Project Report

7-bit Up Counter

EGEN 5301W – Integrated Circuits

Submitted

By

Smit Patel

Student Id:- 101317783

Date:- 28 – 04 – 24

❖ Introduction

(1) Design Overview: -

- In this project, I have designed a 7-bit synchronous up counter. It wil start from a decimal number of 103 (according to my preset state) and will end at a decimal value of 127. After then, it will roll over to the 0 and will count further from 0 to 127. In this way, the counter will cover the whole range from 0 to 127 with a starting state of 103.
- The flip-flop chosen in this project is T-flip flop as it makes the designing for my up-counter less computing. Moreover, I have placed the external signals like enable, preset, and reset to enable the function of flip-flops. The starting state, i.e decimal value of 103, will be achieved using the Preset and Clear signals provided to the flip flops. The reset signal sets the counter to the initial starting state whenever it has been activated and after the the counter will follow the following counting.
- In order to achieve the above-mentioned counting sequence, I have combined the two different up counter circuits such that It will fulfilled my requirement. First, I designed the up-counter circuit such that it will count from a decimal value of 103 to 127 and then roll over to zero. After completing the first circuit, I designed the regular 7-bit up counter circuit that count from a decimal value of 0 to 127. After analyzing these two circuits, I noticed that if I combined these two circuits, I would eventually achieve my target up counter circuit.

(2) Specifications

- The entire schematic of this circuit was designed in Cadence Virtuoso 45nm Technology.
- Below are the specifications of the components that were used.

Component Name	Voltage Level	Width	Length
PMOS	1.1 V	240nm	45nm
NMOS	1.1 V	120nm	45nm

Table 1.1 Transistor Parameters

Voltage Source	Purpose	Voltage level
Vpulse	Input to many Parameters of digital blocks	0 – 1.1 V
Vdc	Power Supply	1.1 V

Table 1.2 Voltage Sources

(3) Starting State :-

- The starting number of the sequence must be a number that is based on my student Id.
- It will be calculated as per the formula given to me.
- Starting State = HI + A + B + C + D + E + F +G.
- For example, my student Id = 101317783,
- **Starting State = 83 + 7 + 7 + 1 + 3 + 1 + 0 + 1 = 103.**

❖ System Design

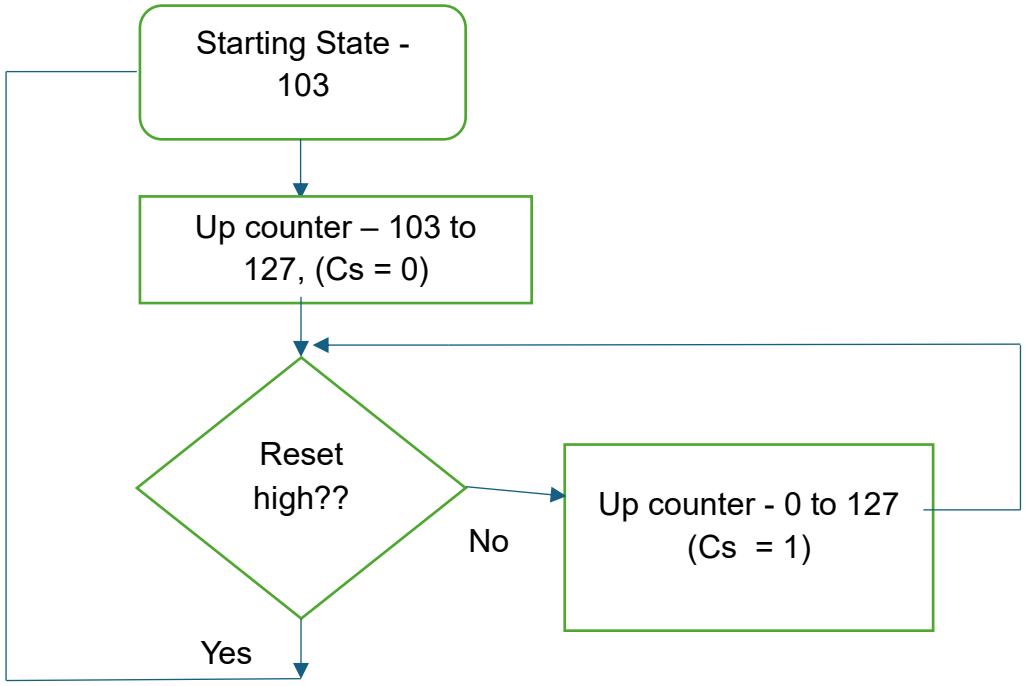

Fig 1.1 Counter Working Flowchart

- After the mathematical Caculation, I have found that there are some different inputs that we must need to give to make the circuit work . I used a Cs signal , namely a control signal, that will change alternatively for these two different circuit inputs such that the whole sequence of counter will be counted

- The counter will start from the starting state of 103, and it will increment to each interation and reach to the 127. When it will reach to 127, the output of all flip - flops will be high

that makes the Cs signal change. After then, the Cs signal allows the flip – flops to take those inputs that are required to count from a decimal value of 0 to 103. Moreover, There is a system which checks the reset signal is high or not.

- If the reset signal is high, the counter will roll over to it's preset state and follow the further seqeunce.

❖ Design Of Individual Blocks

1. Inverter

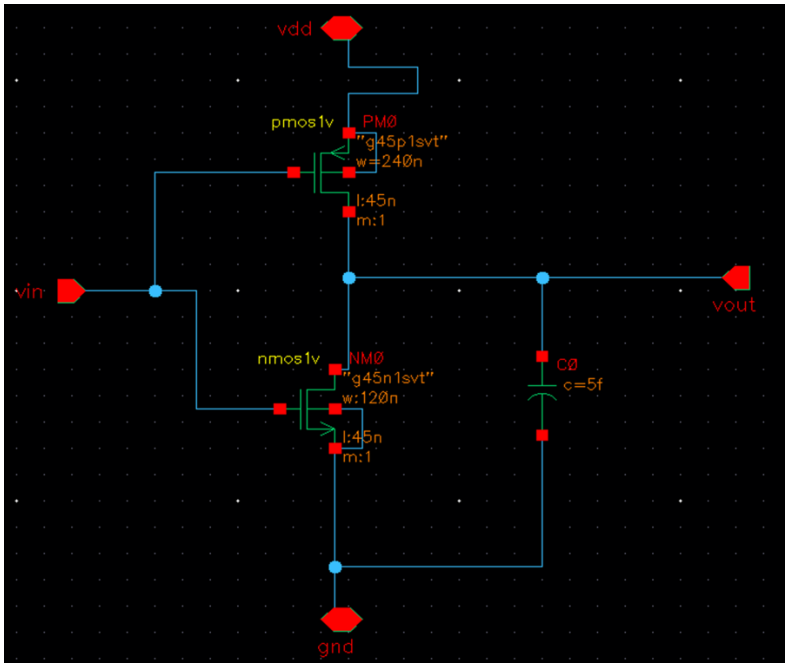

Fig 1.2 Inverter Schematic

Note :- The purpose of load capacitor is to minimize the noise level, increase the output strength and decrease the spike.

Vin	Vin'
1	1
1	0

Table 1.3 Inverter Truth Table

2. AND Gate

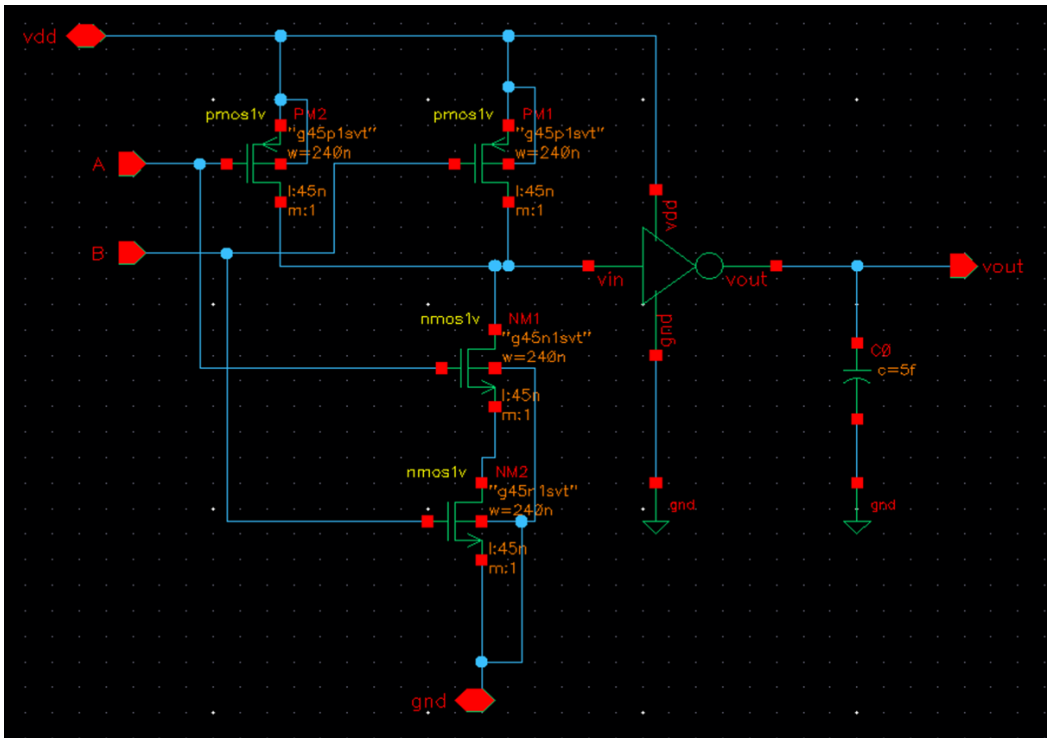

Fig 1.3 AND Gate Schematic

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

Table 1.4 AND gate Truth Table

3. OR Gate

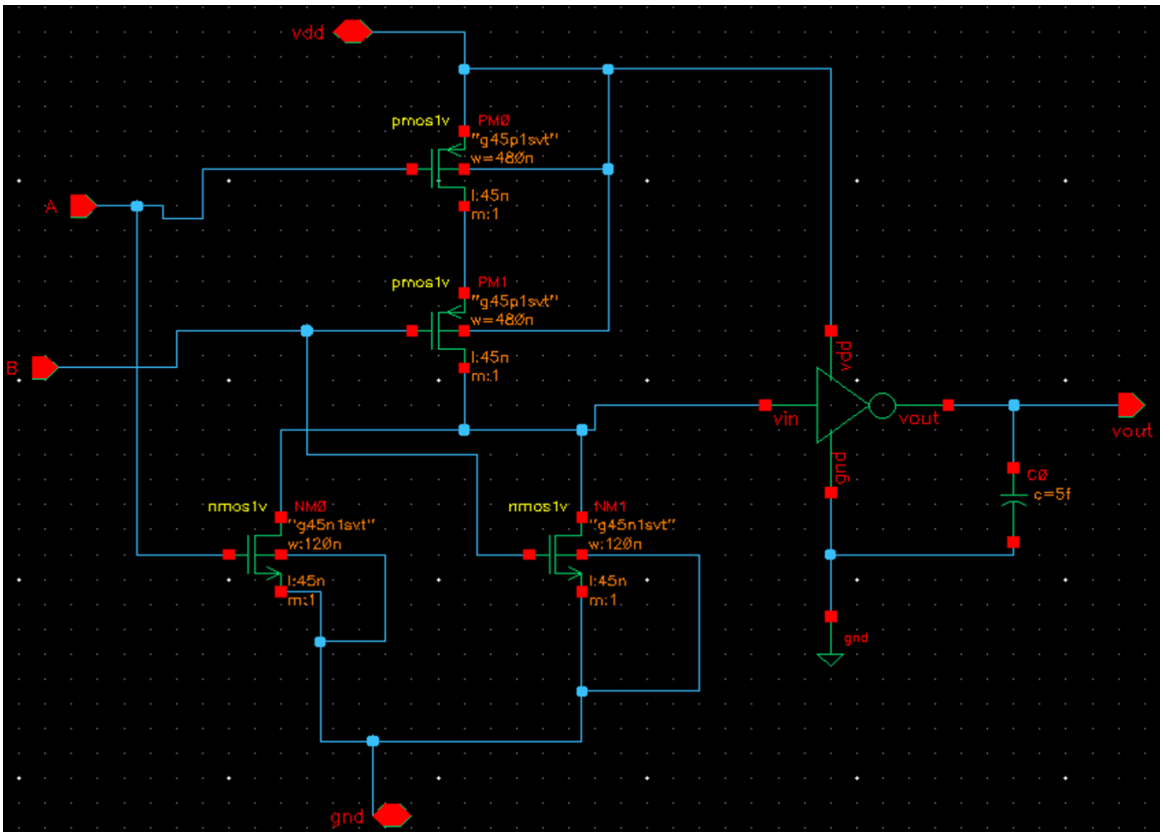

Fig 1.4 OR Gate Schematic

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

Table 1.5 OR gate Truth Table

4. NAND Gate

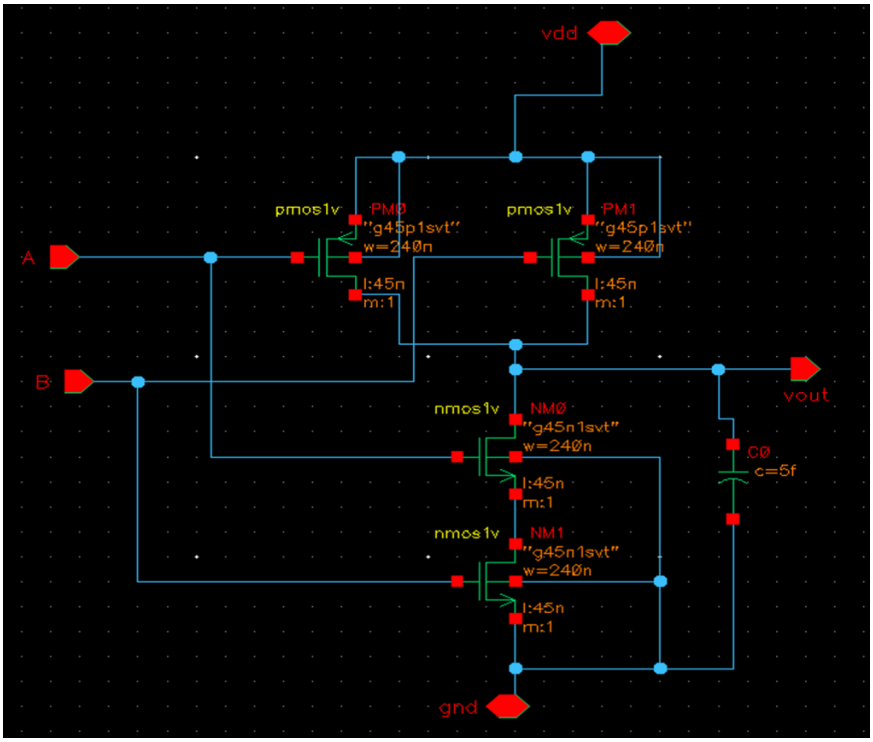

Fig 1.5 NAND Gate Schematic

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

Table 1.6 NAND gate Truth Table

5. NOR Gate

A	B	Y

0	0	1
0	1	0
1	0	0
1	1	0

Table 1.7 NOR gate Truth Table

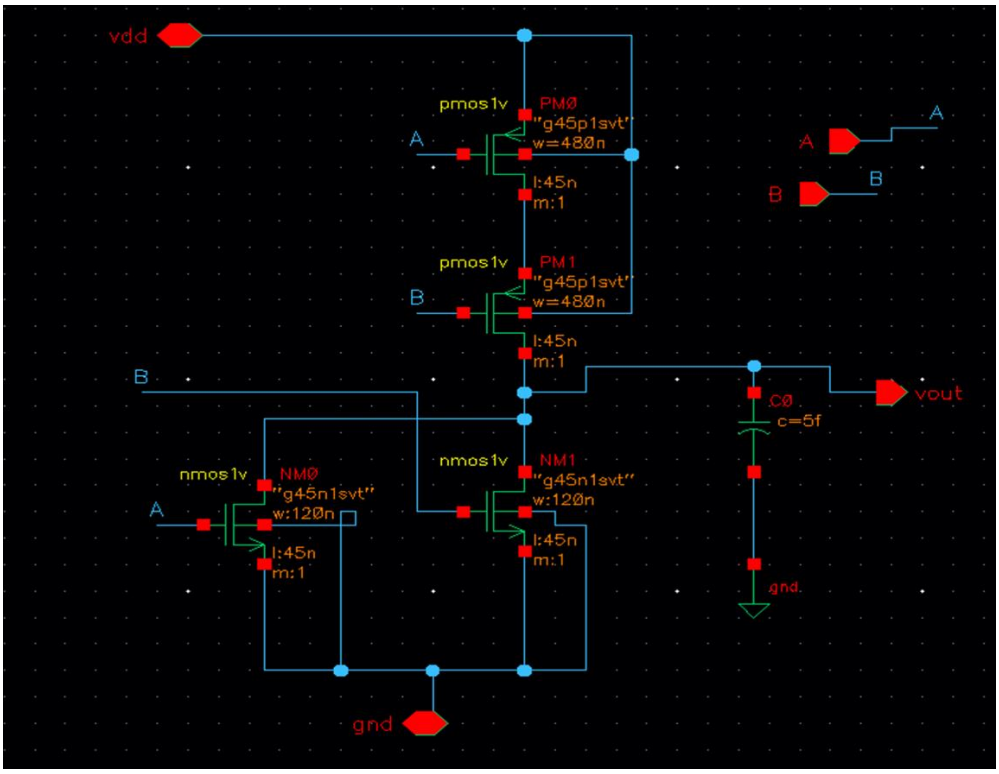

Fig 1.6 NOR Gate Schematic

6. EX - OR Gate

A	B	Y

0	0	0
0	1	1
1	0	1
1	1	0

Table 1.8 EX-OR gate Truth Table

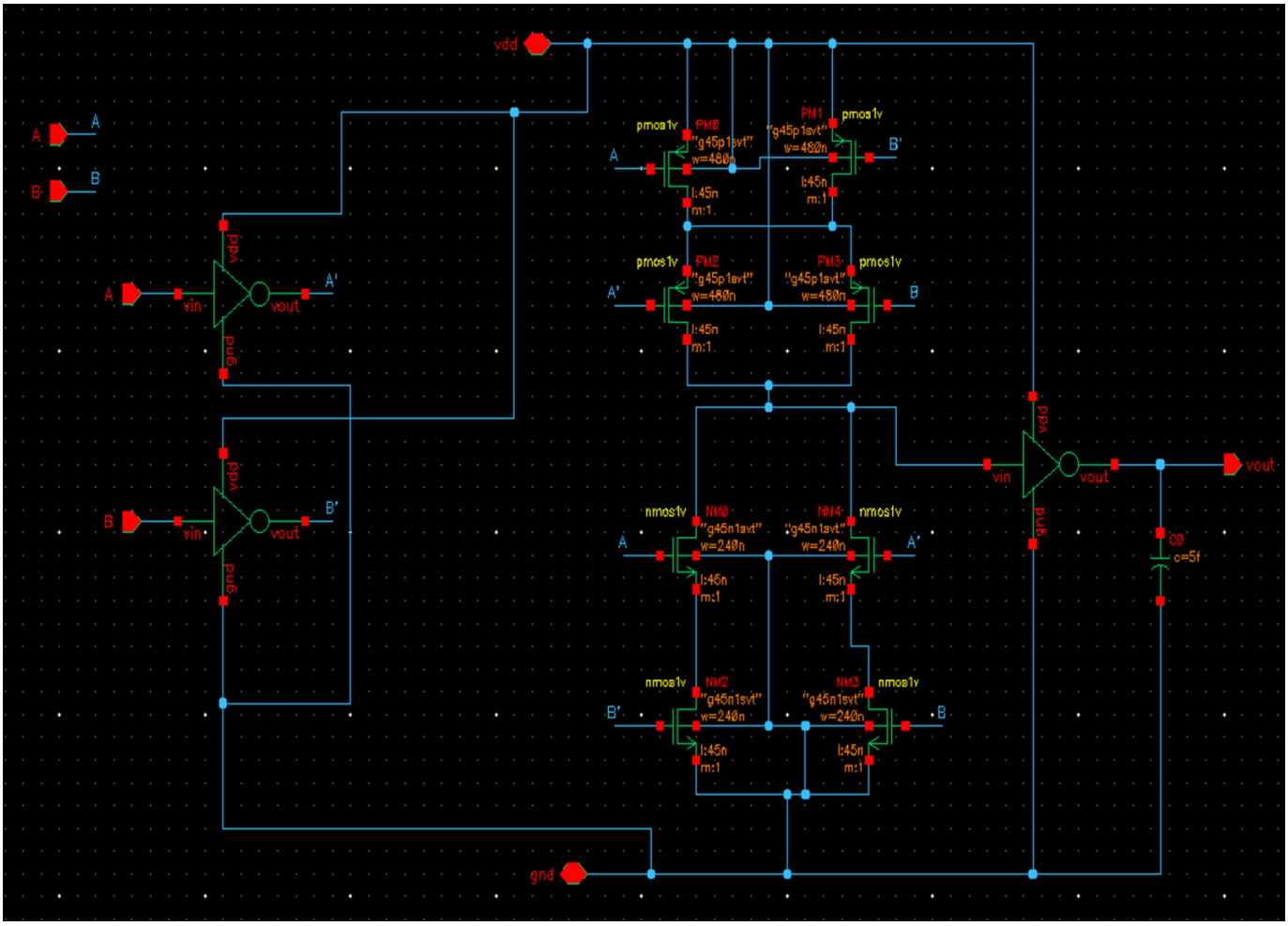

Fig 1.7 EX - OR Gate Schematic

7. EX - NOR Gate

A	B	Y

0	0	1
0	1	0
1	0	0
1	1	1

Table 1.9 EX-NOR gate Truth Table

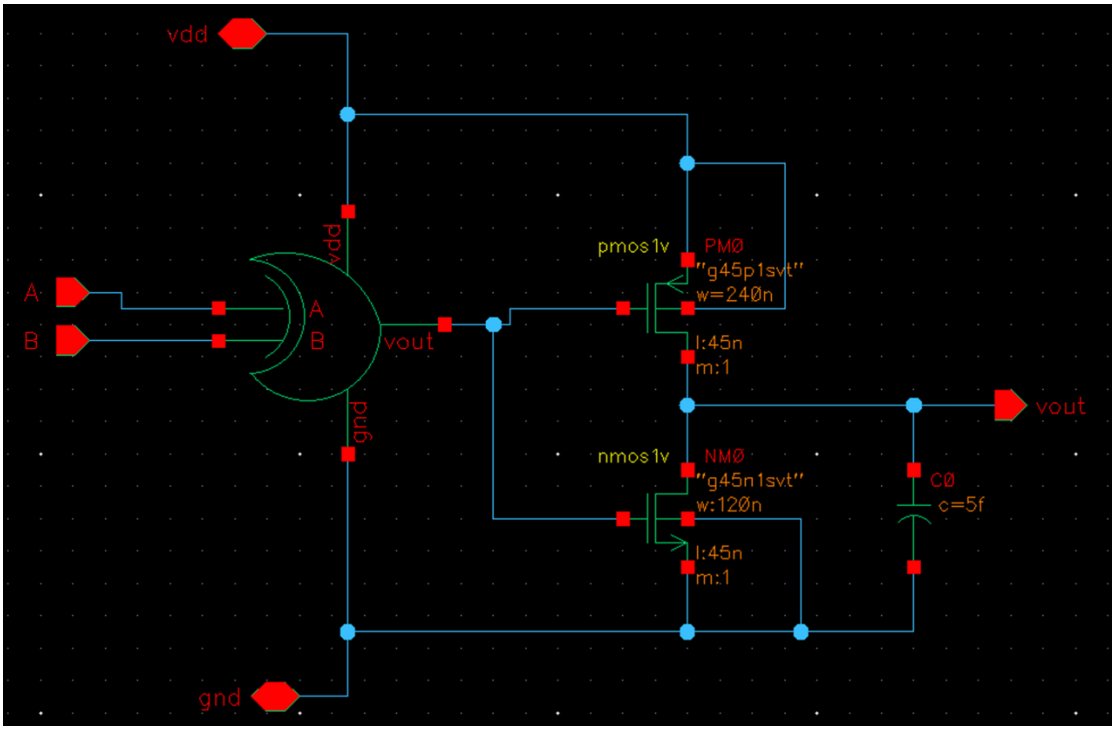

Fig 1.8 EX - NOR Gate Schematic

8. T – Flip Flop

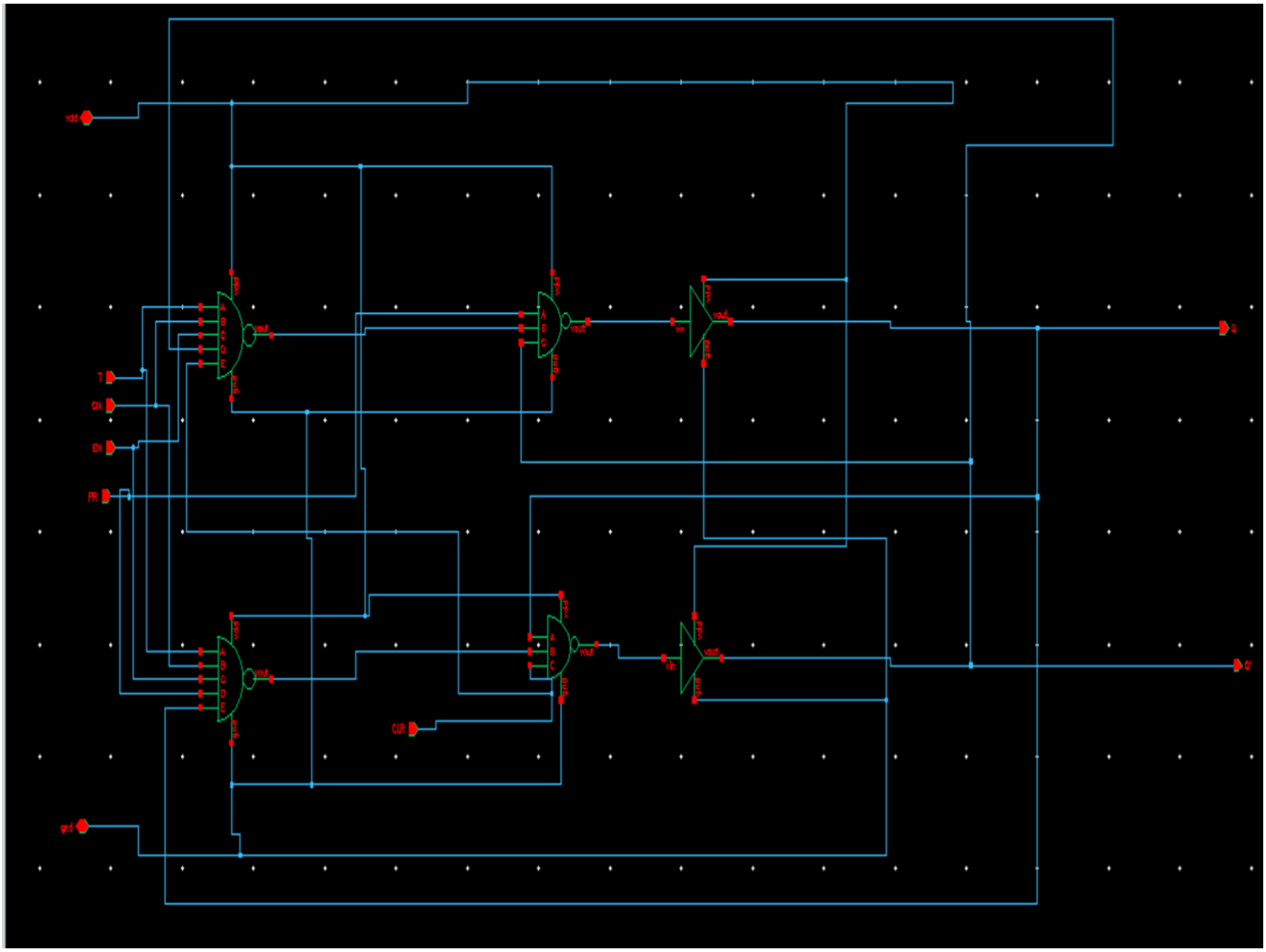

Fig 1.9 T – Flip Flop (based on SR Latch)

9. Buffer

Vin	Vout
0	0
1	1

Table 1.10 Buffer Truth Table

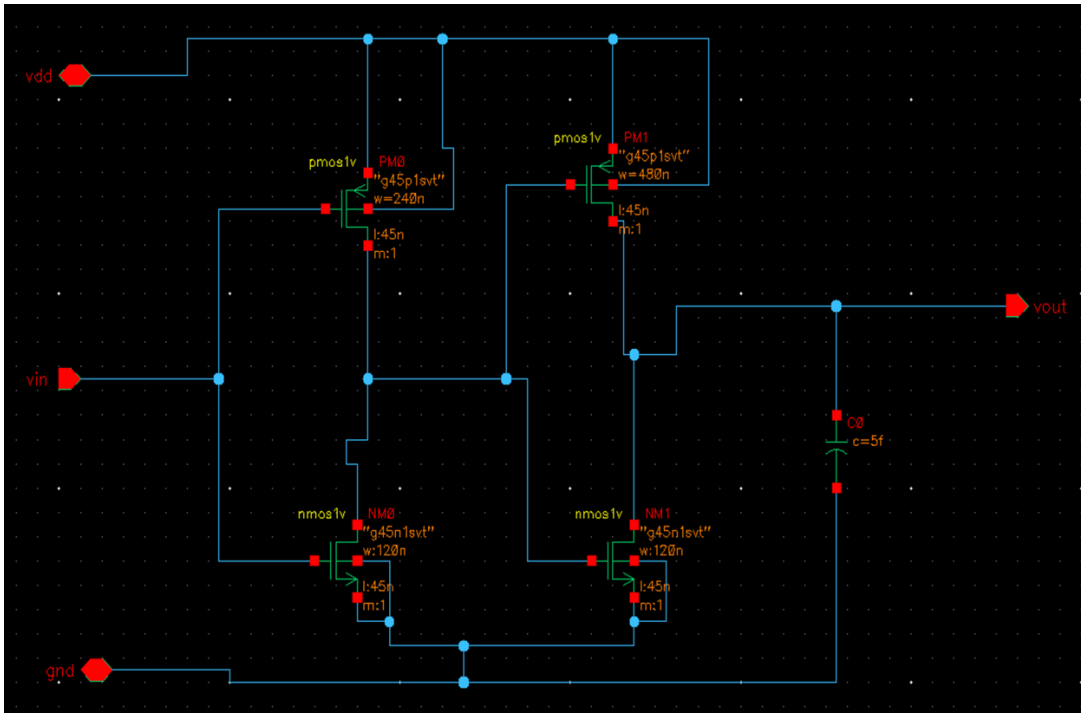

Fig 1.10 Buffer (Single Stage)

10. Multiplexer – 2 x 1

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

Table 1.11 Multiplexer 2 x 1

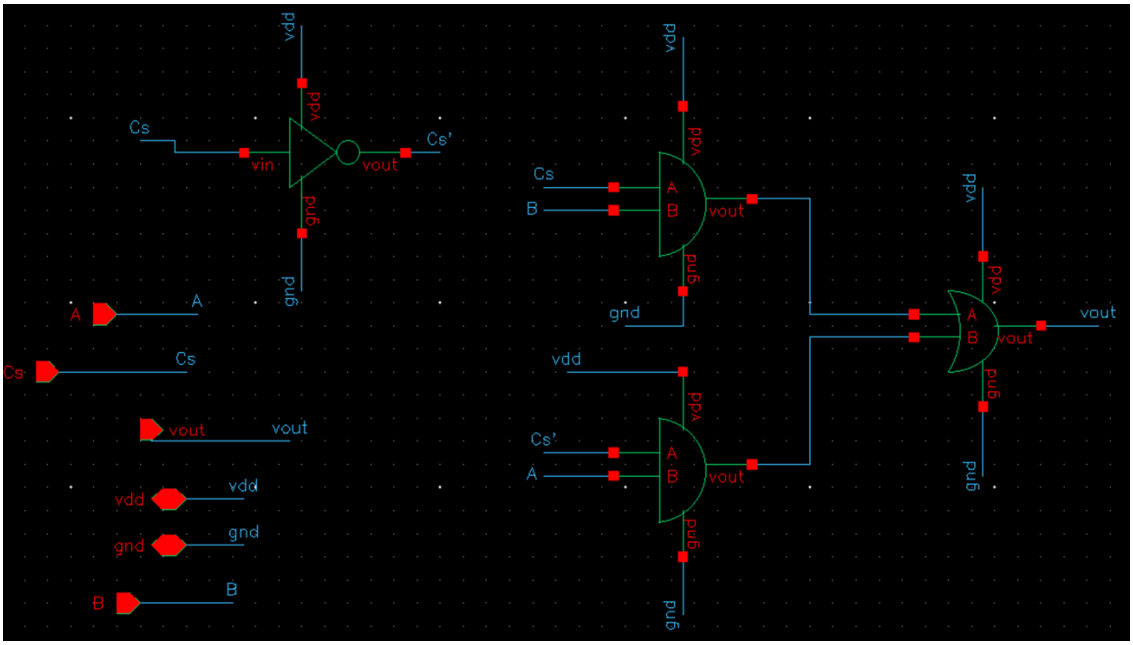

Fig 1.11 2x1 Mux

- **Why I used the T - flip flop ??**

- The reasons for using the T-flip flops are that they offer simplicity, efficiency, and symmetrical operation that will be useful for this counter application.

Advantages :-

Simplicity: Compared to D-flip flops, T-flip flops have a simpler design because they have just one input—the toggle (T) input—which, when activated, toggles the output state. Reduced power consumption and a smaller circuit area can result from this simplicity.

Reduced Gate Count: T-flip flops typically require fewer logic gates compared to D-flip flops, making them more efficient in terms of gate count and layout complexity.

Symmetrical Operation: A T-flip flop operates symmetrically about the input signal. Toggling the T input will always cause the output state to change, regardless of the flip flop's current state. T-flip flops' symmetrical behaviour makes it easier to analyze and design circuits with them.

Disadvantages :-

Constrained Functionality : The input signal is the only way that T-flip flops can switch the state of their output. They cannot, like D-flip flops, directly set or reset the output to a particular state using their data (D) input.

Undefined State: Unlike D-flip flops, which have a clear distinction between the input and output states, T-flip flops may be less intuitive to understand at first glance, especially for beginners. The toggling behavior might make it less obvious how the flip flop will behave under different conditions.

- **Implementation of Enable, Preset and Clear Signals**

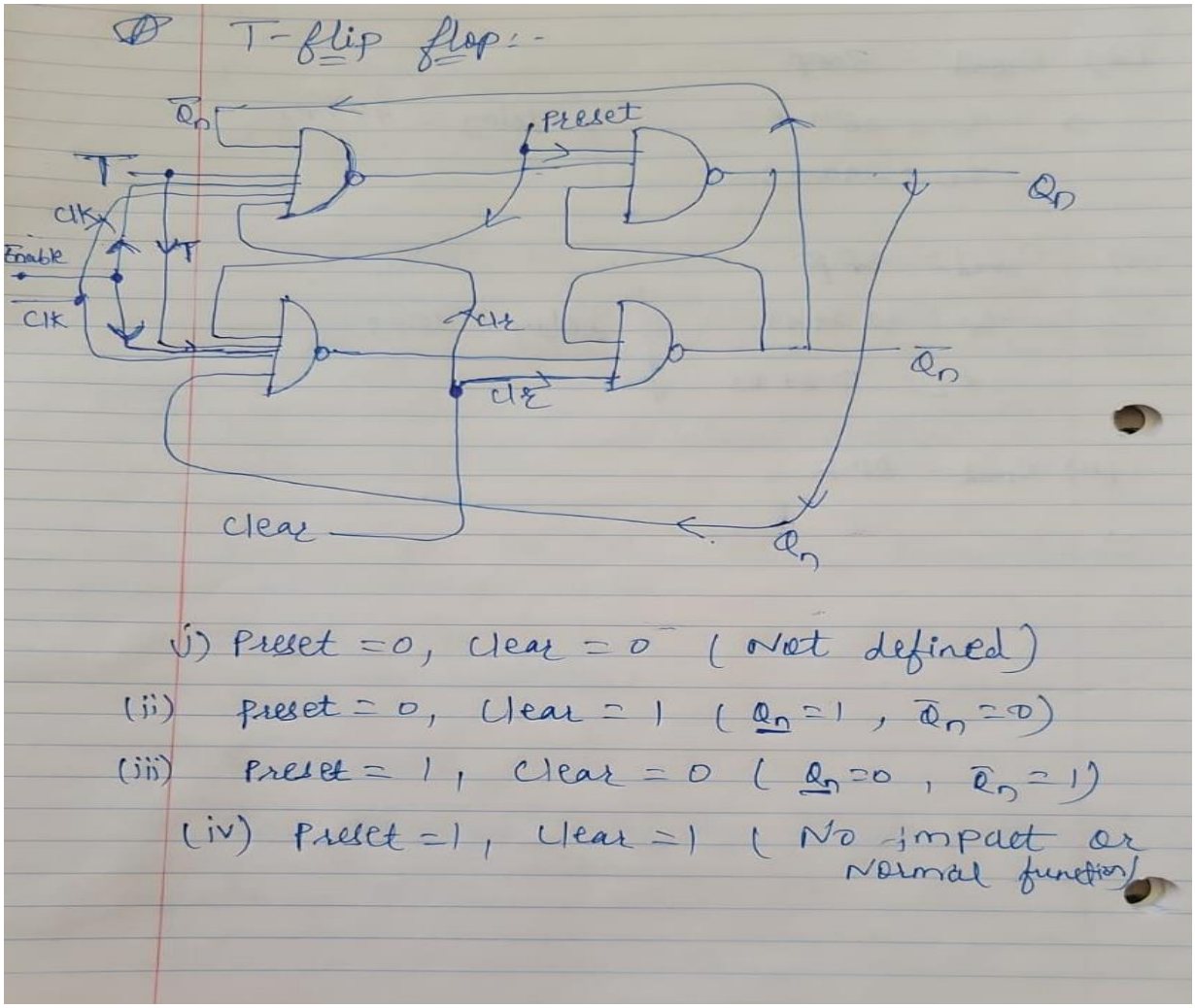

Fig 1.12 T – flip flop theoretical design

- As you can see from the above , It is a circuit of 1 – bit T – flip flop. Let see the implementation of Enable first.

1. **EN = 0 :-** When the EN = 0 , the output of the first two NAND gates is high and it will give the further input to the second NAND gate that makes the output equal to previous state.
2. **EN = 1 :-** However, when the EN = 1, the output of the first two NAND gates is what it was before the ENABLE signal. Therefore, there will not be any change in the output of first two NAND gates and the output of second NAND gates will toggle.

- The Preset and Clear inputs allow us to Set and Clear of a specific flip – flop as per our need. In this project, I have given the Preset and Clear inputs as a function of Reset Signal. Therefore, when the Reset goes high, the output of all flip flops will Set and Clear in such a manner that I will get my starting State.

Preset	Clear	Qn	Qn’
0	0	X	X
0	1	1	0
1	0	0	1
1	1	Qn	Qn’

Table 1.13 Preset and Clear Implementation

❖ High – Level Discussion

- This is a cellview of final made 7 – bit custom Up Counter that counts as per my sequence.
- Inputs are namely Clock, Reset and Enable. However, that are 14 outputs of which 7 are the complements of their actual output.
- Apart from this, VDD and GND trails are there that power up this counter.

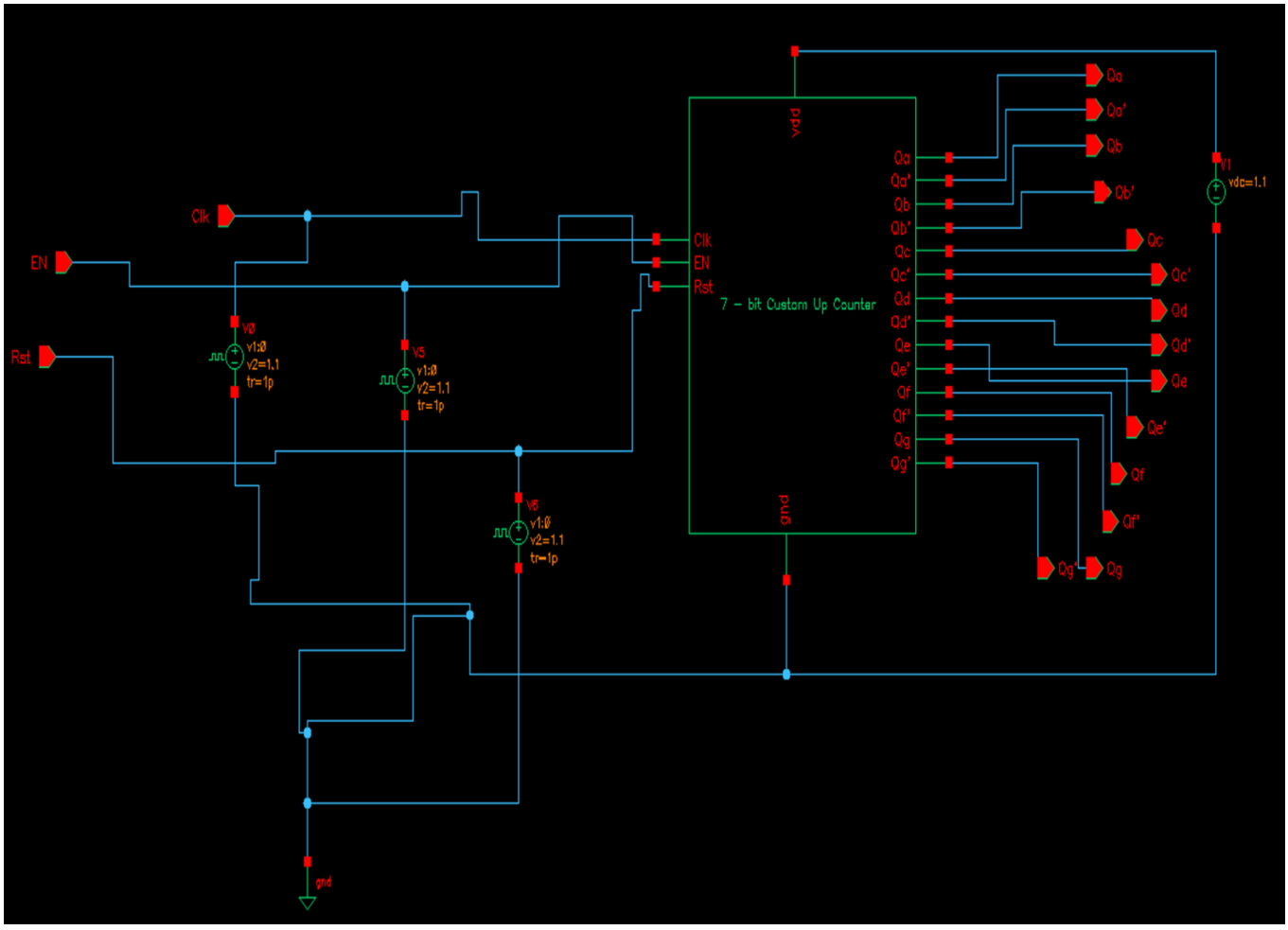

Fig 1.13 Top level Design

❖ Up Counter Circuit Schematic :-

- This whole Up Counter circuit has been divided into sub three blocks , that have a significant role for running this counter circuit. Let's discuss each of ones in brief.

1. T – flip flops Block

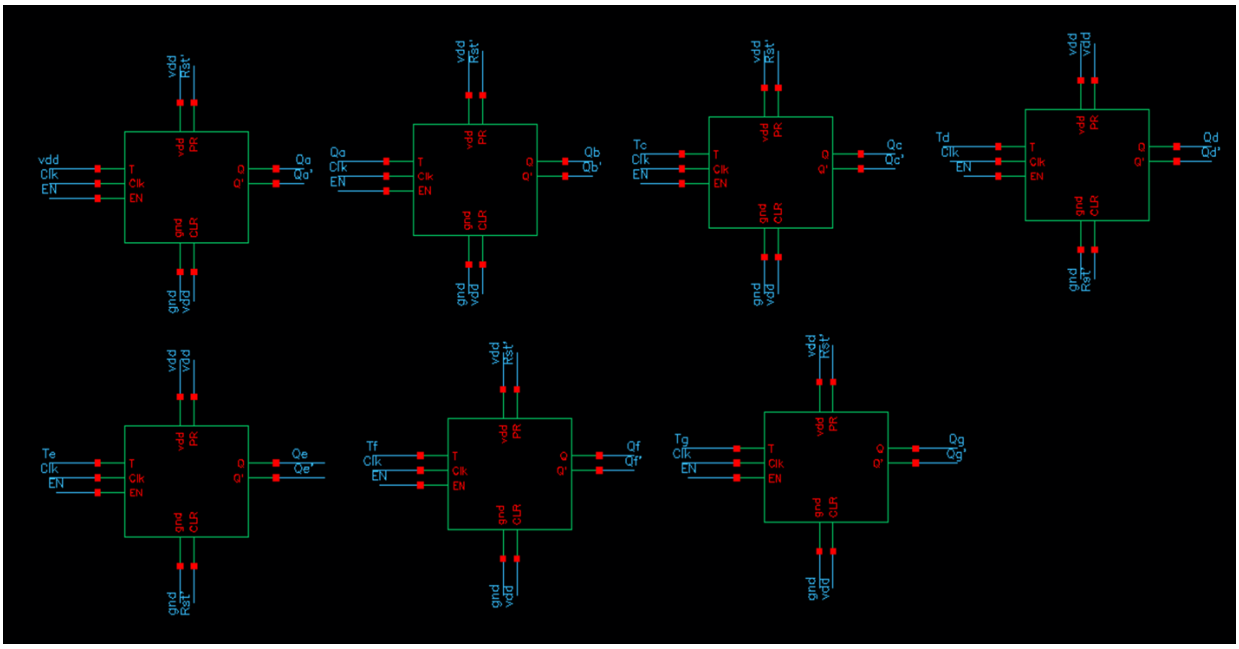

Fig 1.14 T- flip flops Block Design

- This is a T – flip flop block that is connected to different inputs, clock, EN, Preset and Clear signals. This is an actual counter circuit that gives the output response as per our sequence.

2. Multiplexer Logic Block

- The multiplexers logic block is useful for selecting the input for E, F and G flip - flop. The inputs select in such a manner that it allows the counter to follow a proper seqeunce. Ex :- S0 input should be selected , while

counting from 103 to 127 and S1 input should be selected, while counting from the 0 to 127.

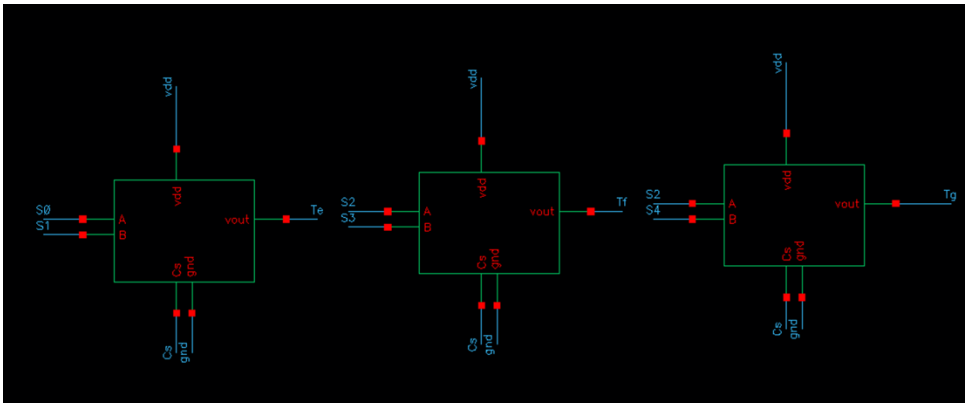

Fig 1.15 Multiplexers Block Design

3. Inputs Logic Block

- The Inputs logic block show the different gates that were used for defining of Flip – Flops.

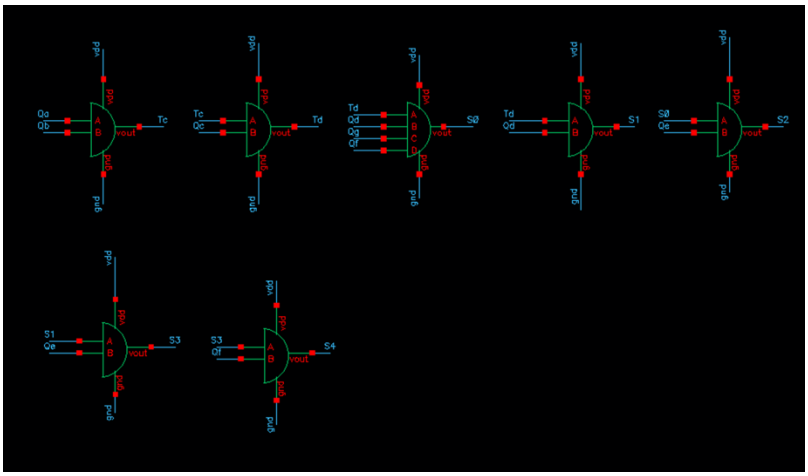

Fig 1.16 Inputs Block Design

4. Reset and Control Signal Logic Block

- This is a twist part or a magic part of the entire counter circuit, that controls the sequence of the Custom design Counter**.**
- **Cs** is the control signal that control the sequence of inputs given to E, F and G Flip – Flop. Initially the value of Cs = 0, that means it will give the input such that the counter starts the count from the number 103 and will end up to 127.
- As soon as the counter reaches to 127, the **Cs = 1** and it will now allow the counter to start from 0 to 127.
- In this schematic, the Cs signal is anded with the **Rst'**, that means whenever reset goes high, the Cs will switch to 0, so that the Counter will start counting from the number 103.
- After then, it is ored with the **S2**. **S2** is the and output of all of the flip – flops outputs, so whenever the counter hits 127, all of the flip – flops output is 1 and eventually **S2** = 1 that will make **Cs** = 1 and it will then count from 0 to 127.

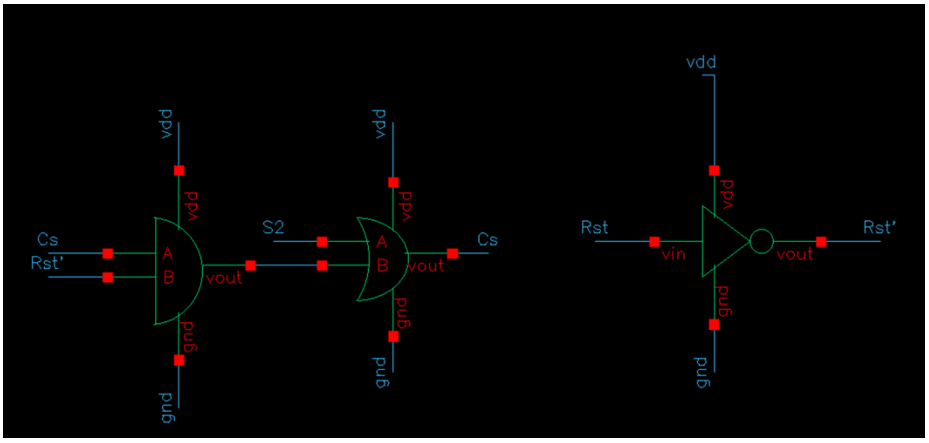

Fig 1.16 Reset and Control Signal Block Design

❖ Output Waveforms :-

- The output waveform I gotten is a different one than what I expected. It was supposed to crystal and clear waveform. However, it starts from the 103 , follows the sequence for the 2 – 3 counts and after that the signals of the flip flop got depleted.
- Upon your advice I have connected the single stage buffer, but it works for a single bit flip flop only.
- I have tried to adjust different (W / L) ratio, tried to connect the two or three stage buffer, but the any of the assumptions didn't works. At last , I stuck with the drive strength of transistor, but the counter that I have designed is a perfect one.

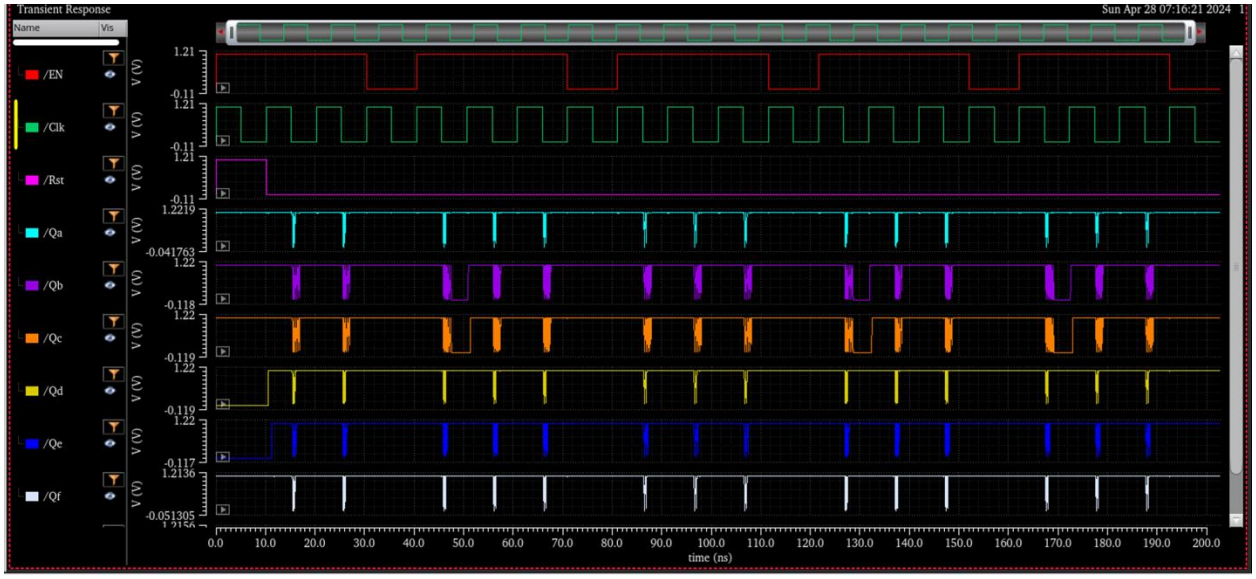

Fig 1.17 Up counter output Waveform

❖ **Delay Estimation and Comparison**

- In this table, Name_x stands for the gate name and X stands for the number of terminals.

Name	Rise Time	Fall Time	Delay
Inverter	47.45ps	69.32ps	58.38ps
AND_2	0.59ns	0.5ns	0.54ns
AND_3	0.649ns	0.51ns	0.579ns
AND_4	0.52ns	0.6ns	0.56ns
OR_2	0.58ns	0.68ns	0.63ns
OR_3	0.46ns	0.47ns	0.46ns
NAND_2	0.24ns	0.238ns	0.239ns
NAND_3	0.55ns	0.50ns	0.525ns
NAND_4	0.45ns	0.292ns	0.371ns
NAND_5	0.59ns	0.37ns	0.48ns
NOR_2	0.25ns	0.31ns	0.28ns
XOR_2	0.68ns	0.427ns	0.55ns
XNOR_2	0.55ns	0.57ns	0.56ns
T_flip flop (Clock at 100 MHz)	0.35ns	0.40ns	0.375ns
T_flip flop (Clock at 1 GHz)	0.25ns	0.23ns	0.24ns
T_flip flop (Clock at 10 Ghz)	0.15ns	0.1ns	0.11ns

Table 1.14 Delay Estimation and Comparison