



Expt. No:

4

Date:

03-09-2020

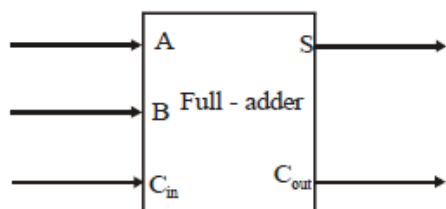
Full Adder and Full Subtractor**AIM:** To design and implement Full Adder and Full Subtractor Circuits.**SOFTWARE TOOLS / OTHER REQUIREMENTS:**

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 (C_{out}). Fig (b) shows the truth table for the full-adder.



(a) logic symbol

Inputs			Outputs	
A	B	C _{in}	S	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b) truth table

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

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

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

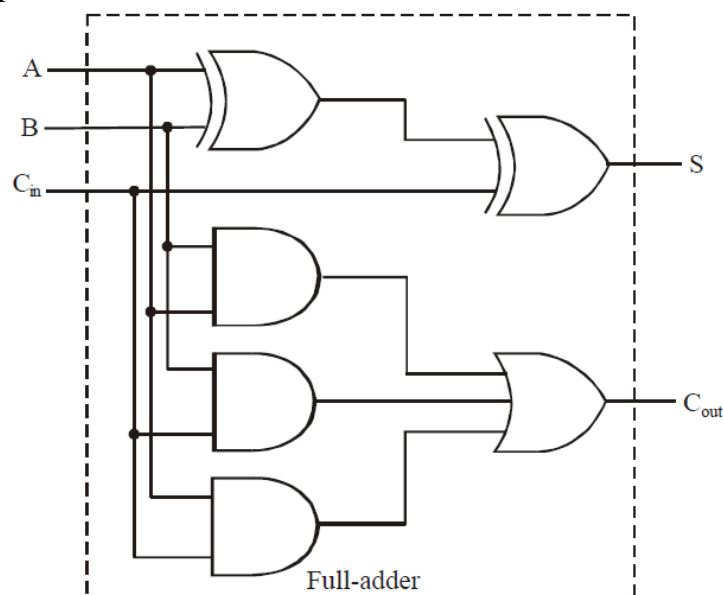


$$S = A \oplus (B \oplus C_{in})$$
$$= A \oplus B \oplus C_{in}$$

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

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

The equations for Sum and Carry can be easily implemented by using logic gates. From equation of Sum we find that to implement the 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.

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

(a) logic symbol

(b) truth table



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

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

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

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

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

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

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

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

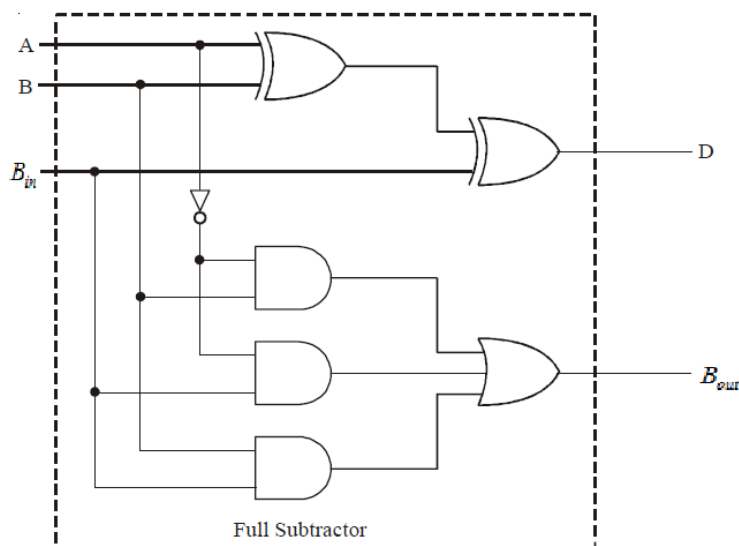
The Boolean expression for the difference bit (D) can be obtained by summing and simplifying all the input combinations from the truth table which have 1 in the corresponding difference column. The final simplified expression for difference is given by

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

and, the Boolean expression for the borrow bit,

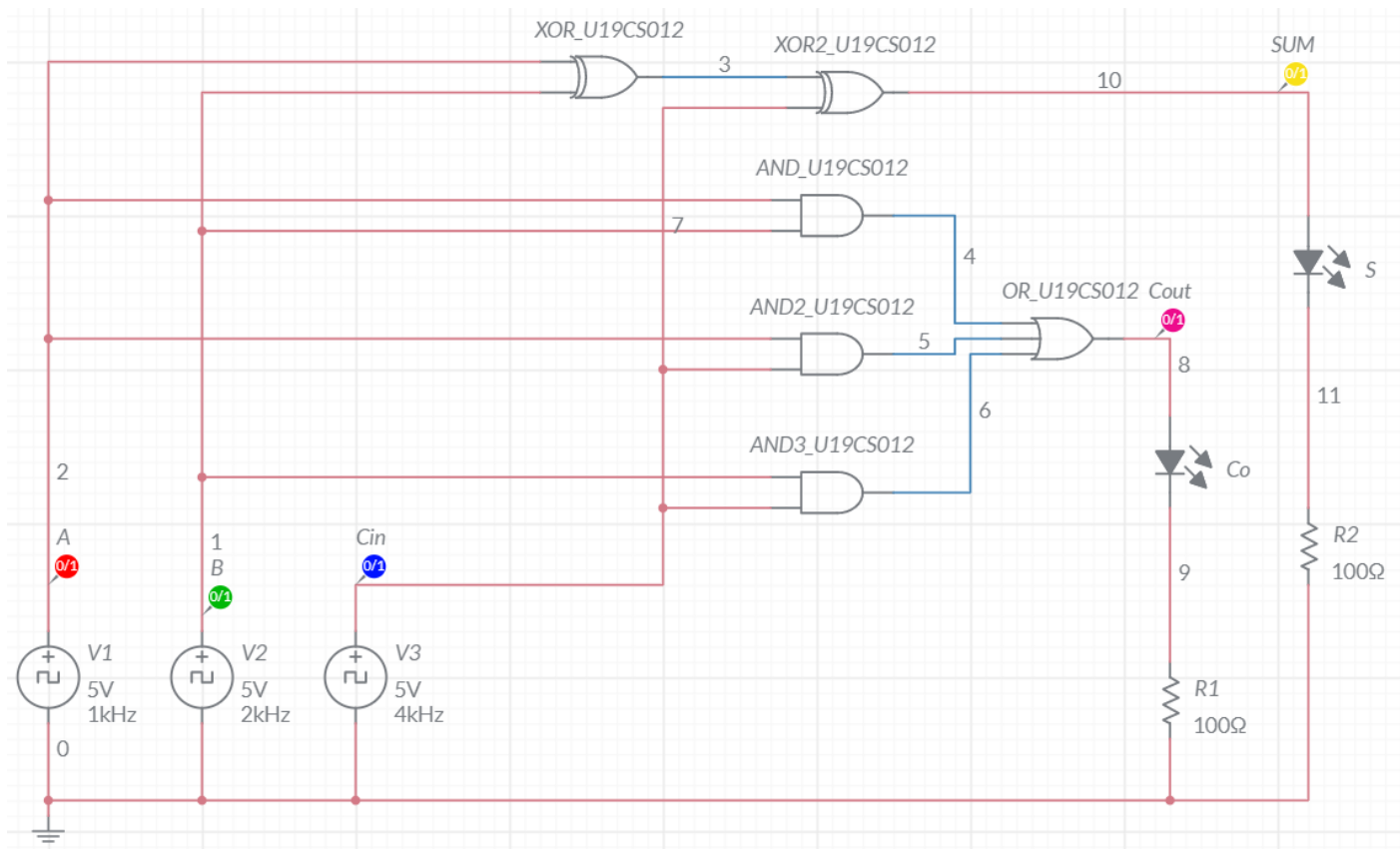
$$B_{out} = \bar{A}B + BB_{in} + \bar{A}B_{in}$$

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.



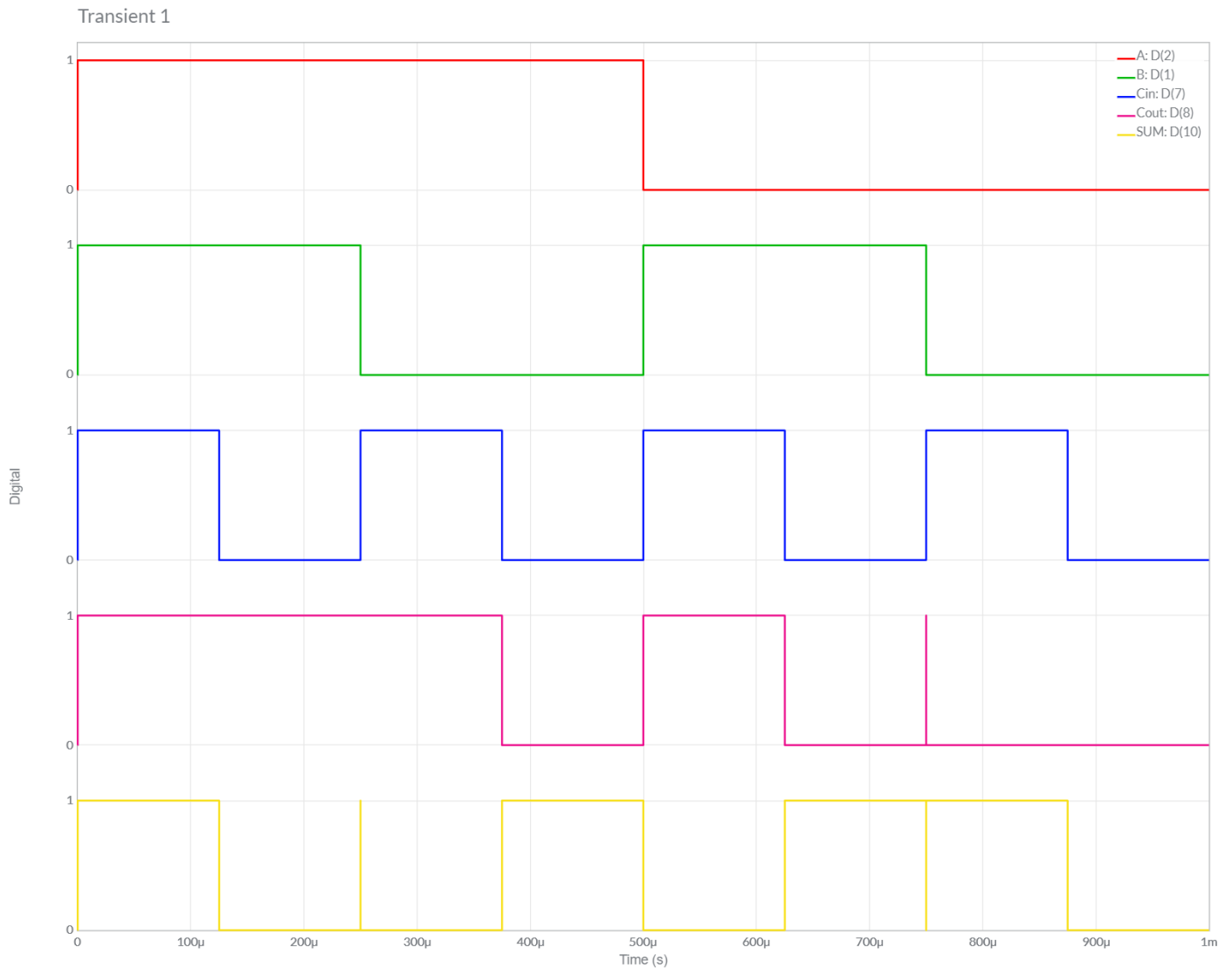


FULL ADDER: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



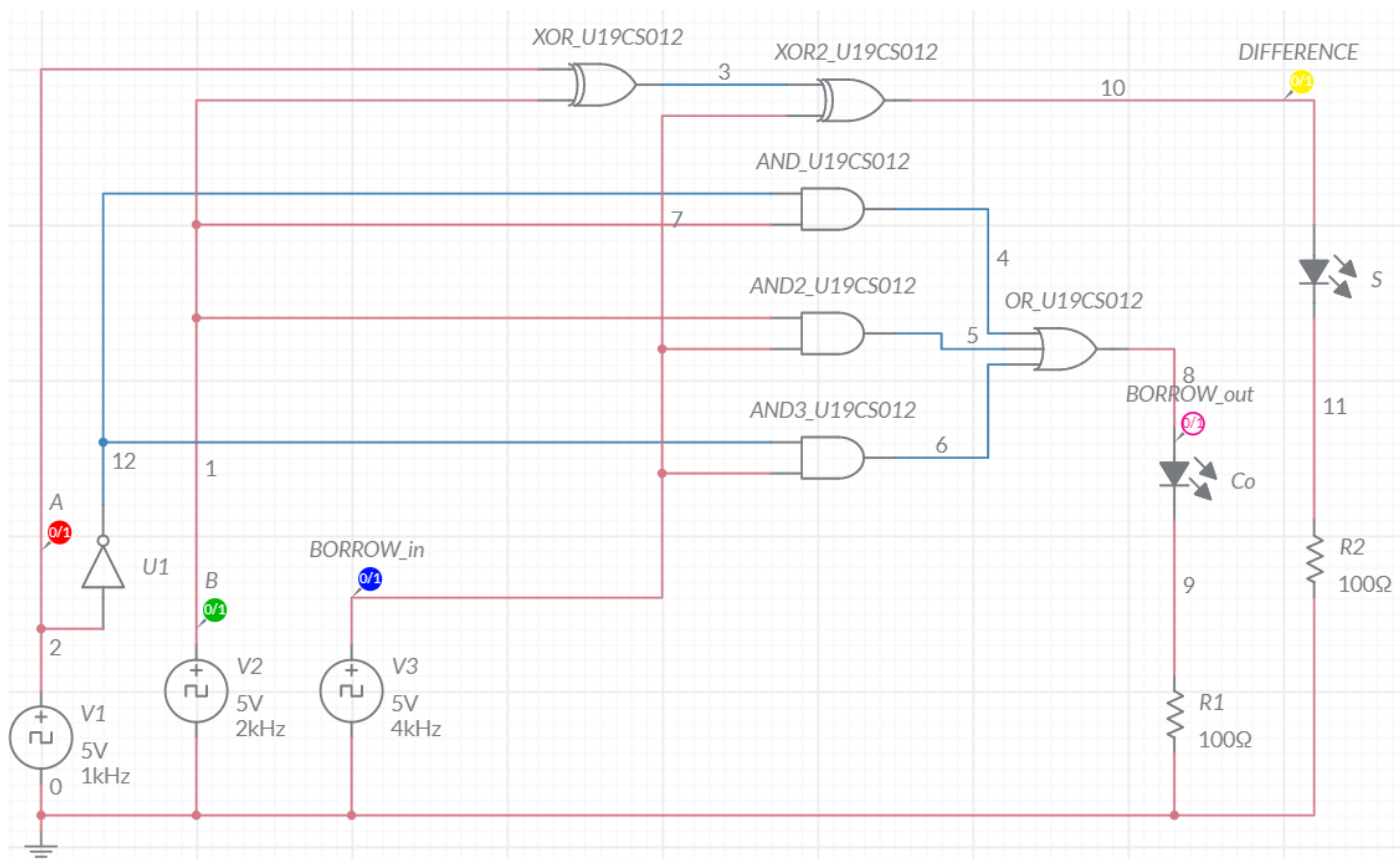


FULL ADDER: OUTPUT PLOTS/WAVEFORMS (FROM MULTISIM):



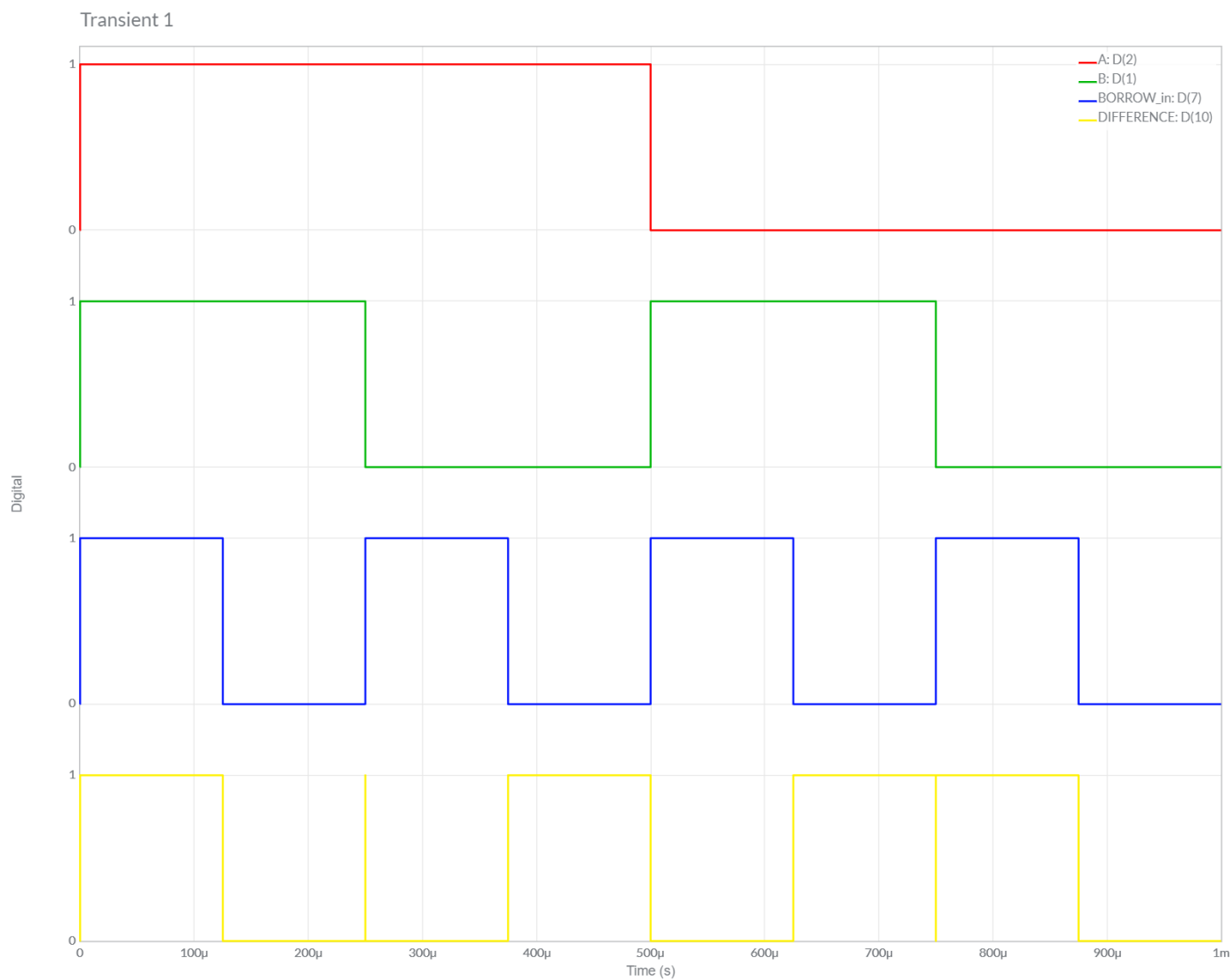


FULL SUBTRACTOR: CIRCUIT/CONNECTION DIAGRAMS (MULTISIM)





FULL SUBTRACTOR: OUTPUT PLOTS/WAVEFORMS (MULTISIM)



CONCLUSIONS

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



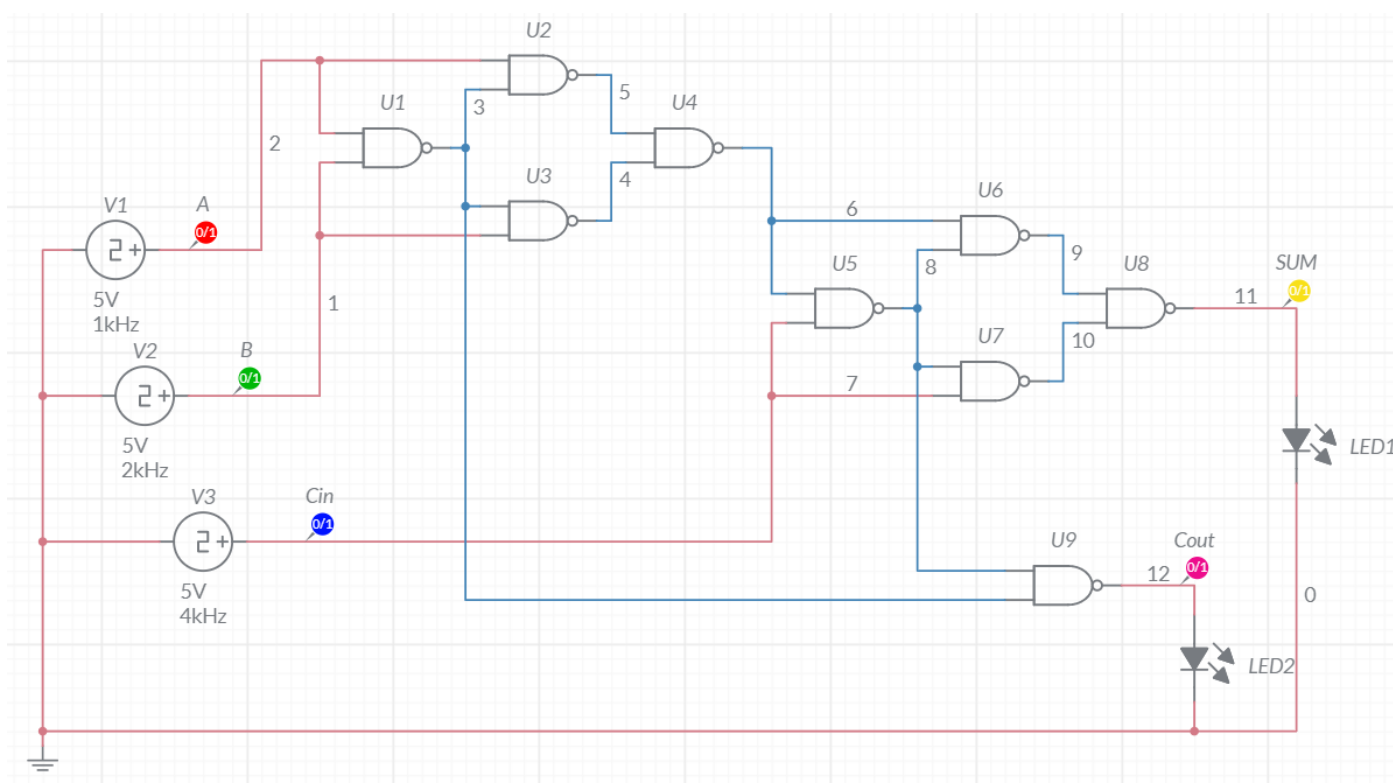
ASSIGNMENT-4

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

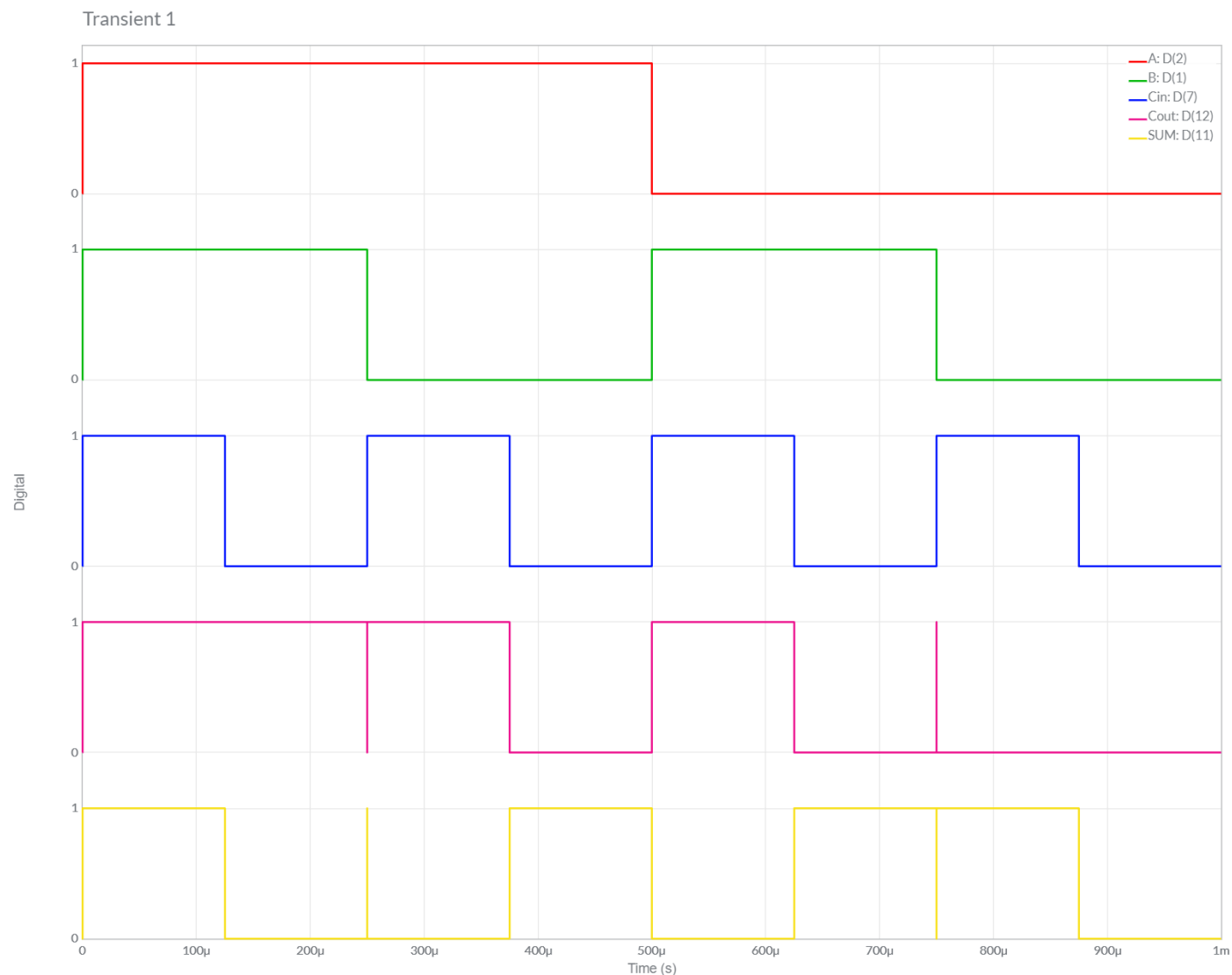


b.) Truth Table

	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

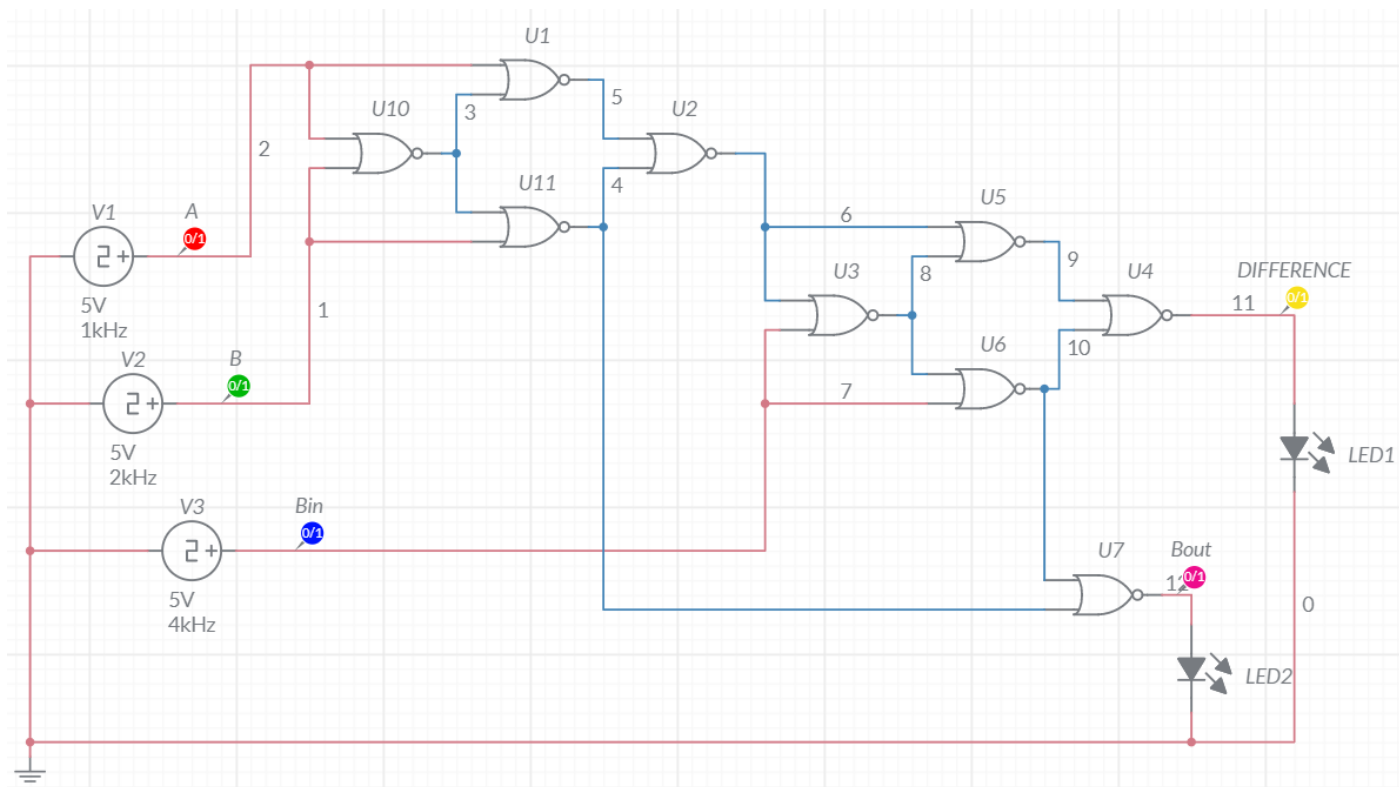
We can observe from Above Graph and Truth Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

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

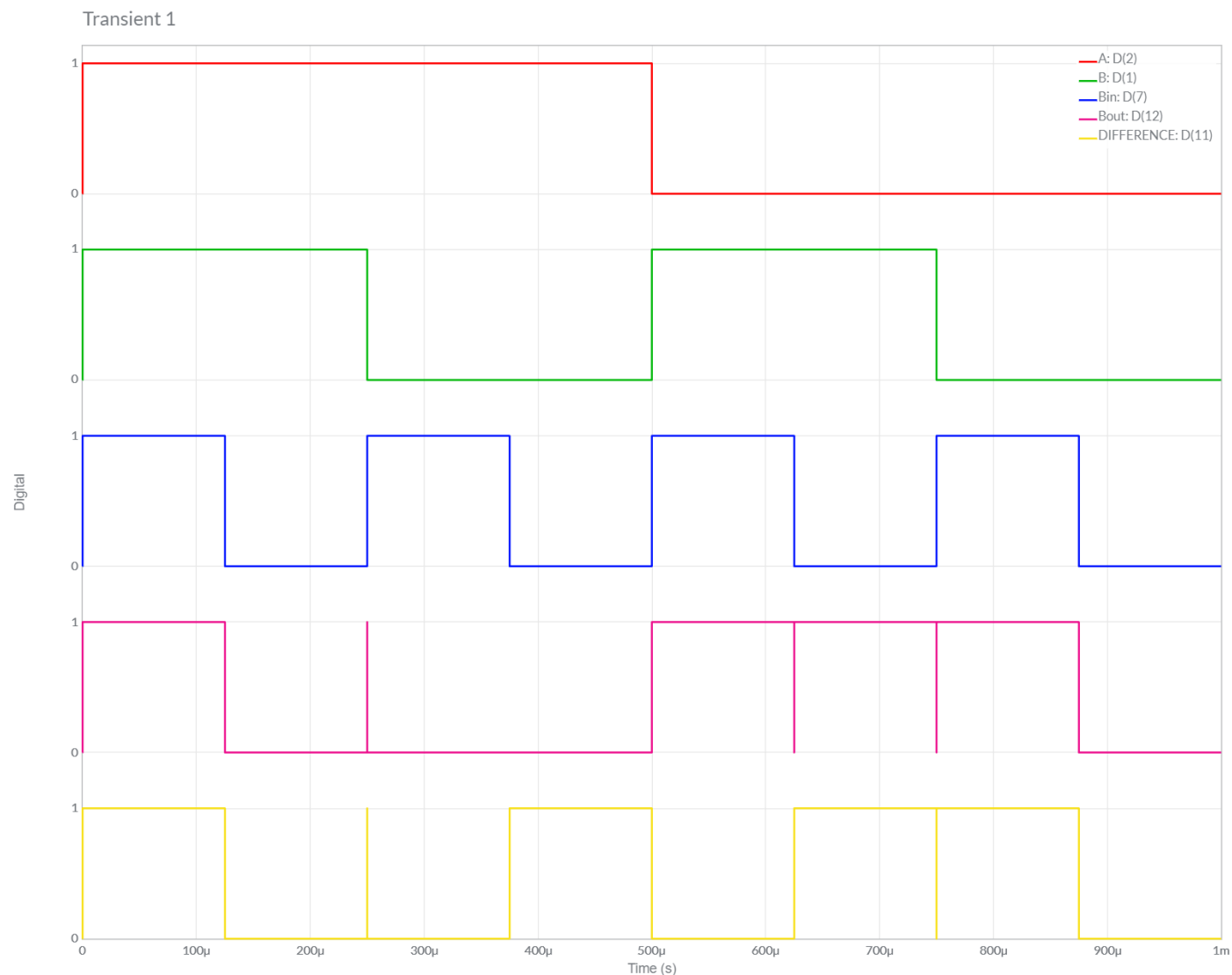


c.) Truth Table

	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

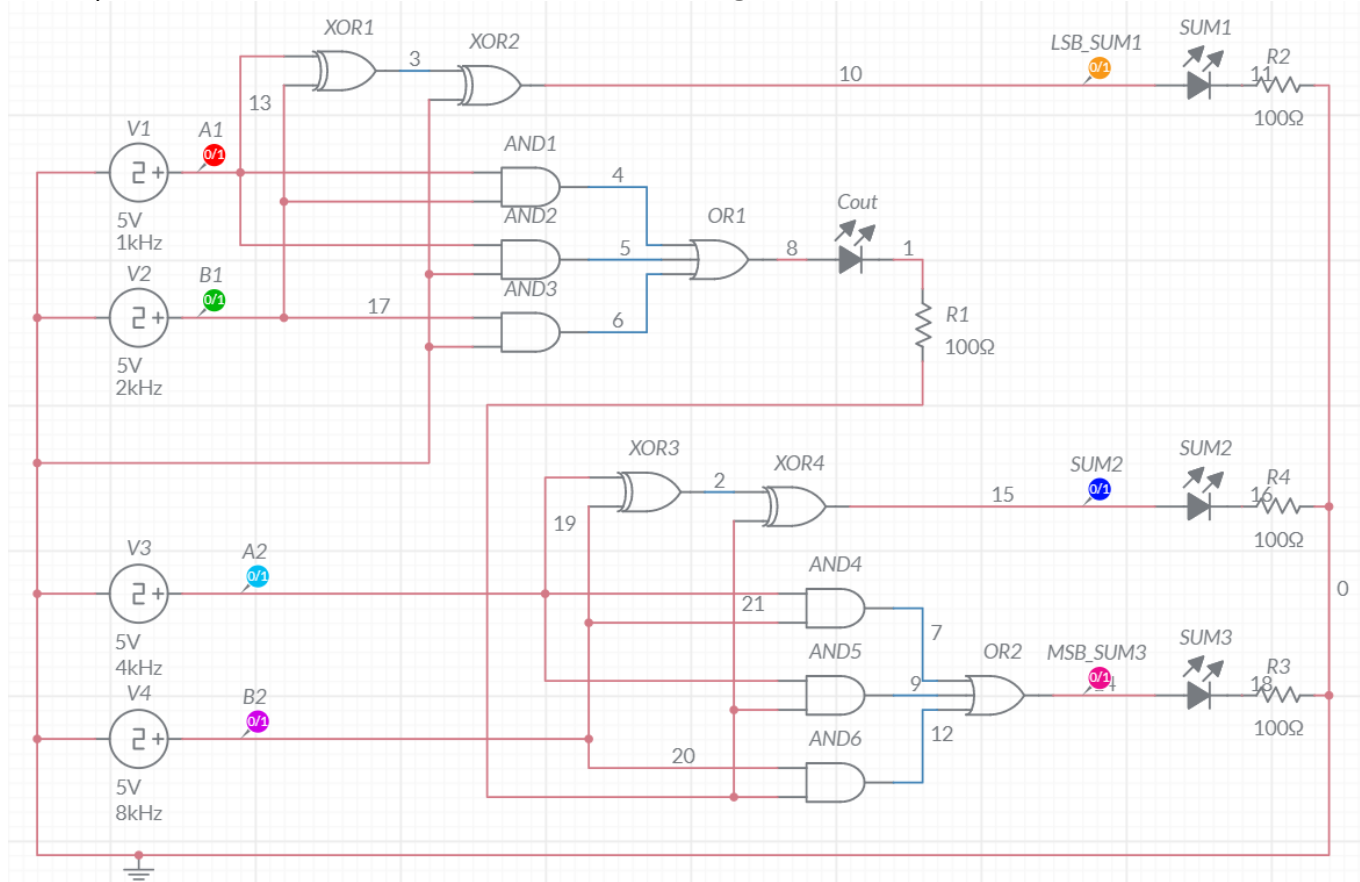
We can observe from Above Graph and Truth Table, Both the *Theoretical* and *Multisim* Values of Given Circuit are **Equal**.

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

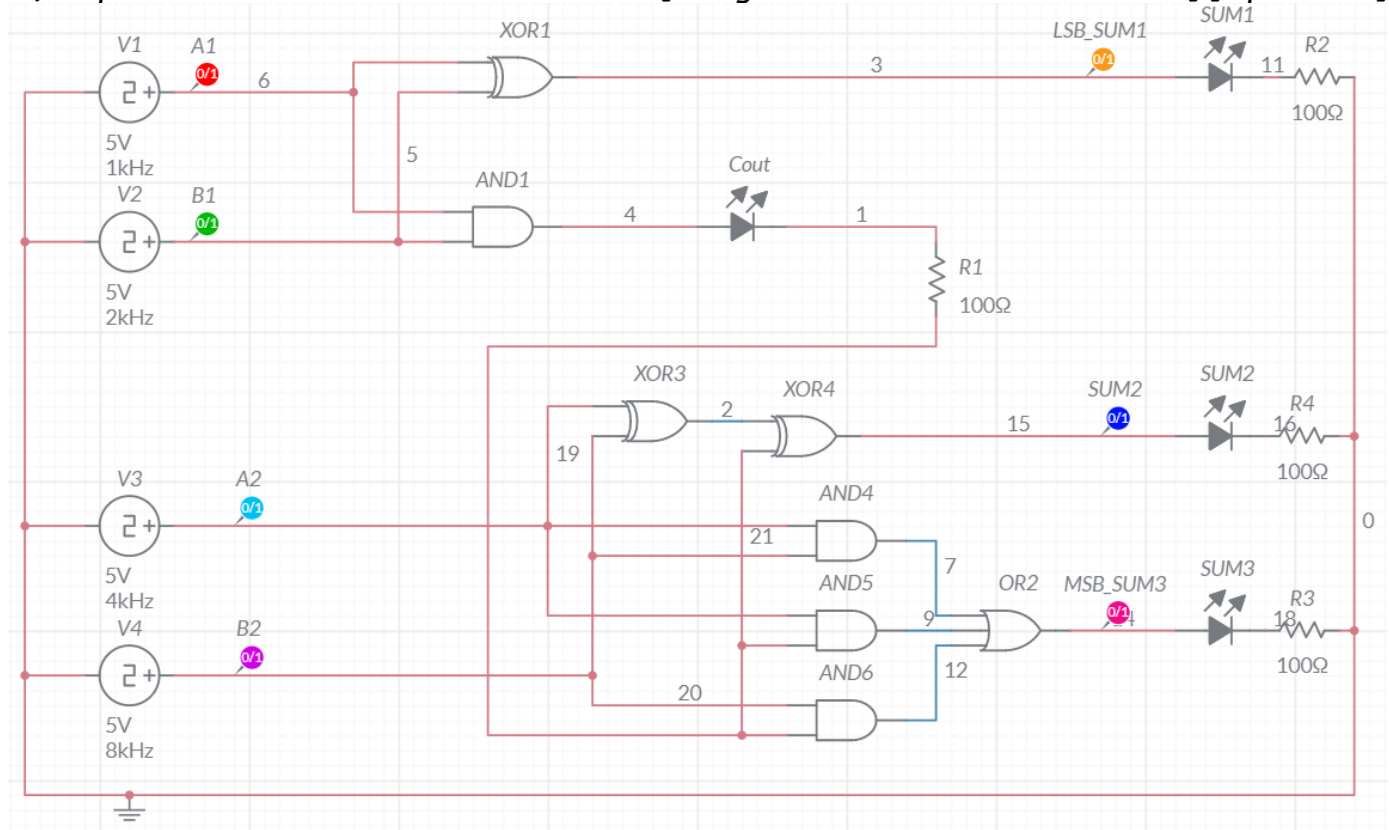


3. Two bit Adder circuit using Full Adders. Attach screenshots for any four input combinations.

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

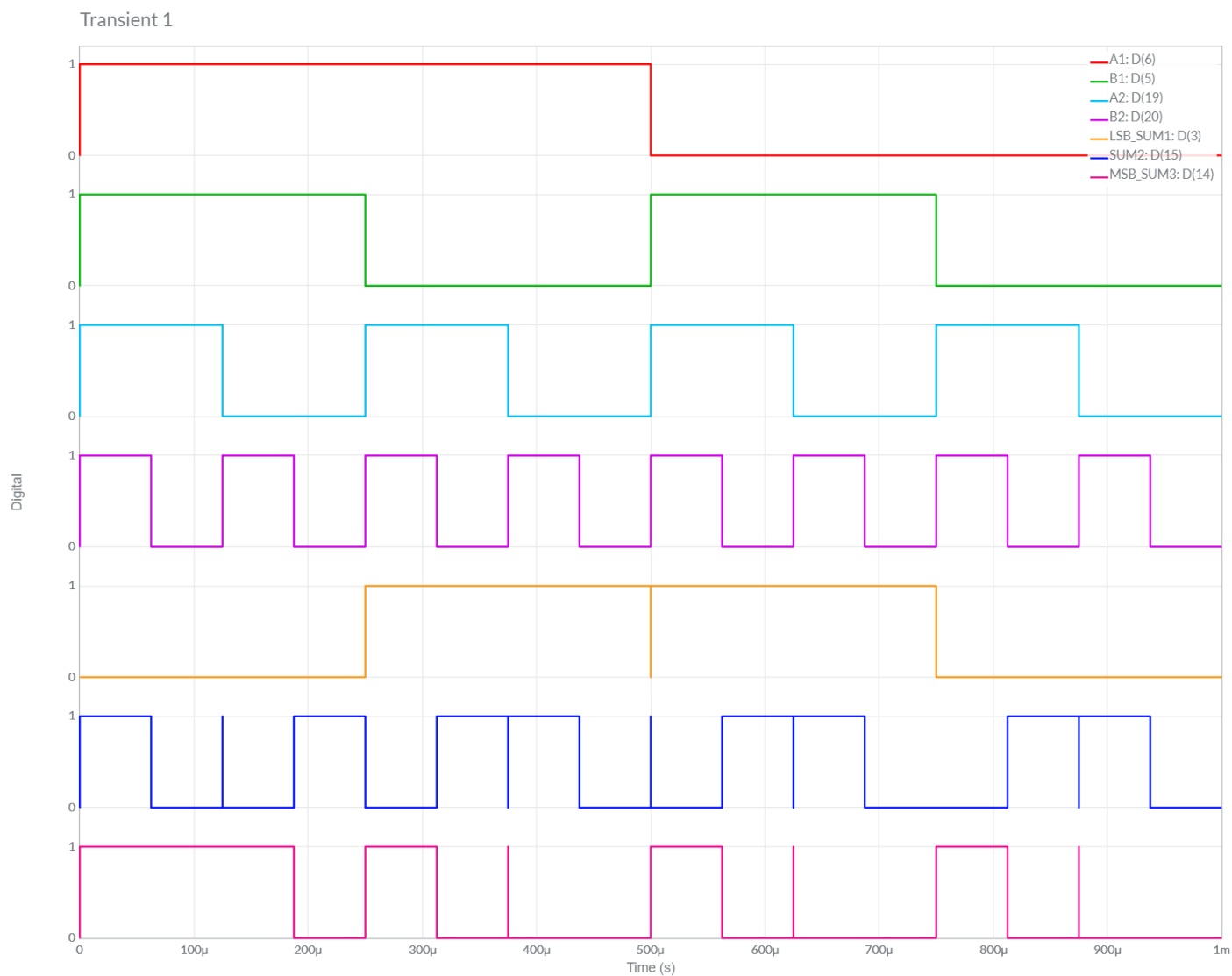


b.) Implement the circuit in Multisim online [Using 1 Full Adders + 1 Half Adder] [Optimized]





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.