

Name: KVS RAKESH

Roll no.: BT18ECE063

Branch: ECE

Semester: 6th

Instructor: Dr. Paritosh Peshwe

CMOS DESIGN
PROJECT REPORT
MOD-4 Synchronous Up-counter

INDEX

S. No.	DESCRIPTION
1	Abstract
2	Introduction
3	Circuit Diagram
4	Simulation Layout
5	Simulation Program
6	Results
7	Conclusions

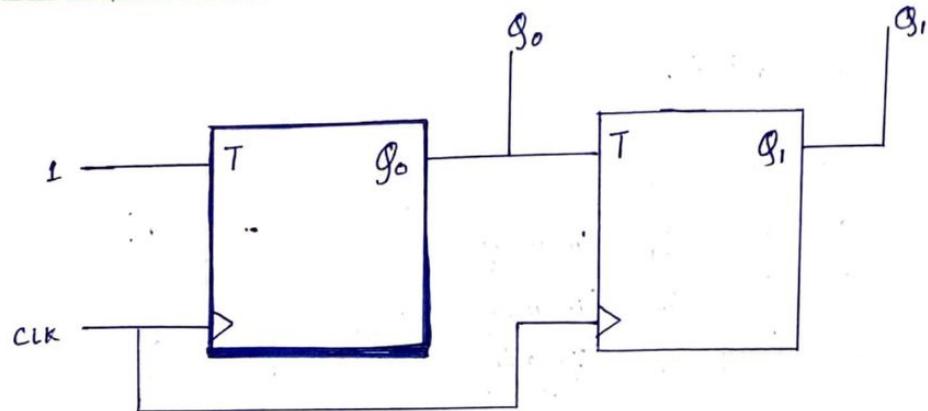
ABSTRACT:

- ✓ PROJECT – implementation of MOD 4 synchronous Up counter using T Flip Flops.
- ✓ Implementation – done by both layout and netlist
- ✓ Counter counts from decimal 0 to 3, that is 4 numbers over and over infinitely when provided with input clock and power to tell when to count up.
- ✓ Implemented MOD-8 also, but let's discuss what happened with that also.

INTRODUCTION:

- ✓ For designing an up counter, I have used 2 T Flip Flops in series.
- ✓ basically T Flip Flops inverts the last output whenever a rising edge of clock appears on the clock input.
- ✓ Every T flip flop slows the input clock by 2 times or it doubles its time period. If we see binary representation of numbers from decimal 0-3, we see that they alternate on alternate clock cycles. This means that if we attach 2 flip flops in series, we can achieve the desired output.
- ✓ This circuit is clock synchronous which means that both the T Flip Flops are given the same clock and thus operate on the same clock.

CIRCUIT DIAGRAM:



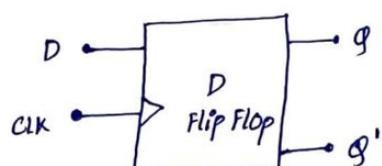
<u>CLK</u>	Q_1	Q_0	
↓	0	0	0
↑	0	1	1
↓	1	0	2
↑	1	1	3

Rising edge

FOR designing single T- flip flop(By using D_{FF} and XOR gate)

$T_{FF} \rightarrow D_{FF} \& \text{XOR}$
 ↓ →
 D-flipflop Gate

D _{FF} Truth table		
CLK	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1

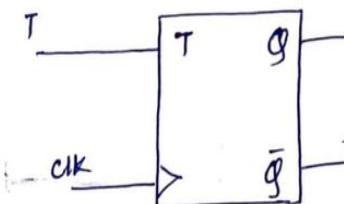


XOR Truth table		
X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

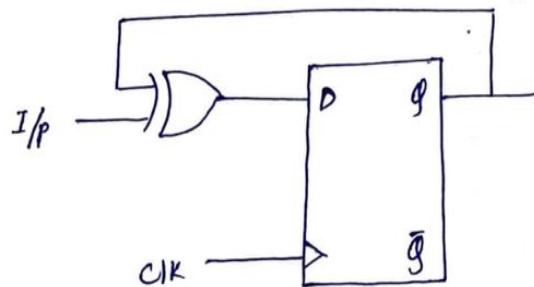


T_{FF} Truth table

clk	T	Q _{n+1}
0	x	Q _n (memory)
1	0	Q _n (memory)
1	1	Q̄ _n (toggling)



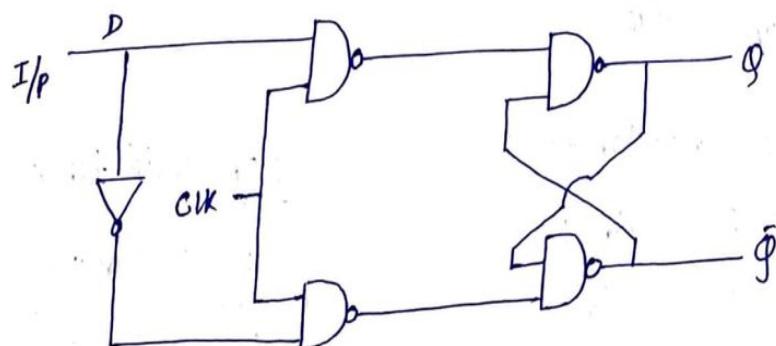
T_{FF} By using XOR Gate & D_{FF}



I/P	clk	Q _n	Q _{n+1}
1	F	0	1
1	F	1	0
0	F	0	0
0	F	1	1

FOR designing single D - flip flops(By using NAND gate representation)

D_{FF} → By using NAND gate

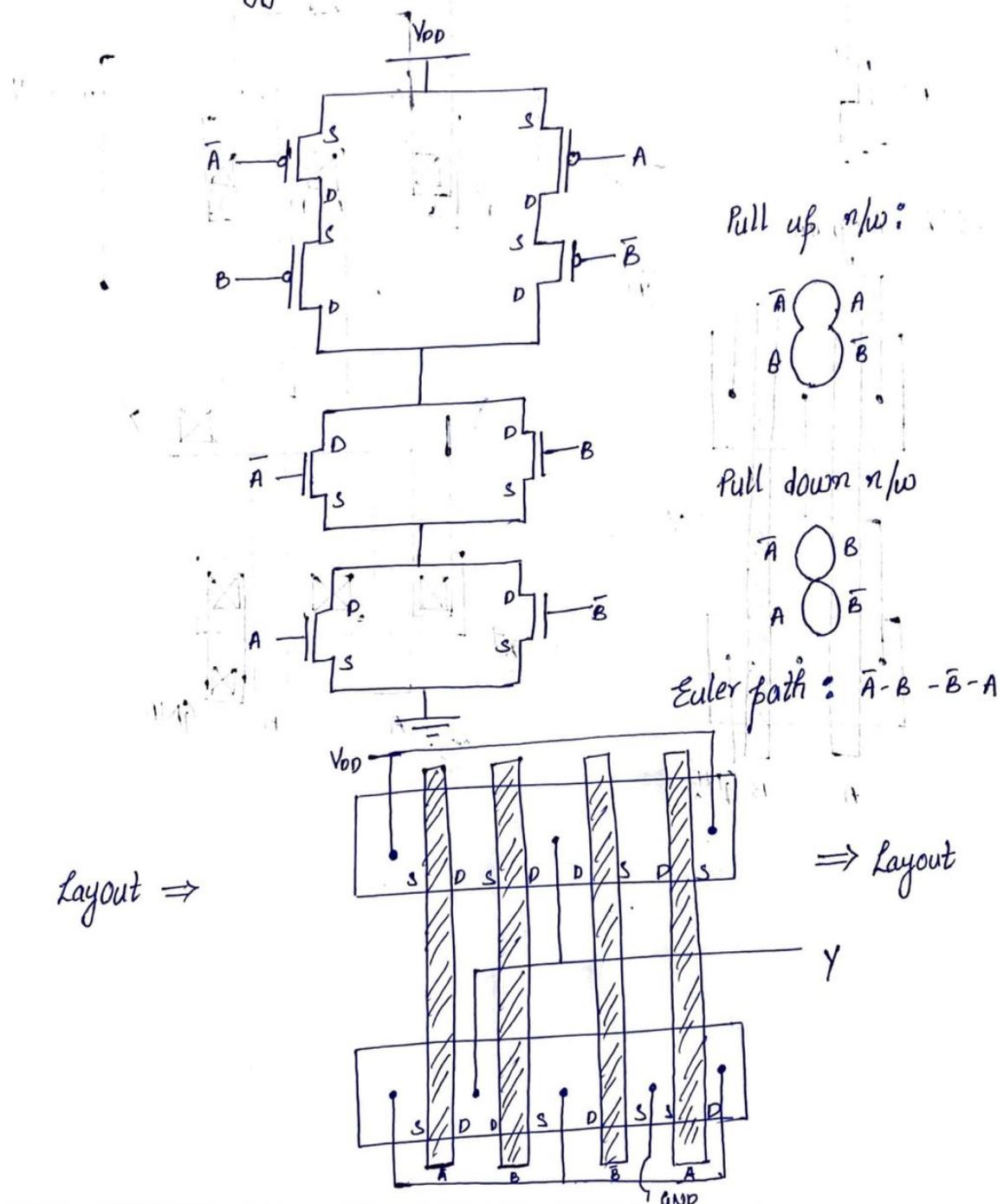


For Designing the XOR gate (by using CMOS technology logic)

XOR Gate

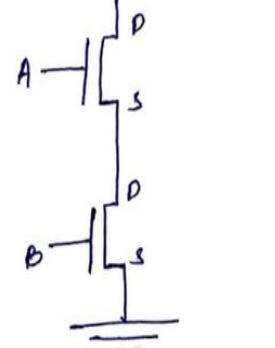
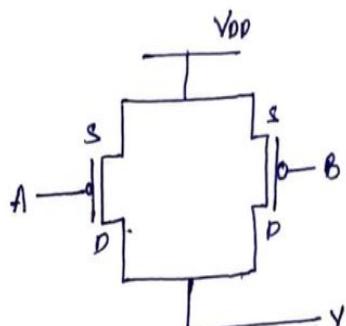
$$\begin{aligned} Y &= A\bar{B} + \bar{A}B \\ &= (\overline{\bar{A}+B}) + (\overline{A+\bar{B}}) \\ &= (\overline{\bar{A}+B})(\overline{A+\bar{B}}) \end{aligned}$$

CMOS technology schematic

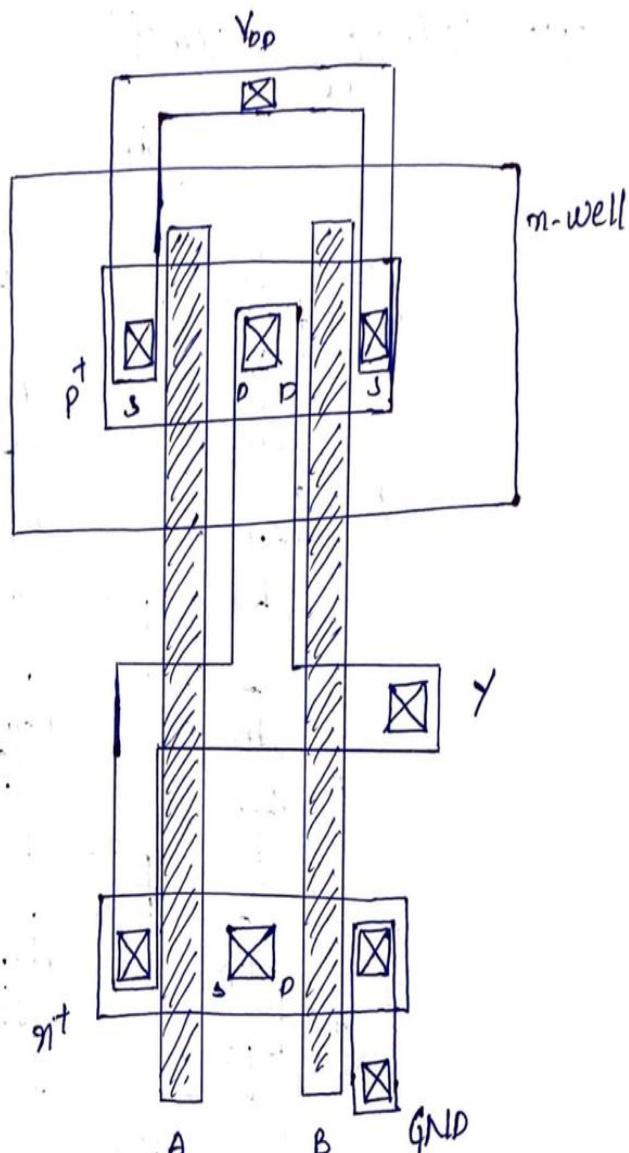
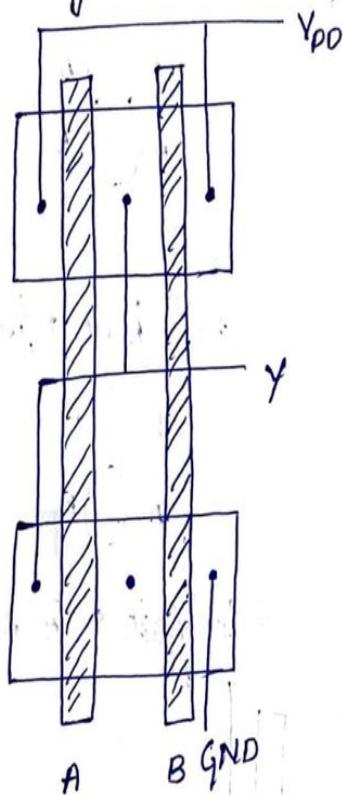


For Designing the NAND gate (by using CMOS technology logic)

NAND GATE

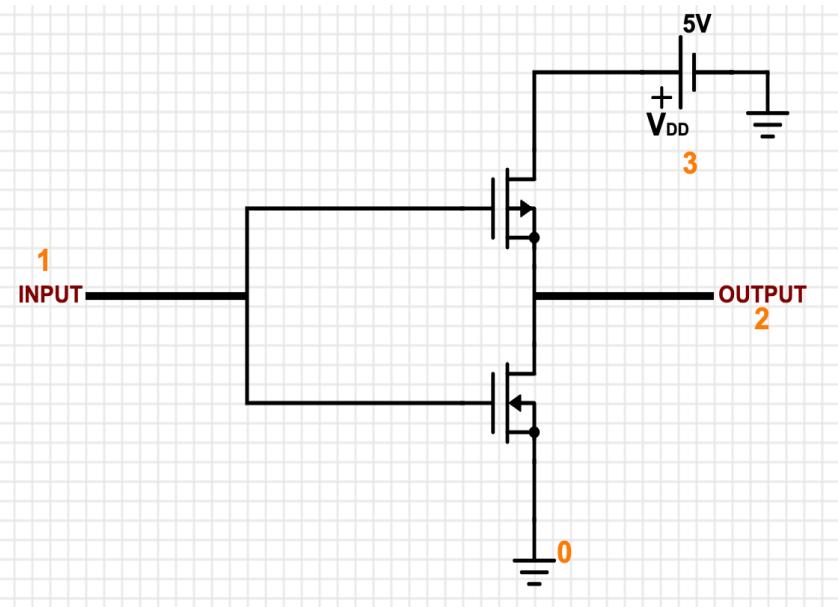


stick diagram

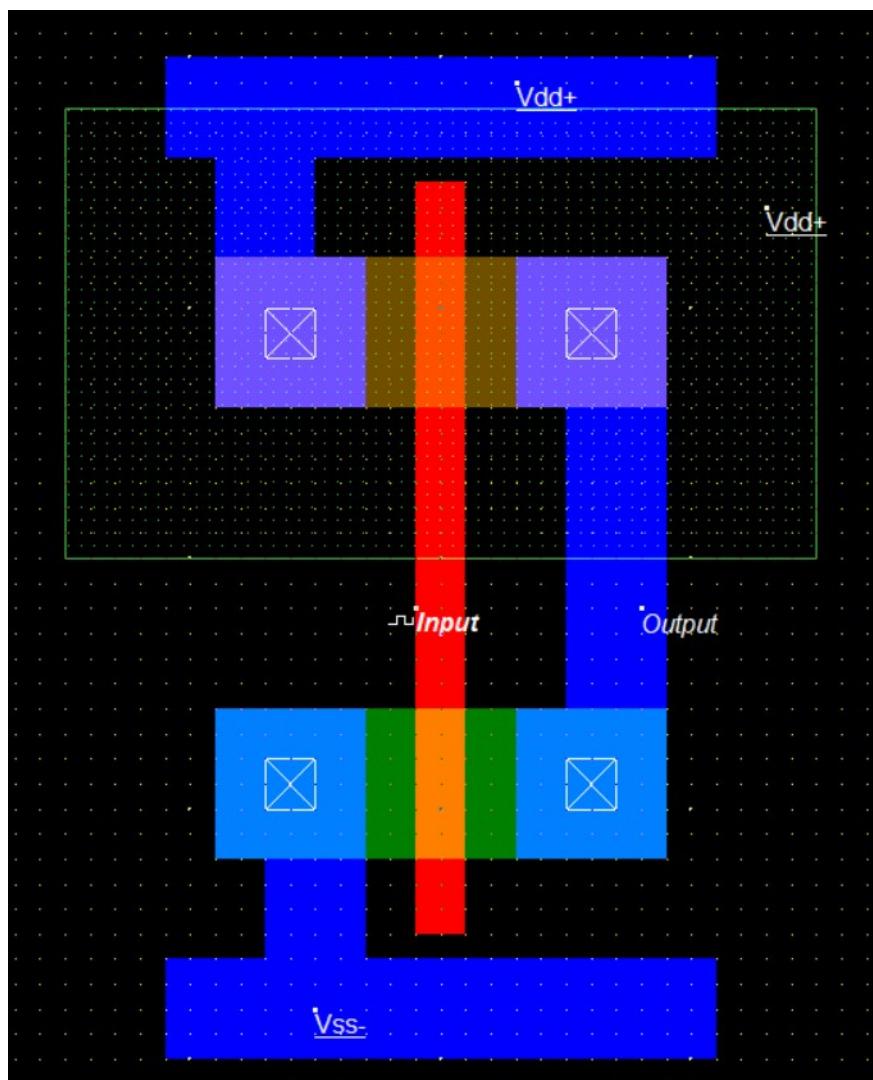


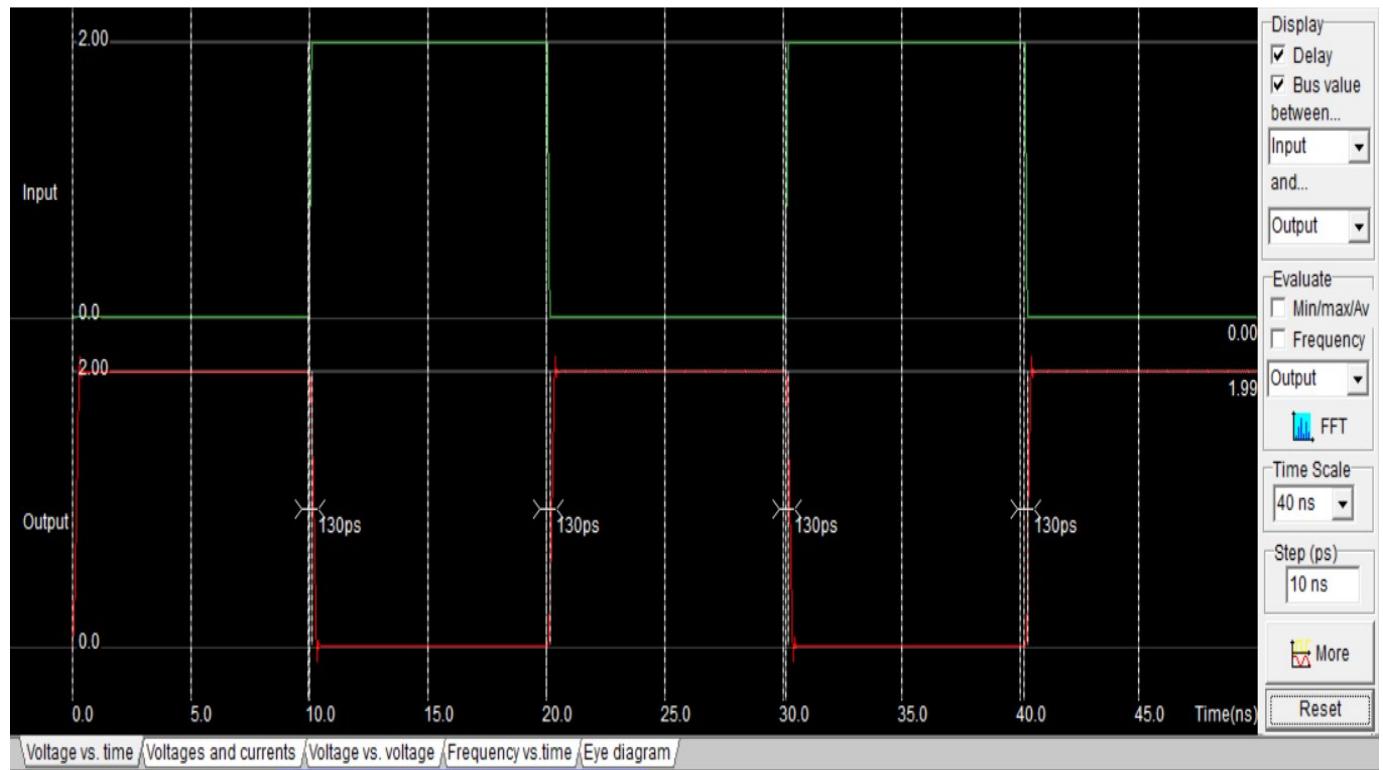
Some of the used circuits in projects:

CMOS INVERTER:

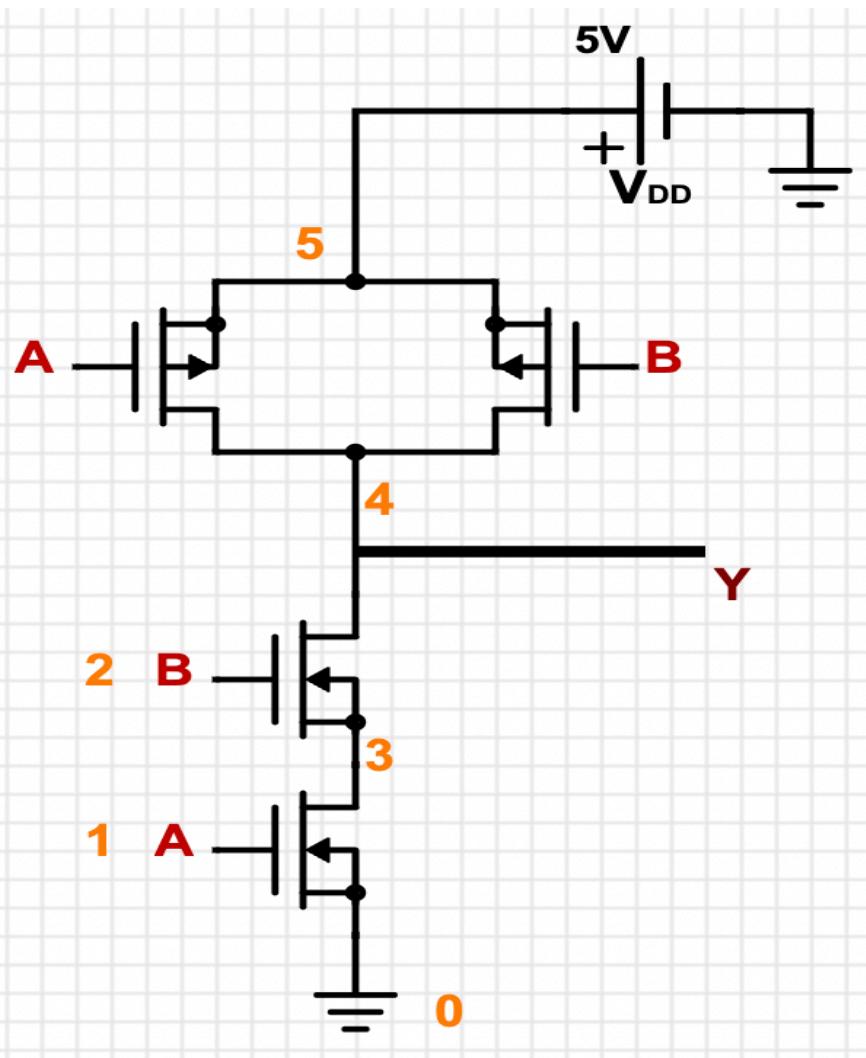


LAYOUT & OUTPUT:

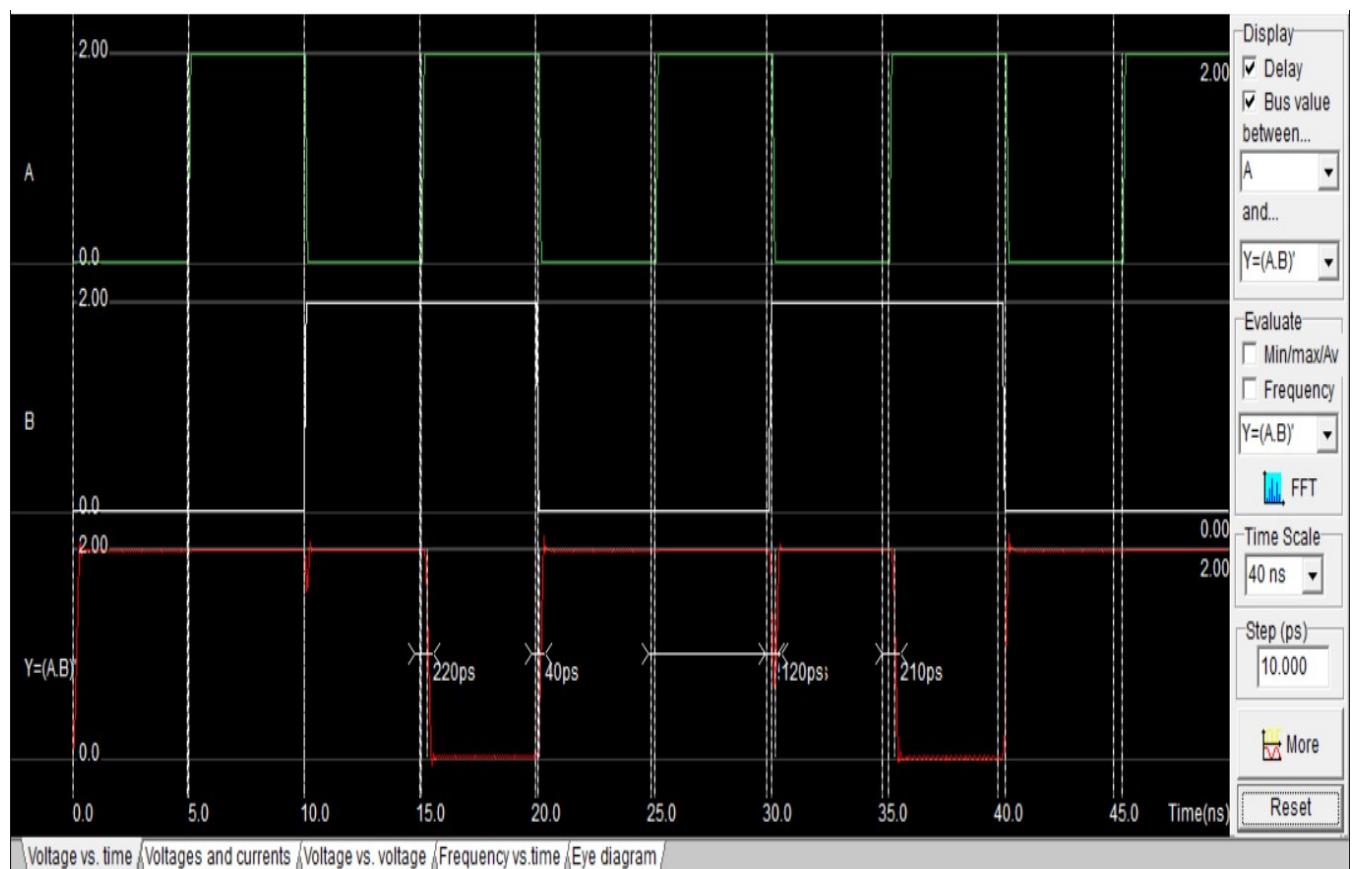
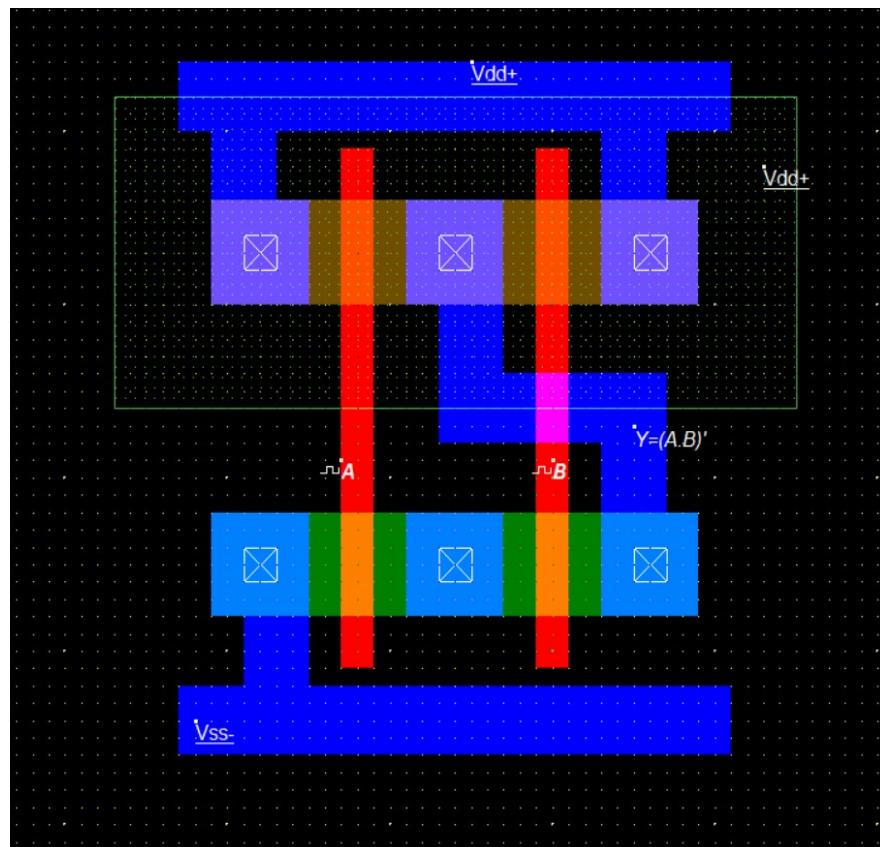




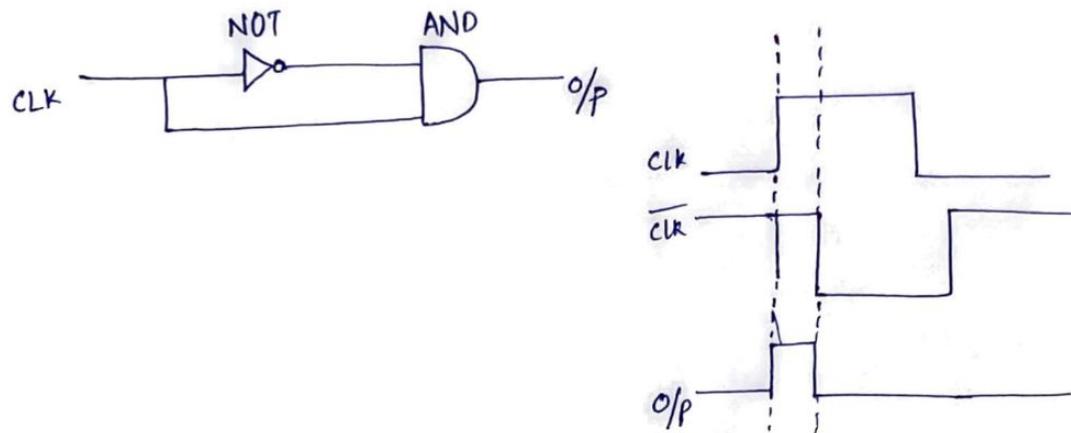
CMOS 2-Input NAND Logic Gate:



LAYOUT & OUTPUT:

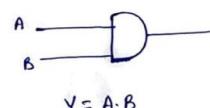


- ✓ Clock cannot be directly given to the Flip Flops as it only takes rising edge detected clock.
- ✓ Hence, Add a sub-circuit for detecting rising edges of clock.
- ✓ That is done by using the following circuit:

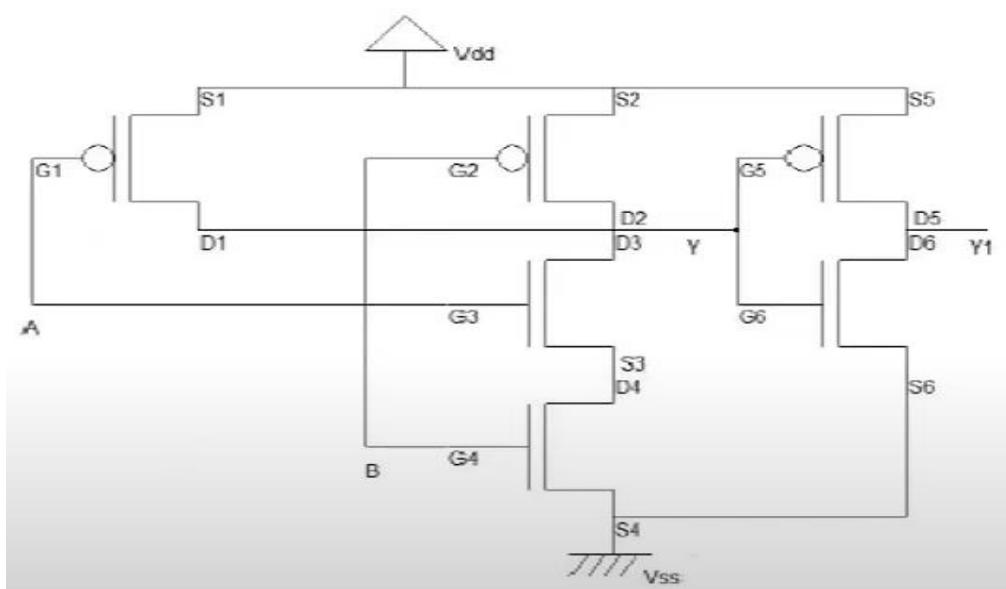


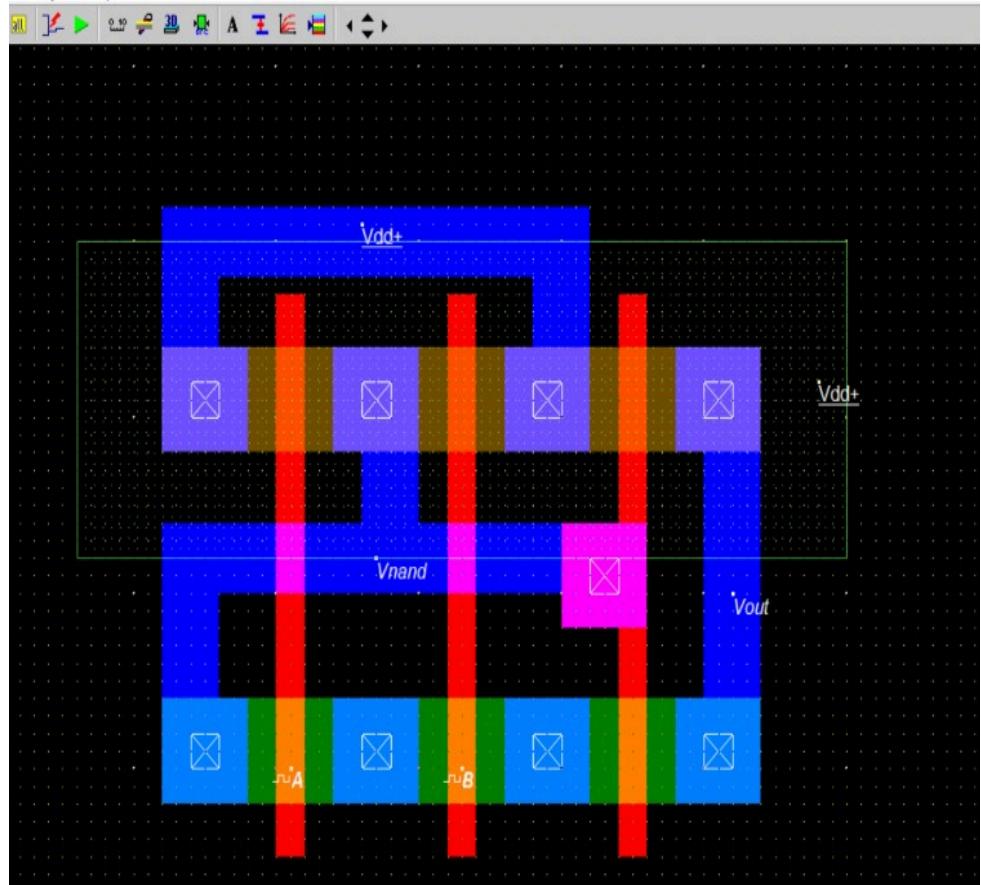
- ✓ NOT gate is for generating propagation delay which then detects the proper rising edge of the clock.
- ✓ NOT gate, hence, was made using 9 inverters using minimum width which is $0.4\mu\text{m}$ for the maximum resistance which will generate the maximum delay.
- ✓ AND gate used is made using following diagrams:

AND GATE

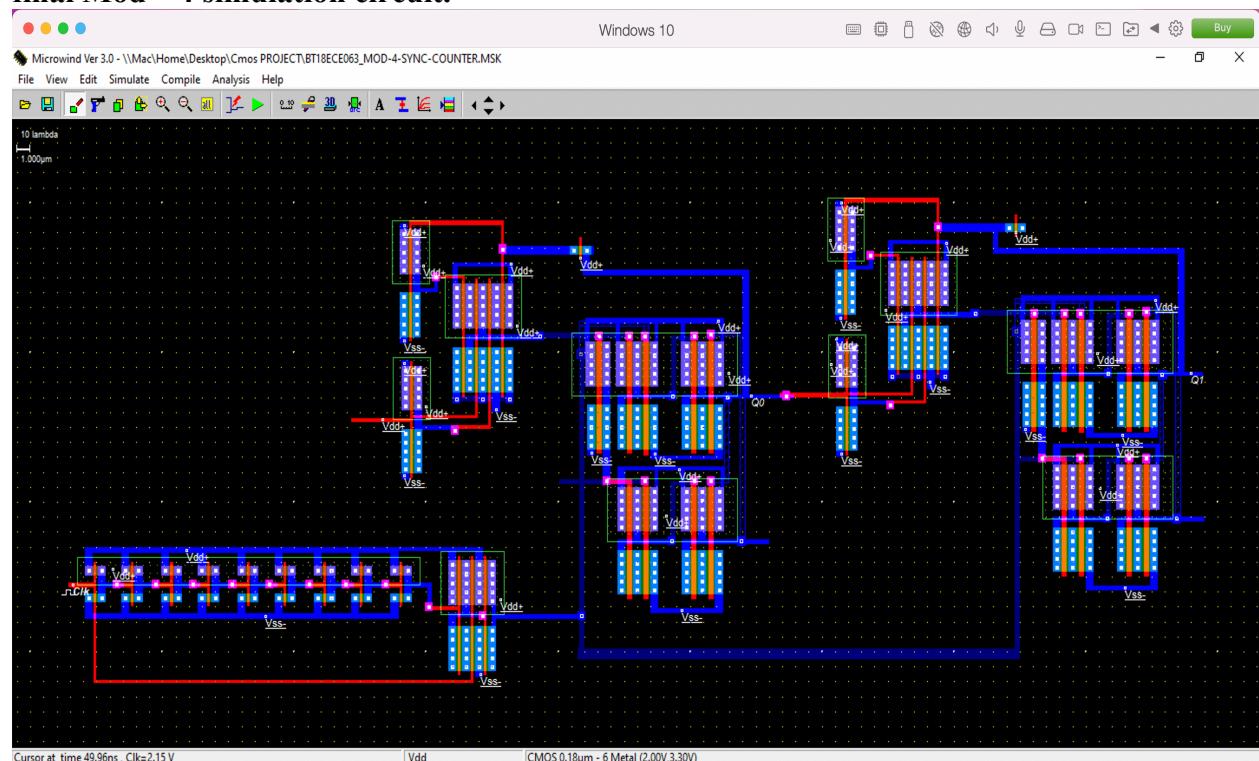


A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

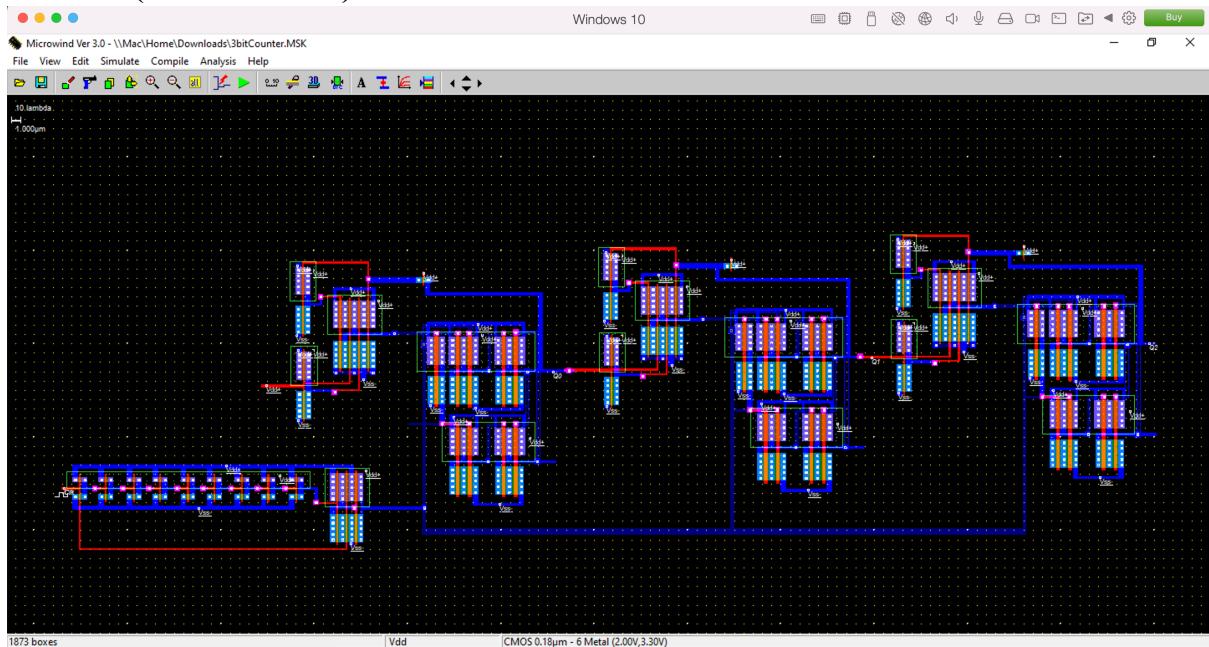




SIMULATION LAYOUT:
final Mod – 4 simulation circuit.



MOD – 8(3 – bit counter)



SIMULATION PROGRAM:

Netlist for the simulation was made using various subcircuits in a ladder like nested structure as each subcircuit was used in another subcircuit for implementation. Starting from subcircuit of inverter, then NAND. Using subcircuit of Inverter, subcircuit of XOR was made, then using it and NAND, subcircuit of D-Flip Flop was made, then using it, subcircuit of T flip flop was made and then finally using 2 T flip flop final desired circuit was made.

NGSPICE CODE:

MOD 4 Synchronous up counter:

```
.subckt inverter 1 2 3
mp 2 1 3 3 pmod w=200u l=1u
mn 2 1 0 0 nmod w=100u l=1u
.model pmod pmos level=54 version=4.7
.model nmod nmos level=54 version=4.7
.ends
```

\$ D G S B

\$write the subckts of different gates that are necessary

```

.subckt nand 1 2 4 5

mp1 4 1 5 5 pmod w=200u l=1u

mp2 4 2 5 5 pmod w=200u l=1u

mn1 3 1 0 0 nmod w=200u l=1u

mn2 4 2 3 3 nmod w=200u l=1u

.model pmod pmos level=54 version=4.7

.model nmod nmos level=54 version=4.7

.ends

.subckt xor 1 2 6 9

xin1 1 3 9 inverter

xin2 2 4 9 inverter

mp1 7 3 9 9 pmod w=400u l=1u

mp2 6 2 7 7 pmod w=400u l=1u

mp3 8 1 9 9 pmod w=400u l=1u

mp4 6 4 8 8 pmod w=400u l=1u

mn1 6 3 5 5 nmod w=200u l=1u

mn2 6 2 5 5 nmod w=200u l=1u

mn3 5 1 0 0 nmod w=200u l=1u

mn4 5 4 0 0 nmod w=200u l=1u

.model pmod pmos level=54 version=4.7

.model nmod nmos level=54 version=4.7

.ends

```

```

.subckt dff 1 3 7 8
xnan1 1 3 4 8 nand
xinv1 1 2 8 inverter
xnan2 3 2 5 8 nand
xnan3 4 6 7 8 nand
xnan4 5 7 6 8 nand
.model pmod pmos level=54 version=4.7
.model nmod nmos level=54 version=4.7
.ends

.subckt tff 1 4 3 5
xd1 2 4 3 5 dff
xx1 3 1 2 5 xor
.model pmod pmos level=54 version=4.7
.model nmod nmos level=54 version=4.7
.ends

.subckt counter 1 2 3 4
xt1 4 1 2 4 tff
xt2 2 1 3 4 tff
.model pmod pmos level=54 version=4.7
.model nmod nmos level=54 version=4.7
.ends

```

Vdd 4 0 dc 5V

Vclk 1 0 pulse(0 5 0 0 0 1ns 15ns)

xcentr 1 2 3 4 counter

.model pmod pmos level=54 version=4.7

.model nmod nmos level=54 version=4.7

.tran 0.1ns 100ns

.control

run

plot V(2) title 'Q0' xlabel 'Q0'

plot V(3) title 'Q1' xlabel 'Q1'

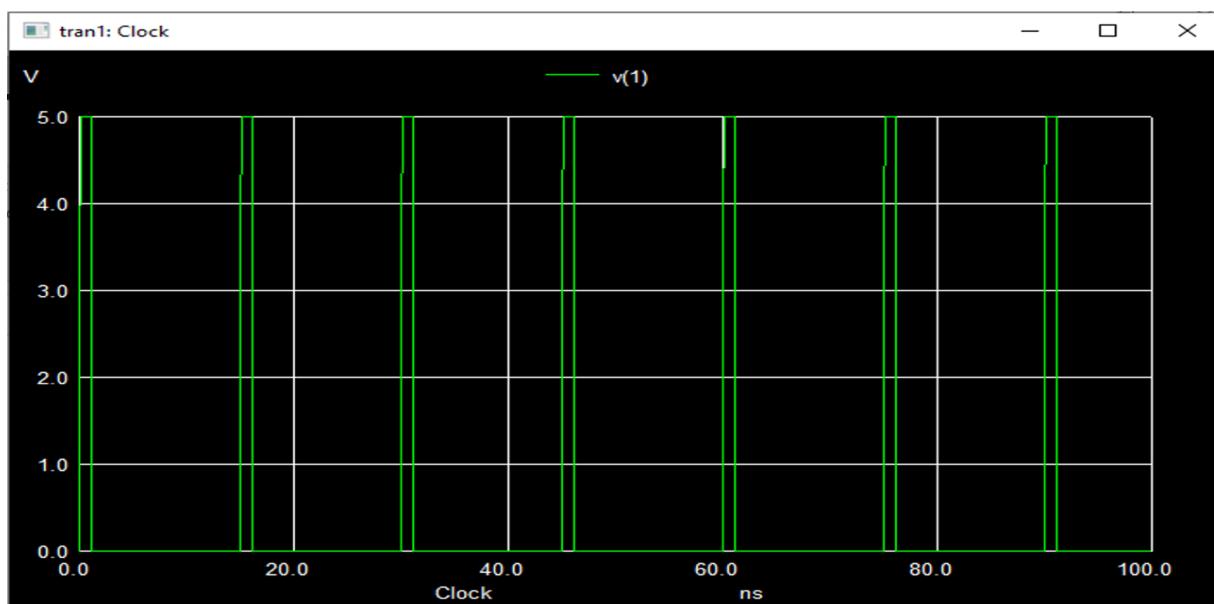
plot V(1) title 'Clock' xlabel 'Clock'

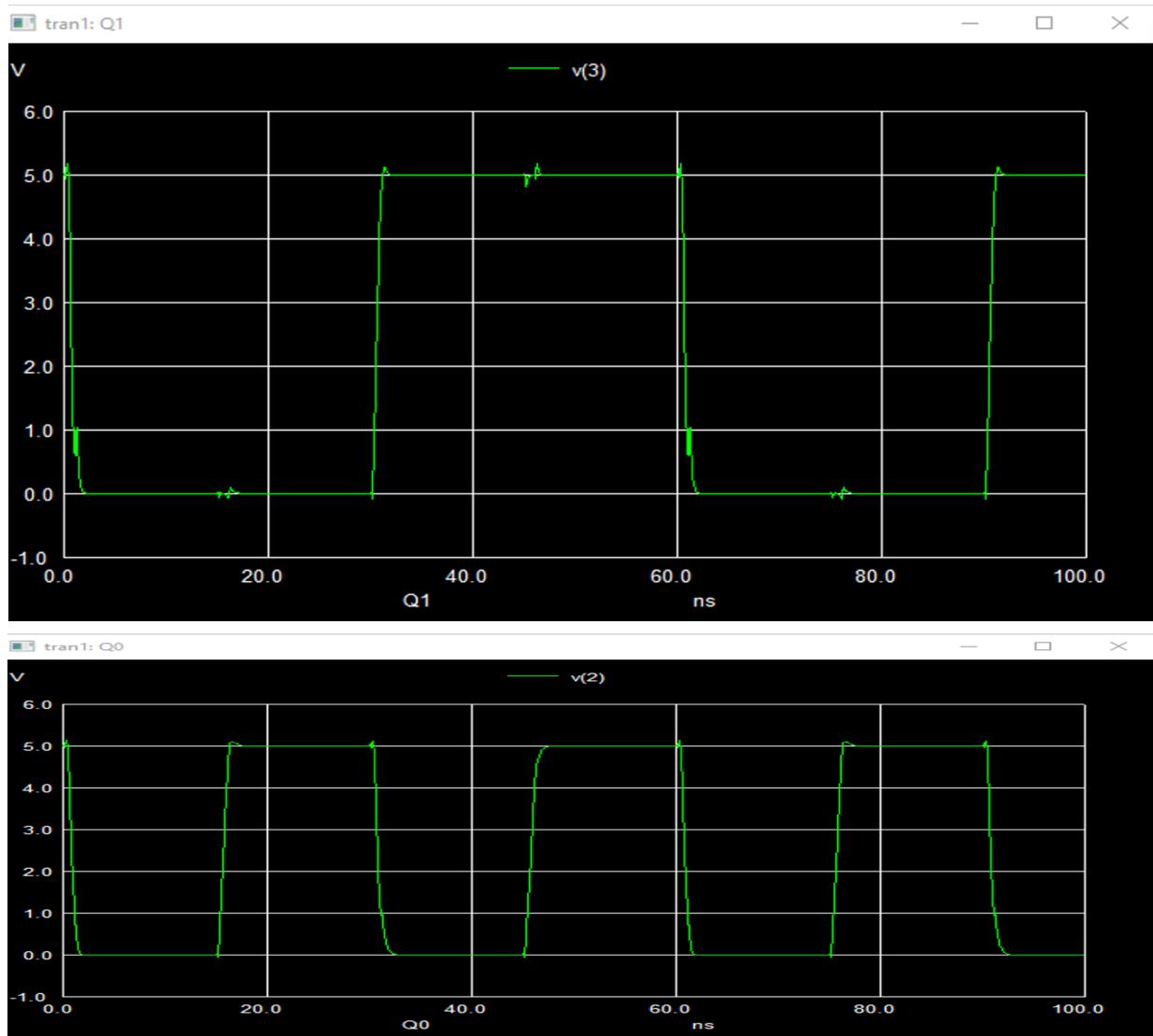
.endc

.end

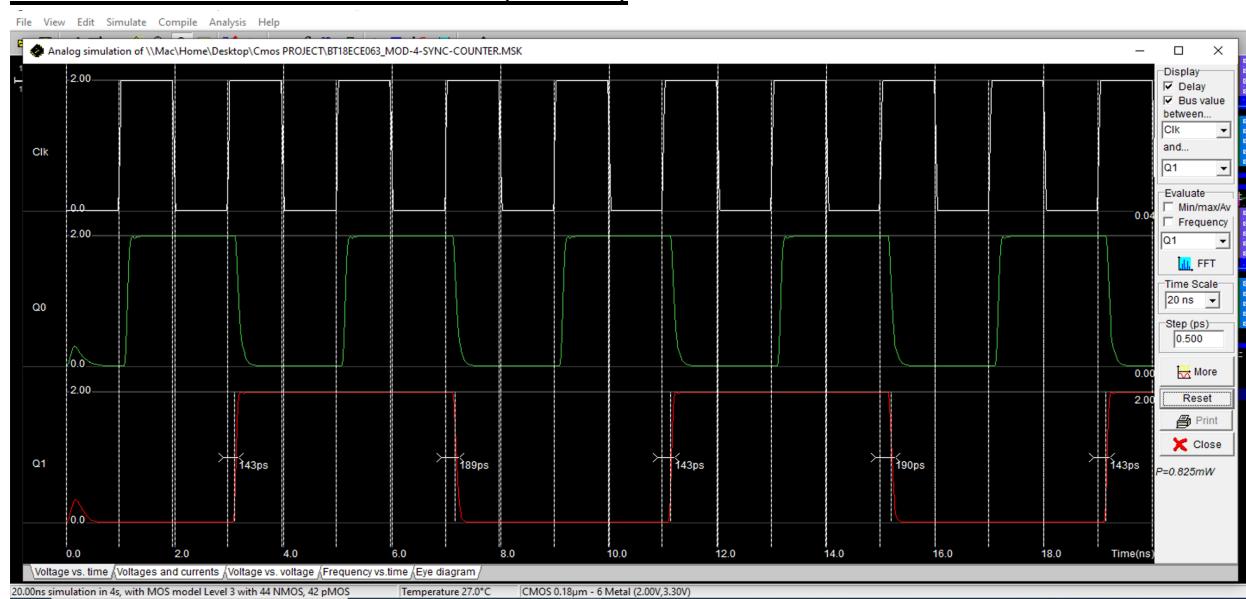
RESULTS:

NGSPICE OUTPUT:

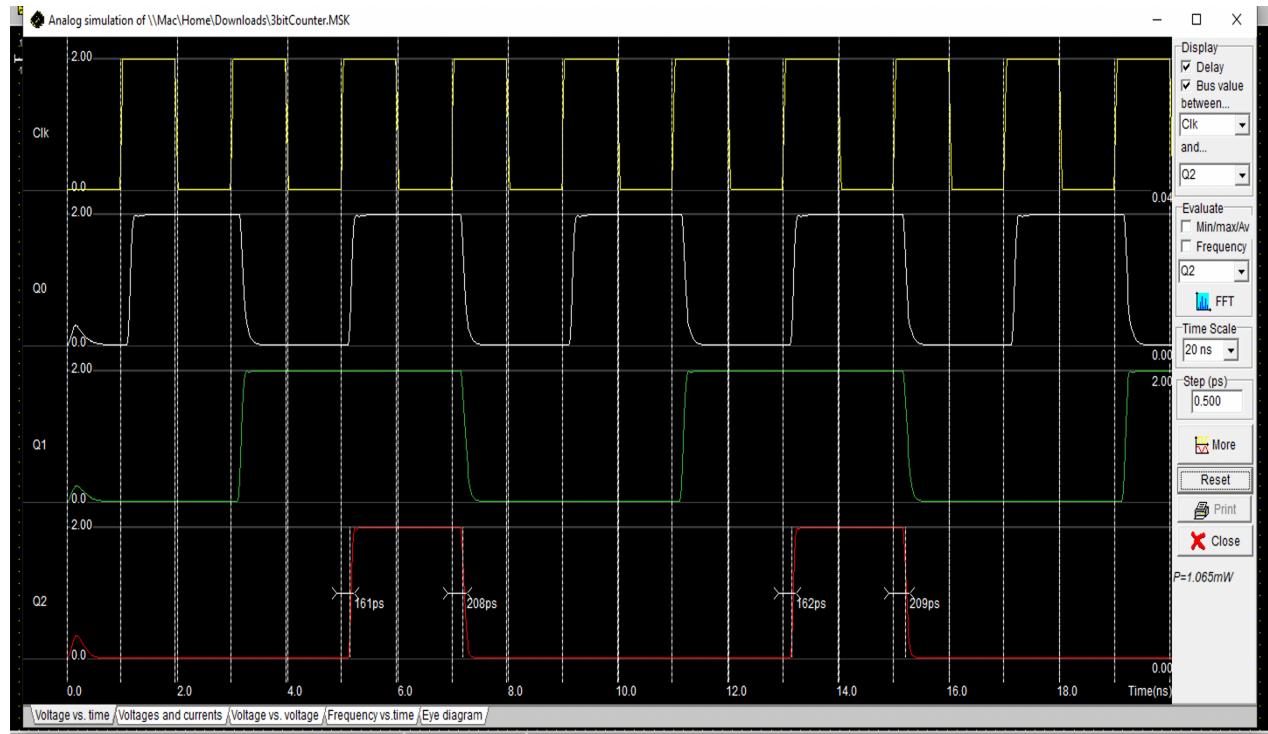




MICROWIND LAYOUT OUTPUT(MOD-4):



MICROWIND LAYOUT OUTPUT(MOD-8):

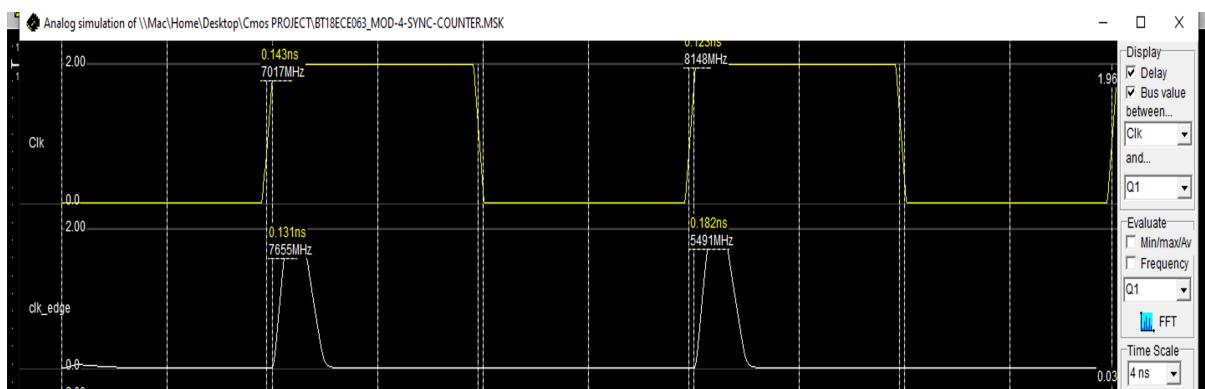


- ✓ Clock in the netlist was given which was observed after the positive clock edge detection circuit, for simplicity.
- ✓ Therefore, each clock pulse denotes a rising edge of the clock.

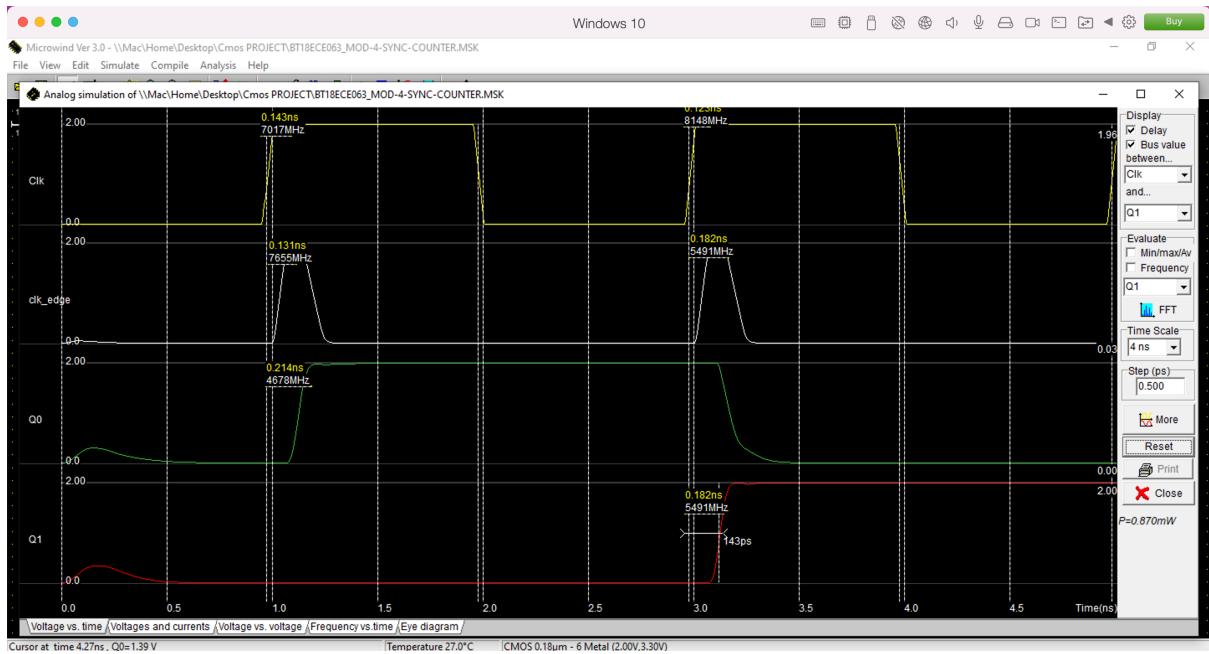
Conclusion:

CLOCK TIMINGS:

There was observed a trade-off between clock speed and the output quality. Due to the clock's edge detection circuit, there was a considerable amount of propagation delay that was introduced between the actual clock that was given as input and the clock that was sent to the T flip flops. This Delay was average of 0.06ns as seen from the below diagram:

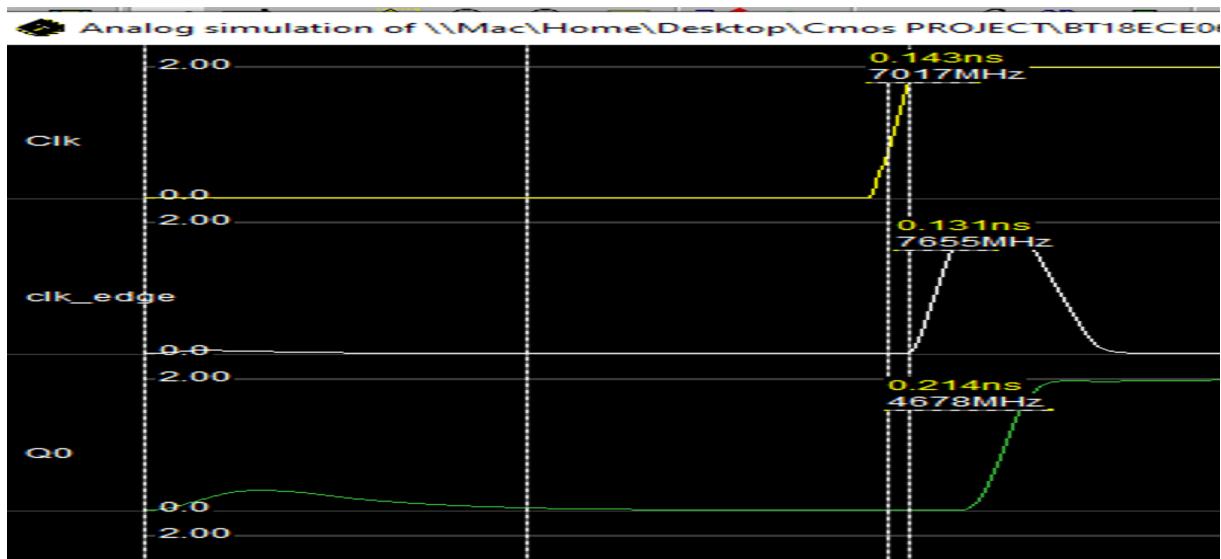


The fastest that the circuit can run on clock is observed to be 300ns input clock period. If it is lesser than 300ns, the clock rising edge detector starts missing the rising edges due to the propagation delay. This can be seen from the following diagram:



Therefore the maximum clock speed that can be used with this circuit is 3.33 MHz, any more frequency and the circuit will start to miss clocks.

The maximum propagation delay that was observed was: 0.084ns which is 28% of the clock period. This is seen from the following diagram:



Thus, the fastest any clock that can be given to this circuit is of 3.33 MHz In another words, this circuit can perform as much as 333,333 count arounds per second.

The widths of all P-MOS and N-MOS were kept as $3\mu\text{m}$ for reducing the propagation delay. The outputs were successfully obtained and this project was a success.

In this hands-on lab session, got familiar with new Software (MicroWind) -layout designing in MicroWind. Designed the Layouts for taken project. obtained the desired output by using MicroWind & ngspice.