NAME: HRISHIKESH G KULKARNI **DATE: 21/04/2022**

REG.NO.: 21BAI1660

BRANCH: BTECH - CSE AND SPEC IN AI/ML – VITCHENNAI

BECE101P_SLOT-L5+L6_EXPERIMENT – 10

FACULTY: PROF. SASITHRADEVI MA'AM

Digital Logic Gates Experiment

AIM: To study, understand and simulate the basic digital logics (logic gates) and an algebraic Boolean expression using LT-spice.

SOFTWARE REQUIRED: LT-Spice

Apparatus required: In LT-Spice we use wires, grounding cables, logic gates of different kinds and voltage sources.

THEORY:

LOGIC GATES

A logic gate is a simple switching circuit that determines whether an input pulse can pass through to the output in digital circuits.

The building blocks of a digital circuit are logic gates, which execute numerous logical operations that are required by any digital circuit. These can take two or more inputs but only produce one output.

The mix of inputs applied across a logic gate determines its output. Logic gates use Boolean algebra to execute logical processes. Logic gates are found in nearly every digital gadget we use on a regular basis. Logic gates are used in the architecture of our telephones, laptops, tablets, and memory devices.

Boolean Algebra

Boolean algebra is a type of logical algebra in which symbols represent logic levels.

The digits (or symbols) 1 and 0 are related to the logic levels in this algebra; in electrical circuits, logic 1 will represent a closed switch, a high voltage, or a device's "on" state. An open switch, low voltage, or "off" state of the device will be represented by logic 0.

At any one time, a digital device will be in one of these two binary situations. A light bulb can be used to demonstrate the operation of a logic gate. When logic 0 is supplied to the switch, it is turned off, and the bulb does not light up. The switch is in an ON state when logic 1 is applied, and the bulb would light up. In integrated circuits (IC), logic gates are widely employed.

- **Truth Table:** The outputs for all conceivable combinations of inputs that may be applied to a logic gate or circuit are listed in a truth table. When we enter values into a truth table, we usually express them as 1 or 0, with 1 denoting True logic and 0 denoting False logic.

Types of Basic Logic Gates:

1. **AND:**

An AND gate has a single output and two or more inputs.

1. When all of the inputs are 1, the output of this gate is 1.
2. The AND gate's Boolean logic is **Y=A.B** if there are two inputs A and B.

2. **OR:**

Two or more inputs and one output can be used in an OR gate.

1. The logic of this gate is that if at least one of the inputs is 1, the output will be 1.
2. The OR gate's output will be given by the following mathematical procedure if there are two inputs A and B: Y=A+B

3. **NOT:**

The NOT gate is a basic one-input, one-output gate.

1. When the input is 1, the output is 0 and vice versa. A NOT gate is sometimes called as an inverter because of its feature.
2. If there is only one input A, the output may be calculated using the Boolean equation Y=A'.

4. **NAND:**

A NAND gate, sometimes known as a 'NOT-AND' gate, is essentially a Not gate followed by an AND gate.

1. This gate's output is 1 only if none of the inputs is 1. Alternatively, when all of the inputs are not high and at least one is low, the output is high.
2. If there are two inputs A and B, the Boolean expression for the NAND gate is Y=(A.B)'

5. **NOR:**

A NOR gate, sometimes known as a "NOT-OR" gate, consists of an OR gate followed by a NOT gate.

1. This gate's output is 1 only when all of its inputs are 0. Alternatively, when all of the inputs are low, the output is high.
2. The Boolean statement for the NOR gate is Y=(A+B)' if there are two inputs A and B.

6. **XOR:**

The Exclusive-OR or 'Ex-OR' gate is a digital logic gate that accepts more than two inputs but only outputs one value.

1. If any of the inputs is 'High,' the output of the XOR Gate is 'High.' If both inputs are 'High,' the output is 'Low.' If both inputs are 'Low,' the output is 'Low.'
2. The Boolean equation for the XOR gate is Y=A'.B+A.B' if there are two inputs A and B.

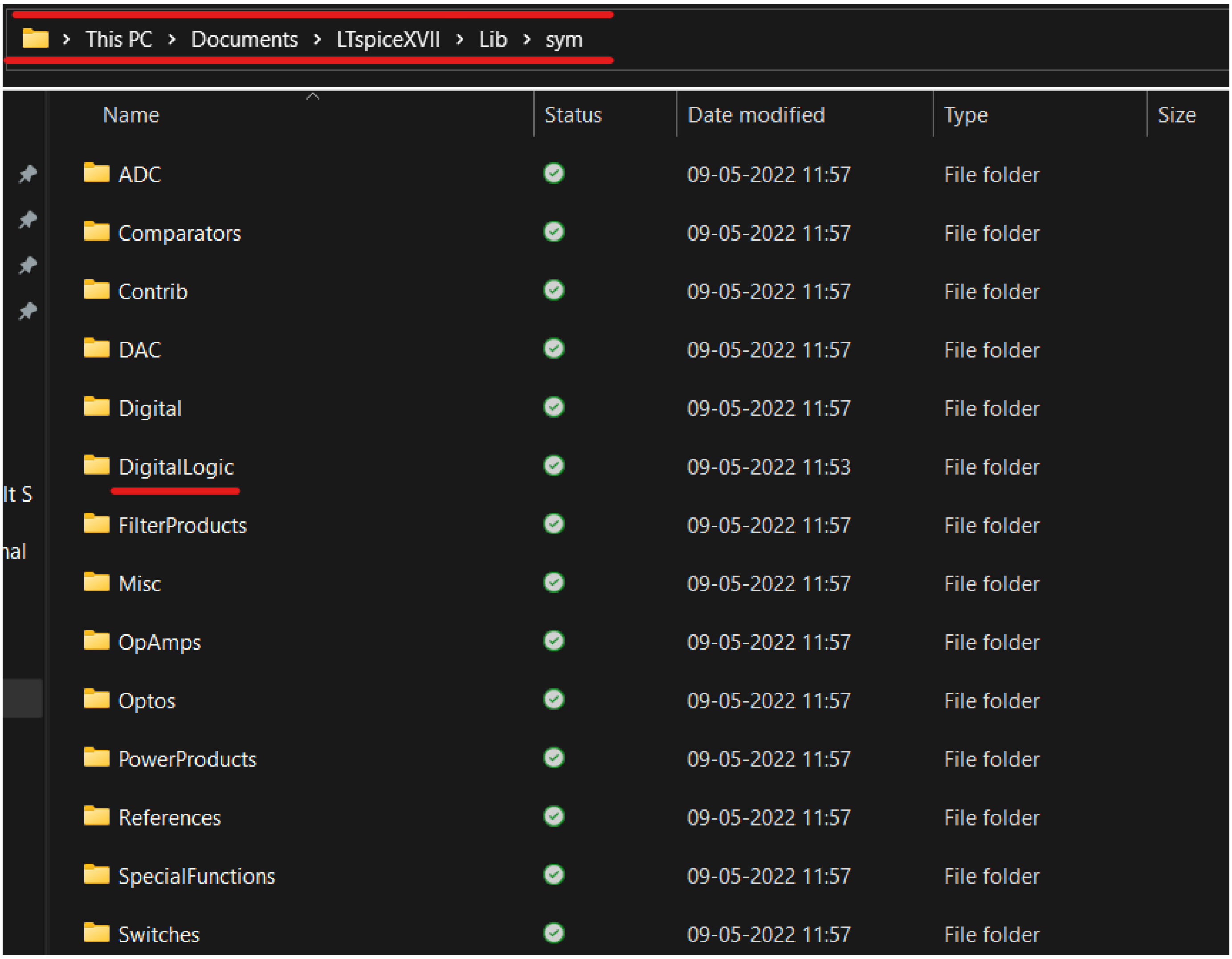
7. **XNOR:**

The Exclusive-NOR or ‘EX-NOR’ gate is a digital logic gate that accepts more than two inputs but only outputs one.

1. If both inputs are ‘High,’ the output of the XNOR Gate is ‘High.’ If both inputs are ‘Low,’ the output is ‘Low.’ If one of the inputs is ‘Low,’ the output is ‘Low.’
2. If there are two inputs A and B, then the XNOR gate’s Boolean equation is: Y=A.B+A’B’.

PROCEDURE

Download the DigitalLogics zip file and place it under the sym file under the lib file inside the LTSPICE folder and then extract the DigitalLogics folder in this location.



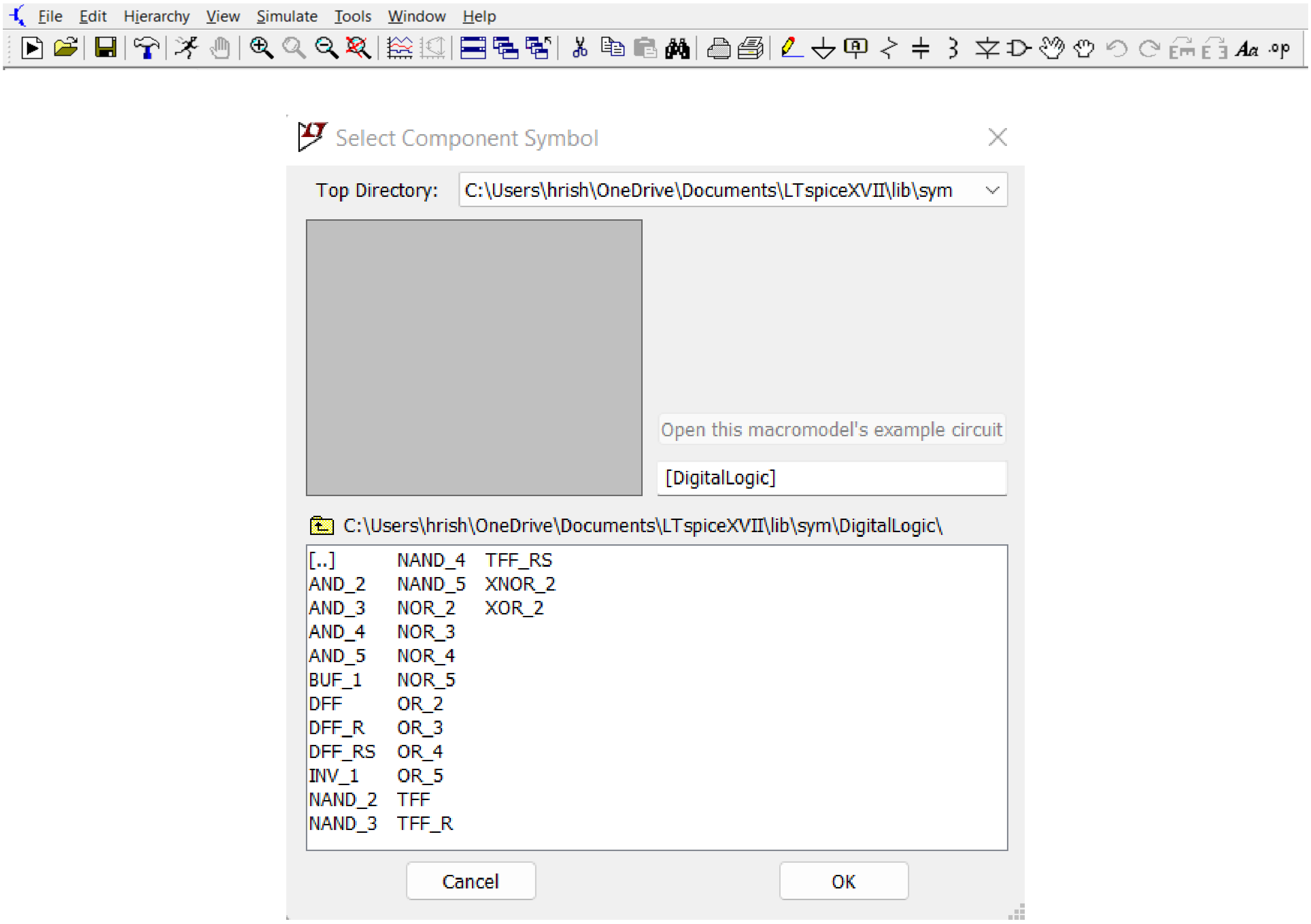
This PC › Documents › LTspiceXVII › Lib › sym

Name	Status	Date modified	Type	Size
ADC	✓	09-05-2022 11:57	File folder	
Comparators	✓	09-05-2022 11:57	File folder	
Contrib	✓	09-05-2022 11:57	File folder	
DAC	✓	09-05-2022 11:57	File folder	
Digital	✓	09-05-2022 11:57	File folder	
DigitalLogic	✓	09-05-2022 11:53	File folder	
FilterProducts	✓	09-05-2022 11:57	File folder	
Misc	✓	09-05-2022 11:57	File folder	
OpAmps	✓	09-05-2022 11:57	File folder	
Optos	✓	09-05-2022 11:57	File folder	
PowerProducts	✓	09-05-2022 11:57	File folder	
References	✓	09-05-2022 11:57	File folder	
SpecialFunctions	✓	09-05-2022 11:57	File folder	
Switches	✓	09-05-2022 11:57	File folder	

FOR BASIC DIGITAL LOGIC GATES:

1. Take 2 voltage sources and place them, connect their negative ends to grounding using wires.

2. Now, click on the component library button on the tool bar above and click on [Digital Logics] and then click on AND logic gate and place it in the circuit.



Select Component Symbol

Top Directory: C:\Users\hrish\OneDrive\Documents\LTspiceXVII\lib\sym

Open this macromodel's example circuit

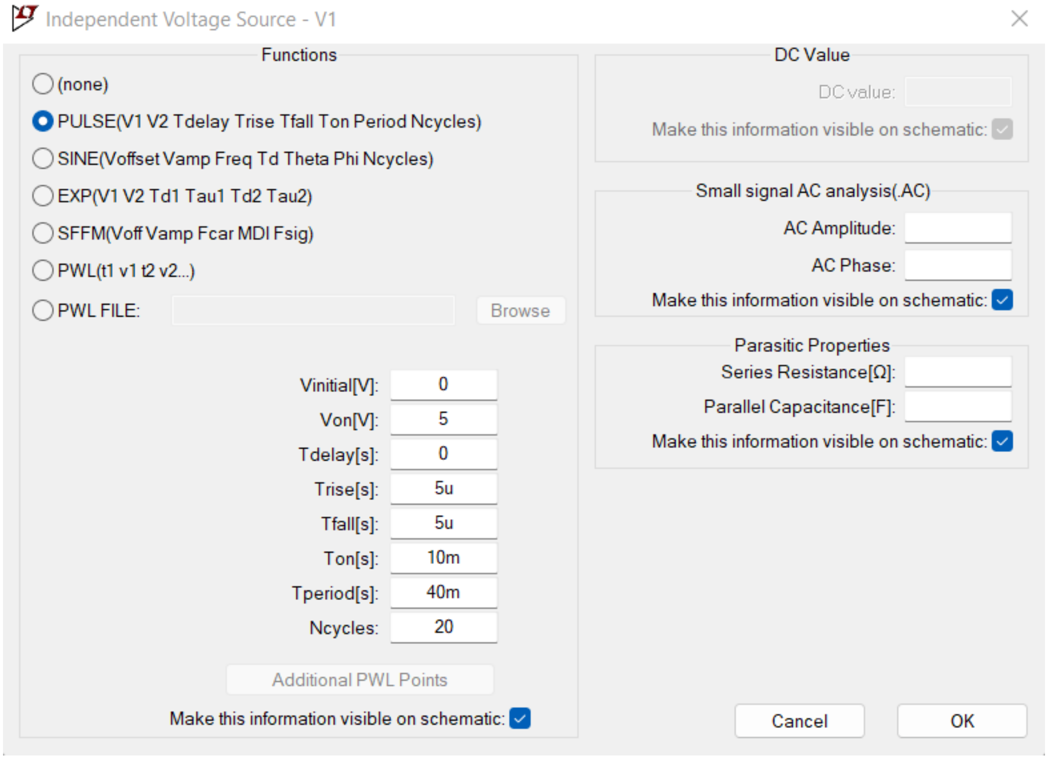
[DigitalLogic]

C:\Users\hrish\OneDrive\Documents\LTspiceXVII\lib\sym\DigitalLogic\

```
[..]      NAND_4   TFF_RS
AND_2     NAND_5   XNOR_2
AND_3     NOR_2    XOR_2
AND_4     NOR_3
AND_5     NOR_4
BUF_1     NOR_5
DFF       OR_2
DFF_R     OR_3
DFF_RS    OR_4
INV_1     OR_5
NAND_2    TFF
NAND_3    TFF_R
```

3. Next, we connect the 2 positive ends of the voltage sources to the 2 ends of the AND logic gate. Also extend the output wire of the logic gate.

4. Now, we give pulse value values as shown for the 2 sources V1 and V2 as shown in the figure below.


Independent Voltage Source - V1

Functions

○ (none)
● PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)
○ SINE(Voffset Vamp Freq Td Theta Phi Ncycles)
○ EXP(V1 V2 Td1 Tau1 Td2 Tau2)
○ SFFM(Voff Vamp Fcar MDI Fsig)
○ PWL(t1 v1 t2 v2...)
○ PWL FILE:

Vinitial[V]:	0
Von[V]:	5
Tdelay[s]:	0
Trise[s]:	5u
Tfall[s]:	5u
Ton[s]:	10m
Tperiod[s]:	40m
Ncycles:	20

Make this information visible on schematic: ☑

DC Value

DC value:

Make this information visible on schematic: ☑

Small signal AC analysis(.AC)

AC Amplitude:

AC Phase:

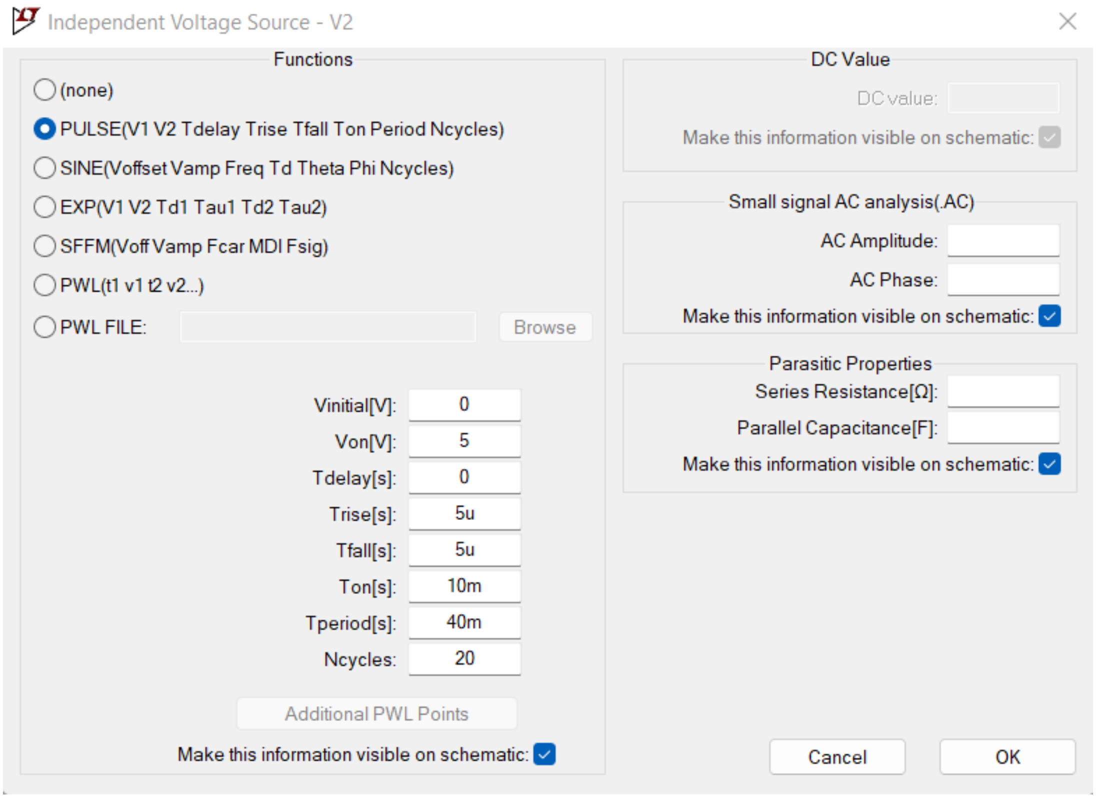
Make this information visible on schematic: ☑

Parasitic Properties

Series Resistance[Ω]:

Parallel Capacitance[F]:

Make this information visible on schematic: ☑



Independent Voltage Source - V2

Functions

○ (none)

● PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)

○ SINE(Voffset Vamp Freq Td Theta Phi Ncycles)

○ EXP(V1 V2 Td1 Tau1 Td2 Tau2)

○ SFFM(Voff Vamp Fcar MDI Fsig)

○ PWL(t1 v1 t2 v2...)

○ PWL FILE:

Vinitial[V]:	0
Von[V]:	5
Tdelay[s]:	0
Trise[s]:	5u
Tfall[s]:	5u
Ton[s]:	10m
Tperiod[s]:	40m
Ncycles:	20

Make this information visible on schematic: ☑

DC Value

DC value:

Make this information visible on schematic: ☑

Small signal AC analysis(.AC)

AC Amplitude:

AC Phase:

Make this information visible on schematic: ☑

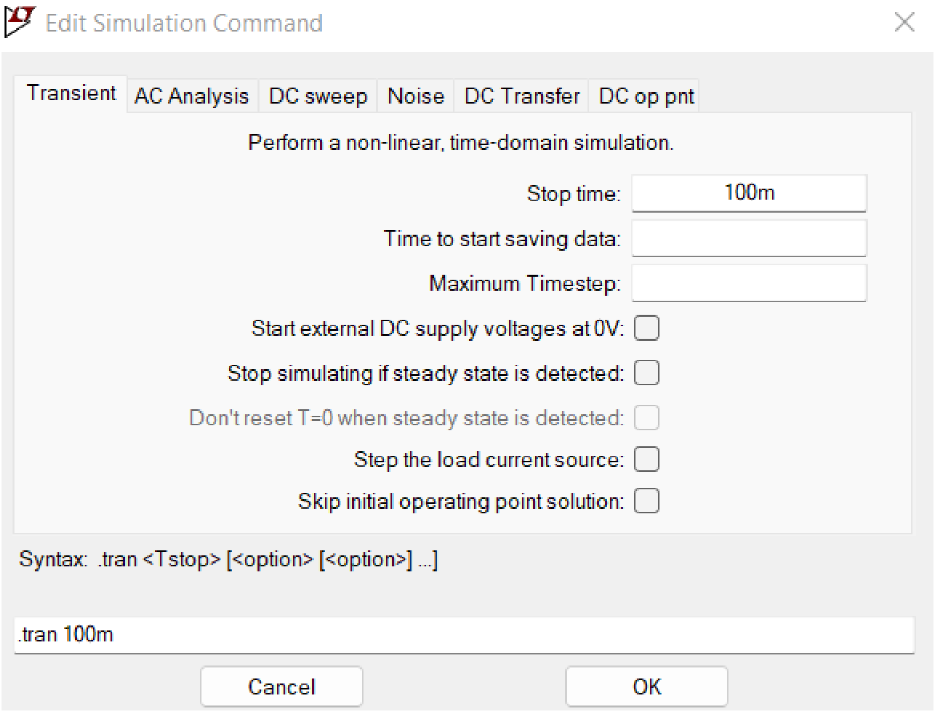
Parasitic Properties

Series Resistance[Ω]:

Parallel Capacitance[F]:

Make this information visible on schematic: ☑

5. Now, go under simulate option and click on edit simulation command and then go under transient and then give the settings as shown below.



Edit Simulation Command

Perform a non-linear, time-domain simulation.

Stop time:	100m
Time to start saving data:	
Maximum Timestep:	

Start external DC supply voltages at 0V: □

Stop simulating if steady state is detected: □

Don't reset T=0 when steady state is detected: □

Step the load current source: □

Skip initial operating point solution: □

Syntax: .tran <Tstop> [<option> [<option>] ...]

.tran 100m

6. Now go under simulation and click on the Run button.

7. After the Graph appears, we left click on the V1 and V2 inputs and then we also left click on the output voltage i.e the output wire coming out of the “AND” gate.

8. Now, we can see the graph for the AND logic gate.

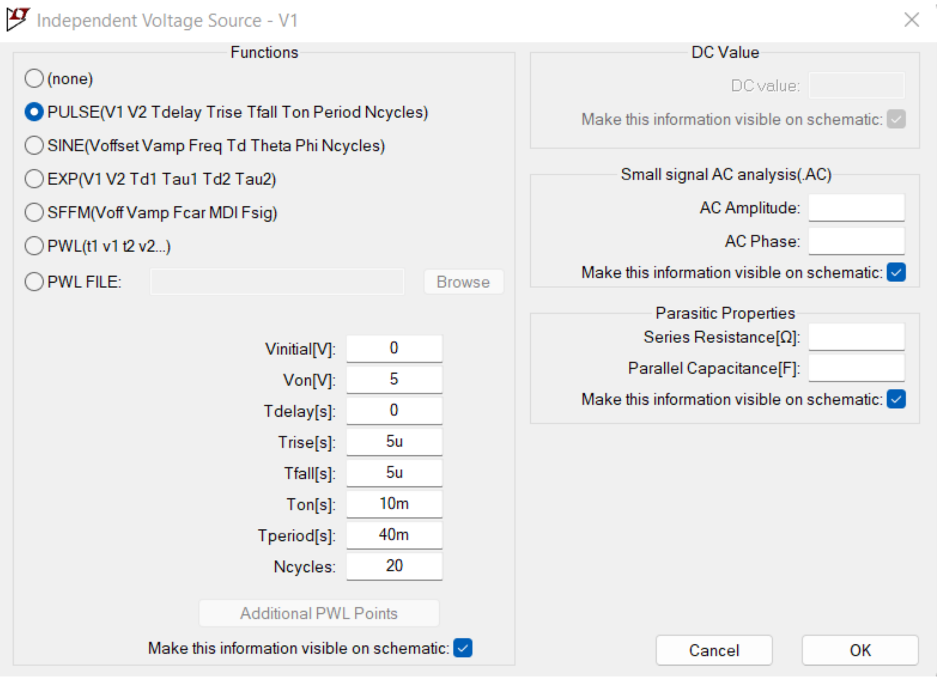
9. We must repeat the same procedure for all the logic gates, keeping the same V1 and V2 values and connections also the same, but just changing the logic gate itself and we record all the output graphs. Repeat for other gates like OR, NAND, NOR, NOT, XOR and XNOR.

10. Only for the "NOT" gate we take single voltage source i.e single input with single wire and also just taking a single output.

FOR BOOLEAN EXPRESSION USING DIGITAL LOGIC GATES:

1. We construct [((V1orV2)&V3)orV2] using the 2 digital logic gates. Using the gates under the component library inside the digital logic gates section.

2. We use 3 source V1,V2 and V3 an give them the pulse value as shown below.



Independent Voltage Source - V1

Functions

○ (none)
● PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)
○ SINE(Voffset Vamp Freq Td Theta Phi Ncycles)
○ EXP(V1 V2 Td1 Tau1 Td2 Tau2)
○ SFFM(Voff Vamp Fcar MDI Fsig)
○ PWL(t1 v1 t2 v2...)
○ PWL FILE:

Vinitial[V]: 0
Von[V]: 5
Tdelay[s]: 0
Trise[s]: 5u
Tfall[s]: 5u
Ton[s]: 10m
Tperiod[s]: 40m
Ncycles: 20

Make this information visible on schematic: ☑

DC Value

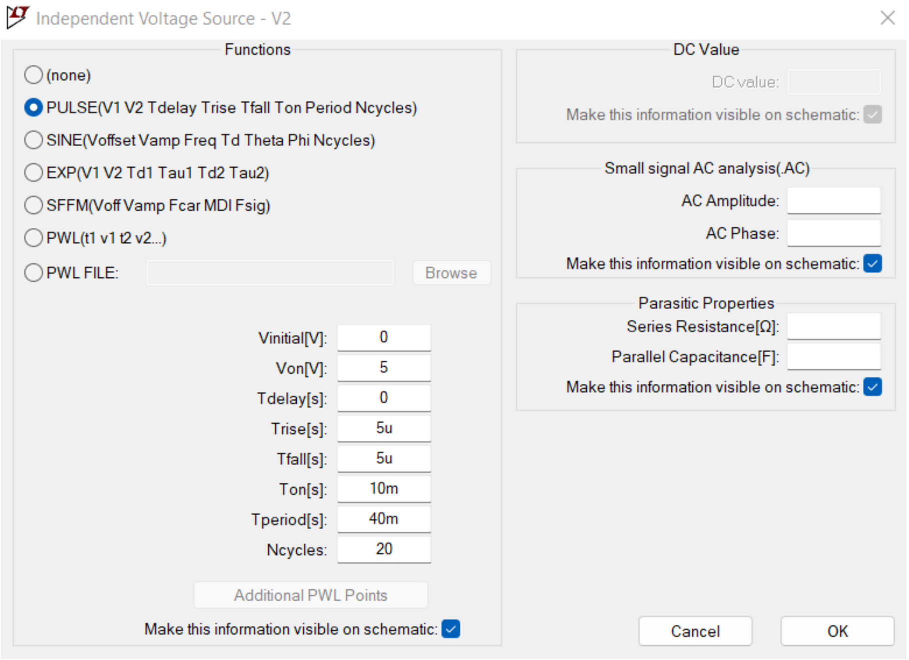
DC value:
Make this information visible on schematic: ☑

Small signal AC analysis(.AC)

AC Amplitude:
AC Phase:
Make this information visible on schematic: ☑

Parasitic Properties

Series Resistance[Ω]:
Parallel Capacitance[F]:
Make this information visible on schematic: ☑



Independent Voltage Source - V2

Functions

○ (none)
● PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)
○ SINE(Voffset Vamp Freq Td Theta Phi Ncycles)
○ EXP(V1 V2 Td1 Tau1 Td2 Tau2)
○ SFFM(Voff Vamp Fcar MDI Fsig)
○ PWL(t1 v1 t2 v2...)
○ PWL FILE:

Vinitial[V]: 0
Von[V]: 5
Tdelay[s]: 0
Trise[s]: 5u
Tfall[s]: 5u
Ton[s]: 10m
Tperiod[s]: 40m
Ncycles: 20

Make this information visible on schematic: ☑

DC Value

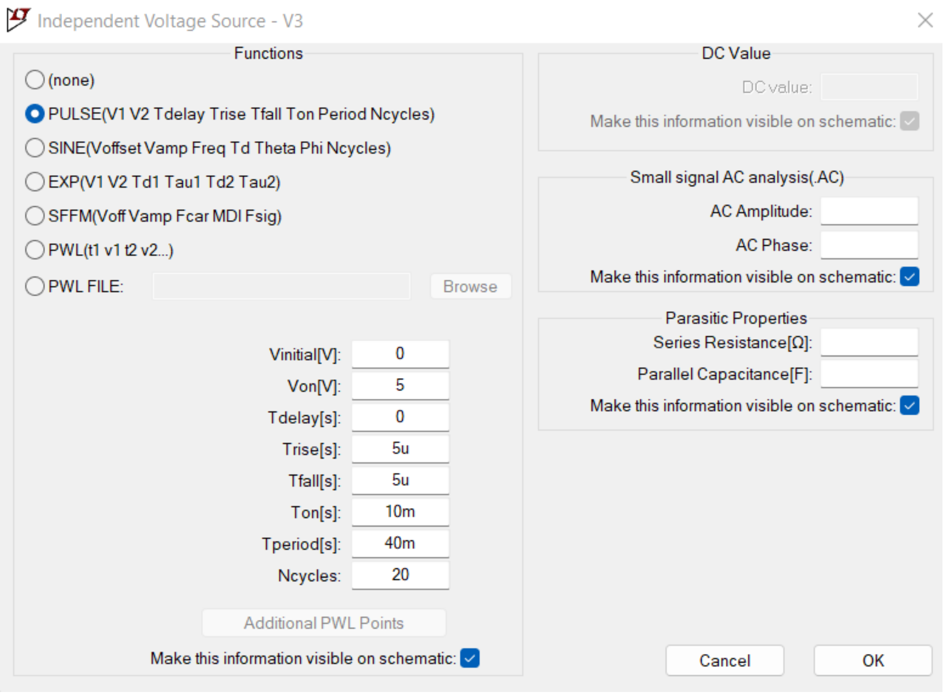
DC value:
Make this information visible on schematic: ☑

Small signal AC analysis(.AC)

AC Amplitude:
AC Phase:
Make this information visible on schematic: ☑

Parasitic Properties

Series Resistance[Ω]:
Parallel Capacitance[F]:
Make this information visible on schematic: ☑



Independent Voltage Source - V3

Functions

○ (none)

● PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)

○ SINE(Voffset Vamp Freq Td Theta Phi Ncycles)

○ EXP(V1 V2 Td1 Tau1 Td2 Tau2)

○ SFFM(Voff Vamp Fcar MDI Fsig)

○ PWL(t1 v1 t2 v2...)

○ PWL FILE:

Vinitial[V]: 0

Von[V]: 5

Tdelay[s]: 0

Trise[s]: 5u

Tfall[s]: 5u

Ton[s]: 10m

Tperiod[s]: 40m

Ncycles: 20

Make this information visible on schematic: ☑

DC Value

DC value:

Make this information visible on schematic: ☑

Small signal AC analysis(.AC)

AC Amplitude:

AC Phase:

Make this information visible on schematic: ☑

Parasitic Properties

Series Resistance[Ω]:

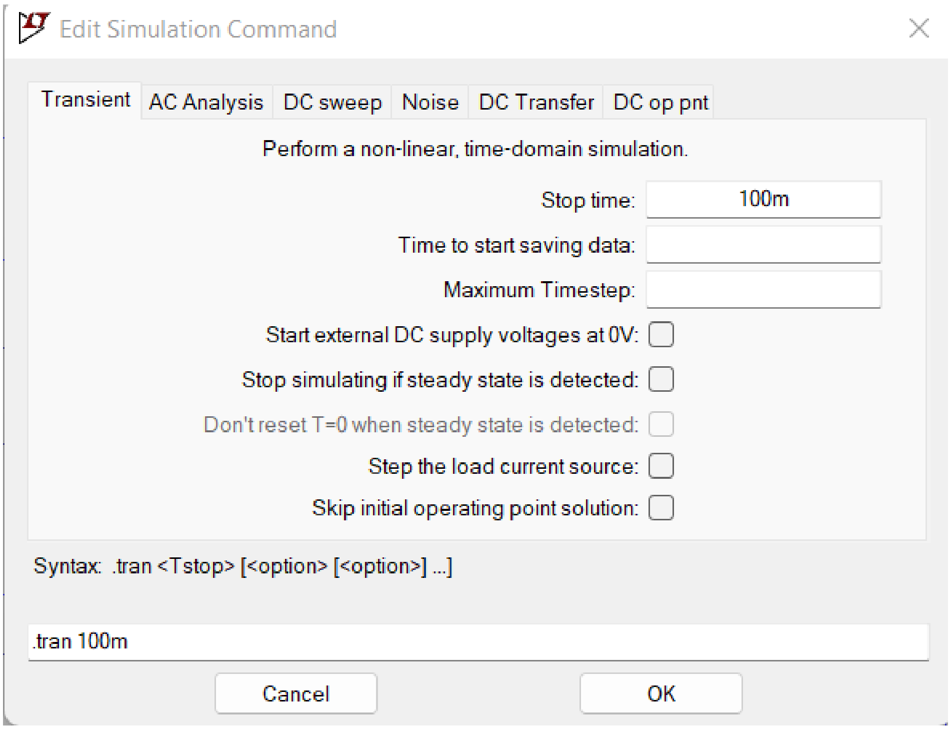
Parallel Capacitance[F]:

Make this information visible on schematic: ☑

3. We connect V1,V2 and V3 negative terminals to the ground.

4. Now, we make the circuit according to the expression. (Circuit shown later)

5. Now, under the simulate we click on the edit simulation command and then give the transient settings as follows.



Edit Simulation Command

Perform a non-linear, time-domain simulation.

Stop time: 100m

Time to start saving data:

Maximum Timestep:

Start external DC supply voltages at 0V: □

Stop simulating if steady state is detected: □

Don't reset T=0 when steady state is detected: □

Step the load current source: □

Skip initial operating point solution: □

Syntax: .tran <Tstop> [<option> [<option>] ...]

```
.tran 100m
```

6. Next, we go under the simulate tab and click on the RUN button.

7. When the graph appears, we left click on the input voltages(click on the wire to the connected to the positive ends of V1, V2 and V3).

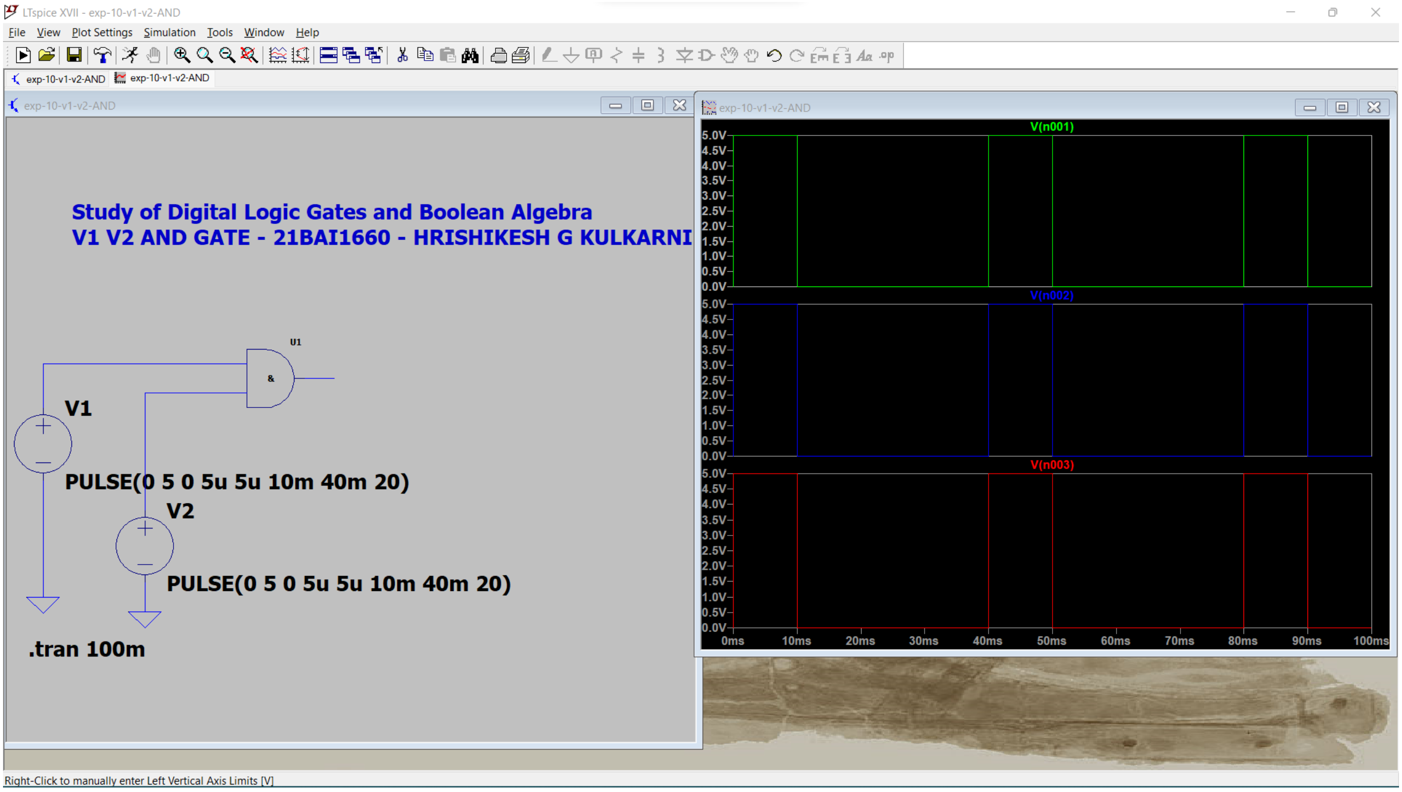
8. And for the output, we left click on the wire to rightmost which is coming out of the OR gate.

9. We get the output graph.

CIRCUITS AND OUTPUT GRAPHS

For BASIC LOGIC GATES

AND:



Study of Digital Logic Gates and Boolean Algebra
V1 V2 AND GATE - 21BAI1660 - HRISHIKESH G KULKARNI

PULSE(0 5 0 5u 5u 10m 40m 20)

PULSE(0 5 0 5u 5u 10m 40m 20)

.tran 100m

Right-Click to manually enter Left Vertical Axis Limits [V]

NEXT PAGE

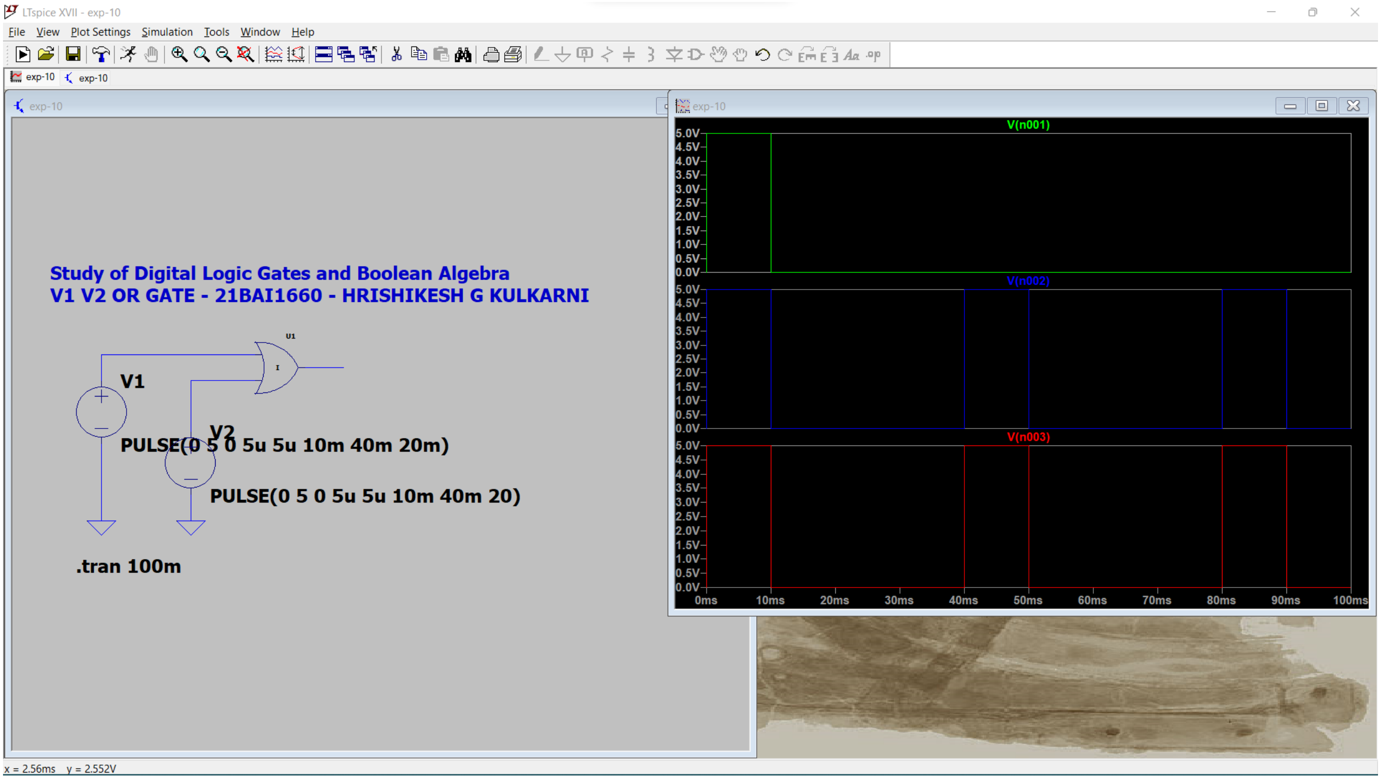
|

|

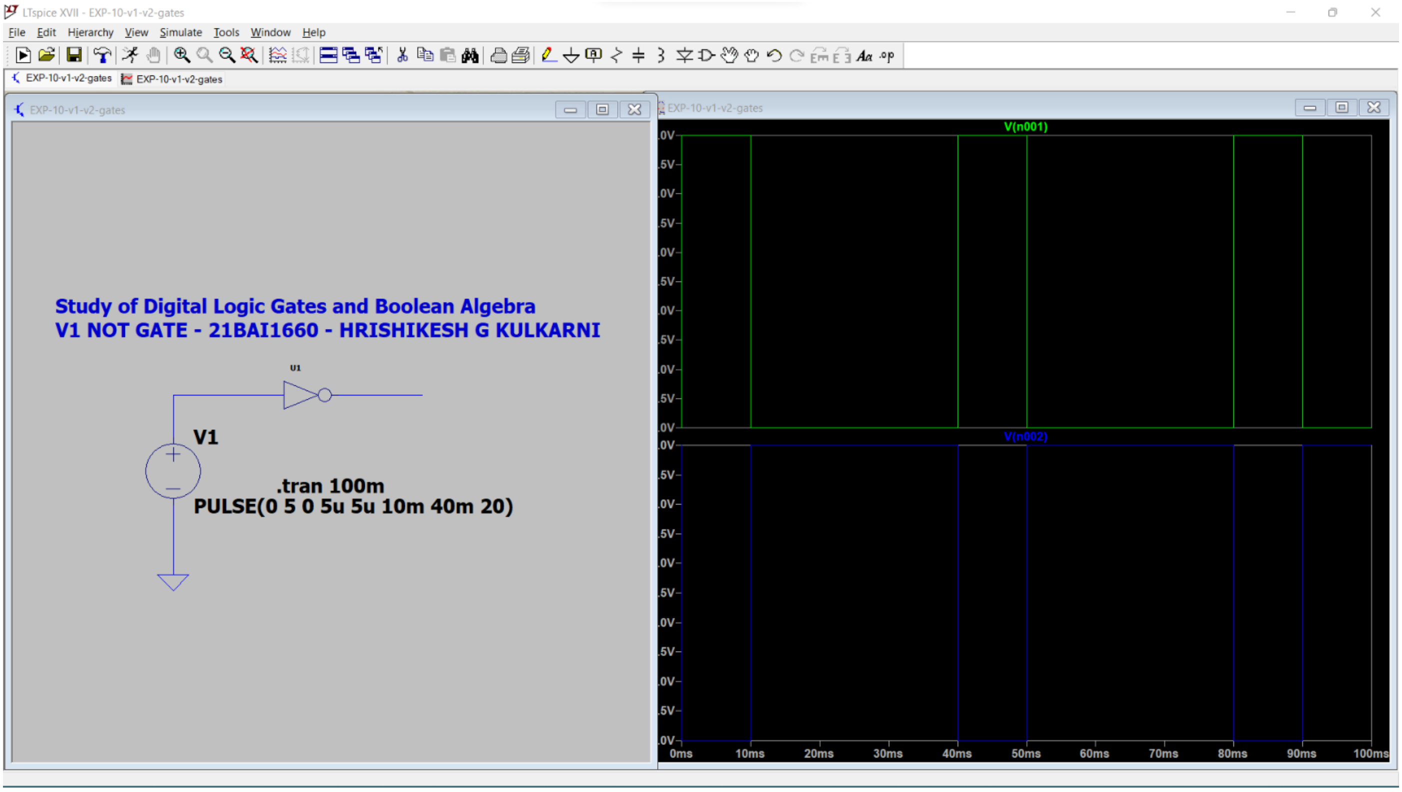
|

V

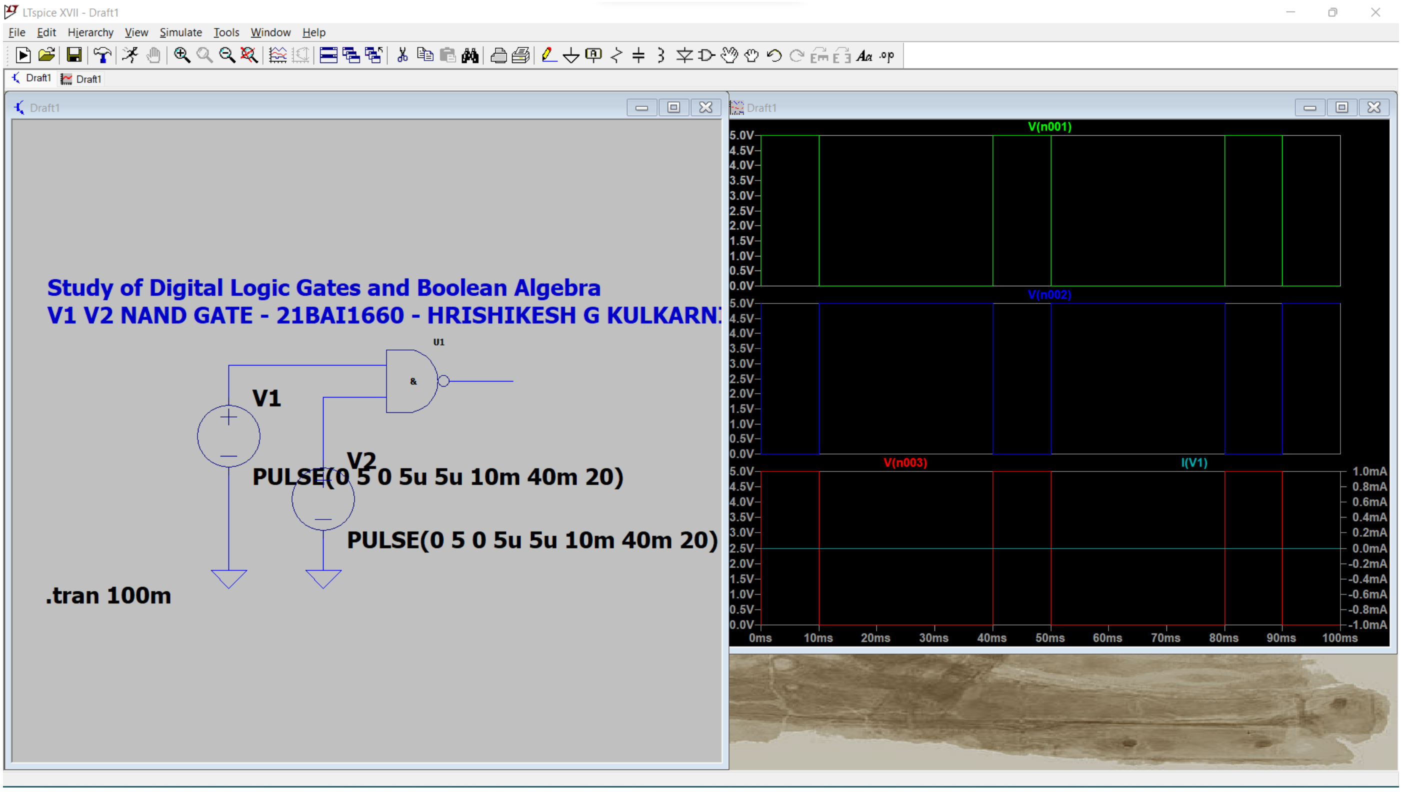
OR:


LTspice XVII - exp-10
File View Plot Settings Simulation Tools Window Help
exp-10
exp-10
Study of Digital Logic Gates and Boolean Algebra
V1 V2 OR GATE - 21BAI1660 - HRISHIKESH G KULKARNI
U1
V1
V2
PULSE(0 5 0 5u 5u 10m 40m 20m)
PULSE(0 5 0 5u 5u 10m 40m 20)
.tran 100m
V(n001)
V(n002)
V(n003)
5.0V
4.5V
4.0V
3.5V
3.0V
2.5V
2.0V
1.5V
1.0V
0.5V
0.0V
0ms
10ms
20ms
30ms
40ms
50ms
60ms
70ms
80ms
90ms
100ms
x = 2.56ms y = 2.552V

NOT:


LTspice XVII - EXP-10-v1-v2-gates
File Edit Hierarchy View Simulate Tools Window Help
EXP-10-v1-v2-gates
EXP-10-v1-v2-gates
Study of Digital Logic Gates and Boolean Algebra
V1 NOT GATE - 21BAI1660 - HRISHIKESH G KULKARNI
U1
V1
.tran 100m
PULSE(0 5 0 5u 5u 10m 40m 20)
V(n001)
V(n002)
0ms
10ms
20ms
30ms
40ms
50ms
60ms
70ms
80ms
90ms
100ms

NAND:



Study of Digital Logic Gates and Boolean Algebra
V1 V2 NAND GATE - 21BAI1660 - HRISHIKESH G KULKARNI

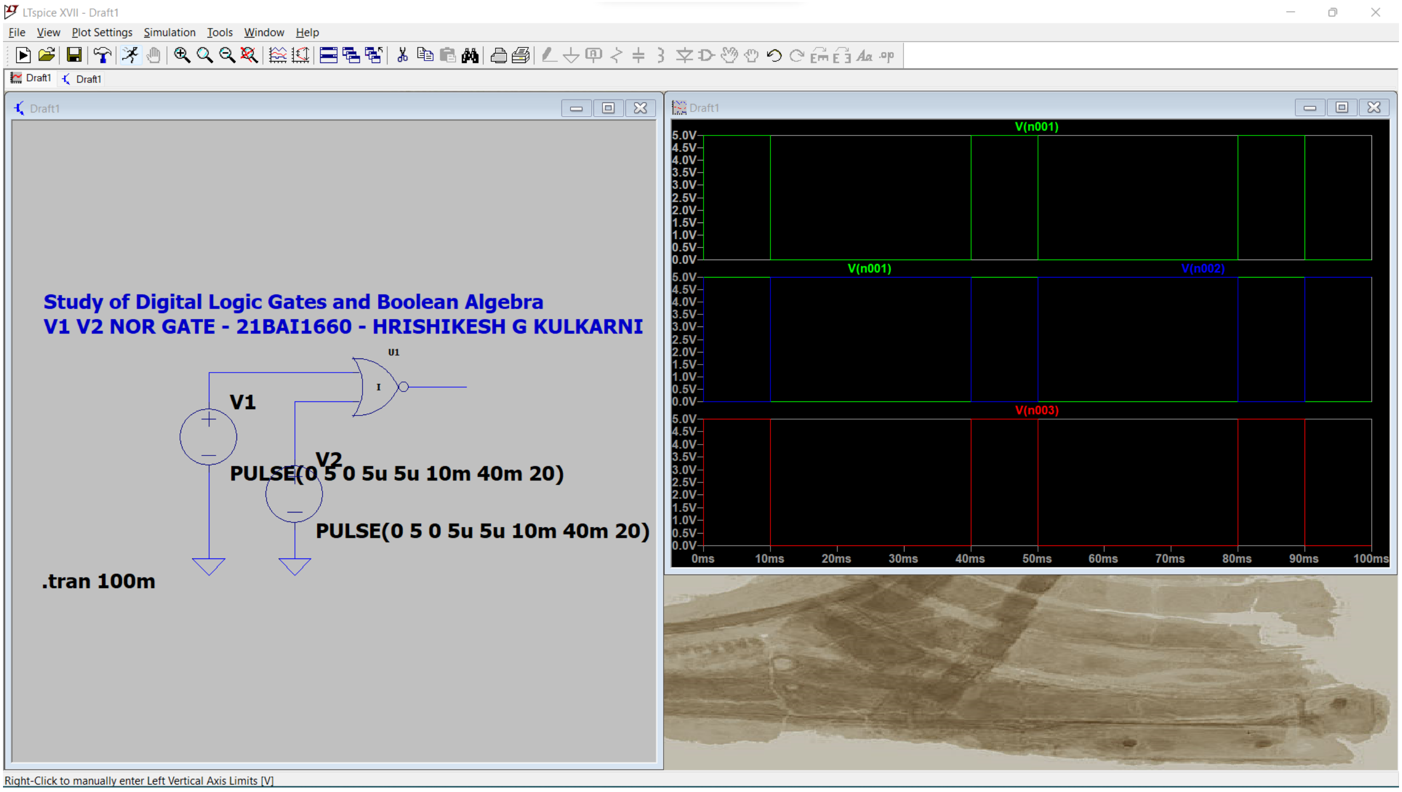
PULSE(0 5 0 5u 5u 10m 40m 20)

PULSE(0 5 0 5u 5u 10m 40m 20)

.tran 100m

LTspice XVII - Draft1
File Edit Hierarchy View Simulate Tools Window Help
Draft1
Draft1
Draft1
Draft1
U1
&
V1
V2
V(n001)
V(n002)
V(n003)
I(V1)
5.0V 4.5V 4.0V 3.5V 3.0V 2.5V 2.0V 1.5V 1.0V 0.5V 0.0V
1.0mA 0.8mA 0.6mA 0.4mA 0.2mA 0.0mA -0.2mA -0.4mA -0.6mA -0.8mA -1.0mA
0ms 10ms 20ms 30ms 40ms 50ms 60ms 70ms 80ms 90ms 100ms

NOR:



Study of Digital Logic Gates and Boolean Algebra
V1 V2 NOR GATE - 21BAI1660 - HRISHIKESH G KULKARNI

PULSE(0 5 0 5u 5u 10m 40m 20)

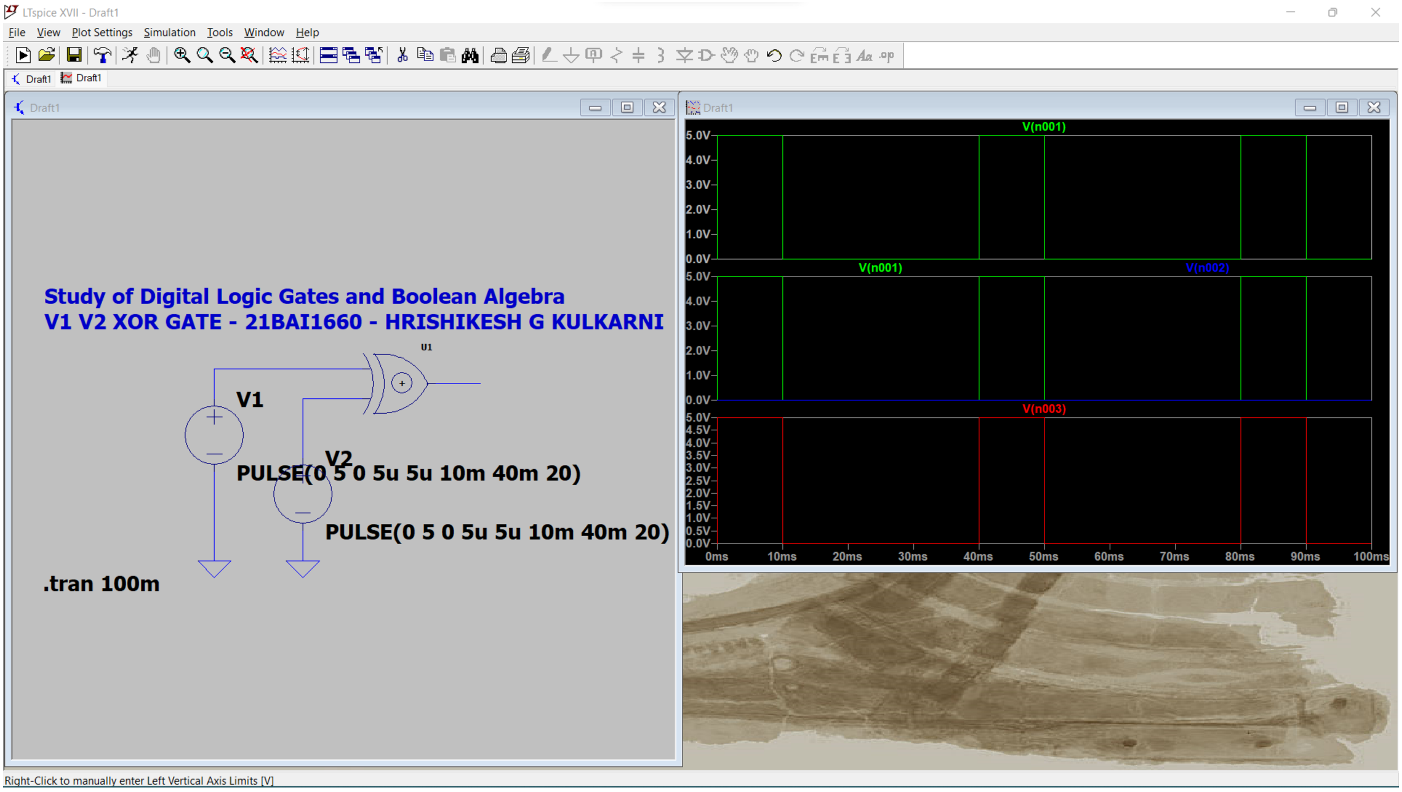
PULSE(0 5 0 5u 5u 10m 40m 20)

.tran 100m

Right-Click to manually enter Left Vertical Axis Limits [V]

LTspice XVII - Draft1
File View Plot Settings Simulation Tools Window Help
Draft1
Draft1
Draft1
Draft1
U1
1
V1
V2
V(n001)
V(n001)
V(n002)
V(n003)
5.0V 4.5V 4.0V 3.5V 3.0V 2.5V 2.0V 1.5V 1.0V 0.5V 0.0V
0ms 10ms 20ms 30ms 40ms 50ms 60ms 70ms 80ms 90ms 100ms

XOR:



Study of Digital Logic Gates and Boolean Algebra
V1 V2 XOR GATE - 21BAI1660 - HRISHIKESH G KULKARNI

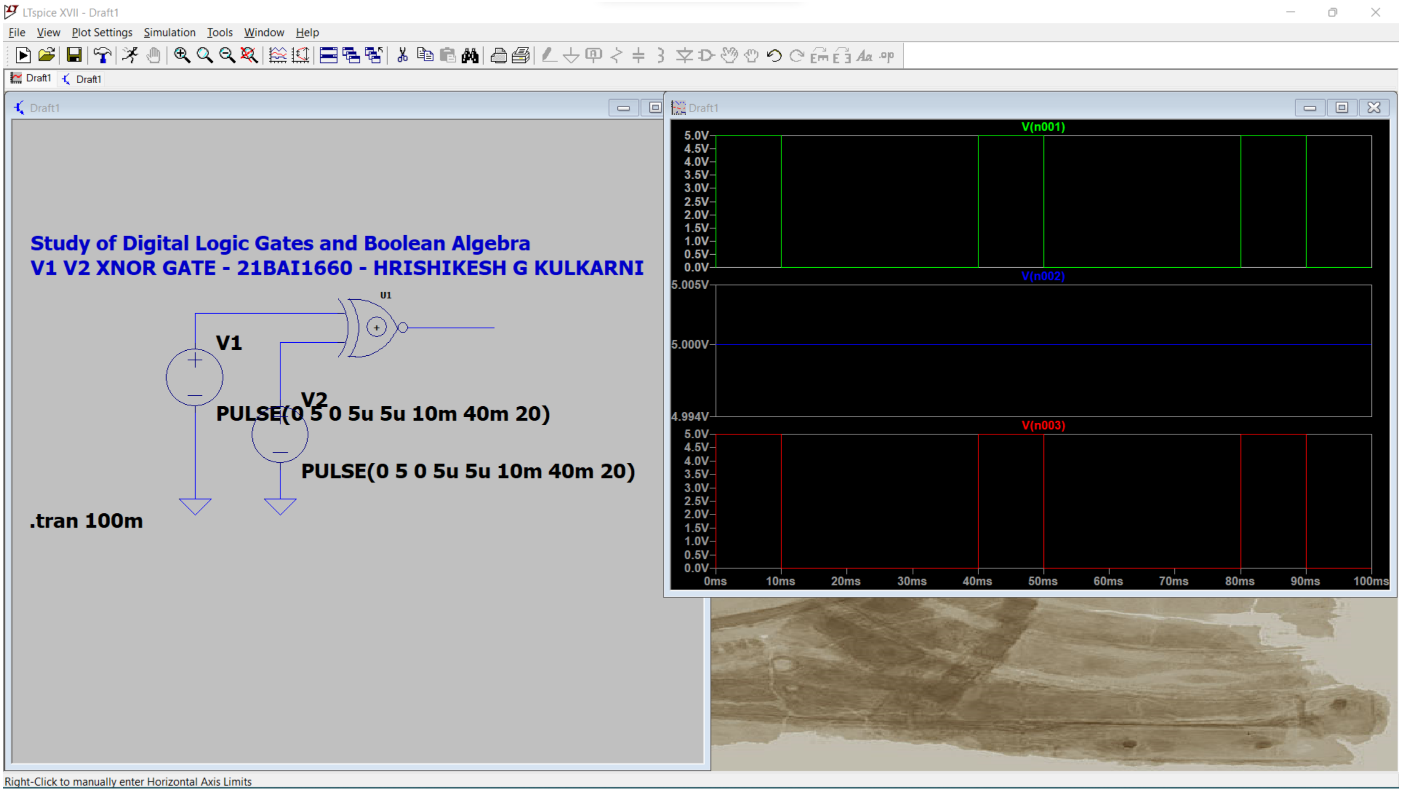
V1 PULSE(0 5 0 5u 5u 10m 40m 20)

V2 PULSE(0 5 0 5u 5u 10m 40m 20)

.tran 100m

Right-Click to manually enter Left Vertical Axis Limits [V]

XNOR:



Study of Digital Logic Gates and Boolean Algebra
V1 V2 XNOR GATE - 21BAI1660 - HRISHIKESH G KULKARNI

V1 PULSE(0 5 0 5u 5u 10m 40m 20)

V2 PULSE(0 5 0 5u 5u 10m 40m 20)

.tran 100m

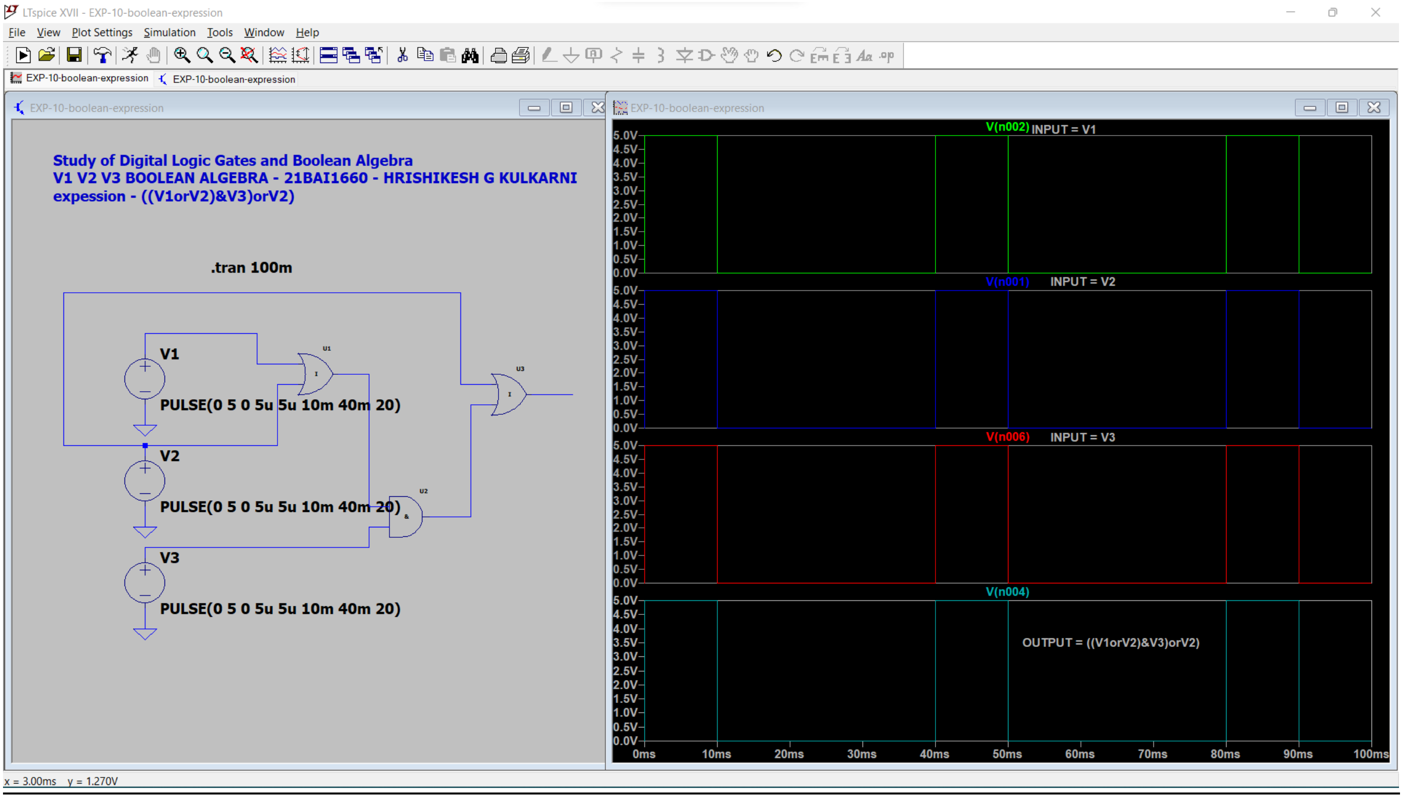
Right-Click to manually enter Horizontal Axis Limits

NEXT PAGE

|

V

For Boolean Expression using LOGIC GATES:



LTspice XVII - EXP-10-boolean-expression

Study of Digital Logic Gates and Boolean Algebra
V1 V2 V3 BOOLEAN ALGEBRA - 21BAI1660 - HRISHIKESH G KULKARNI
expression - ((V1orV2)&V3)orV2)

.tran 100m

V1 PULSE(0 5 0 5u 5u 10m 40m 20)

V2 PULSE(0 5 0 5u 5u 10m 40m 20)

V3 PULSE(0 5 0 5u 5u 10m 40m 20)

x = 3.00ms y = 1.270V

RESULT AND INFERENCE

- **We acquired the basic knowledge of Logic gates.**
- **We were able to study, understand and simulate the various basic logic gates and also the algebraic Boolean expression using LT-Spice software.**

=====THE END=====