NAME : ADITYA RAJ

ADMISSION NO.: U20CS100

Expt. No: **8**

Date: **14/10/2021**

IMPLEMENTATION OF ADDERS AND SUBTRACTOR

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

SOFTWARE TOOLS / OTHER REQUIREMENTS:

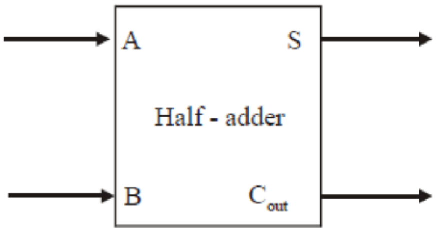
1. Multisim Simulator
2. Logic Gates (NAND and NOR Gates)

THEORY:

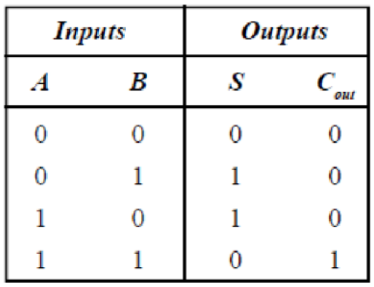
HALF ADDER:

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 (Cout). 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.

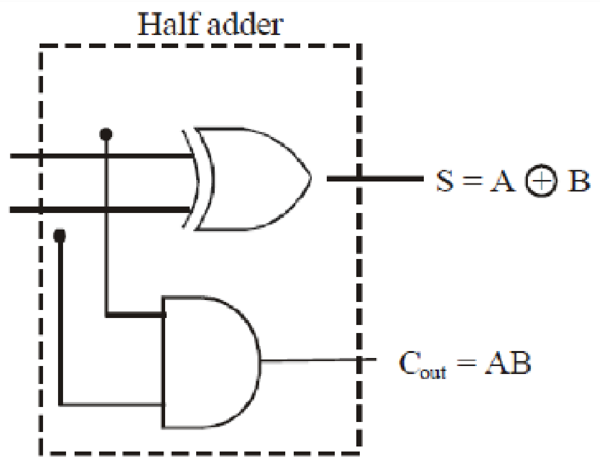
$$S = A\overline{B} + \overline{A}B$$

$$= A \oplus B$$

and the Boolean expression for the carry output by,

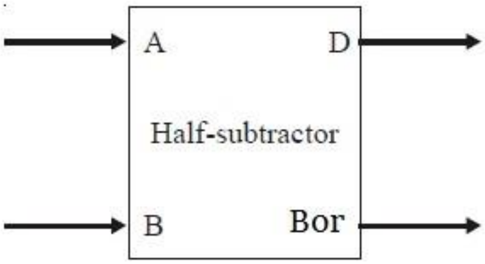
$$C_{ont} = AB$$

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 excluse-OR gate while the carry output (Cout) is the output of a two-input AND gate.

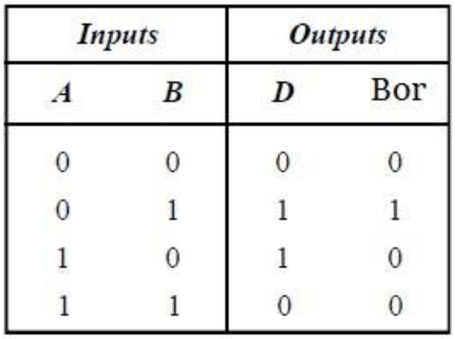


HALF SUBTRACTOR:

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 it 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:

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

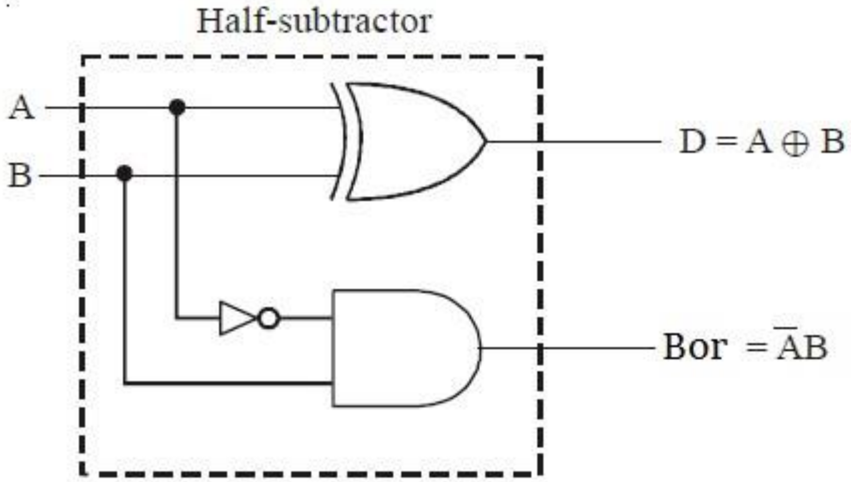
The Boolean expression for the difference bit (D) can be expressed by the equation.

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

and the Boolean expression for the borrow bit,

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

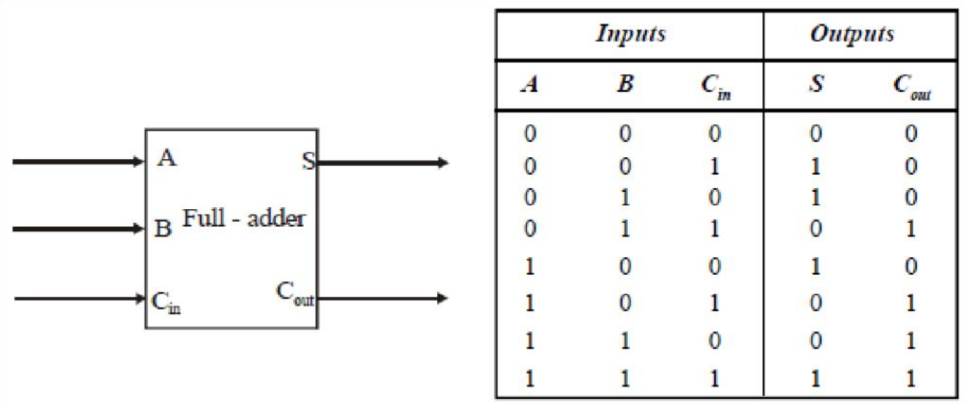
The above two expressions can be implemented by a logic circuit shown in Fig. below. As seen from this figure, the difference output (D) is obtained from an exclusiveOR gate while the borrow bit (Bor) is the output of a two-input AND gate.



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.



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

(*a*) logic symbol (*b*) truth table

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

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

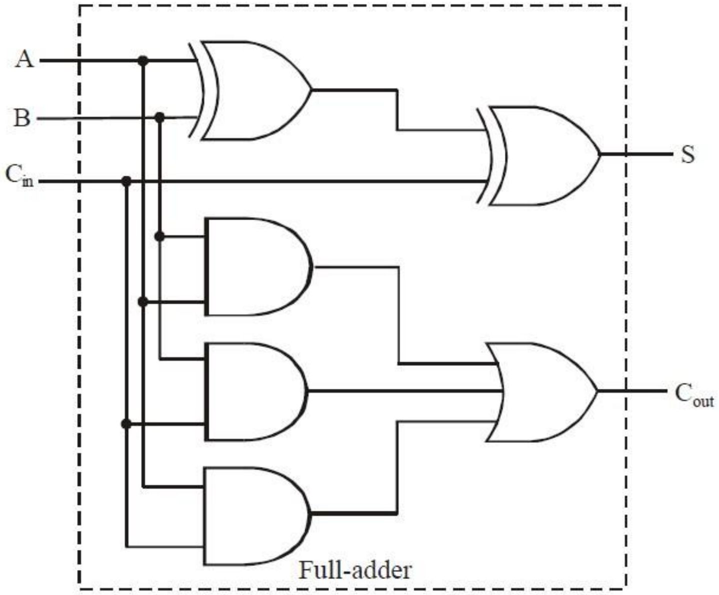
The Boolean expression for the sum output (S) can be obtained from the above truth table by summing and then simplifying the terms for which S=1. Thus, the sum is

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

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

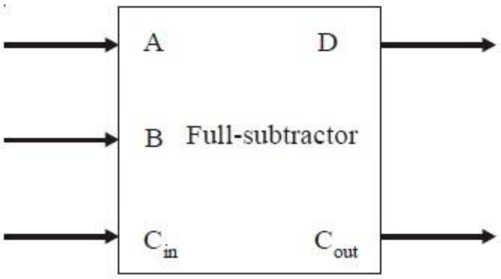
$$C_{ont} = BC_{in} + AC_{in} + AB$$

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 2input Exclusive–OR gates can be used. The first Exclusive–OR gate generates the term *A 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.

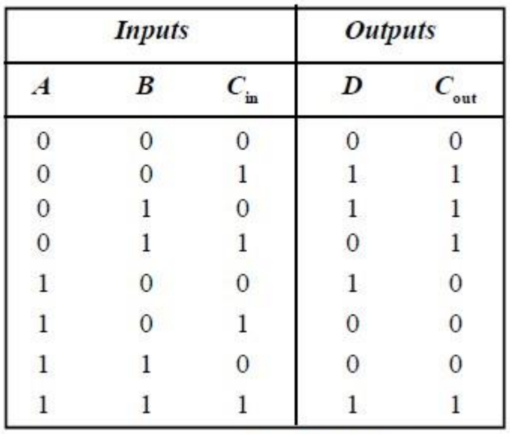


FULL SUBTRACTOR:

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:

$0 - 0 - 0 = 0$ with borrow 0

$0 - 0 - 1 = 1$ with borrow 1

$0 - 1 - 0 = 1$ with borrow 1

$0 - 1 - 1 = 0$ with borrow 1

$1 - 0 - 0 = 0$ with borrow 0

$1 - 1 - 0 = 0$ with borrow 0

$1 - 1 - 1 = 1$ with borrow 1

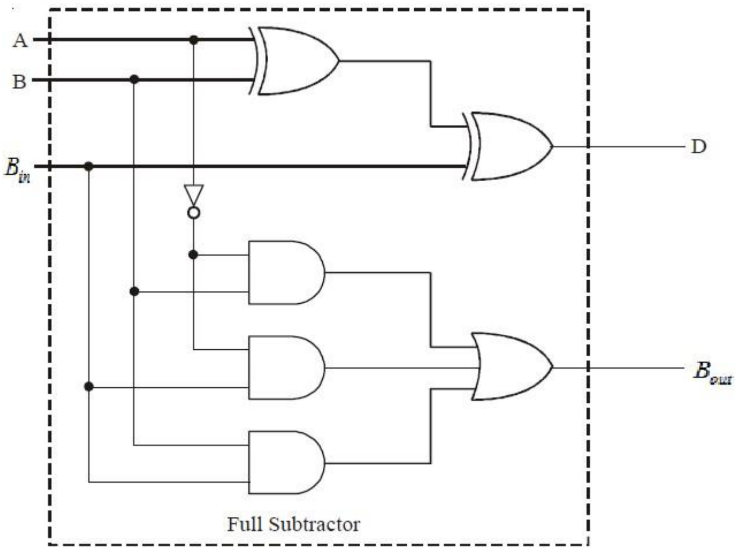
The Boolean expression for the difference bit (D) can be obtained by summing and simplifying all the input combinations from the truth table which have 1 in the corresponding difference column. The final simplified expression for difference is given by

$$D\ A\ B\ B = \square\ \square\ _{in}$$

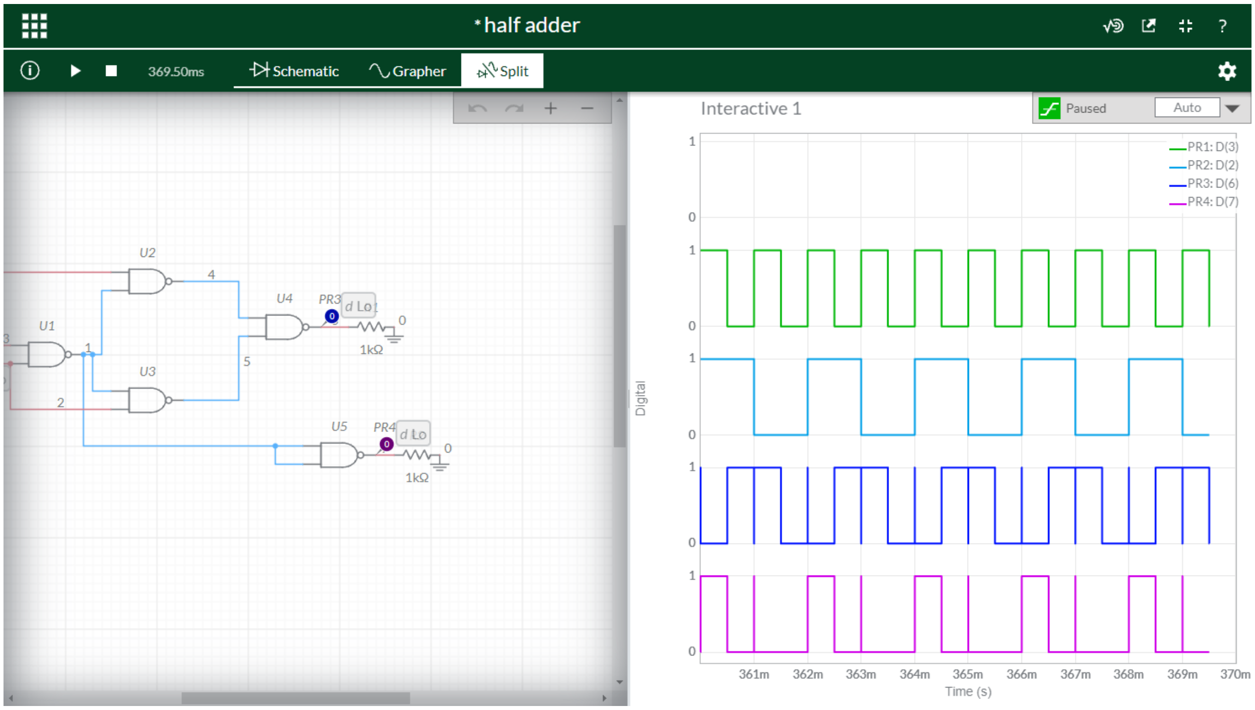
and, the Boolean expression for the borrow bit,

$$B_{out} = AB\ BB\ AB + {}_{in} + {}_{in}$$

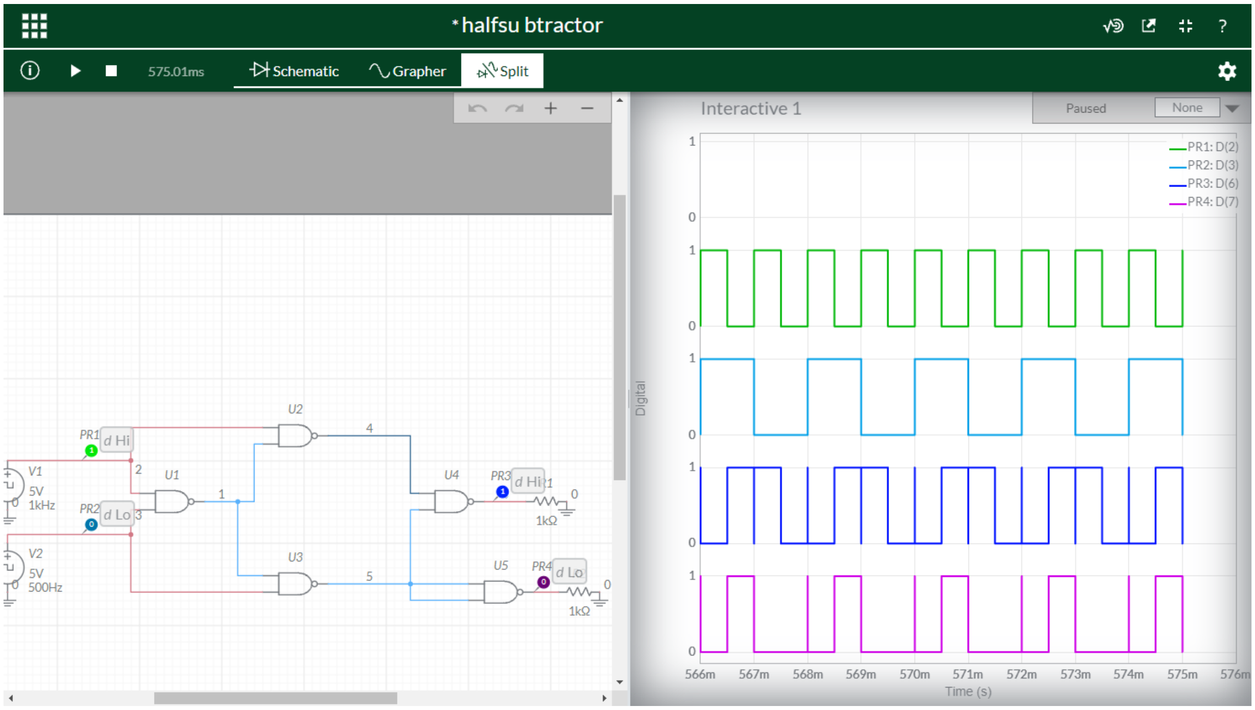
From the above two expressions we find that to implement full-subtractor's difference output function, two-2 input Exclusive–OR gates can be used. Similarly from equation of borrow we find that to implement the full-subtractor's borrow output function, a NOT gate, three 2-input AND gates followed by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



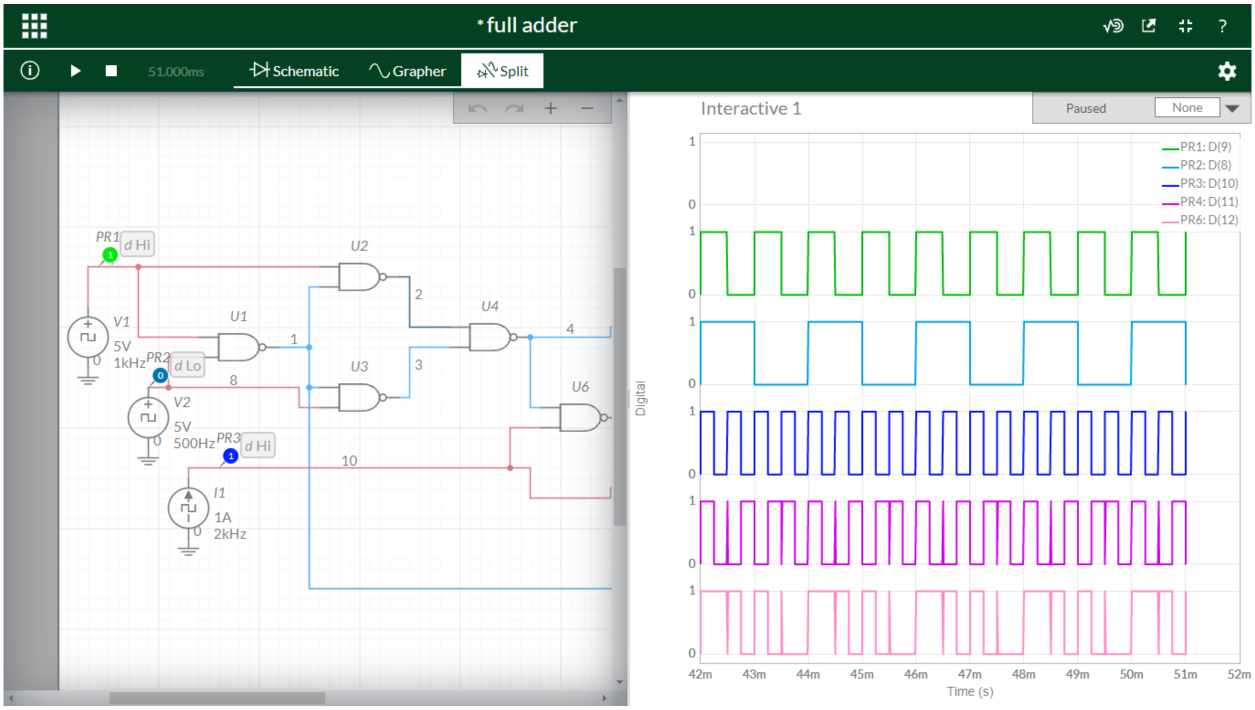
HALF ADDER: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



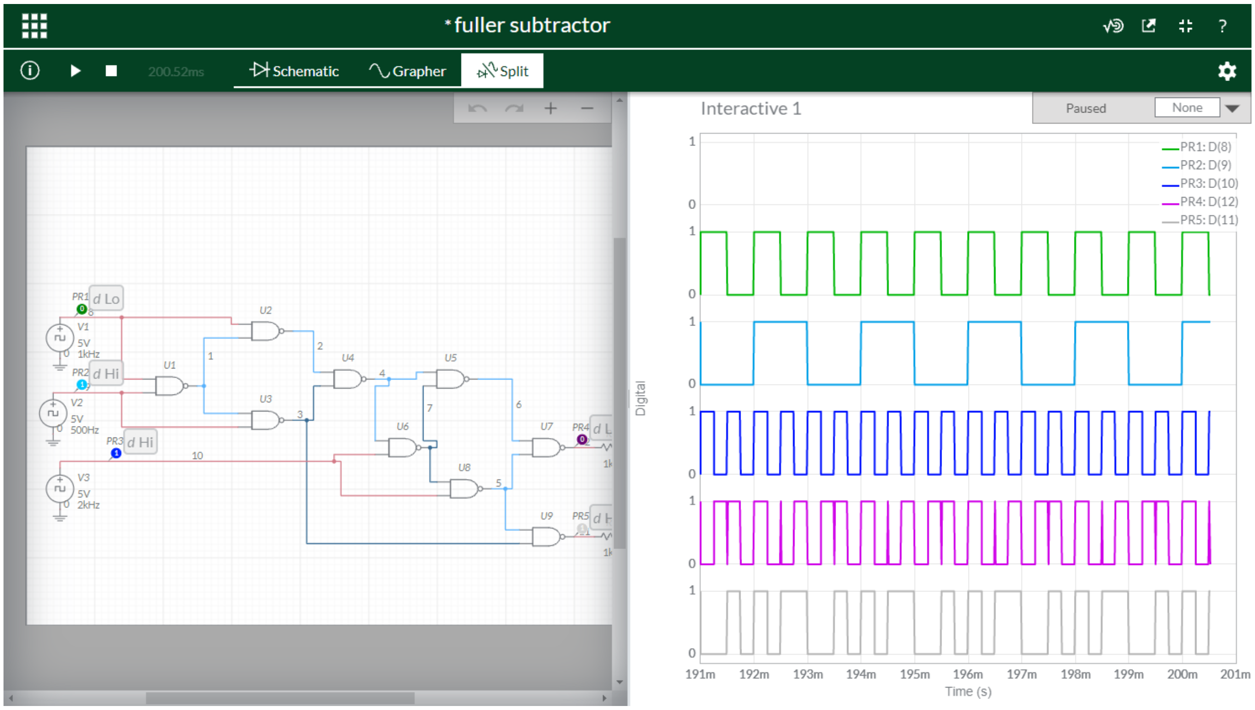
HALF SUBTRACTOR: CIRCUIT/CONNECTION DIAGRAMS (MULTISIM)



FULL ADDER: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

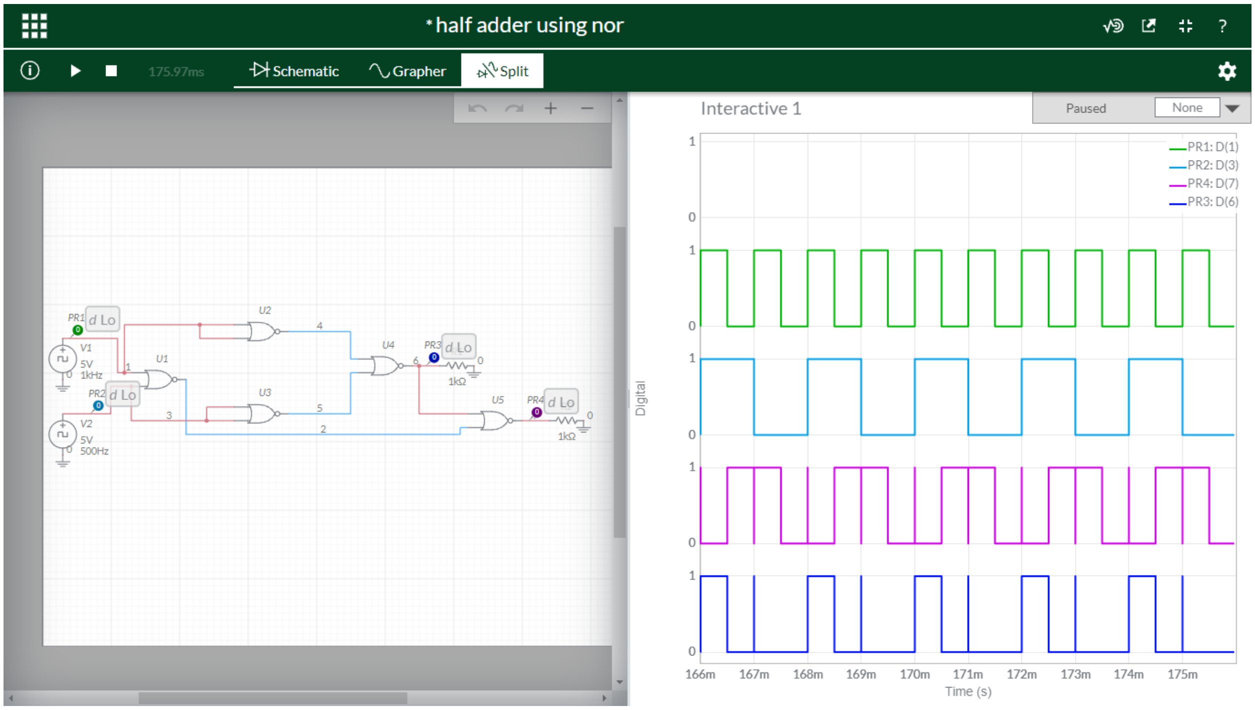


FULL SUBTRACTOR: CIRCUIT/CONNECTION DIAGRAMS (MULTISIM)

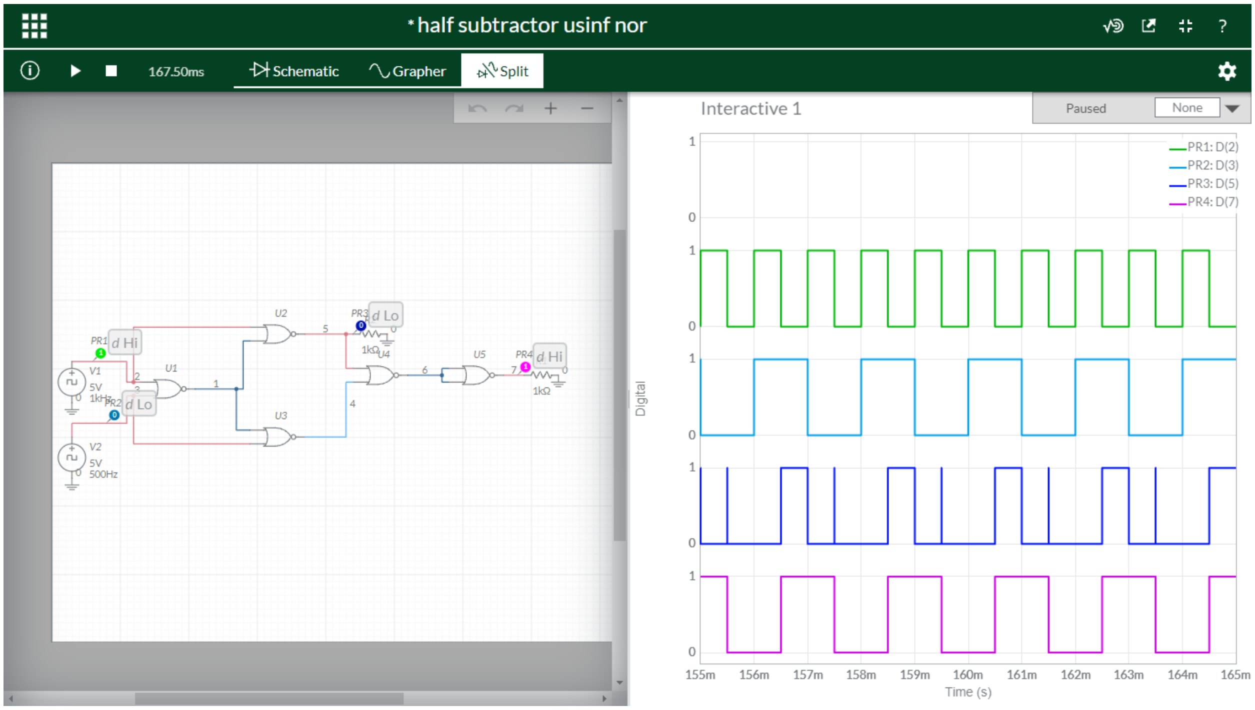


ASSGNMENT QUESTIONS:

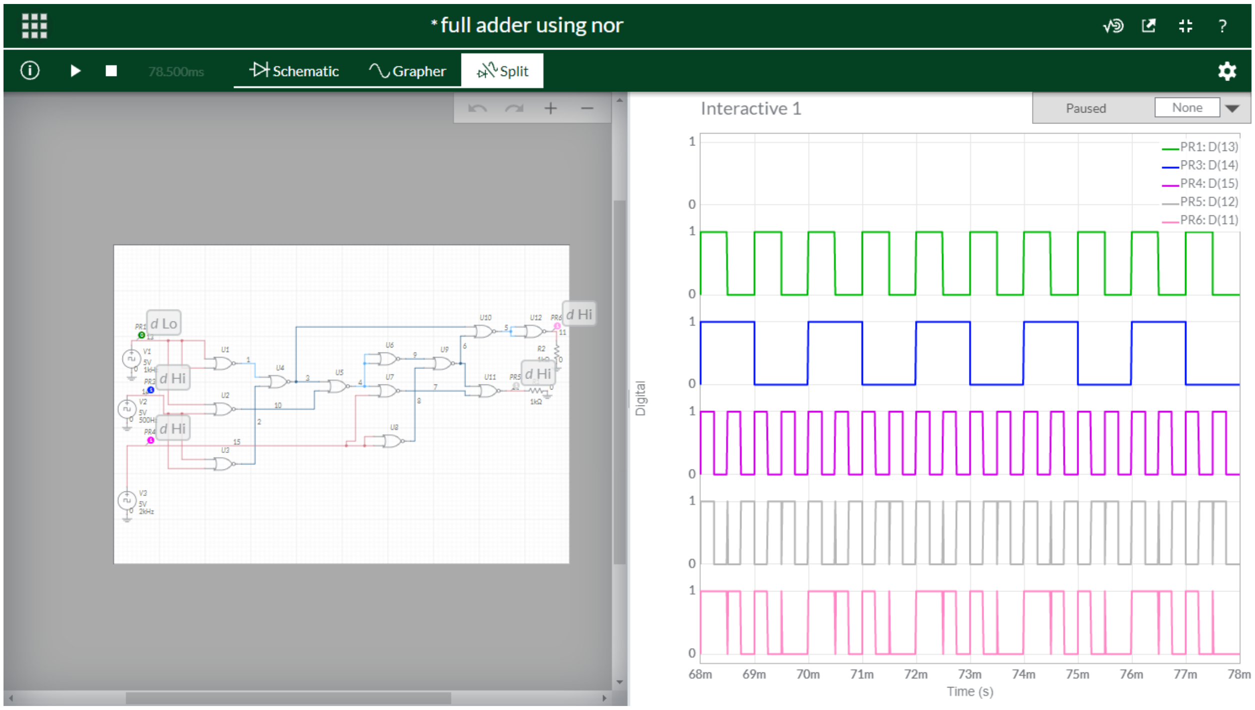
HALF ADDER USING NOR: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



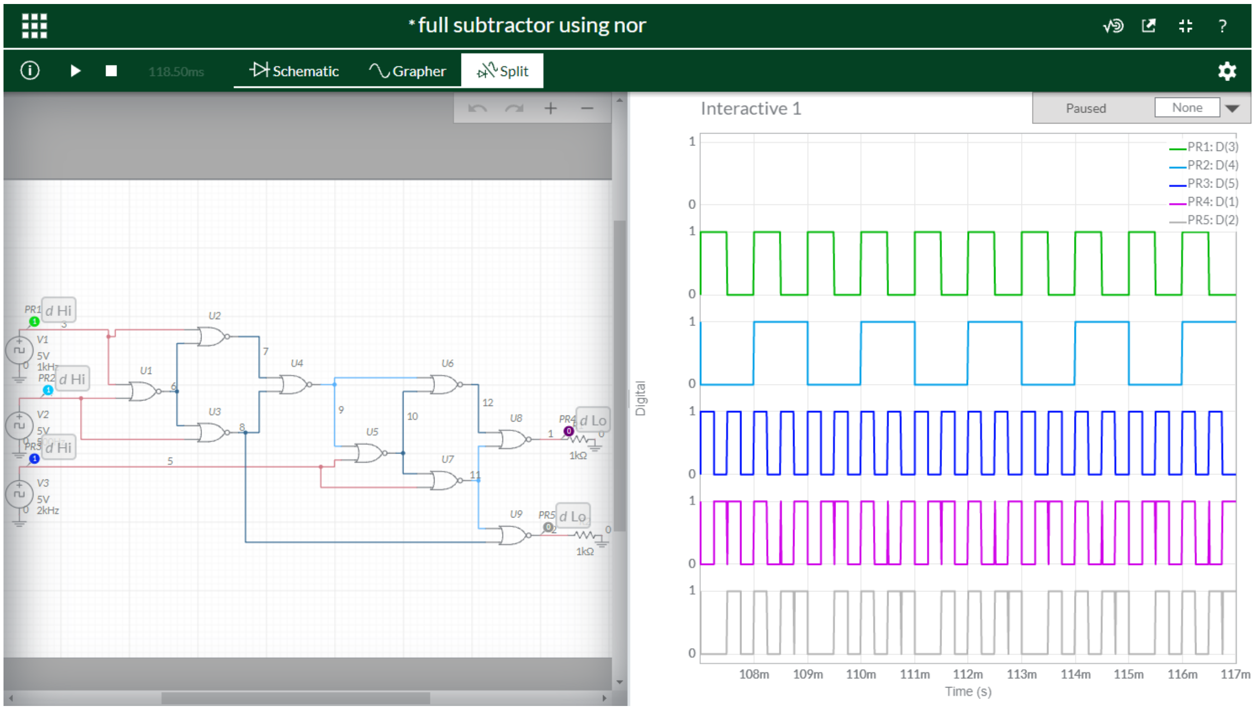
HALF SUBTRACTOR USING NOR: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



FULL ADDER USING NOR: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



FULL SUBTRACTOR USING NOR: CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



CONCLUSIONS

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