



INDIAN INSTITUTE OF  
INFORMATION  
TECHNOLOGY

at(1.5)

# VLSI Assignment

Vth Semester

EC301: Introduction to VLSI

*Aalekh Prasad*  
*20BEC001*

Under the guidance of  
Asst. Prof. Dr. Jagadish D N  
Date: October 2, 2022

---

## Contents

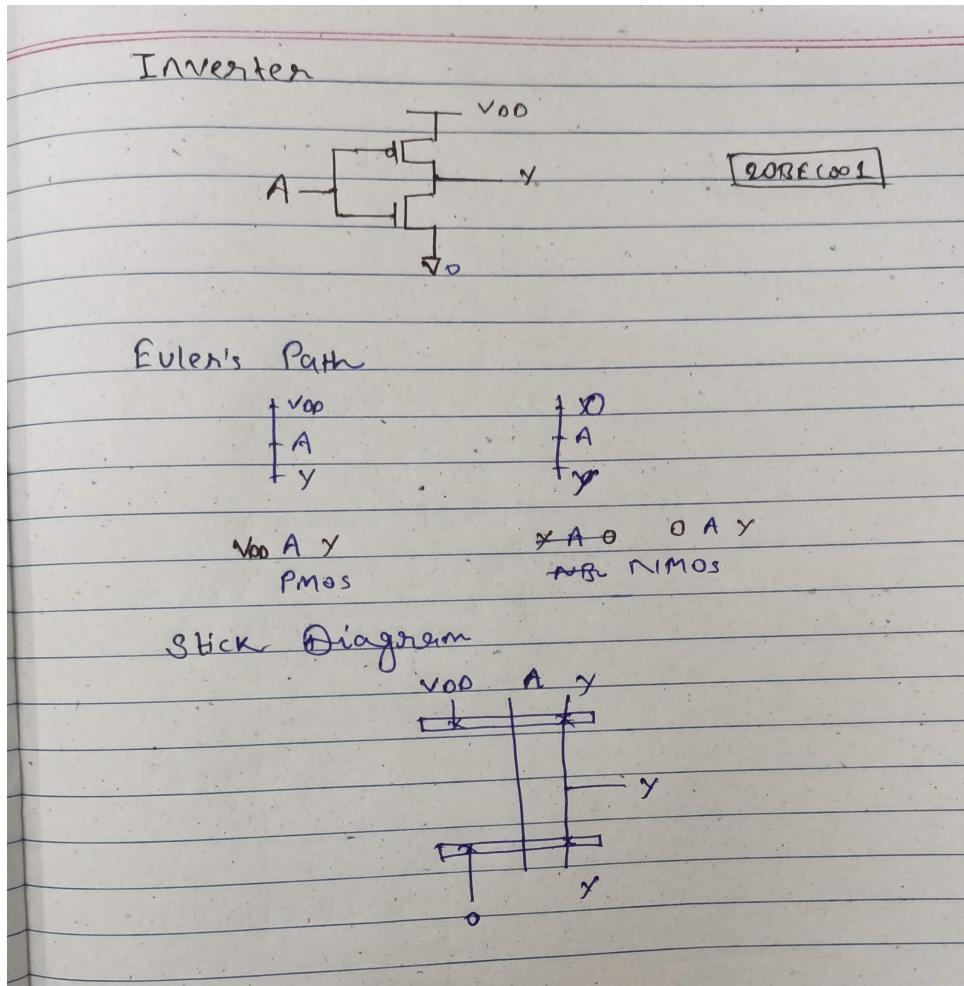
<b>1 INVERTER</b>	<b>1</b>
<b>2 NOR 2 Input</b>	<b>4</b>
<b>3 NOR 3 Input</b>	<b>7</b>
<b>4 NAND 2 Input</b>	<b>10</b>
<b>5 NAND 3 Input</b>	<b>13</b>
<b>6 Pass And</b>	<b>16</b>
<b>7 Pass OR</b>	<b>19</b>
<b>8 Transmission Gate</b>	<b>22</b>
<b>9 Footed Dynamic Cmos</b>	<b>25</b>
<b>10 D-Flip Flop</b>	<b>28</b>

## 1 INVERTER

In digital logic, an inverter or NOT gate is a logic gate which implements logical negation. In mathematical logic it is equivalent to the logical negation operator or Bar as complement.

It inverts the input, if input is logic 1 then the output will be logic 0 and vice versa.

### Schematic Diagram



### Ngspice Code

```
.title Inverter(DC) 20bec001
.include techfile130.txt

vdd vdd 0 1.2

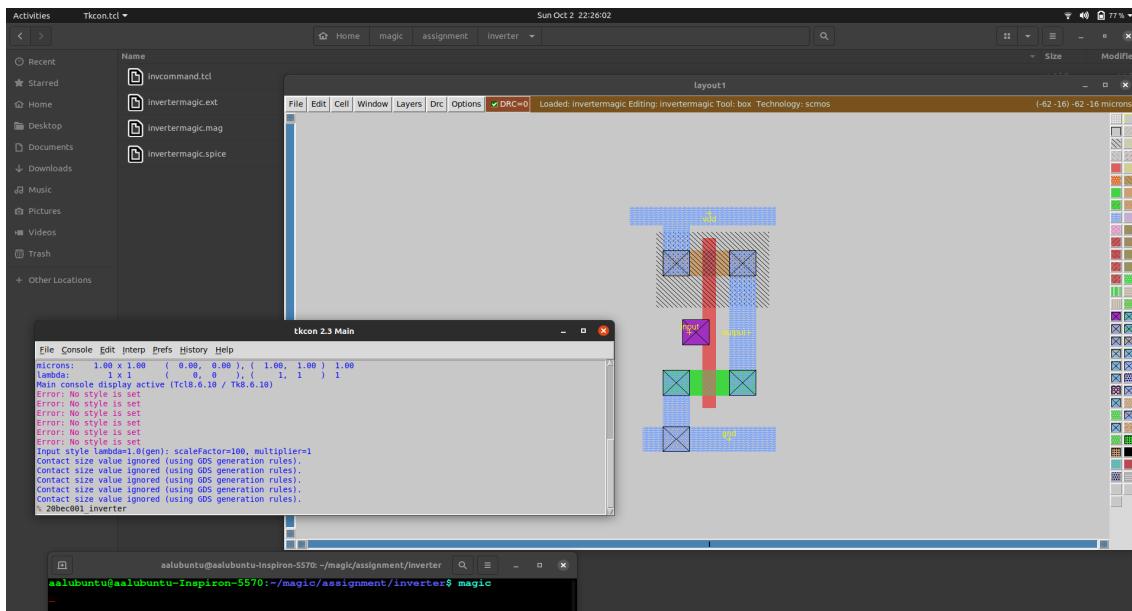
vin vi 0 PULSE(0 1.2 2NS 2NS 2NS 100NS 200NS)

Mp vout vi vdd vdd pmos l=130n w=3000n
Mn vout vi 0 0 nmos l=130n w=1000n
```

```
.tran 0.1n 500n 0 0.1n  
//.dc vin 0 1.2 0.1
```

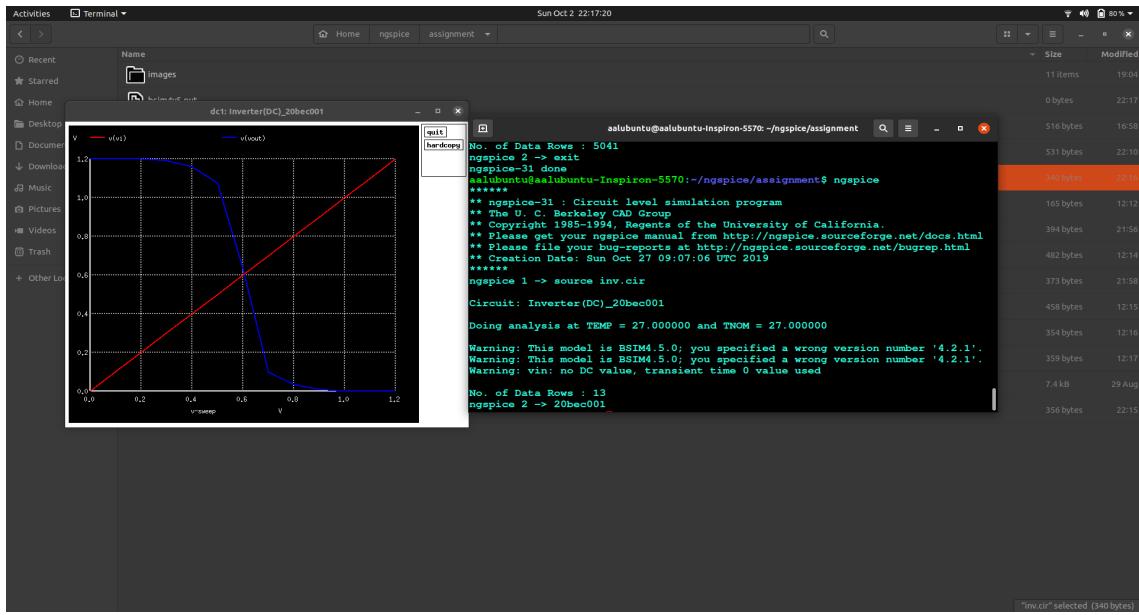
```
.control  
run  
plot v(vi) v(vout)  
.endc  
.end
```

## Magic Screenshot

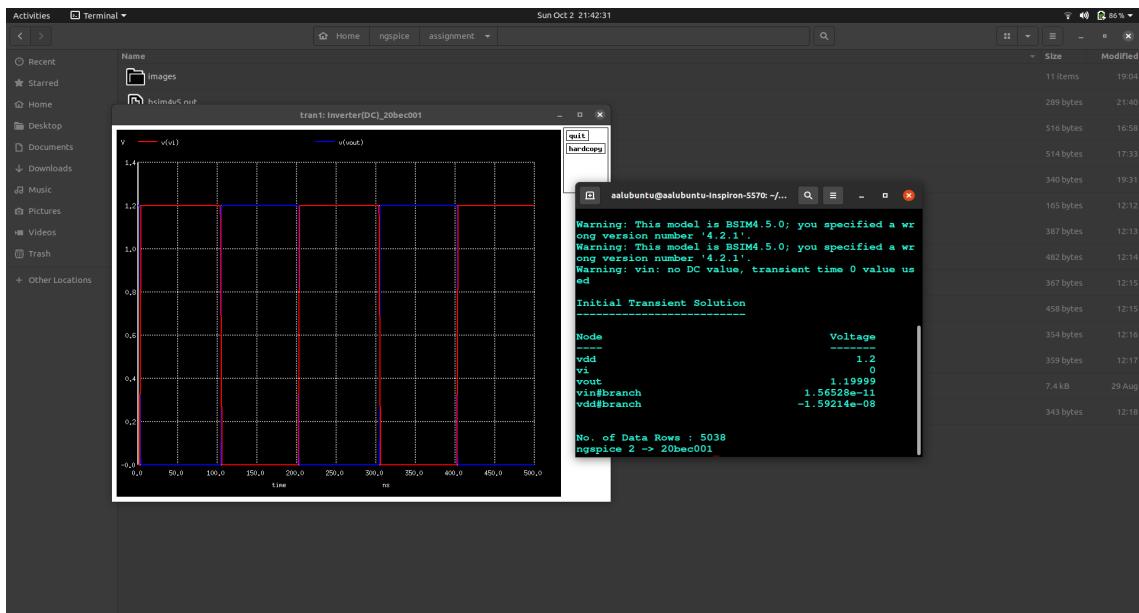


## Ngspice Simulation

## DC Analysis



## AC Analysis



## Conclusion

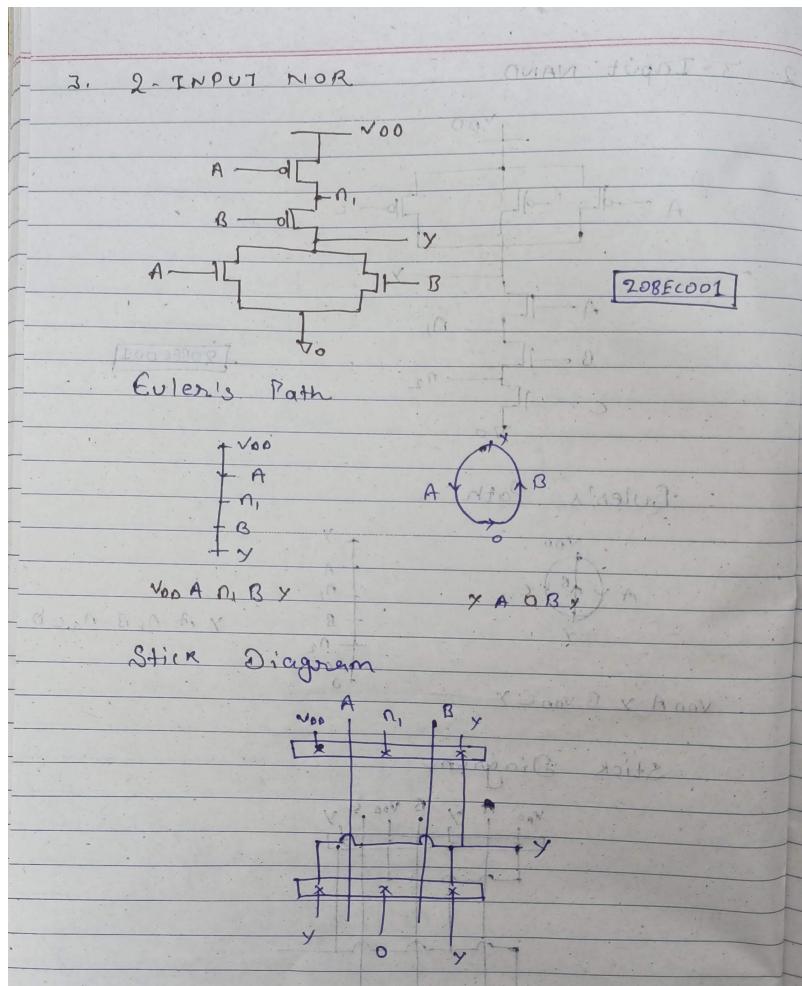
Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.

## 2 NOR 2 Input

It is a digital circuit that has two inputs and produces an output, which is the inversion of logical OR of all those inputs.

Logic NOR Gates are available using digital circuits to produce the desired logical function and is given a symbol whose shape is that of a standard OR gate with a circle, sometimes called an “inversion bubble” at its output to represent the NOT gate symbol with the logical operation of the NOR gate.

### Schematic Diagram



### Ngspice Code

```
.title NOR2inp 20bec001
.include techfile130.txt

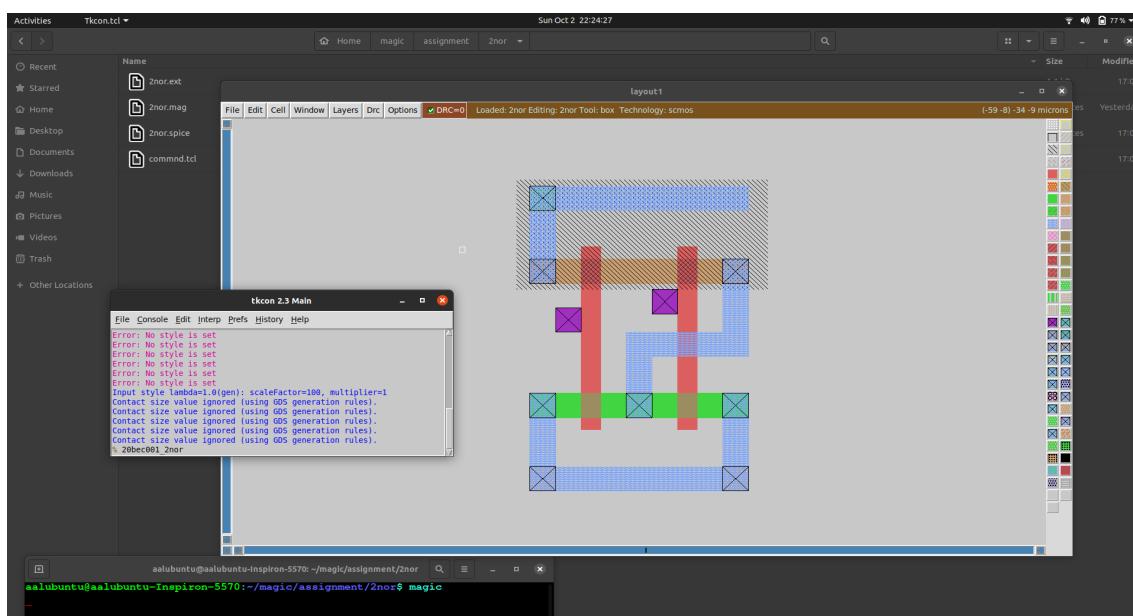
MP1 n1 A Vdd Vdd pmos w=200n l=130n
MP2 Y B n1 Vdd pmos w=260n l=130n
Mn1 Y A 0 0 nmos w=130n l=130n
Mn2 Y B 0 0 nmos w=130n l=130n
```

Vdd Vdd 0 1.2

VA A 0 PULSE (0 1.2 0 1n 1n 100n 200n)  
VB B 0 PULSE (0 1.2 0 1n 1n 200n 400n)

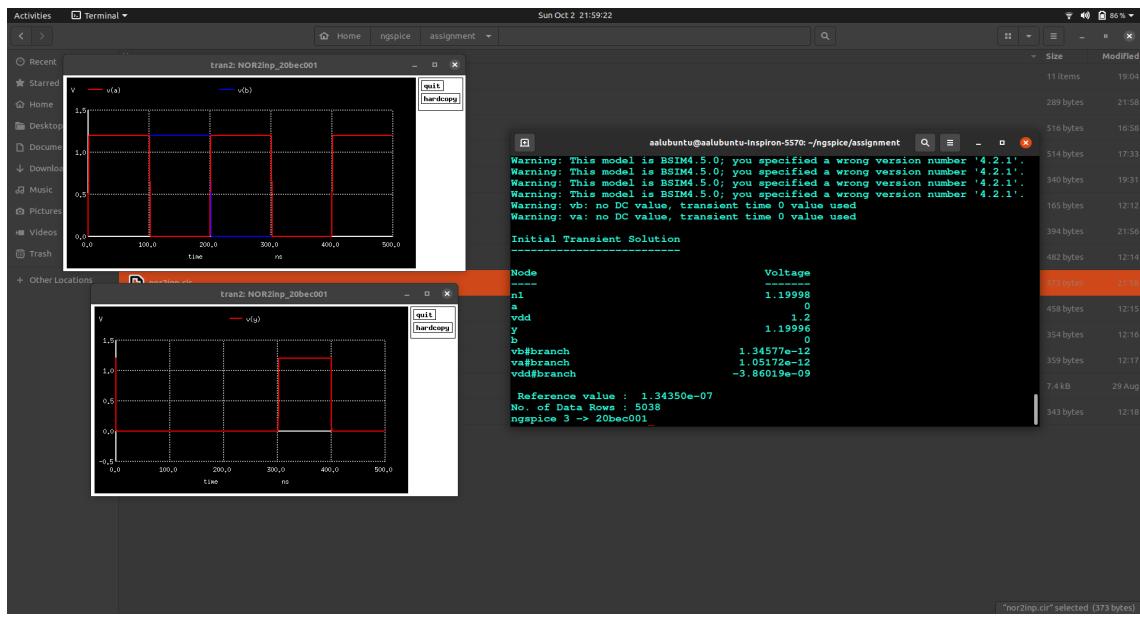
```
.tran 0.1n 500n 0 0.1n
.control
run
plot V(Y)
plot V(A) V(B)
.endc
```

### Magic Screenshot



### Ngspice Simulation

### AC Analysis



## Conclusion

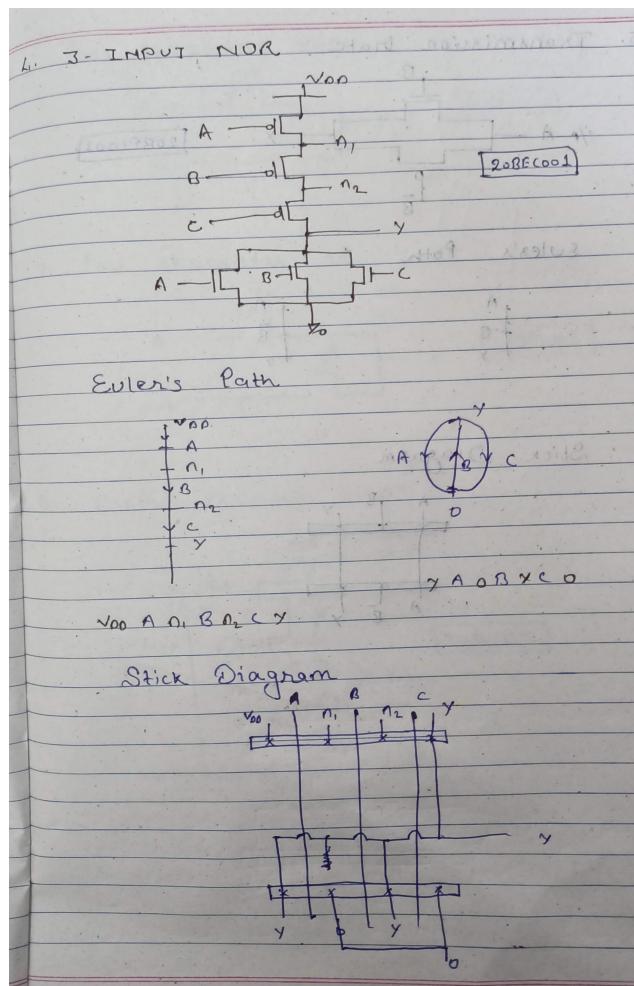
Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.

### 3 NOR 3 Input

It is a digital circuit that has three inputs and produces an output, which is the inversion of logical OR of all those inputs.

Logic NOR Gates are available using digital circuits to produce the desired logical function and is given a symbol whose shape is that of a standard OR gate with a circle, sometimes called an “inversion bubble” at its output to represent the NOT gate symbol with the logical operation of the NOR gate.

#### Schematic Diagram



#### Ngspice Code

```
.title 3inputNOR 20bec001  
.include techfile130.txt
```

Mp1 n1 A Vdd Vdd pmos w=260n l=130n

Mp2 n2 B n1 Vdd pmos w=260n l=130n

Mp3 Y C n2 Vdd pmos w=260n l=130n

```
Mn1 Y A 0 0 nmos w=130n l=130n  
Mn2 Y B 0 0 nmos w=130n l=130n  
Mn3 Y B 0 0 nmos w=130n l=130n
```

```
Vdd Vdd 0 1.2
```

```
VA A 0 PULSE (0 1.2 0 1n 1n 50n 100n)  
VB B 0 PULSE (0 1.2 0 1n 1n 100n 200n)  
VC C 0 PULSE (0 1.2 0 1n 1n 200n 400n)
```

```
.tran 0.1n 500n 0 0.1n
```

```
.control
```

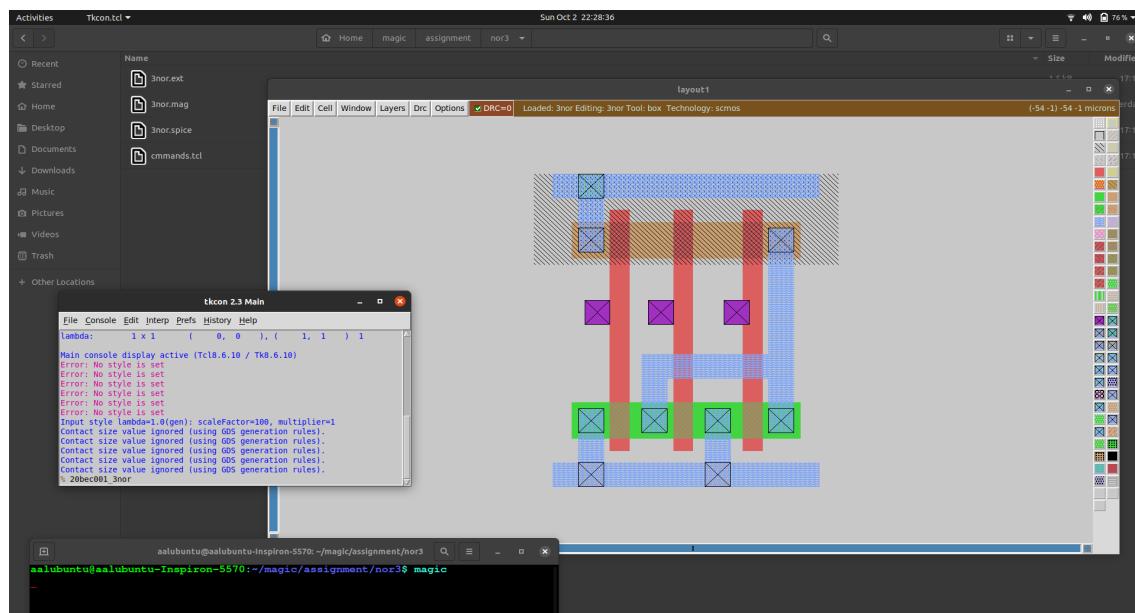
```
run
```

```
plot V(A) V(B) V(C)
```

```
plot V(Y)
```

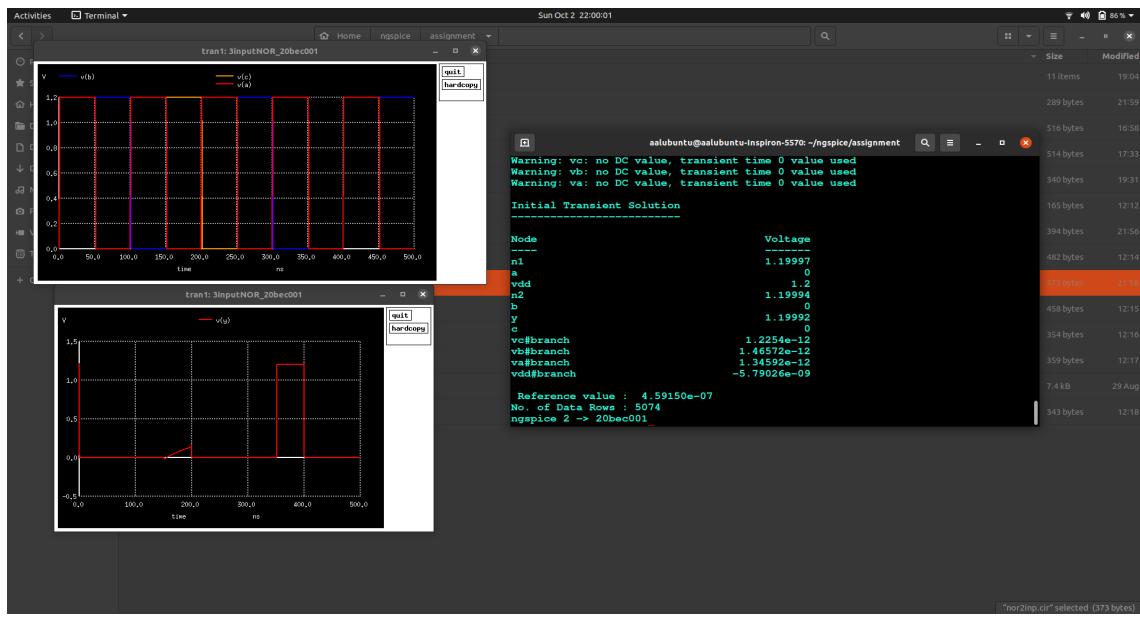
```
.endc
```

## Magic Screenshot



## Ngspice Simulation

## AC Analysis



## Conclusion

Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.

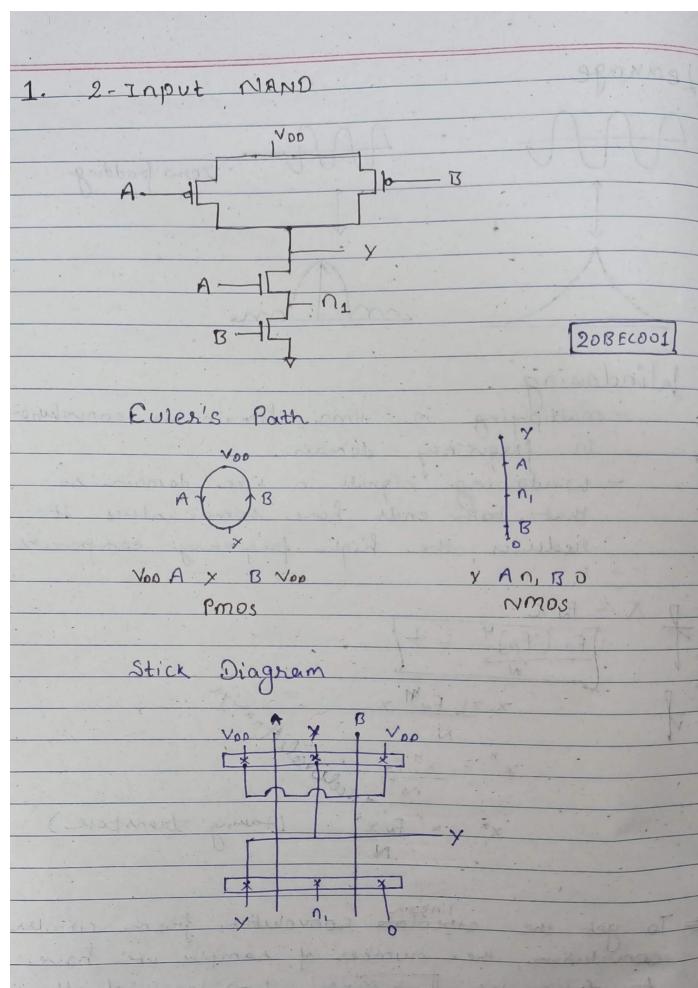
## 4 NAND 2 Input

The NAND gate or “NotAND” gate is the combination of two basic logic gates, the AND gate and the NOT gate connected in series. The NAND gate and NOR gate can be called the universal gates since the combination of these gates can be used to accomplish any of the basic operations.

The output of a NAND gate is high when either of the inputs is high or if both the inputs are low. In other words, the output is always high and goes low only when both the inputs are high. The logic NAND function is given by the Boolean expression

$$Y = A * B$$

### Schematic Diagram



### Ngspice Code

```
.title 2inputNAND 20bec001  
.include techfile130.txt
```

```
**MOS d g s b
```

MP1 Y A Vdd Vdd pmos w=260n l=130n  
MP2 Y B Vdd vdd pmos w=260n l=130n  
Mn1 Y A n1 0 nmos w=130n l=130n  
Mn2 n1 B 0 0 nmos w=130n l=130n

Vdd Vdd 0 1.2

VA A 0 PULSE (0 1.2 0 1n 1n 100n 200n)  
VB B 0 PULSE (0 1.2 0 1n 1n 198n 400n)

.tran 0.1n 500n 0 0.1n

.control

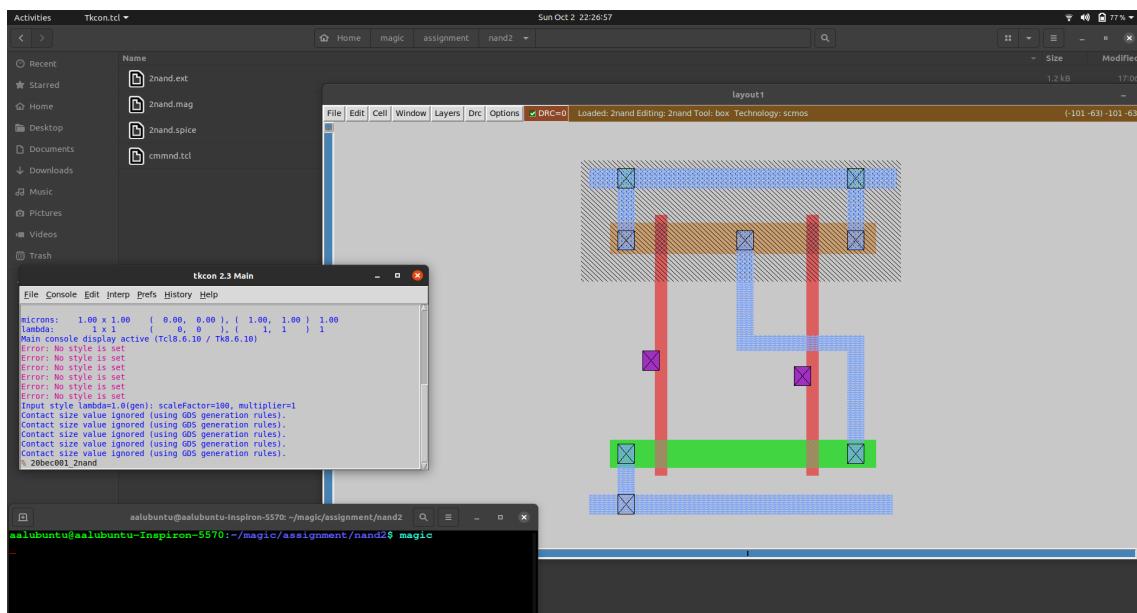
run

plot V(A) V(B)

plot V(Y)

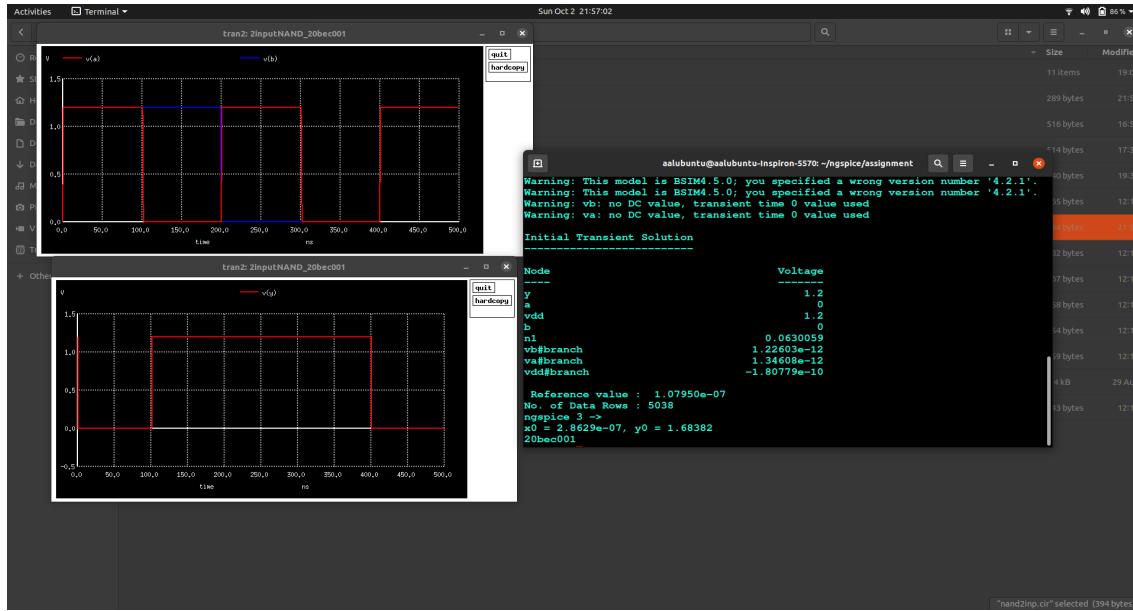
.endc

## Magic Screenshot



## Ngspice Simulation

### AC Analysis



### Conclusion

Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.

## 5 NAND 3 Input

The NAND gate or “NotAND” gate is the combination of two basic logic gates, the AND gate and the NOT gate connected in series. The NAND gate and NOR gate can be called the universal gates since the combination of these gates can be used to accomplish any of the basic operations.

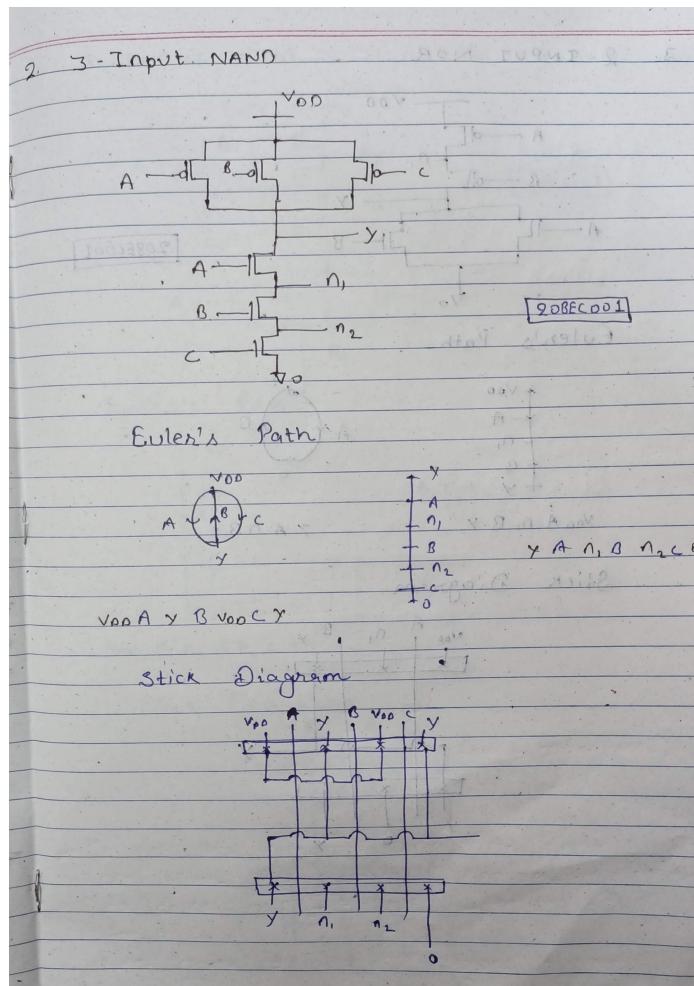
It is same as two input NAND gate but only difference is that it has three input.

The output of a NAND gate is high when either of the inputs is high or if both the inputs are low. In other words, the output is always high and goes low only when both the inputs are high.

The logic NAND function is given by the Boolean expression

$$Y = A * B * C$$

### Schematic Diagram



### Ngspice Code

```
.title 3inputNAND 20bec001  
.include techfile130.txt
```

\*\*MOs d g s b

Mp1 Y A Vdd Vdd pmos w=260n l=130n  
MP2 Y B Vdd vdd pmos w=260n l=130n  
MP3 Y C Vdd vdd pmos w=260n l=130n

Mn1 Y A n1 0 nmos w=130n l=130n  
Mn2 n1 B n2 0 nmos w=130n l=130n  
Mn3 n2 C 0 0 nmos w=130n l=130n

Vdd Vdd 0 1.2

VA A 0 PULSE (0 1.2 0 1n 1n 50n 100n)  
VB B 0 PULSE (0 1.2 0 1n 1n 98n 200n)  
VC C 0 PULSE (0 1.2 0 1n 1n 198n 400n)

.tran 0.1n 500n 0 0.1n

.control

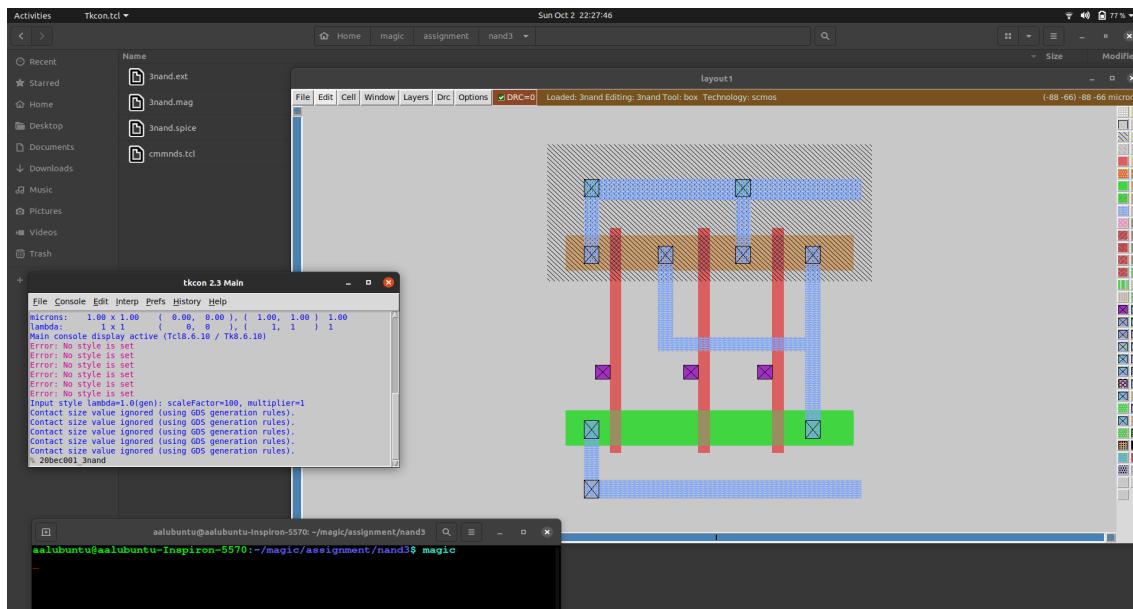
run

plot V(A) V(B) V(C)

plot V(Y)

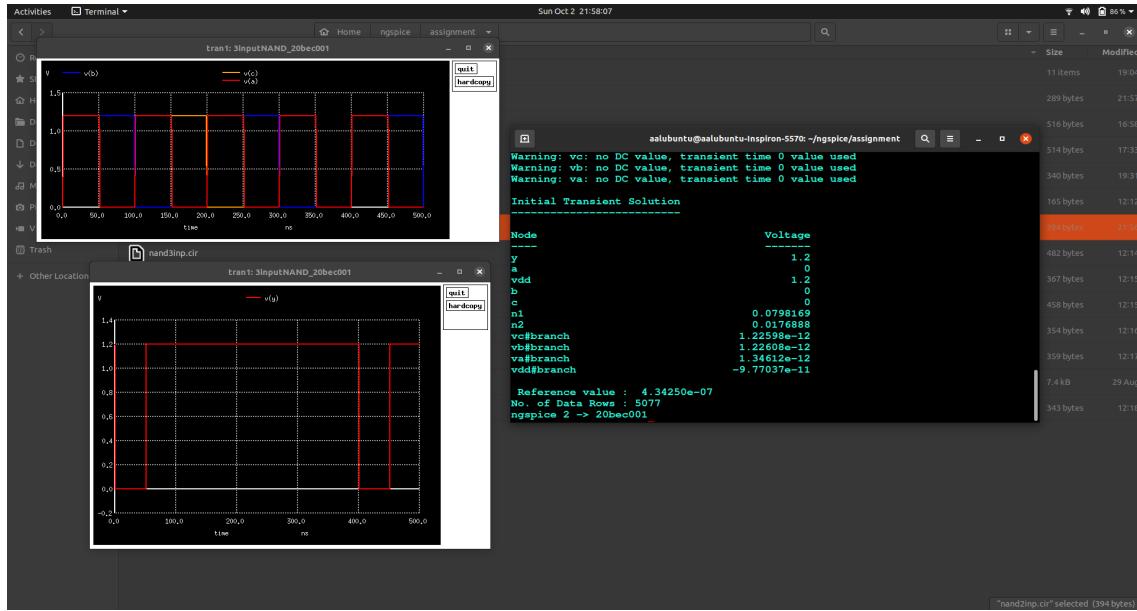
.endc

## Magic Screenshot



## Ngspice Simulation

### AC Analysis



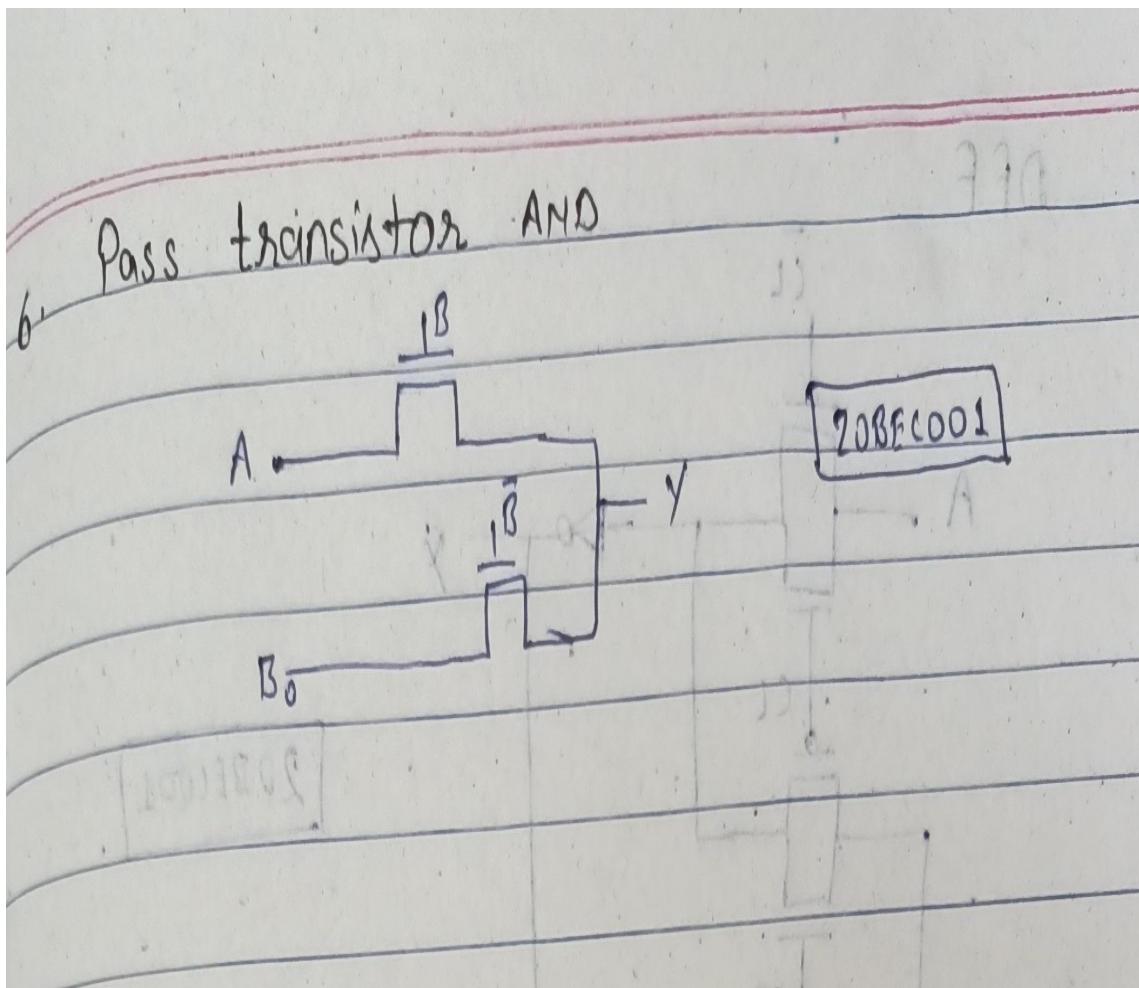
### Conclusion

Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.

## 6 Pass And

the complementary CMOS gate as switching the output pin to one of power or ground. A slightly more general gate is obtained if we switch the output to one of power; ground; or any of the input signals. In such designs the MOSFET is considered to be a pass transistor. Here pass transistor acts as simple AND gate.

### Schematic Diagram



### Ngspice Code

```
.title pass transistor 20bec001
.include techfile130.txt
.include Inv.lib
```

```
*****MOS d g s b
```

```
Mn1 Y B A 0 nmos w=130n l=130n
```

Mn2 Y 1 B 0 nmos w=130n l=130n

x1 B 1 Vdd inv

Vdd Vdd 0 1.2

VA A 0 PULSE (0 1.2 0 2n 2n 100n 200n)  
VB B 0 PULSE (0 1.2 0 2n 2n 198n 400n)

.tran 0.1n 500n 0 0.1n

.control

run

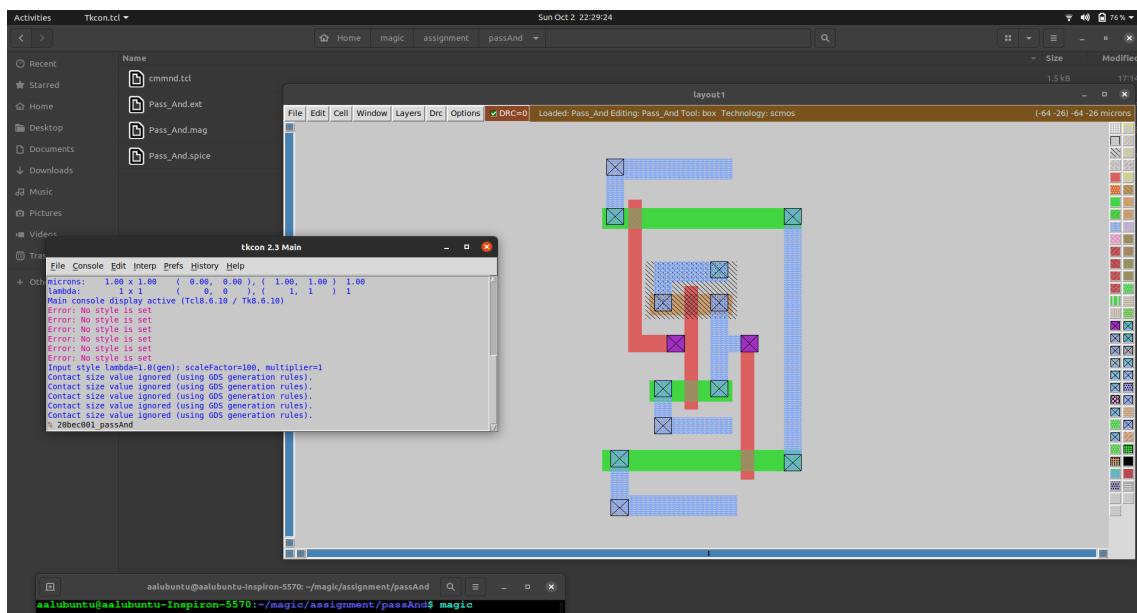
plot V(A)

plot V(B)

plot V(Y)

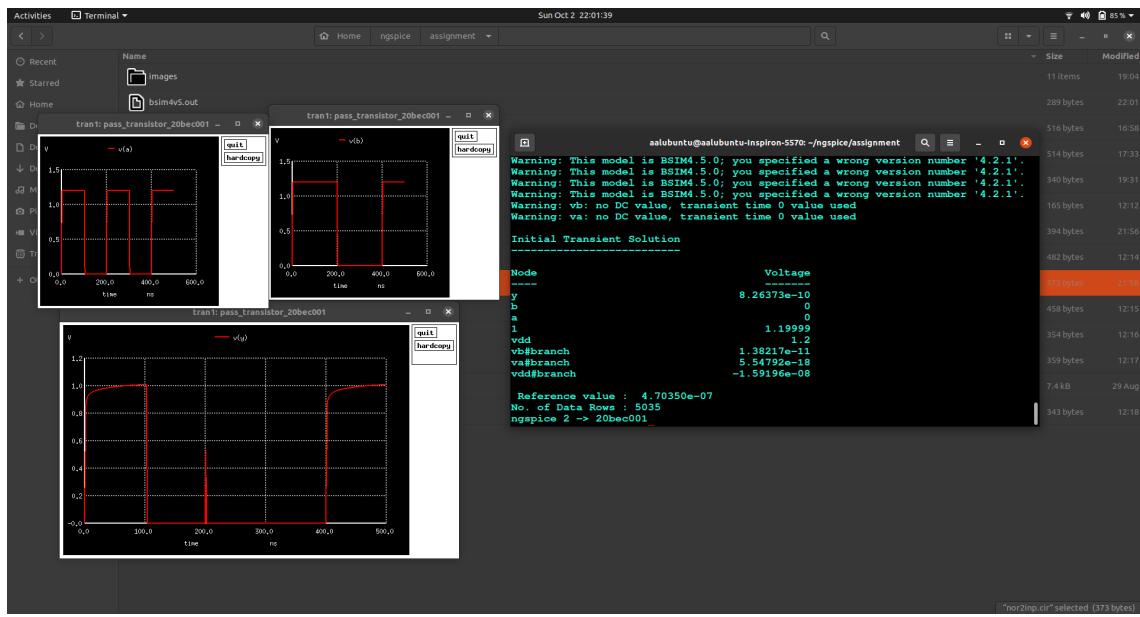
.endc

## Magic Screenshot



## Ngspice Simulation

## AC Analysis



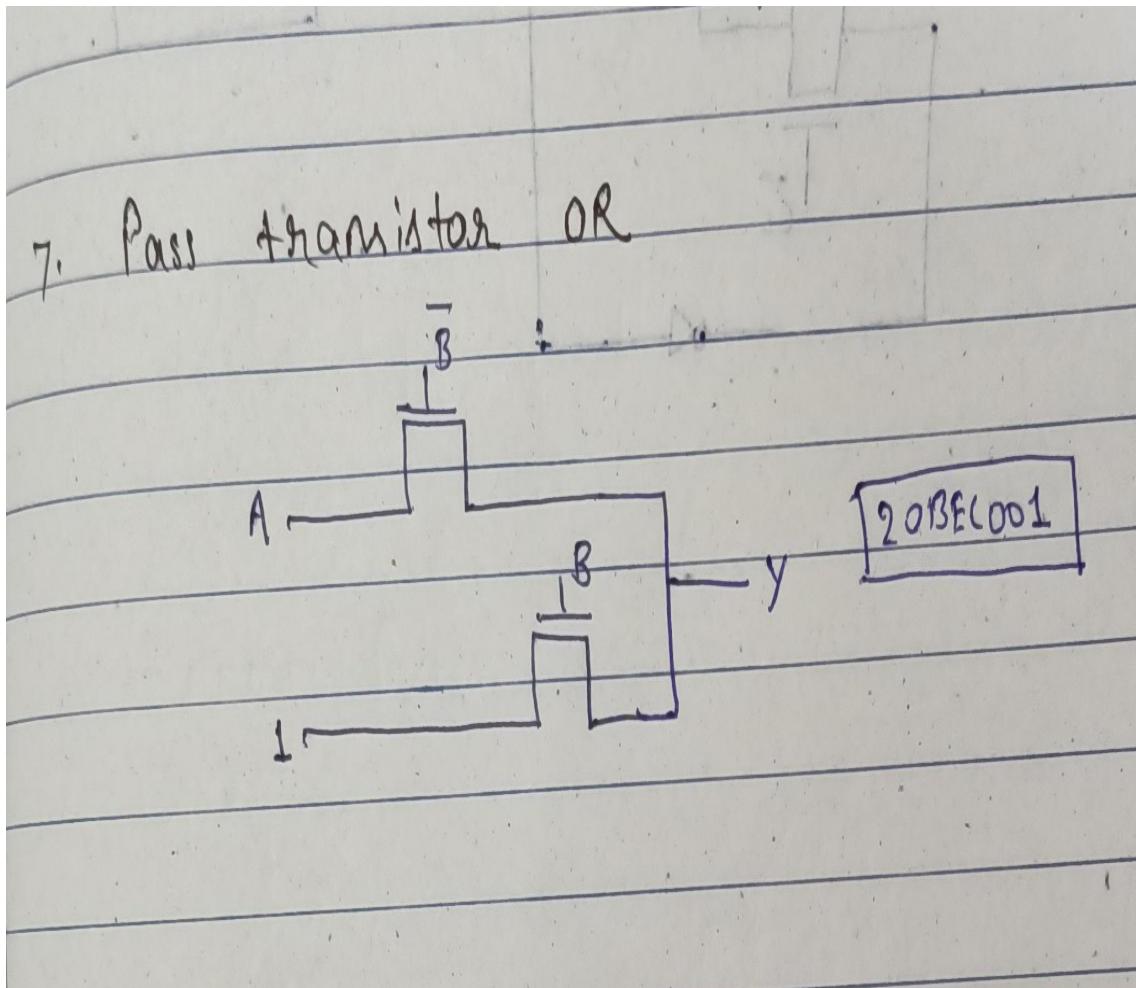
## Conclusion

Hence we got the Rise and Fall time for the output are nearly double as per the ngspice simulation.

## 7 Pass OR

the complementary CMOS gate as switching the output pin to one of power or ground. A slightly more general gate is obtained if we switch the output to one of power; ground; or any of the input signals. In such designs the MOSFET is considered to be a pass transistor. Here pass transistor acts as simple OR gate.

### Schematic Diagram



### Ngspice Code

```
.title pass transistor OR 20bec001
.include techfile130.txt
.include Inv.lib
```

```
**MOS d g s b
```

```
Mn1 Y ino A 0 nmos w=130n l=130n
```

Mn2 Y B vdd 0 nmos w=130n l=130n

x1 B ino Vdd inv

Vdd Vdd 0 1.2

VA A 0 PULSE (0 1.2 0 1n 1n 100n 200n)  
VB B 0 PULSE (0 1.2 0 1n 1n 200n 400n)

.tran 0.1n 500n 0 0.1n

.control

run

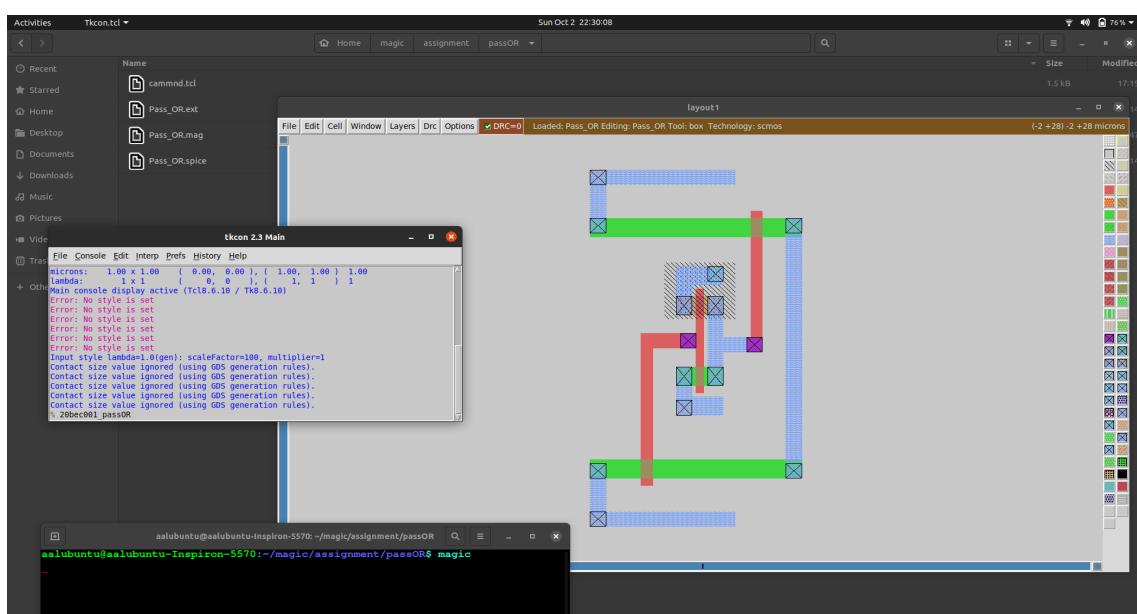
plot V(A)

plot V(B)

plot V(Y)

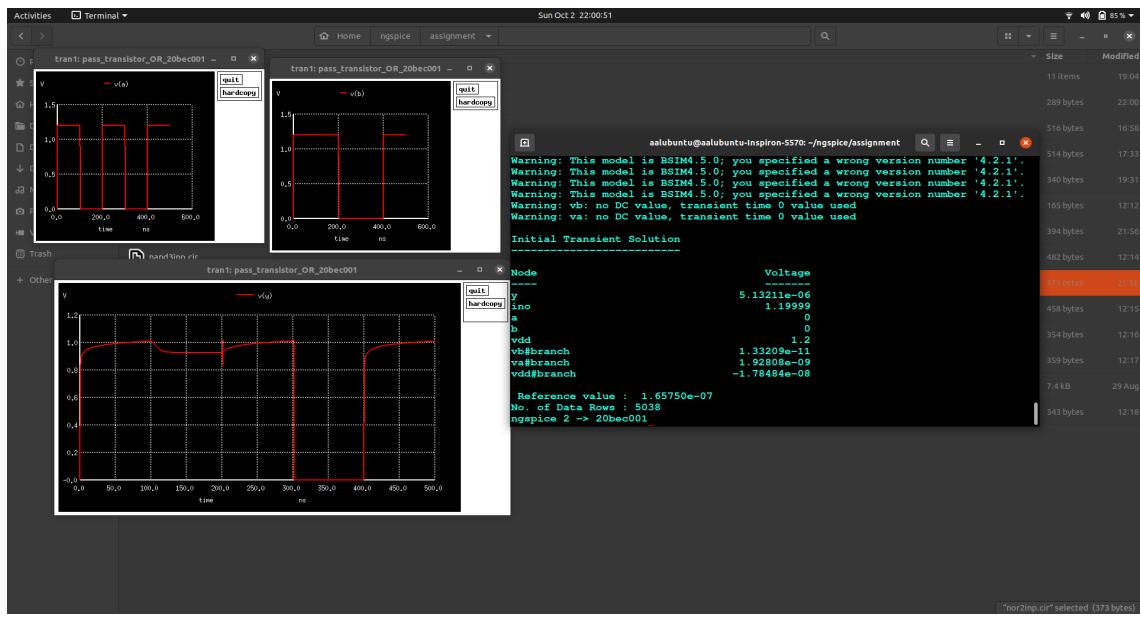
.endc

## Magic Screenshot



## Ngspice Simulation

## AC Analysis



## Conclusion

Hence we got the Rise and Fall time for the output are nearly double as per the ngspice simulation.

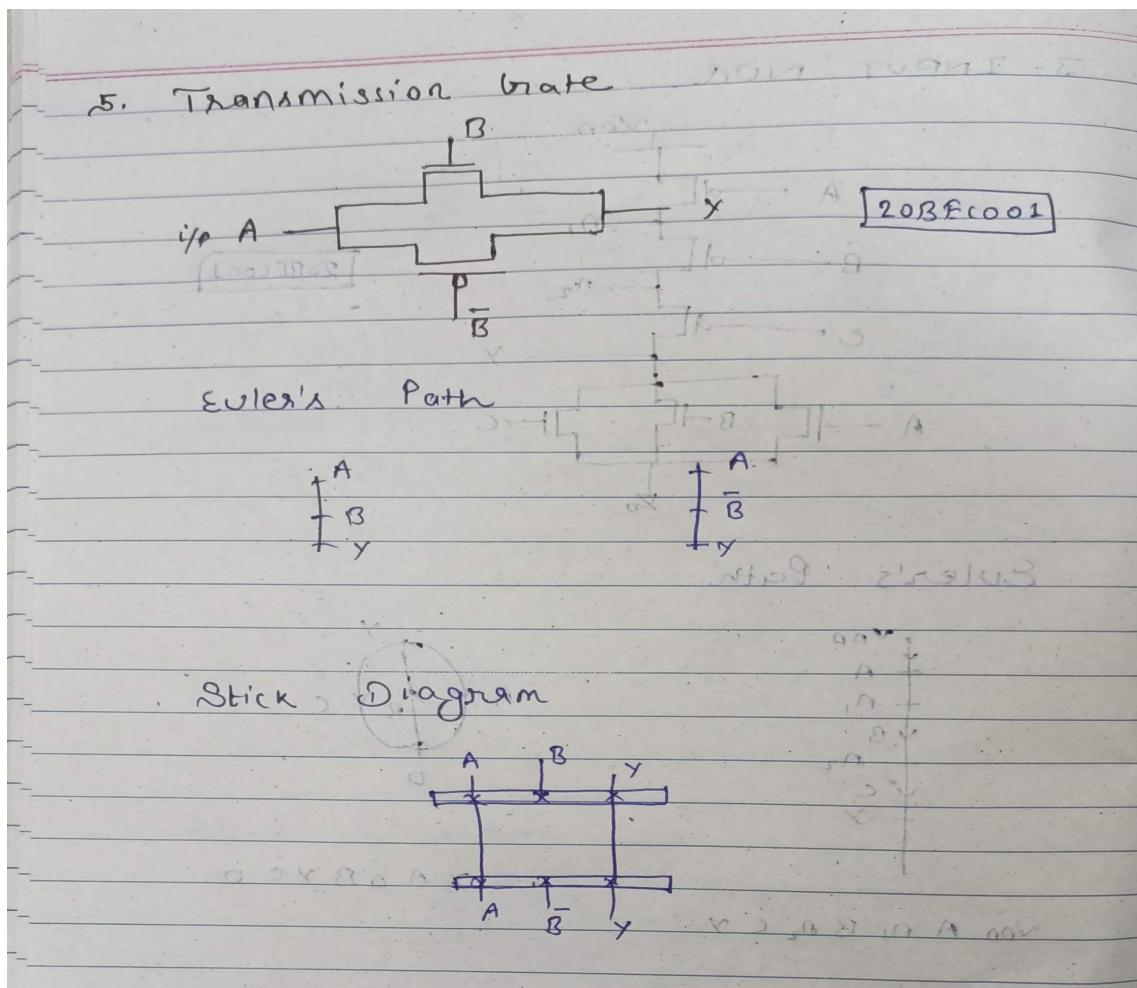
## 8 Transmission Gate

The transmission gate logic is used to solve the voltage drop problem of the pass transistor logic.

This technique uses the complementary properties of NMOS and PMOS transistors. i.e. NMOS devices passes a strong '0' but a weak '1' while PMOS transistors pass a strong '1' but a weak '0'.

The transmission gate is mainly a bi-directional switch enabled by the gate signal 'C'. When C = 1 both MOSFETs are ON and the signal pass through the gate i.e. A = B if C = 1. Whereas C = 0 makes the MOSFETs cut off creating an open circuit between nodes A and B.

### Schematic Diagram



### Ngspice Code

```
.title transmission gate 20bec001
.include techfile130.txt
.include Inv.lib
```

\*\*MOs d g s b

Mn1 Y B A 0 nmos w=130n l=130n

Mp2 Y 1 A Vdd pmos w=260n l=130n

x1 B 1 Vdd inv

Vdd Vdd 0 1.2

VA A 0 PULSE (0 1.2 0 2n 2n 100n 200n)  
VB B 0 PULSE (0 1.2 0 2n 2n 200n 300n)

Cload Y 0 70f

```
.tran 0.1n 500n 0 0.1n
```

.control

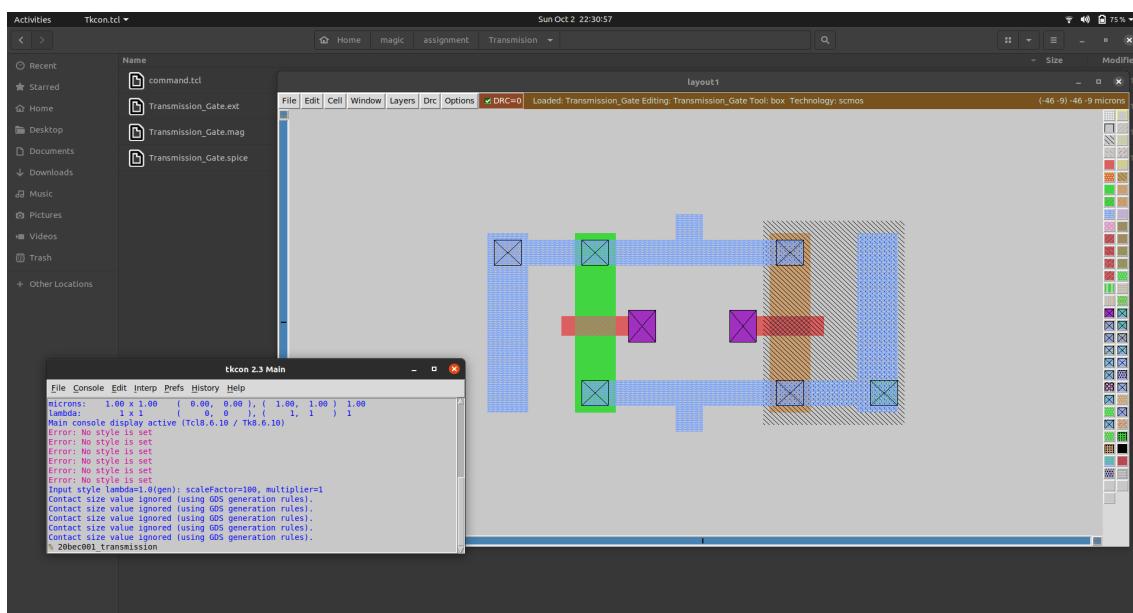
run

plot V(A) V(B)

plot V(Y)

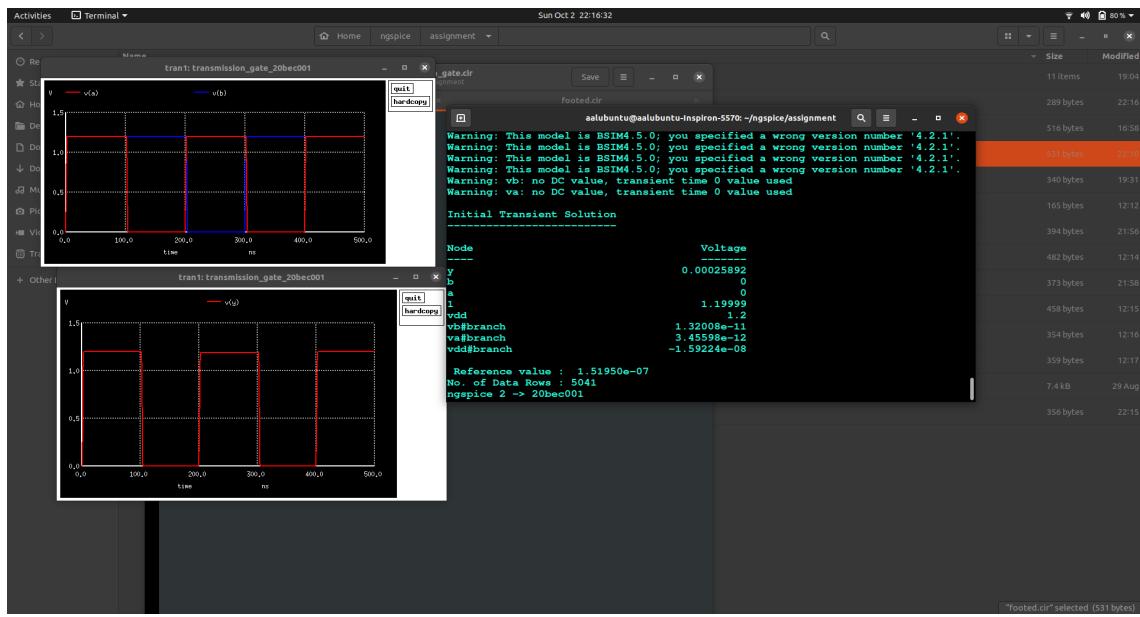
.endc

## Magic Screenshot



## Ngspice Simulation

## AC Analysis



## Conclusion

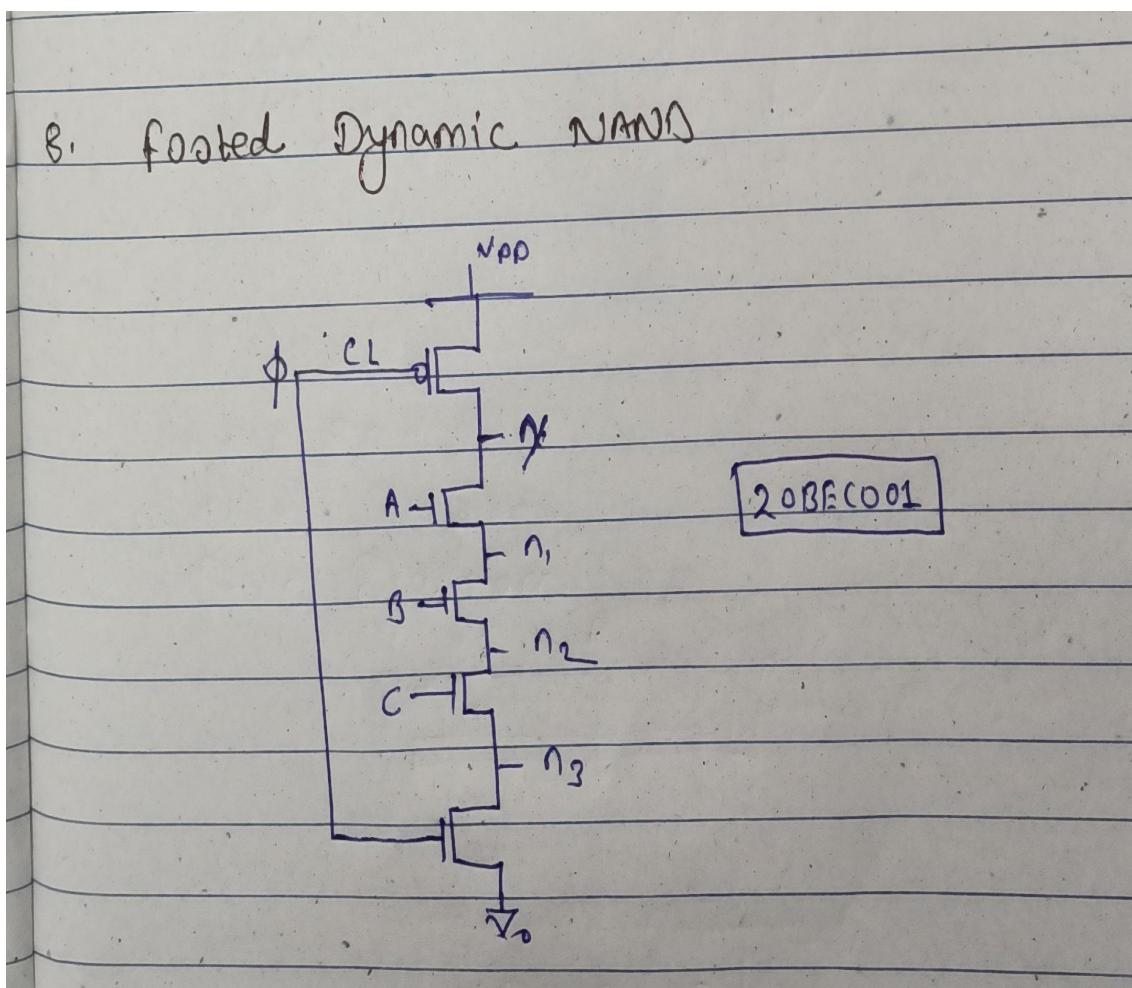
Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.

## 9 Footed Dynamic Cmos

A dynamic logic gate uses clocking and charge storage properties of MOSFETs to implement logic operation. The clock provides a synchronized data flow which makes the technique useful in designing sequential networks. The characterising feature of a dynamic logic gate is that the result of a calculation is valid only for a short period of time.

if the input is '1' during precharge, contention will take place because both the pMOS and nMOS transistors will be ON. When the input cannot be guaranteed to be '0' during precharge, an extra clocked evaluation transistor can be added to the bottom of the nMOS stack to avoid contention. The extra transistor is sometimes called a foot

## Schematic Diagram



## Ngspice Code

.title footed-dynam nand 20bec001  
.include techfile130.txt

\*\*\*Mos d g s b

Mp1 Y CL Vdd Vdd pmos w=4000n l=130n

Vdd Vdd 0 1.2

Mn1 Y A n1 0 nmos w=3000n l=200n

Mn2 n1 B n2 0 nmos w=3000n l=200n

Mn3 n2 C n3 0 nmos w=3000n l=200n

Mn4 n3 CL 0 0 nmos w=3000n l=200n

Cload Y 0 100f

VCL CL 0 PULSE (0 1.2 0 1n 1n 50n 100n)

VA A 0 PULSE (0 1.2 0 1n 1n 100n 150n)

VB B 0 PULSE (0 1.2 0 1n 1n 68n 200n)

VC C 0 PULSE (0 1.2 0 1n 1n 198n 400n)

.tran 0.1n 500n

.control

run

plot V(CL)

plot V(A)

plot V(B)

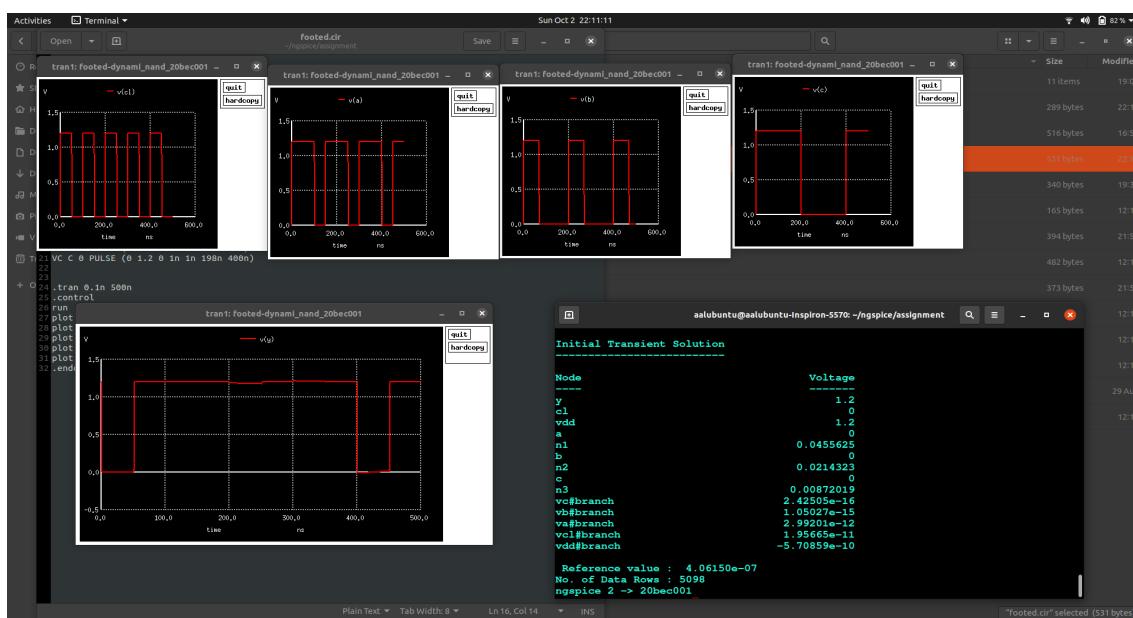
plot V(C)

plot V(Y)

.endc

## Ngspice Simulation

### AC Analysis



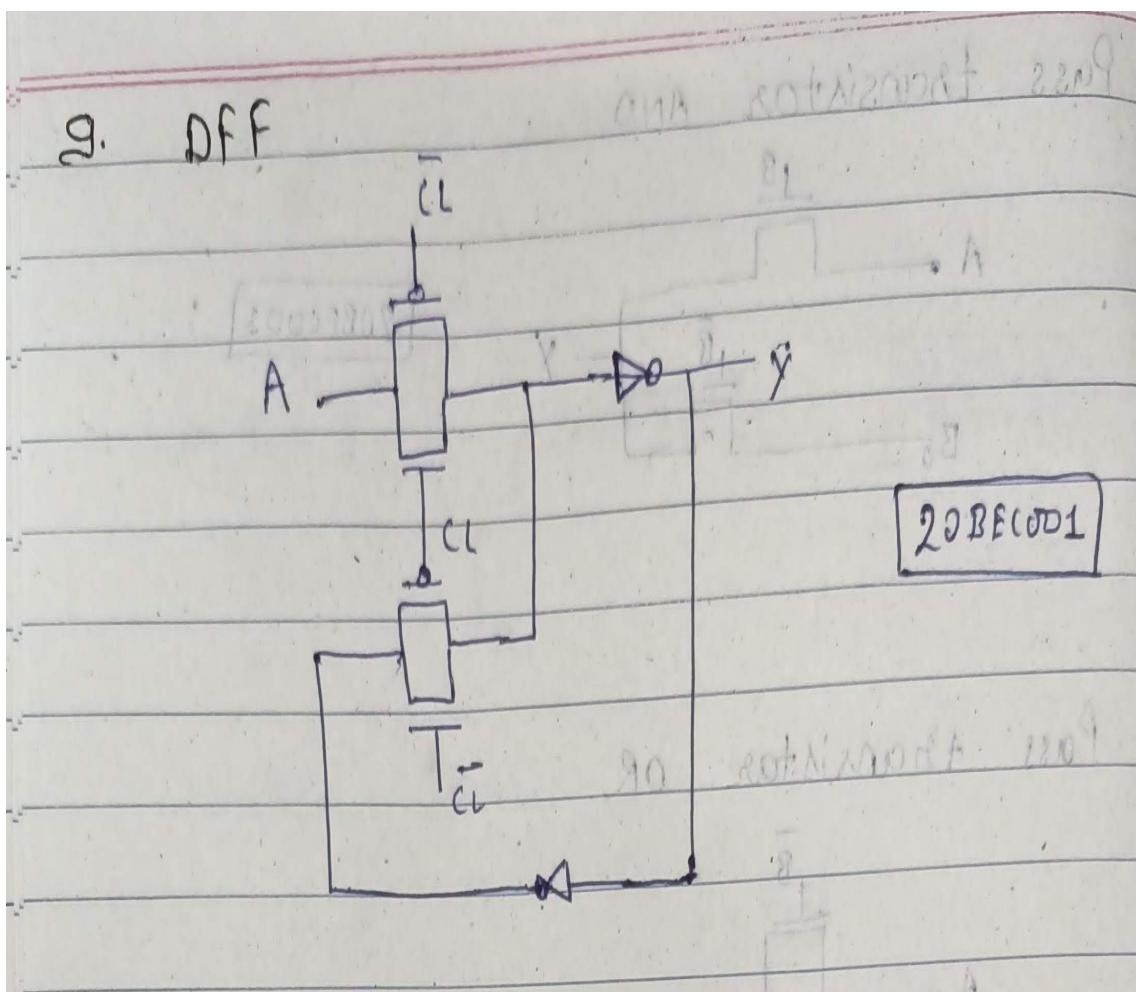
## **Conclusion**

Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.

## 10 D-Flip Flop

A D-type flip-flop operates with a delay in input by one clock cycle. Thus, by cascading many D-type flip-flops delay circuits can be created, which are used in many applications such as in digital television systems. A D-type flip-flop is also known as a D flip-flop or delay flip-flop. The D flip flop is the most important flip flop from other clocked types. It ensures that at the same time, both the inputs, i.e., S and R, are never equal to 1. The Delay flip-flop is designed using a gated SR flip-flop with an inverter connected between the inputs allowing for a single input D(Data).

### Schematic Diagram



### Ngspice Code

```
.title DFF 20bec001
.include techfile130.txt
.include Inv.lib

**MOs d g s b
Mp1 n1 1 A Vdd pmos w=260n l=130n
```

Mp2 n1 CL n2 Vdd pmos w=260n l=130n

Mn1 n1 CL A 0 nmos w=130n l=130n

Mn2 n1 1 n2 0 nmos w=130n l=130n

x1 CL 1 Vdd inv

x3 Y n2 Vdd inv

x4 n1 Y Vdd inv

Vdd Vdd 0 1.2

VA A 0 PULSE (0 1.2 0 1n 1n 100n 200n)

VCL CL 0 PULSE (0 1.2 0 1n 1n 50n 100n)

.tran 0.1n 1000n 0 0.1n

.control

run

plot V(A)

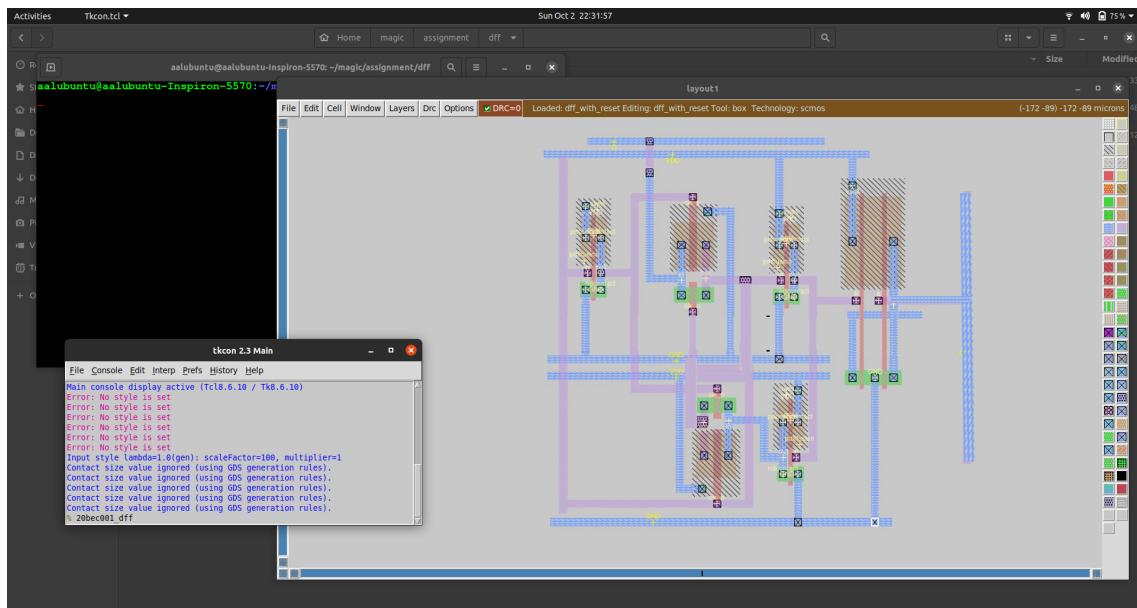
plot V(CL)

plot V(1)

plot V(Y)

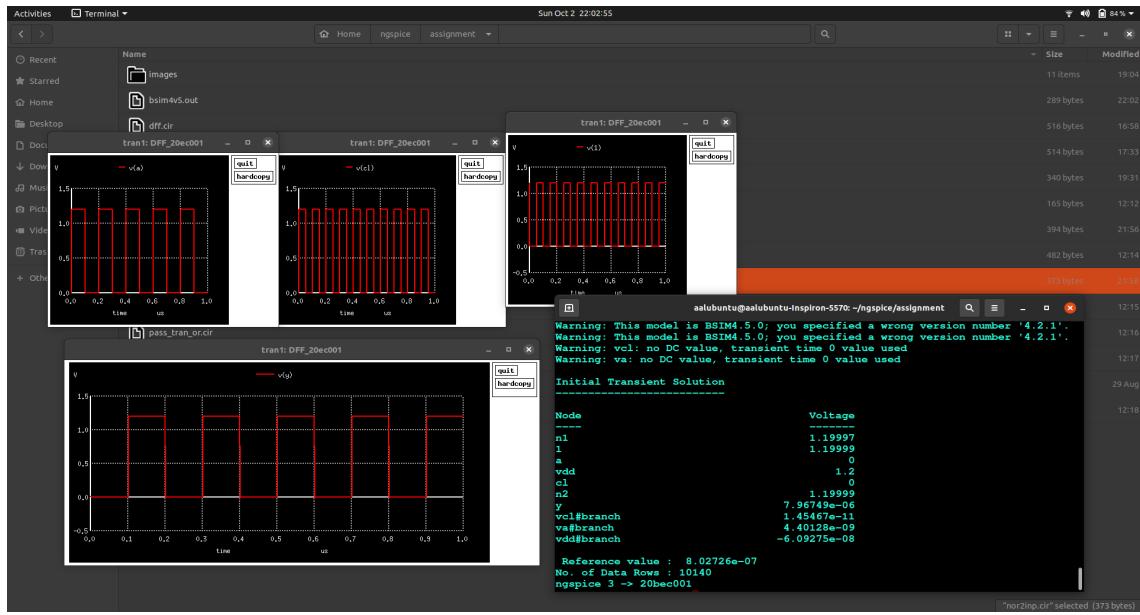
.endc

## Magic Screenshot



## Ngspice Simulation

### AC Analysis



## Conclusion

Hence we got the Rise and Fall time for the output are nearly same as per the ngspice simulation.