Circuit Simulation Project

https://esim.fossee.in/circuit-simulation-project

Name of the Participant: Preeti Pallavi (20BCE1828)
Project Guide: Dr R. Maheswari

Title: Design of XOR gate using only NOR gates

Theory:

XOR gate is the Exclusive OR gate. This is not frequently used as inclusive OR gate, which is the regular OR gate. XOR represents the inequality function, i.e., the output is true if the inputs are not alike otherwise the output is false. A way to remember XOR is "must have one or the other but not both".

XOR can also be viewed as addition modulo 2. As a result, XOR gates are used to implement binary addition in computers. Other uses include subtractors, comparators, and controlled inverters.

An XOR gate can be designed by using basic logic gates like NAND gate and NOR gate as they are universal gates. An XOR gate is made by considering the conjunctive normal form, noting from De Morgan's Law that a NOR gate is an inverted-input AND gate. This construction entails a propagation delay three times that of a single NOR gate and uses five gates.

The truth table for XOR gate is as follows:

XOR Truth Table		
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Using the above truth table, we can now derive the Boolean Expression for XOR Gate. If A and B are the inputs of the XOR gate, its output is given as:

$$A' B + A B'$$

The XOR output is represented as: $\mathbf{A \oplus B}$
It can also be written as:

$$(A + B) (A' + B')$$

By applying De Morgan's law, the above Boolean expression can also be written as:

$$(A + B) (A B)'$$

The output of 2 input XOR gate is HIGH only when one of its inputs are HIGH. If both the inputs are same, then the output is LOW.

Using NOR Gates

To implement the XOR Gate using NOR Gates, the above XOR Boolean Equation can be re-written as:

$$Q = A' B + A B'$$

$$Q = A' B + A B' + A A' + B B'$$

$$Q = (A' + B') (A + B)$$

Taking complement on both sides, we get:

$$Q' = ((A' + B') (A + B))'$$

Using de Morgan's Law, we get:

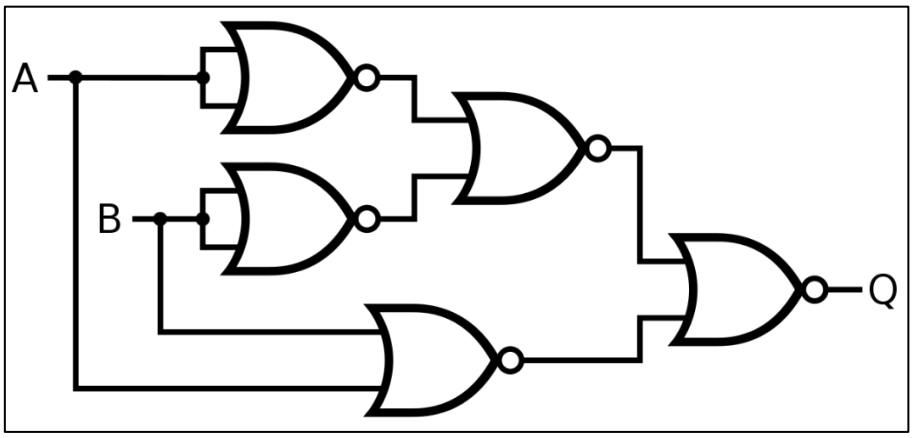
$$Q' = (A' + B')' + (A + B)'$$

Once again taking complement on both sides, we get:

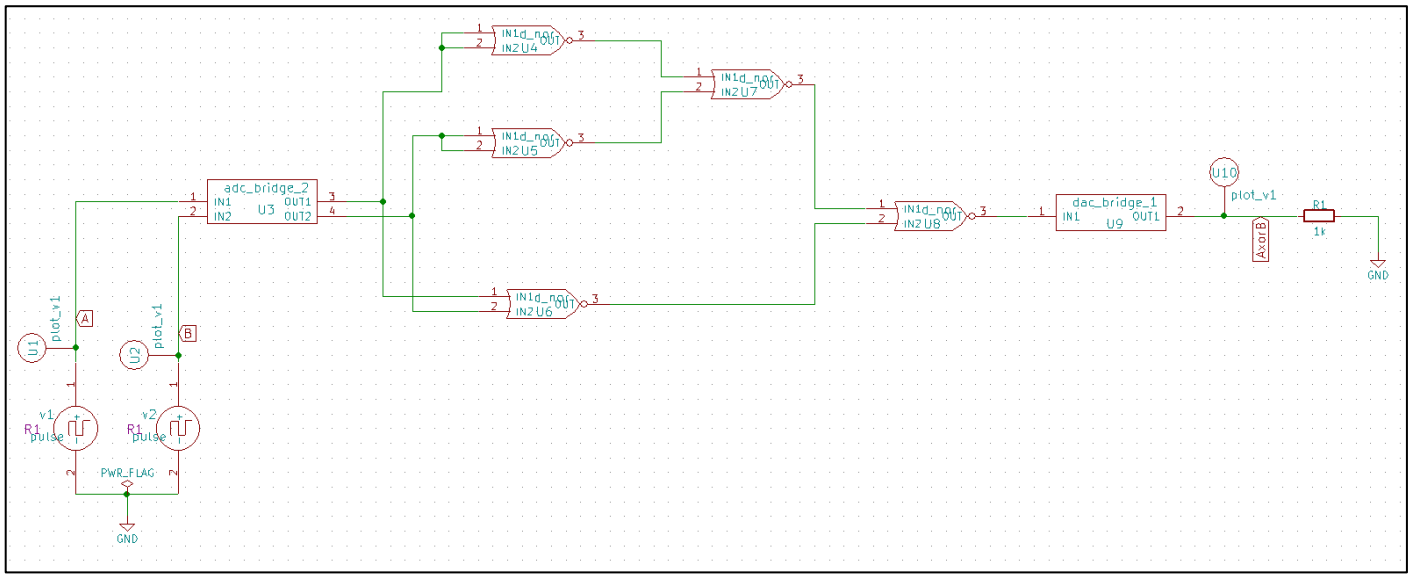
$$Q = ((A' + B')' + (A + B)')'$$

This equation looks like it can be implemented using NOR Gates. We need totally five NOR gates (two for inverting A and B, one for NOR of A and B, one for NOR of A' and

B’ and the final one to obtain the above equation). The following image shows the XOR Gate implemented using NOR Gates.

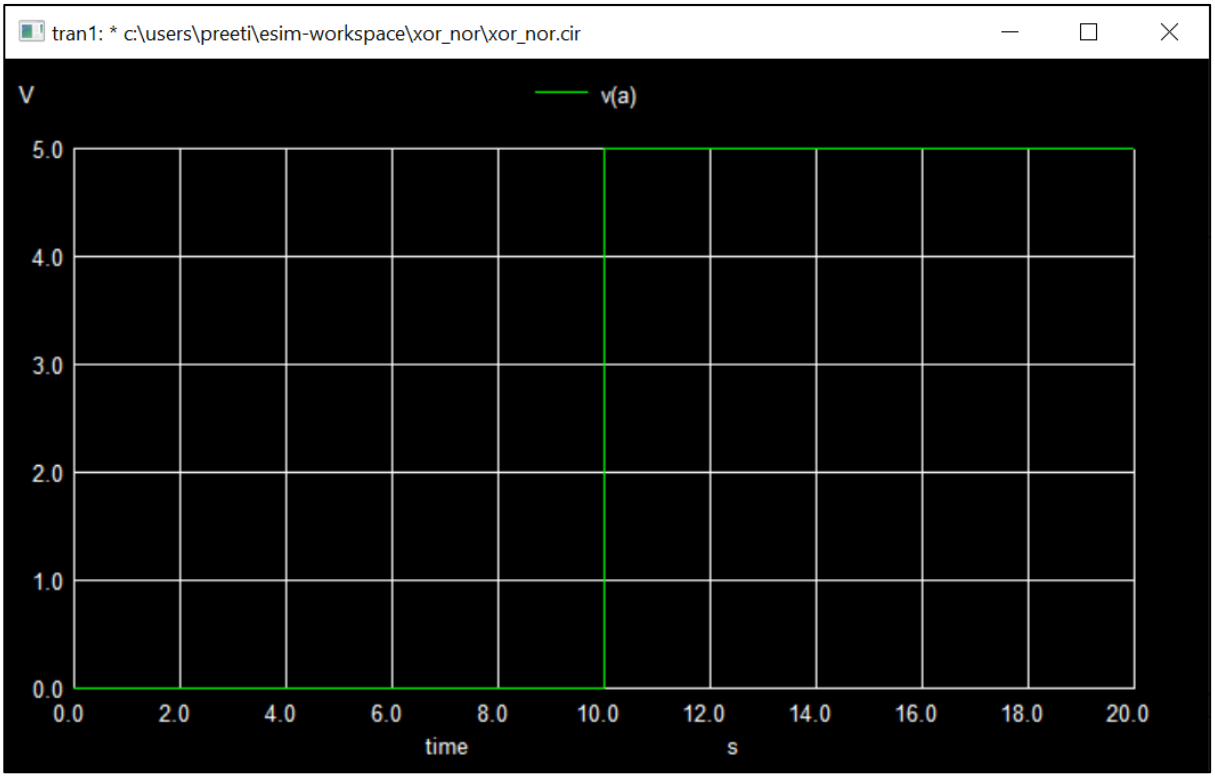


Schematic diagram:

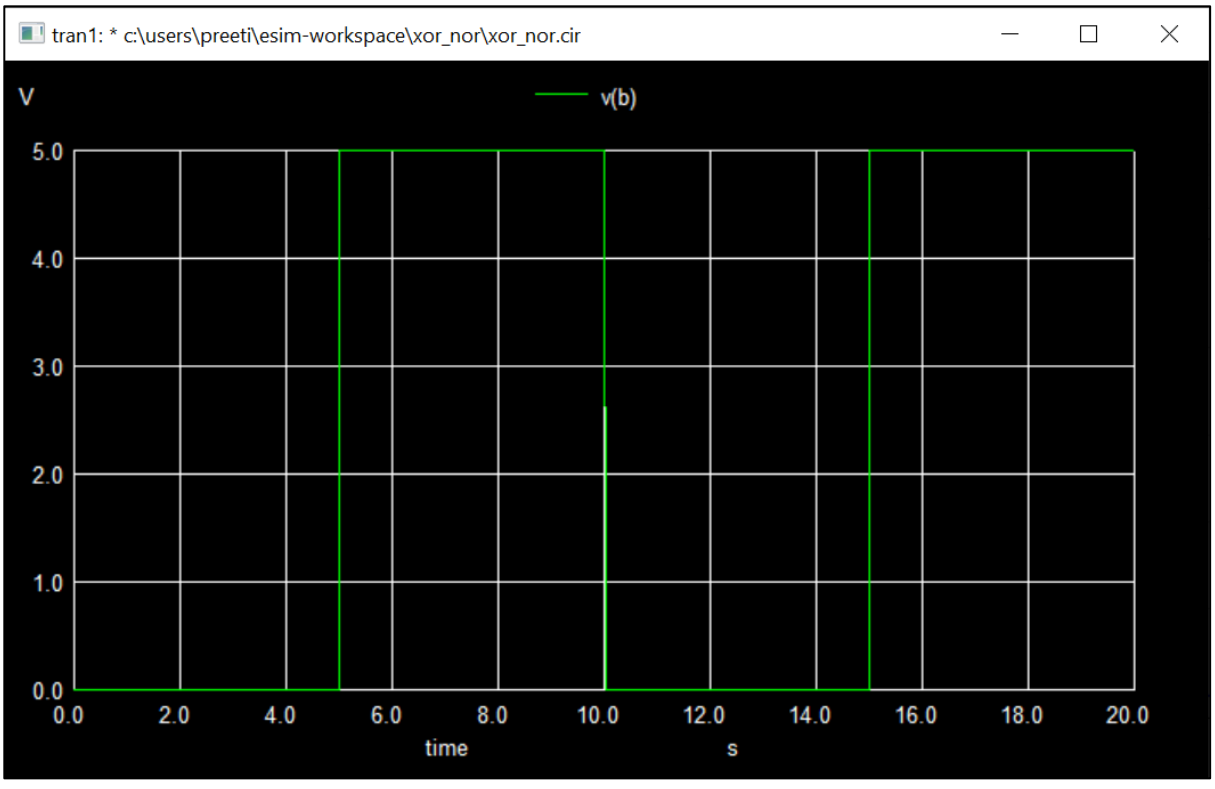


Plots:

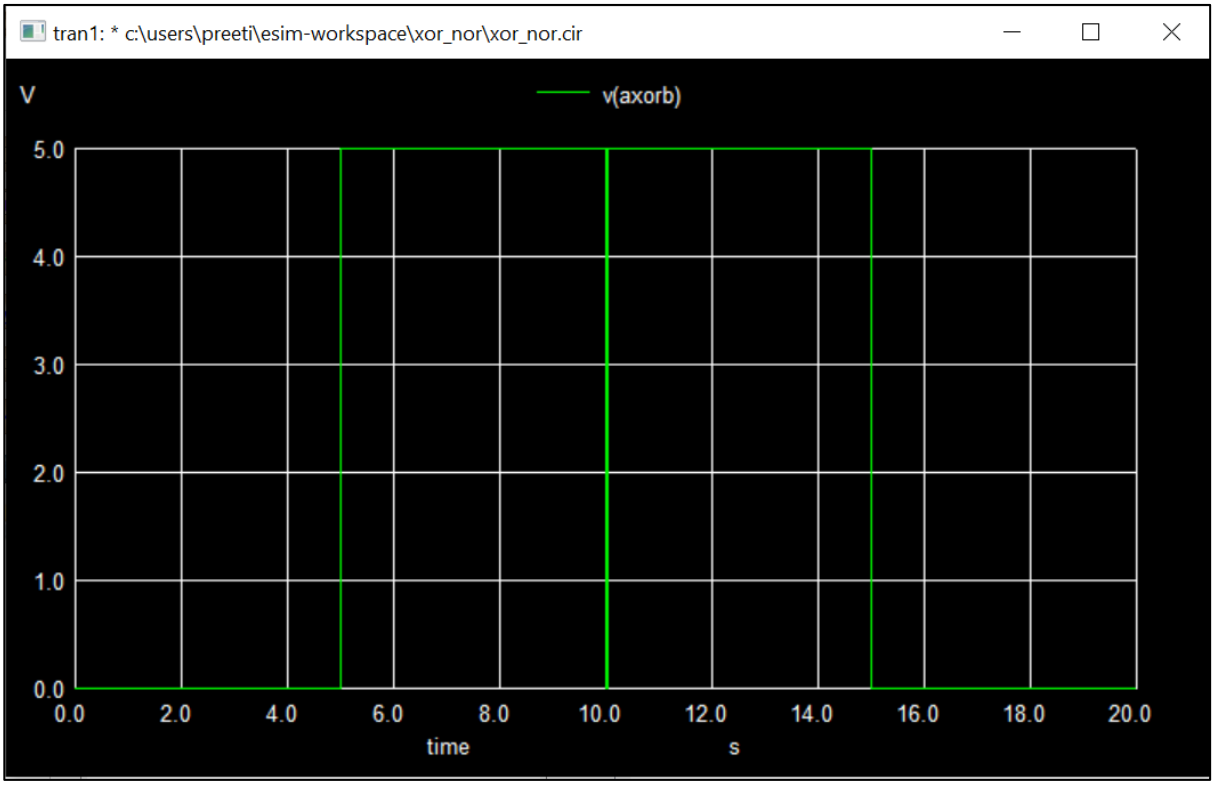
A:


tran1: * c:\users\preeti\esim-workspace\xor_nor\xor_nor.cir
V
v(a)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
time
s

B:

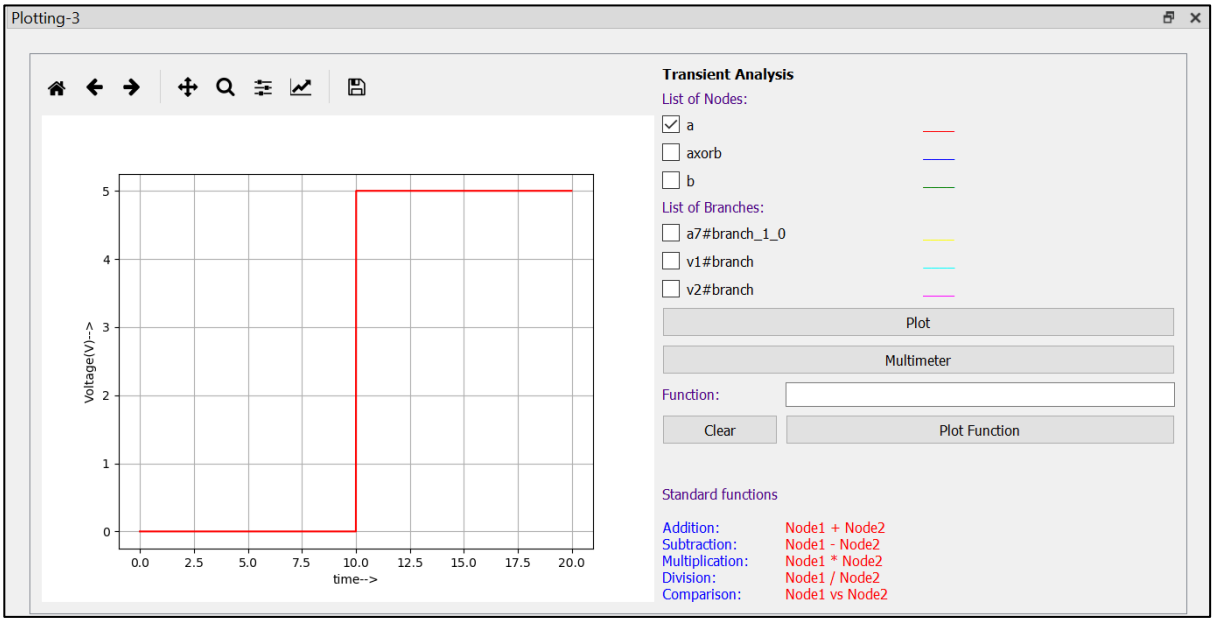

tran1: * c:\users\preeti\esim-workspace\xor_nor\xor_nor.cir
V
v(b)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
time
s

A XOR B:

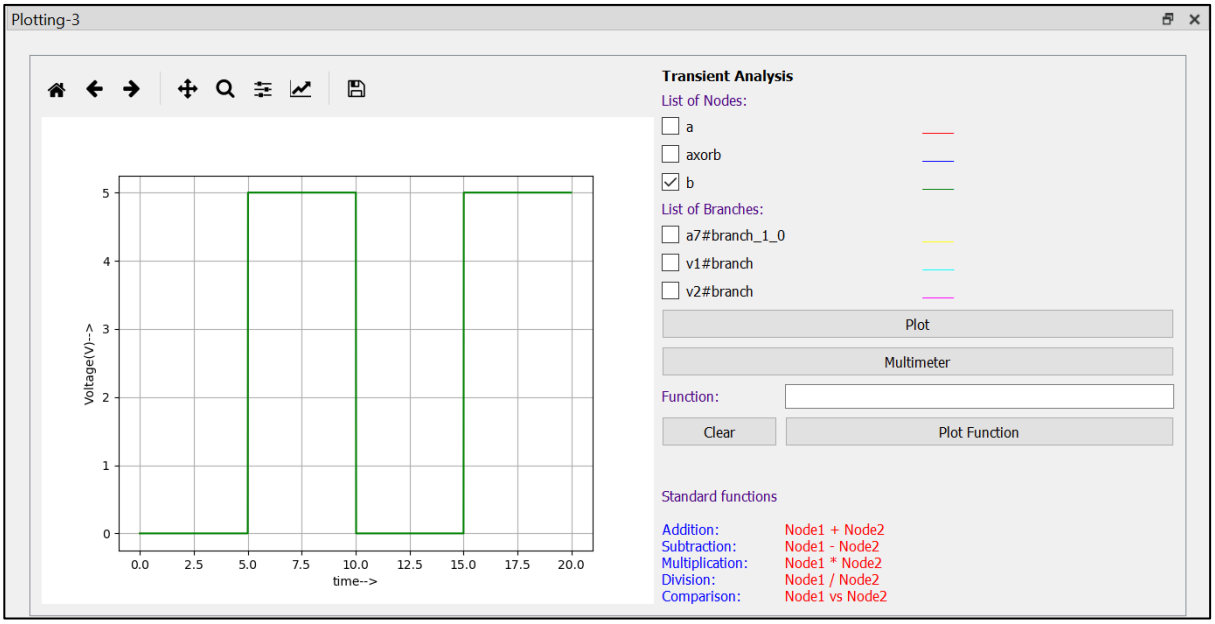


Python Plots:

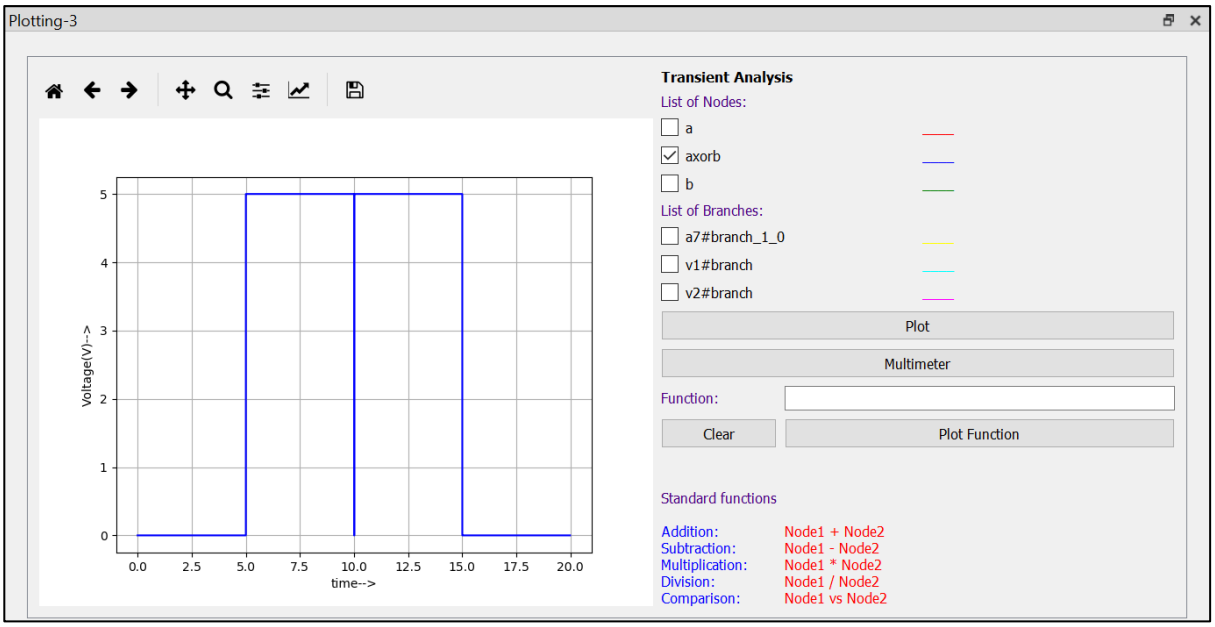
A:



B:



A XOR B:



Conclusion:

Hence, the XOR logic gate was designed using only NOR gates and resulting plots were verified on Esim.

References:

1. https://en.wikipedia.org/wiki/XOR_gate
2. https://www.electronicshub.org/exclusive-or-gatexor-gate/
3. https://en.wikipedia.org/wiki/NOR_logic#XOR