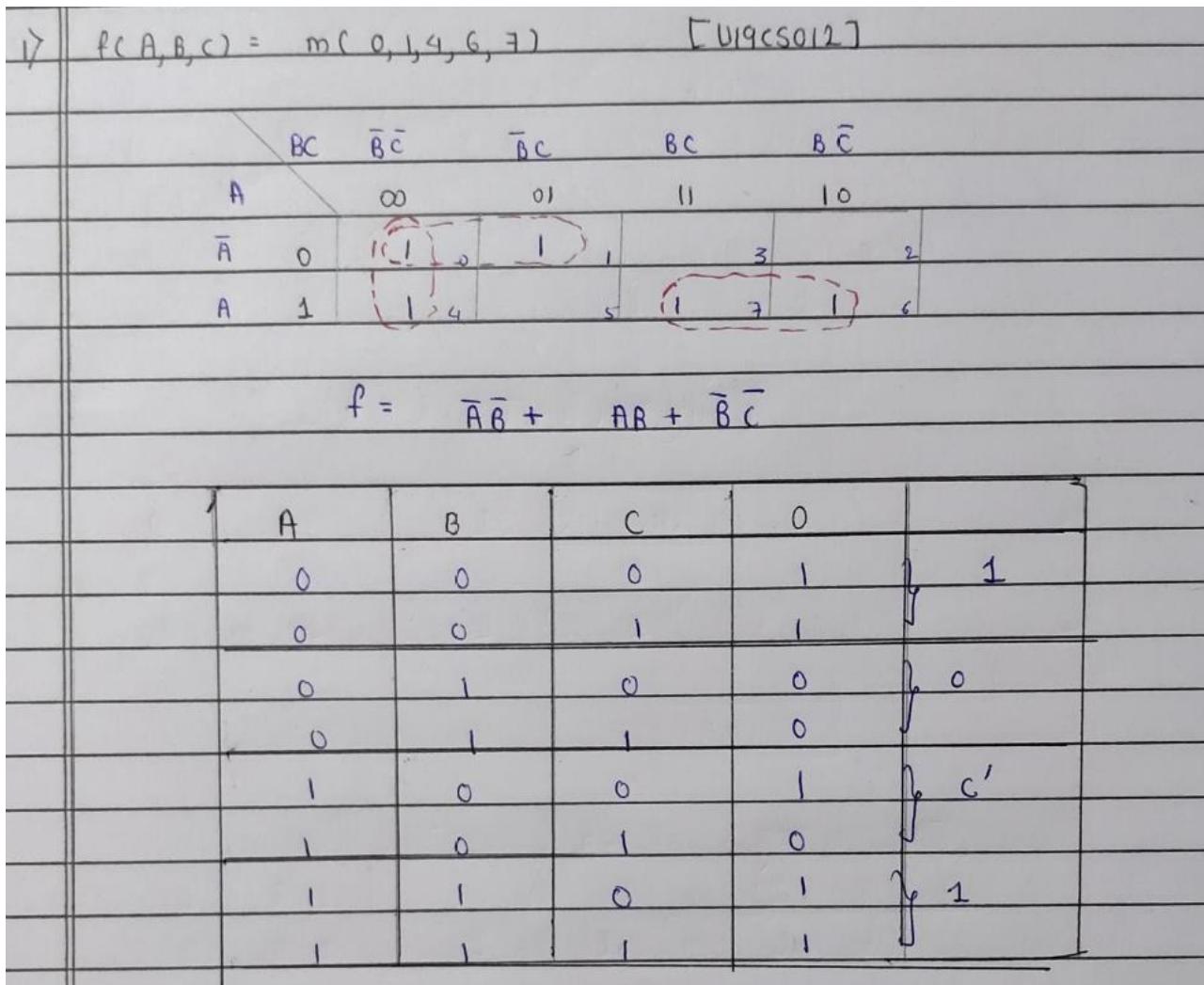
### WAVEFORMS (FROM MULTISIM)



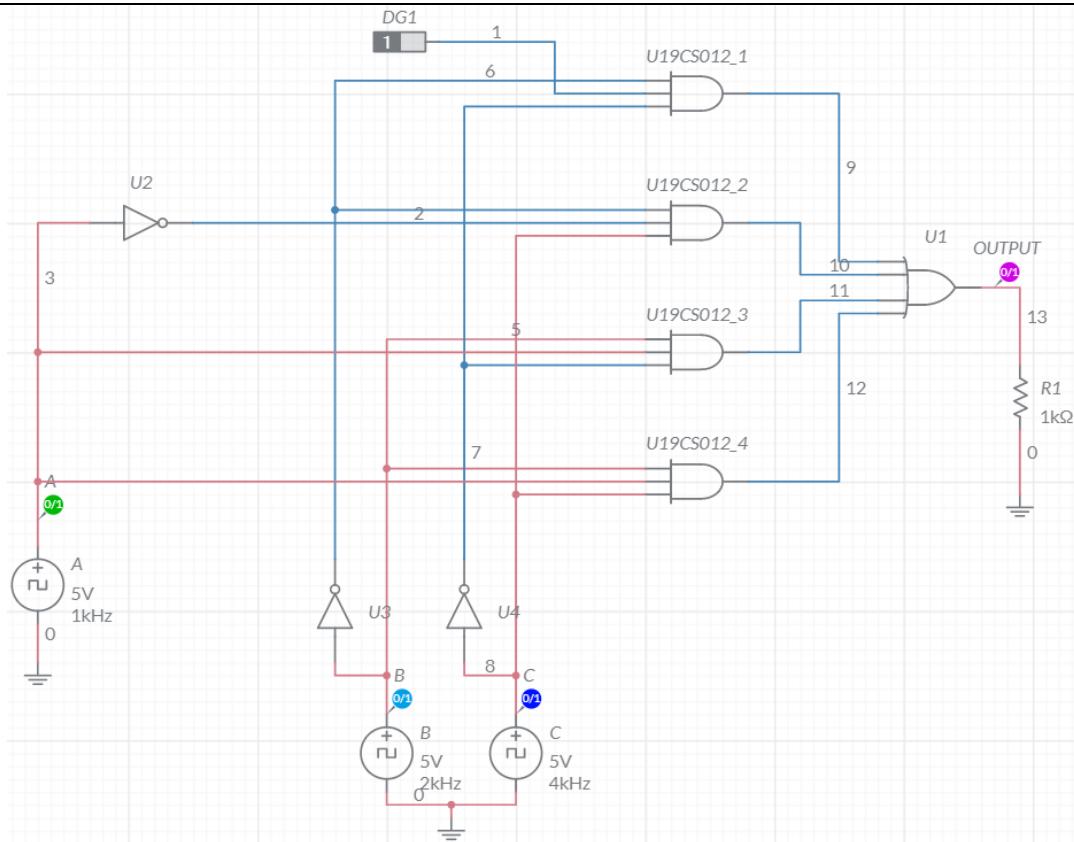


### Function Implementation using Multiplexers

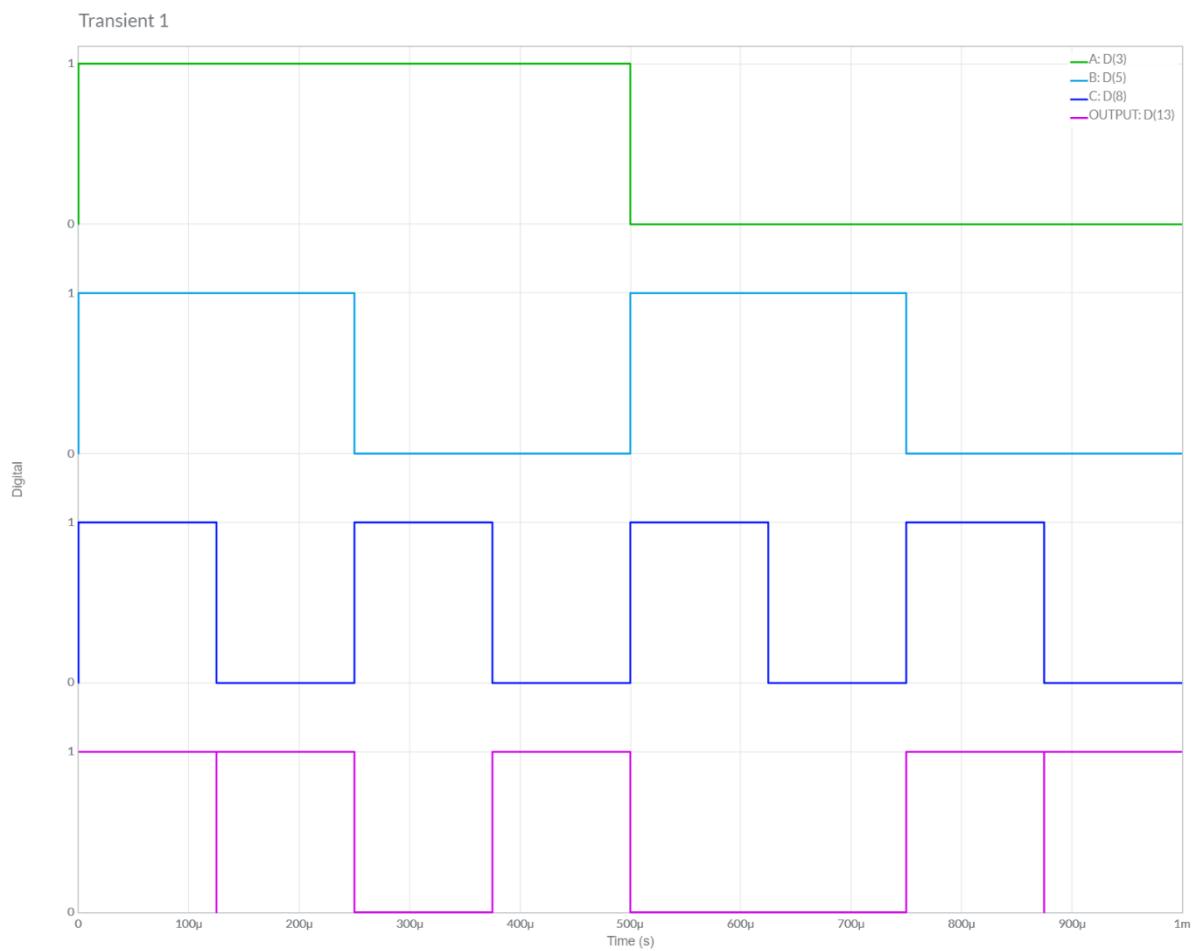
$$1) f(A, B, C) = m(0, 1, 4, 6, 7)$$



**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



**WAVEFORMS (FROM MULTISIM)**





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

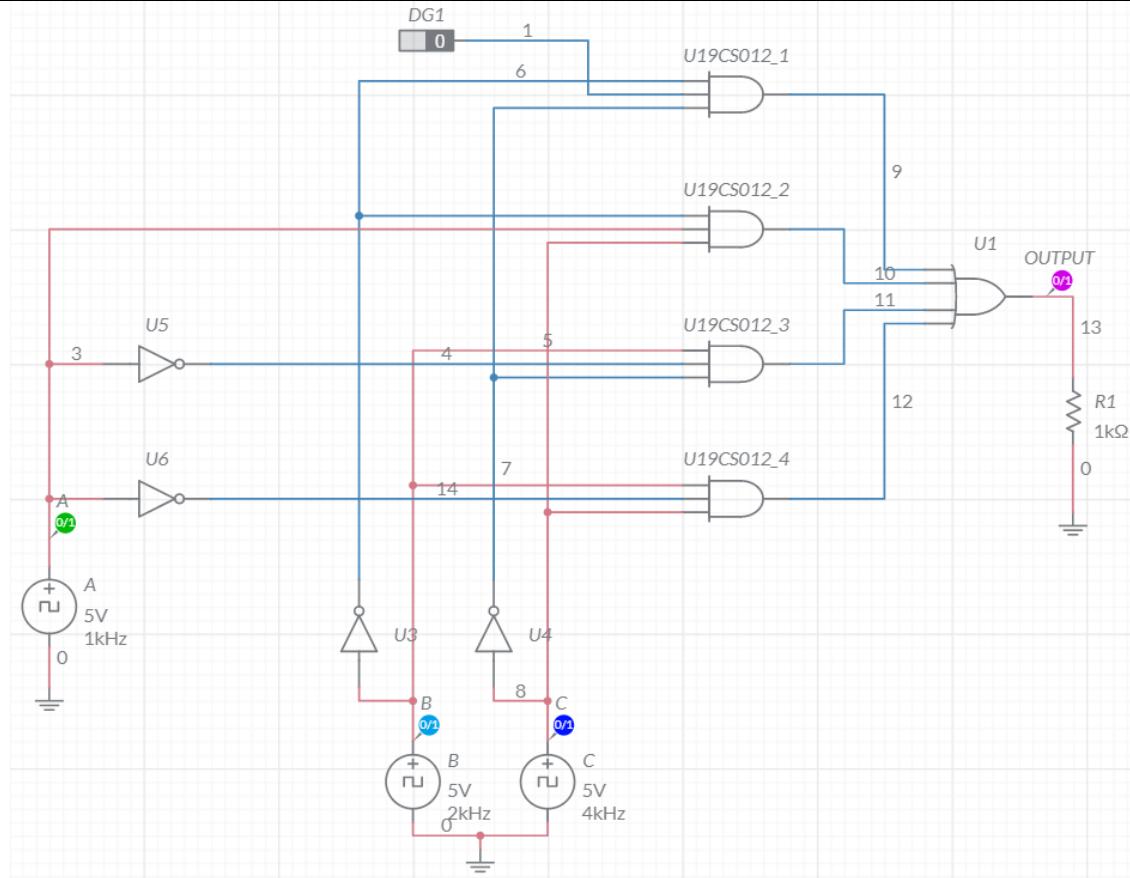
2)  $f(A, B, C) = M(0, 1, 4, 6, 7)$  [U19CS012]

		BC	$(B+C)$	$(B+\bar{C})$	$(\bar{B}+\bar{C})$	$(\bar{B}+C)$
		A	00	01	11	10
A	B	0	(0)	(0)	(0)	(0)
		1	(0)	(1)	(1)	(0)

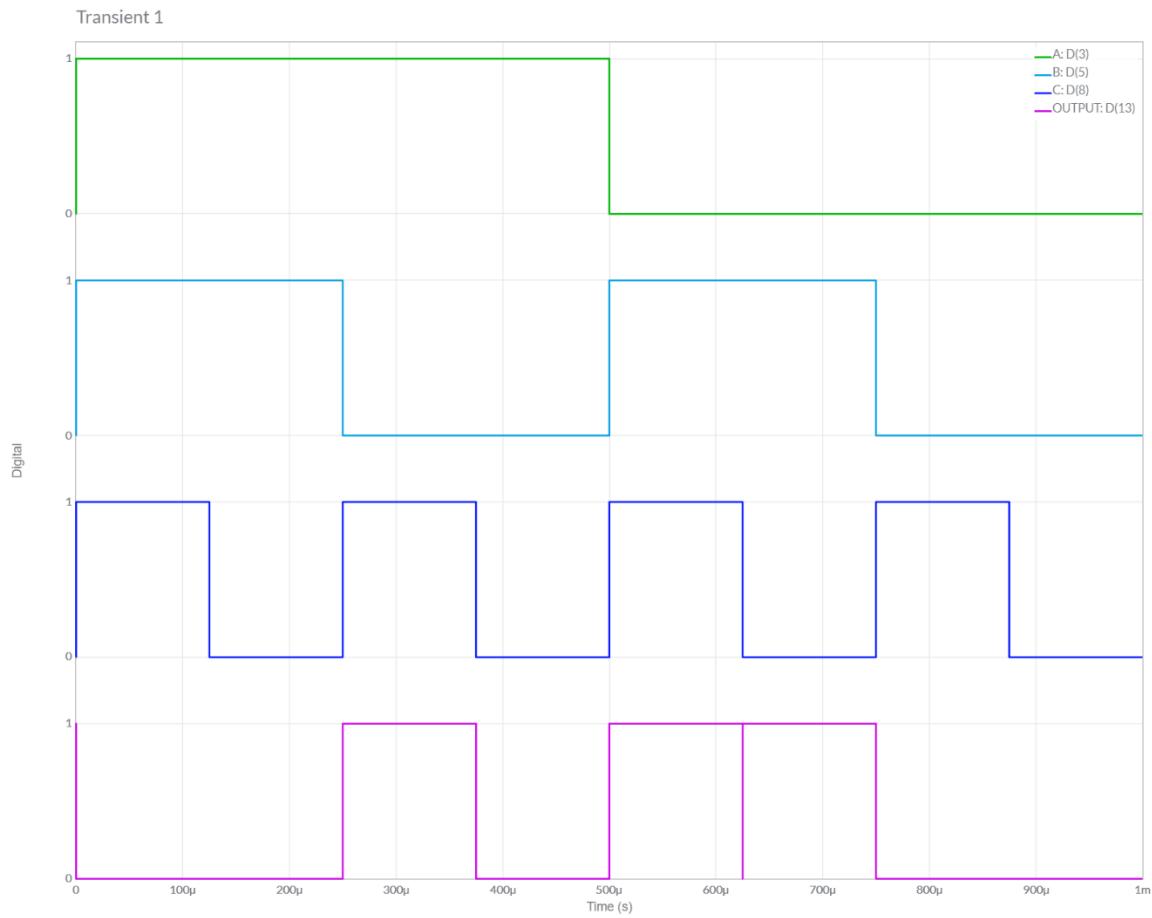
$$f = (A+B) \cdot (\bar{A}+\bar{B}) \cdot (B+C)$$

A	B	C	O	
0	0	0	0	0
0	0	1	0	
0	1	0	1	1
0	1	1	1	
1	0	0	0	C
1	0	1	1	
1	1	0	0	0
1	1	1	0	

**CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)**



**WAVEFORMS (FROM MULTISIM)**



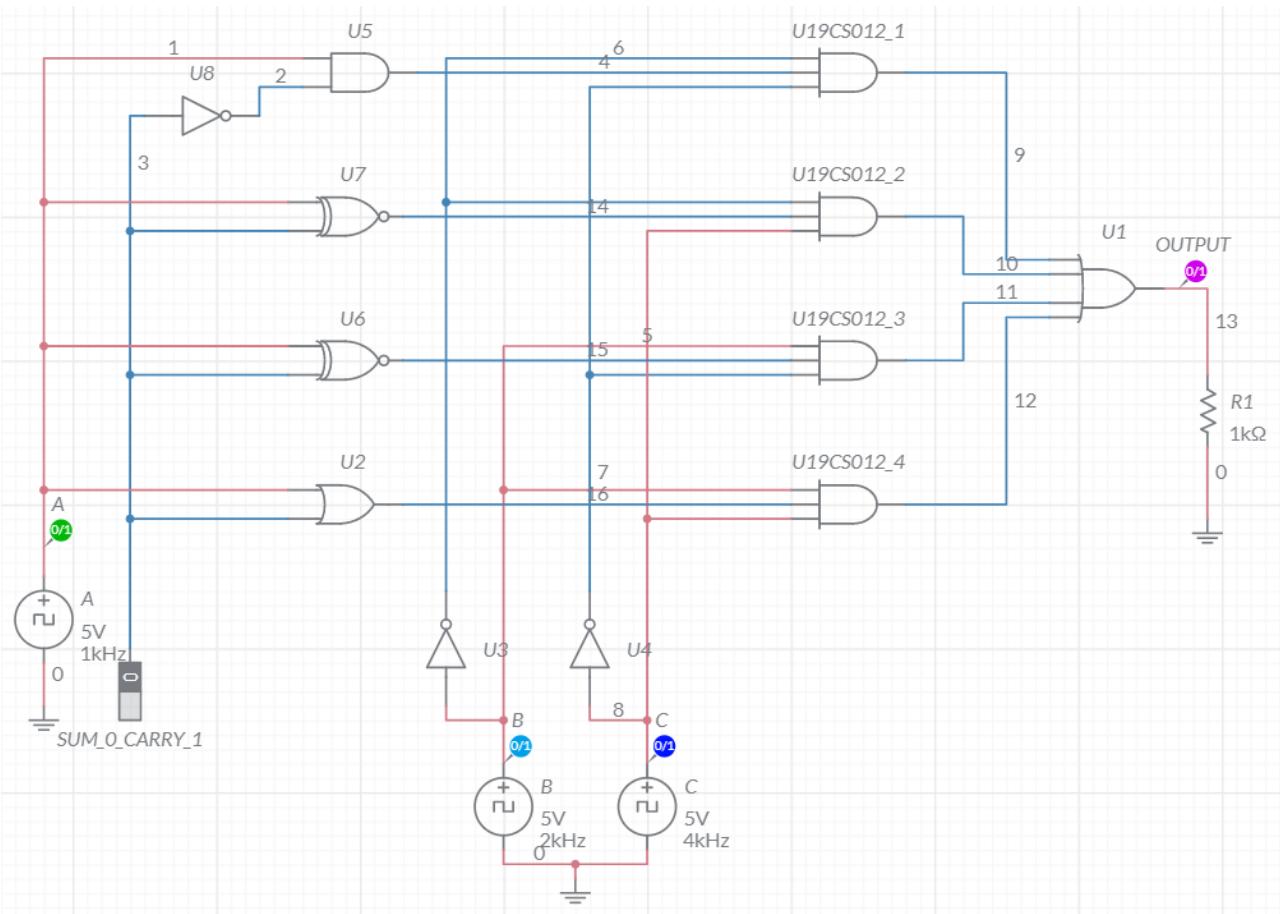


### 3) Full Adder

3.7 Full Adder [U19CS012]

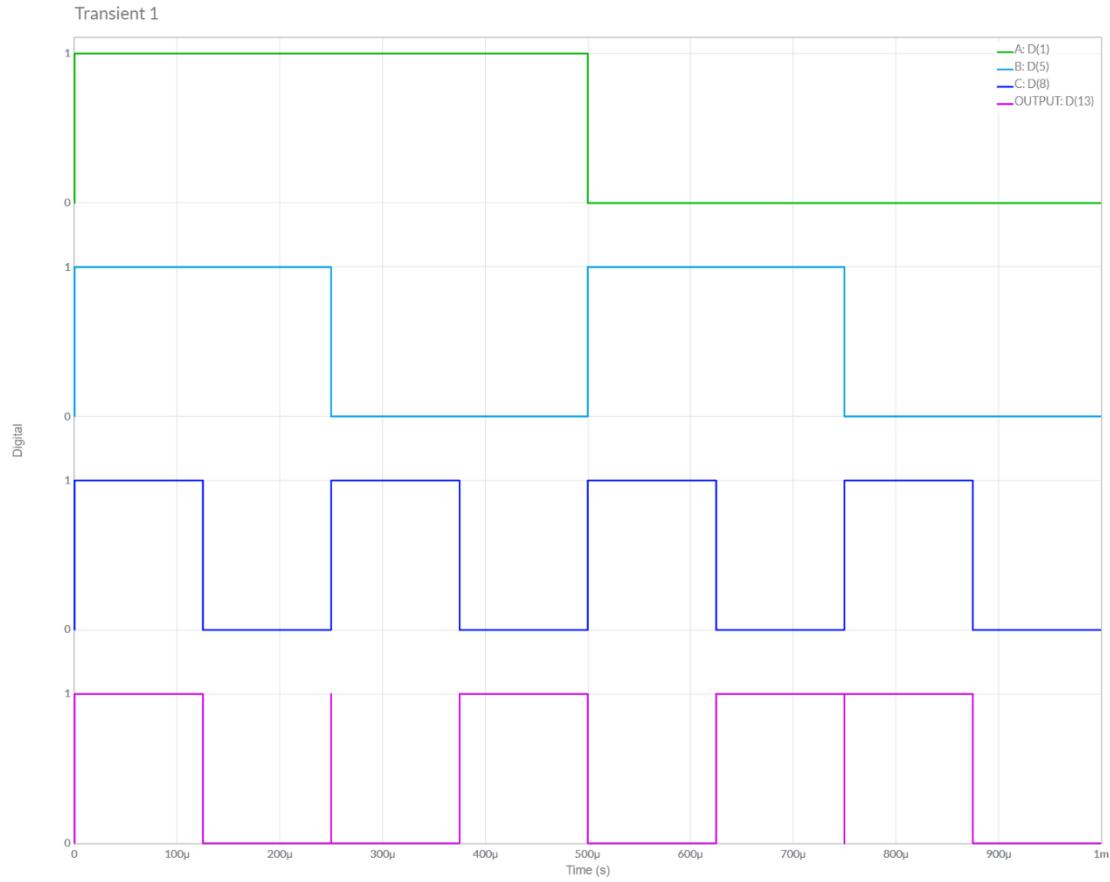
A	B	C <sub>in</sub>	Sum	carry
0	0	0	0 } c	0 } 0
0	0	1	1	0
0	1	0	1 } c'	0 } c
0	1	1	0 } c	1 } 1
1	0	0	1 } c'	0 } c
1	0	1	0	1 } 1
1	1	0	0 } c	1 } 1
1	1	1	1	1 } ①

### CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

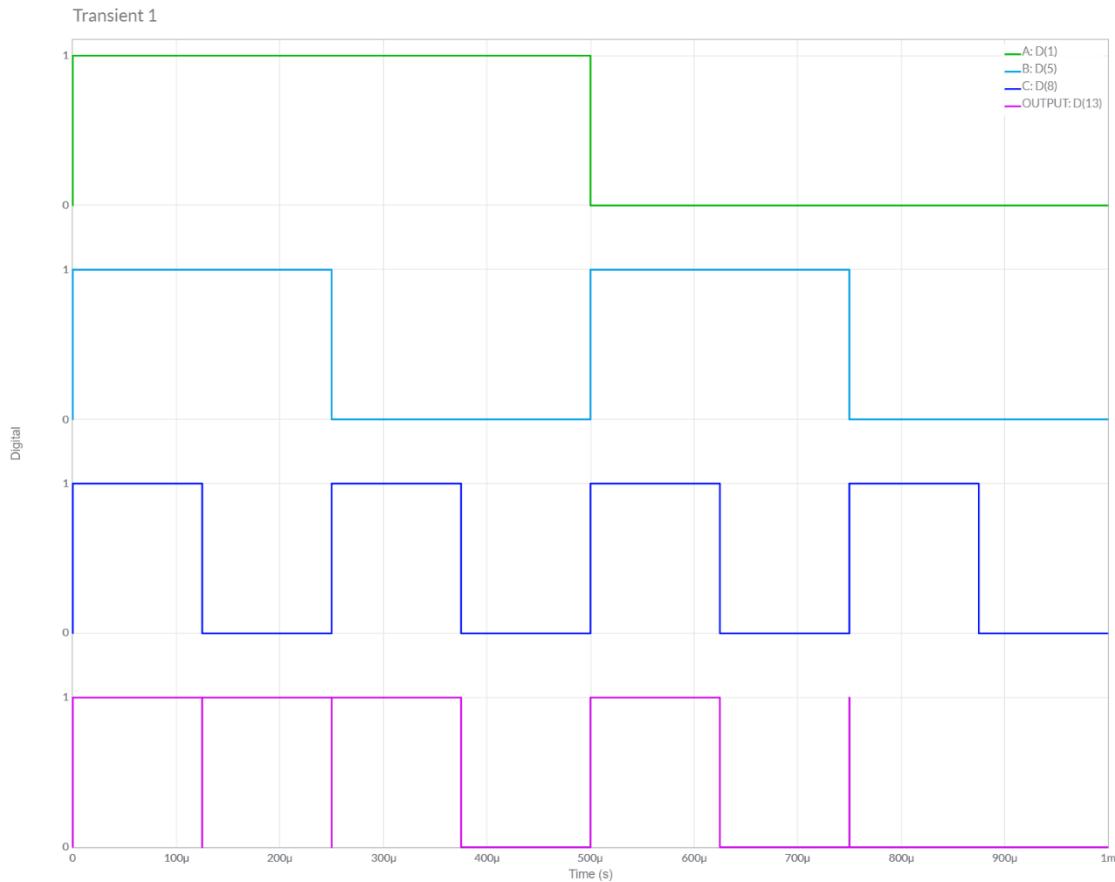




**WAVEFORMS (SUM)**



**WAVEFORMS (CARRY)**





## **CONCLUSIONS**

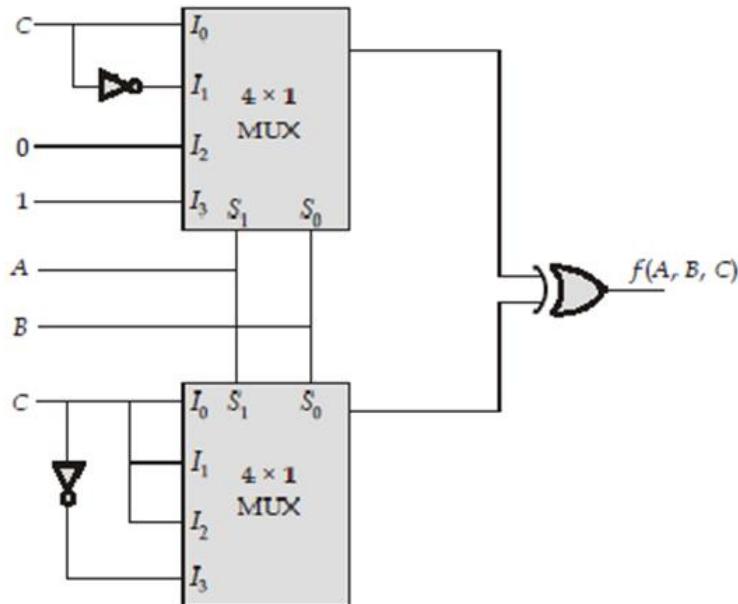
- 1.) In this Experiment, We have studied about Code-Converters[Both Binary to Gray and Gray to Binary], Multiplexors ( $2 \times 1$  &  $4 \times 1$ ), Realized All Basic Gates using  $2 \times 1$  Multiplexor and Some Functional Implementation using Multiplexor on Multisim.
- 2.) We **Verified** the Theoretical Knowledge gained above by *implementing the above Circuits in Multisim [verified it with their Truth Table]* and successfully Got the **Desired Output**. Hence the *Experiment has Been Completed Successfully*.



# **ASSIGNMENT-12**

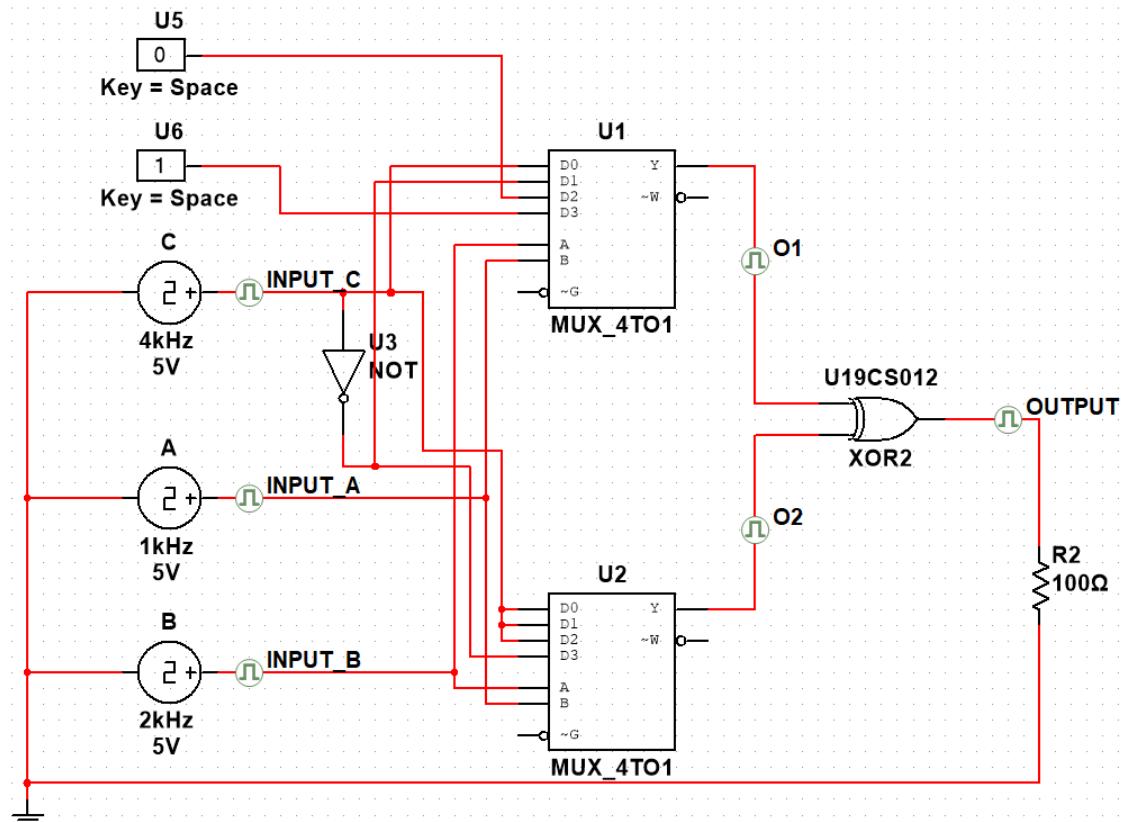
U19CS012

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



**"A" Batch Question**

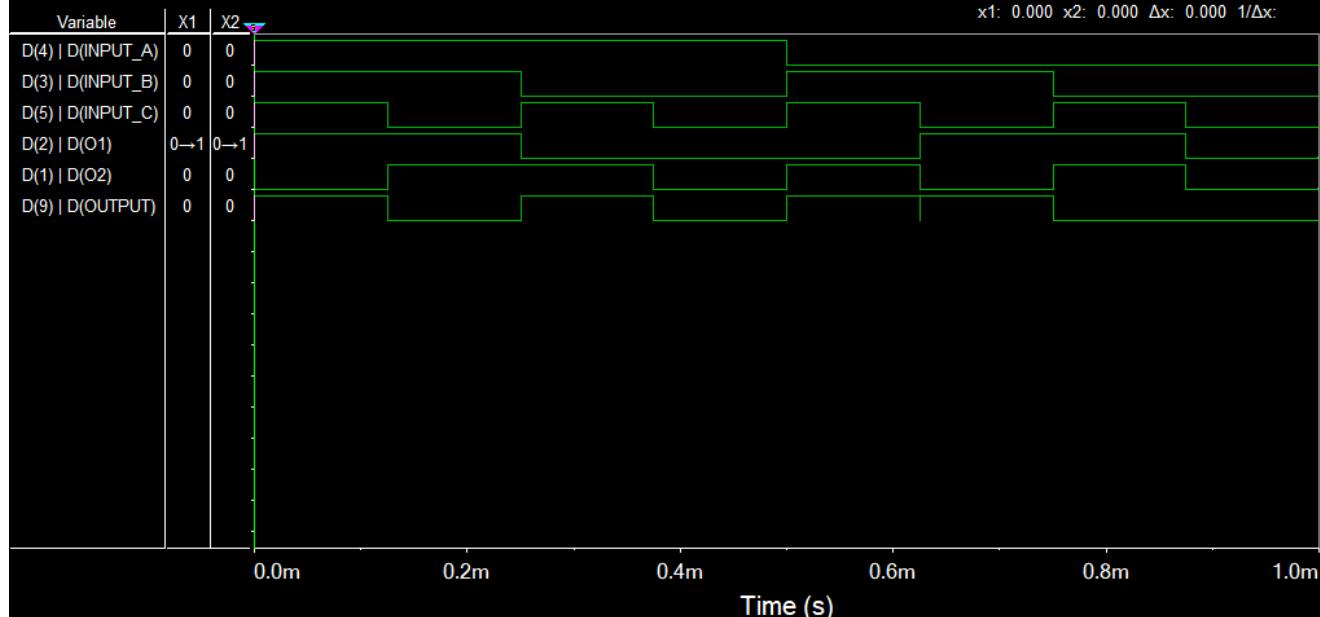
### **Circuit Diagram [Multisim Implementation]**



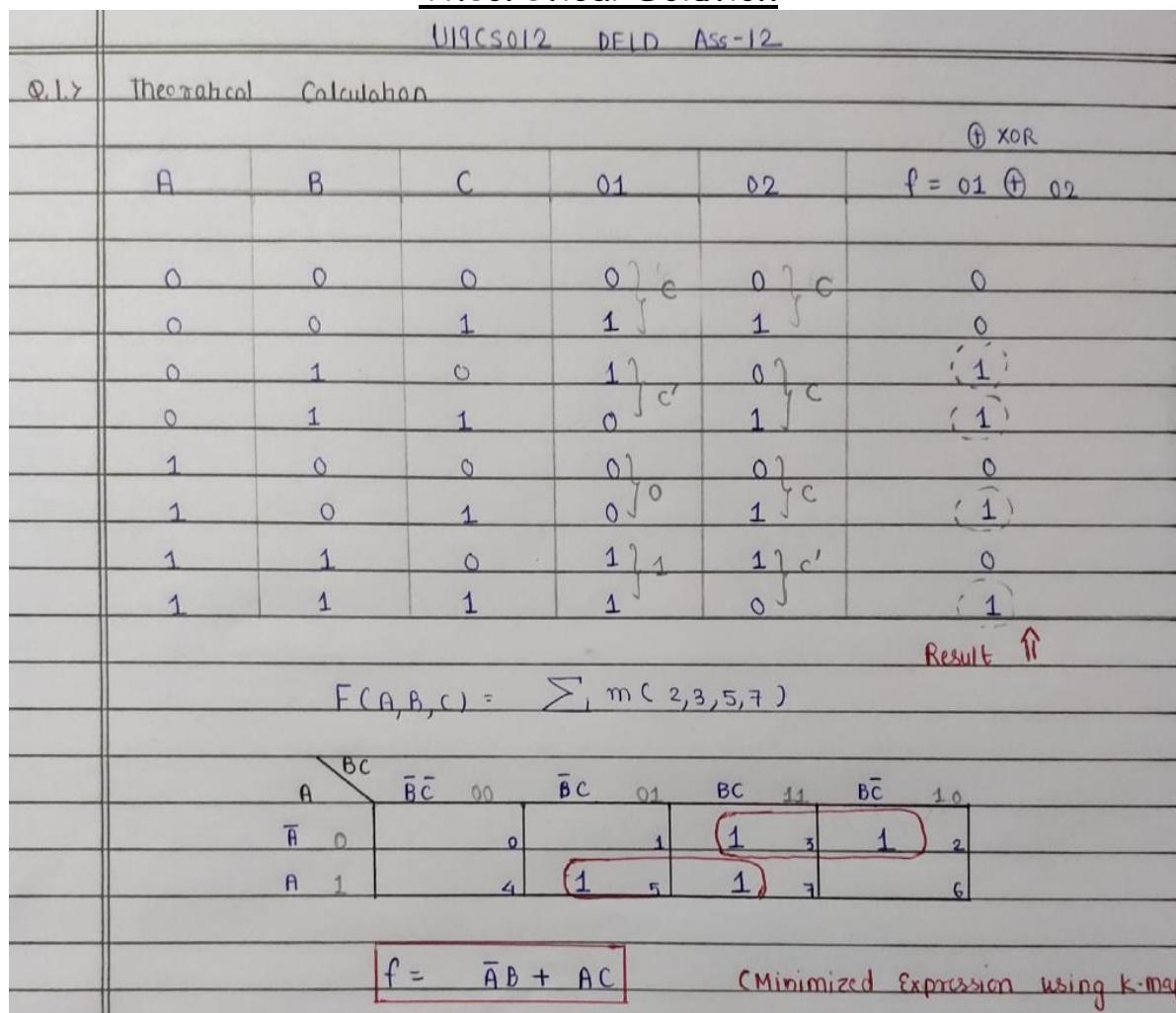
**Grapher Image [Transient]**



**Assignment\_12\_Q1  
Transient**



**Theoretical Solution**

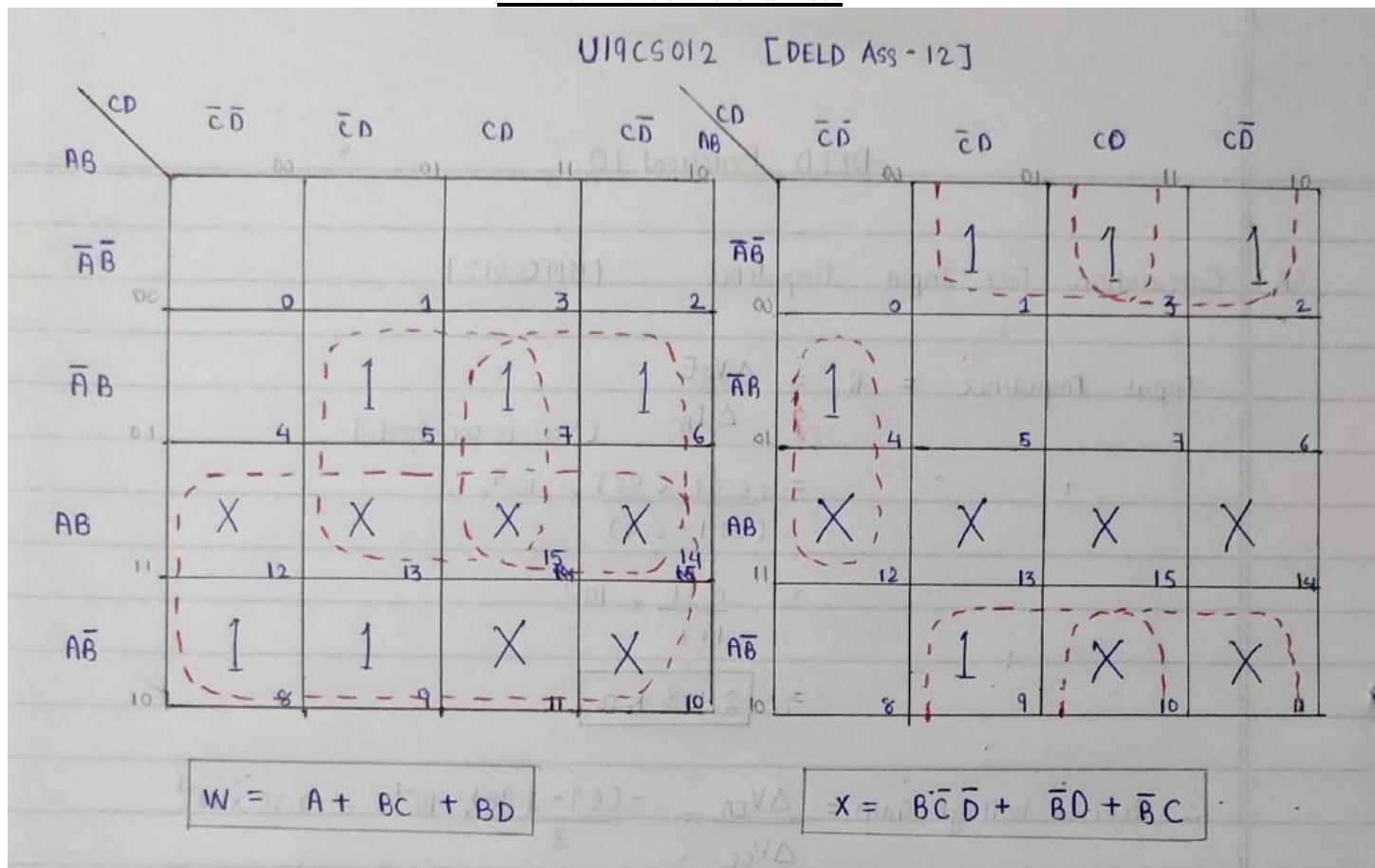


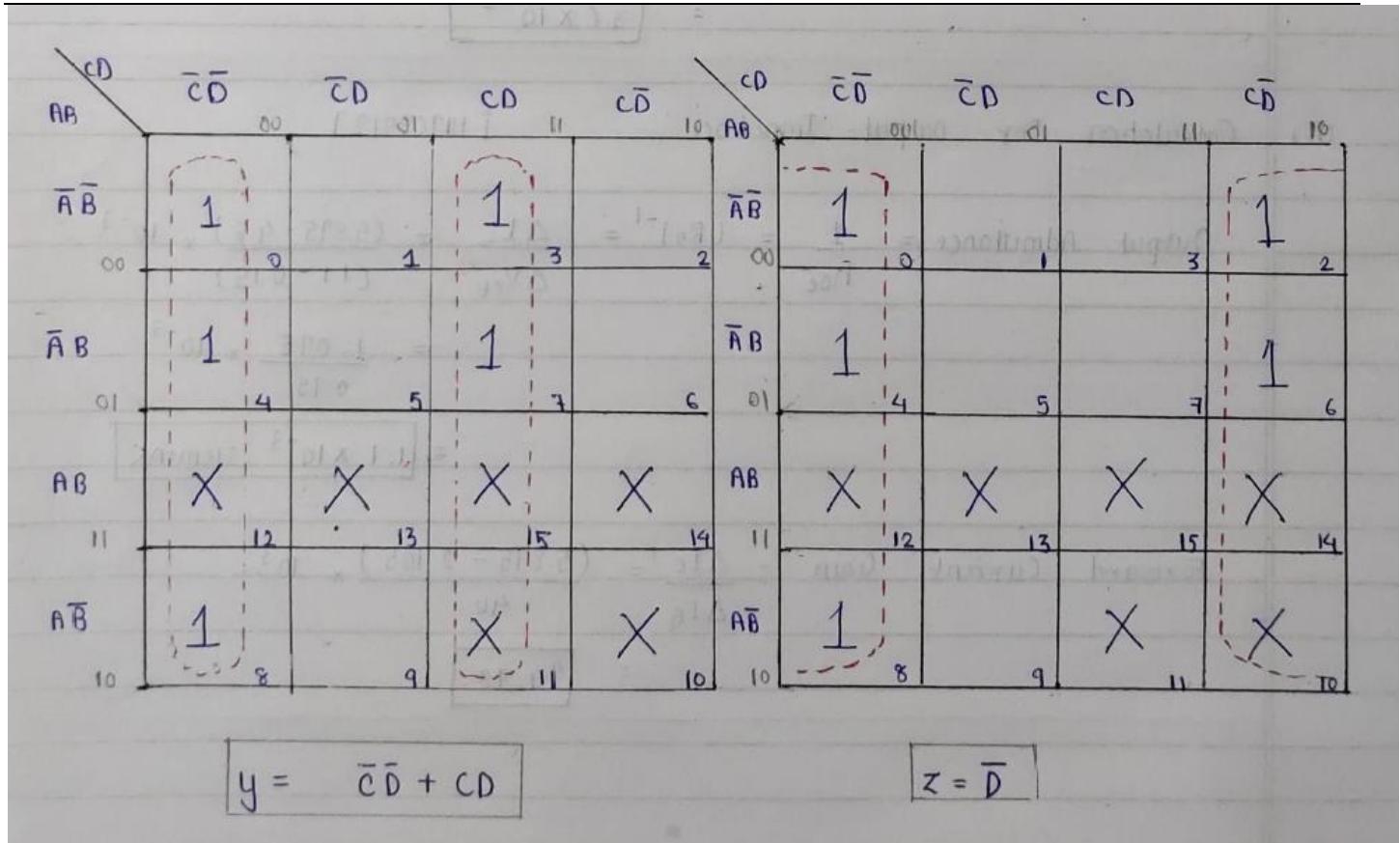


**2. Design, implement and verify using Multisim: BCD to Excess - 3 Code Converter**

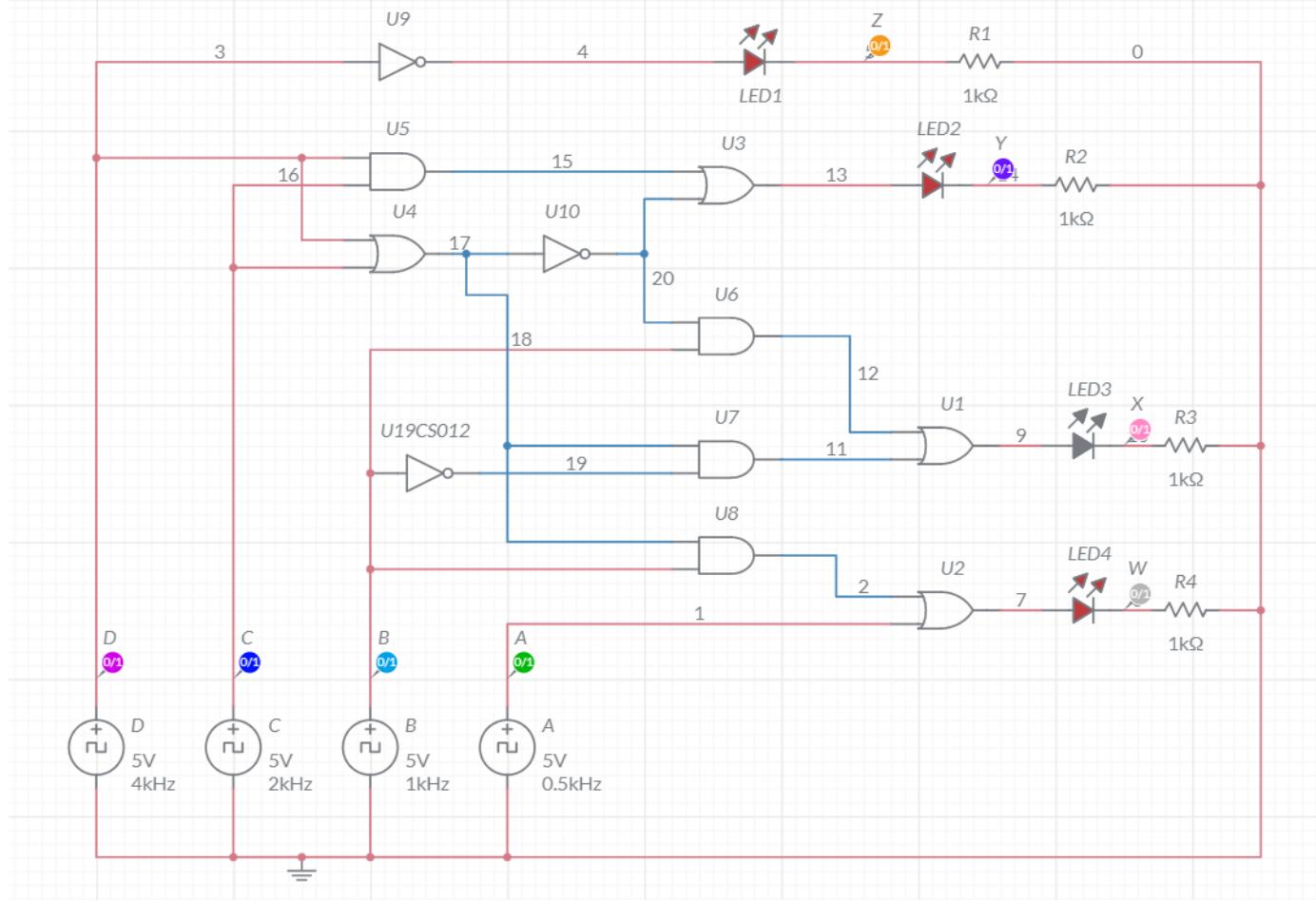
BCD(8421)				Excess-3			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

**Theoretical Solution**



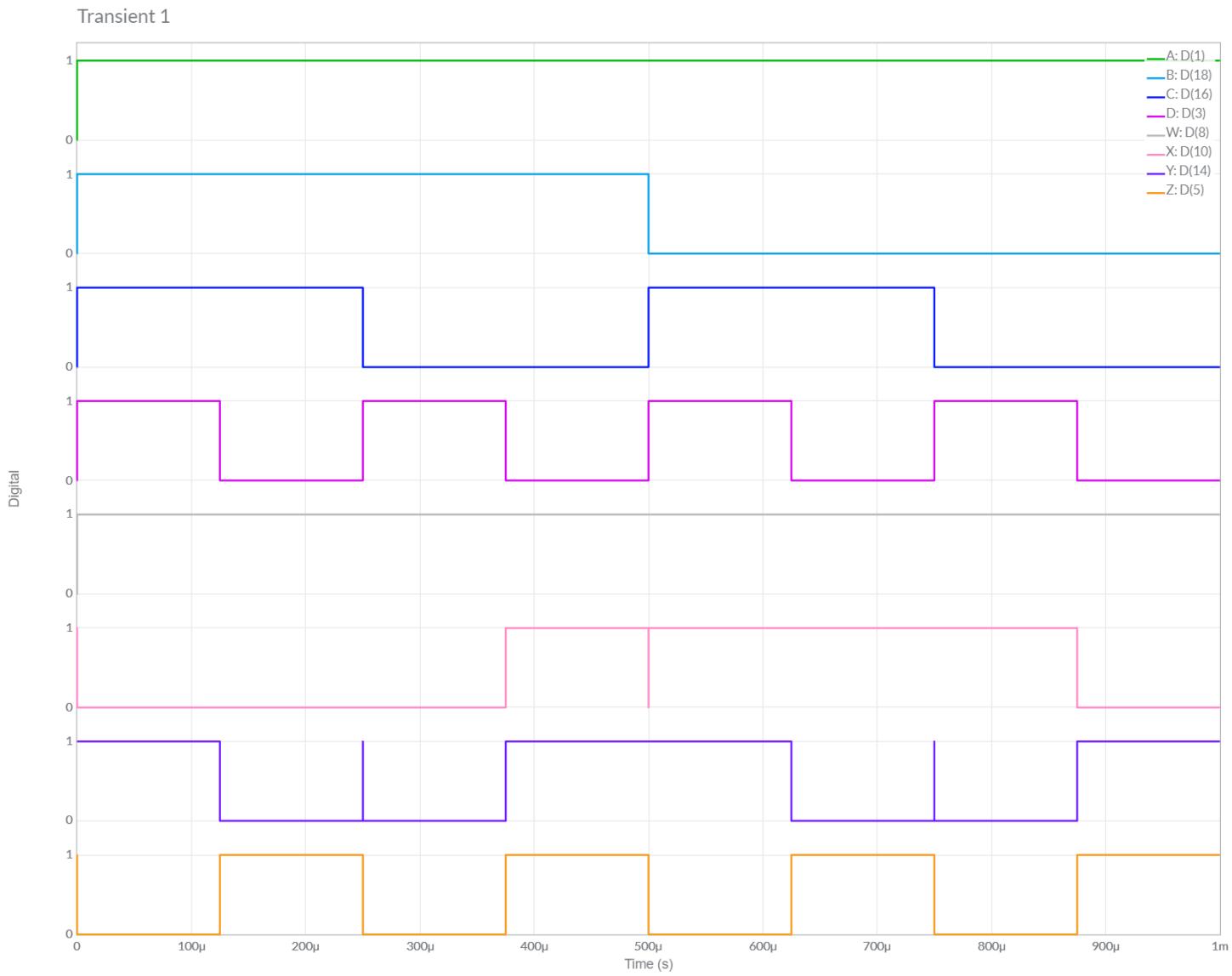


**Circuit Diagram [Multisim Implementation]**





### Grapher Image [Transient]



### D.) CONCLUSION:

We have Successfully Implemented Particular Circuit [ $A'B + AC$ ] and BCD to Excess-3 Convertor and verified our **MULTISIM Outputs** and **Results** from **Theoretical Calculations**.

Hence Results Both Theoretical Calculation and Multisim Implementation have been verified to be same and The Experiment has Been Successfully Performed.



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:**

Filters may be classified as either digital or analog.

Digital filters are implemented using a digital computer or special purpose digital hardware.

Analog filters may be classified as either passive or active and are usually implemented with R, L, and C components and operational amplifiers.

An Active filter is one that, along with R, L, and C components, also contains an energy source, such as that derived from an operational amplifier.

A Passive filter is one that contains only R, L, and C components. It is not necessary that all three be present.

L is often omitted (on purpose) from passive filter design because of the size and cost of inductors - and they also carry along an R that must be included in the design.

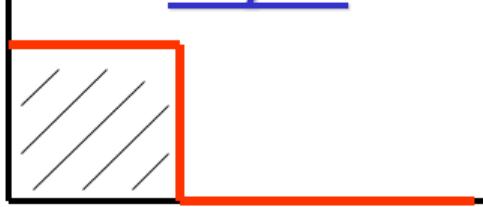
It will be shown later that the ideal filter, sometimes called a "brickwall" filter, can be approached by making the order of the filter higher and higher.

The order here refers to the order of the polynomial(s) that are used to define the filter.

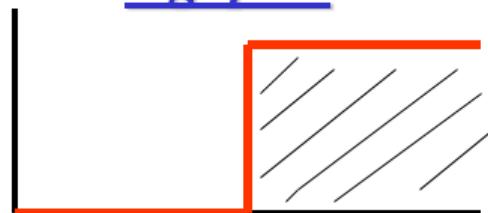


**Four types of filters - "Ideal"**

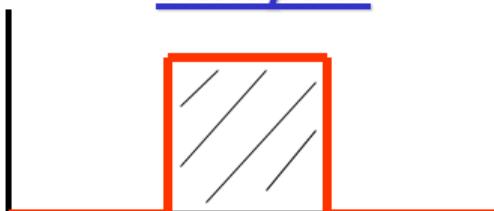
**lowpass**



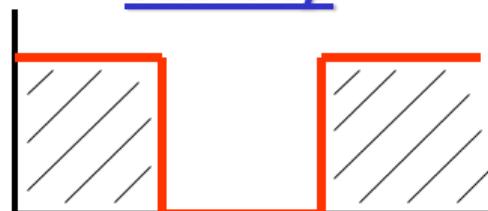
**highpass**



**bandpass**

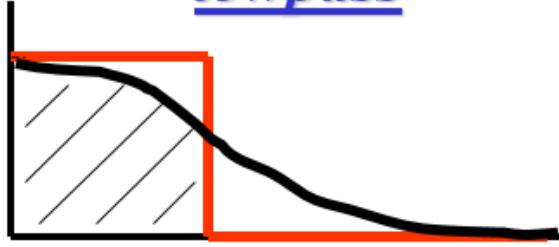


**bandstop**

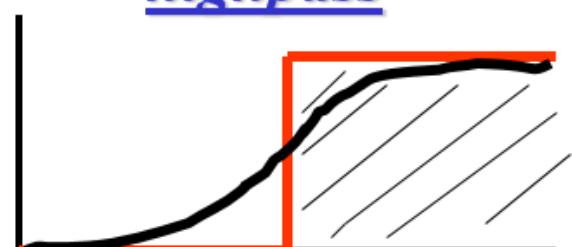


**Realistic Filters:**

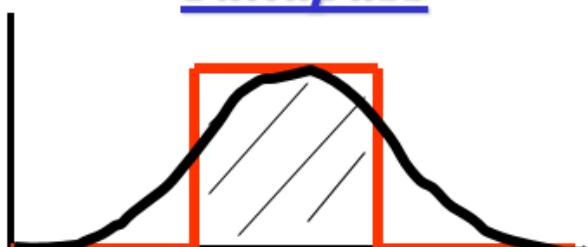
**lowpass**



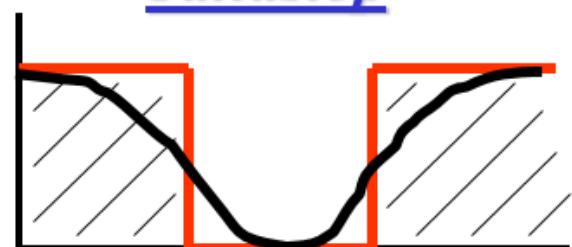
**highpass**



**bandpass**



**bandstop**



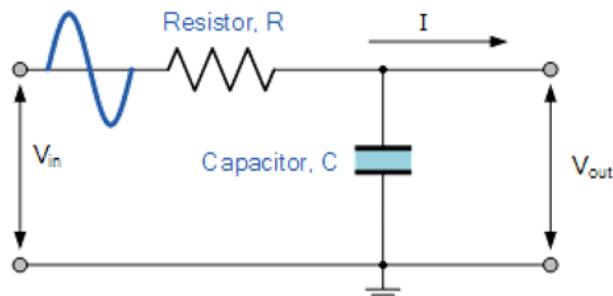


### (A) RC LOW-PASS FILTER

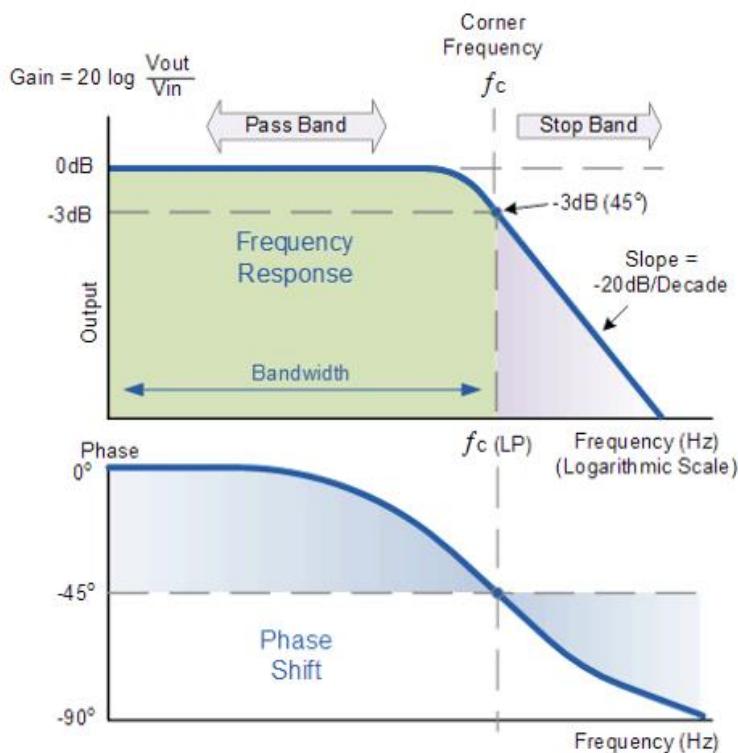
A low-pass filter allows for easy passage of low-frequency signals from source to load, and difficult passage of high-frequency signals.

The cutoff frequency for a low-pass filter is that frequency at which the output (load) voltage equals 70.7% of the input (source) voltage.

Above the cutoff frequency, the output voltage is lower than 70.7% of the input, and vice versa.



First order low pass filter



Cut-off Frequency and Phase Shift

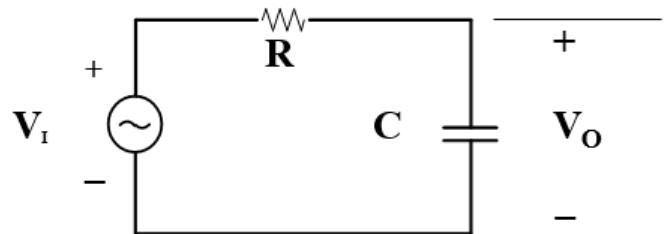
$$f_c = \frac{1}{2\pi RC}$$

$$\text{Phase Shift } \varphi = -\arctan(2\pi f RC)$$



## Low Pass Filter

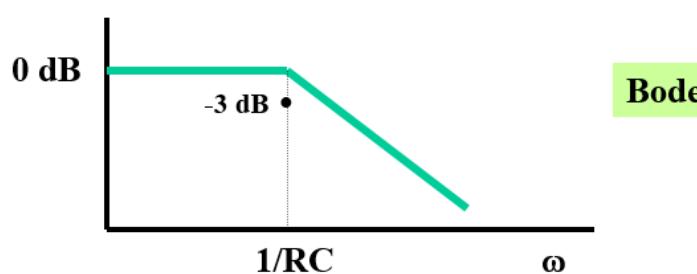
Consider the circuit below.



**Low pass filter circuit**

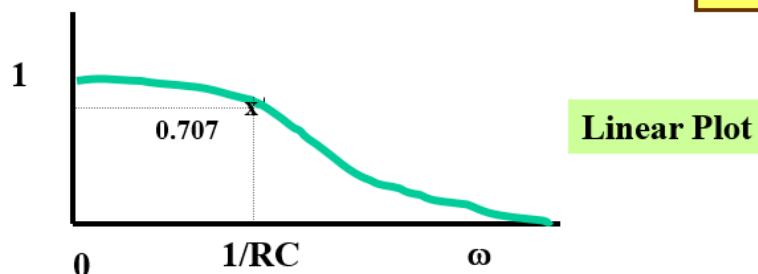
$$\frac{V_o(jw)}{V_i(jw)} = \frac{\frac{1}{jwC}}{R + \frac{1}{jwC}} = \frac{1}{1 + jwRC}$$

## Low Pass Filter



**Bode**

**Passes low frequencies  
Attenuates high frequencies**



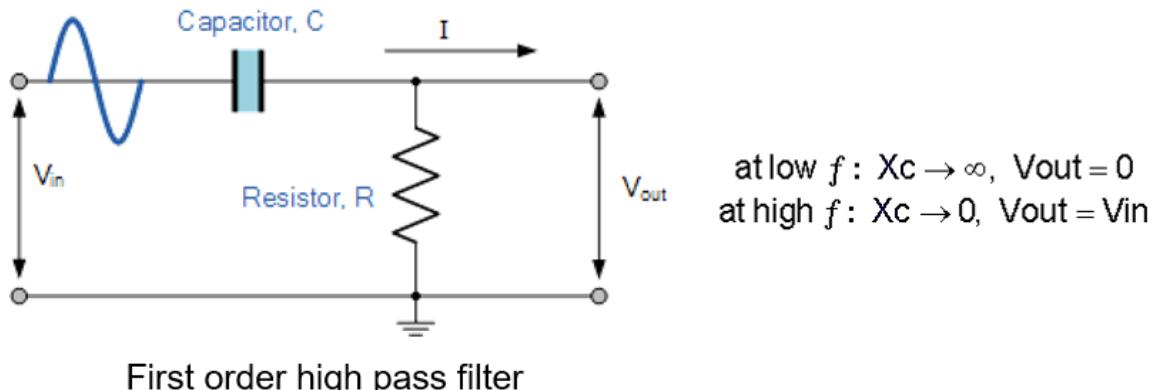
**Linear Plot**

**(B) RC HIGH-PASS FILTER**

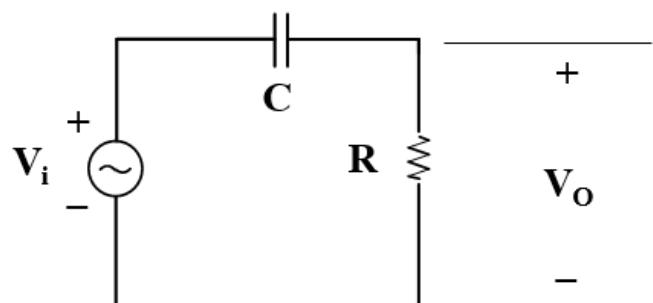
A high-pass filter allows for easy passage of high-frequency signals from source to load, and difficult passage of low-frequency signals.

The cutoff frequency for a high-pass filter is that frequency at which the output (load) voltage equals 70.7% of the input (source) voltage.

Above the cutoff frequency, the output voltage is greater than 70.7% of the input, and vice versa.

**High Pass Filter**

Consider the circuit below.

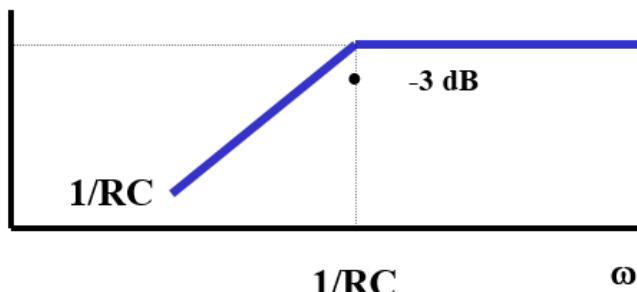
**High Pass Filter**

$$\frac{V_o(jw)}{V_i(jw)} = \frac{R}{R + \frac{1}{jwC}} = \frac{jwRC}{1 + jwRC}$$



## High Pass Filter

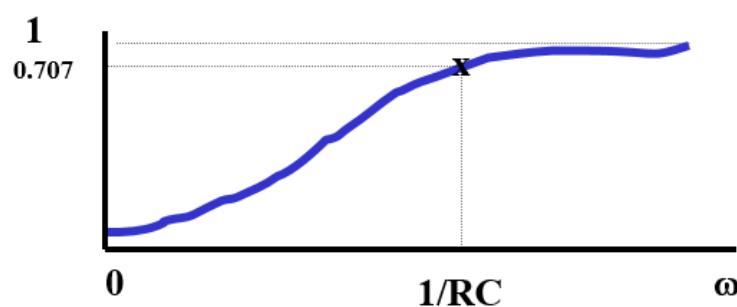
**Bode**



**Passes high frequencies**

**Attenuates low frequencies**

**Linear**

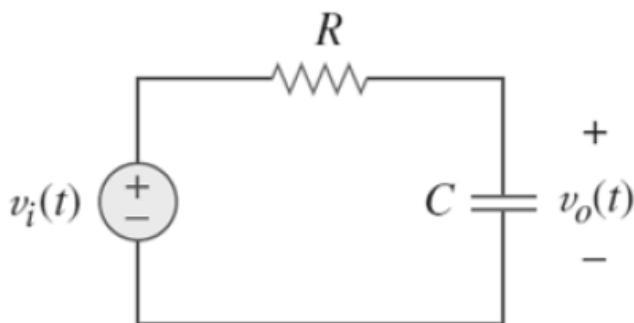


### (C) LOW PASS AS AN INTEGRATOR

"If the time constant value is much greater than the time period of the input signal than the RC low pass circuit will act as an **Integrator**"

$$RC \gg T$$

Under this circumstances the voltage drop across  $C$  will be very small in comparison to the drop across  $R$ .



$$V_i = iR$$

$$i = \frac{V_i}{R}$$

$$V_o = \frac{1}{C} \int idt$$

$$V_o = \frac{1}{C} \int \frac{V_i}{R} dt$$

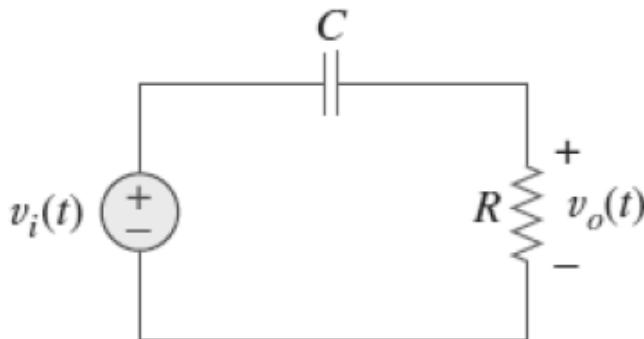
$$V_o = \frac{1}{RC} \int V_i dt$$

**(D) HIGH PASS AS A DIFFERENTIATOR**

"If the time constant value is much smaller than the time period of the input signal than the RC High pass circuit will act as a **Differentiator**"

$$RC \ll T$$

Under this circumstances the voltage drop across R will be very small in comparison with the drop across C. Hence we may consider that the total input  $V_i$  appears across C, so that the current is determined entirely by the capacitance.



$$i = C \frac{dV_i}{dt}$$

$$V_o = iR$$

$$V_o = RC \frac{dV_i}{dt}$$

**RC Integrator**

- ▶ Sine – (-Cosine)
- ▶ Triangular – Sine
- ▶ Rectangular Wave - Triangle

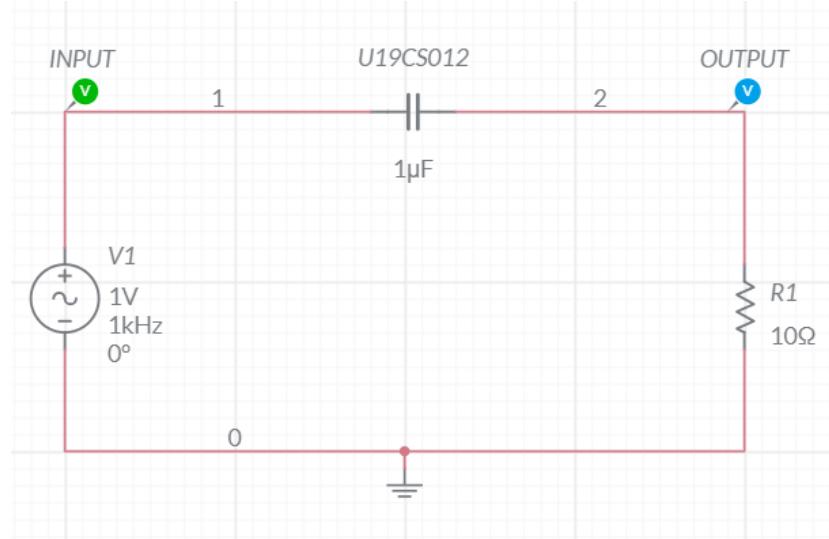
**RC Differentiator**

- ▶ Sine – Cosine
- ▶ Triangular Wave – Square Wave
- ▶ Rectangular Wave - Spikes

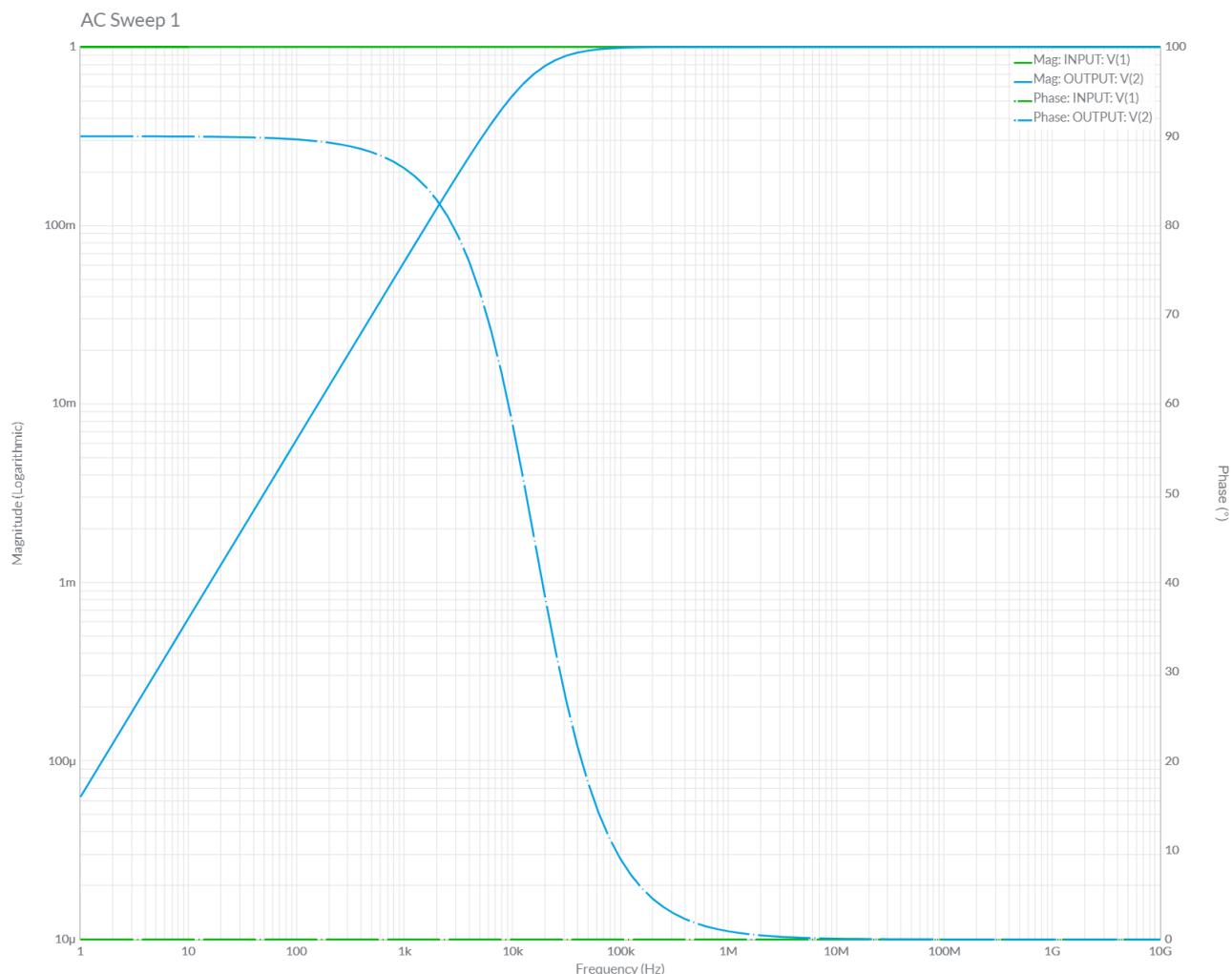


**SIMULATION SCREENSHOTS**

**Circuit Diagram of RC - High Pass Filter**



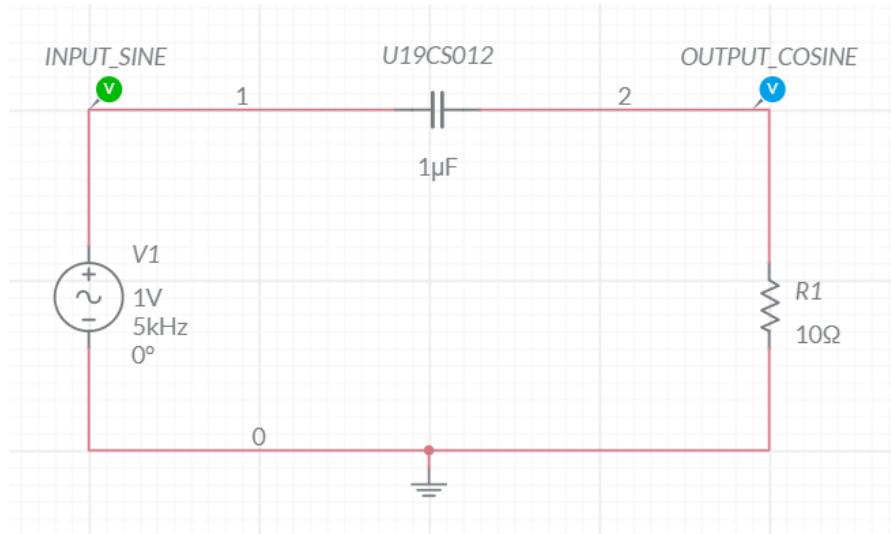
**Frequency Response Plot of High Pass Filter**



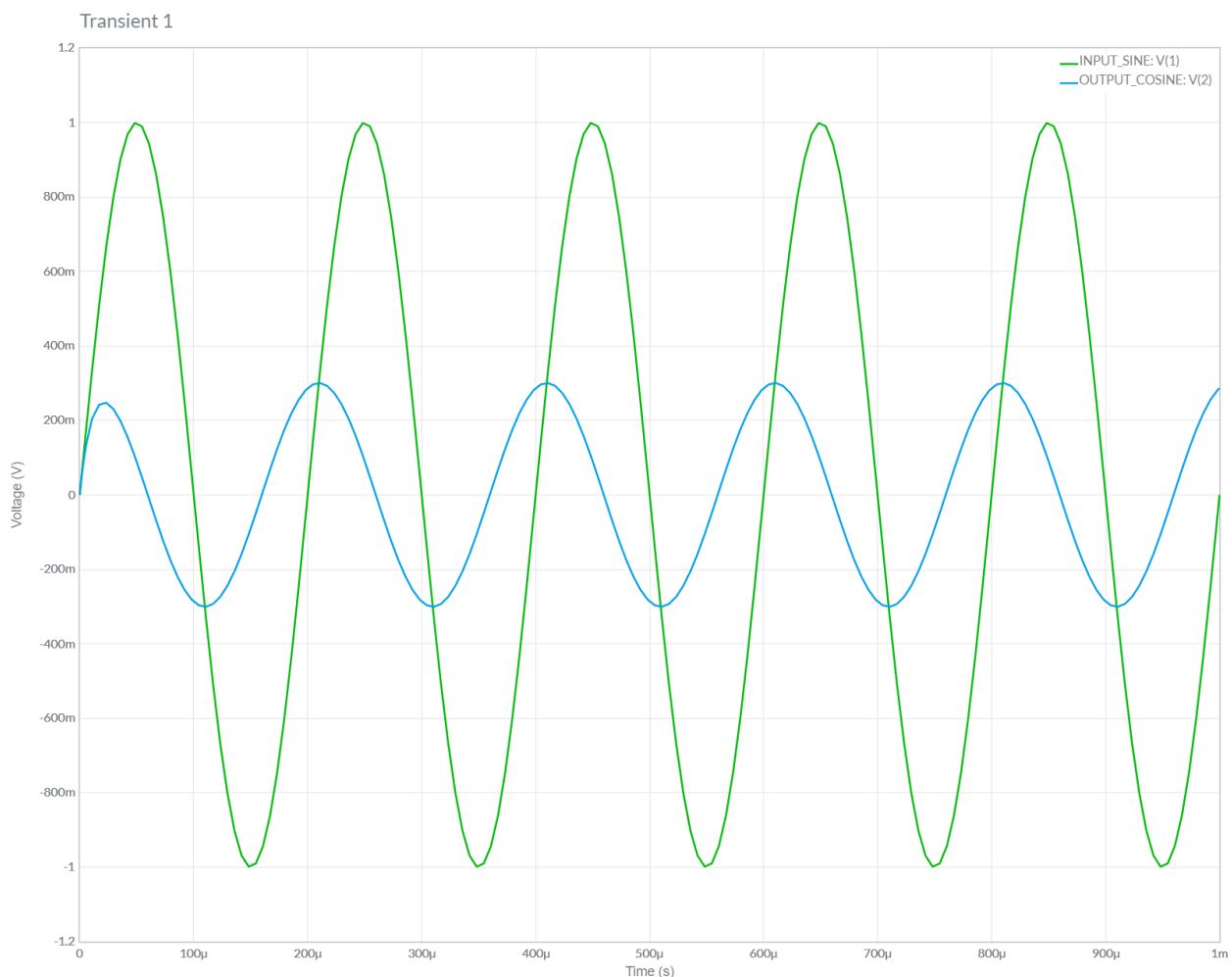


**Output of Differentiator [T > > RC]**

**Input - Sine Wave [Circuit Diagram]**

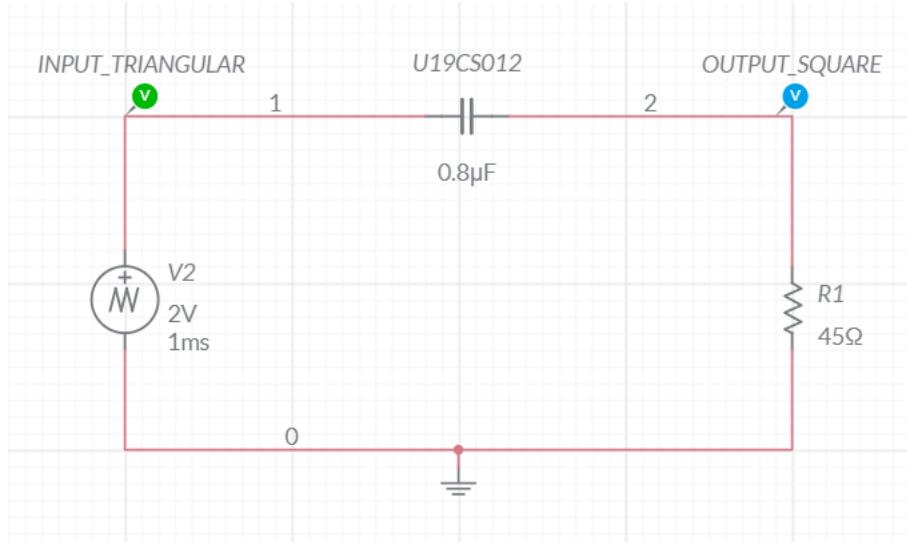


**Grapher Image [Differentiation of (Sine) = Cosine]**

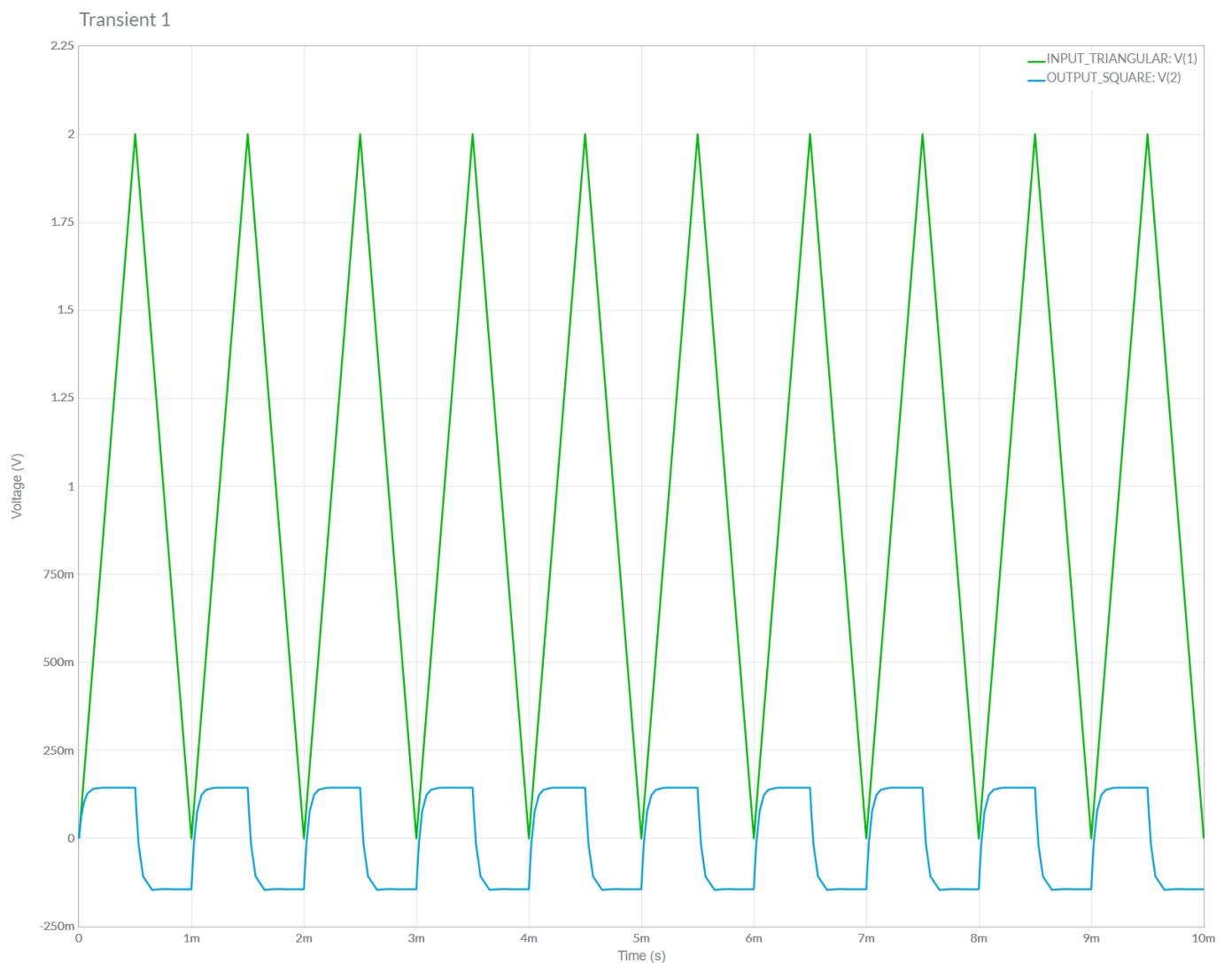




**Input - Triangular Wave [Circuit Diagram]**

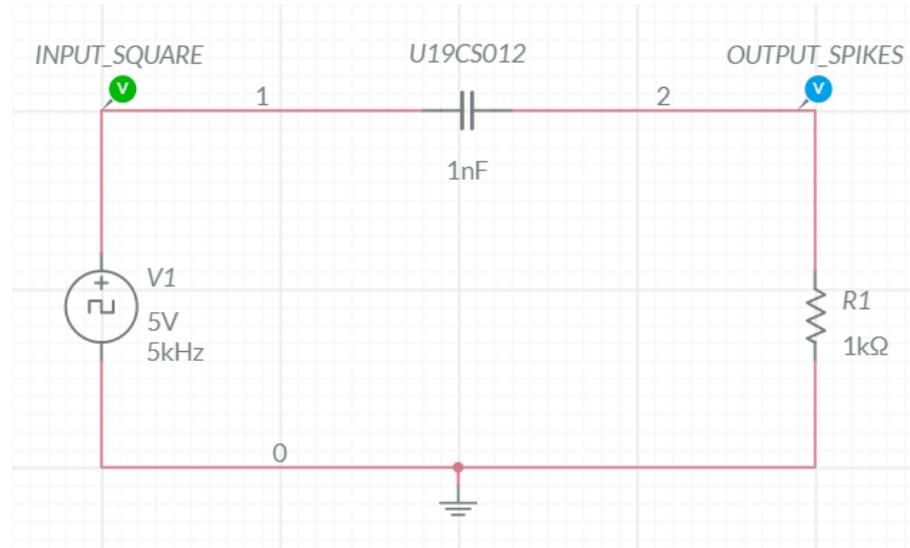


**Grapher Image [Differentiation of (Triangular) = Square]**

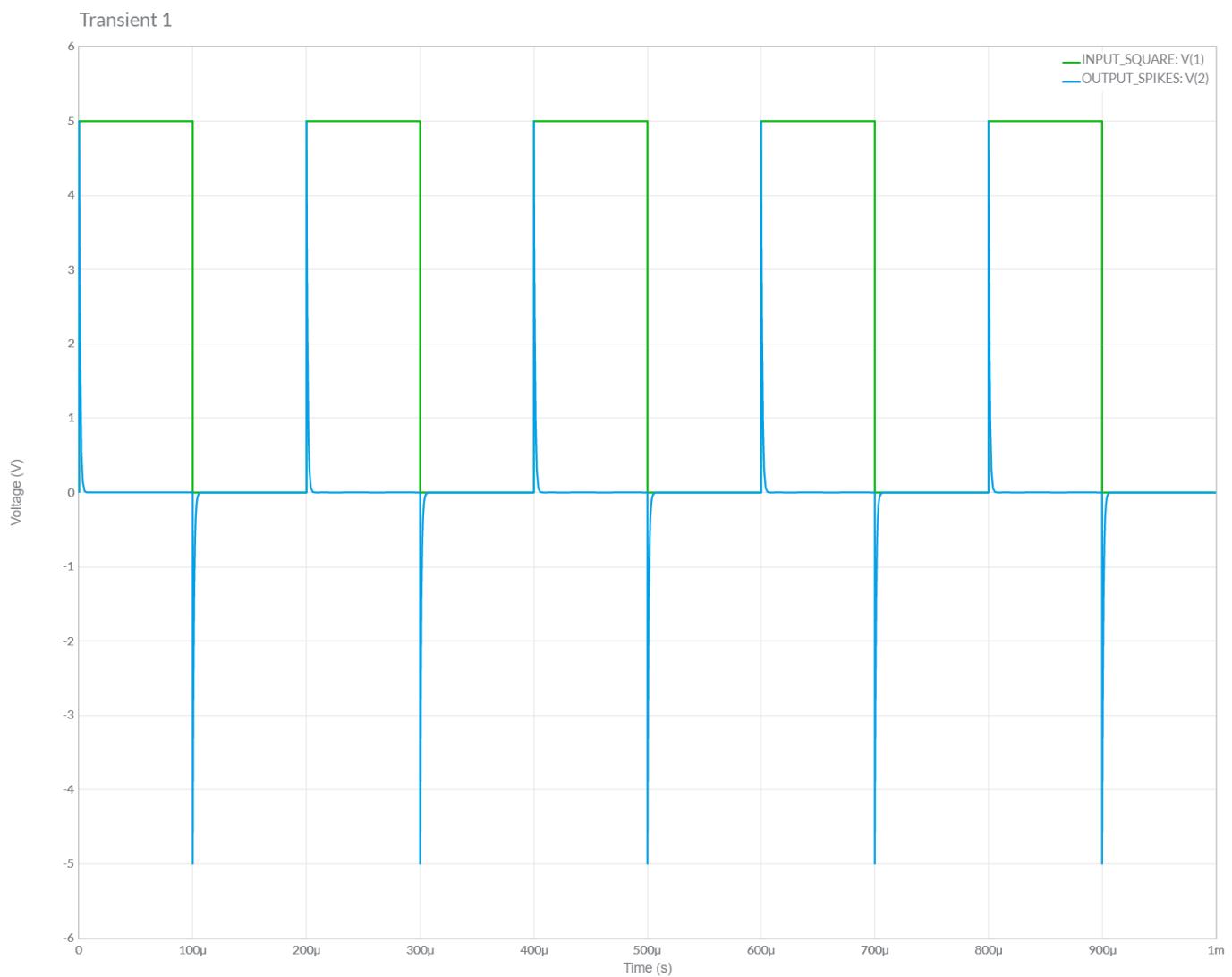




**Input - Square Wave [Circuit Diagram]**

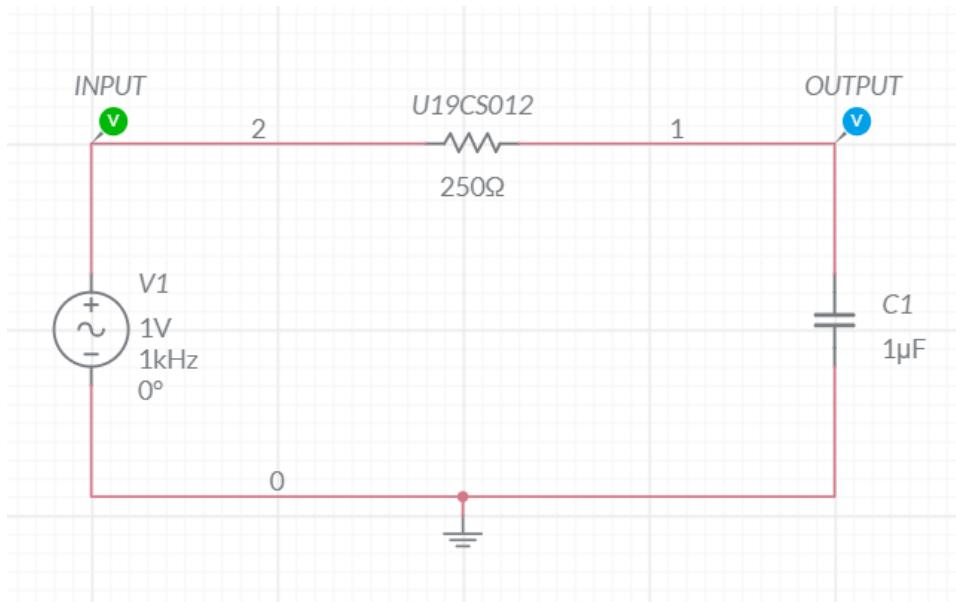


**Grapher Image [Differentiation of (Rectangular) = Spikes]**

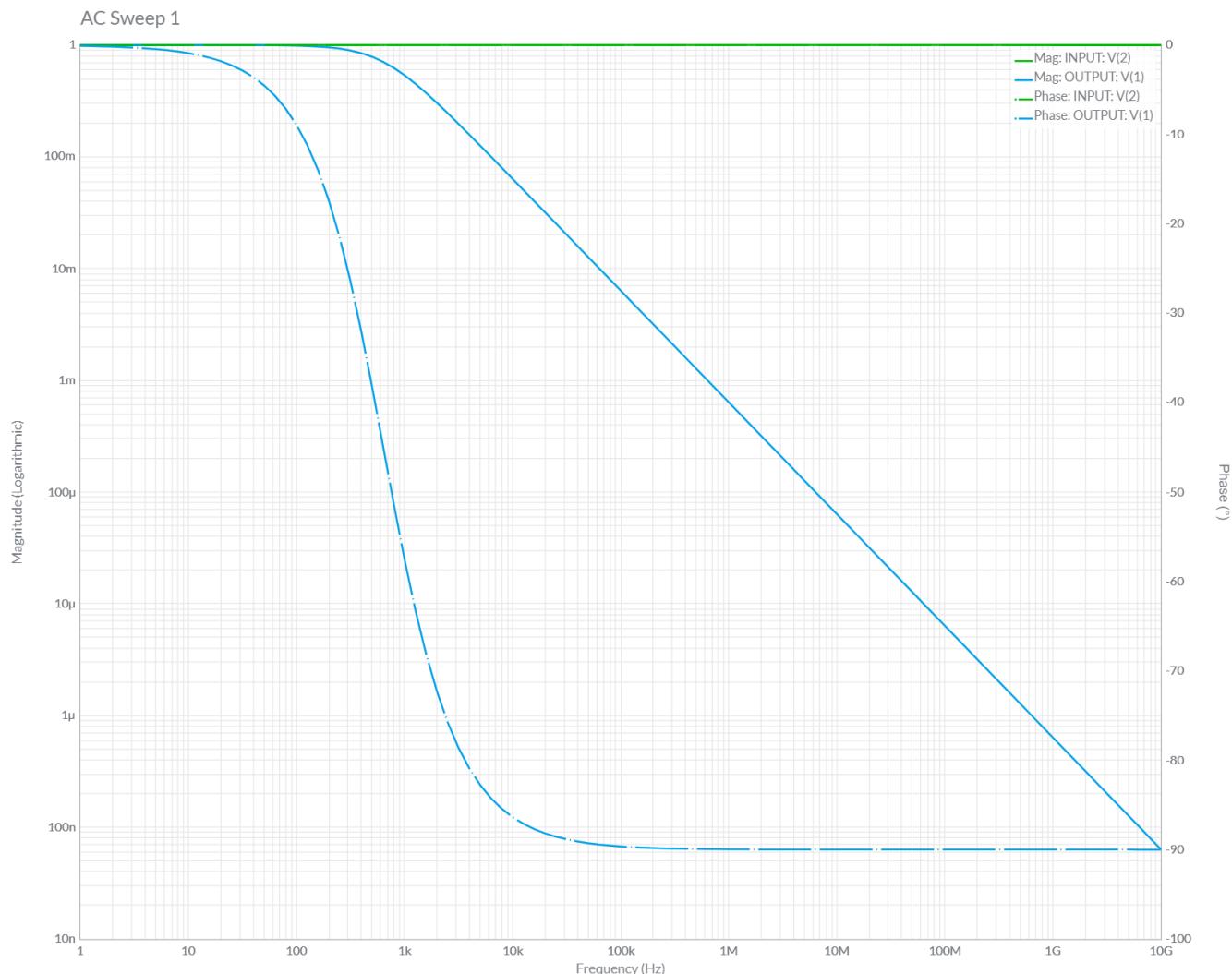




### Circuit Diagram of RC - Low Pass Filter



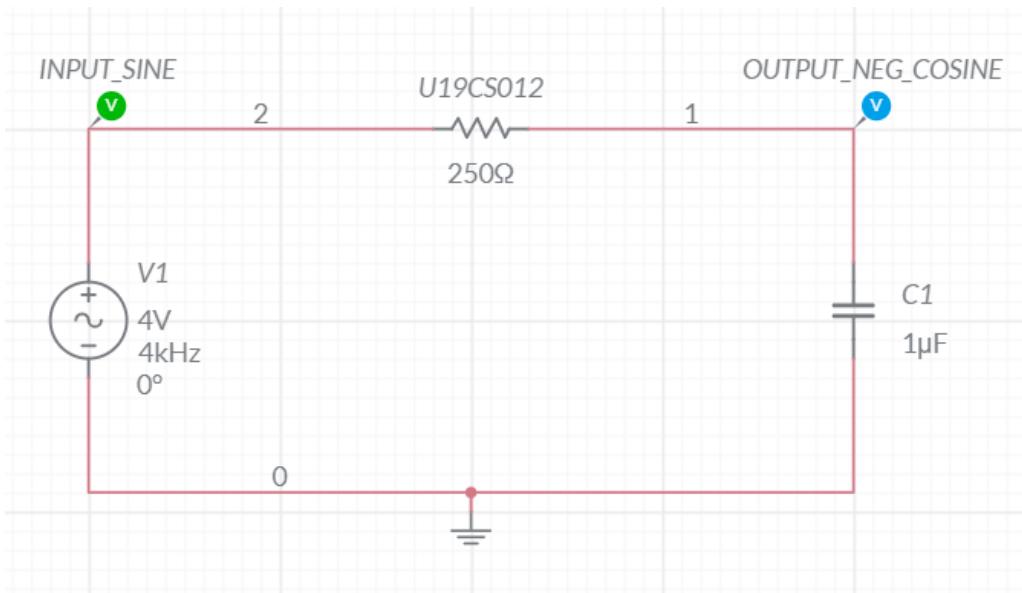
### Frequency Response Plot of Low Pass Filter



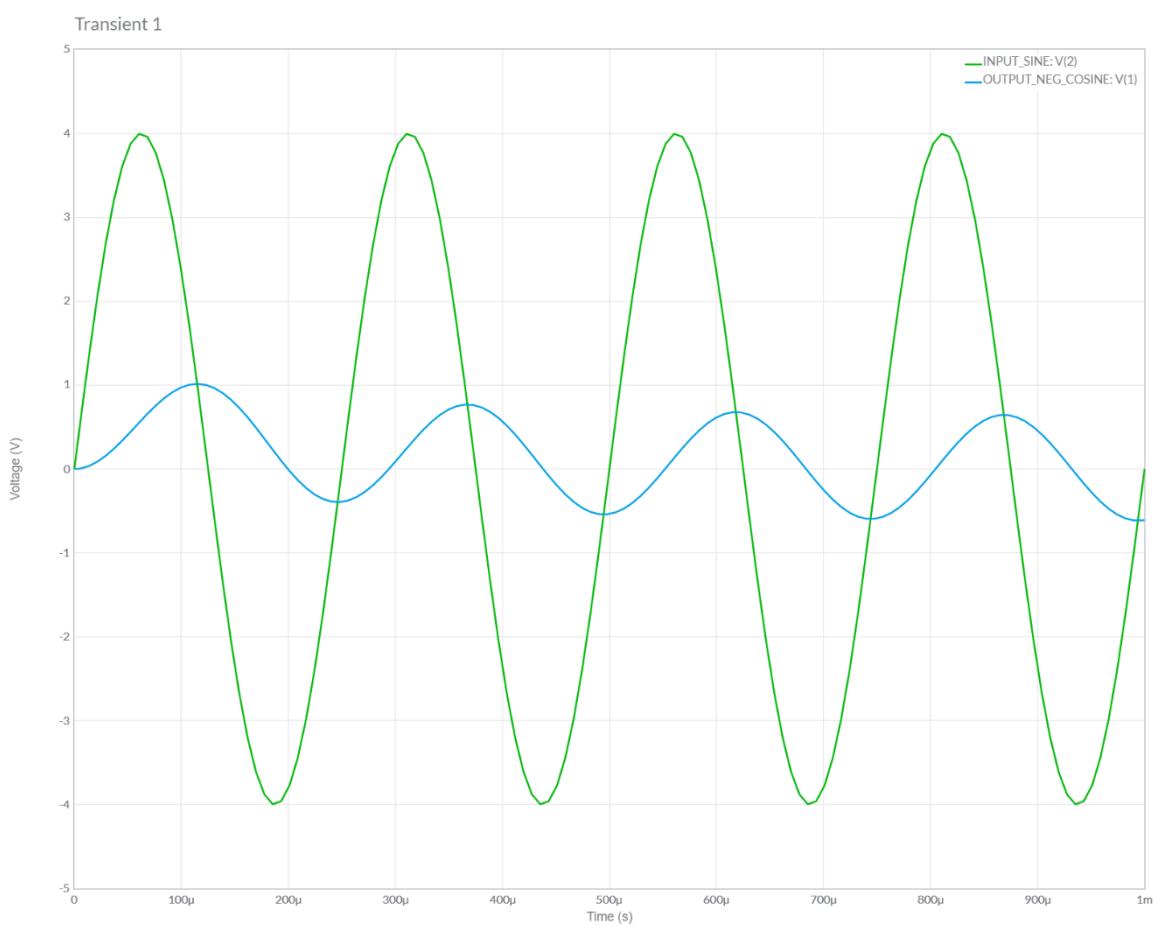


**Output of Integrator [RC > T]**

**Input - Sine Wave [Circuit Diagram]**

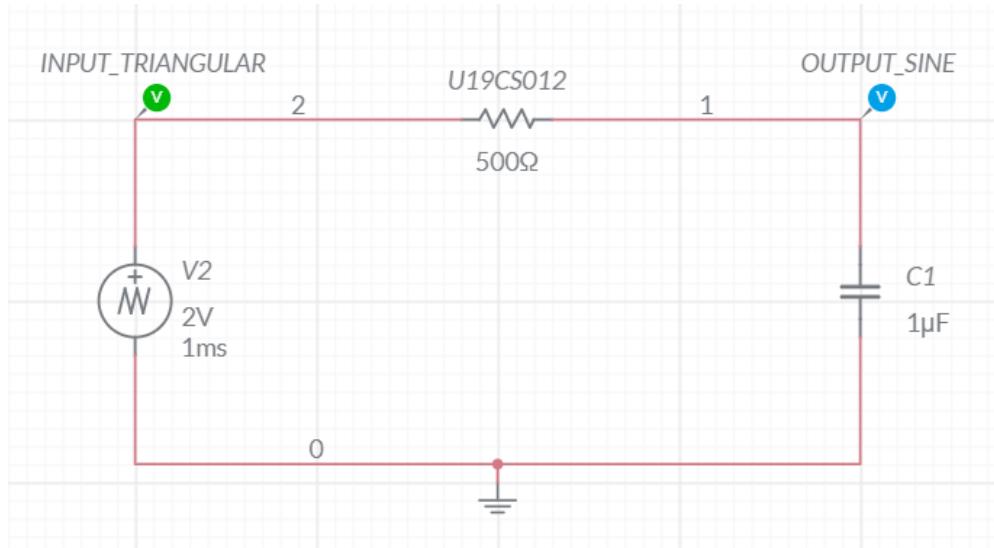


**Grapher Image [Integration of (Sine) = - Cosine]**

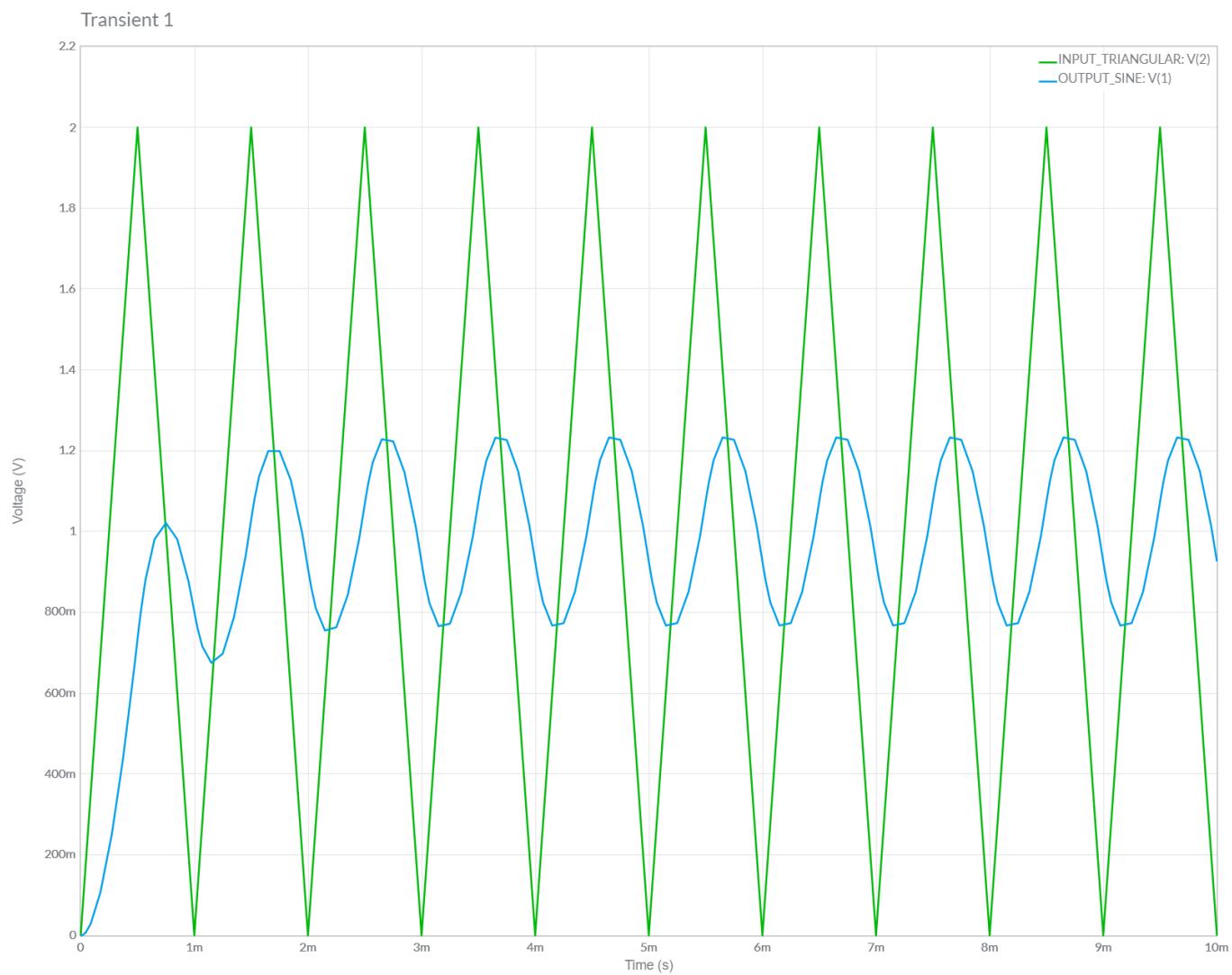




**Input - Triangular Wave [Circuit Diagram]**

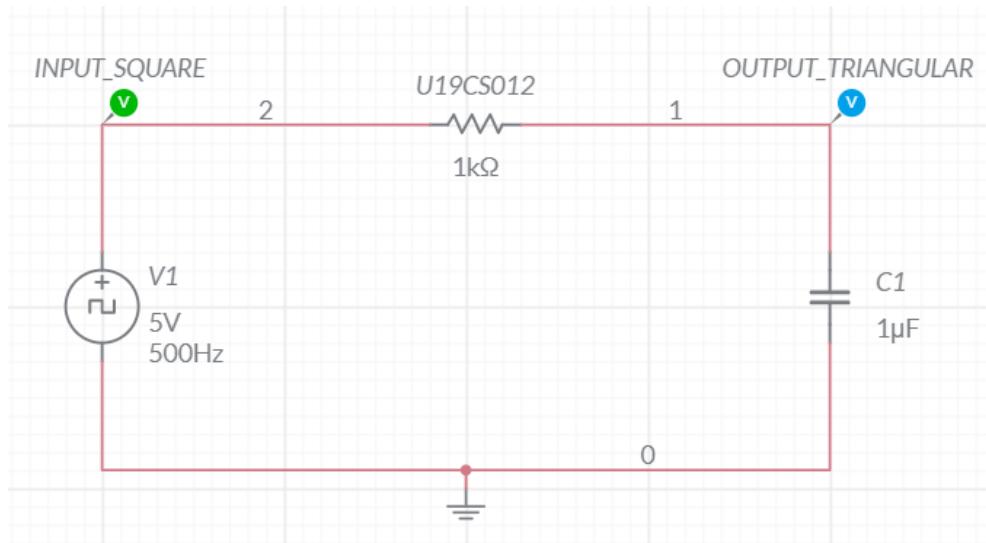


**Grapher Image [Integration of (Triangular) = Sine]**

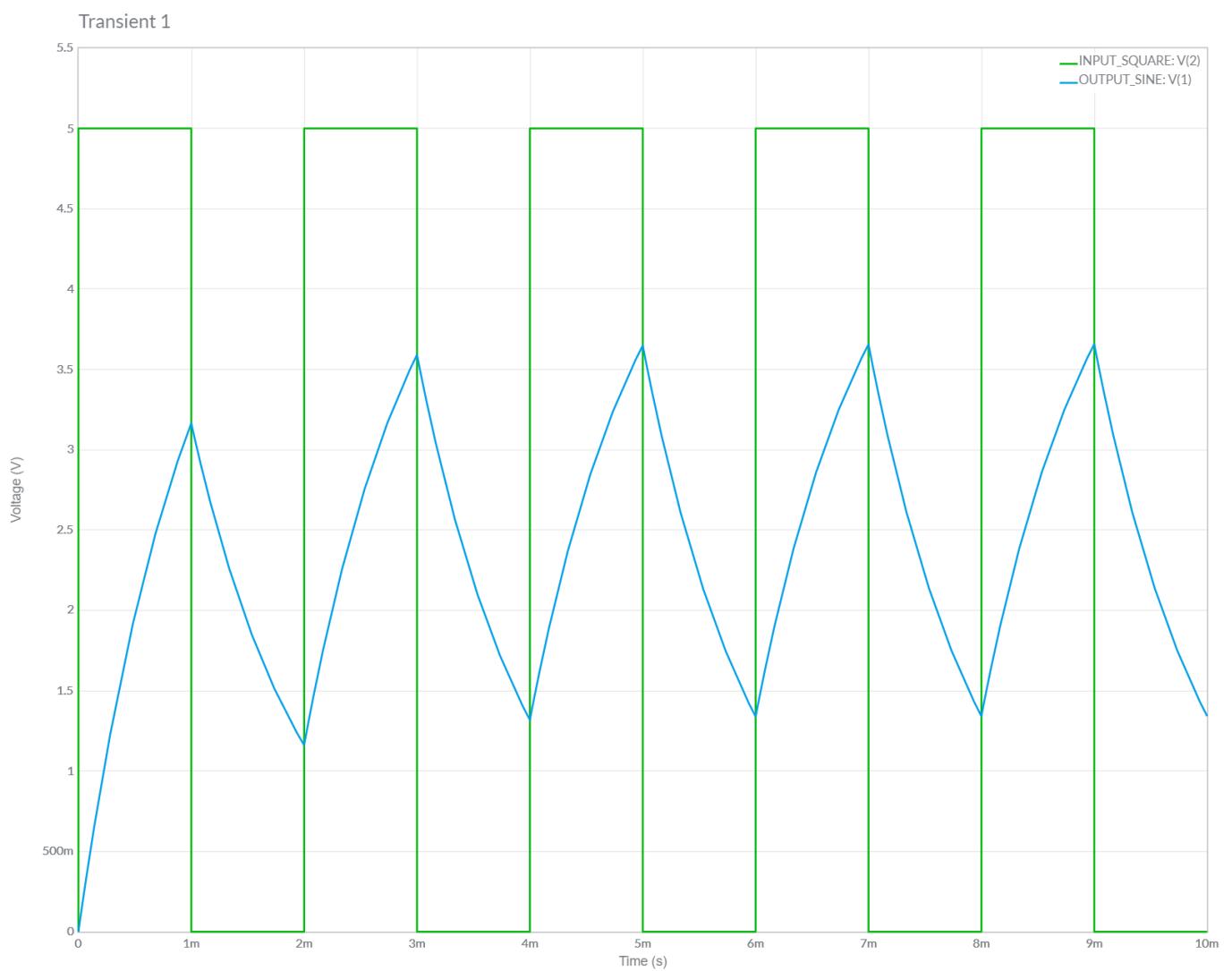




**Input - Square Wave [Circuit Diagram]**



**Grapher Image [Integration of (Square) = Triangular]**





## **CONCLUSIONS**

1.) In this Experiment, We have studied about Passive RC High Pass and Low Pass Filter and Also Observed the Working of Low Pass Filter as an Integrator and High Pass Filter as a Differentiator.

2.) We Verified the Theoretical Knowledge gained above by *implementing the above Circuits in all Three Input Cases: Sine Wave, Triangular Wave and Square Wave* in Multisim and successfully got the Desired Output.  
Hence the Experiment has Been Completed Successfully.

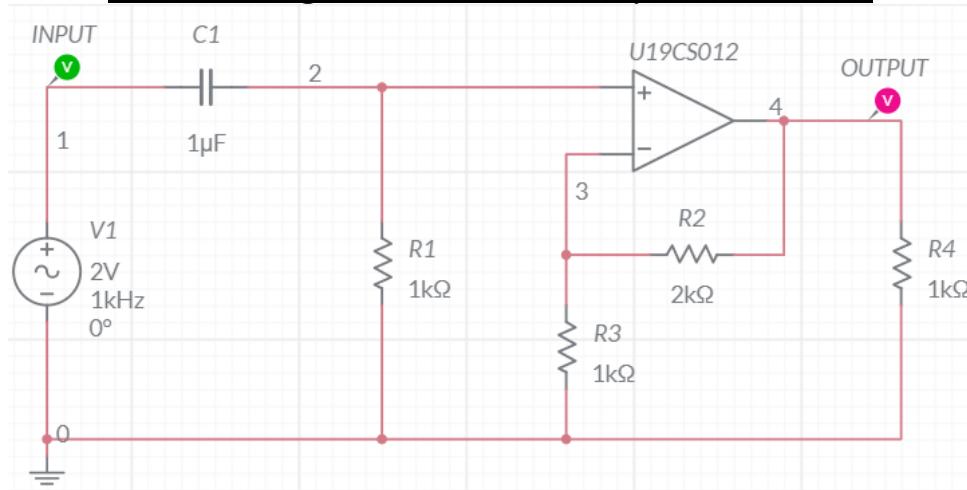


# **ASSIGNMENT-13**

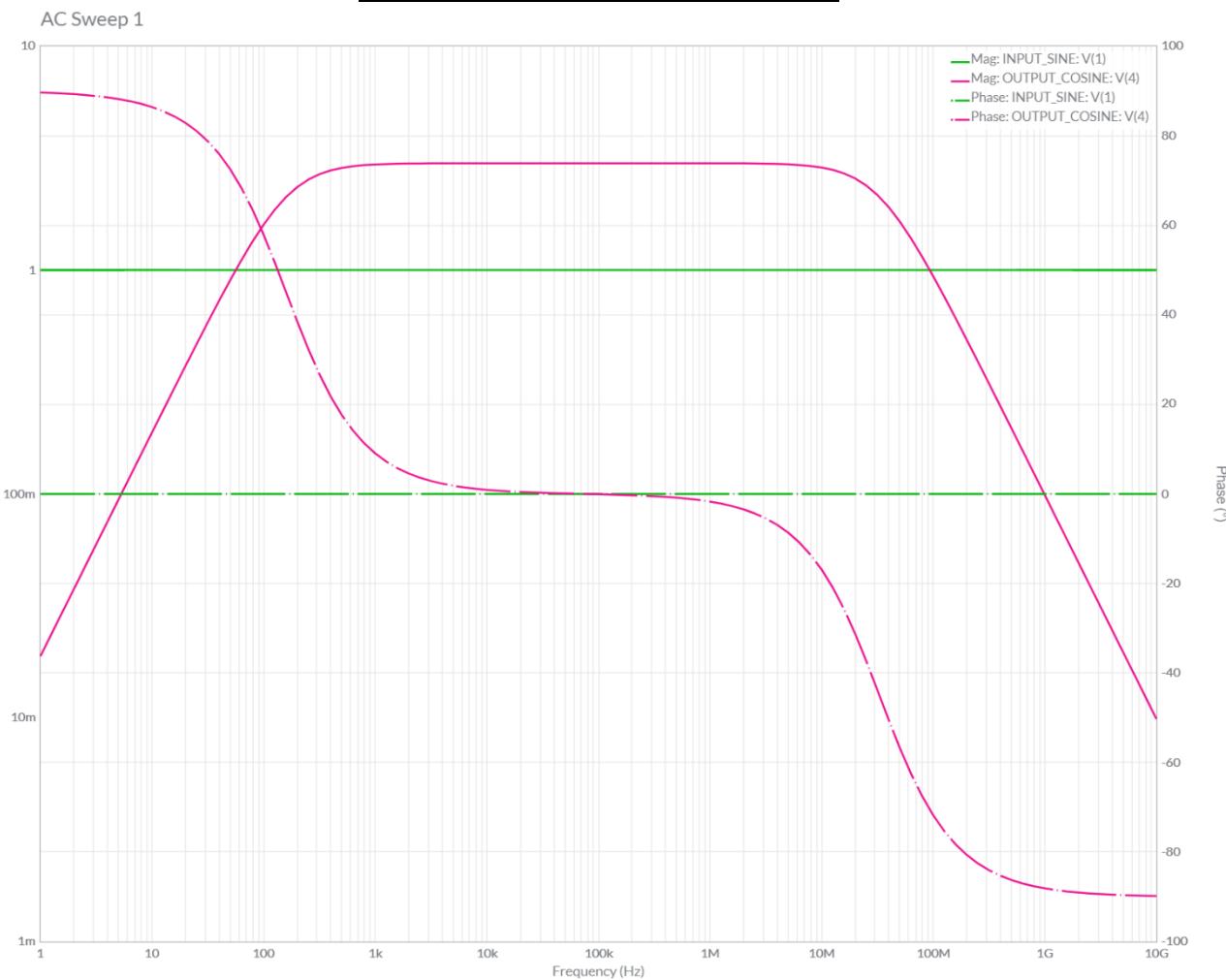
U19CS012

1. Simulate an Active High Pass Filter Circuit [High Pass Filter using OPAMP]  
[OPAMP with Amplification Factor = 3]

**Circuit Diagram [Multisim Implementation]**

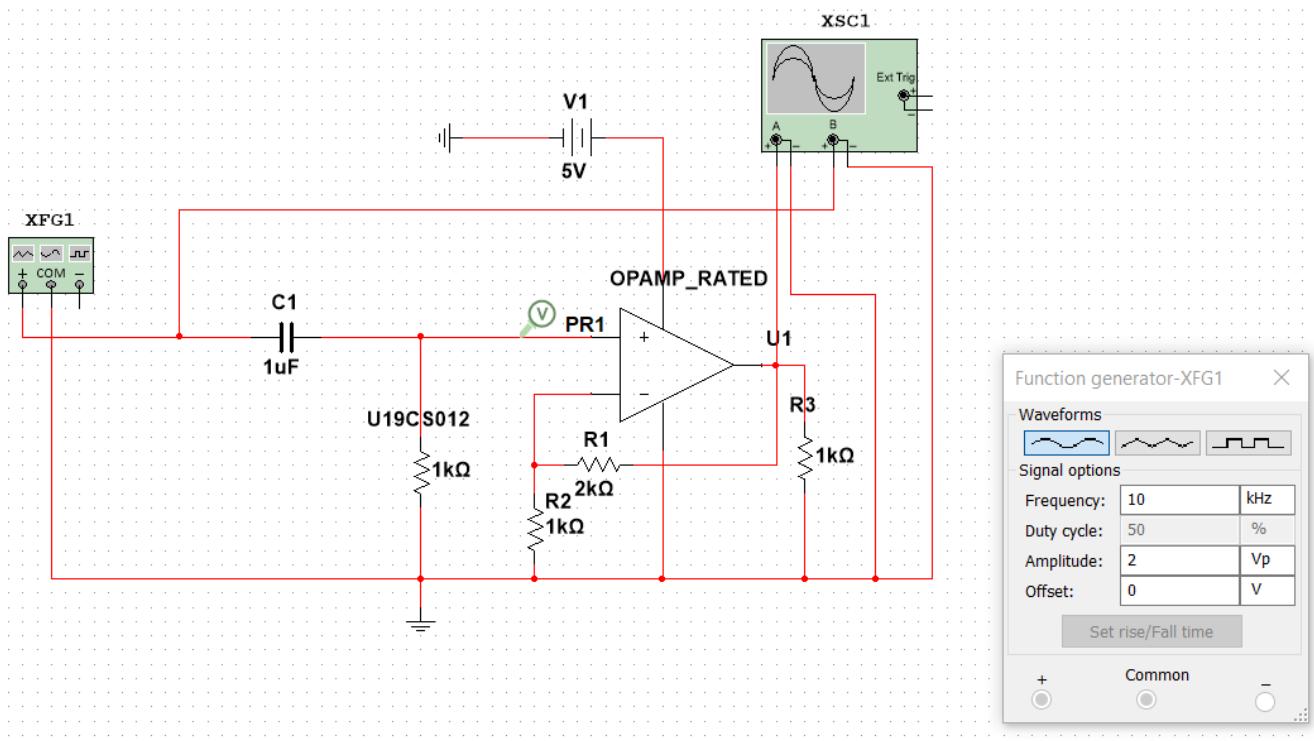


**Grapher Image [AC Sweep]**

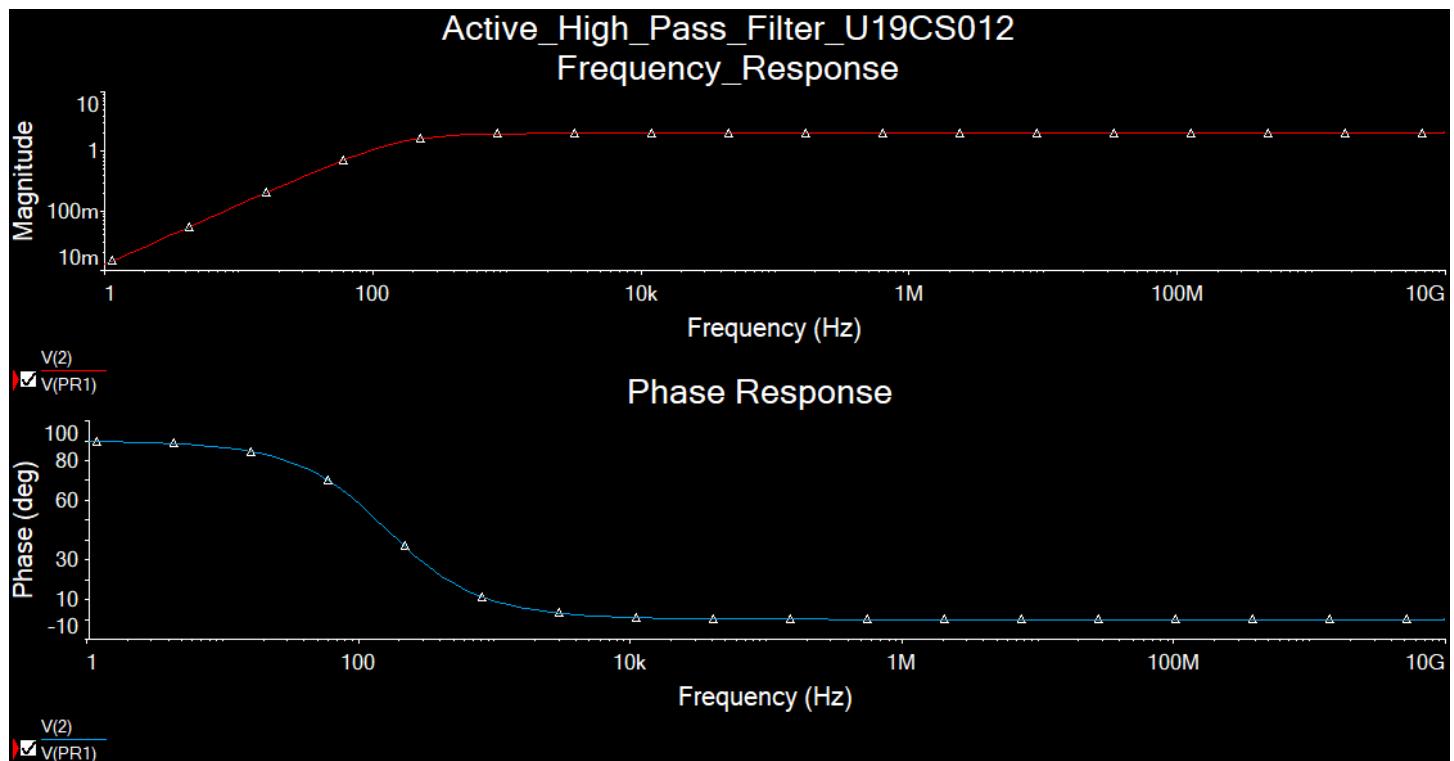




### Circuit Diagram [Offline Multisim Implementation]



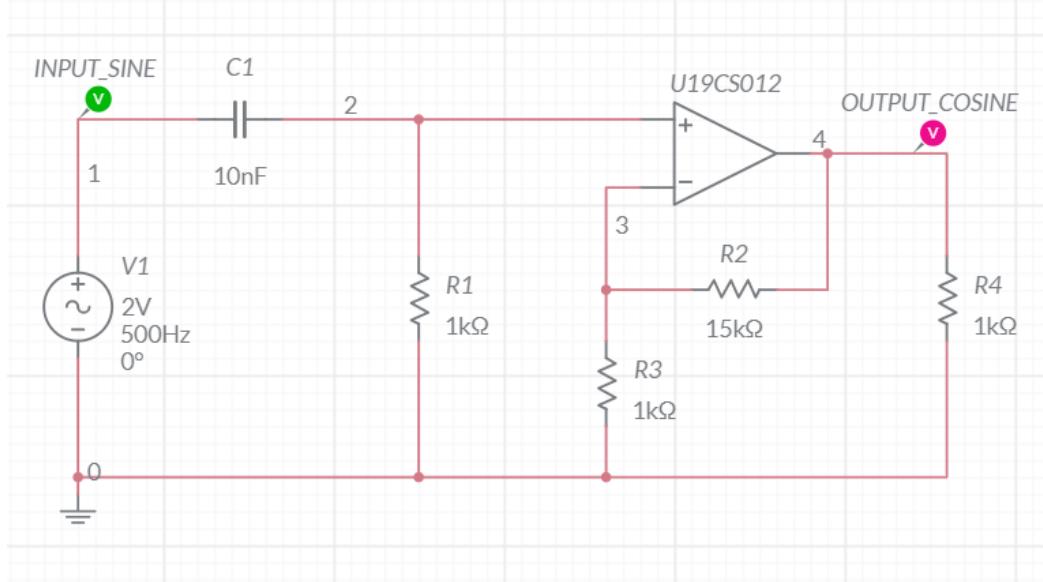
### Grapher Image [Offline Multisim AC Sweep]



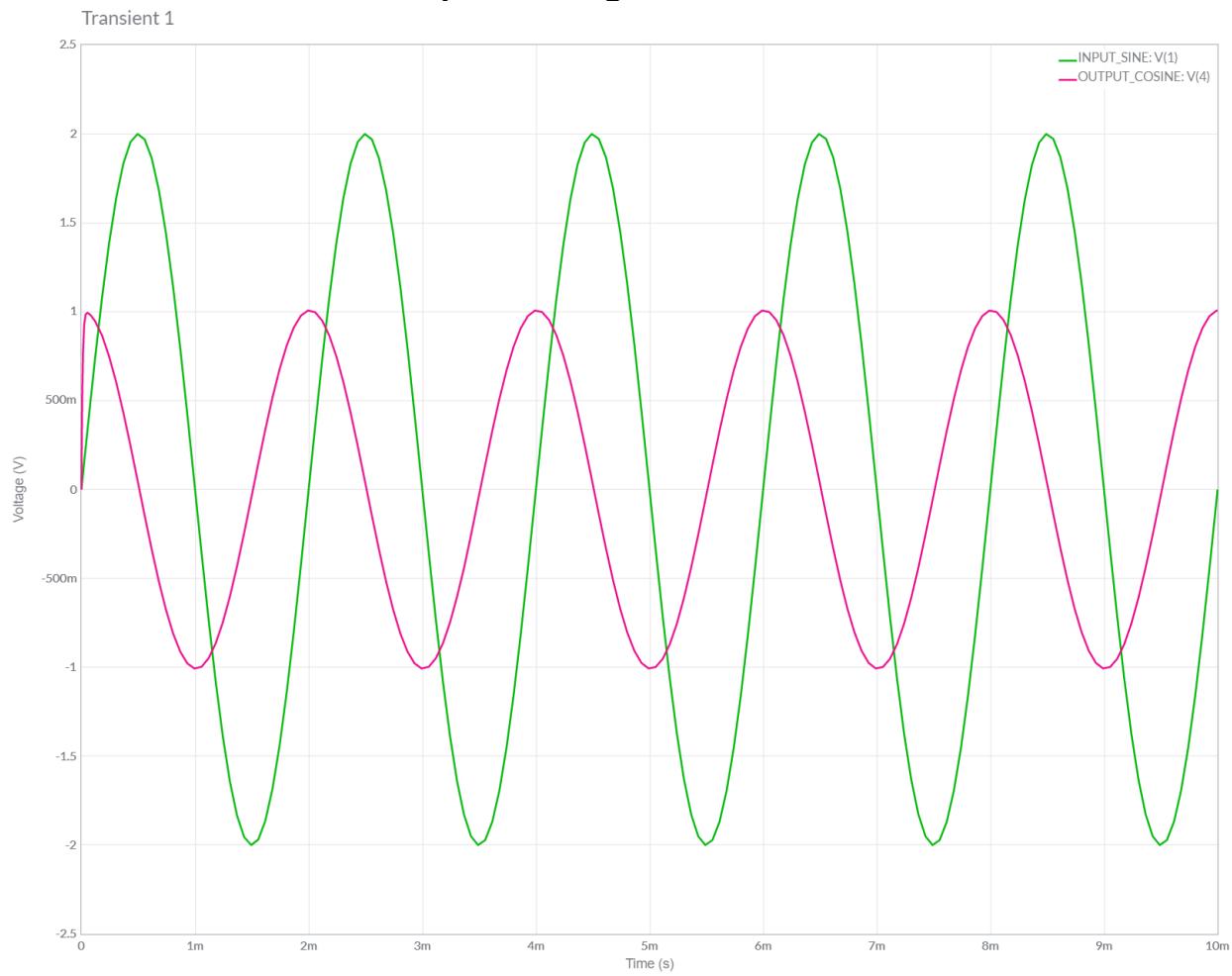


**2. Simulate an Active High Pass Filter Circuit as Differentiator  
 [Differentiation of (Sine Wave) = + Cosine Wave]**

**Circuit Diagram [Multisim Implementation]**



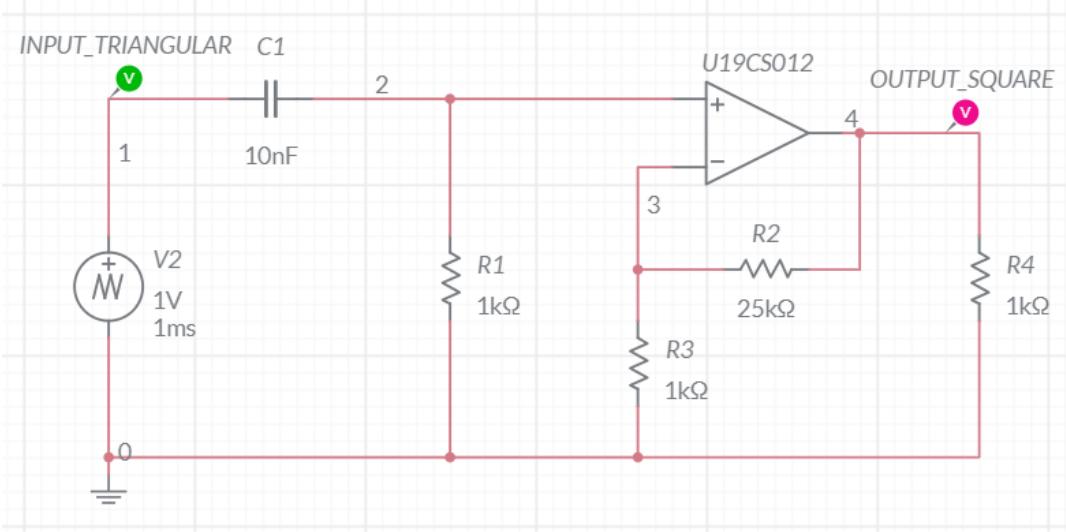
**Grapher Image [Transient]**





**3. Simulate an Active High Pass Filter Circuit as Differentiator**  
**[Differentiation of (Triangular Wave) = Square Wave]**

**Circuit Diagram [Multisim Implementation]**



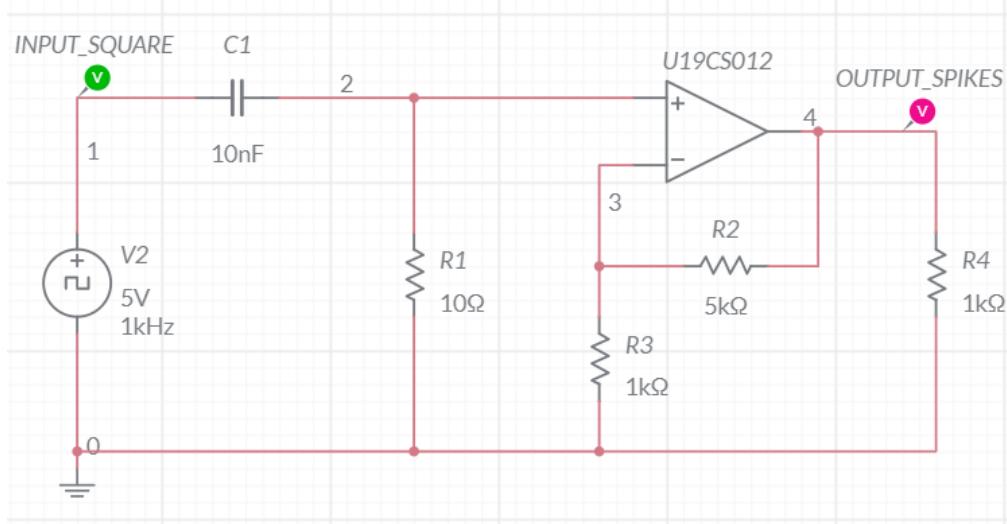
**Grapher Image [Transient]**



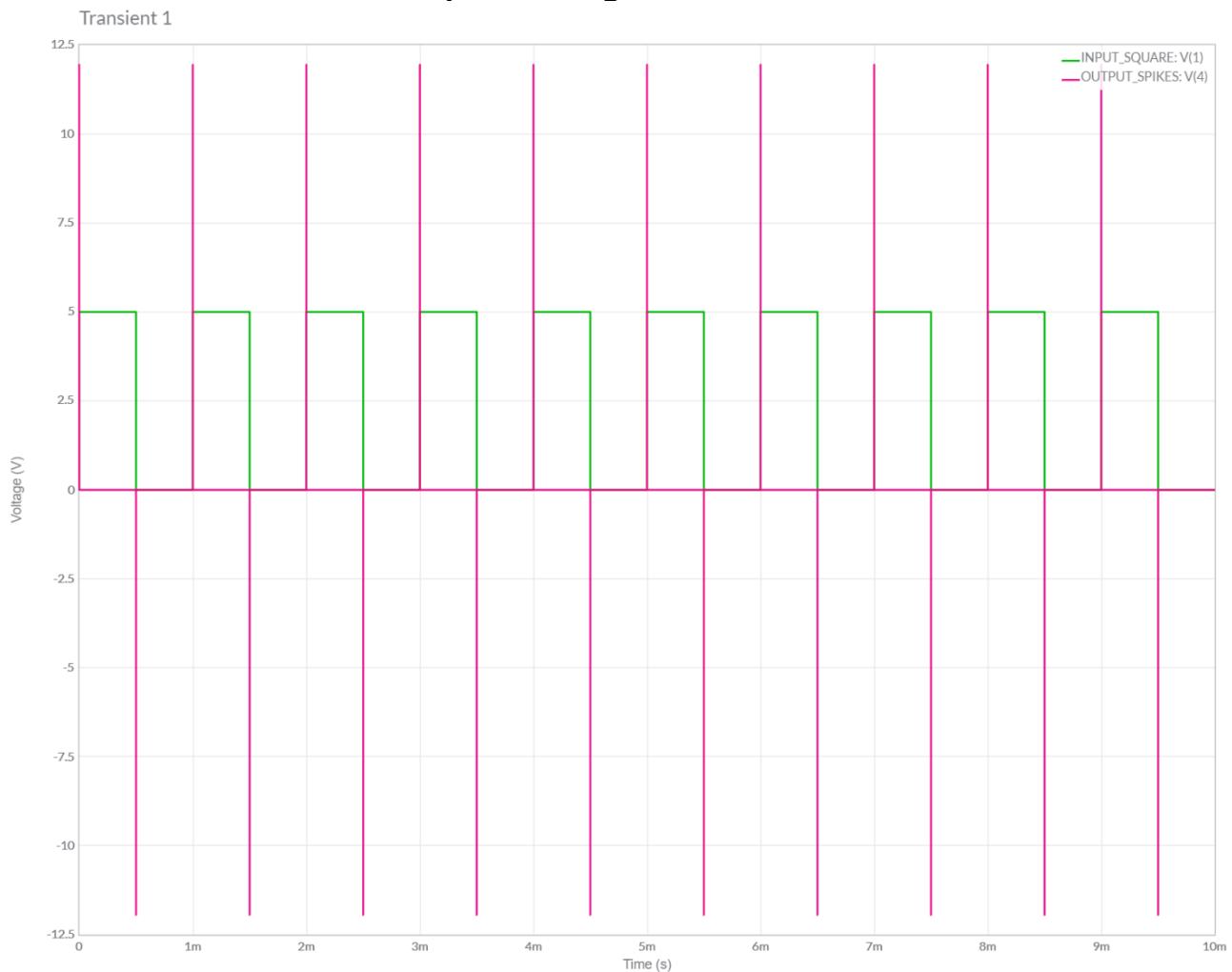


4. Simulate an Active High Pass Filter Circuit as Differentiator  
 [Differentiation of (Square Wave) = Spikes Wave]

**Circuit Diagram [Multisim Implementation]**



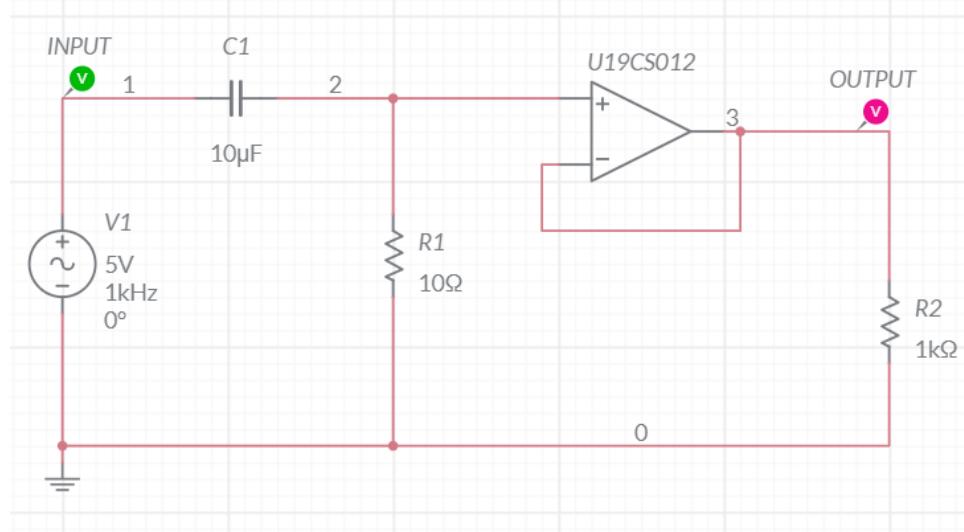
**Grapher Image [Transient]**





5. Simulate an Active Low Pass Filter Circuit [High Pass Filter using OPAMP]  
 [EXTRA] [OPAMP as Buffer]

**Circuit Diagram [Multisim Implementation]**



**Grapher Image [AC Sweep]**

