SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT
ELECTRONICS ENGINEERING DEPARTMENT

| Expt. No: | 2 | Study of Basic and Universal Gates |
|---|---|---|
| Date: | 20-08-2020 | |

**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 differnt 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.

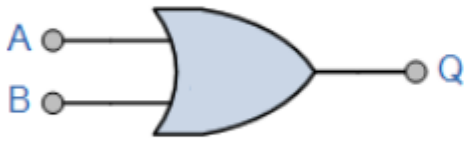
**AND GATE:** The Logic AND Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when all of its inputs are HIGH. The output state of a digital logic AND gate only

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 | | |
|---|---|---|---|
|  2-input AND Gate | A | B | Q |
| | 0 | 0 | 0 |
| | 0 | 1 | 0 |
| | 1 | 0 | 0 |
| | 1 | 1 | 1 |

## OR GATE:

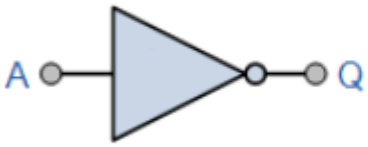
The Logic OR Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when one or more of its inputs are HIGH. The output, Q of a "Logic OR Gate" only returns "LOW" again when **ALL** of its inputs are at a logic level "0". In other words for a logic OR gate, any "HIGH"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

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

## NOT GATE:

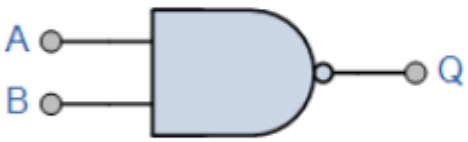
The Logic NOT Gate is the most basic of all the logical gates and is often referred to as an Inverting Buffer or simply an Inverter. Inverting NOT gates are single input devices which have an output level that is normally at logic level "1" and goes "LOW" to a logic level "0" when its single input is

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 |

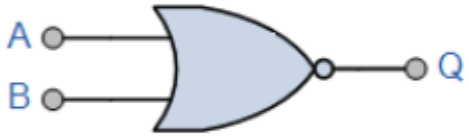
## NAND GATE:

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 |

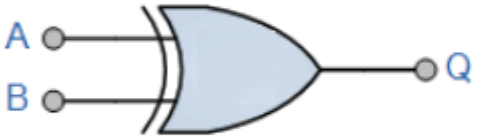
## NOR GATE:

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 | | |
|---|---|---|---|
|  2-input NOR Gate | A | B | Q |
| | 0 | 0 | 1 |
| | 0 | 1 | 0 |
| | 1 | 0 | 0 |
| | 1 | 1 | 0 |

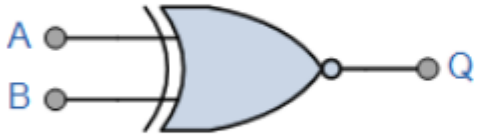
## EX-OR GATE:

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

| Symbol | Truth Table | | |
|---|---|---|---|
|  2-input Ex-OR Gate | A | B | Q |
| | 0 | 0 | 0 |
| | 0 | 1 | 1 |
| | 1 | 0 | 1 |
| | 1 | 1 | 0 |

## EX-NOR GATE:

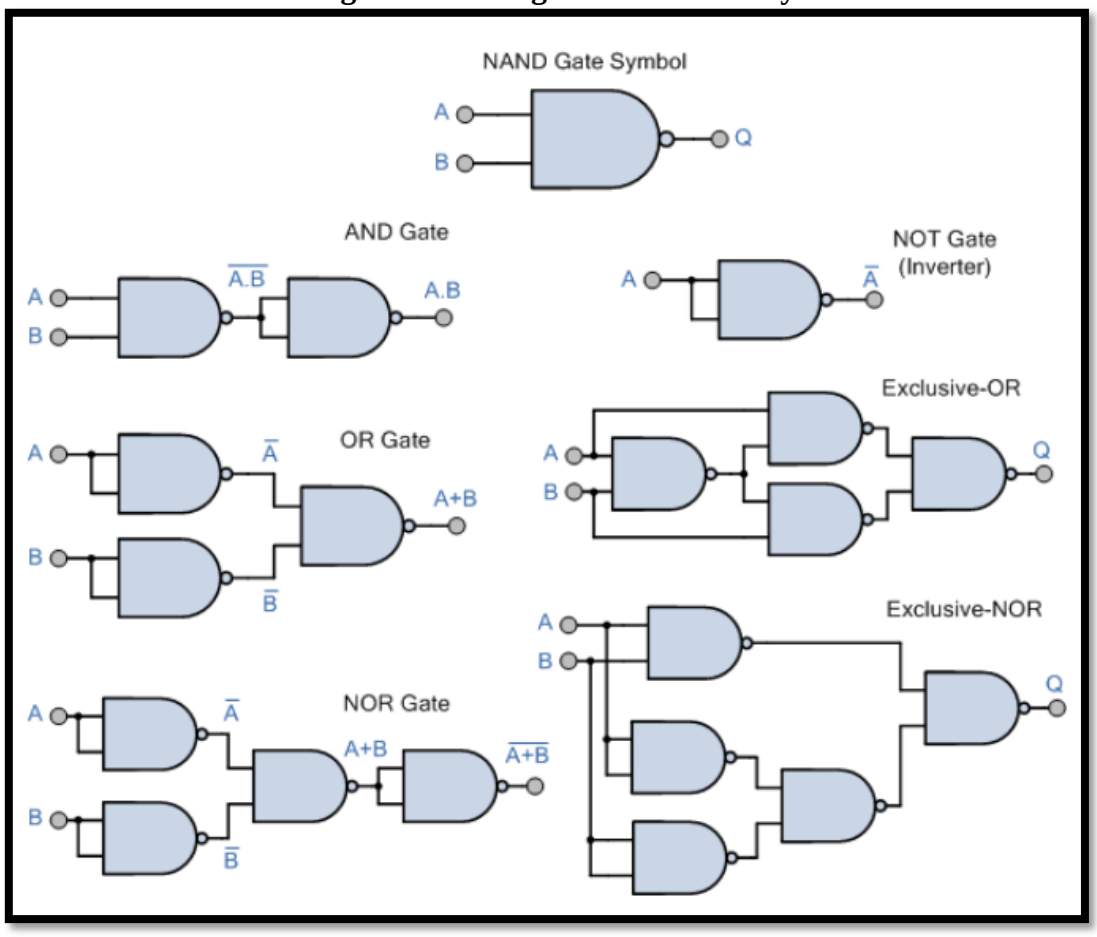
The Exclusive-NOR Gate function is a digital logic gate that is the reverse or complementary form of the Exclusive-OR function. This gate gives output "1" when its inputs are "*logically equal*" or "*equivalent*" to each other, which is why an **Exclusive-NOR** gate is sometimes called an **Equivalence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

| Symbol | Truth Table | | |
|---|---|---|---|
|  2-input Ex-NOR Gate | A | B | Q |
| | 0 | 0 | 1 |
| | 0 | 1 | 0 |
| | 1 | 0 | 0 |
| | 1 | 1 | 1 |

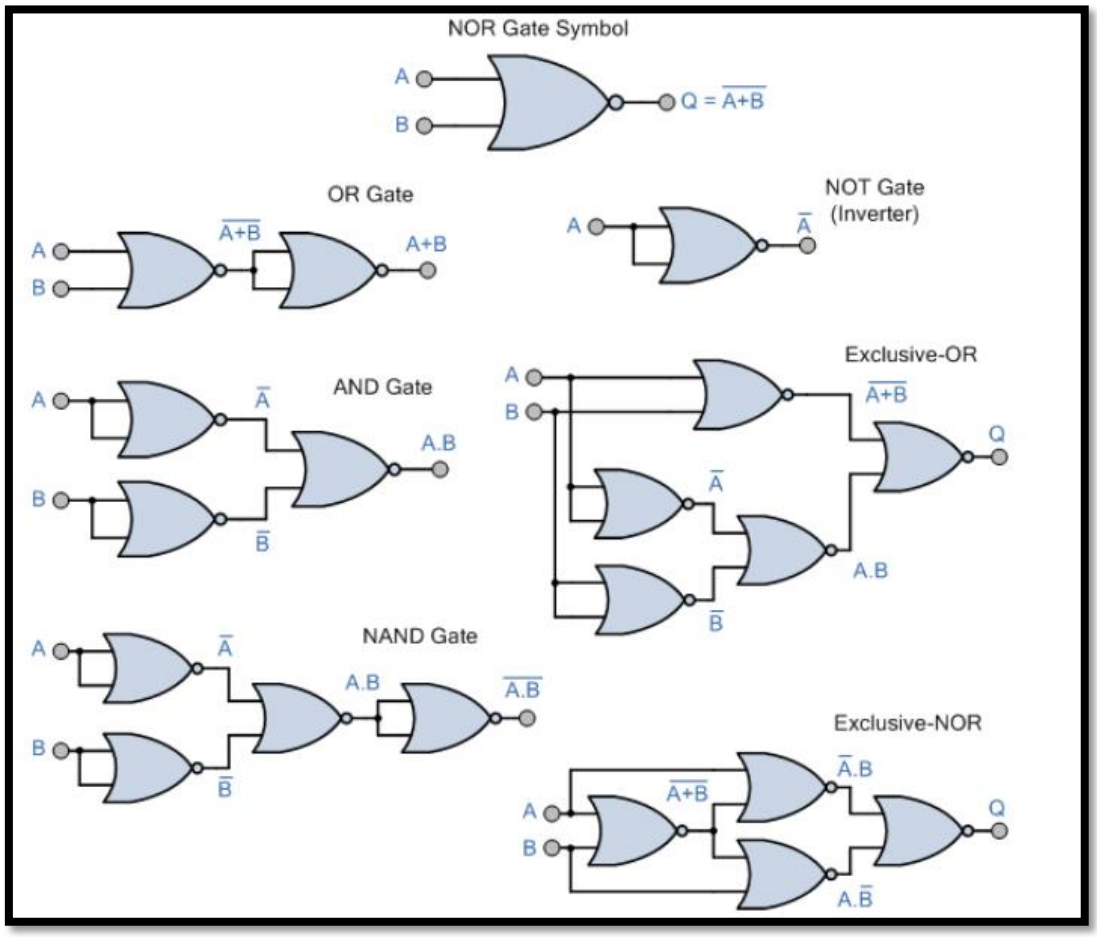
## UNIVERSAL GATES:

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:**

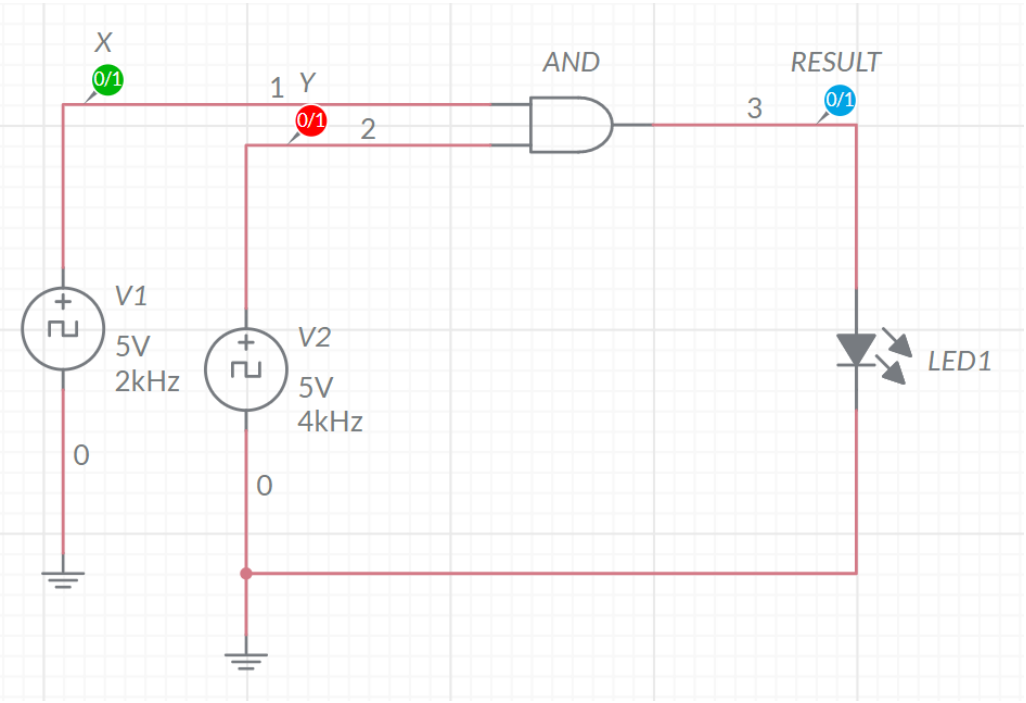


## Logic Gates using NOR Gate Only:

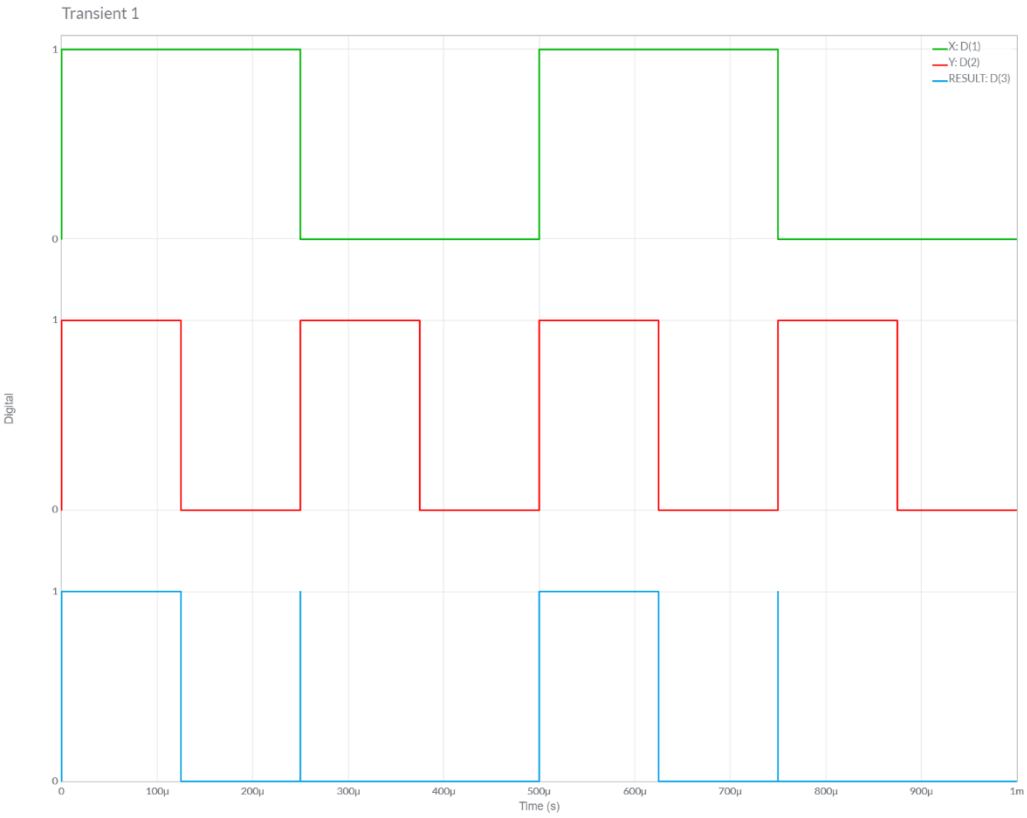


## 1.) 2-INPUT AND

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

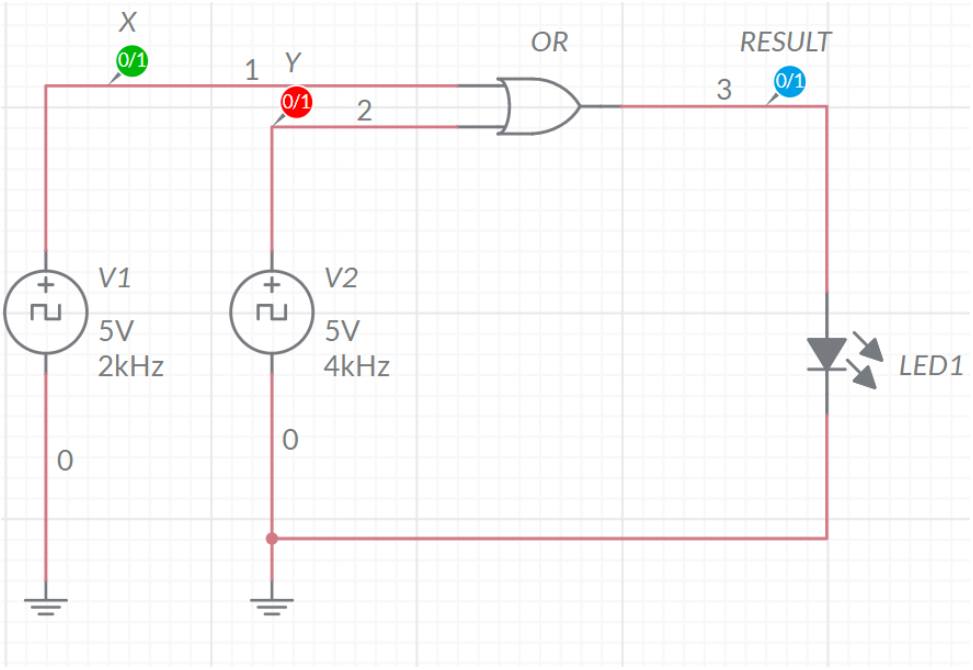



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

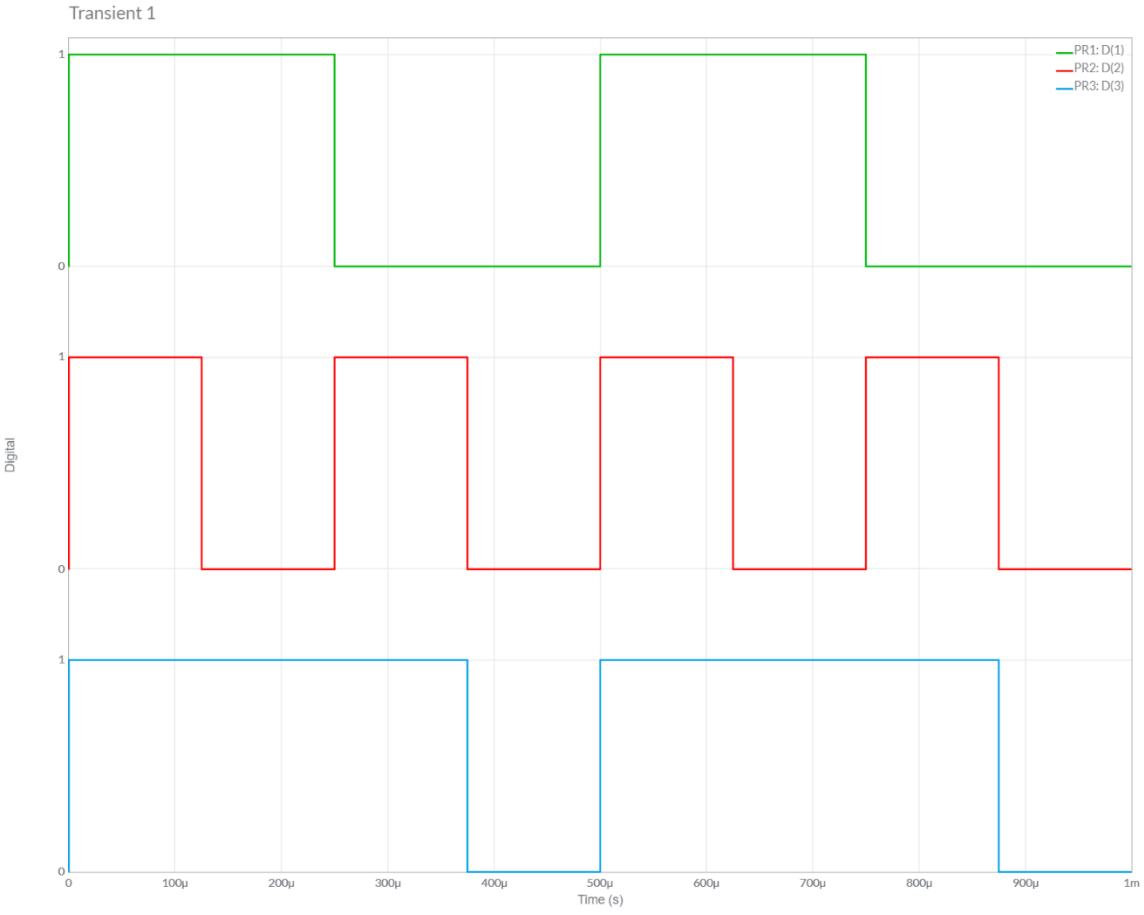


## 2.) 2-INPUT OR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

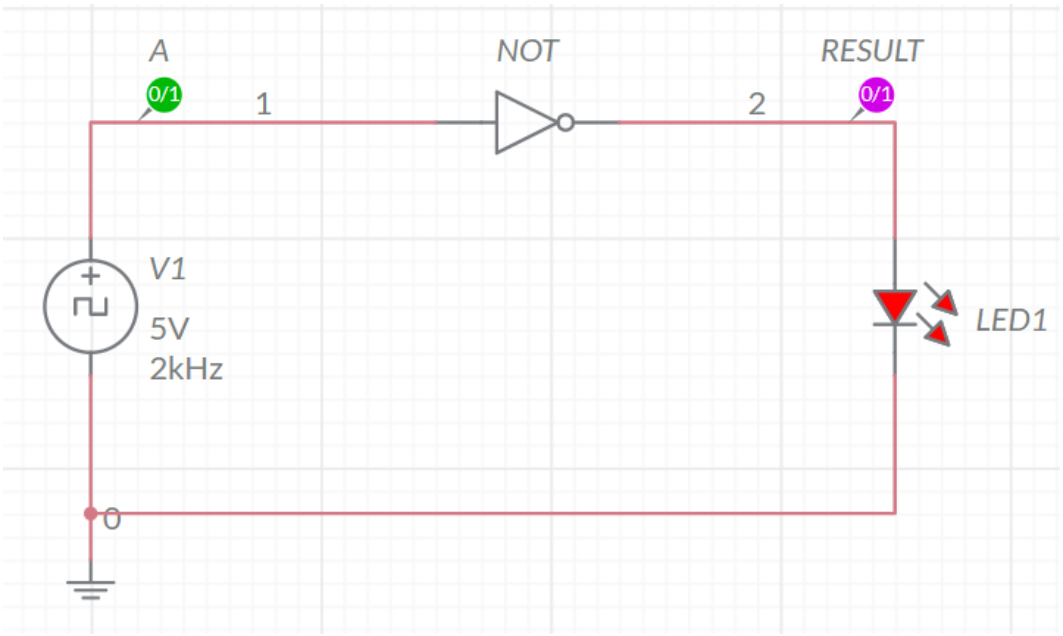



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

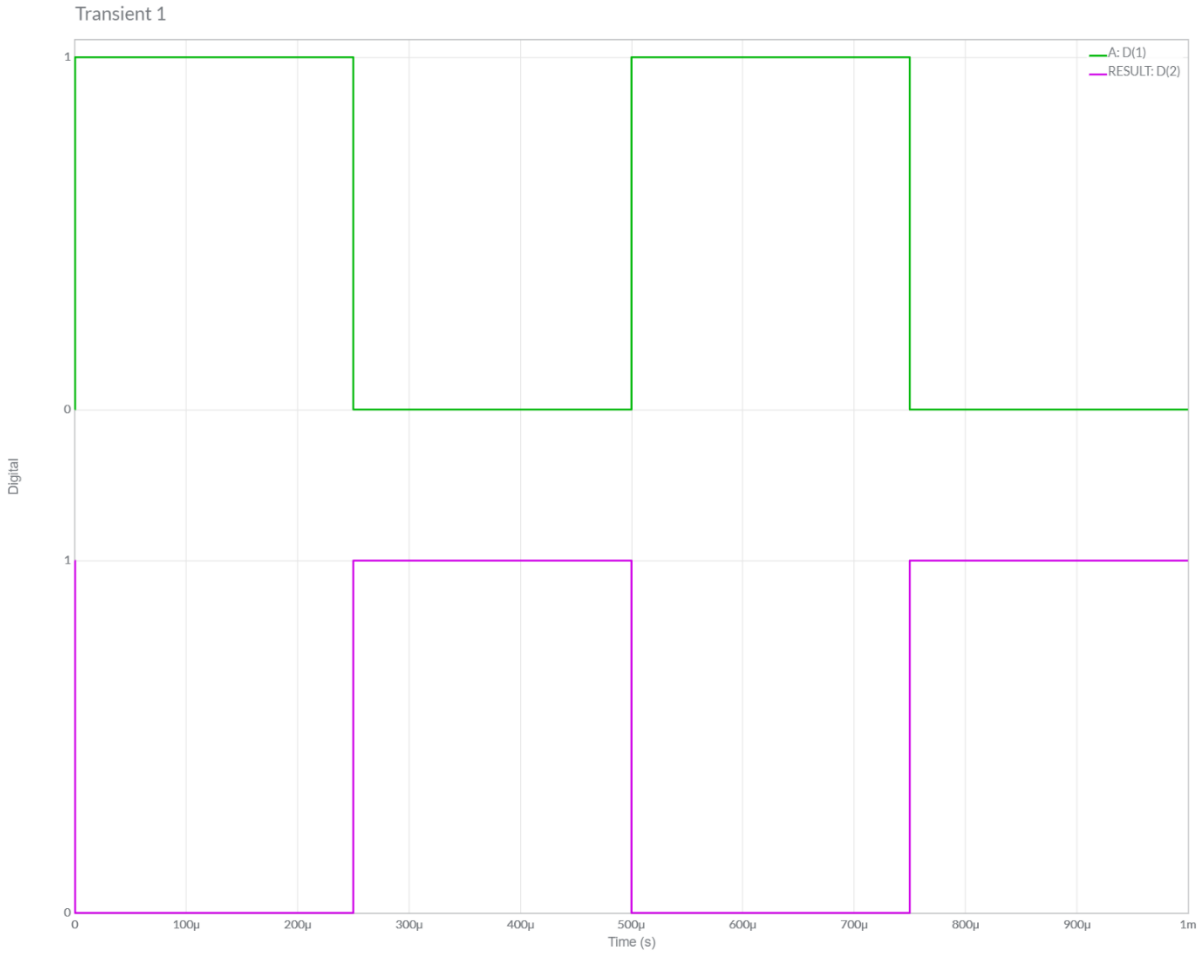


## 3.) NOT

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

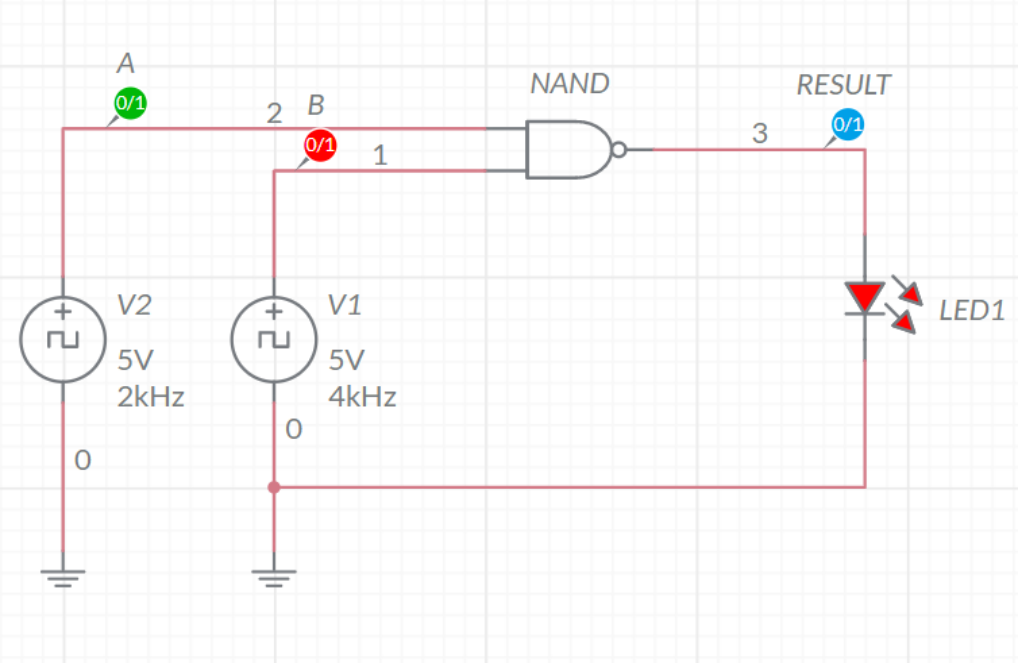



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

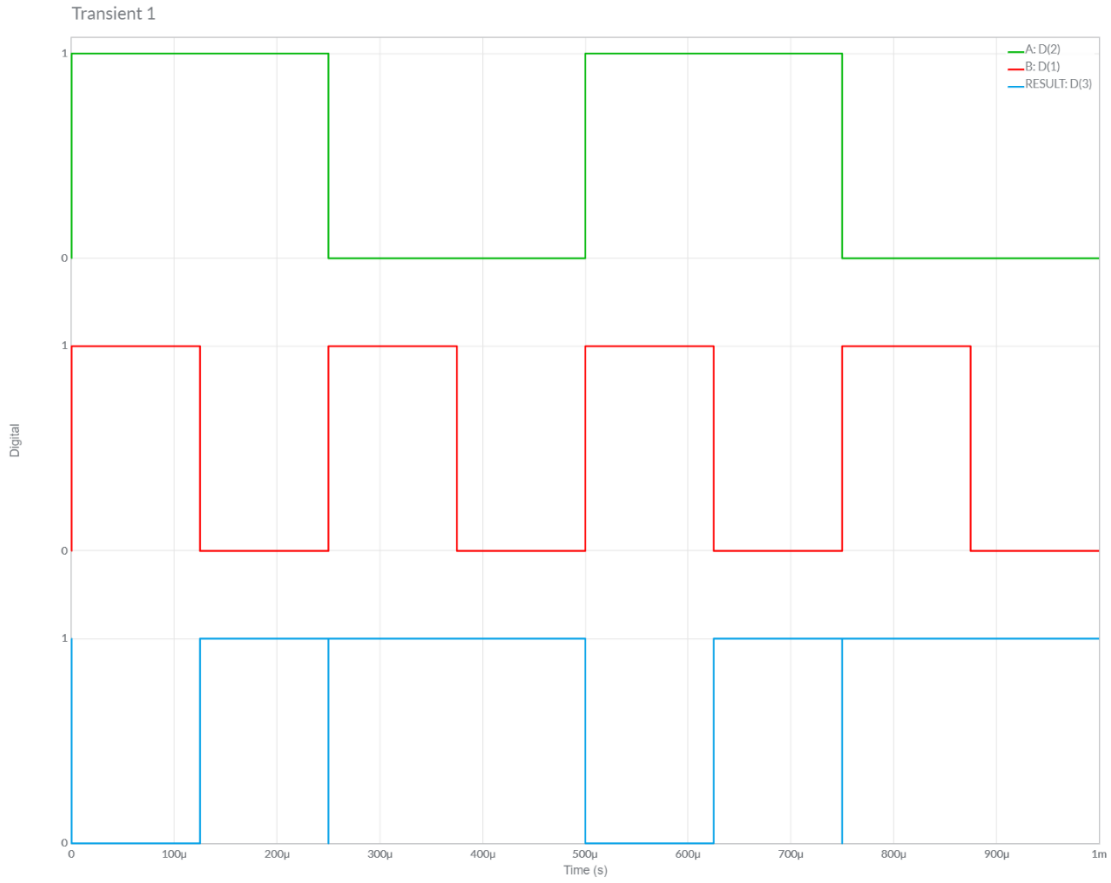


## 4.) 2-INPUT NAND

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

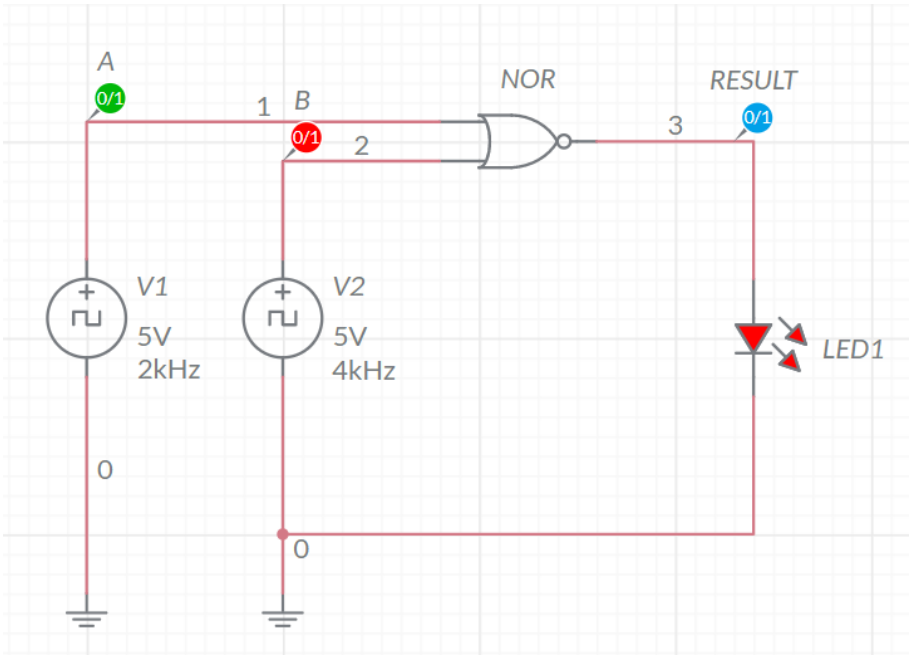



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

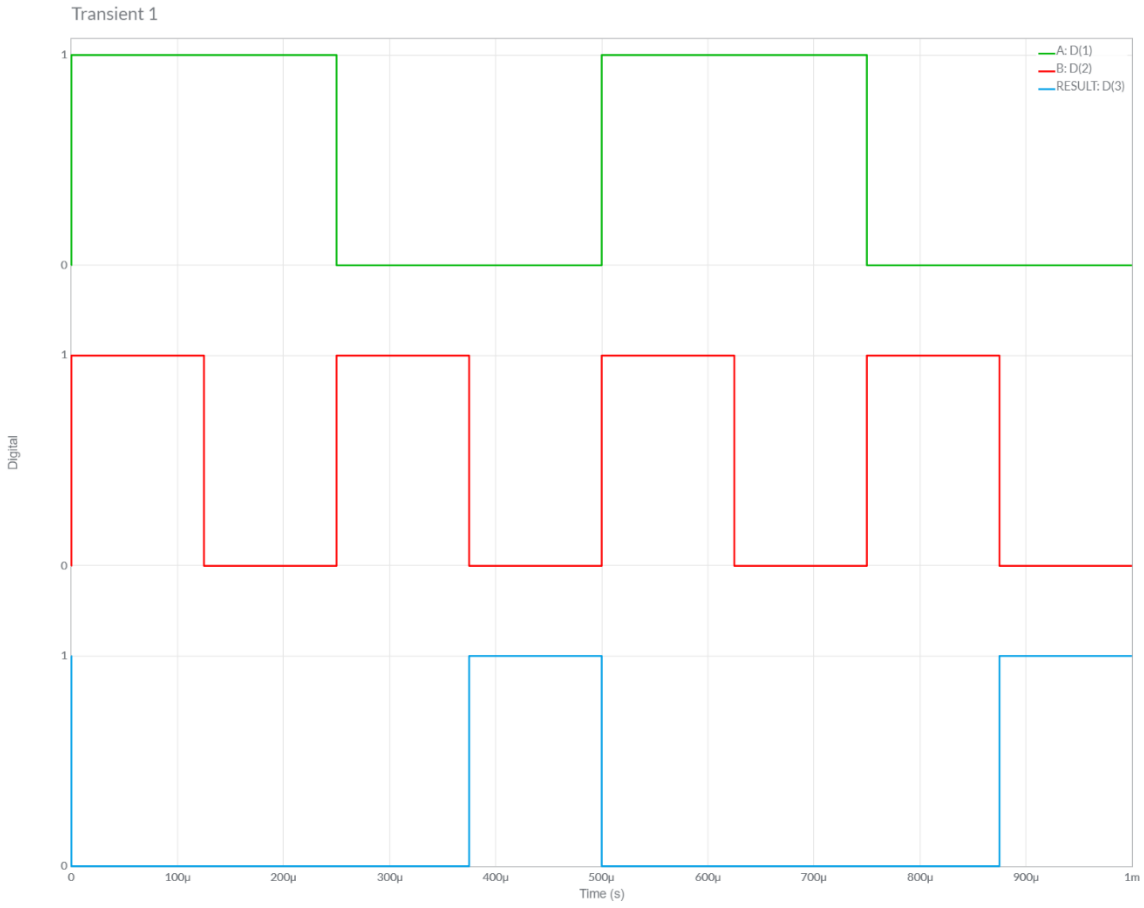


## 5.) 2-INPUT NOR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

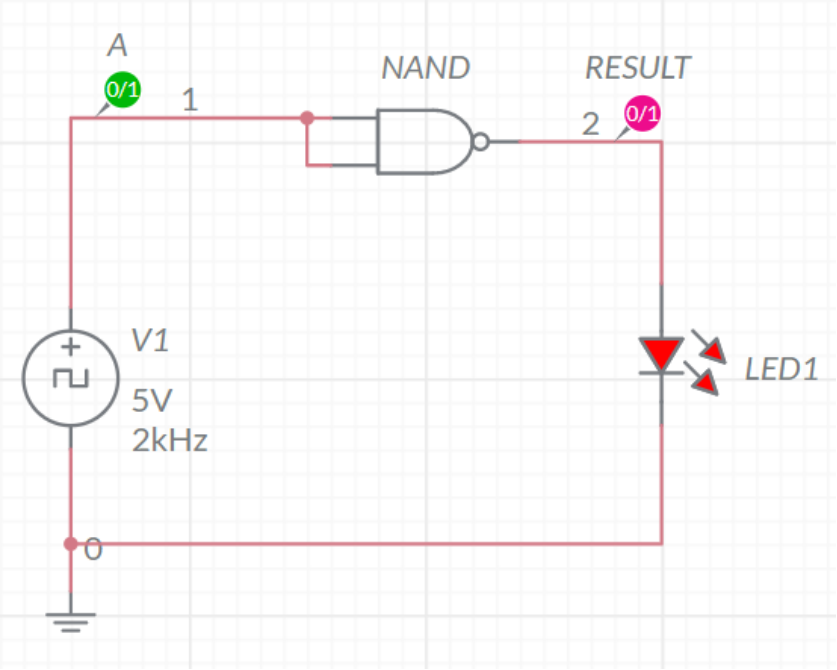



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

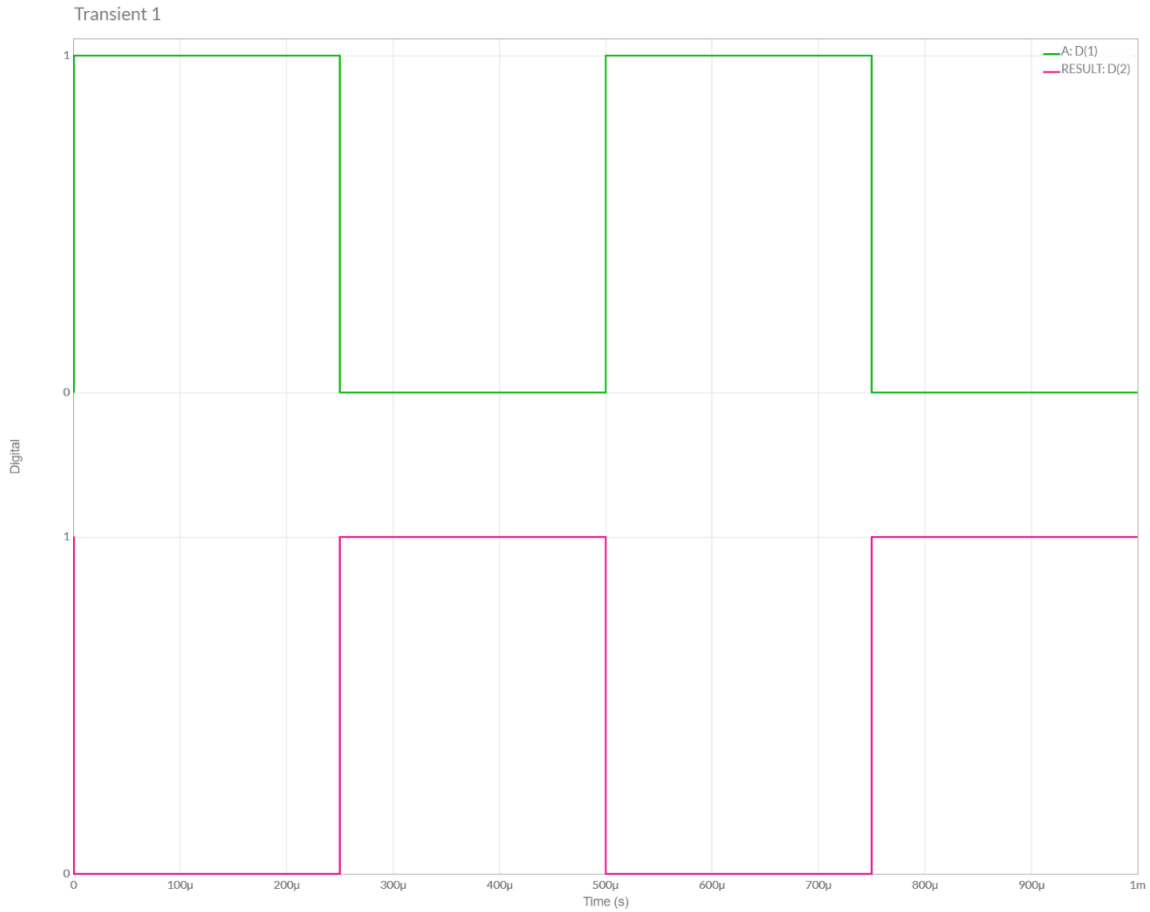


## 6.) CREATE INVERTOR USING NAND

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

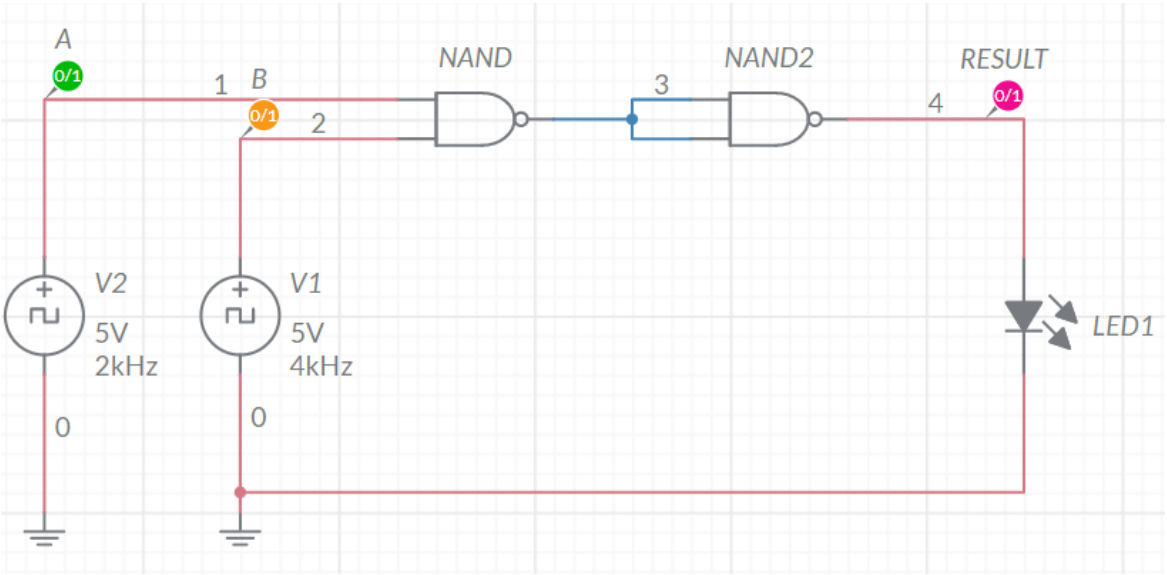



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

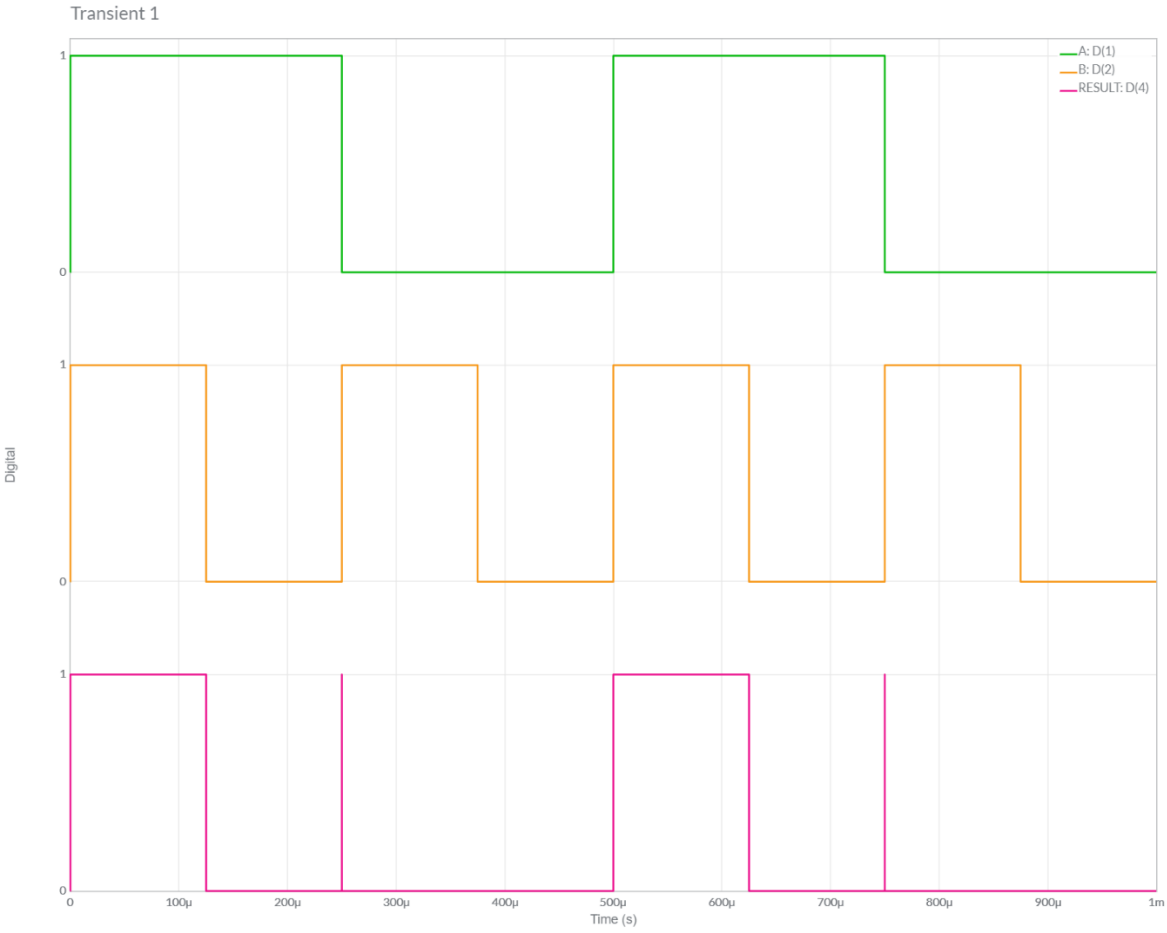


## 7.) CREATE AND USING NAND

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

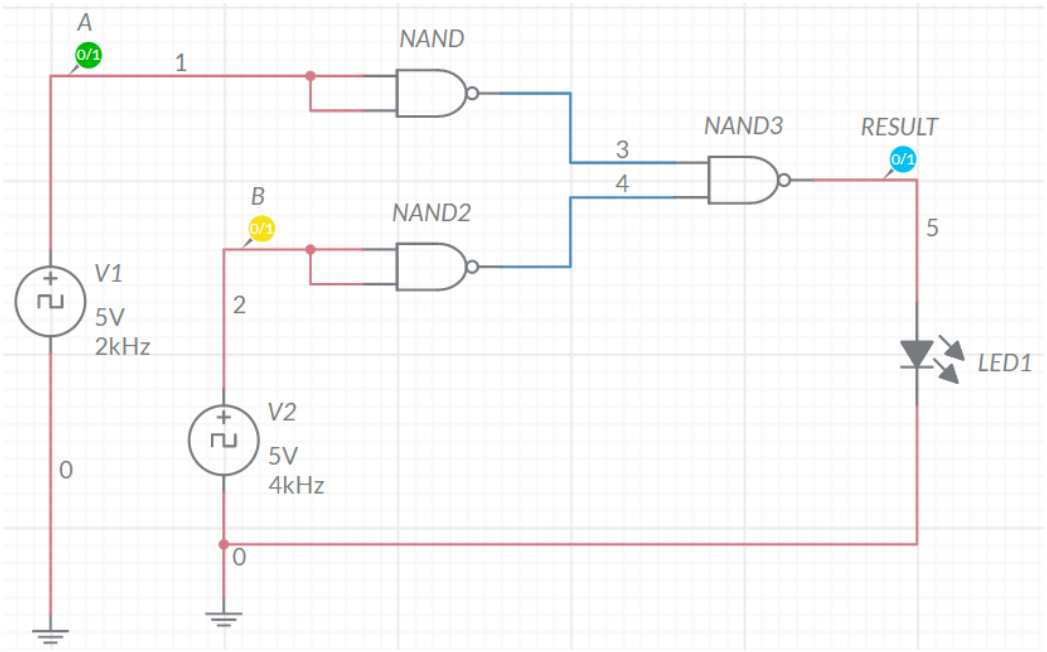



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

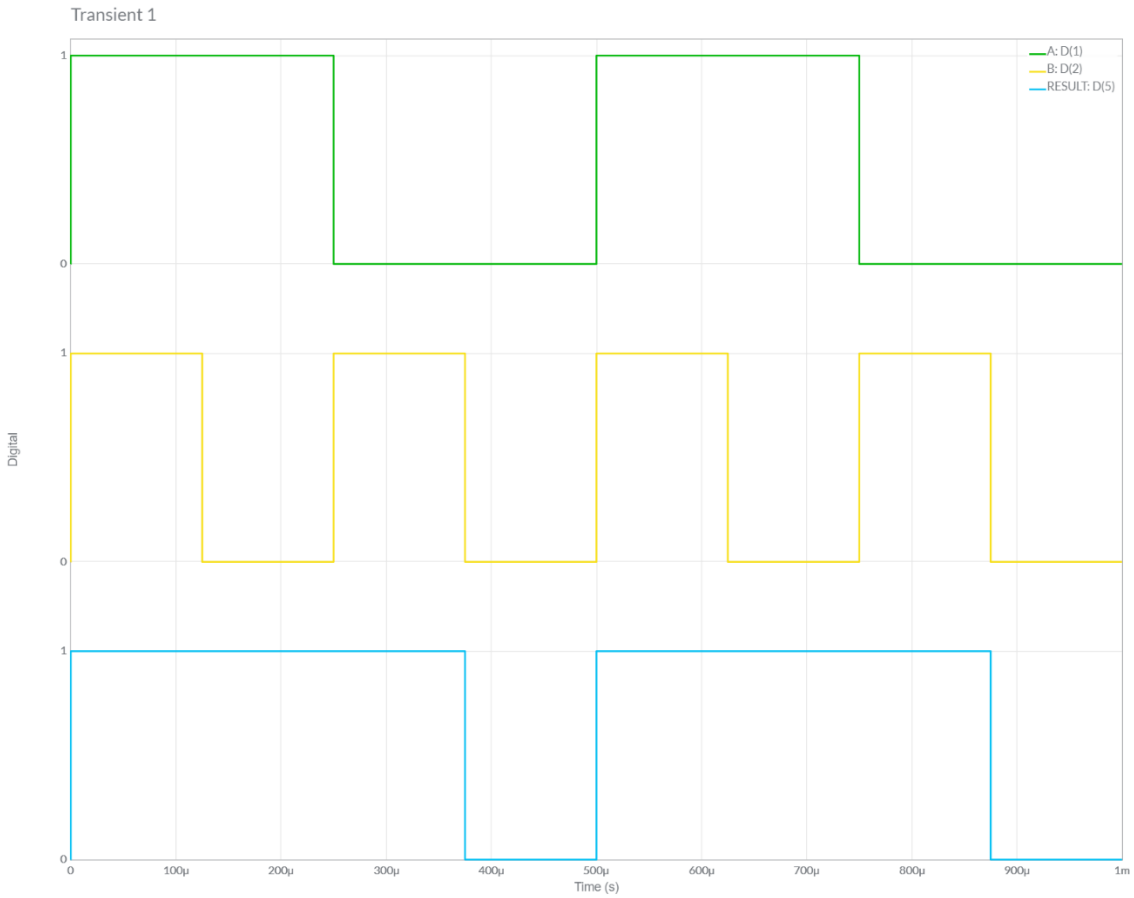


## 8.) CREATE OR USING NAND

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

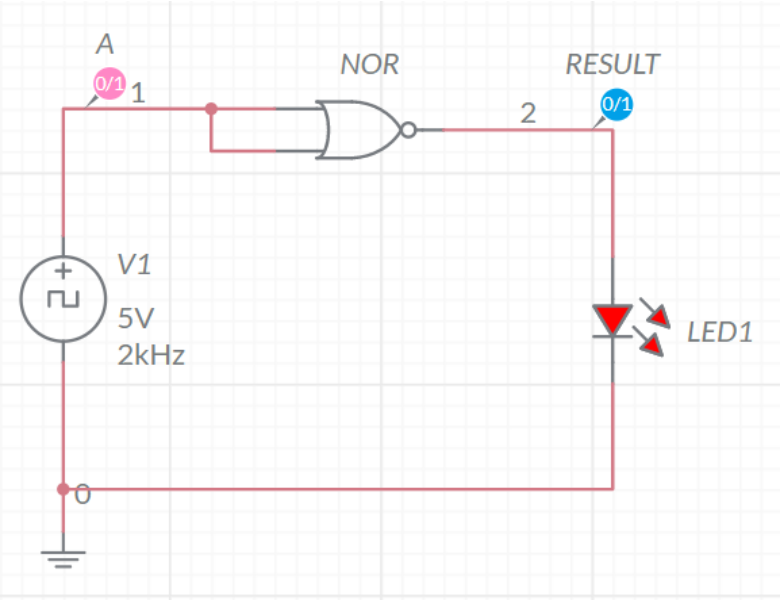



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

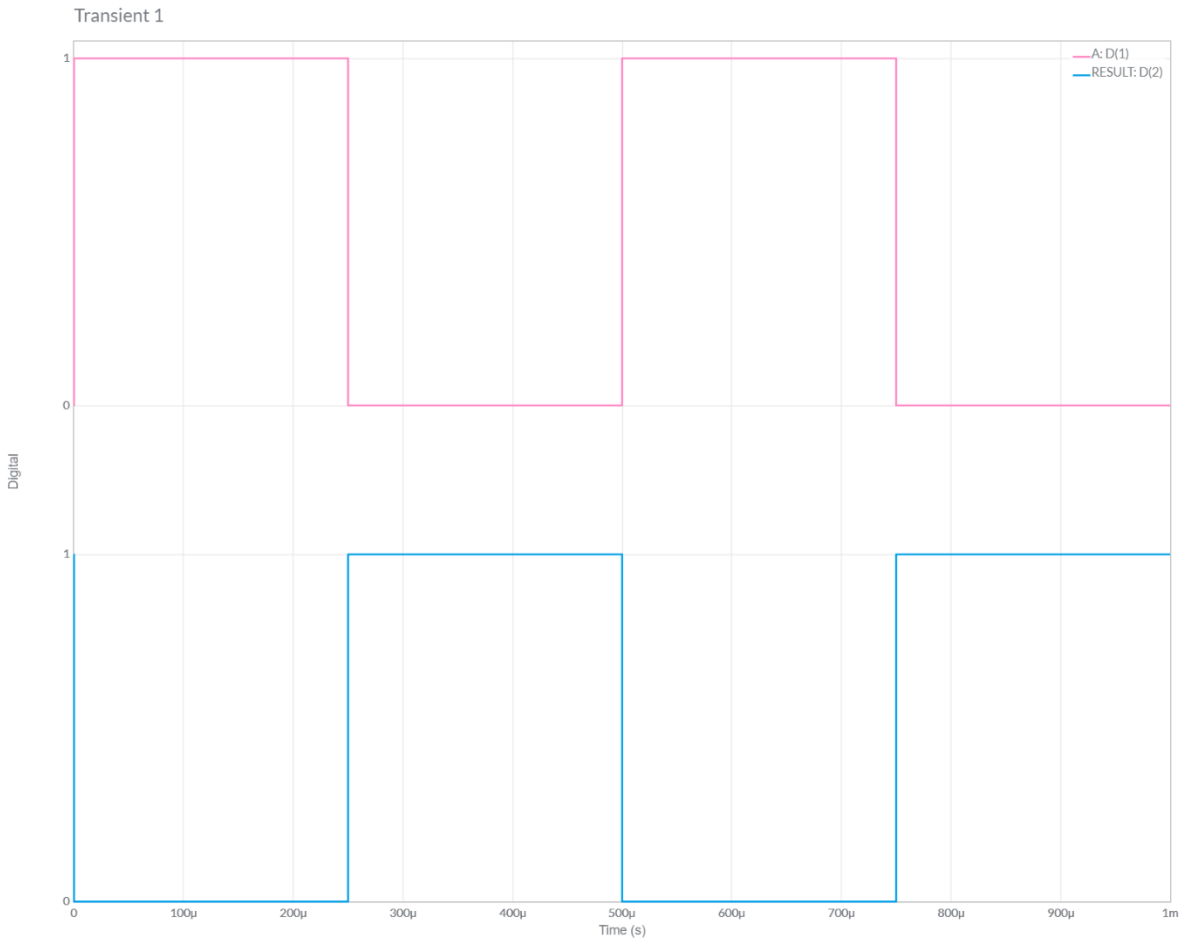


## 9.) CREATE INVERTOR USING NOR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

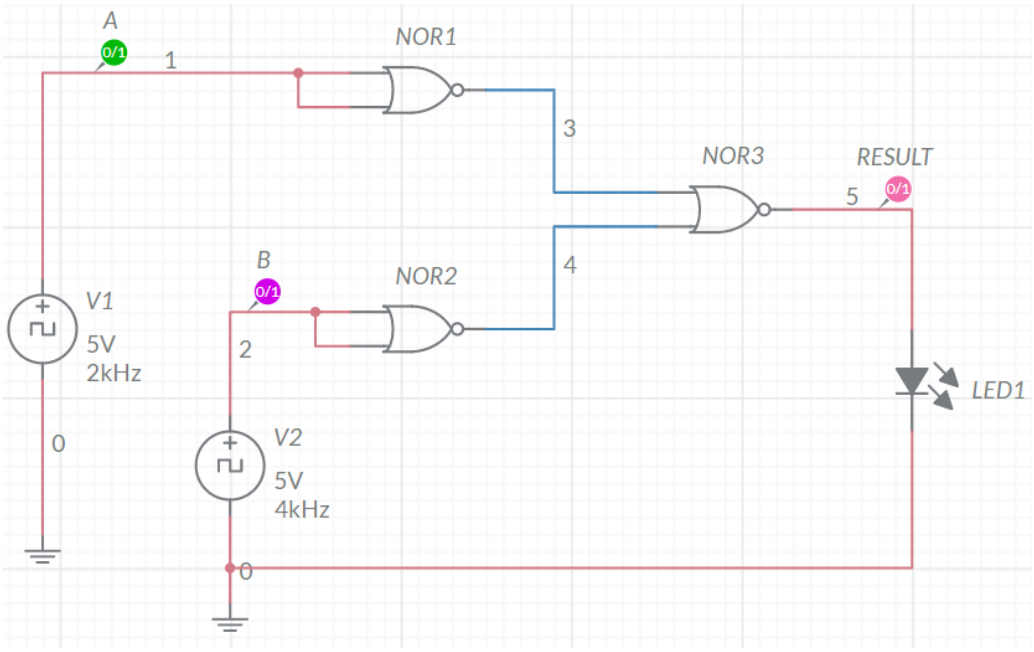



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

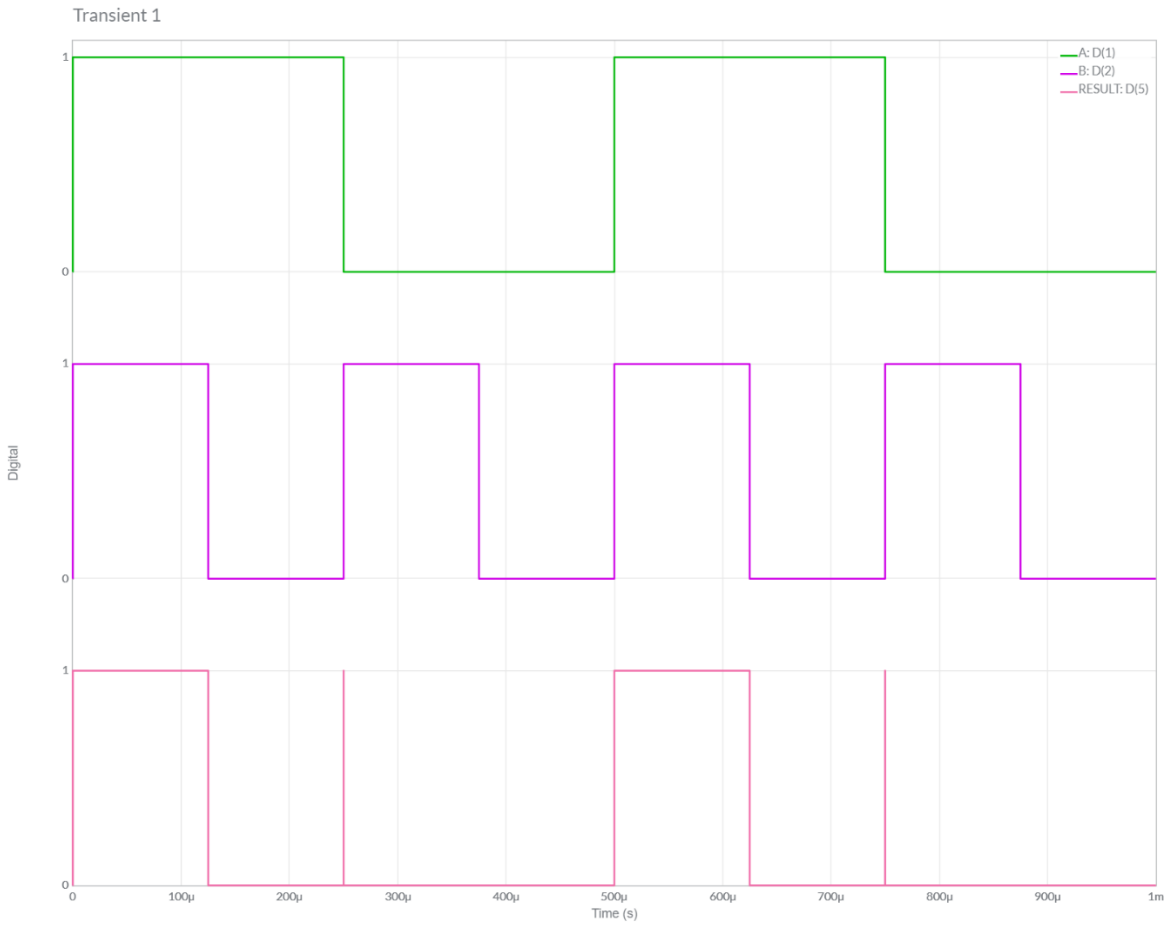


## 10.) CREATE AND USING NOR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

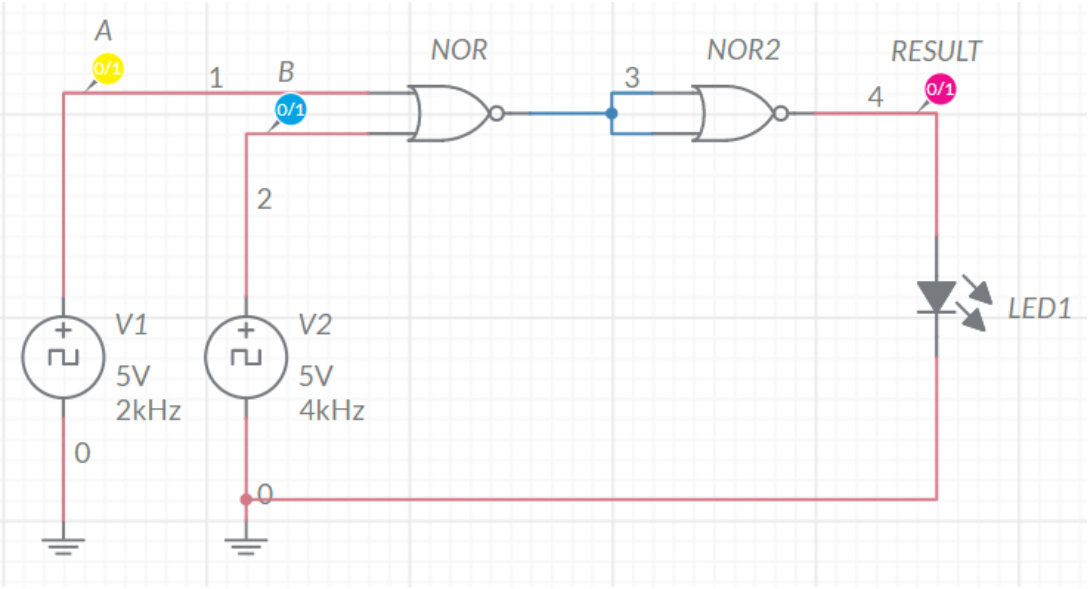



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

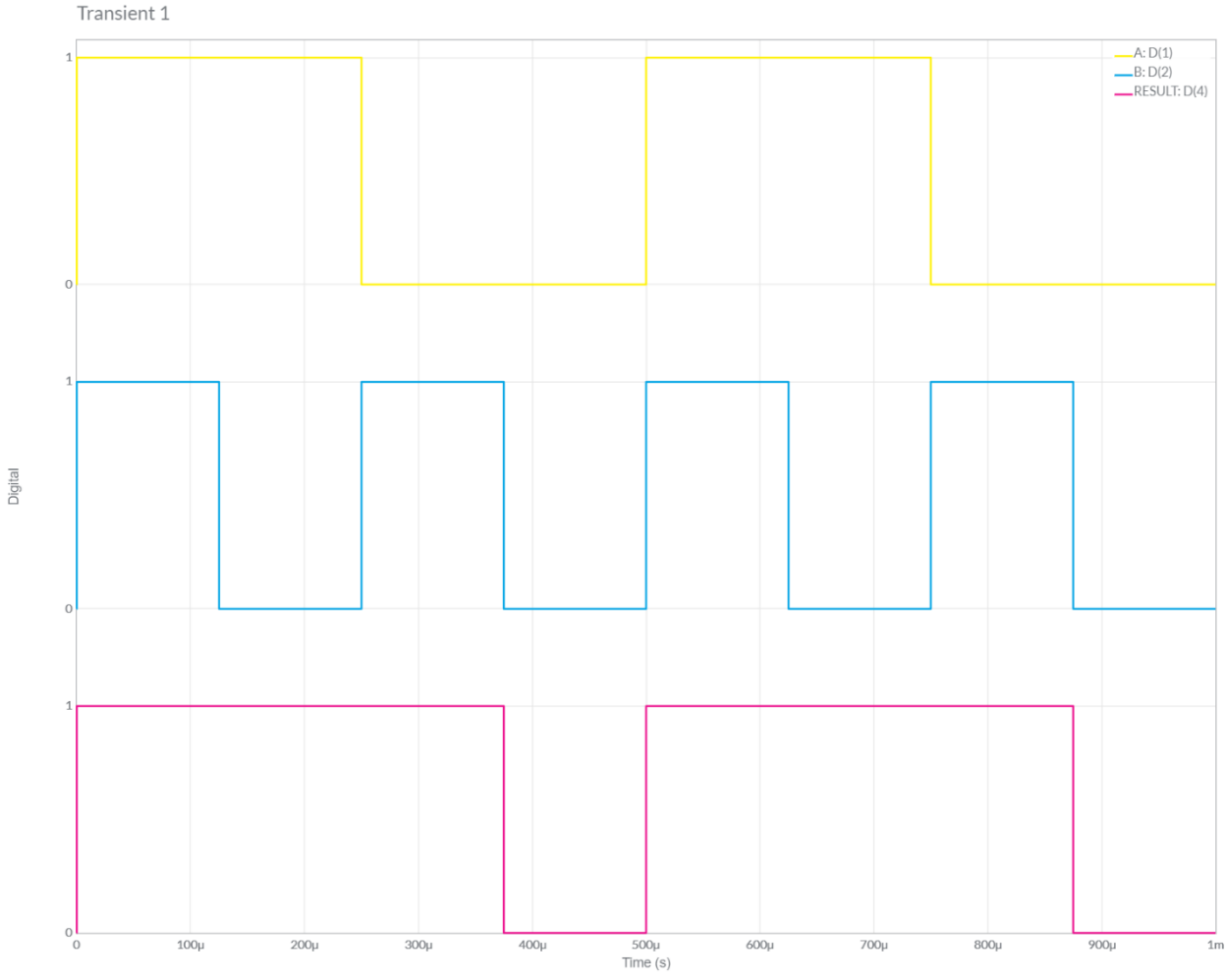


## 11.) CREATE OR USING NOR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

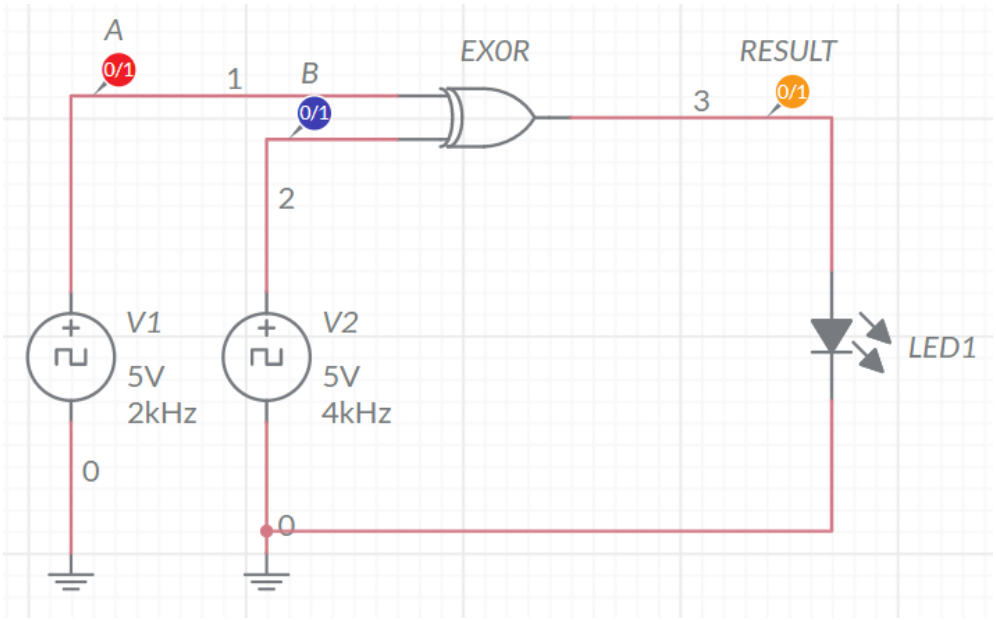



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

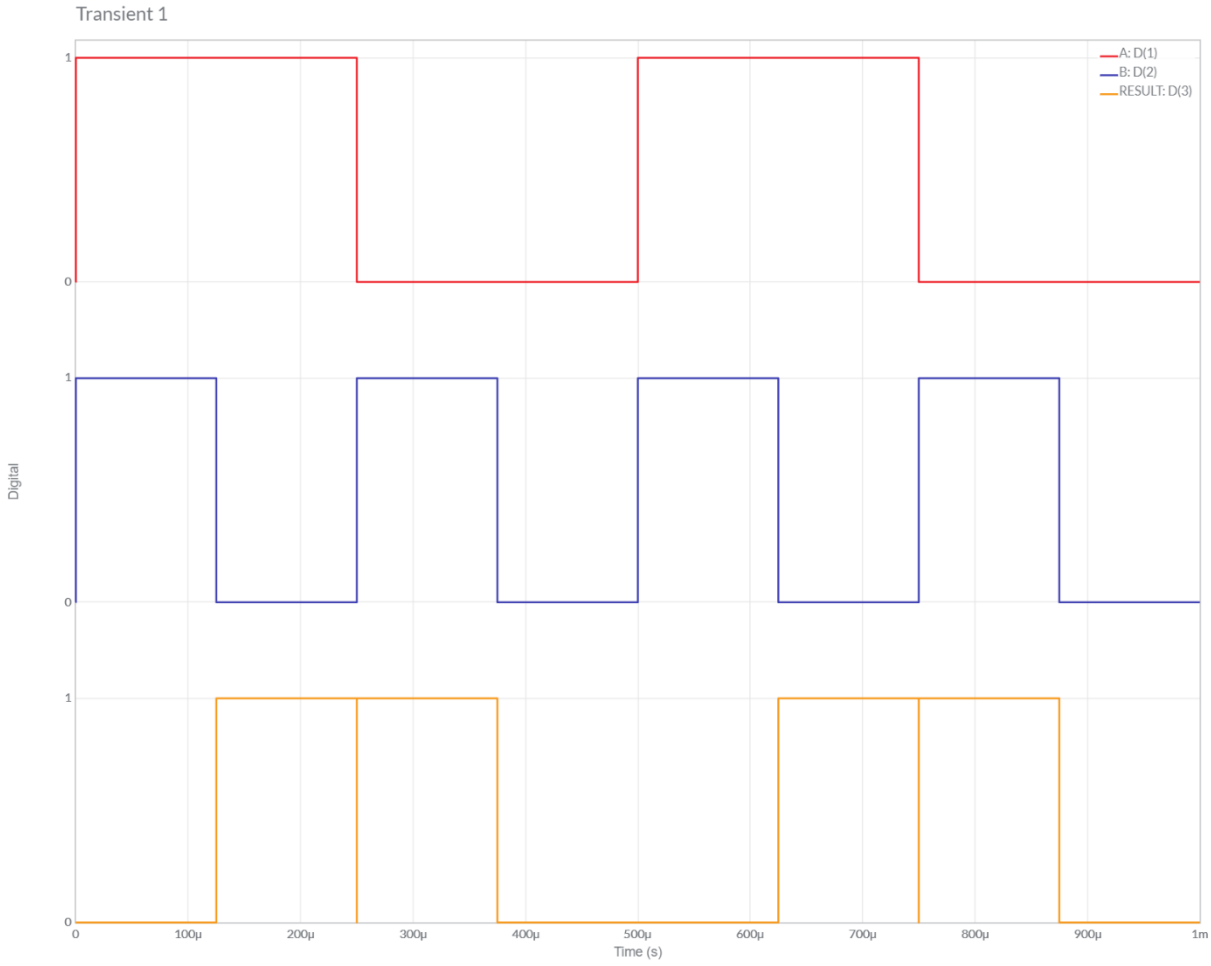


## 12.) 2-INPUT EX-OR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

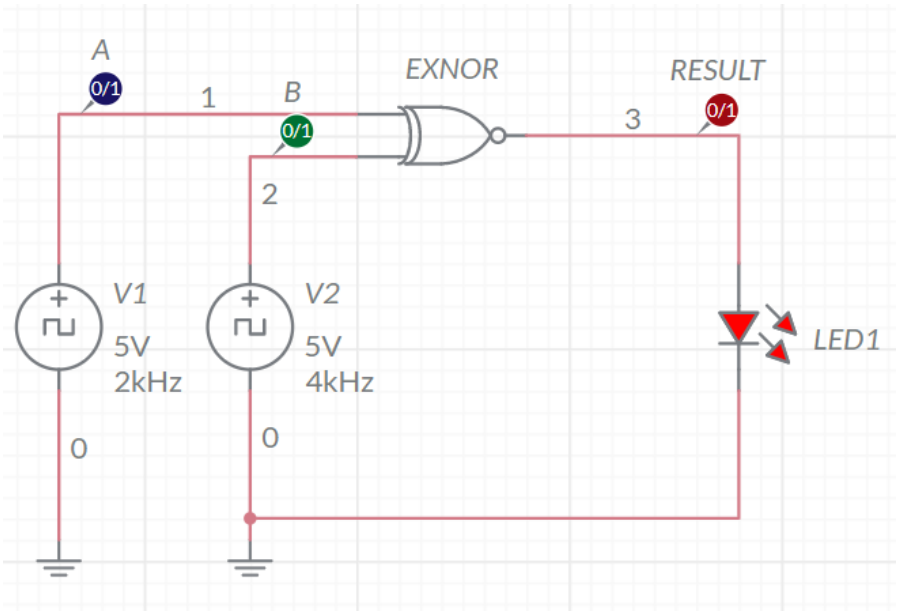



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

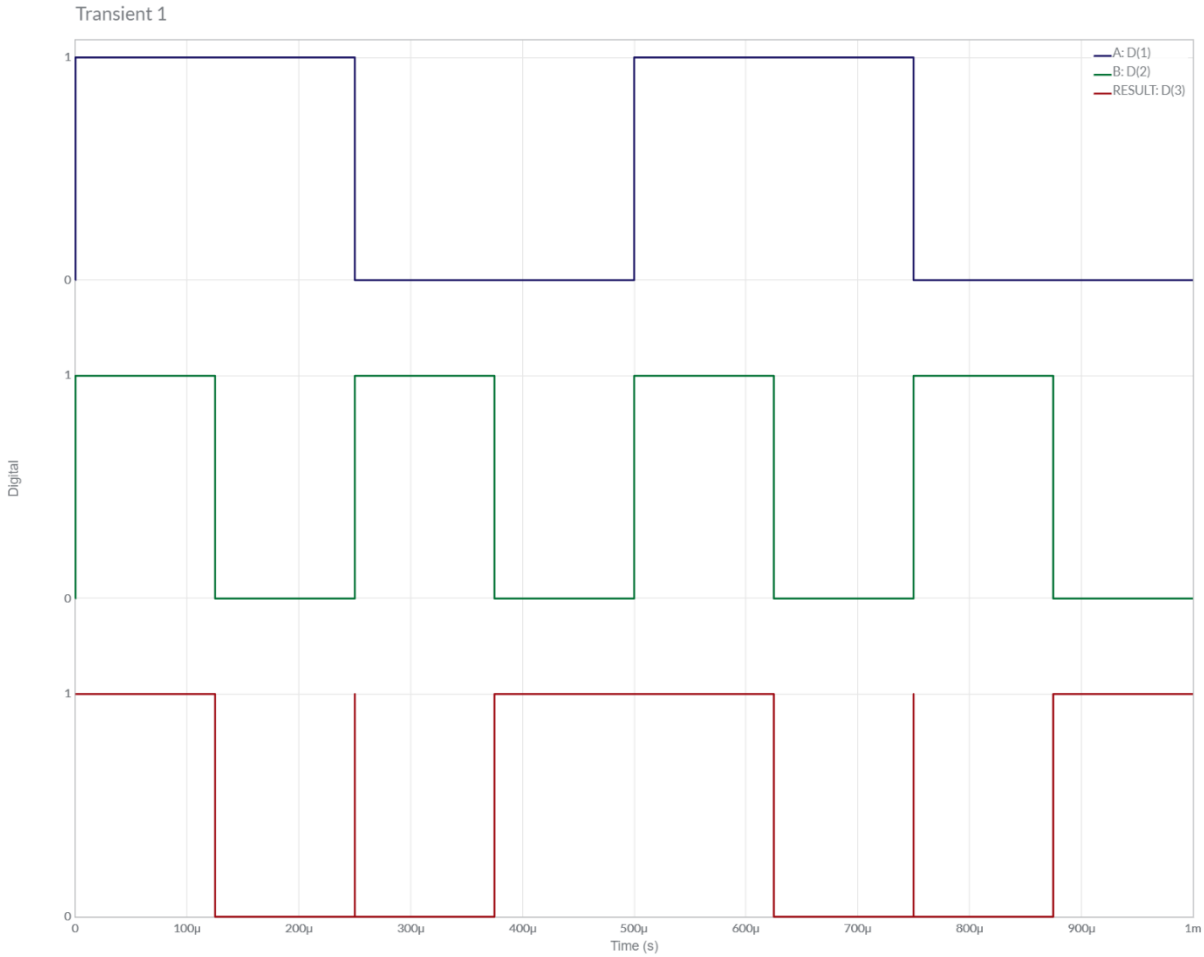


## 13.) 2-INPUT EX-NOR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

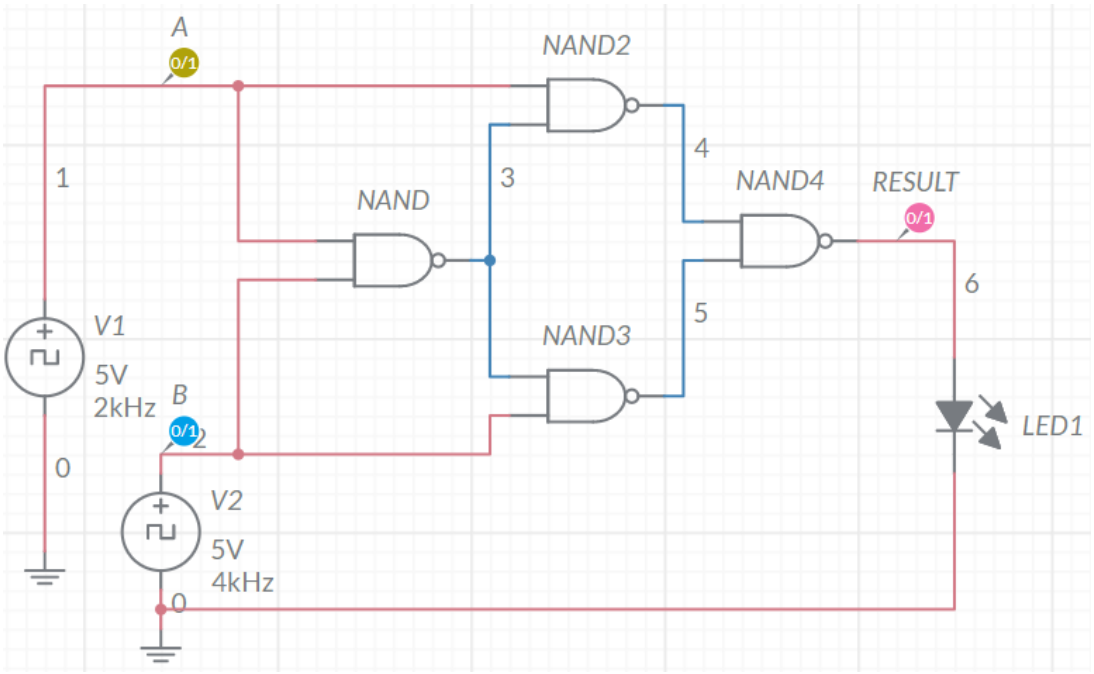



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

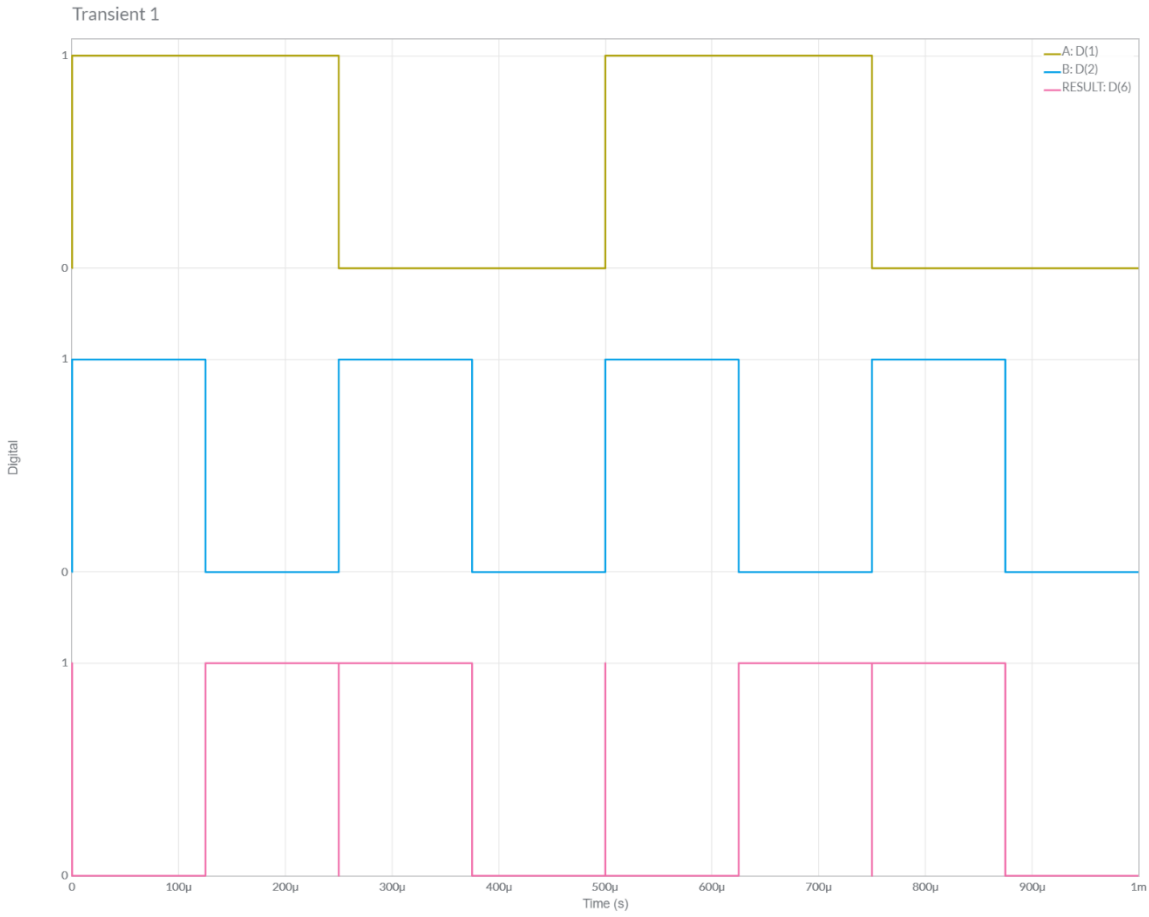


## 14.) 2-INPUT EX-OR USING NAND

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

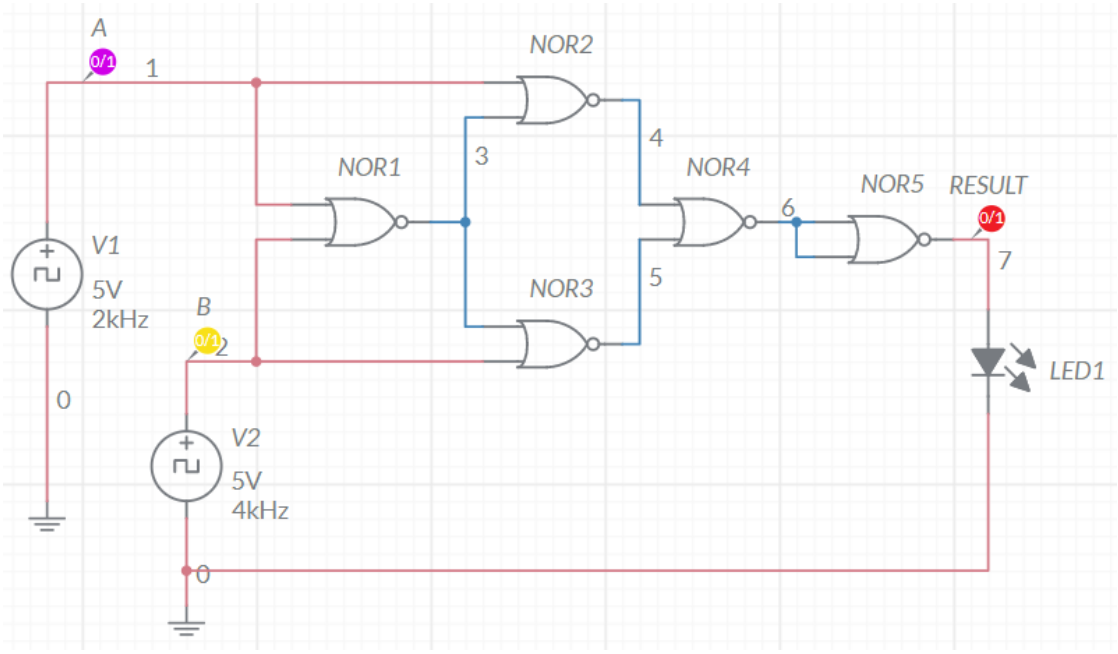



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

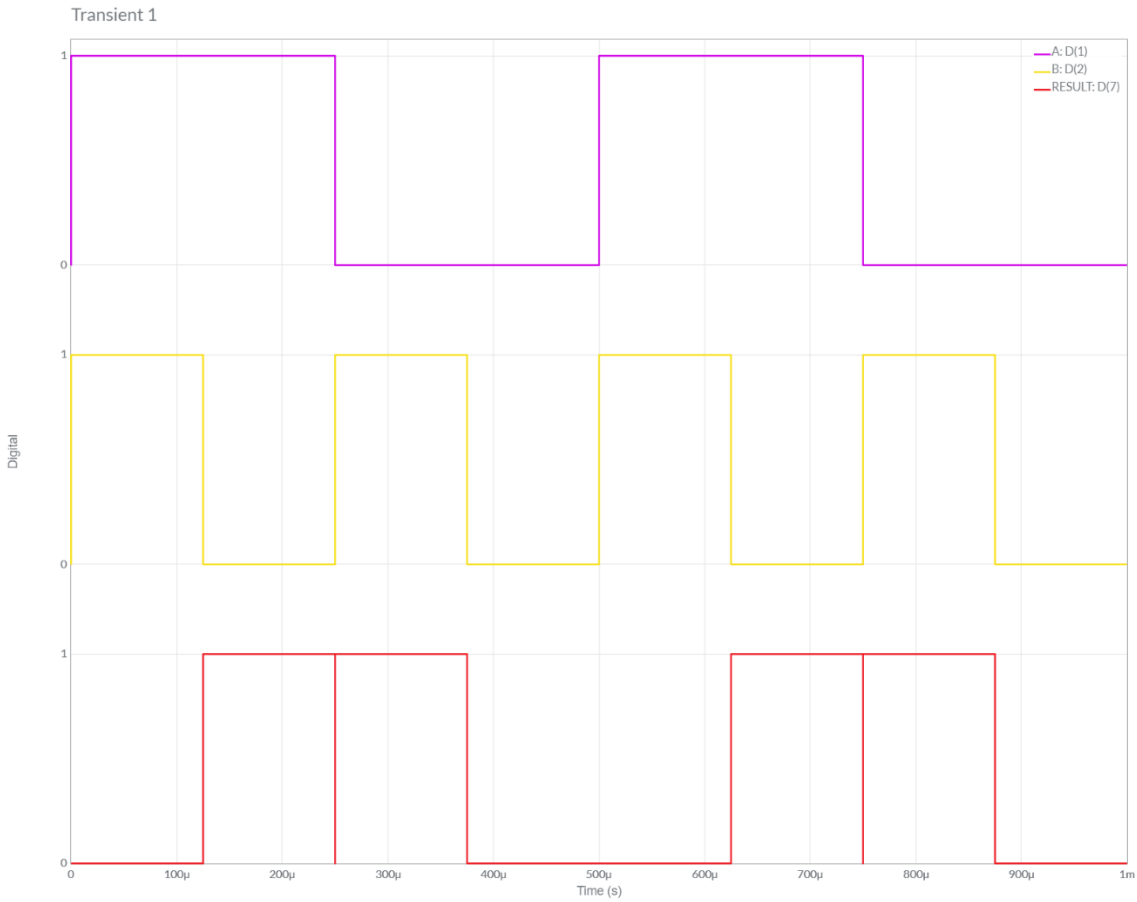


## 15.) 2-INPUT EX-OR USING NOR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

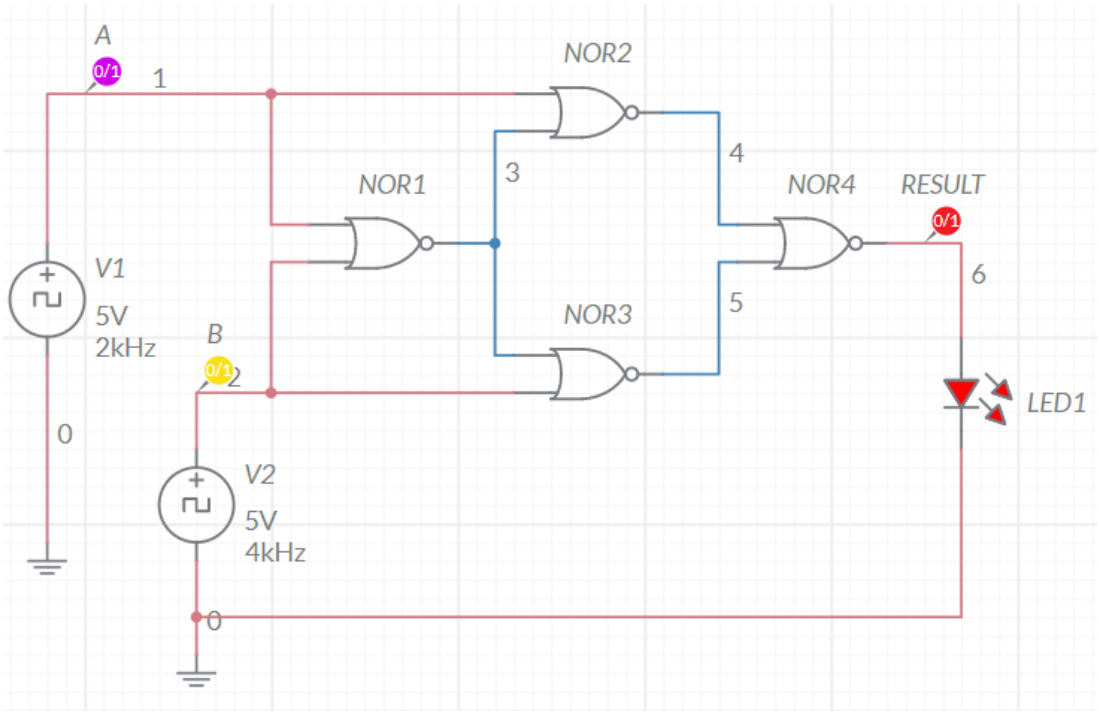



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

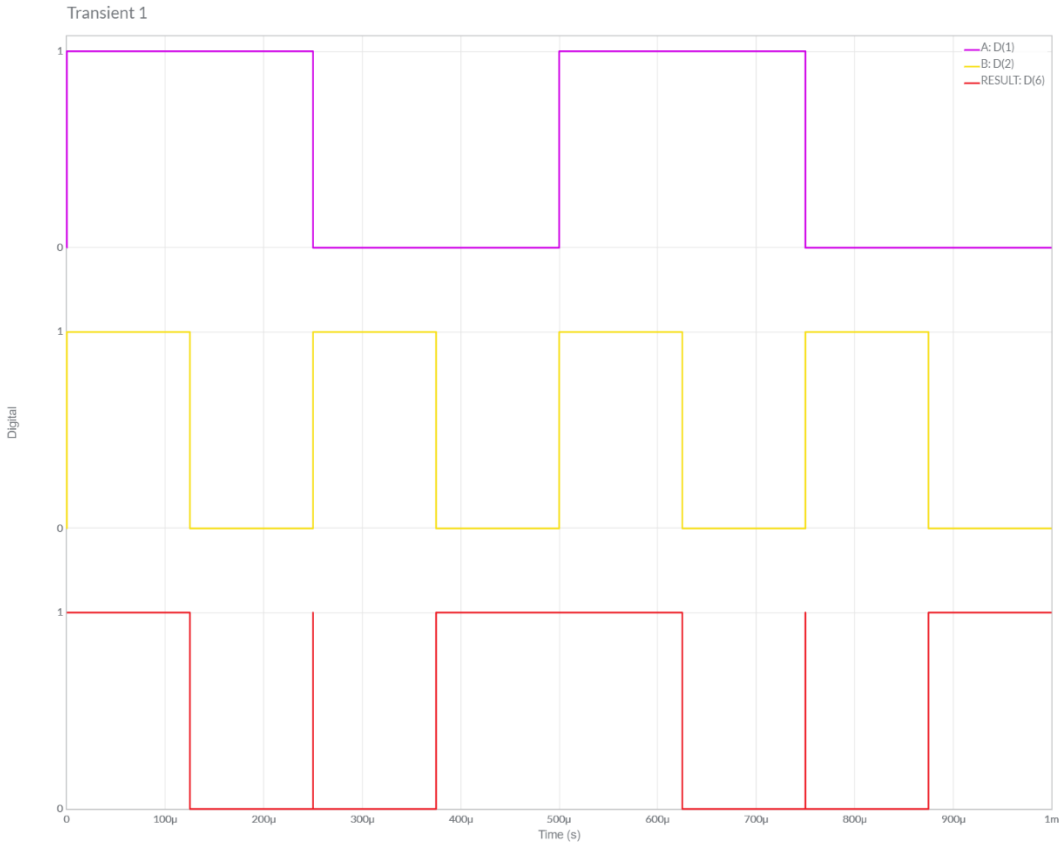


## 16.) 2-INPUT EX-NOR USING NOR

## CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

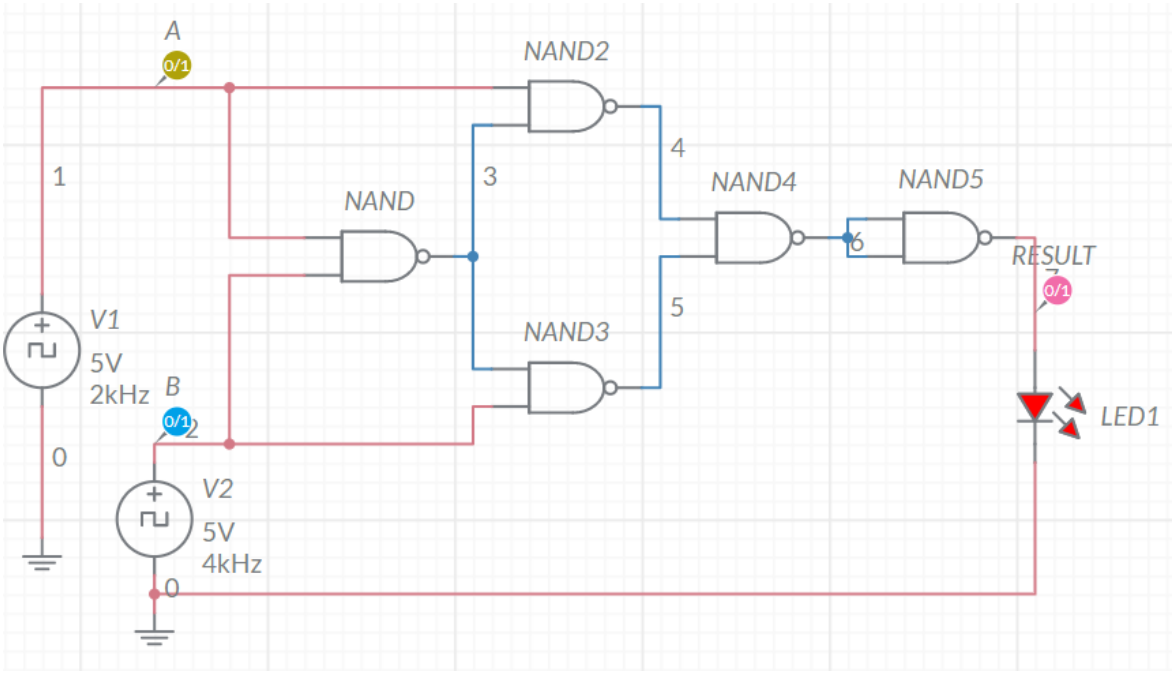



## OUTPUT PLOTS/SCREENSHOTS/WAVEFORMS (FROM MULTISIM):

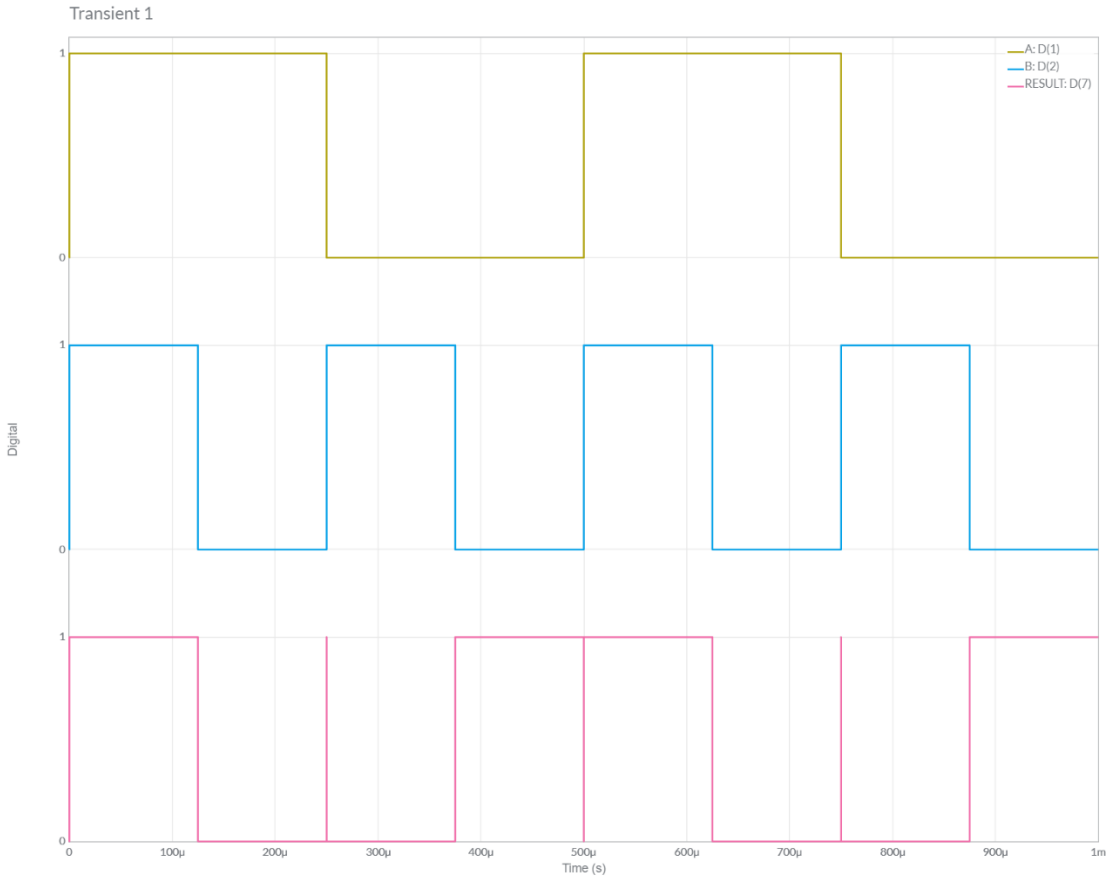


## 17.) 2-INPUT EX-NOR USING NAND

## 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**.

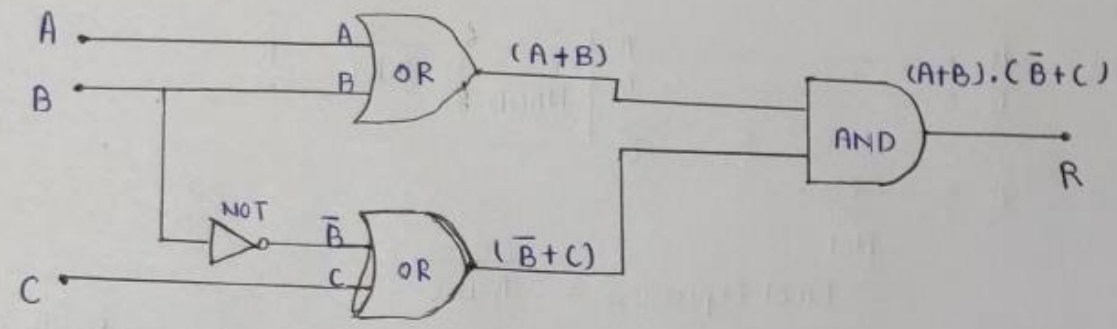
2.) We can also Conclude that *All Logic Gates* can be created using *either of NAND or NOR Gates Combination*. Therefore, *NAND* and *NOR* are also Called **Universal Gates**.

# ASSIGNMENT SECTION

## Question -1

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

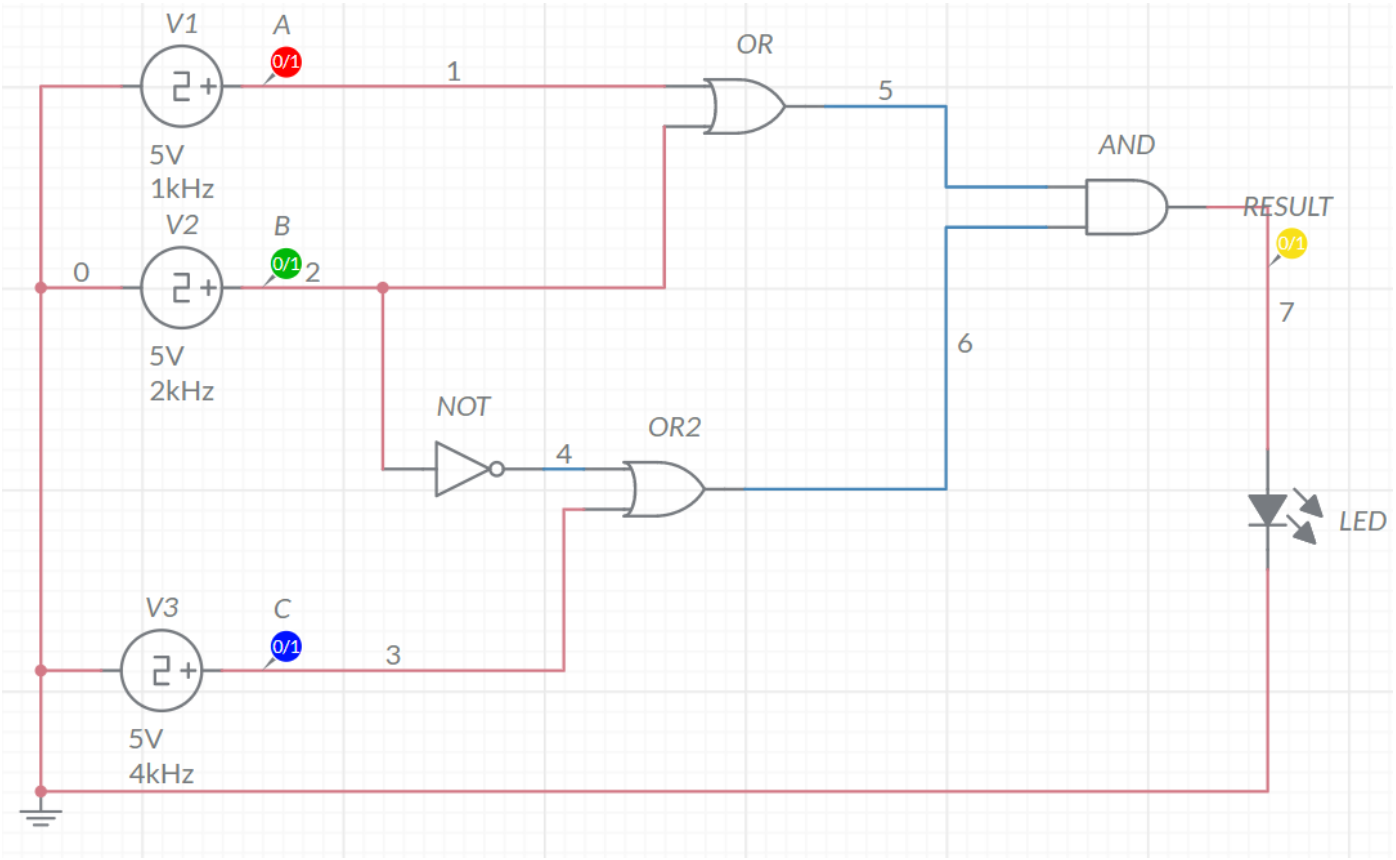
(A) Circuit 1



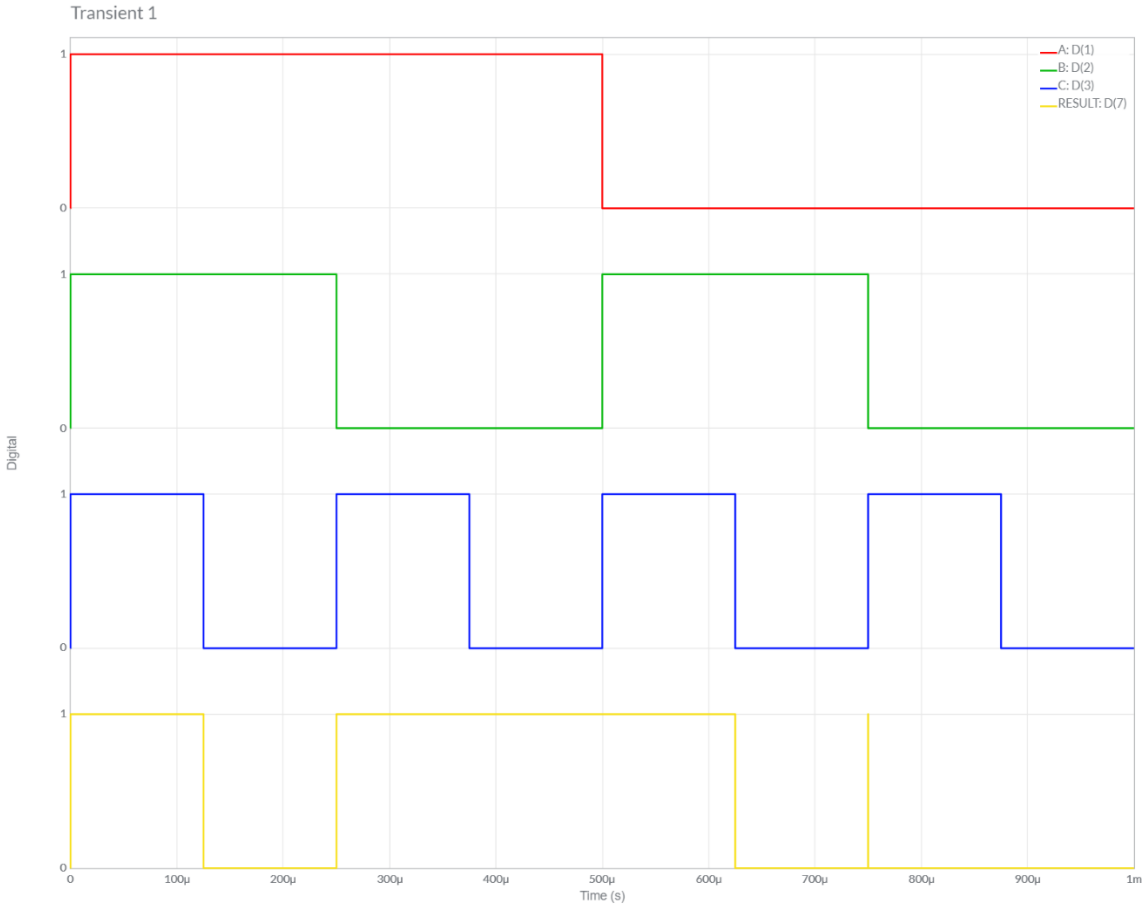

Final Expression = $(A+B).(\bar{B}+C)$

| Sr No | A | B | C | $\bar{B}$ | A+B | $\bar{B}+C$ | Result : $(A+B).(\bar{B}+C)$ |
|---|---|---|---|---|---|---|---|
| 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 | Theoratical 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:*

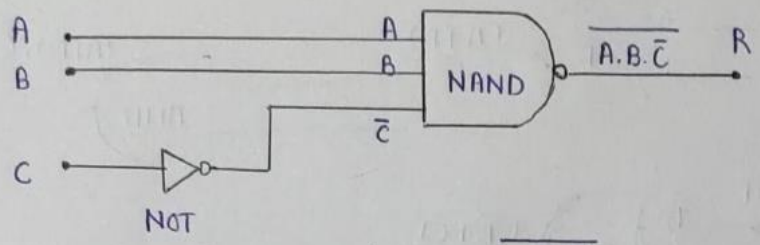
We can observe from Above Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

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

# Question -2

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

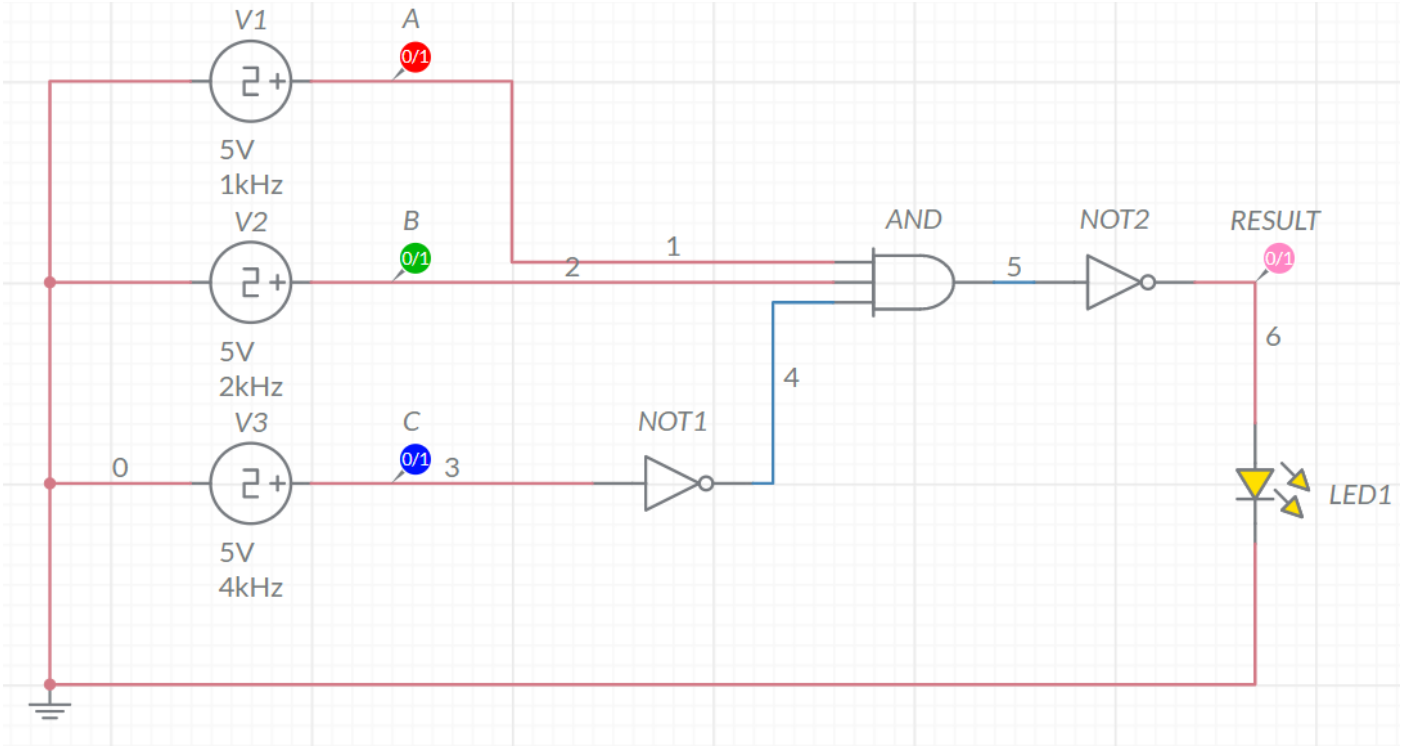
Circuit 2 :



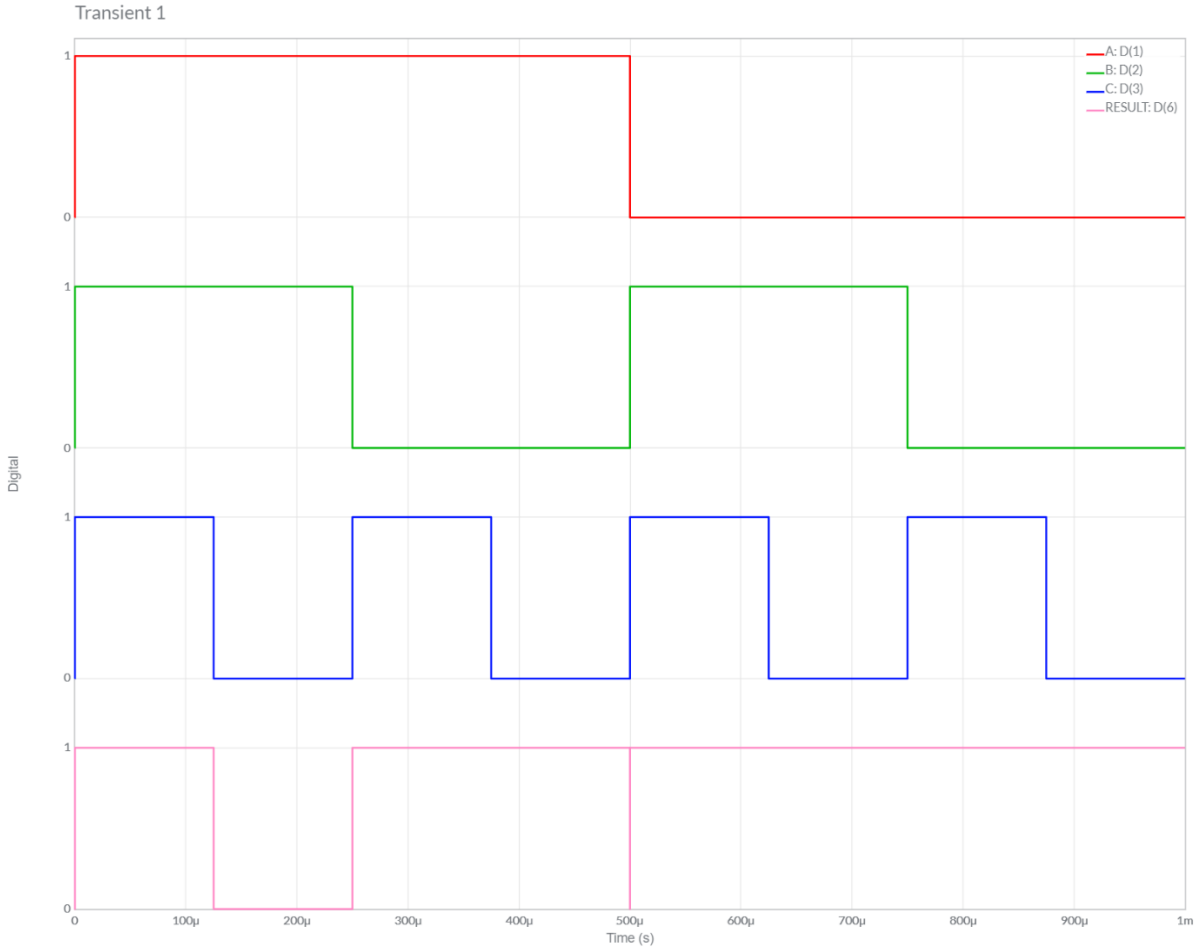

Final Expression = $\overline{A.B.\bar{C}}$

| Sr No | A | B | C | $\bar{C}$ | $A.B.\bar{C}$ | Result : $\overline{A.B.\bar{C}}$ |
|---|---|---|---|---|---|---|
| 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 | Theoratical 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:*

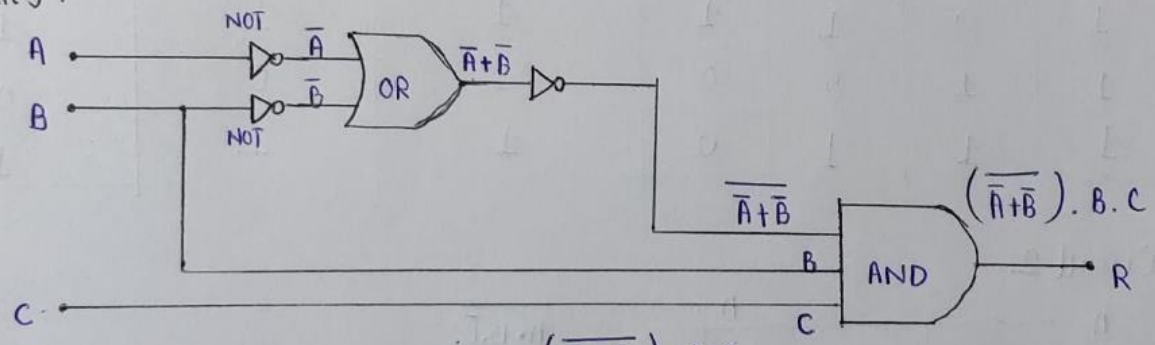
We can observe from Above Graph, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

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

# Question -3

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

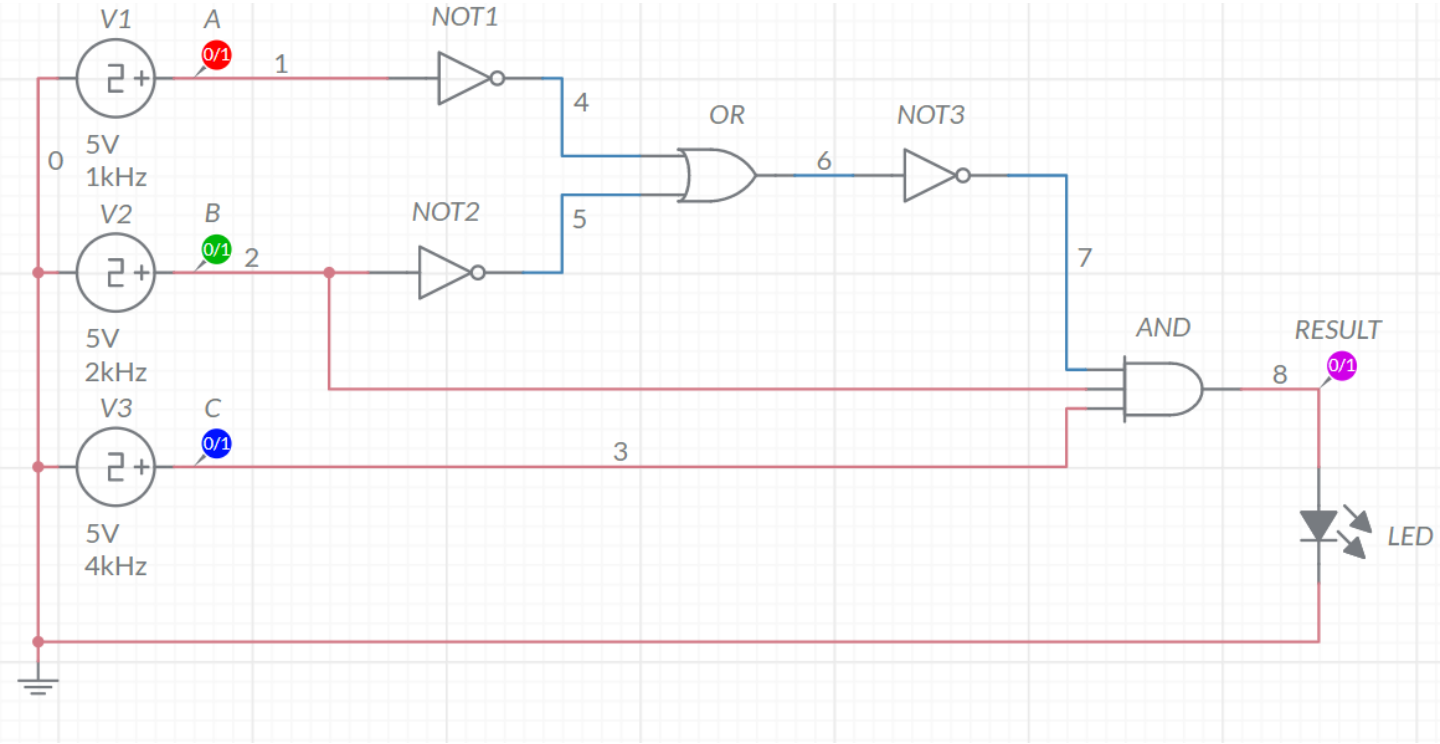
Circuit 3 :



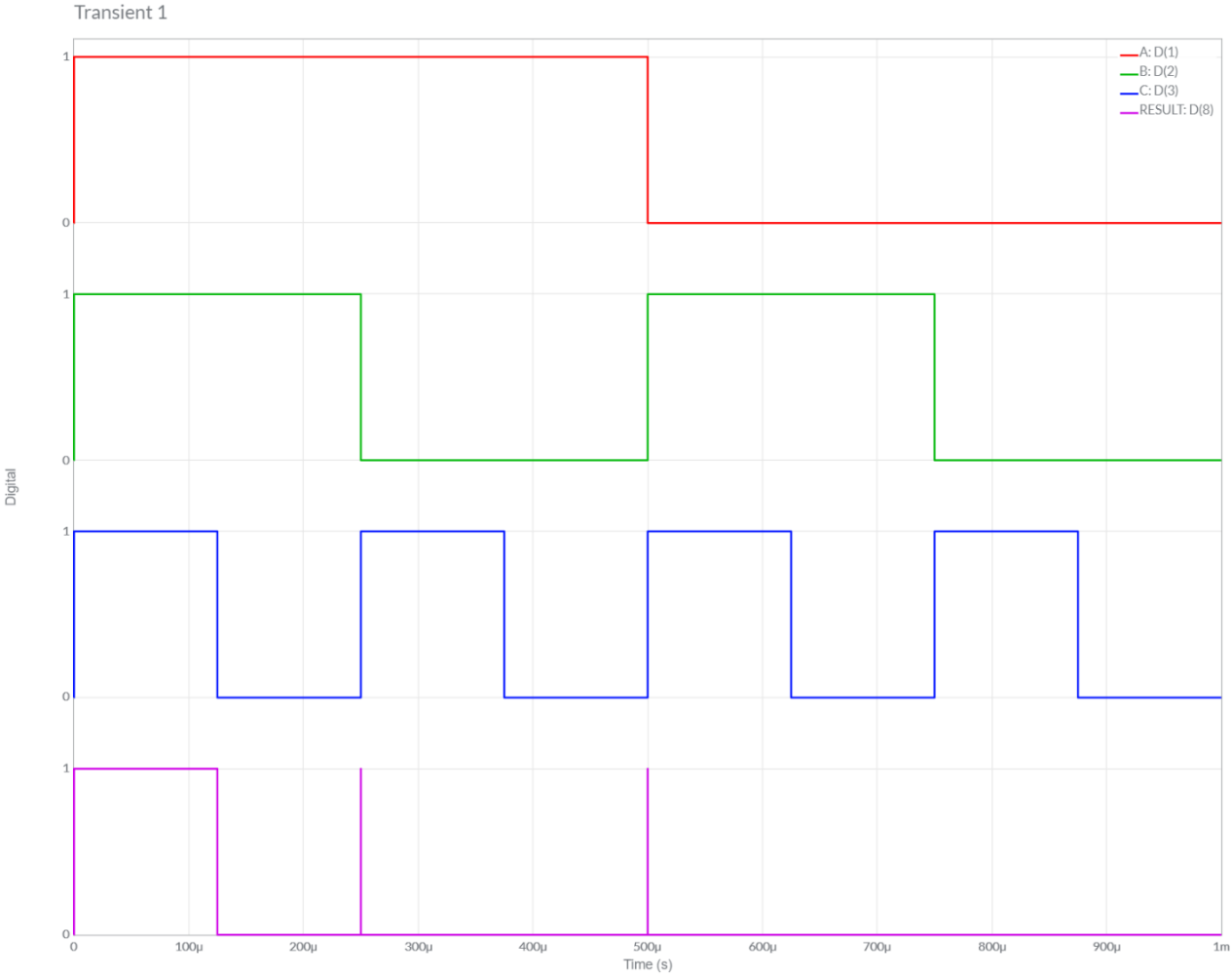

Final Expression : $(\overline{\bar{A}+\bar{B}}).B.C$

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