



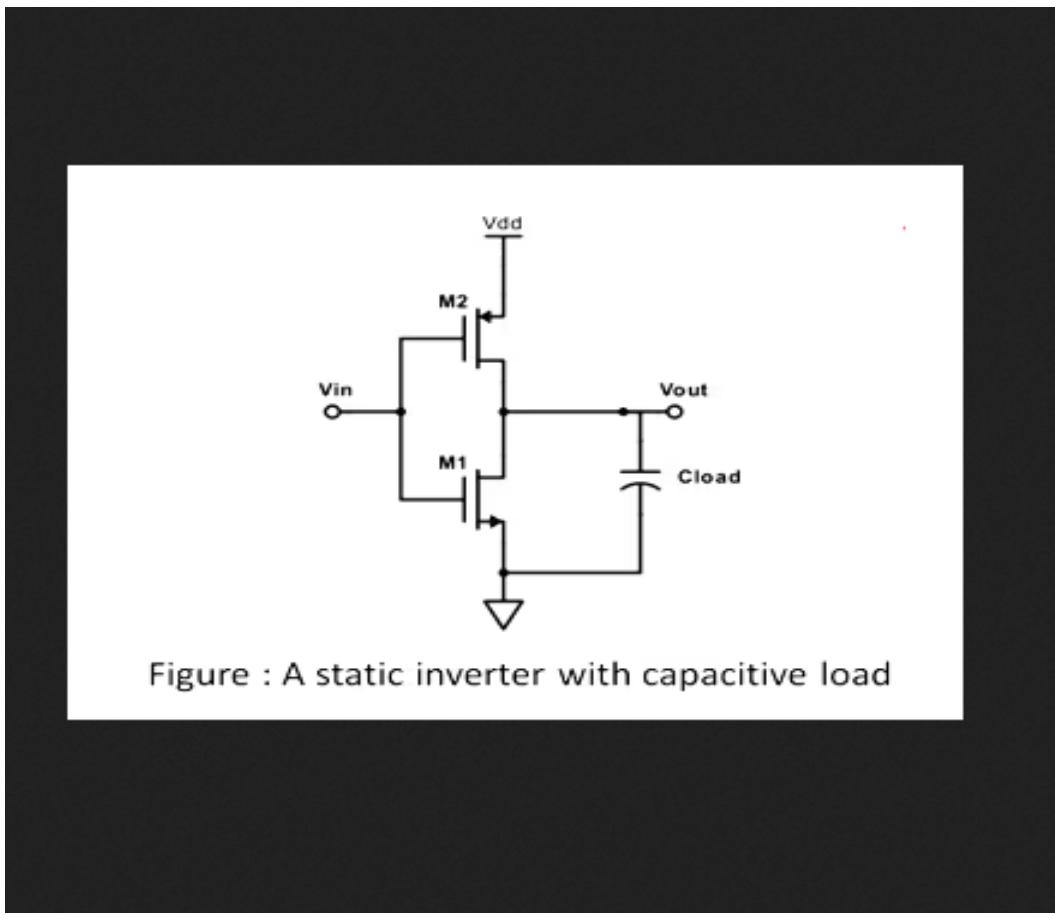
VLSI

Assignment Report I

Logic Families

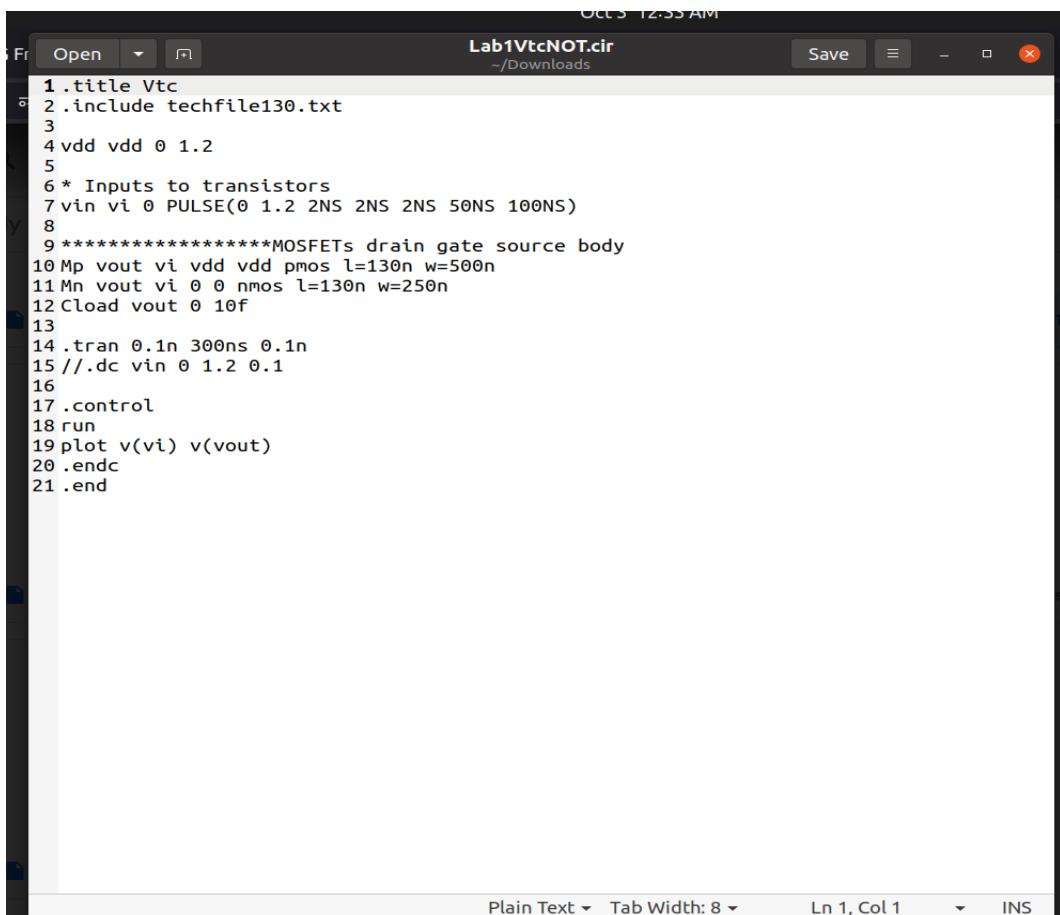
Neel Shahakar
20BEC026

1. Inverter



The basic structure of a Complementary Metal oxide semiconductor inverter consists of an n-MOS transistor and p-MOS transistor as a load and the gates of the two transistors are shorted at the input and the drains of the two transistors are also shorted where the output is obtained. The source n-MOS and p-MOS transistors of the CMOS Inverter are connected to the ground and supply respectively. Its operation is readily understood with the help of the simple switch model of the MOS Transistor. The transistor is nothing more than a switch with an infinite off resistance (for $|V_{GS}| < |V_T|$) and a finite on-resistance (for $|V_{GS}| > |V_T|$). This leads to the following interpretation of the inverter.

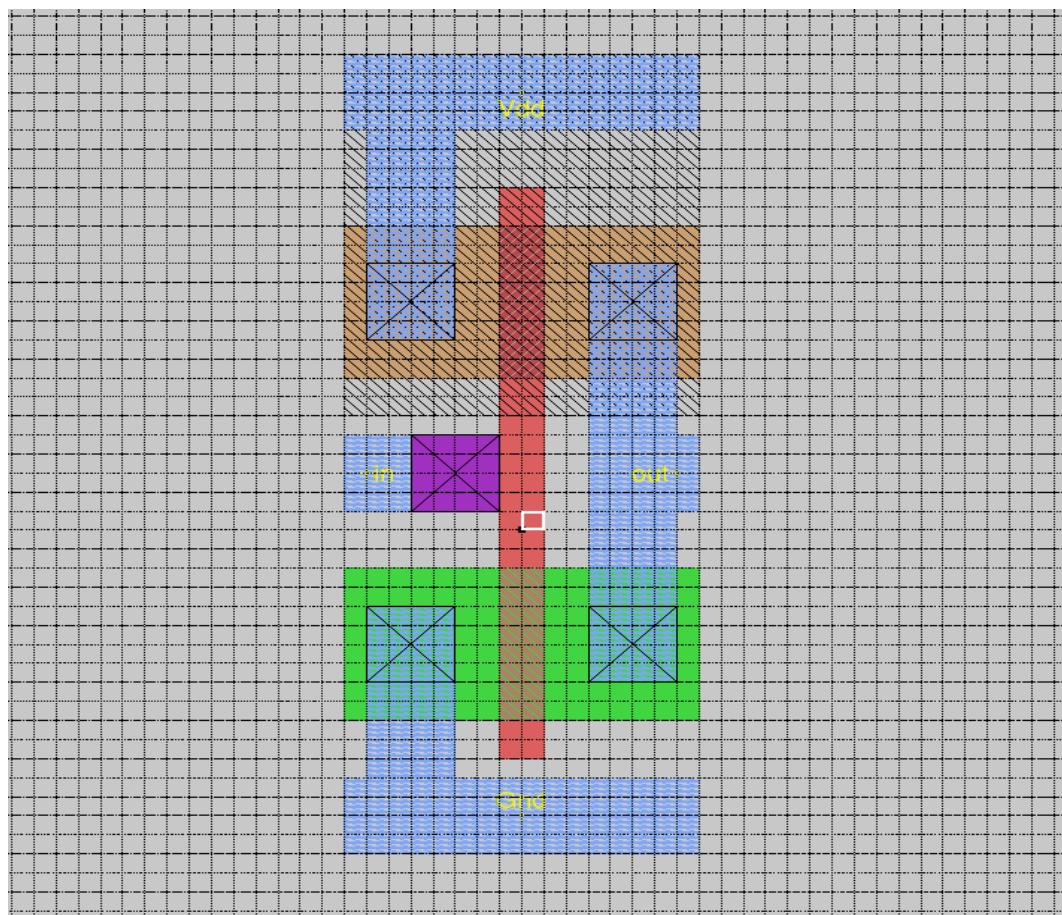
Netlist



The screenshot shows a terminal window with a dark theme running a text editor. The file is titled "Lab1VtcNOT.cir" and is located in the "/Downloads" directory. The timestamp at the top right is "Oct 5 12:55 AM". The editor interface includes a menu bar with "File", "Open", and a file icon. On the right side, there are buttons for "Save", "Minimize", "Maximize", and "Close". The bottom of the window has status bars for "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS". The main content area contains the following SPICE netlist:

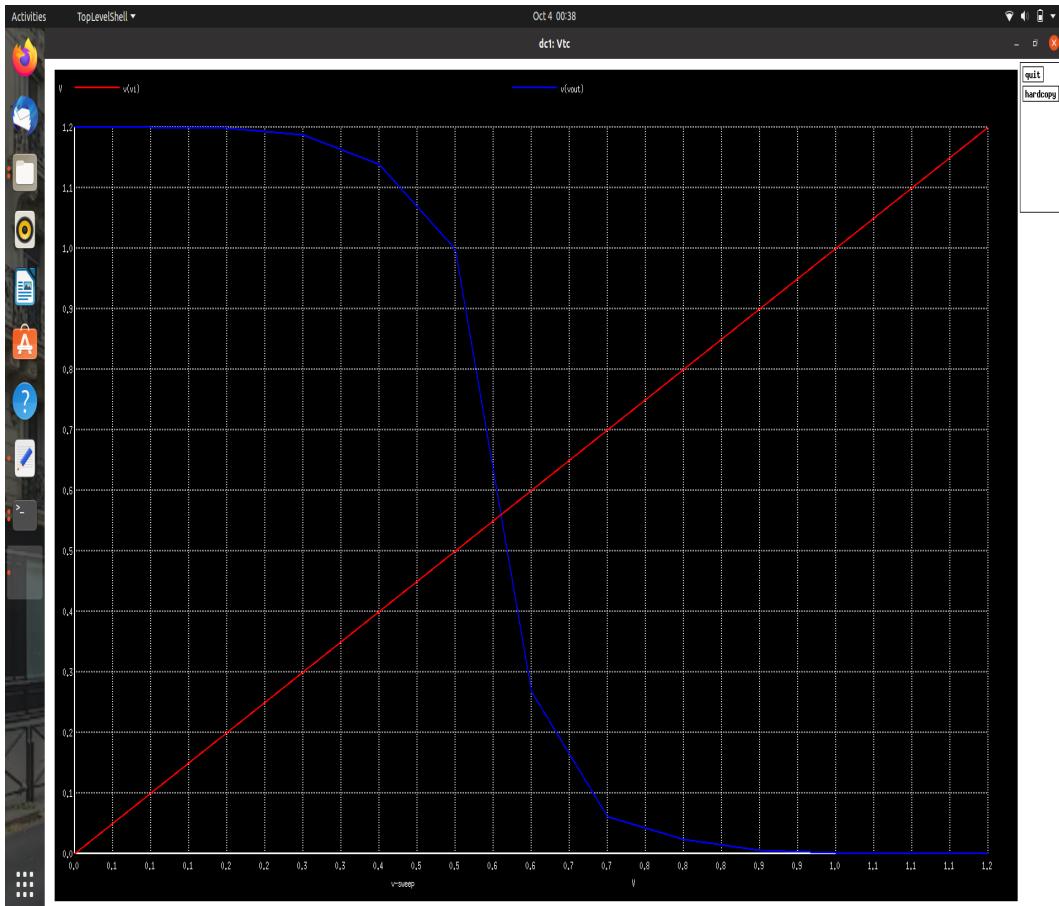
```
1 .title Vtc
2 .include techfile130.txt
3
4 vdd vdd 0 1.2
5
6 * Inputs to transistors
7 vin vi 0 PULSE(0 1.2 2NS 2NS 2NS 50NS 100NS)
8
9 *****MOSFETs drain gate source body
10 Mp vout vi vdd vdd pmos l=130n w=500n
11 Mn vout vi 0 0 nmos l=130n w=250n
12 Cload vout 0 10f
13
14 .tran 0.1n 300ns 0.1n
15 //dc vin 0 1.2 0.1
16
17 .control
18 run
19 plot v(vi) v(vout)
20 .endc
21 .end
```

Layout

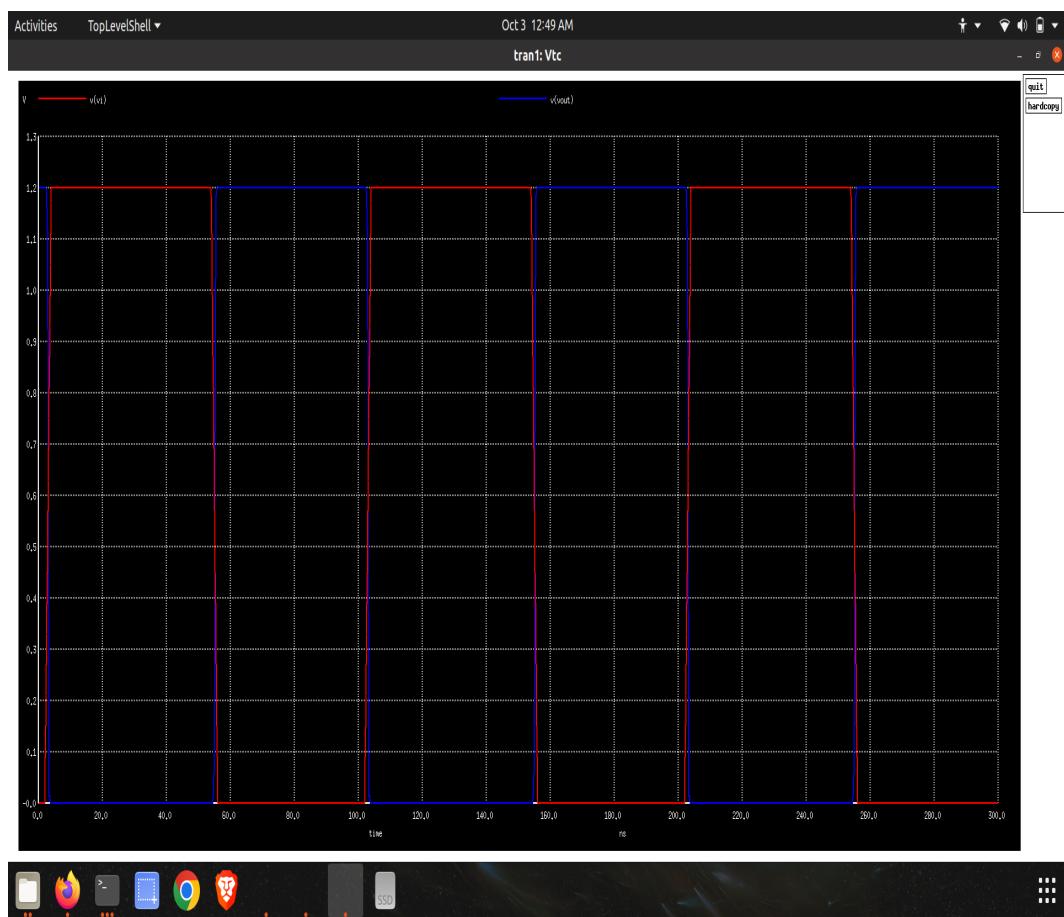


Simulation Plots

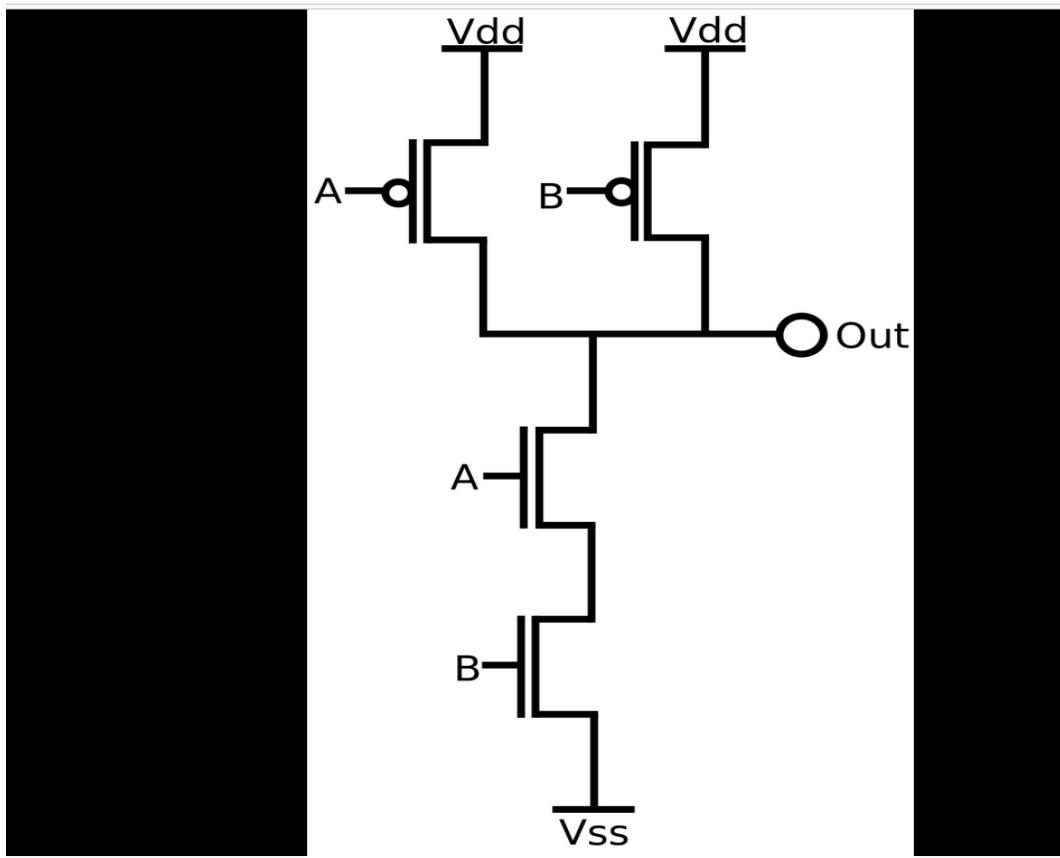
DC analysis



Transient analysis

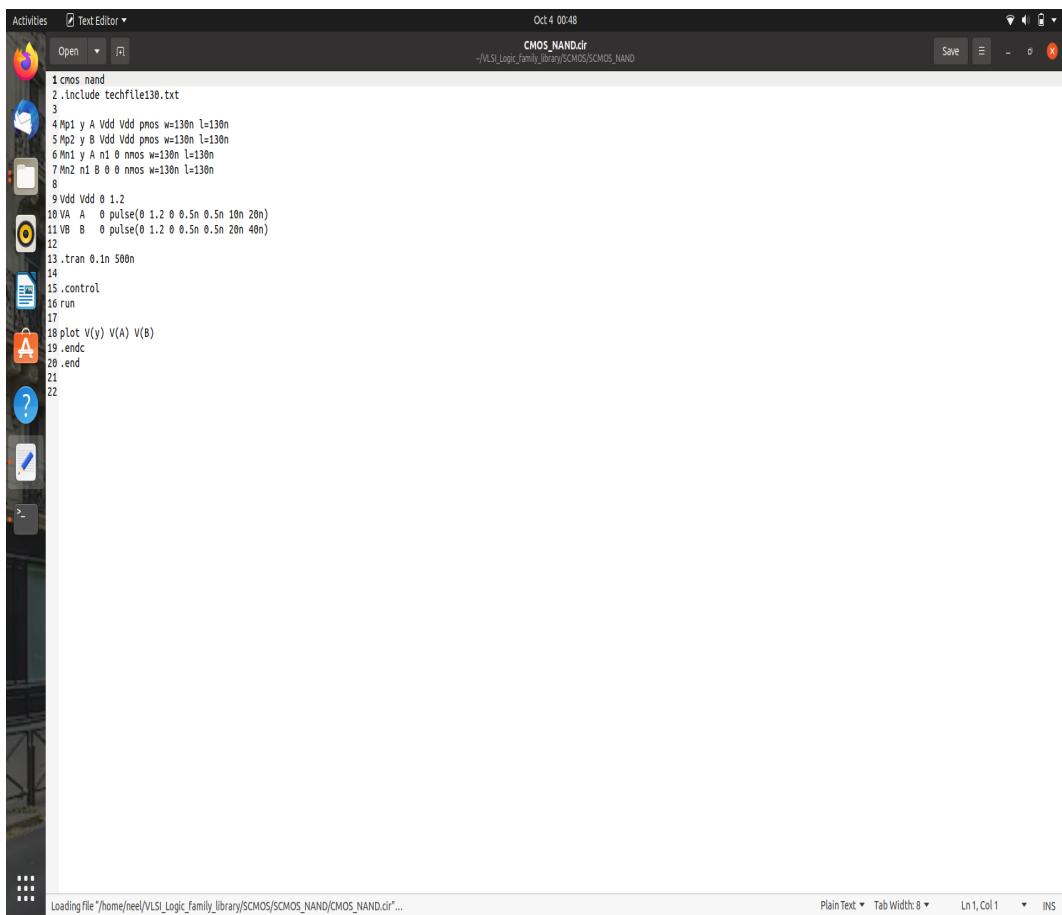


2. 2 i/p SCMOS NAND



NAND gate (NOT-AND) is a logic gate which produces an output which is false only if all its inputs are true; thus its output is complement to that of an AND gate. A LOW (0) output results only if all the inputs to the gate are HIGH (1); if any input is LOW (0), a HIGH (1) output results. A NAND gate is made using transistors and junction diodes. By De Morgan's laws, a two-input NAND gate's logic may be expressed as $A \cdot B = A + B$, making a NAND gate equivalent to inverters followed by an OR gate.

Netlist

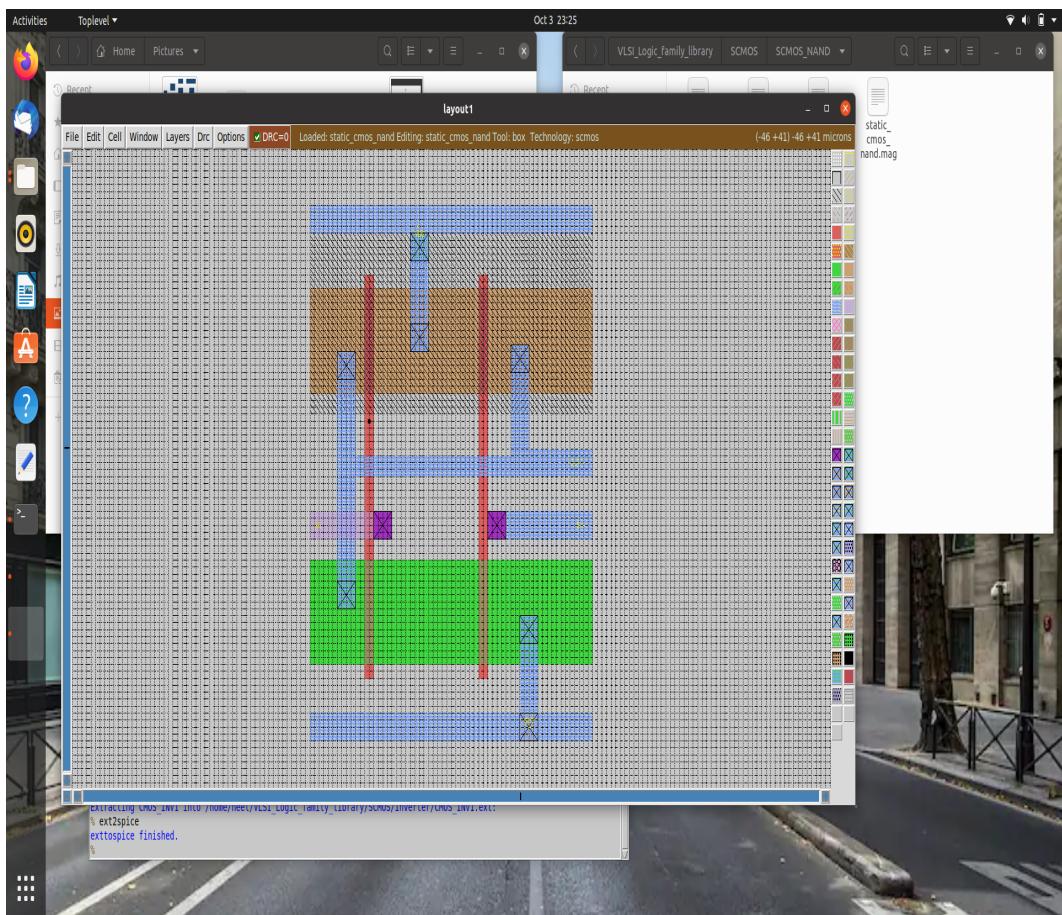


The screenshot shows a terminal window titled "CMOS_NAND.drc" with the following content:

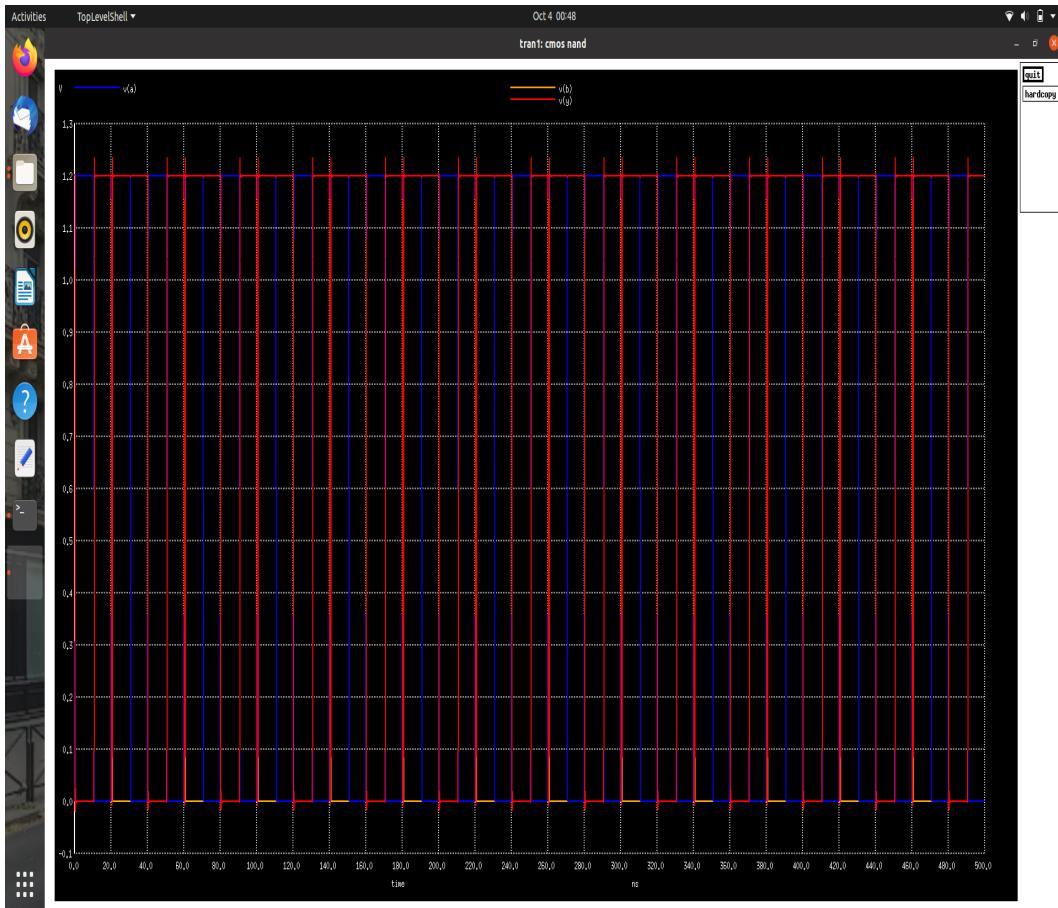
```
1 cmos nand
2 .include techfile130.txt
3
4 Mp1 y A Vdd Vdd pmos w=130n l=130n
5 Mp2 y B Vdd Vdd pmos w=130n l=130n
6 Mn1 y A n1 0 nnos w=130n l=130n
7 Mn2 n1 B 0 0 nnos w=130n l=130n
8
9 Vdd Vdd 0 1.2
10 VA A 0 pulse(0 1.2 0 0.5n 0.5n 10n 20n)
11 VB B 0 pulse(0 1.2 0 0.5n 0.5n 20n 40n)
12
13 .tran 0.1n 500n
14
15 .control
16 run
17
18 plot V(y) V(A) V(B)
19 .endc
20 .end
21
22
```

The terminal window is part of the "Text Editor" application in the "Activities" overview. The status bar at the bottom indicates "Loading file /home/neel/VLSI_Library/SCMOS/SCMOS_NAND/CMOS_NAND.drc" and shows options for "Plain Text", "Tab Width: 8", "Ln 1 Col 1", and "INS".

Layout

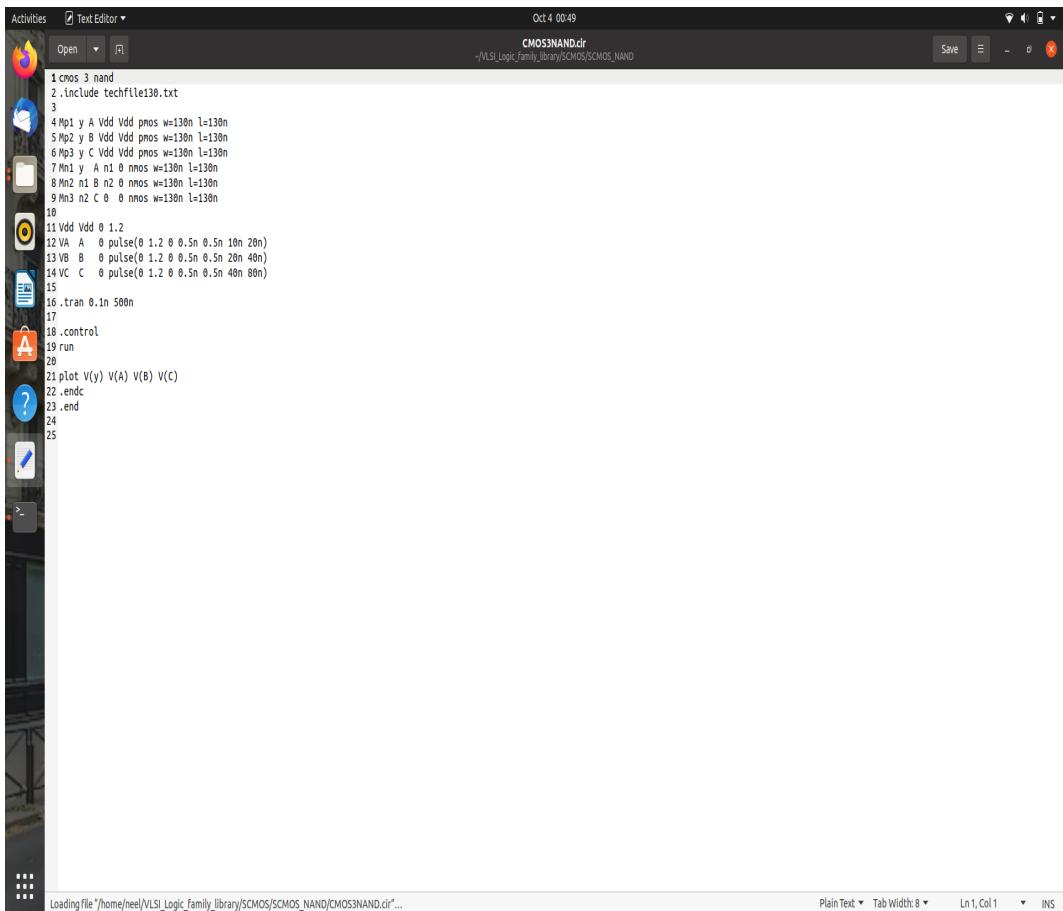


Simulation Plots



3. 3 i/p NAND

Netlist

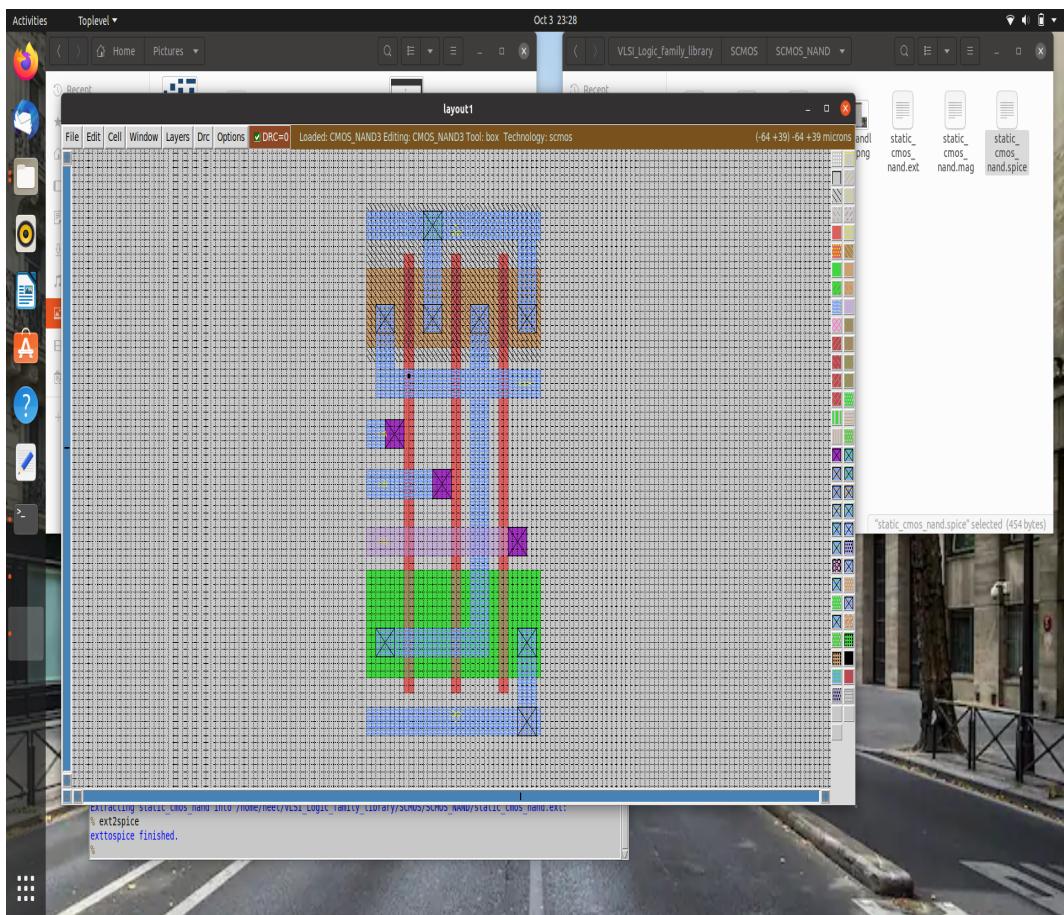


```
Oct 4 00:49
CMOS3NAND.drc
~/VLSI_Logic_Family_Library/SCMOS/SCMOS_NAND

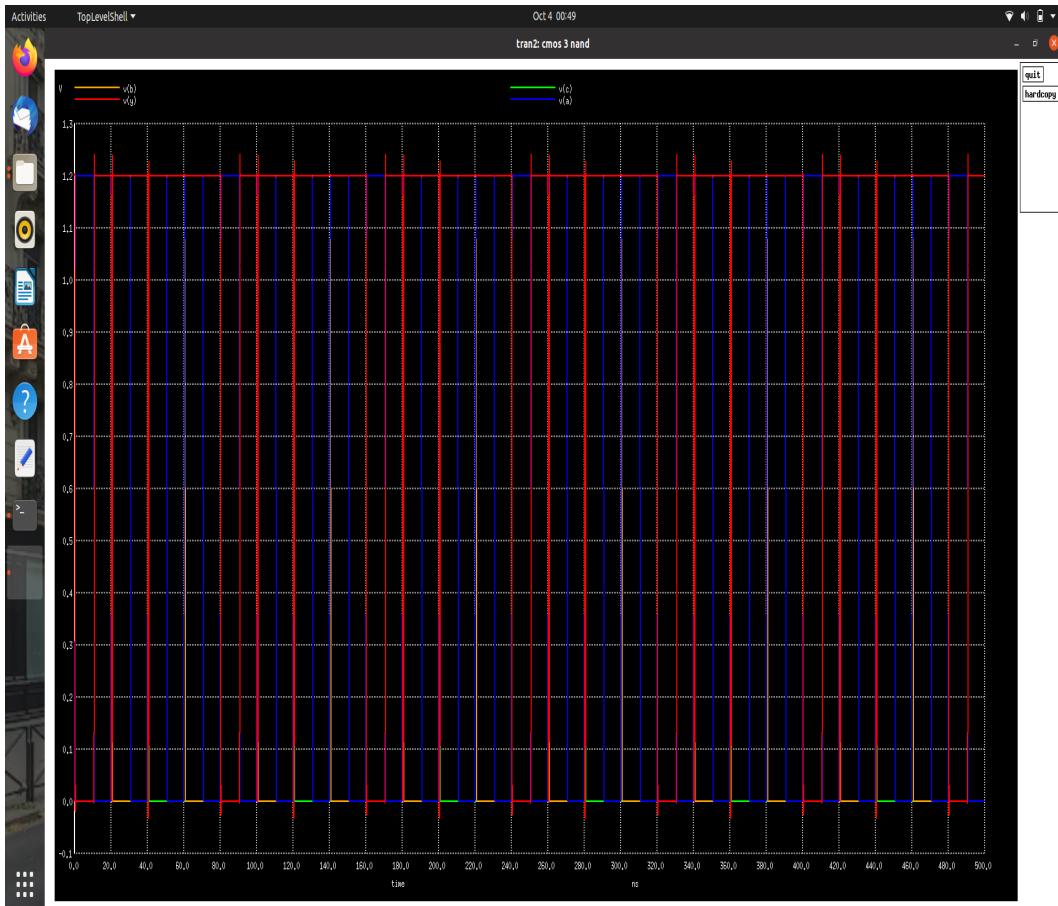
1 cmos 3 nand
2 .include techfile130.txt
3
4 Mp1 y A Vdd Vdd pmos w=130n l=130n
5 Mp2 y B Vdd Vdd pmos w=130n l=130n
6 M03 y C Vdd Vdd pmos w=130n l=130n
7 Mo1 y A n1 0 nmos w=130n l=130n
8 Mn2 n1 B n2 0 nmos w=130n l=130n
9 Mn3 n2 C 0 0 nmos w=130n l=130n
10
11 Vdd Vdd 0 1.2
12 VA A 0 pulse(0 1.2 0 0.5n 0.5n 10n 20n)
13 VB B 0 pulse(0 1.2 0 0.5n 0.5n 20n 40n)
14 VC C 0 pulse(0 1.2 0 0.5n 0.5n 40n 80n)
15
16 .tran 0.1n 500n
17
18 .control
19 run
20
21 plot V(y) V(A) V(B) V(C)
22 .endc
23 .end
24
25

Loading file ~/home/neel/VLSI_Logic_Family_Library/SCMOS/SCMOS_NAND/CMOS3NAND.cir"...
Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS
```

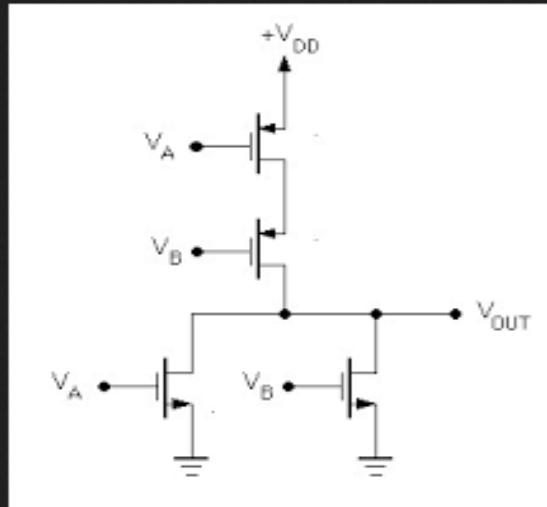
Layout



Simulation Plots



4. 2 i/p NOR



The NOR gate is a digital logic gate that implements logical NOR - it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results. NOR is the result of the negation of the OR operator. It can also in some senses be seen as the inverse of an AND gate. NOR is a functionally complete operation—NOR gates can be combined to generate any other logical function. It shares this property with the NAND gate. By contrast, the OR operator is monotonic as it can only change LOW to HIGH but not vice versa.

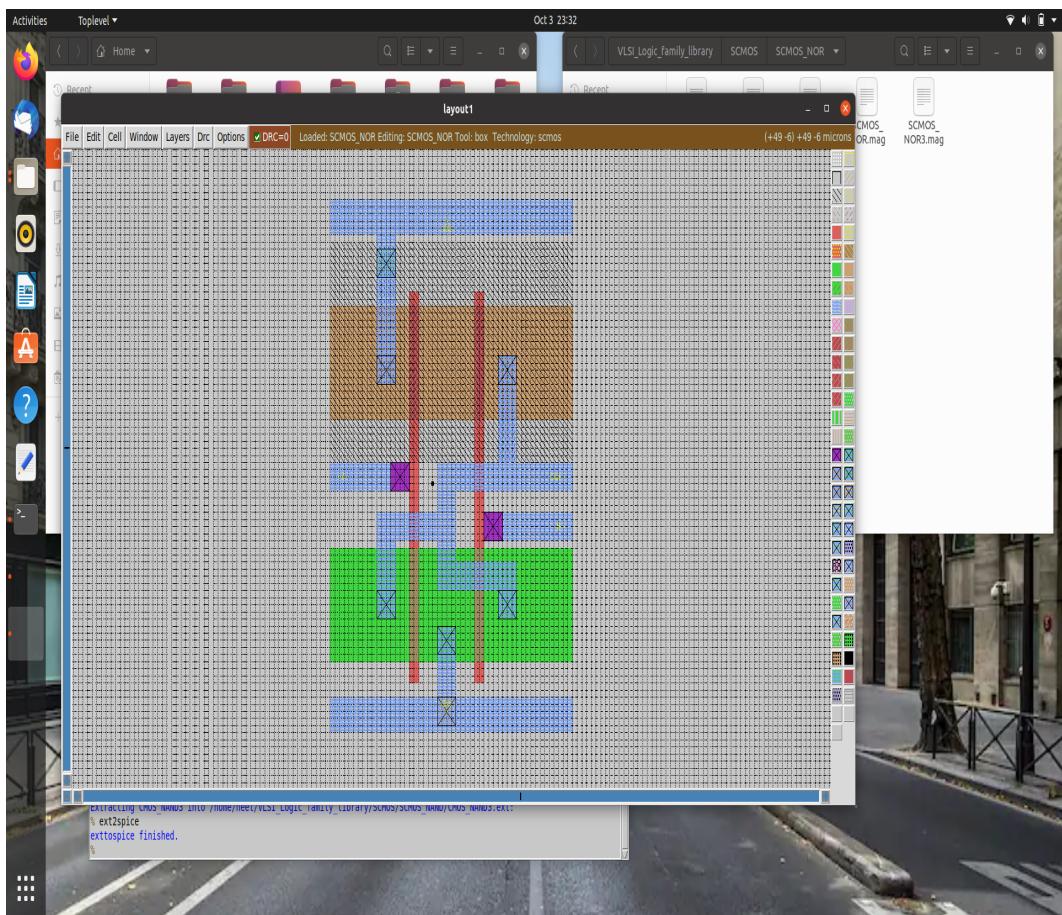
Netlist

The screenshot shows a terminal window titled "Text Editor" with the file "lab2CMOSNOR.cir" open. The terminal window has a dark theme with light-colored text. The netlist code is as follows:

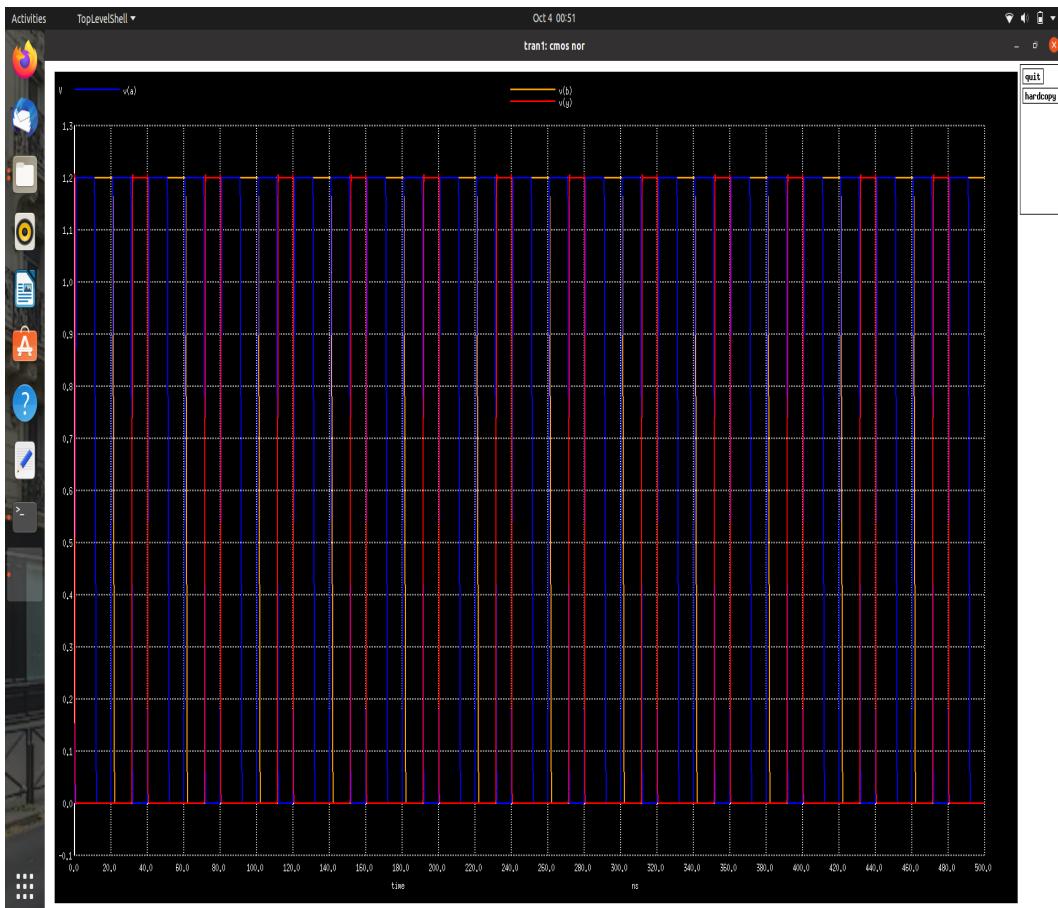
```
1 CMOS NOR
2 .include techfile130.txt
3
4 M1 n1 A Vdd Vdd nmos w=130n l=130n
5 M2 Y B n1 Vdd nmos w=130n l=130n
6 M1 Y A 0 0 nmos w=130n l=130n
7 M2 Y B 0 0 nmos w=130n l=130n
8 Vdd Vdd 0 1.2
9 VA A B Pulse(0 1.2 0 in in 10n 20n)
10 VB B 0 Pulse(0 1.2 0 in in 20n 40n)
11
12 .tran 0.1n 500n
13 .control
14 run
15
16 plot V(y) V(A) V(B)
17 .endc
18 .end
```

The terminal window also displays the message "Loading file /home/neel/VLSI_Library/SCMOS/SCMOS_NOR/lab2CMOSNOR.cir..." at the bottom.

Layout

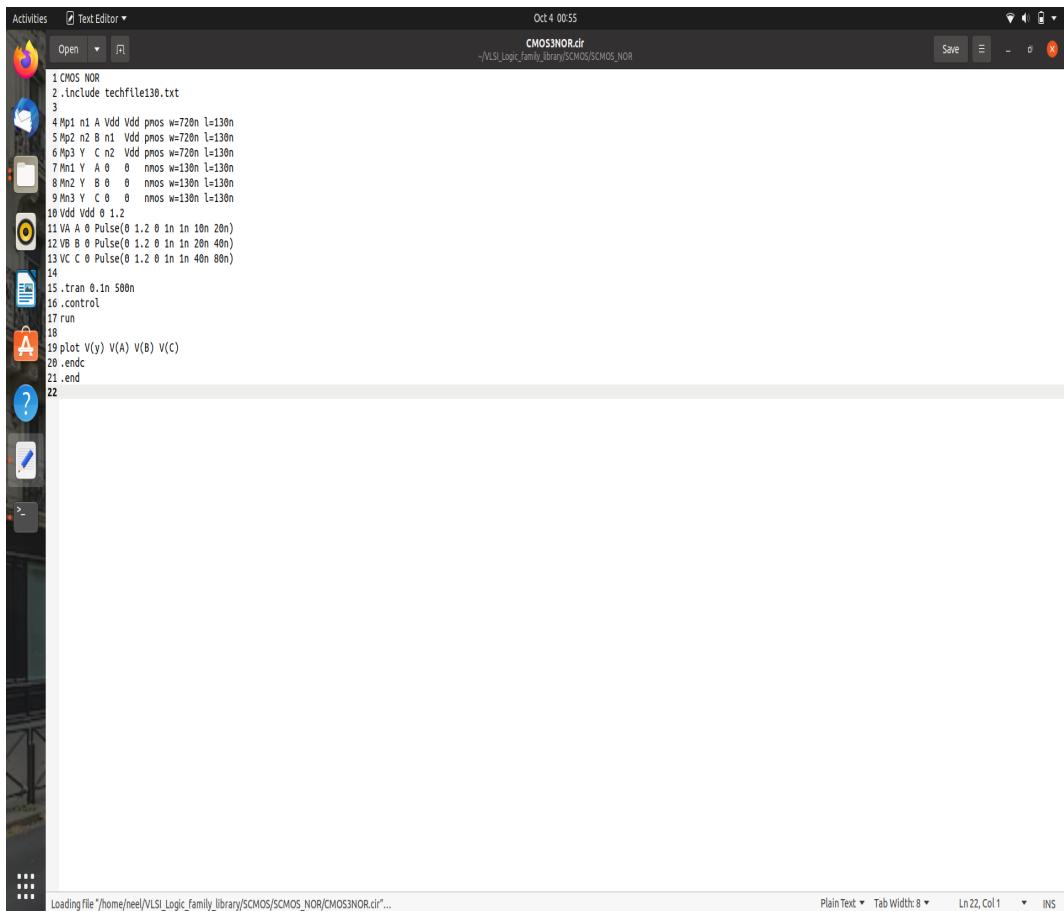


Simulation Plots



5. 3 i/p NOR

Netlist



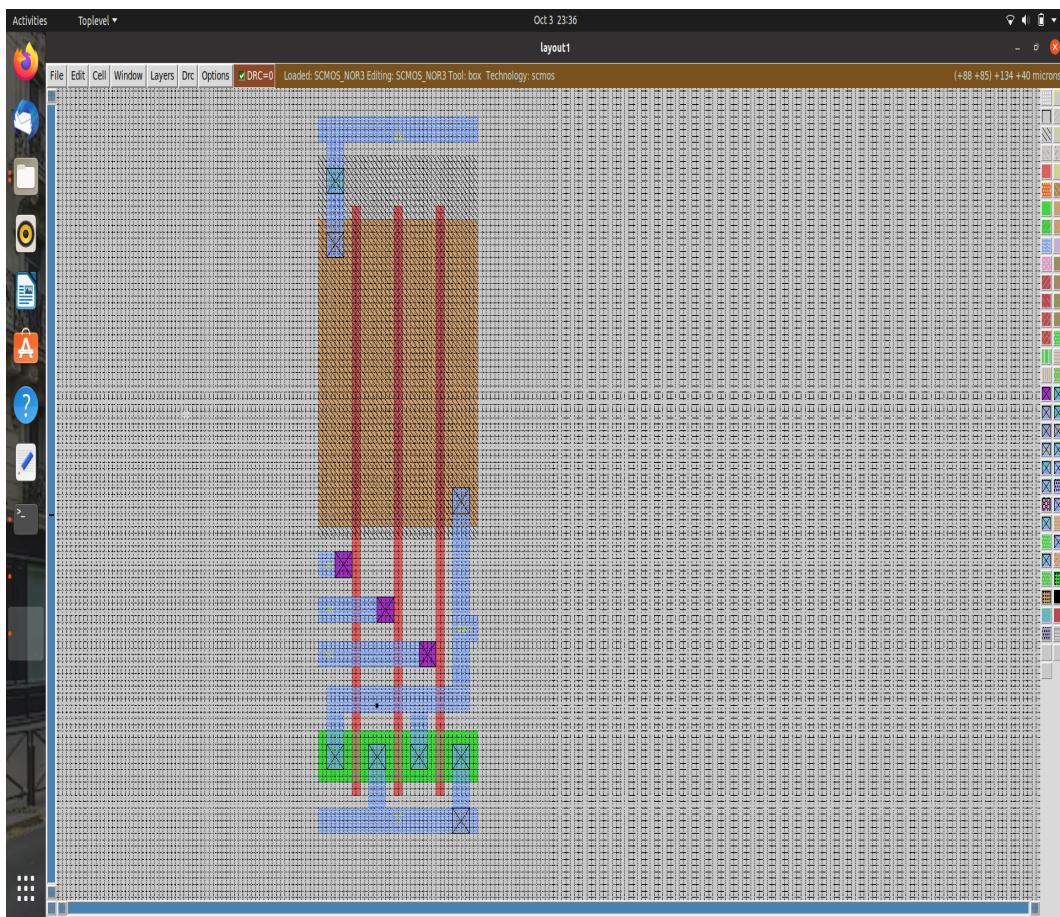
```
Activities Text Editor Oct 4 00:55
CMOS3NOR.cir
-/VLSI_Logic_Family_library/SCMOS/SCMOS_NOR

1 CMOS NOR
2 .include techfile130.txt
3
4 Mp1 n1 A Vdd Vdd pmos w=720n l=130n
5 Mp2 n2 B n1 Vdd pmos w=720n l=130n
6 Mo3 Y C n2 Vdd pmos w=720n l=130n
7 Mo1 Y A 0 0 nmos w=130n l=130n
8 Mo2 Y B 0 0 nmos w=130n l=130n
9 Mo3 Y C 0 0 nmos w=130n l=130n
10 Vdd Vdd 0 1.2
11 VA A 0 Pulse(0 1.2 0 1n 1n 10n 20n)
12 VB B 0 Pulse(0 1.2 0 1n 20n 40n)
13 VC C 0 Pulse(0 1.2 0 1n 40n 80n)
14
15 .tran 0.1n 500n
16 .control
17 run
18
19 plot V(y) V(A) V(B) V(C)
20 .endc
21 .end
22
```

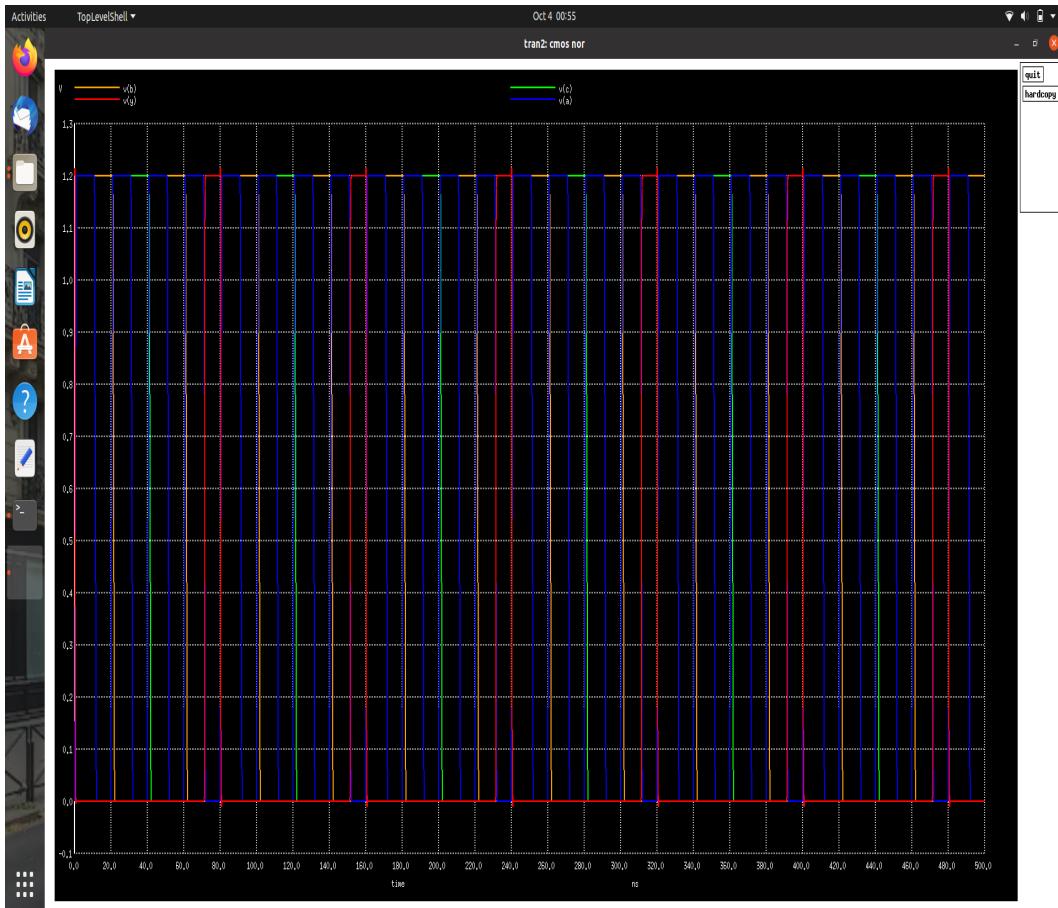
Loading file "/home/neel/VLSI_Library/SCMOS/SCMOS_NOR/CMOS3NOR.cir"...

Plain Text ▾ Tab Width: 8 ▾ Ln 22, Col 1 ▾ INS

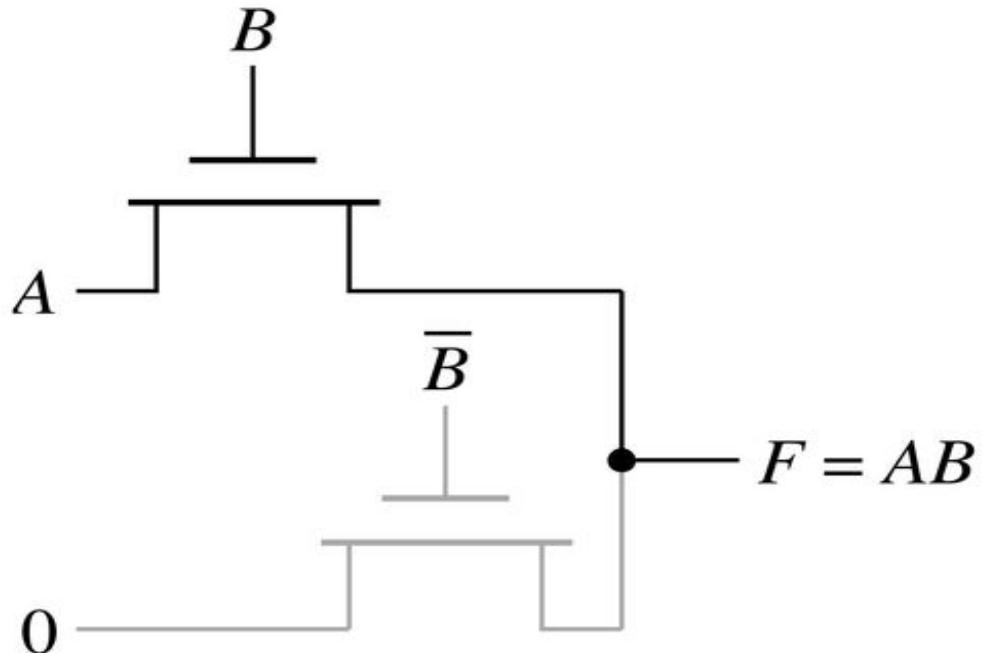
Layout



Simulation Plots

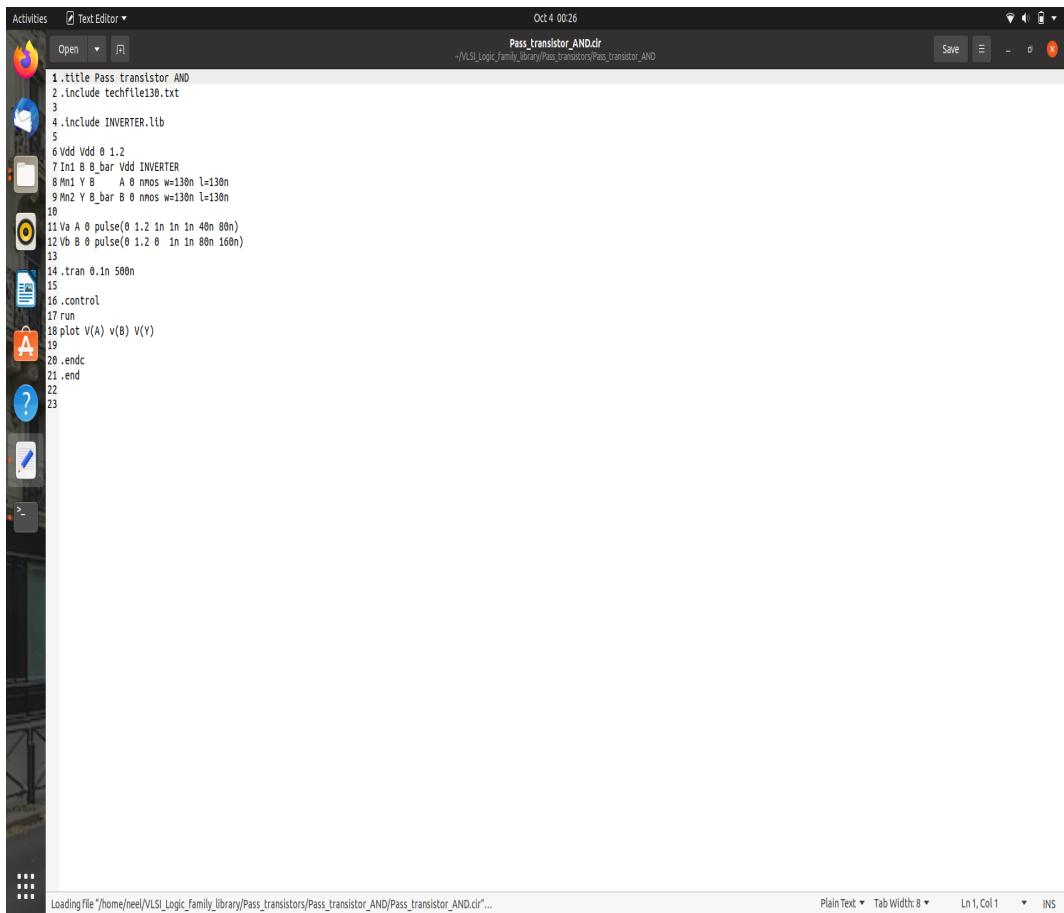


6. Pass Transistor AND



A pass transistor takes input in both source and gate unlike cmos which takes only gate voltage as input. Pass logic Requires 4 logic gates (needs an inverter) less than CMOS logic would require 6 logic gates. The gate can be static.pass AND takes A and B in nmos1 drain and gate and B and \bar{B} in nmos2 drain and gate.

Netlist



The screenshot shows a terminal window titled "Text Editor" with the file "Pass_transistor_AND.cir" open. The terminal interface includes a toolbar with icons for Open, Save, and Close, and a status bar at the bottom. The code in the terminal is as follows:

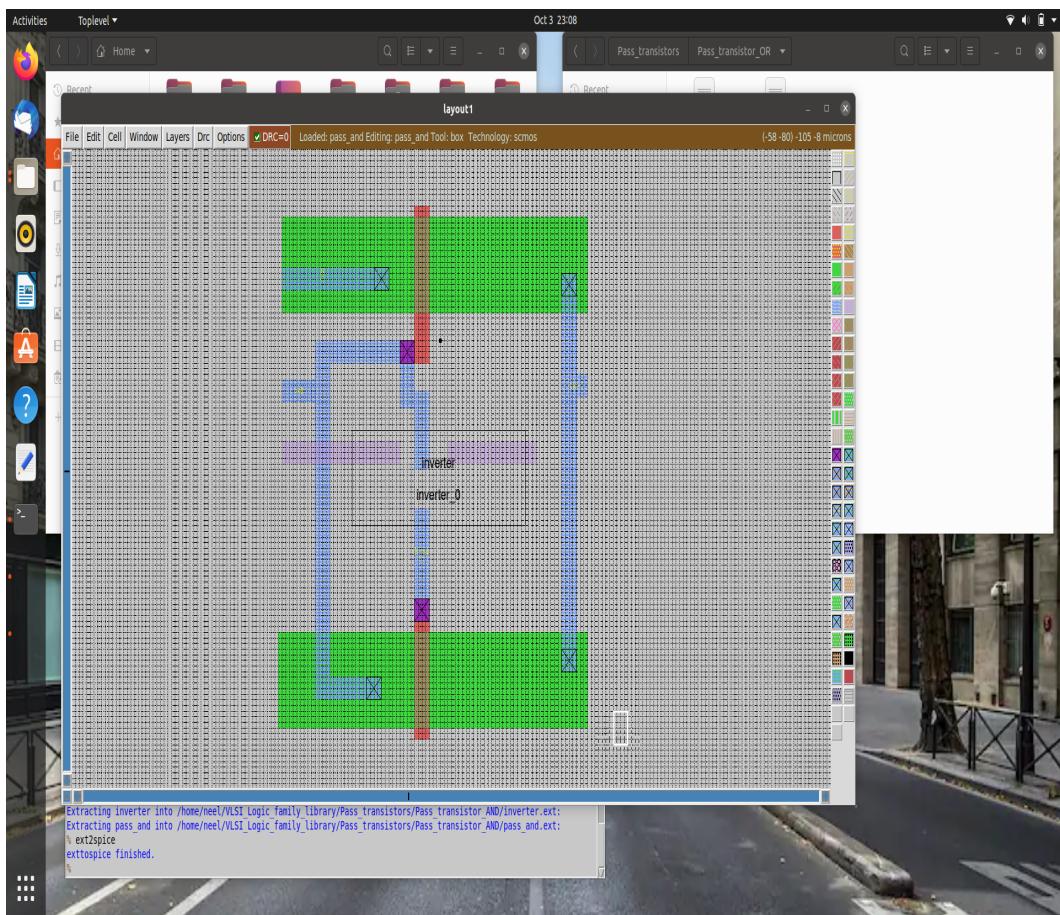
```
Oct 4 00:26
Pass_transistor_AND.cir
-/VLSI/Logic_family/library/Pass_transistors/Pass_transistor_AND

1 .title Pass transistor AND
2 .include techfile130.txt
3
4 .include INVERTER.lib
5
6 Vdd Vdd 0 1.2
7 M1 Y B _bar Vdd INVERTER
8 M1 Y B 0 nmos w=130n l=130n
9 M2 Y B_bar B 0 nmos w=130n l=130n
10
11 Va A 0 pulse(0 1.2 1n 1n 40n 80n)
12 Vb 0 0 pulse(0 1.2 0 1n 80n 160n)
13
14 .tran 0.1n 500n
15
16 .control
17 run
18 plot V(A) v(B) V(Y)
19
20 .endc
21 .end
22
23

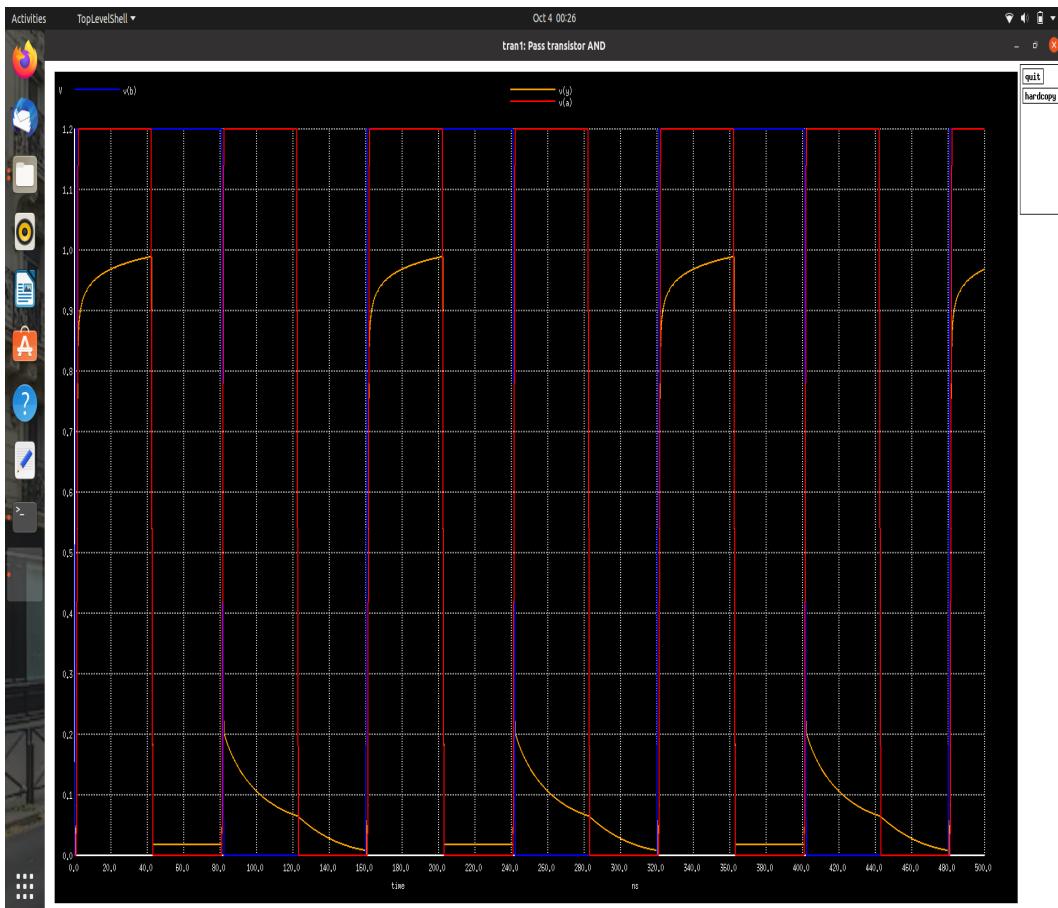
Loading file "/home/neel/VLSI_Logic_family/library/Pass_transistors/Pass_transistor_AND/Pass_transistor_AND.cir"...
```

The status bar at the bottom right shows "Plain Text" and "Tab Width: 8".

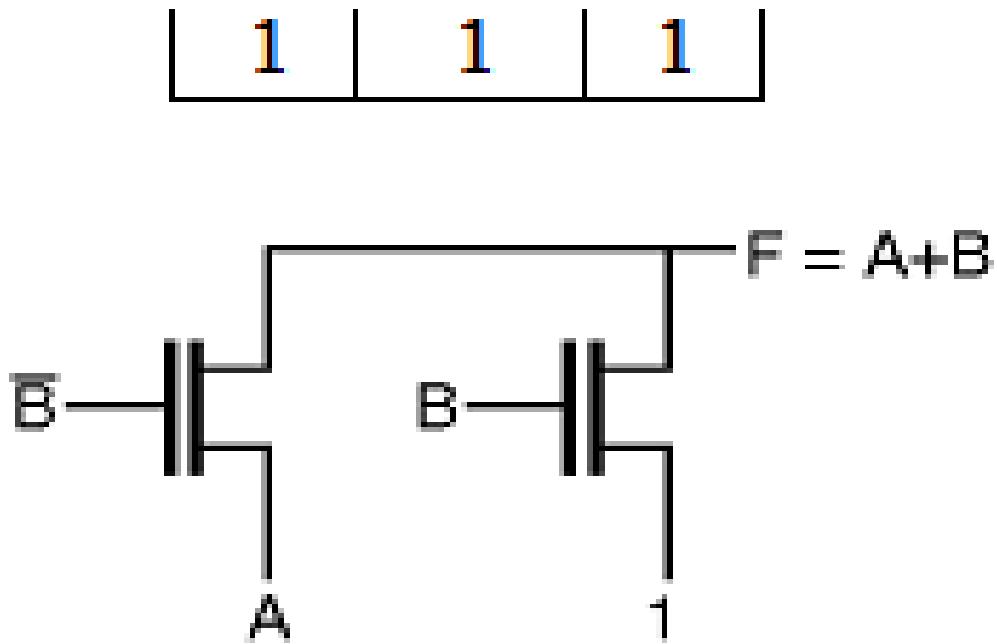
Layout



Simulation Plots



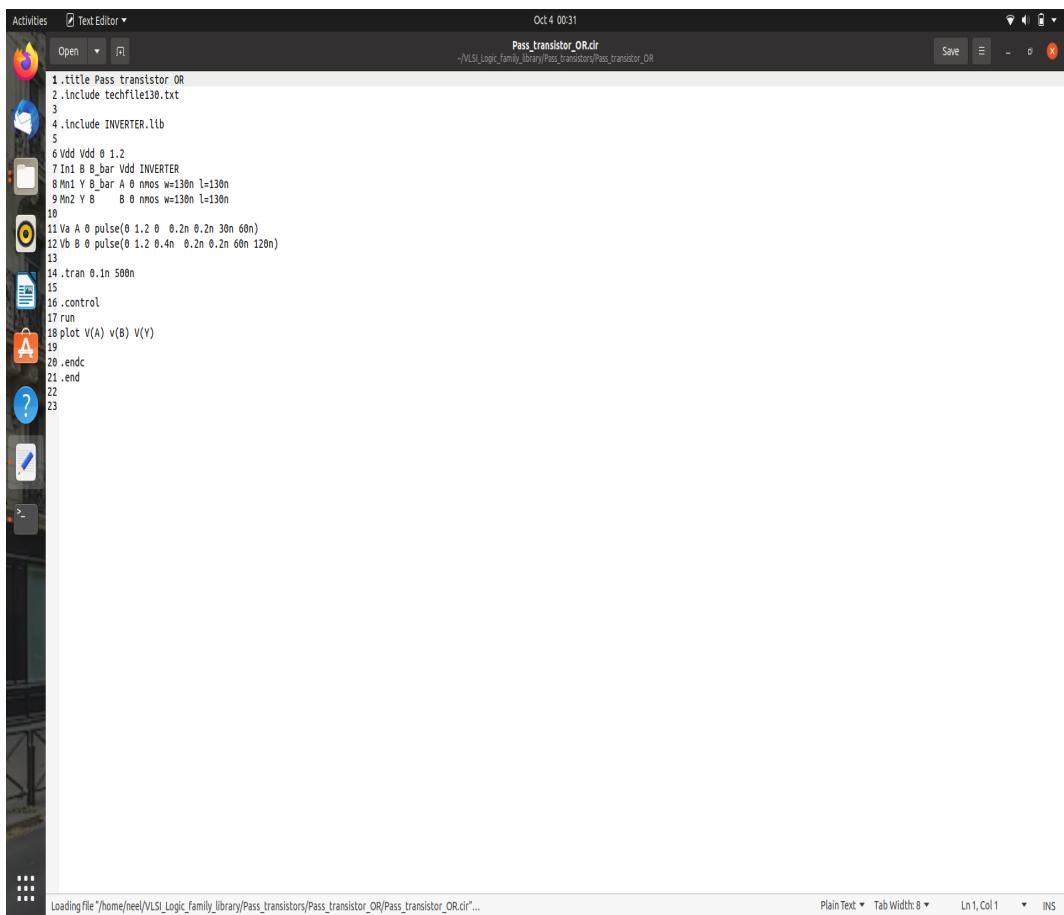
7. Pass Transistor OR



‘OR’ gate using pass transistor

A pass transistor takes input in both source and gate unlike CMOS which takes only gate voltage as input. Pass logic Requires 4 logic gates (needs an inverter) less than CMOS logic would require 6 logic gates. The gate can be static. pass OR takes A and B_b in nmos 1 drain and gate and B_d in nmos 2 drain and gate.

Netlist



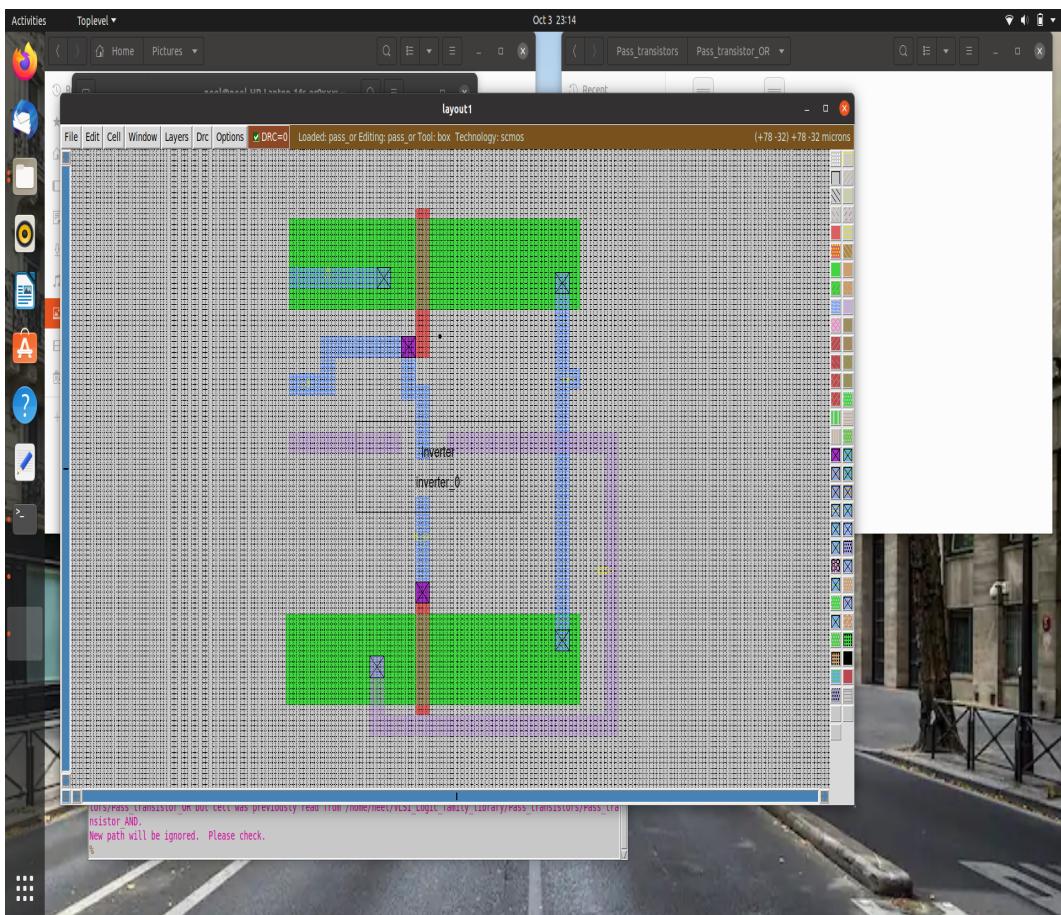
The screenshot shows a terminal window titled "Text Editor" with the file "Pass_transistor_OR.cir" open. The window displays a netlist for a Pass transistor OR circuit. The code includes definitions for a title block, component includes, power supply Vdd, and two NMOS transistors M1 and M2. It also specifies pulse inputs A and B, a tran analysis, and a plot command. The terminal status bar at the bottom indicates the file is loading and shows options for Plain Text, Tab Width, Line, Column, and Insert mode.

```
Oct 4 00:31
Pass_transistor_OR.cir
~/VLSI_Logic_Family/Library/Pass_Transistors/Pass_transistor_OR

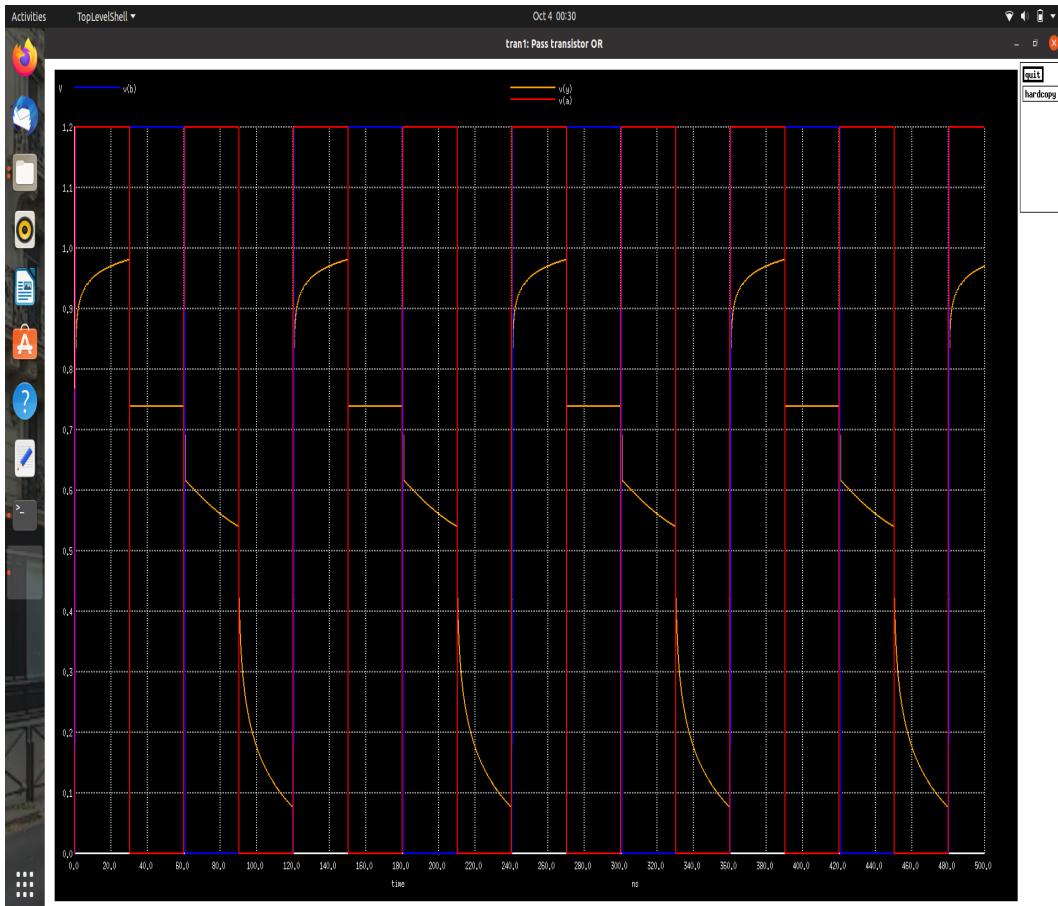
1 .title Pass transistor OR
2 .include techfile130.txt
3
4 .include INVERTER.lib
5
6 Vdd Vdd 0 1.2
7 M1 Y B_bar B Vdd INVERTER
8 M1 Y B_bar B 0 nmos w=130n l=130n
9 M2 Y B B 0 nmos w=130n l=130n
10
11 Va A 0 pulse(0 1.2 0 0.2n 0.2n 30n 60n)
12 Vb B 0 pulse(0 1.2 0.4n 0.2n 0.2n 60n 120n)
13
14 .tran 0.1n 500n
15
16 .control
17 run
18 plot V(A) V(B) V(Y)
19
20 .endc
21 .end
22
23

Loading file "/home/neel/VLSI_Logic_Family/Library/Pass_Transistors/Pass_transistor_OR/Pass_transistor_OR.cir" ...
Plain Text ▾ Tab Width: 8 ▾ Ln 1 Col 1 ▾ INS
```

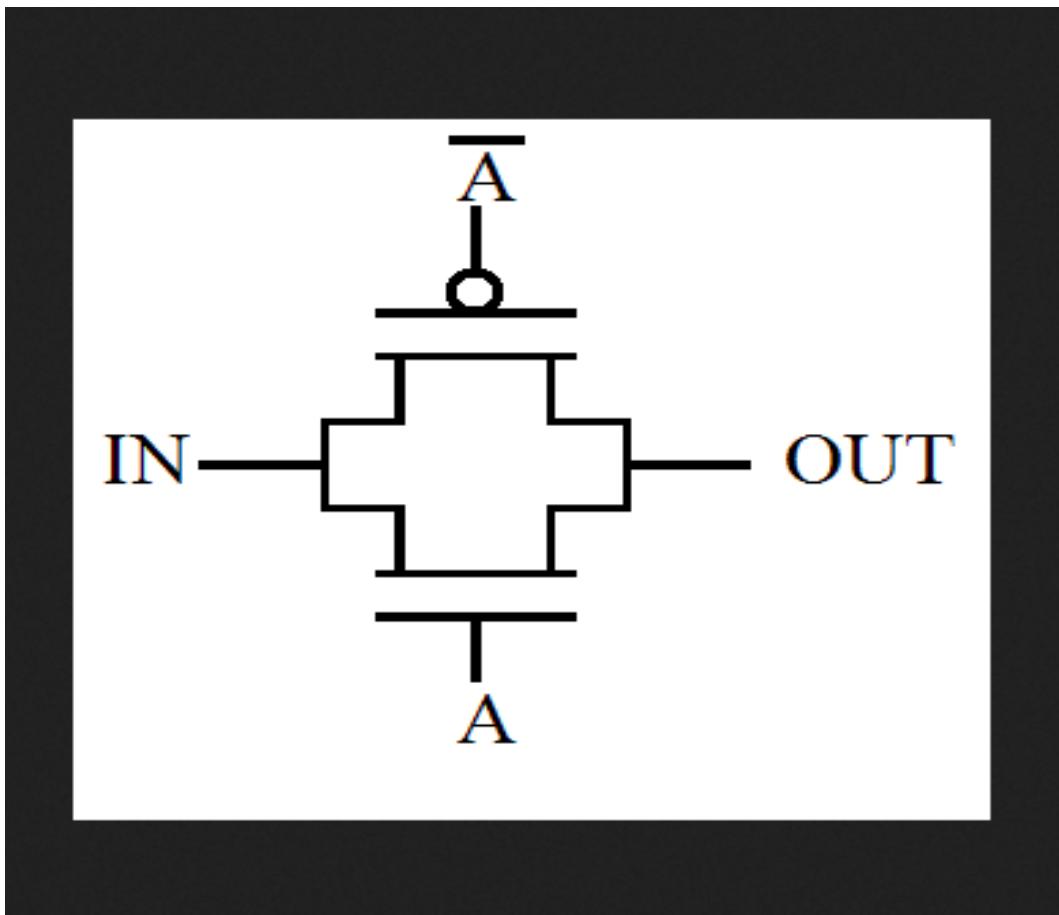
Layout



Simulation Plots

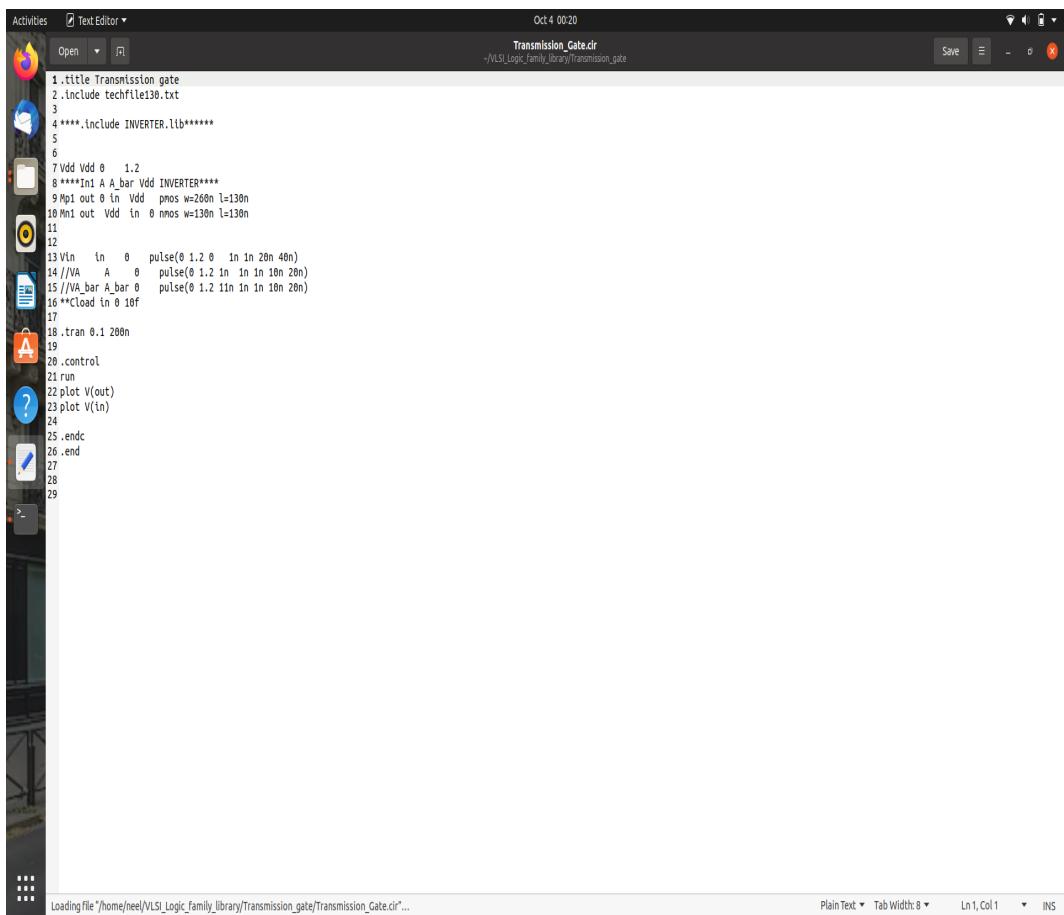


8. Transmission Gate



A transmission gate (TG) is an analog gate similar to a relay that can conduct in both directions or block by a control signal with almost any voltage potential.[1] It is a CMOS-based switch, in which PMOS passes a strong 1 but poor 0, and NMOS passes strong 0 but poor 1. Both PMOS and NMOS work simultaneously.

Netlist

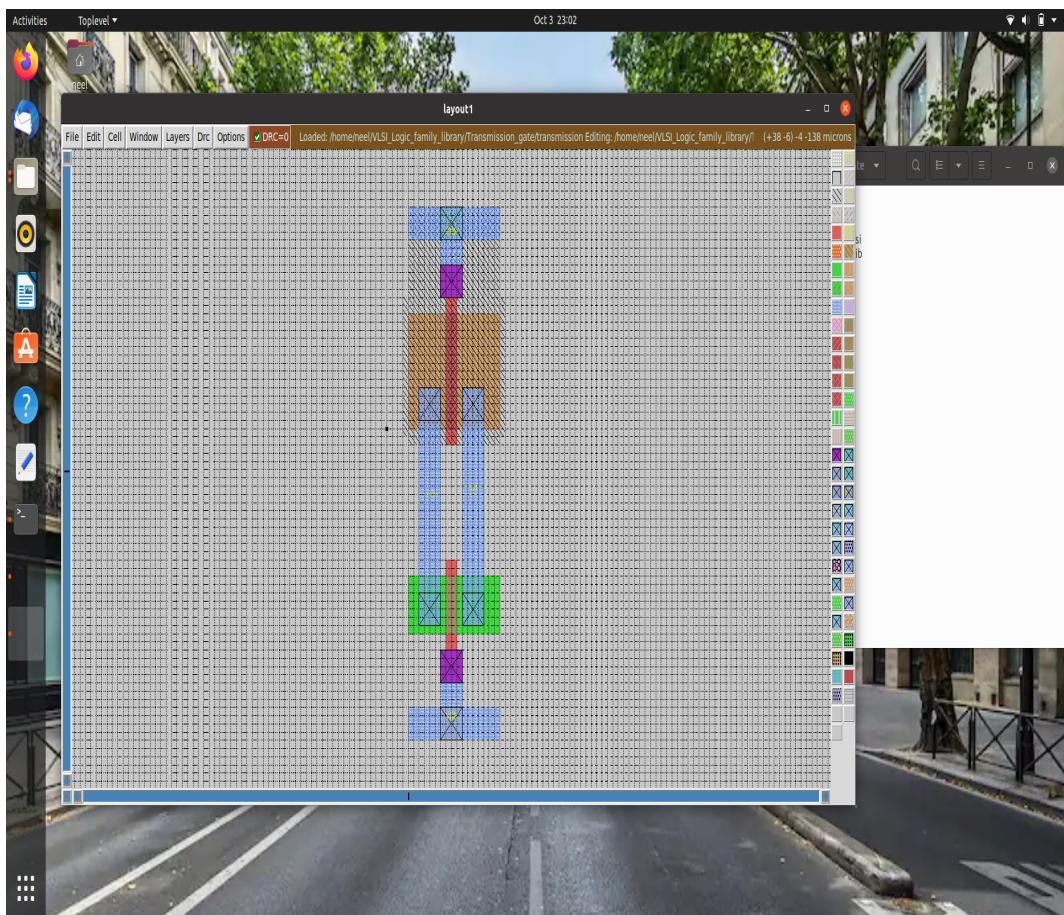


```
Activities Text Editor Oct 4 00:20
Transmission_Gate.cir
~/VLSI_Logic_Family_Library/Transmission_gate
Save X

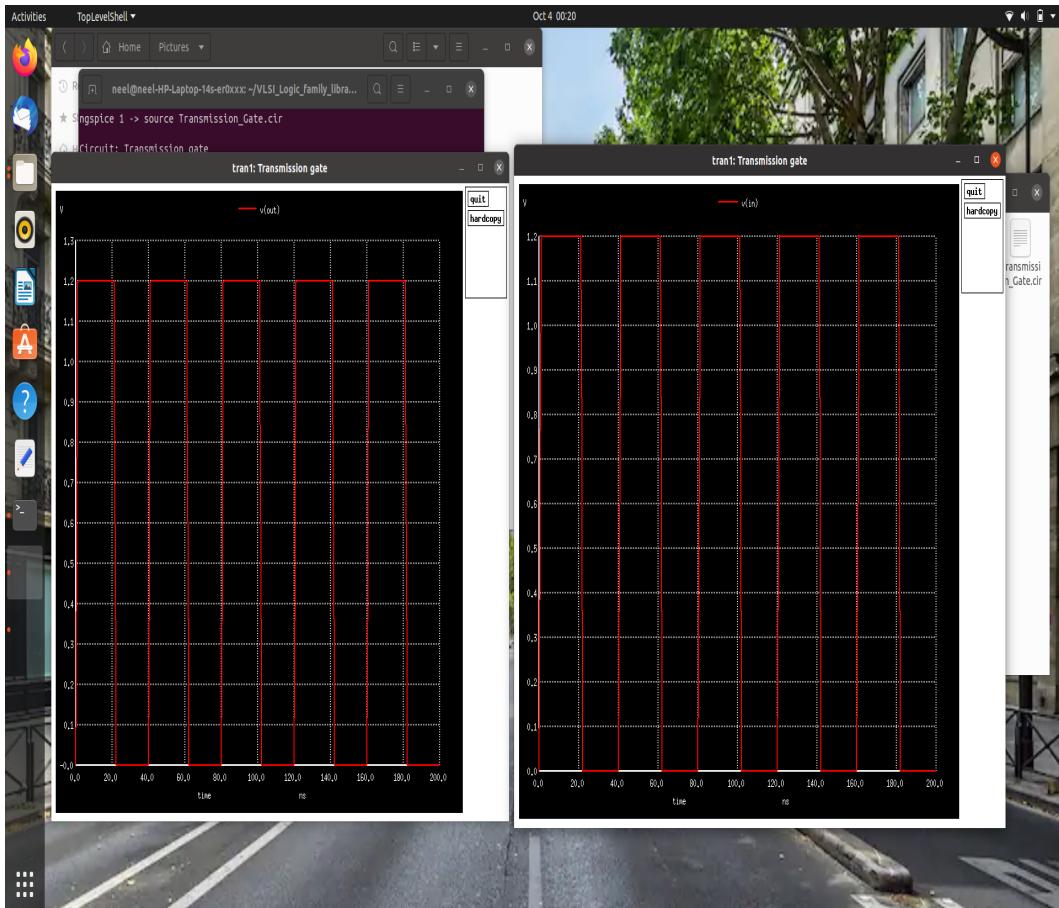
1 .title Transmission gate
2 .include techfile130.txt
3
4 ***.include INVERTER.lib*****
5
6
7 Vdd Vdd 0 1.2
8 ***.Init A A_bar Vdd INVERTER****
9 M1 out 0 in Vdd phos w=20n l=130n
10 M1 out Vdd in 0 nmos w=130n l=130n
11
12
13 Vin in 0 pulse(0 1.2 0 1n 20n 40n)
14 /VA A 0 pulse(0 1.2 1n 1n 10n 20n)
15 /VA_bar A_bar 0 pulse(0 1.2 11n 1n 10n 20n)
16 *Cload in 0 10f
17
18 .tran 0.1 200n
19
20 .control
21 run
22 plot V(out)
23 plot V(in)
24
25 .endc
26 .end
27
28
29

Loading file "/home/neel/VLSI_Logic_Family_Library/Transmission_gate/Transmission_Gate.cir"...
Plain Text Tab Width: 8 Ln 1 Col 1 INS
```

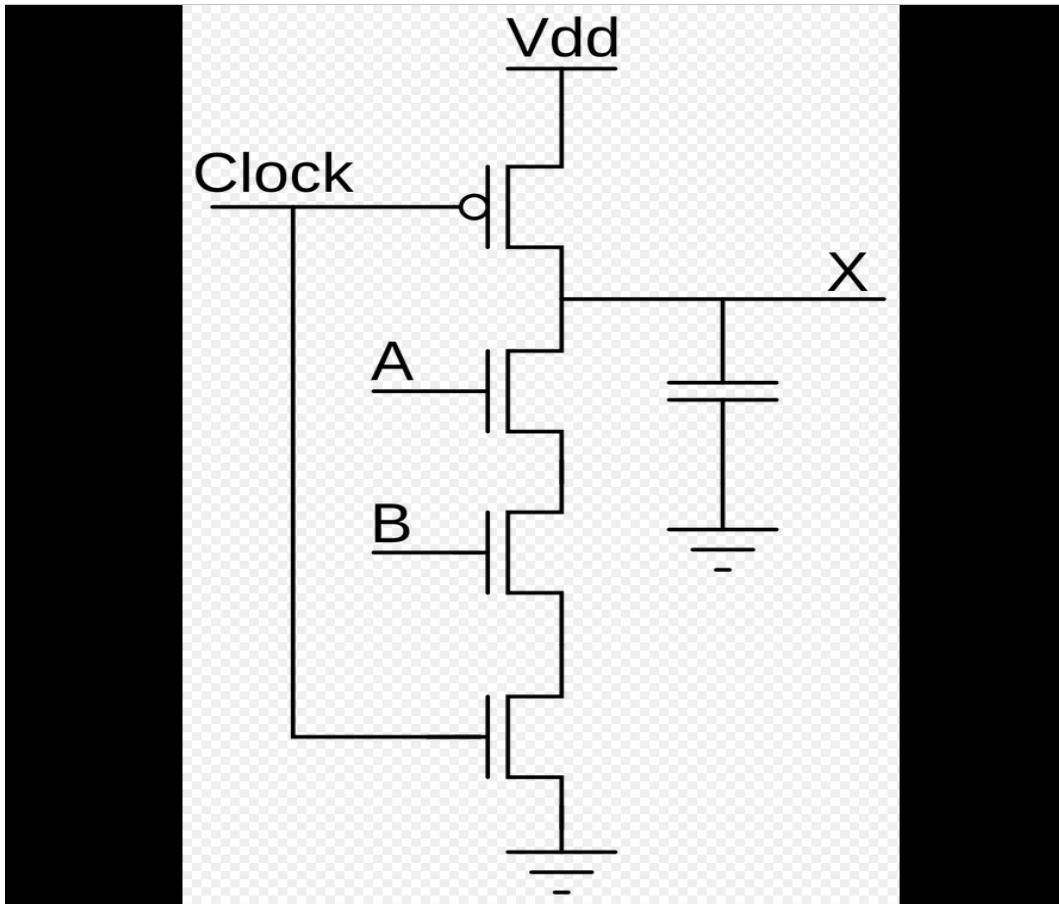
Layout



Simulation Plots



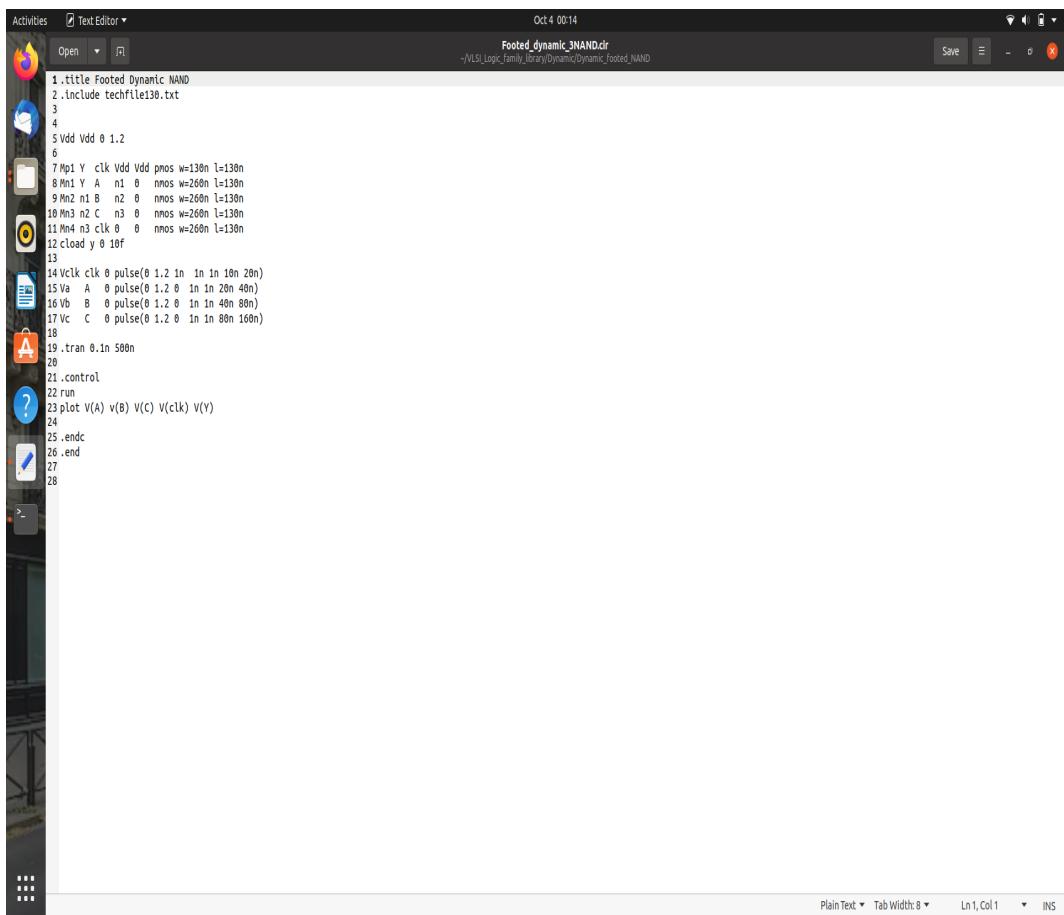
9. Footed Dynamic NAND gate



During the evaluation phase, Clock is high. If A and B are also high, the output will be pulled low. Otherwise, the output stays high (due to the load capacitance). The dynamic logic circuit requires two phases. The first phase, when Clock is low, is called the setup phase or the precharge phase and the second phase, when Clock is high, is called the evaluation phase. In the setup phase, the output is driven high unconditionally (no matter the values of the inputs A and B). The capacitor, which represents the load capacitance of this gate, becomes charged. Because the transistor at the bottom is turned off, it is impossible for the output to be driven low during this phase.

During the evaluation phase, Clock is high. If A and B are also high, the output will be pulled low. Otherwise, the output stays high (due to the load capacitance).

Netlist

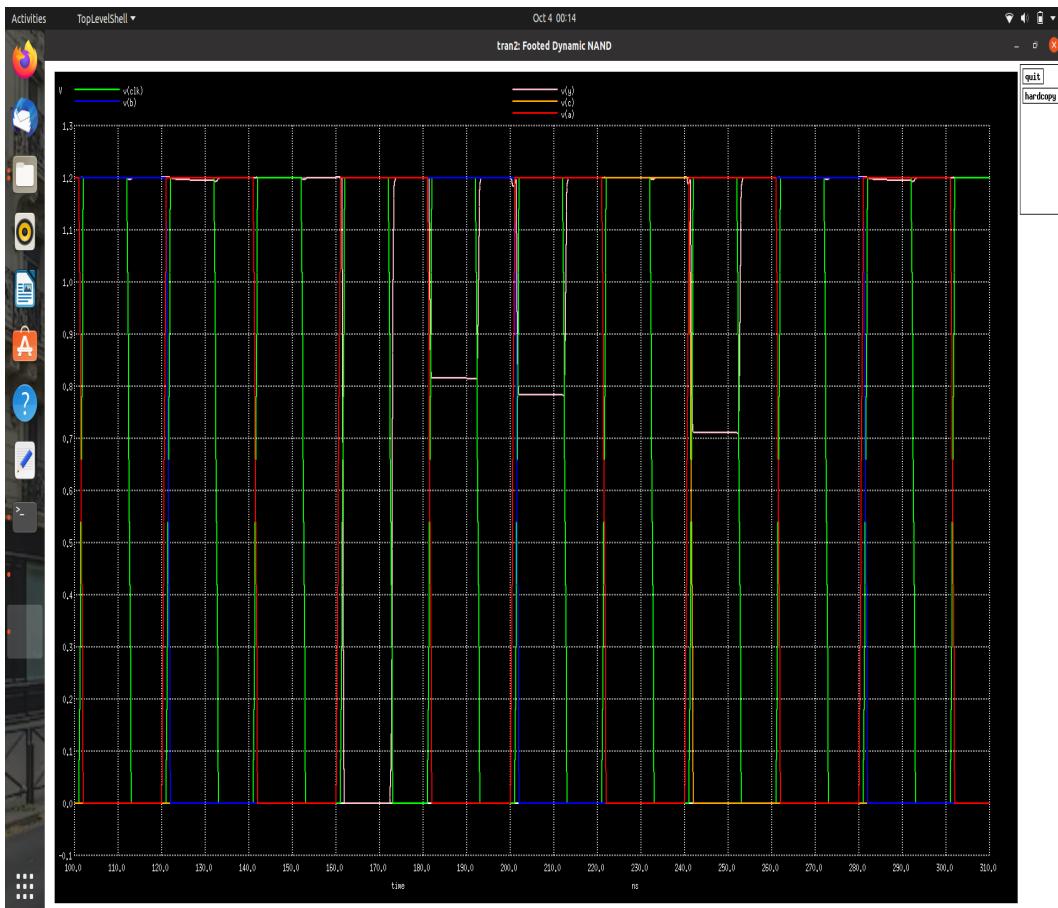


A screenshot of a Linux desktop environment showing a terminal window titled "Text Editor". The window contains a netlist script for a 3NAND gate. The script includes definitions for power supply nodes (Vdd, Vss), logic gates (M1-M4), and control signals (clk, Y). It also includes pulse definitions for inputs A, B, and C, and a plot command to visualize the output Y over time.

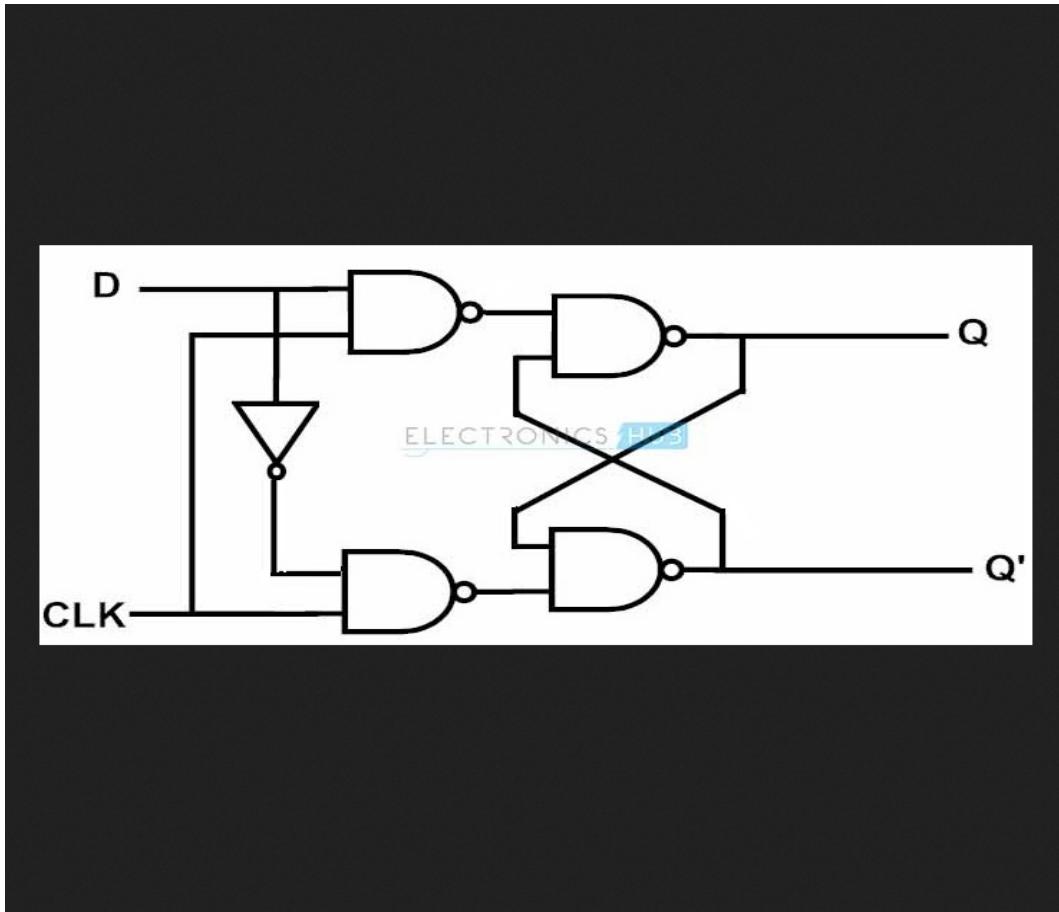
```
Activities Text Editor Oct 4 00:14
Footed_dynamic_3NAND.dir
~/VLSI_Logic_Family/Library/Dynamic/Dynamic_footed_NAND
Save X

1 .title Footed Dynamic NAND
2 .include techfile130.txt
3
4
5 Vdd Vdd 0 1.2
6
7 M1 Y clk Vdd Vdd pmos w=130n l=130n
8 M1 Y A n1 0 nmos w=260n l=130n
9 M2 n1 B n2 0 nmos w=260n l=130n
10 M3 n2 C n3 0 nmos w=260n l=130n
11 M4 n3 clk 0 0 nmos w=260n l=130n
12 cloud y 0 10f
13
14 Vclk clk 0 pulse(0 1.2 0 1n 1n 10n 20n)
15 Va A 0 pulse(0 1.2 0 1n 20n 40n)
16 Vb B 0 pulse(0 1.2 0 1n 40n 80n)
17 Vc C 0 pulse(0 1.2 0 1n 80n 160n)
18
19 .tran 0.1n 500n
20
21 .control
22 run
23 plot V(A) V(B) V(C) V(clk) V(Y)
24
25 .endc
26 .end
27
28
```

Simulation Plots

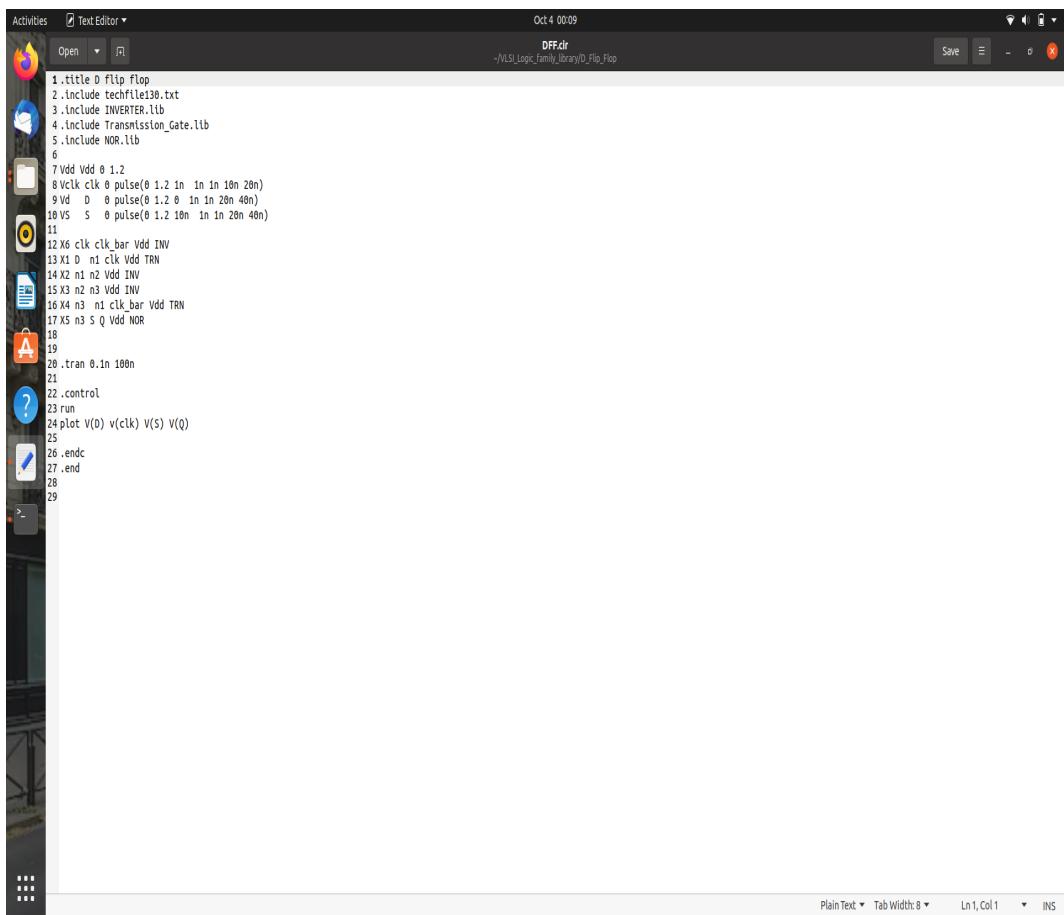


10. Dynamic Flip Flop with Reset



The D-Type Flip-Flop with Set/Reset models a generic clocked data-type Flip-Flop with either asynchronous or synchronous set and reset inputs. The Q and QN outputs can change state only on the specified clock edge unless the asynchronous set or reset is asserted. The clock edge trigger can be set with the Trigger Condition parameter to be either rising edge ($0_T O_1$) or falling edge ($1_T O_0$). If set and reset inputs are not required, the D - Type Flip - Flop can be used.

Netlist

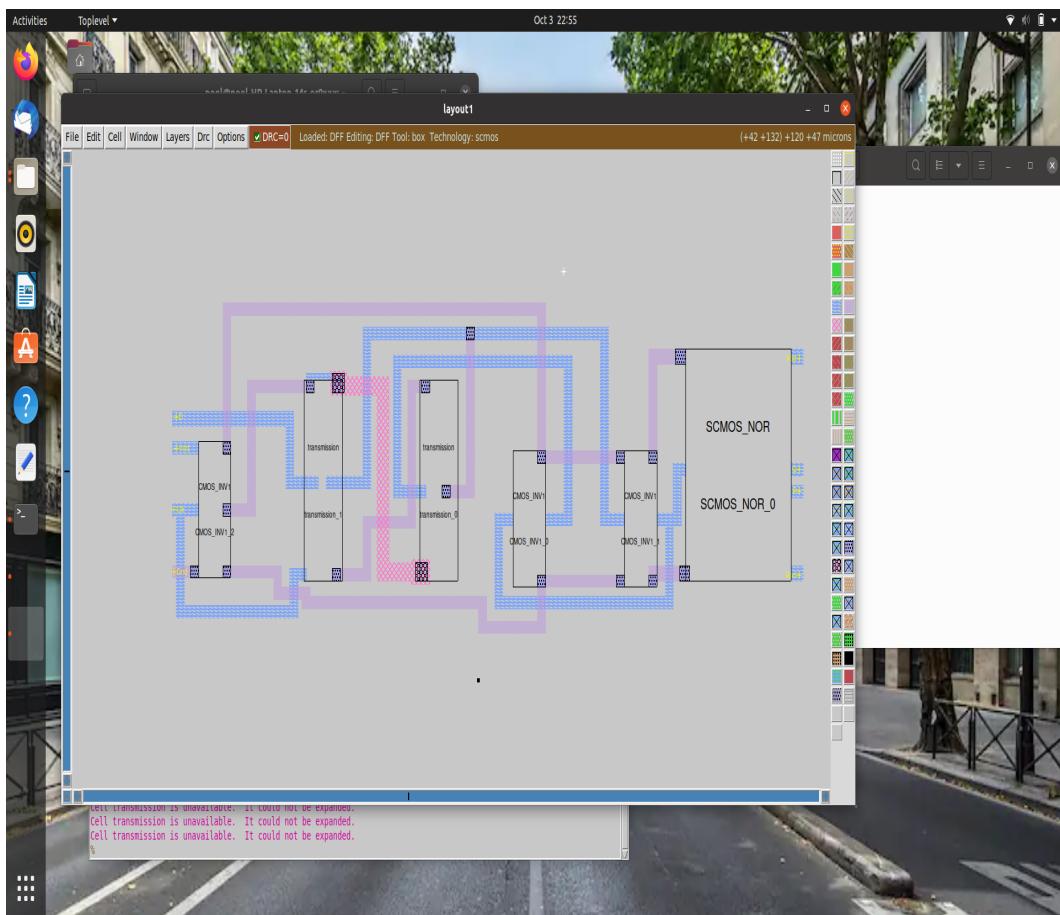


The screenshot shows a terminal window titled "TextEditor" with the file "DFF.v" open. The terminal window has a dark theme and includes a dock on the left side with icons for various applications like a browser, file manager, and terminal. The terminal content is a Verilog-like netlist for a D flip-flop:

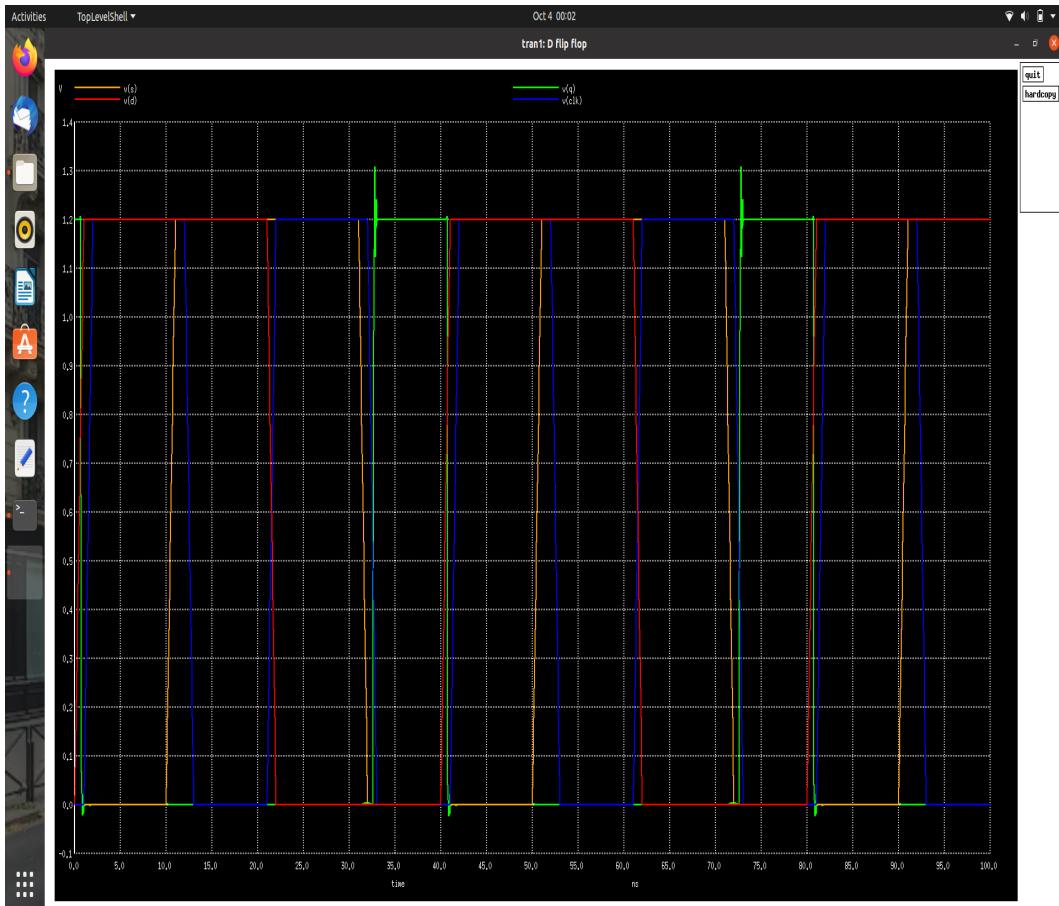
```
1 .title D flip flop
2 .include techfile130.txt
3 .include INVERTER.lib
4 .include Transmission_Gate.lib
5 .include NOR.lib
6
7 Vdd Vdd 0 1.2
8 Vclk clk 0 pulse(0 1.2 in in in 10n 20n)
9 Vd D 0 pulse(0 1.2 0 in in 20n 40n)
10 VS S 0 pulse(0 1.2 10n in in 20n 40n)
11
12 X6 clk clk_bar Vdd INV
13 X1 D n1 clk Vdd TRN
14 X2 n1 n2 Vdd INV
15 X3 n2 n3 Vdd INV
16 X4 n3 n1 clk_bar Vdd TRN
17 X5 n3 S Q Vdd NOR
18
19
20 .tran 0.1n 100n
21
22 .control
23 run
24 plot V(D) v(clk) V(S) V(Q)
25
26 .endc
27 .end
28
29
```

The terminal also shows status information at the bottom: "Plain Text" and "Tab Width: 8".

Layout



Simulation Plots



END