DIGITAL ELECTRONICS AND LOGIC DESIGN [EC - 207]

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT ELECTRONICS ENGINEERING DEPARTMENT

Half Adder/Subtractor and Full Adder/Subtractor

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

SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator ,
2. Logic Gates (NAND AND NOR)

THEORY:

In the combinational circuits, different logic gates are used to design encoder, multiplexer, decoder & de-multiplexer. These circuits have some characteristics like the output of this circuit mainly depends on the levels which are there at input terminals at any time. This circuit doesn’t include any memory. The earlier state of the input doesn’t have any influence on the current state of this circuit. The inputs and outputs of a combinational circuit are ‘n’ no. of inputs & ‘m’ no. of outputs. Some of the combinational circuits are half adder and full adder, subtractor, encoder, decoder, multiplexer, and demultiplexer.

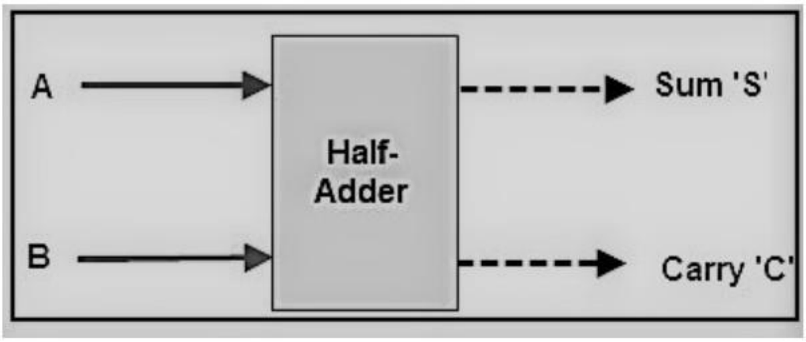
- What is an Adder?

An adder is a digital logic circuit in electronics that is extensively used for the addition of numbers. In many computers and other types of processors, adders are even used to calculate addresses and related activities and calculate table indices in the ALU and even utilized in other parts of the processors.

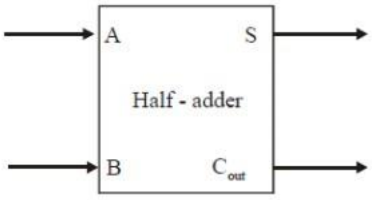
Adders are basically classified into two types: Half Adder and Full Adder.

- **Half Adder :-**

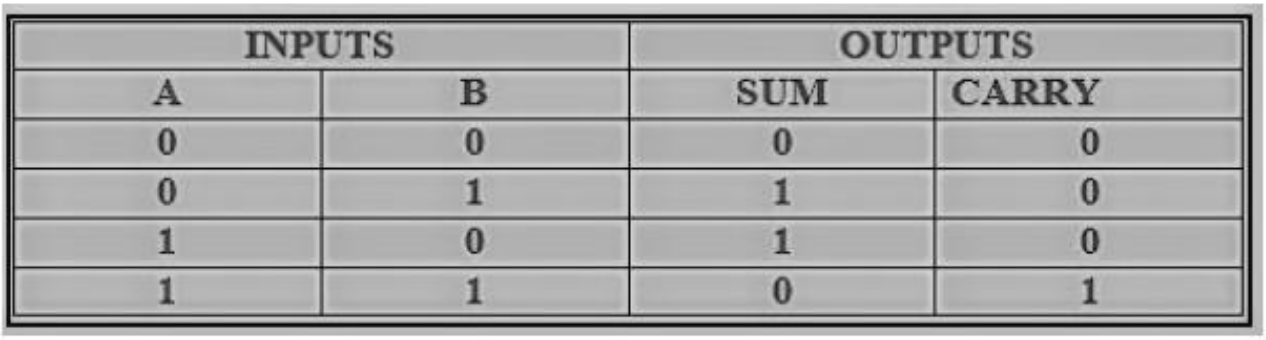
It adds two binary digits where the input bits are termed as augend and addend and the result will be two outputs one is the sum and the other is carry. To perform the sum operation, XOR is applied to both the inputs, and AND gate is applied to both inputs to produce carry.



By using a half adder, you can design simple addition with the help of logic gates.



(*a*) logic symbol



INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half Adder Truth Table

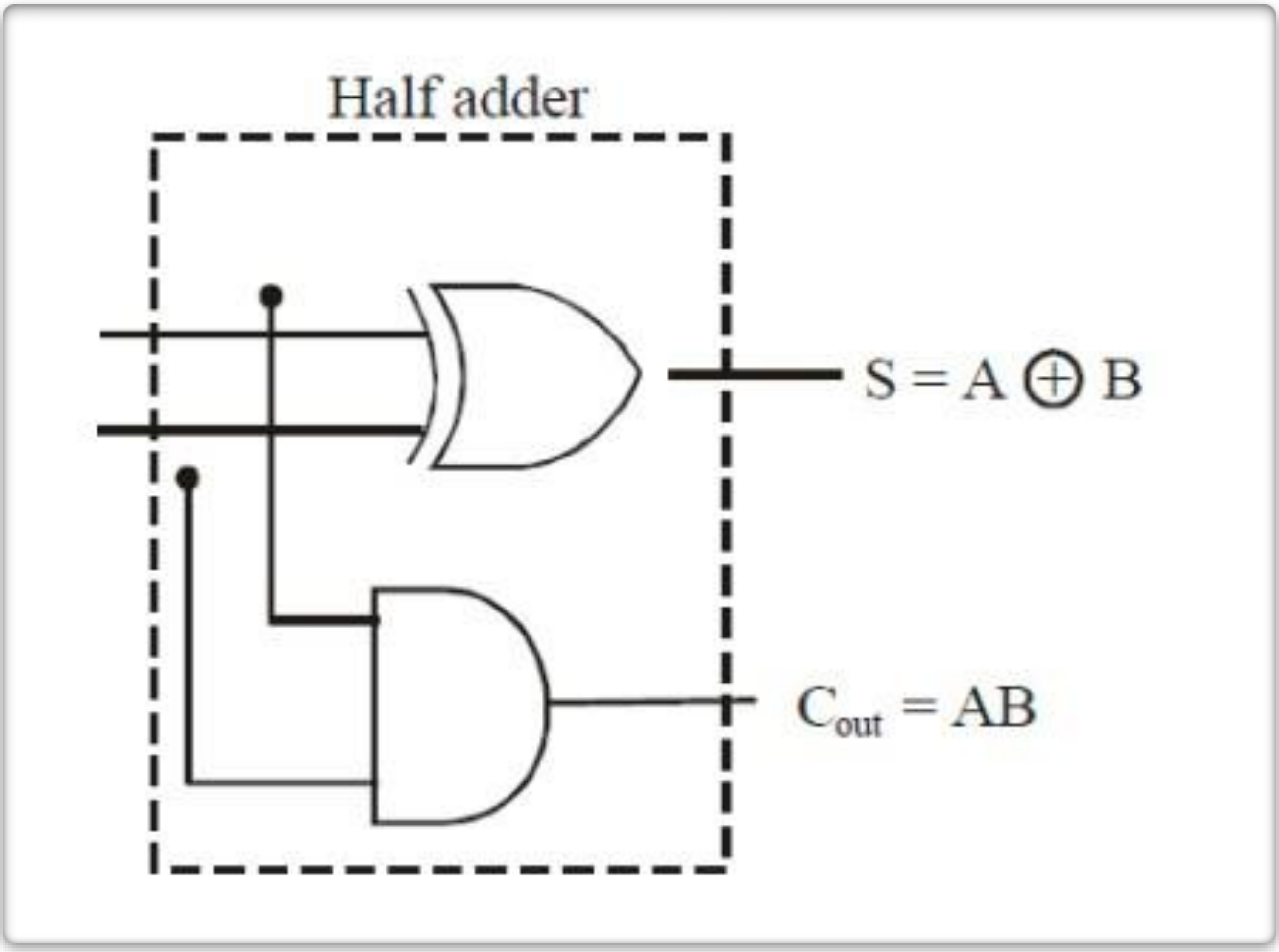
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

These are the least possible single-bit combinations. But the result for 1+1 is 10, the sum result must be re-written as a 2-bit output. Thus, the equations can be written as

0 + 0 = 00
0 + 1 = 01
1 + 0 = 01
1 + 1 = 10

The output ‘1’of ‘10’ is carry-out. ‘SUM’ is the normal output and ‘CARRY’ is the carry-out. The half-adder is useful when you want to add one binary digit quantities.

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

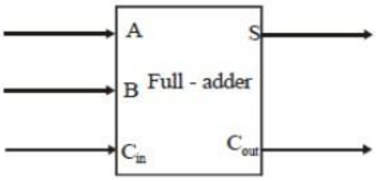


And an equivalent expression in terms of the basic AND, OR, and NOT is:

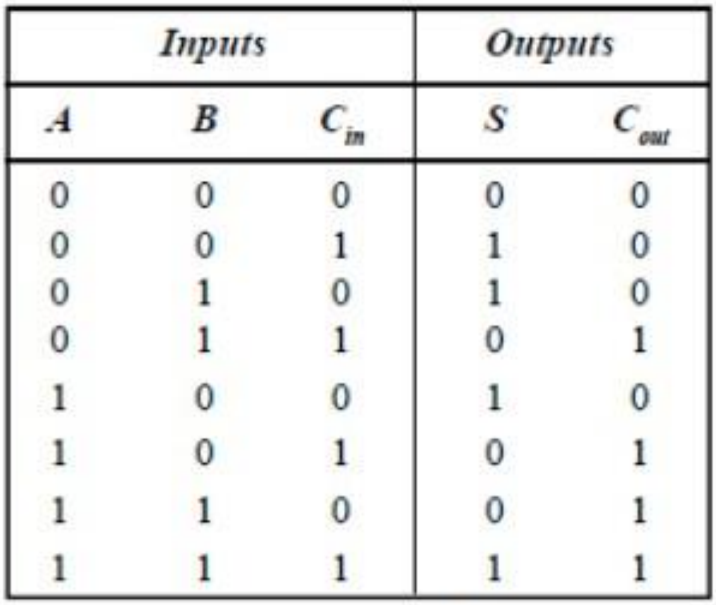
SUM=A.B+A.B’

FULL ADDER:

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.



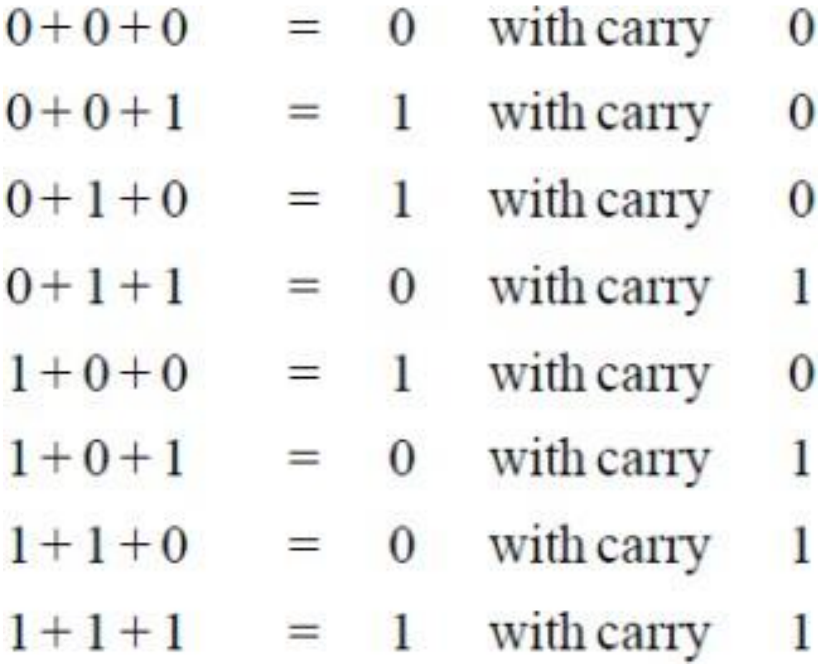
(*a*) logic symbol



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

(*b*) truth table

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



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

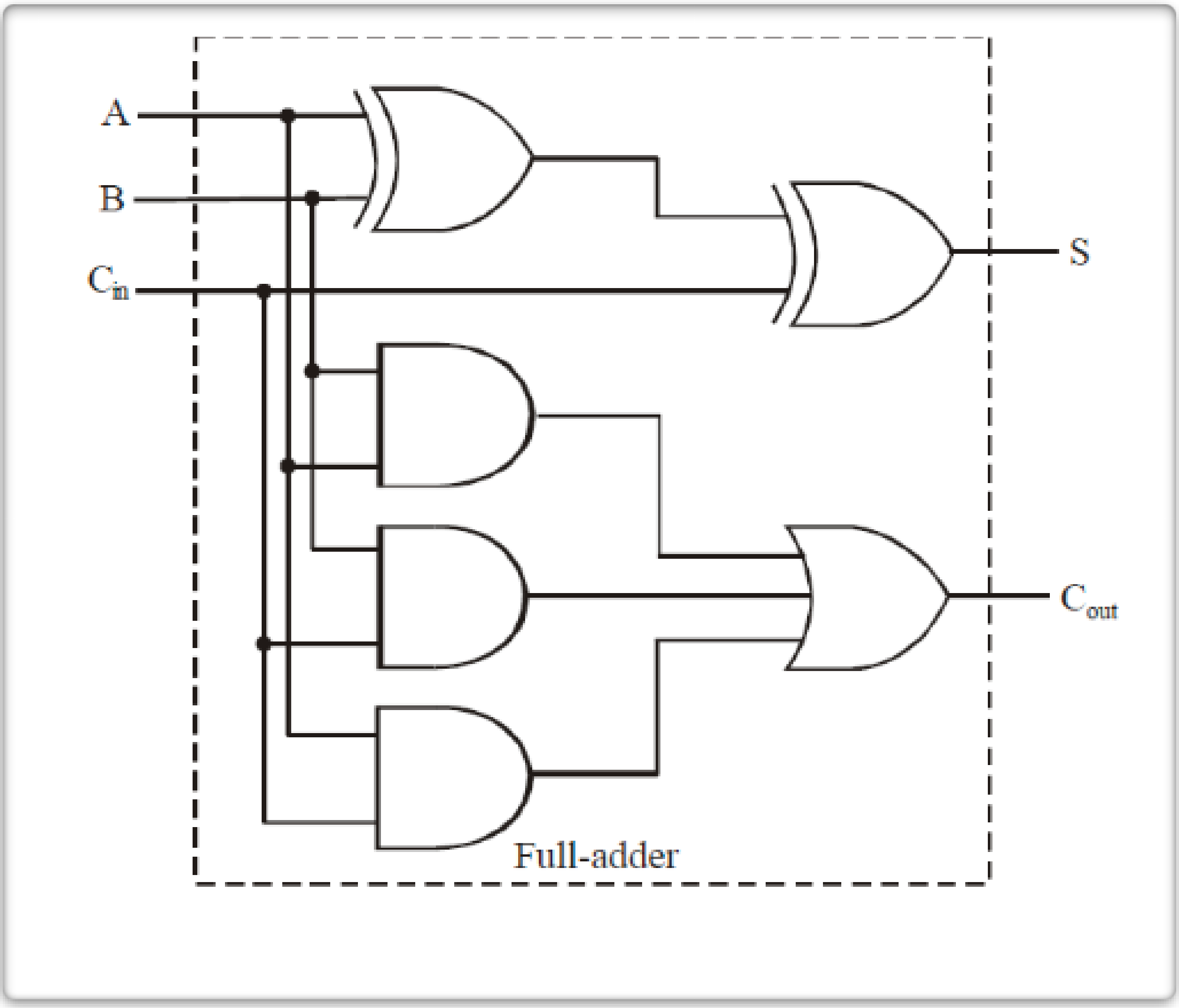
The Boolean expression for the sum output (S) can be obtained from the above truth table by summing and then simplifying the terms for which S=1. Thus, the sum is

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

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

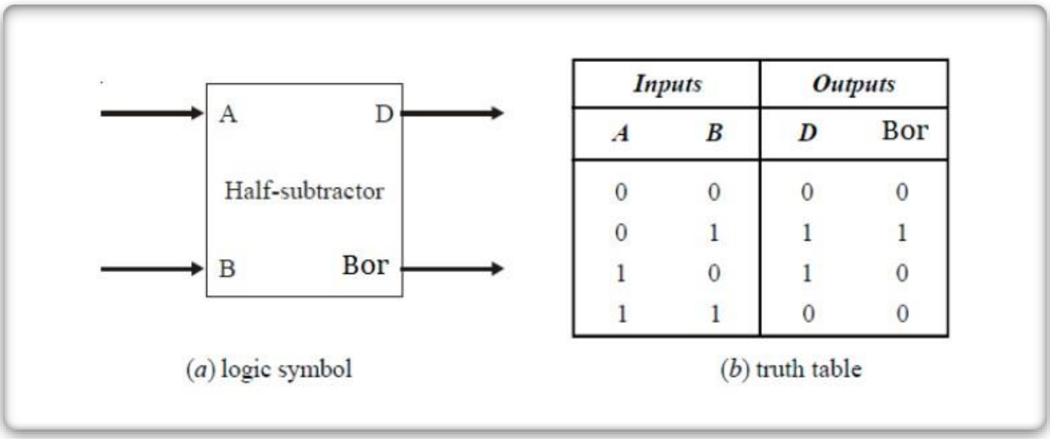


• HALF SUBTRACTOR :-

Half subtractor is the most essential combinational logic circuit which is used in digital electronics. Basically, this is an electronic device or in other terms, we can say it as a logic circuit. This circuit is used to perform two binary digits subtraction.Similarly, the subtractor circuit uses binary numbers (0,1) for the subtraction. The circuit of the half subtractor can be built with two logic gates namely NAND and EX-OR gates. This circuit gives two elements such as the difference as well as they borrow.

As in binary subtraction, the major digit is 1, we can generate borrow while the subtrahend 1 is superior to minuend 0 and due to this, borrow will need. The following example gives the binary subtraction of two binary bits.

As in binary subtraction, the major digit is 1, we can generate borrow while the subtrahend 1 is superior to minuend 0 and due to this, borrow will need. The following example gives the binary subtraction of two binary bits.



(*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:

$0 - 0 = 0$

$0 - 1 = 1$ with a borrow of 1

$1 - 0 = 1$

$1 - 1 = 0$

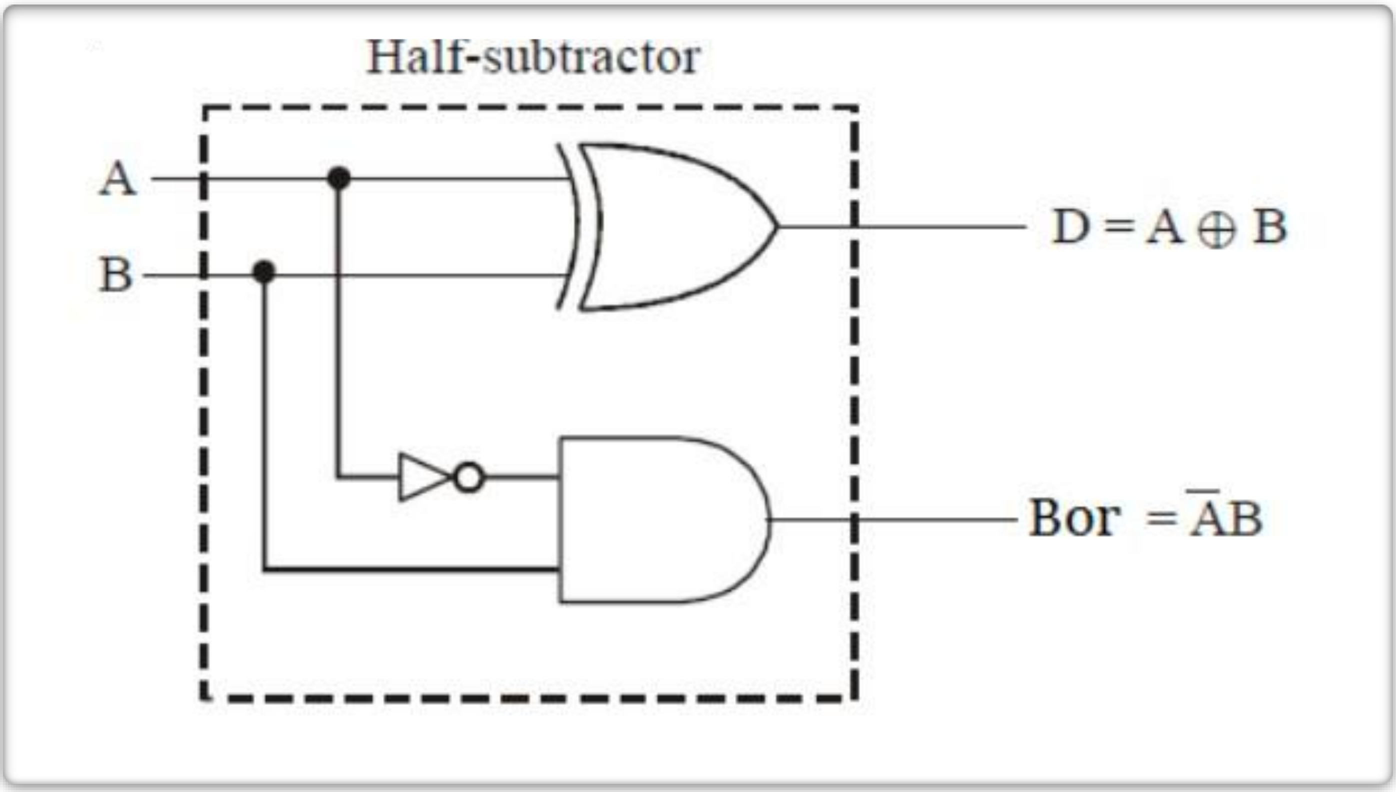
The Boolean expression for the difference bit (D) can be expressed by the equation.

$$D = A\overline{B} + \overline{A}B$$
$$= A \oplus B$$

and the Boolean expression for the borrow bit,

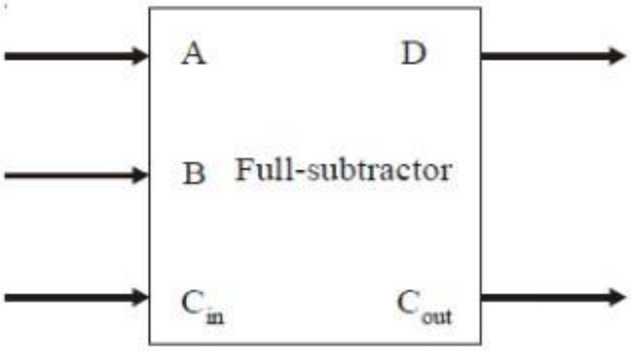
$$\text{Bor} = \overline{A}B$$

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 exclusive- OR gate while the borrow bit (Bor) is the output of a two-input AND gate.

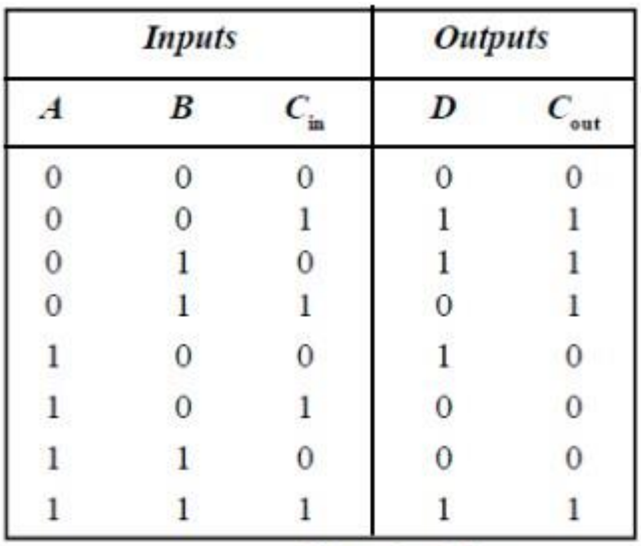


• FULL SUBTRACTOR:

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:

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

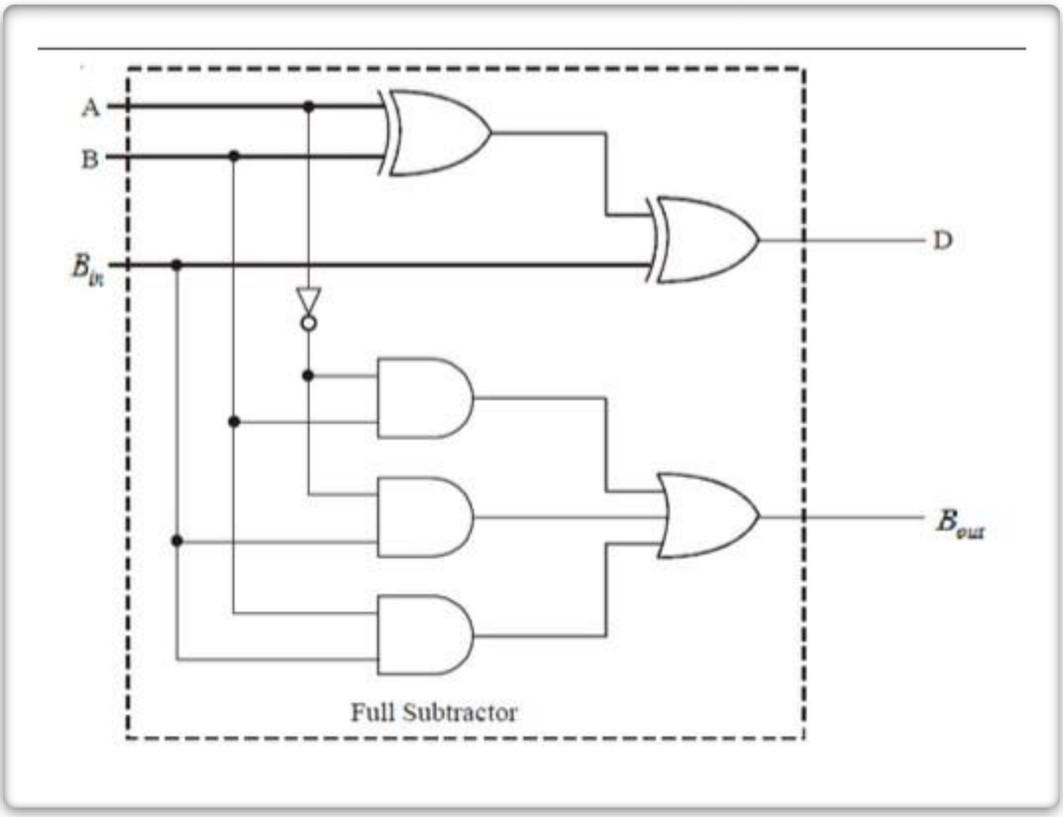
The Boolean expression for the difference bit (D) can be obtained by summing and simplifying all the input combinations from the truth table which have 1 in the corresponding difference column. The final simplified expression for difference is given by

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

and, the Boolean expression for the borrow bit,

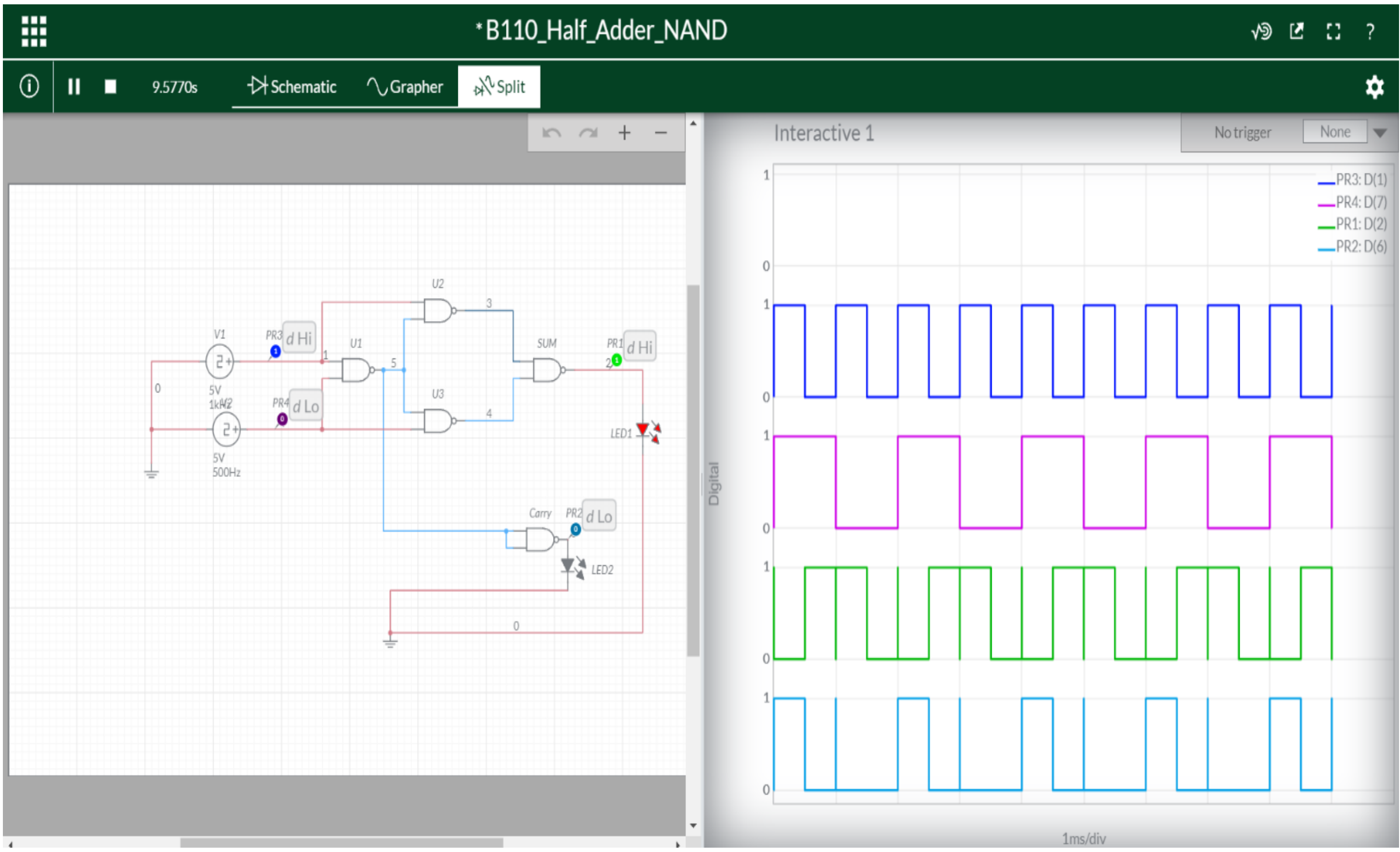
$$B_{out} = \overline{A}B + BB_{in} + \overline{A}B_{in}$$

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.

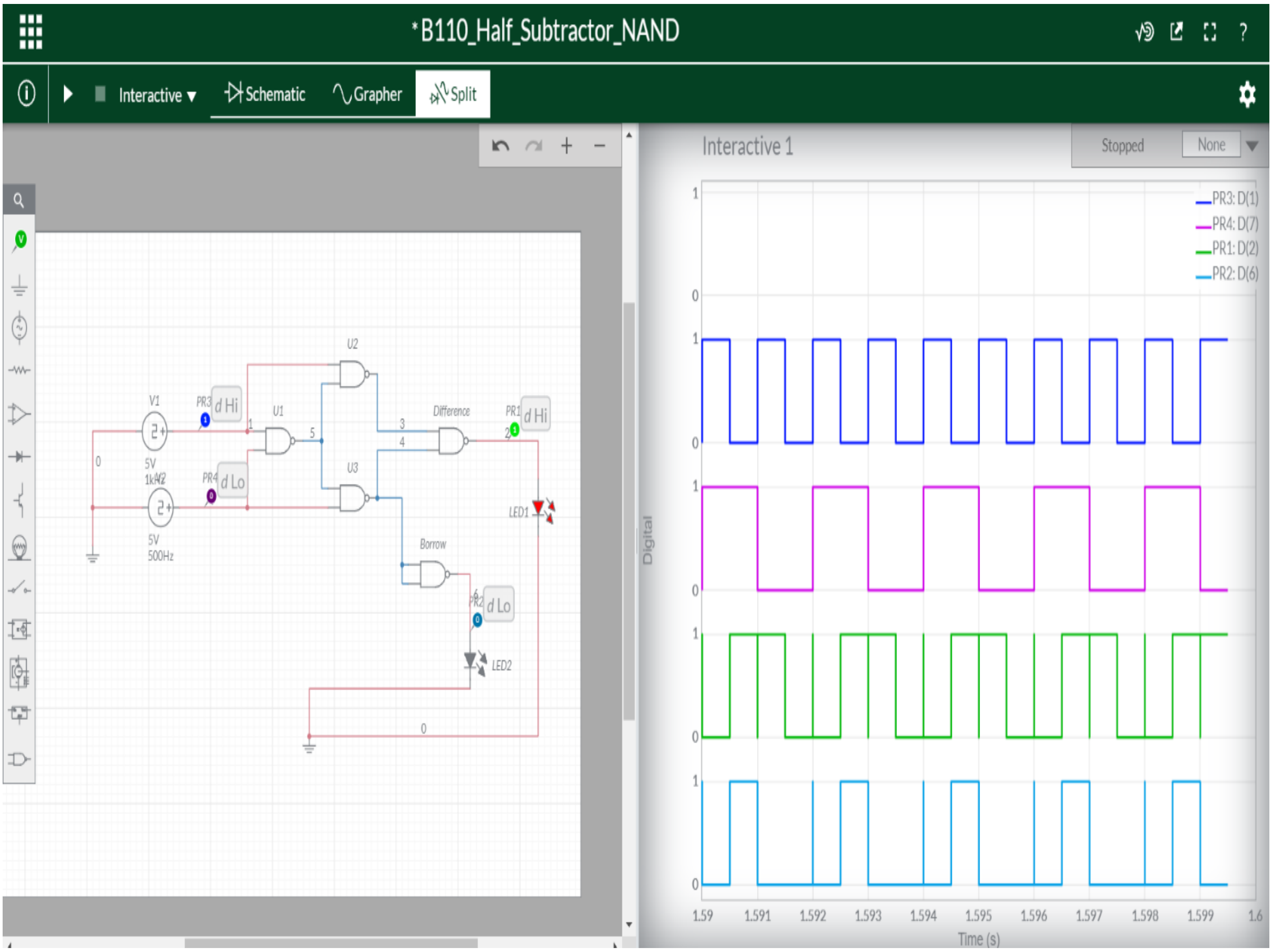


CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM) ALONG WAVEFORMS (FROM MULTISIM)

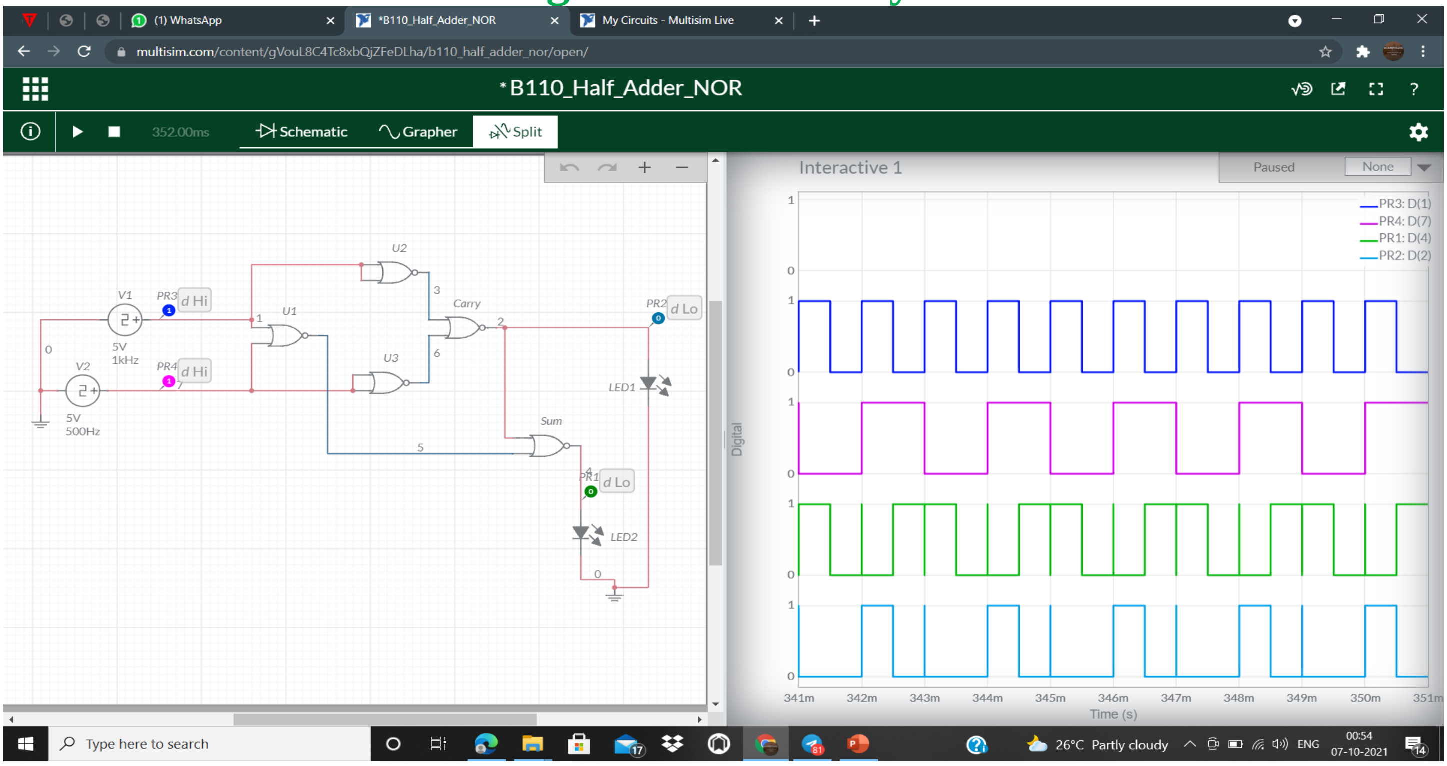
1) HALF ADDER USING NAND GATE ONLY



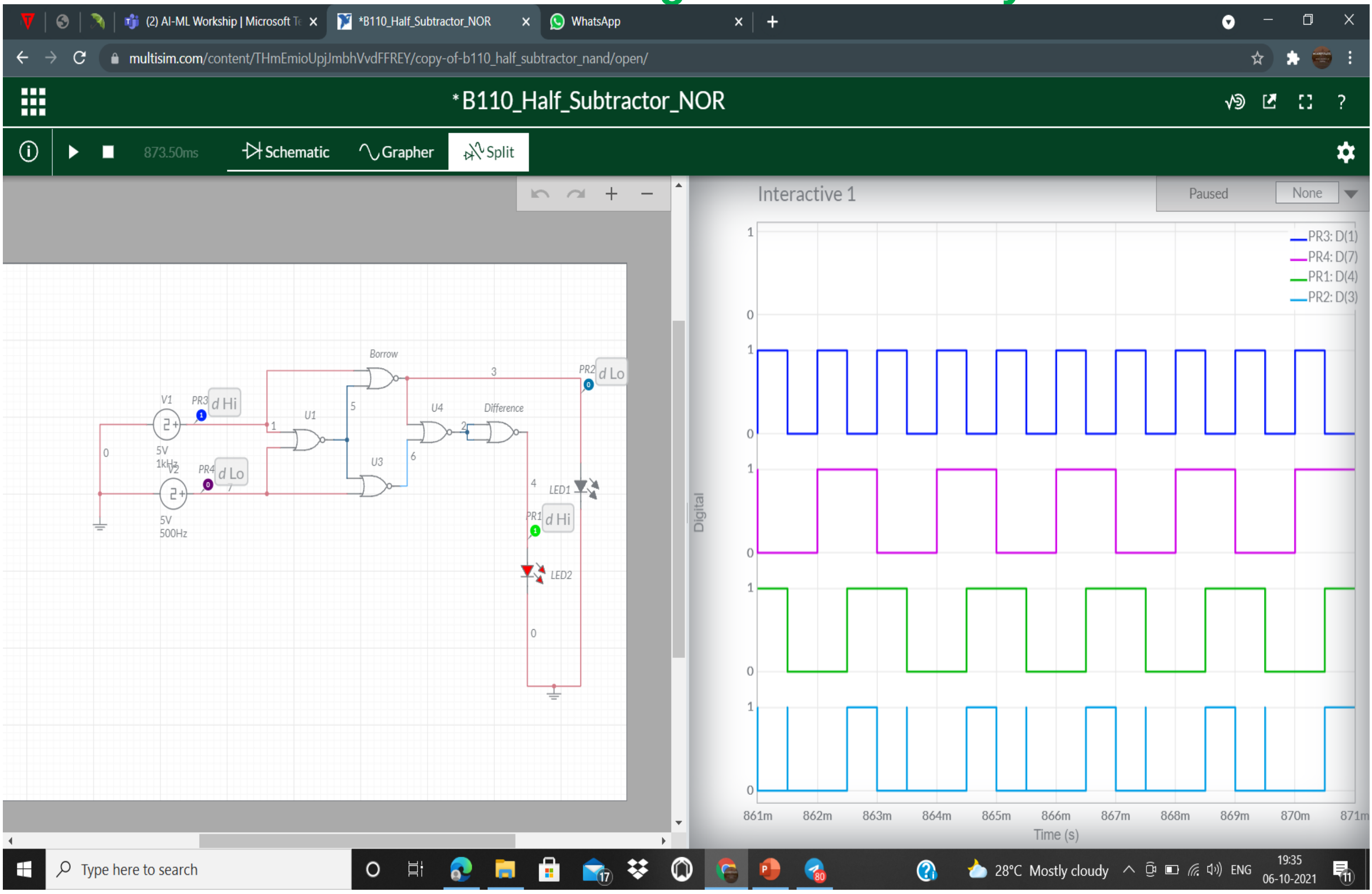
Half Subtractor Using NAND Gate Only



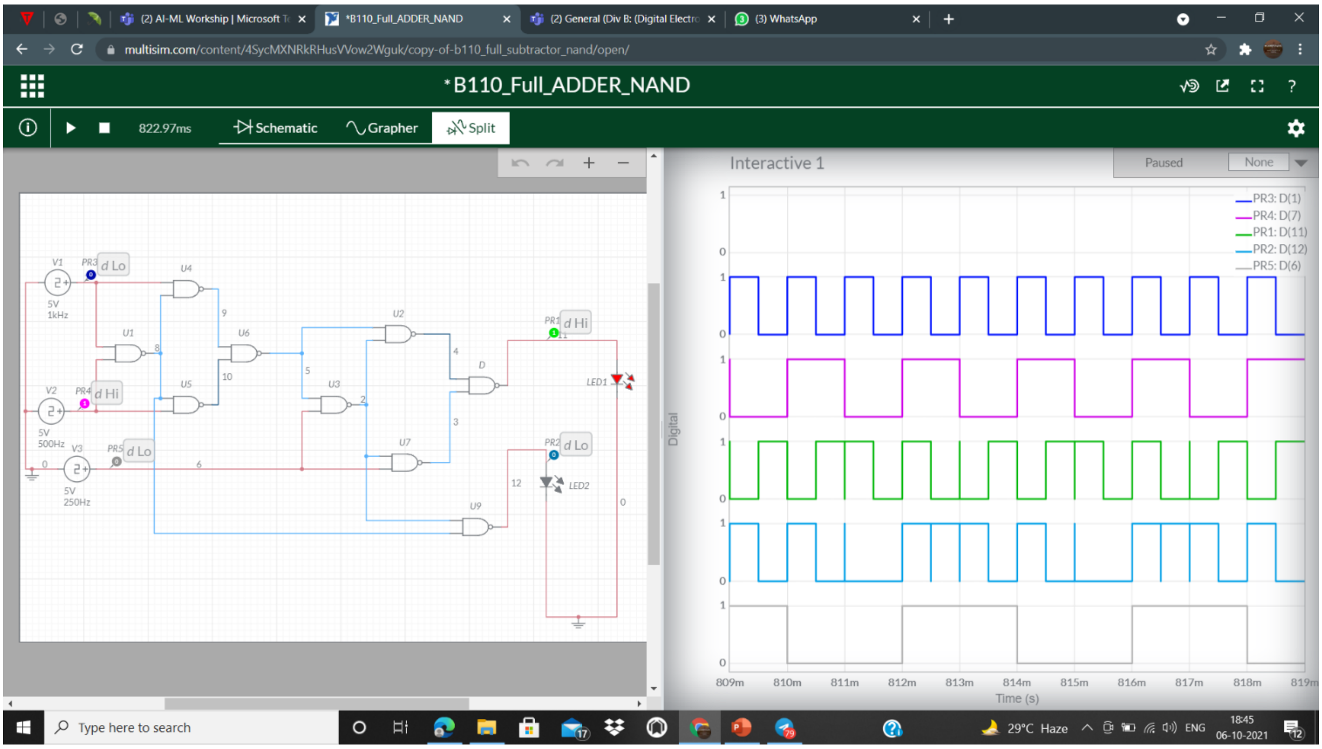
Half Adder Using NOR Gate Only



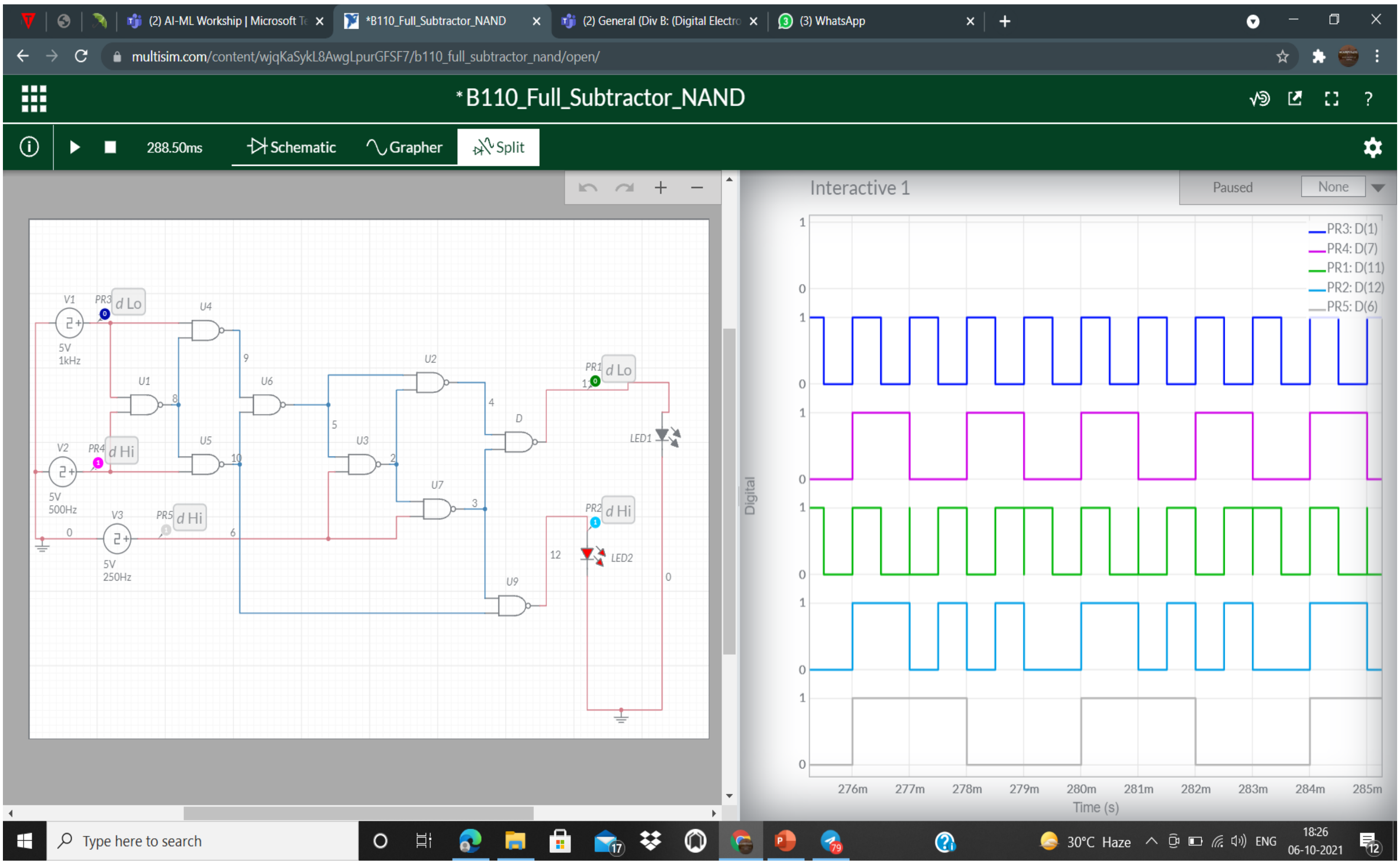
Half Subtractor Using NOR Gate Only



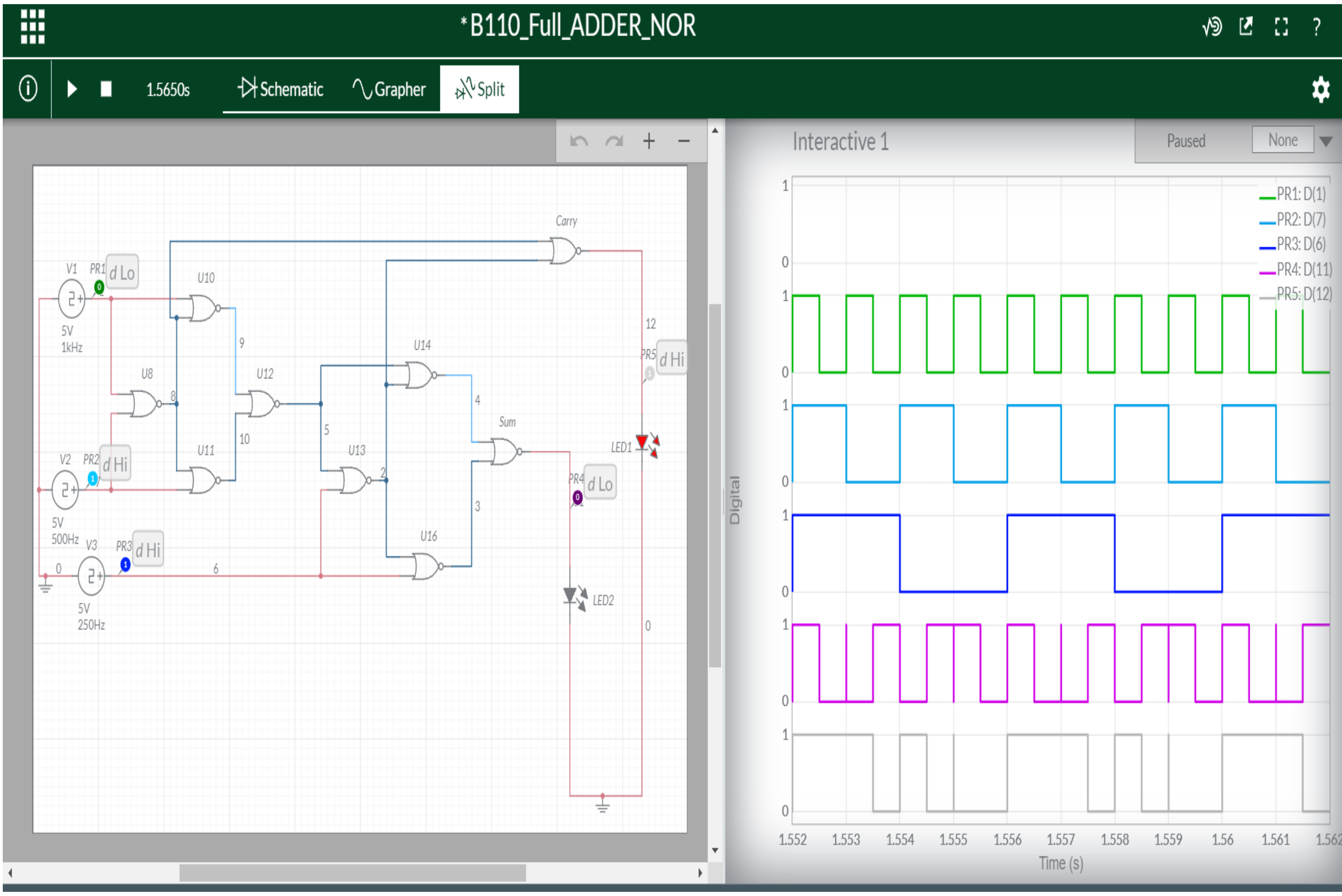
FULL ADDER USING NAND GATE ONLY



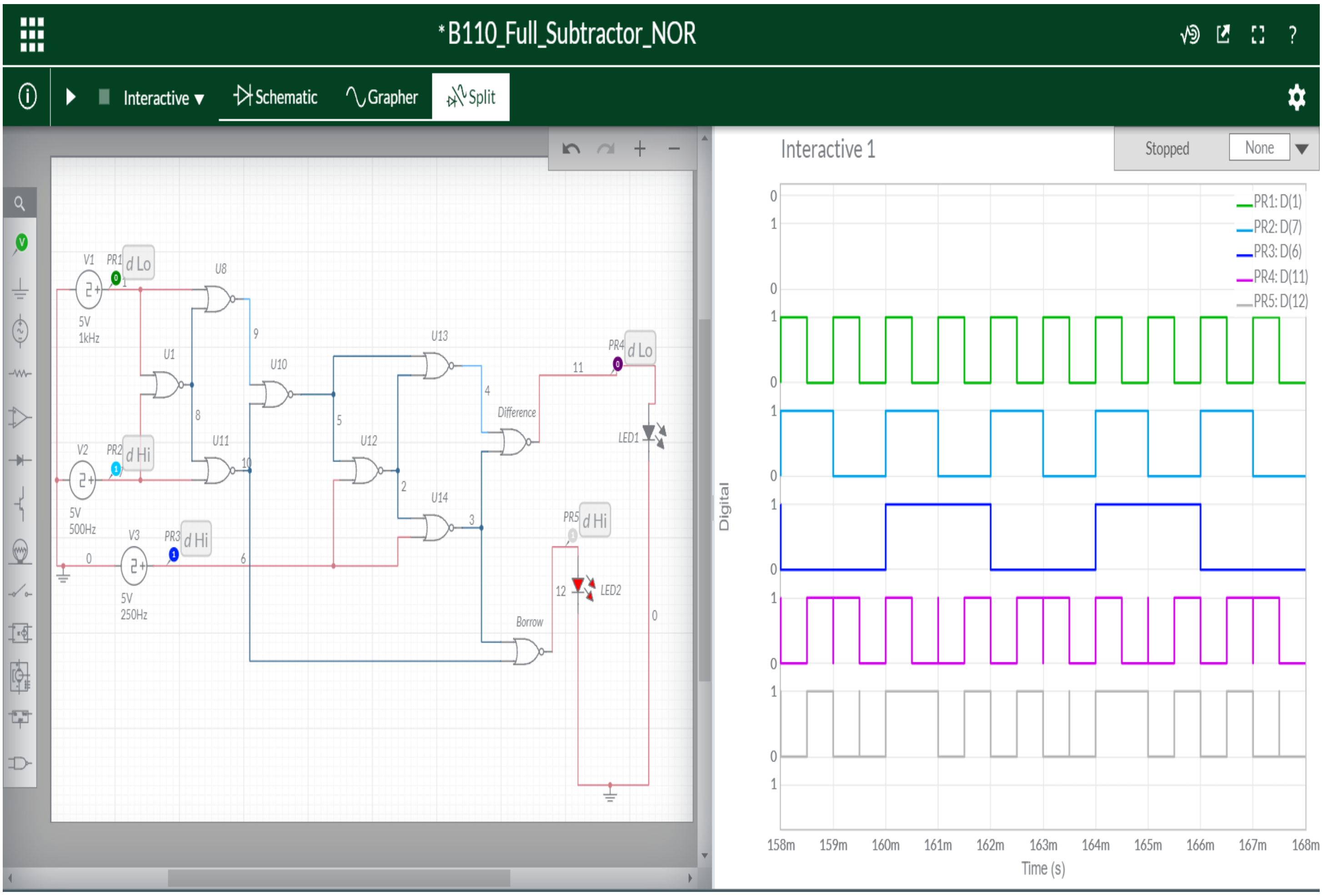
FULL SUBTRACTOR USING NAND GATE ONLY



FULL ADDER USING NOR GATE ONLY



Full Subtractor Using NOR Gate Only



• CONCLUSIONS: -

THE TRUTH TABLE IN THEORY AND THE SIMULATION OF THE HALF/FULL ADDER AND HALF/FULL SUBTRACTOR CIRCUIT USING NAND GATE ONLY AND NOR GATE ONLY ON MULTISIM LIVE BOTH ARE EQUAL. HENCE VERIFIED.