An optimised 4-bit carry look ahead adder to minimize delay

Kaamya Dasika

2023102034
UG2, ECE

IIIT Hyderabad

Abstract*— In this paper, an optimized 4-bit carry look-ahead adder (CLA) design is proposed to minimize propagation delay in digital circuits. The conventional ripple carry adder is transformed to utilize two-level carry logic, significantly reducing delay while maintaining accurate output at the next clock edge. Key optimizations include the use of CMOS static logic gates, transistor sizing for balanced rise and fall times, and the adoption of edge-triggered flip-flops to align with the timing diagram. The design prioritizes speed by assuming stable inputs within specific timing constraints, offering a trade-off between performance and input variability tolerance. Simulation results from NGSPICE demonstrate reduced delays and improved clock speed, validated through pre-layout and post-layout analyses. Implementation on FPGA further verifies the design's performance, showcasing efficient addition with high-speed operation.

Keywords: Carry Look-Ahead Adder, CMOS Logic, Delay Optimization, FPGA, NGSPICE, Digital Design

I. Introduction

A carry look-ahead adder reduces the propagation delay by introducing more complex hardware. In this design, the ripple carry design is suitably transformed such that the carry logic over fixed groups of bits of the adder is reduced to two-level logic. We shall design a CLA considering that input bits are available before the rising edge of the clock and the output should be computed and present at the next rising edge of the clock. We achieve this using D flip flops and the input and output. To reduce the delay, we optimize the propagate and generate terms.

II. Conventional Structure

By generating carry signals quickly, the CLA enhances the speed of binary addition.

A. Inputs and Outputs

Inputs: Two 4-bit binary numbers A=(A3,A2,A1,A0) and B=(B3,B2,B1,B0), along with an initial carry-in Cin.
Outputs: A 4-bit sum S=(S3,S2,S1,S0) and a final carry-out Cout.

B. Generate and Propagate terms

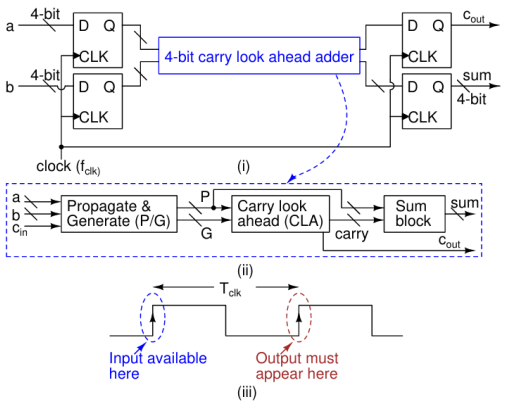
- For each bit position i, Generate and Propagate signals are derived as:
  Gi=Ai XOR Bi Pi=Ai+Bi

These signals are used to calculate carry signals independently of previous bits, allowing carry information to be determined concurrently.

C. Carry Lookahead Logic

- Carry-out signals are generated for each bit position in advance:
  C1=G0+(P0·Cin)
  C2=G1+(P1·C1)=G1+(P1·(G0+P0·Cin))
  C3=G2+(P2·C2)
  Cout=G3+(P3·C3)
  Using the carry signals, each bit of the sum Si is computed as: Si=Pi⊕Ci
- This parallelized calculation of carry signals reduces the delay associated with traditional ripple-carry adders.

III. Proposed Structure



The proposed structure has OR logic for finding Pi terms.

Since the result of XOR and OR differ in the cases when both A and B are 1, and in this case the Ci+1 is independent of Ci and P, we don't have to worry about any mismatch. This reduces the complexity of the design. Since complementary logic gates are inverting in CMOS static logic, it is more efficient to deal with NAND and NOR(universal gates) in CMOS rather than AND and OR. So we do the following changes using DeMorgan's theorem-

$$P_i' = \overline{A_i + B_i}$$

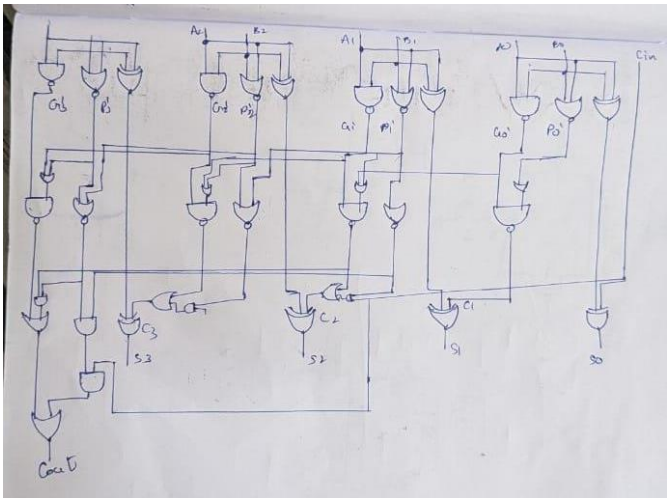
$$G_i' = \overline{A_i \cdot B_i}$$

$$C_1 = \overline{G_0' \cdot (P_0' + \overline{C_0})}$$
$$C_2 = \overline{G_1' \cdot (P_1' + G_0')} + (\overline{P_1' + P_0'} \cdot C_0)$$
$$C_3 = \overline{G_2' \cdot (P_2' + G_1')} + (\overline{P_2' + P_1'} \cdot \overline{G_0'(P_0' + \overline{C_0})})$$
$$C_4 = \overline{P_3' + P_2'} \cdot \overline{P_1' + P_0'} \cdot C_0 + \overline{G_3' \cdot (P_3' + G_2')} + (\overline{P_3' + P_2'} \cdot \overline{G_1'(P_1' + G_0')})$$

Ci	Ai	Bi	Ai XOR Bi	Ai OR Bi	Ai AND Bi	Ci+1	Carry type
0	0	0	0	0	0	0	None
0	0	1	1	1	0	0	None
0	1	0	1	1	0	0	None
0	1	1	0	1	1	1	Generate
1	0	0	0	0	0	0	None
1	0	1	1	1	0	1	Propagate
1	1	0	1	1	0	1	Propagate
1	1	1	0	1	1	1	Generate/Propagate

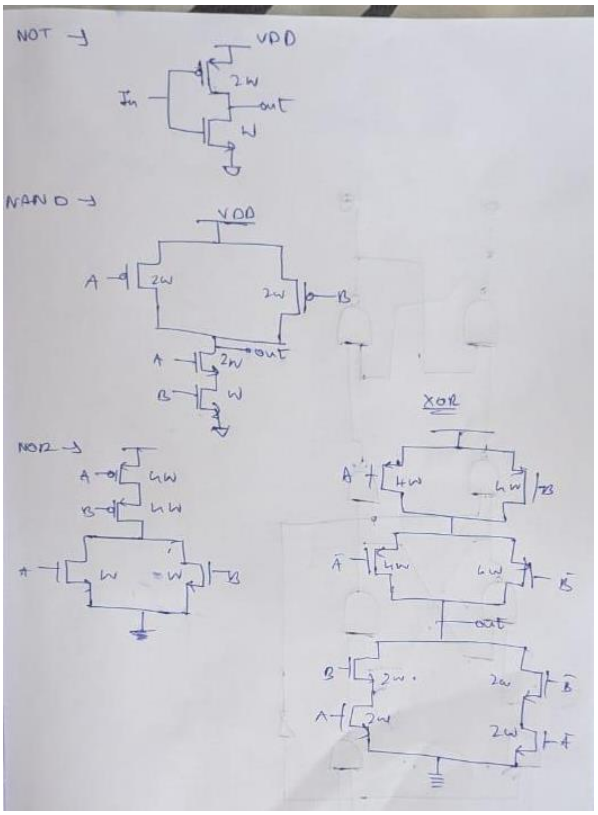
In this design, a negative edge flipflop has been used at the input side and a positive edge flip flop at the output side to get the results according to the timing diagram. So since inuts are available at the +ve edge itselff, we consider them at the immediate -ve edge. Although any changes in the inut in this time span between a concecutive +ve and -ve edge will affect the output, but the design has been made with an assumption that no change occurs, This decreases the delay of the circuit as we give outputs immediately at the upcoming +ve edge. This is the trade off of the implementation- more speed/ less delay with the input snot changing till the -ve edge after they have been provided.

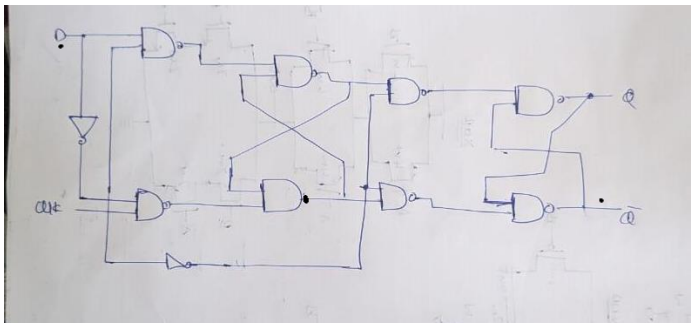
A. Circuit Diagram



IV. Design Details

- The width of the transistors in each of the gates have been decided to match that of the ideal inverter in order to obtain equal rise and fall times and minimize delays and uncertainties. The drive a 20 LAMBDA/10 LAMBDA inverter.
- CMOS static logic is being used for all gates in the circuit, the sizes are as follows-

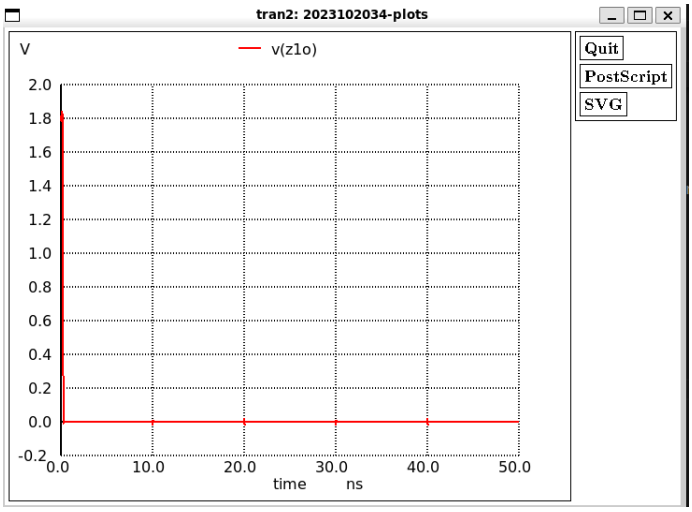


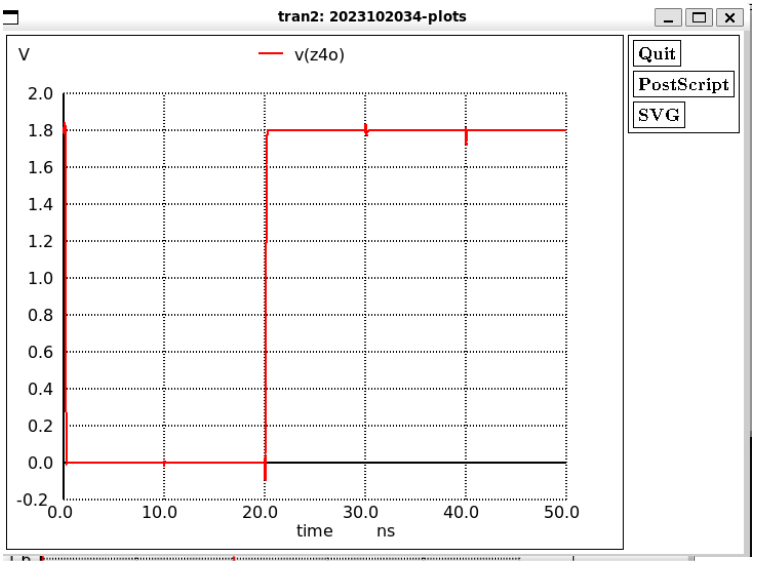


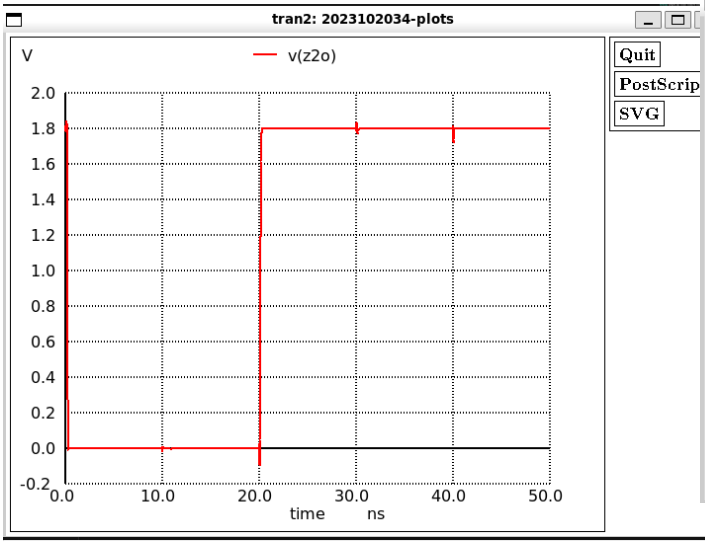
V. NGSPICE SIMULATIONS

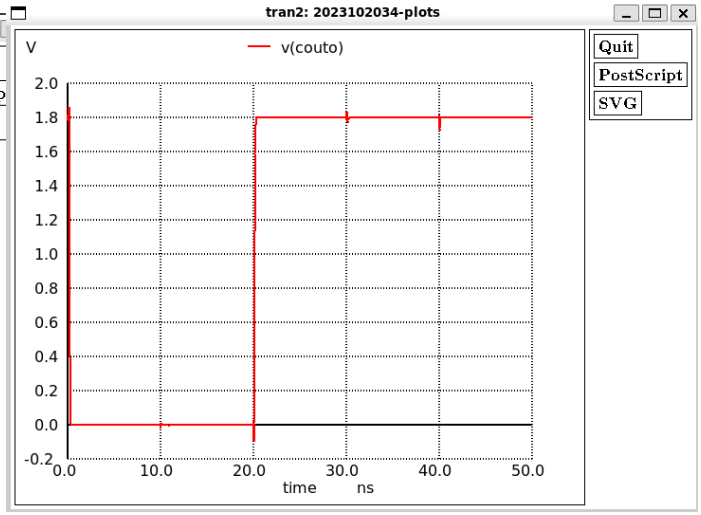
```
.tran 1n 600n
.measure tran tpdr1
+TRIG v(y1) VAL='0.50*SUPPLY' RISE=1 TARG v(p1) VAL='0.50*SUPPLY' RISE=1
.measure tran tpdf1
+TRIG v(y1) VAL='0.50*SUPPLY' FALL=1 TARG v(p1) VAL='0.50*SUPPLY' FALL=1
.measure tran tpd1
+param='(tpdr1+tpdf1)/2' goal=0
```

A. Adder block-

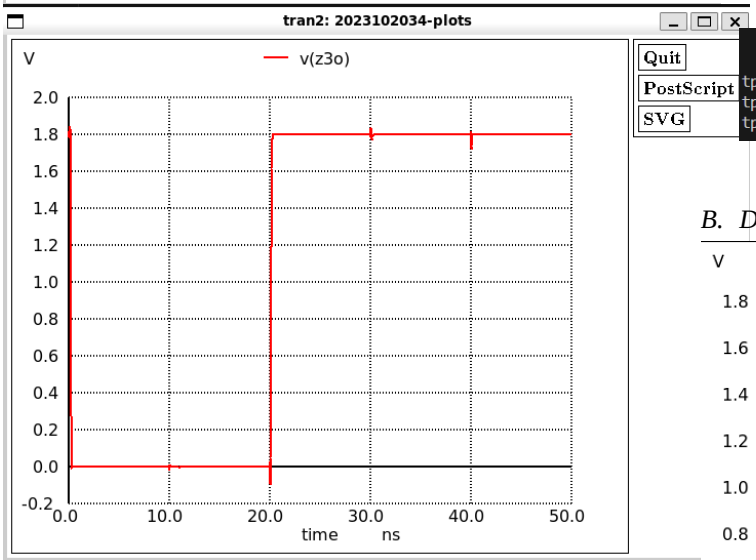






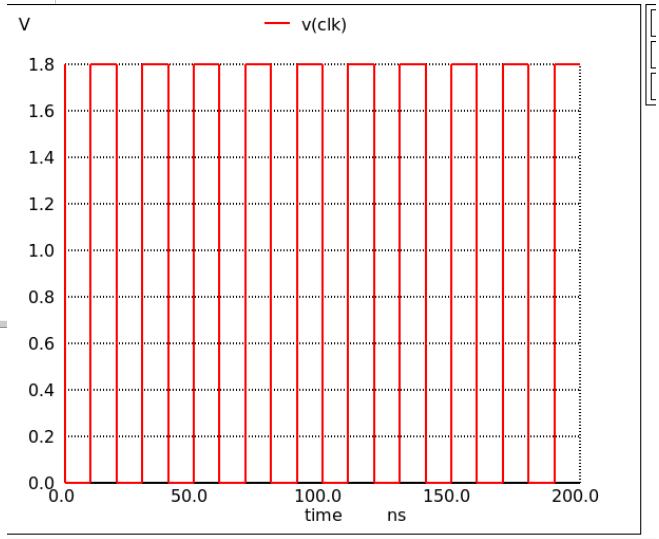


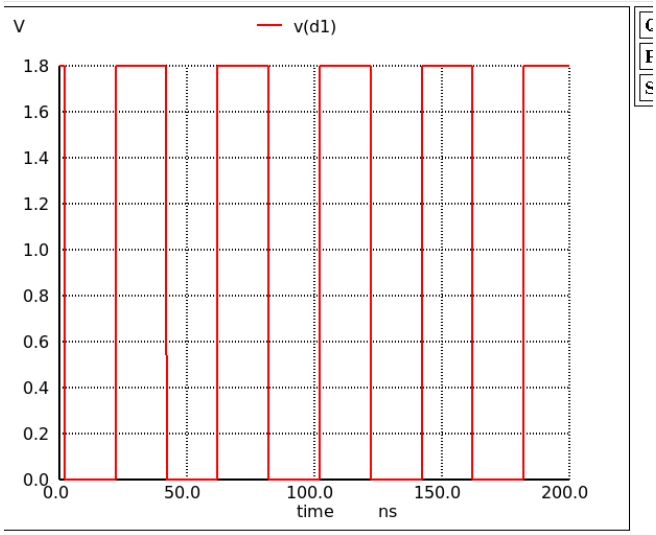
Delay-

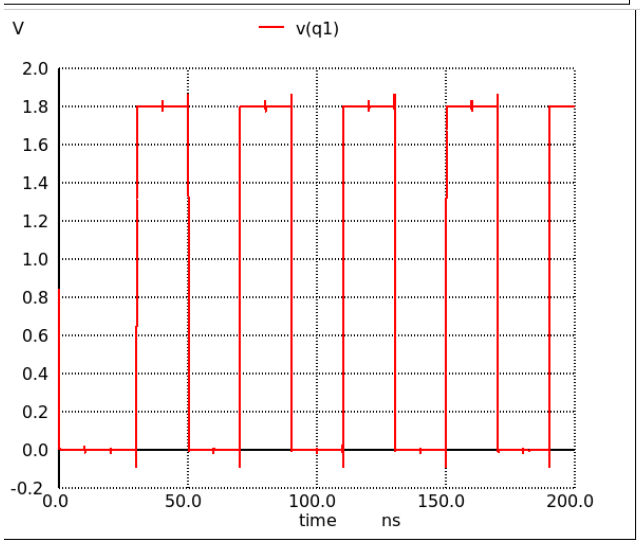



```
Measurements for Transient Analysis

tpdr1                =  3.146814e-10 targ=  2.329681e-09 trig=  2.015000e-09
tpdf1                =  4.278118e-09 targ=  4.283118e-09 trig=  5.000000e-12
tpd1                 =  2.29640e-09
```

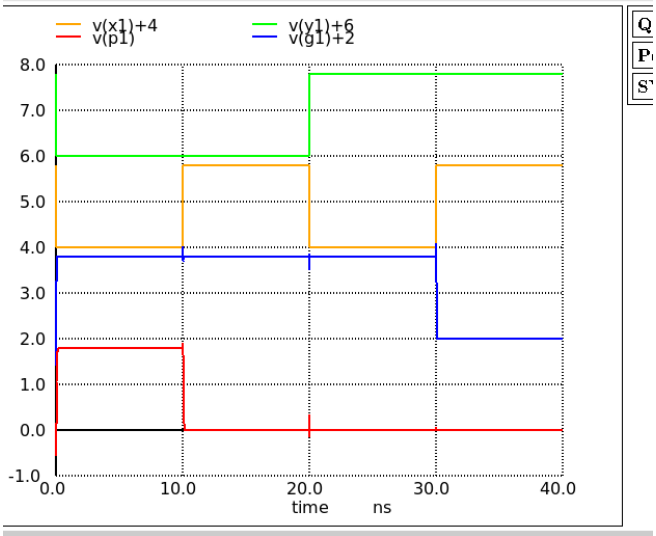
B. D flip flop







C. P, G terms-




```
  Measurements for Transient Analysis

tpdr1              =  -1.968730e-09 targ=  4.626997e-11 trig=  2.015000e-09
tpdf1              =  2.039457e-09 targ=  2.044457e-09 trig=  5.000000e-12
tpd1               =  3.53633e-11
```

VI. D FLIP FLOP DELAYS

A. Setup time-

```
No. of Data Rows : 238

  Measurements for Transient Analysis

setup_time         =  2.500000e-10 targ=  6.500000e-09 trig=  6.250000e-09

ngspice 22 ->
```

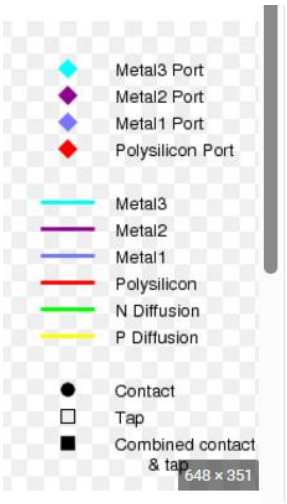
B. Hold time-

```
No. of Data Rows : 238

  Measurements for Transient Analysis

hold_time          =  0.000000e+00 targ=  6.500000e-09 trig=  6.500000e-09

ngspice 41 -> d1.cir
```

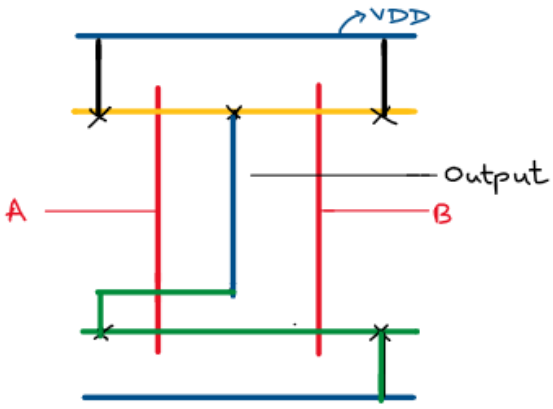
C. TPCQ delay(contamination delay)

```
ngspice 43
xff.x9.1                              0.0384786
vd#branch                                     0
vclk1#branch                                  0
vdd#branch                        -1.23538e-10

No. of Data Rows : 238

  Measurements for Transient Analysis

clktq_time         =  2.540509e-10 targ=  1.675405e-08 trig=  1.650000e-08

ngspice 1 ->
```

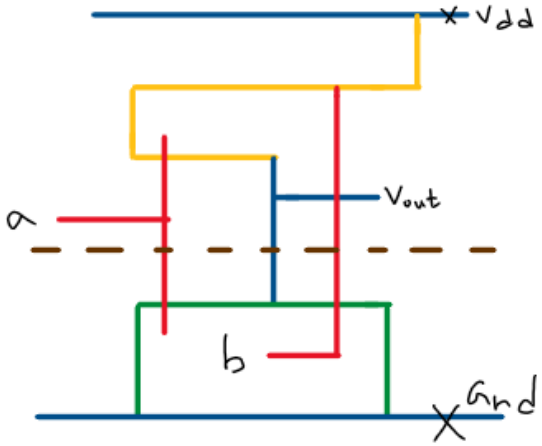
VII. STICK DIAGRAMS



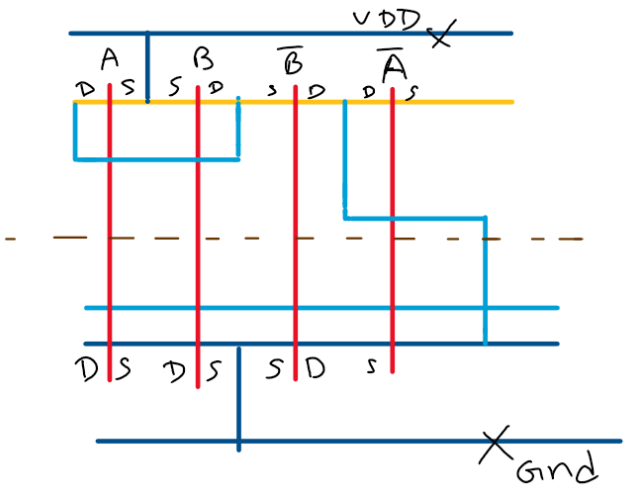
A. NAND



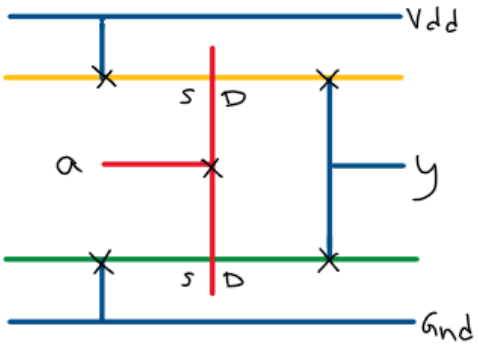
B. NOR



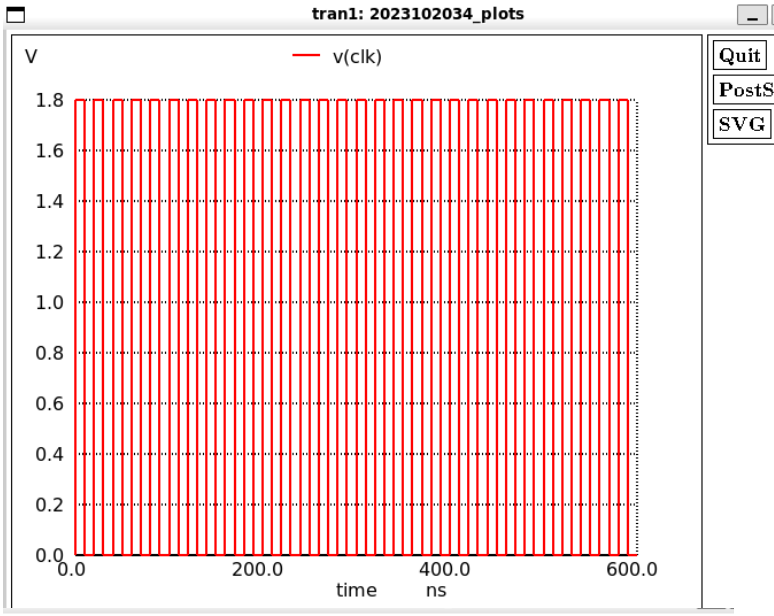
C. XOR

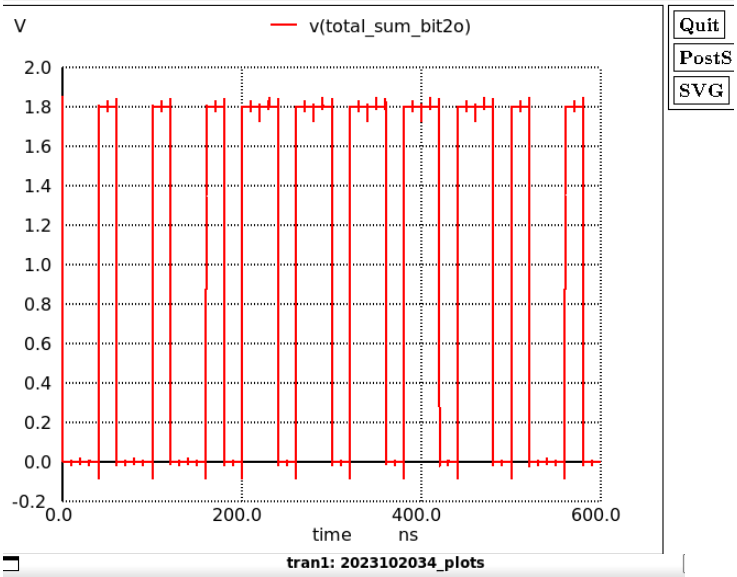


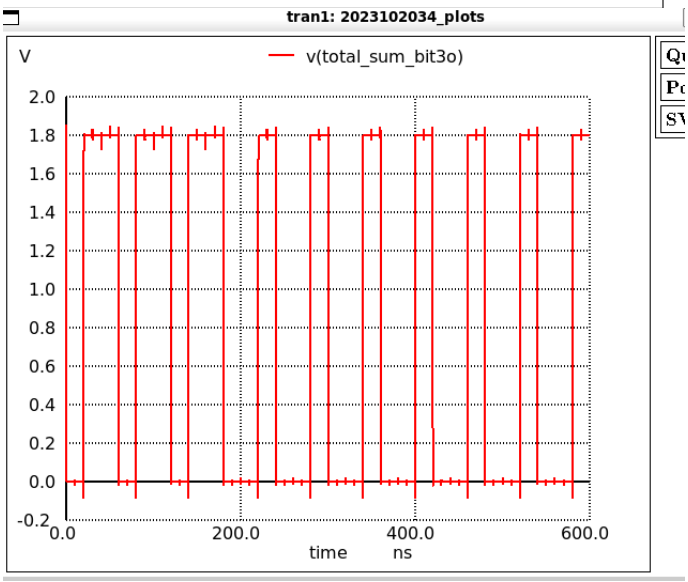
D. INVERTER

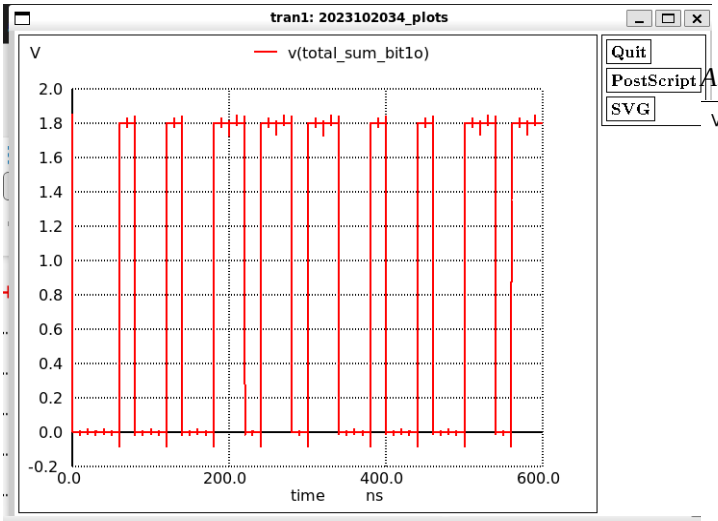


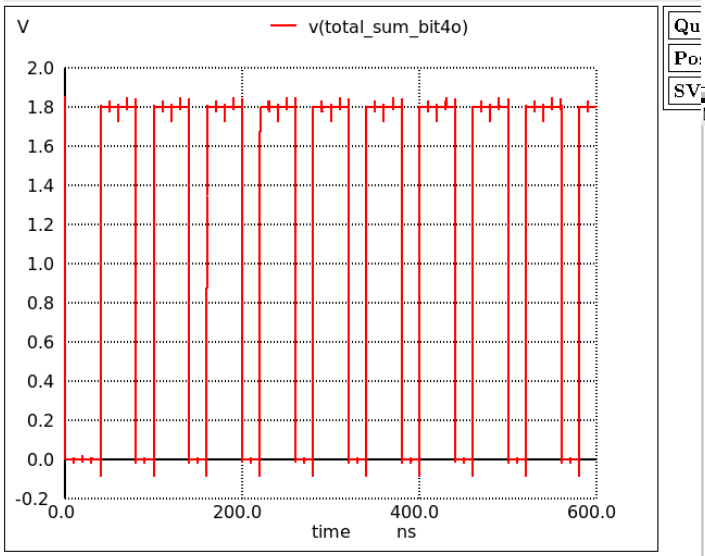
VIII. FINAL CIRCUIT PRE LAYOUT





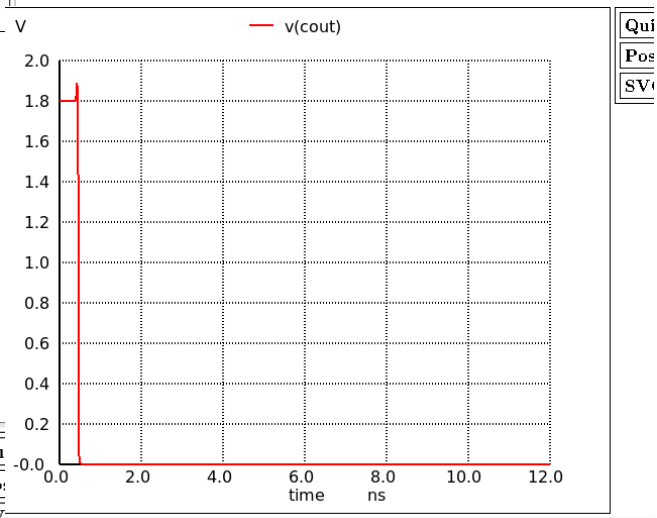


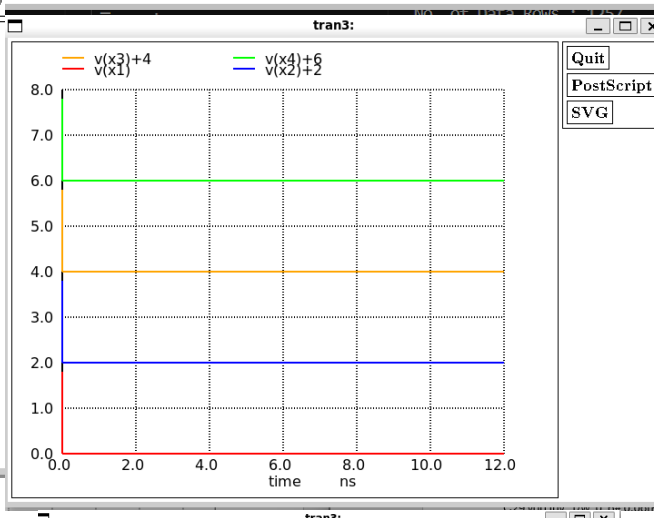


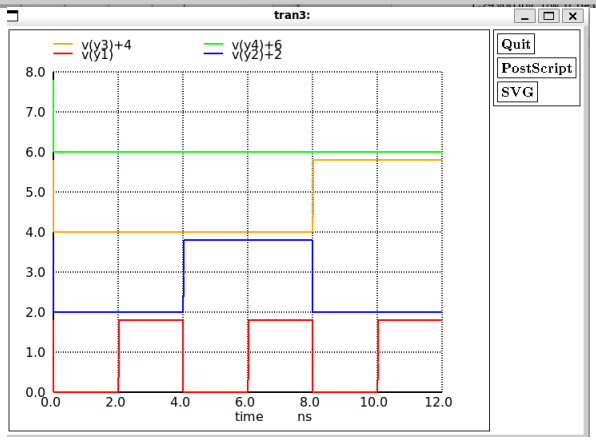


IX. POST LAYOUT

A. *Adder block-*







A. *Delay of full circuit-*

```
No. of Data Rows : 6973

  Measurements for Transient Analysis

tpdr1                =  1.981090e-07 targ=  2.001240e-07 trig=  2.015000e-09
tpdf1                =  1.769200e-10 targ=  1.819200e-10 trig=  5.000000e-12
tpd1                 =  9.91430e-08

.hardcopy: no such command available in ngspice
```

B. *Worst case delay of adder-*

```
tpdr1                =  3.146814e-10 targ=  2.329681e-09 trig=  2.015000e-09
tpdf1                =  4.278118e-09 targ=  4.283118e-09 trig=  5.000000e-12
tpd1                 =  2.29640e-09
```

C. *Maximum Clock speed-*

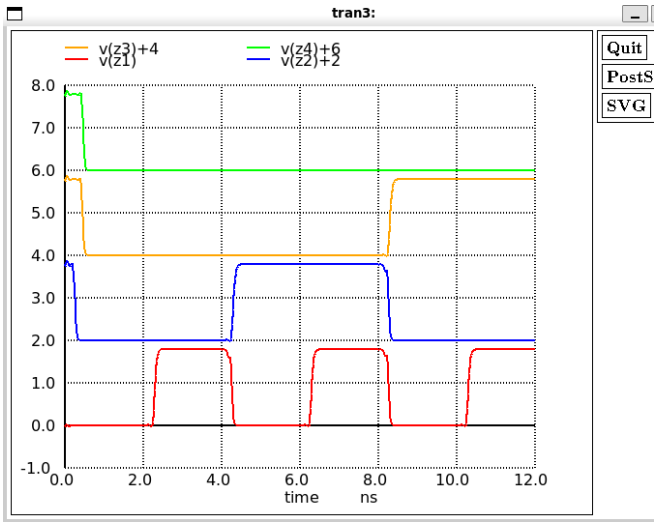
Tclk min= 2.7 ns

$$T_{clk} >= t_{PCQ\ max} + t_{PD\ max} + t_{setup}$$

I can see that the value matches with this and there is no setup time violation as $t_{PCQ\ max} = 0.25$ ns

$t_{PD\ max} = 2.2$ ns

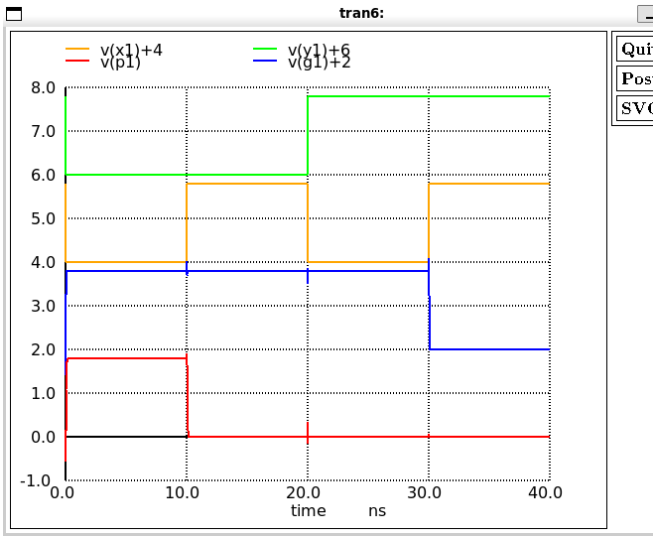
$t_{setup} = 0.25$ ns



Delay-

```
tpdr1              =  2.814415e-10 targ=  2.296442e-09 trig=  2.015000e-09
tpdf1              =  4.253997e-09 targ=  4.258997e-09 trig=  5.000000e-12
tpd1               =  2.26772e-09
```

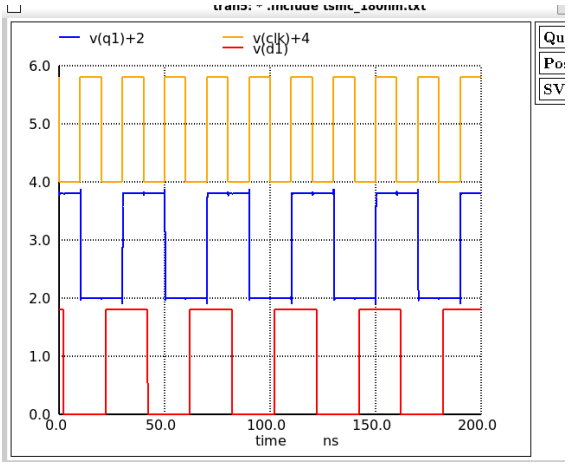
B. P, G block



Delay-

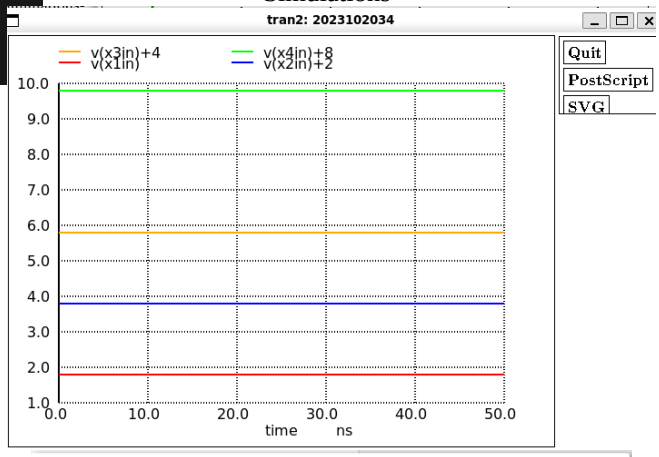
```
  Measurements for Transient Analysis

tpdr1              =  -1.964972e-09 targ=  5.002753e-11 trig=  2.015000e-09
tpdf1              =  2.043485e-09 targ=  2.048485e-09 trig=  5.000000e-12
tpd1               =  3.92561e-11
```

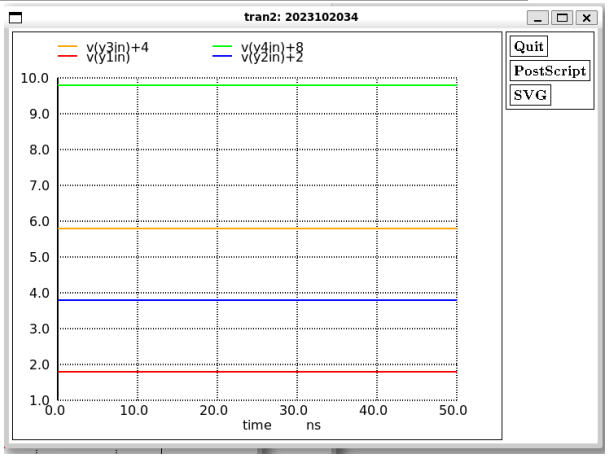
C. D flip flop

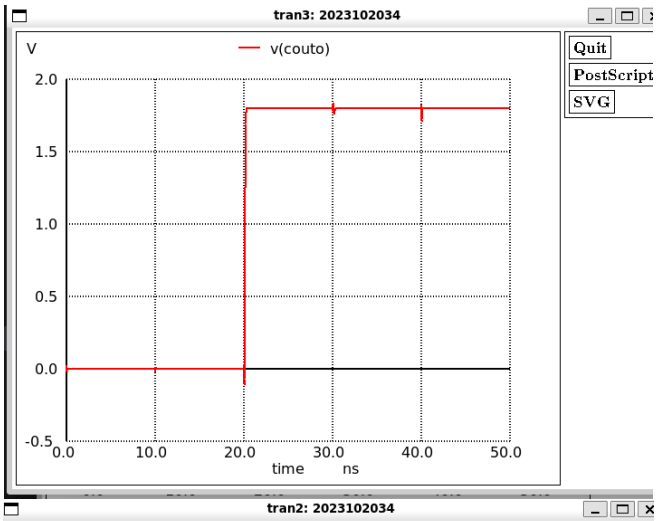


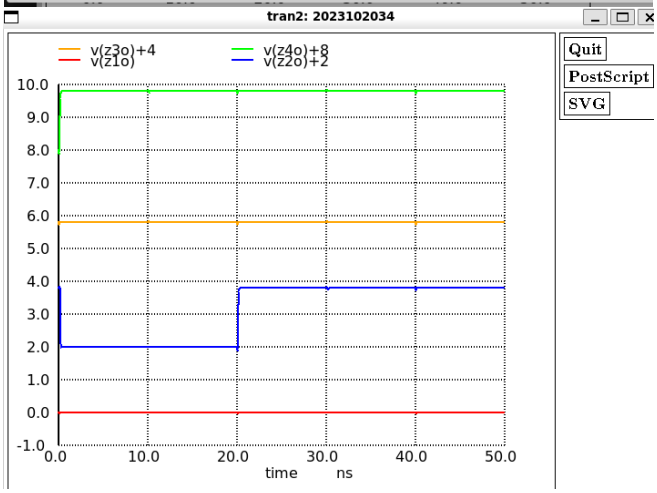
X. FINAL CIRCUIT POST LAYOUT

Simulations-









Delay-

```
  Measurements for Transient Analysis

tpdr1              =  1.980999e-07 targ=  2.001149e-07 trig=  2.015000e-09
tpdf1              =  2.201641e-07 targ=  2.201691e-07 trig=  5.000000e-12
tpd1               =  2.09132e-07

.hardcopy: no such command available in ngspice
v(y1in)+4*input_offset: no such command available in ngspice
v(y4in)+7*input_offset: no such command available in ngspice
```

XI. D FLIP FLOP POST LAYOUT DELAY

Delay-

$T_{clk} >= t_{PCQ\ max} + t_{PD\ max} + t_{setup}$

$t_{PCQ\ max}$ = 0.38 ns
$t_{PD\ max}$ = 2.2 ns
t_{setup} = 0.25 ns
thus Tclk min is 2.83 ns

A. *Hold time-*

```
No. of Data Rows : 238

  Measurements for Transient Analysis

hold_time          =  2.000950e-21 targ=  6.500000e-09 trig=  6.500000e-09

ngspice 42 ->
```

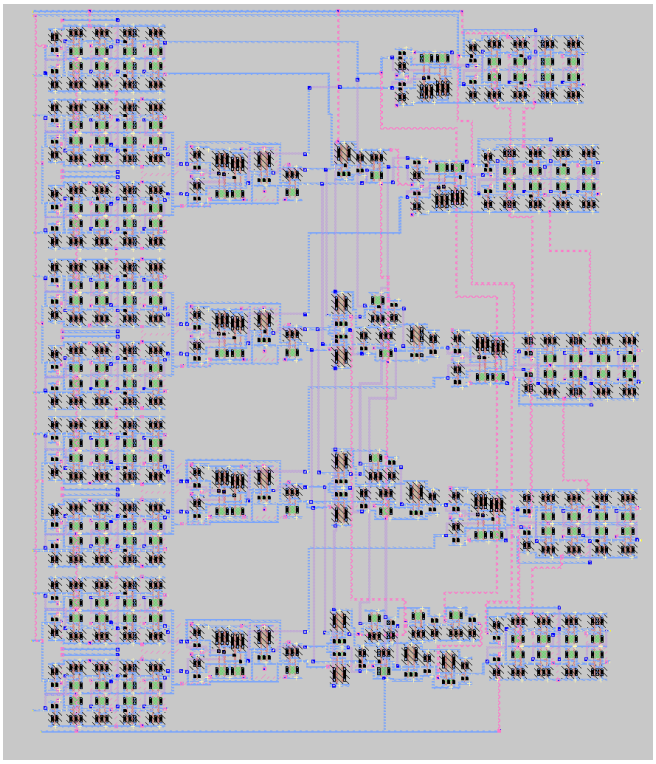
B. *Tpcq-*

```
nand_1/a                        0.701268
vd#branch                   -1.65886e-07
vclk1#branch                -1.94991e-07
vdd#branch                  -0.000245982

No. of Data Rows : 238

  Measurements for Transient Analysis

clktq_time          =  3.844780e-10 targ=  6.884478e-09 trig=  6.500000e-09

ngspice 1 ->
```

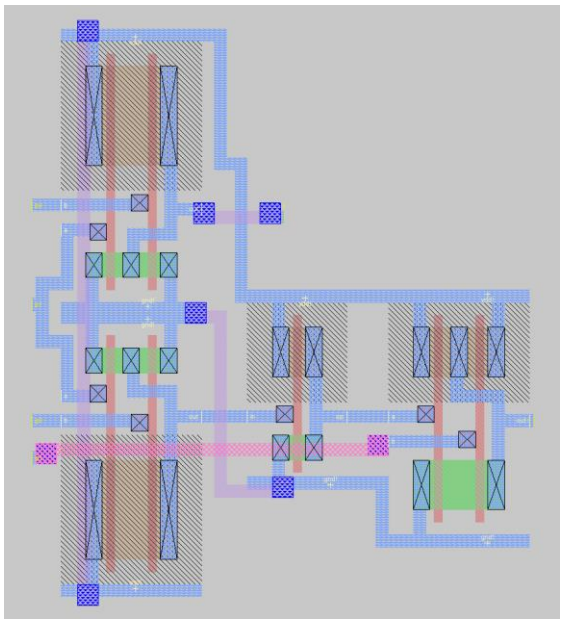
XII. MAGIC

A. *Layouts-*

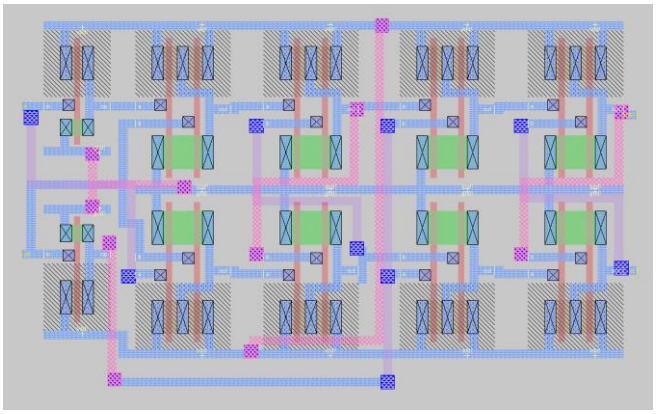
Full circuit-

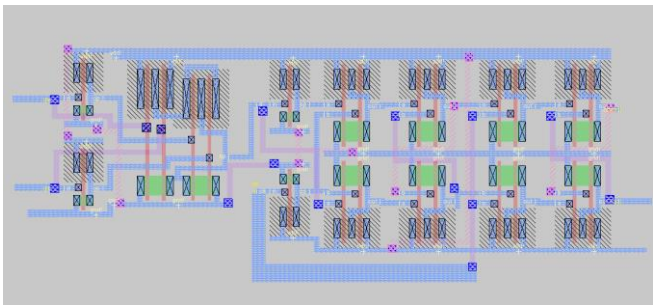


Adder-

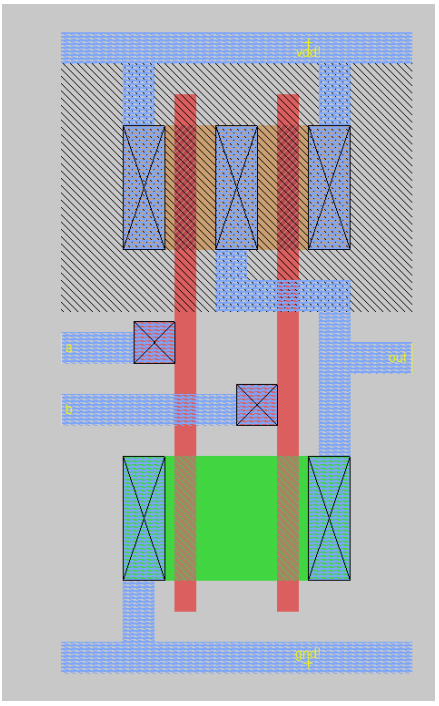


Flip flop-

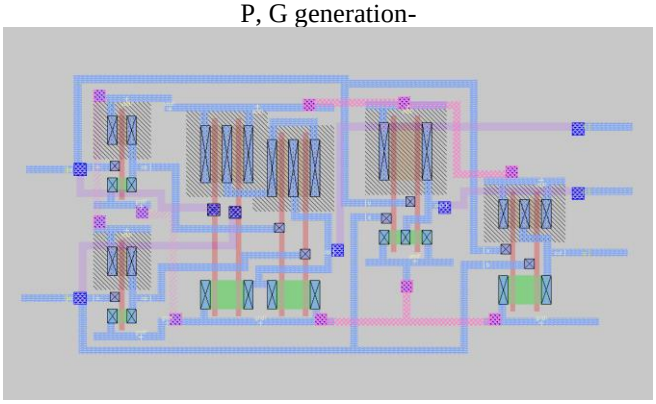




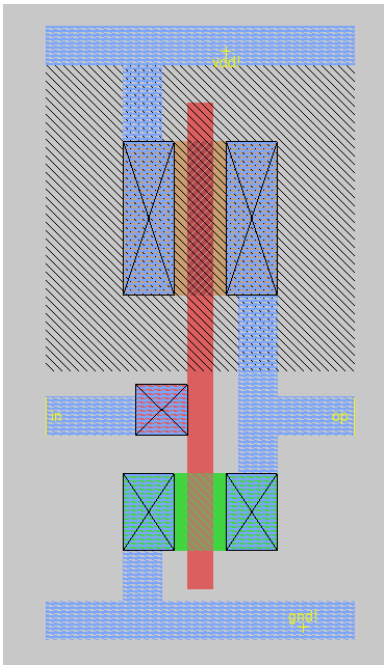
Nand-



P, G generation-

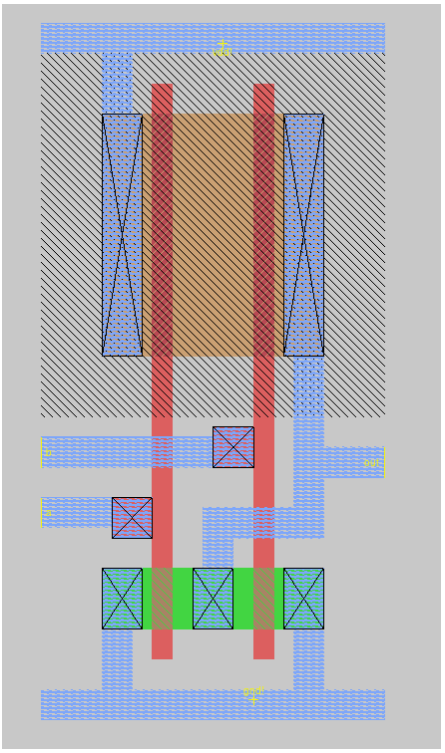


Inverter-

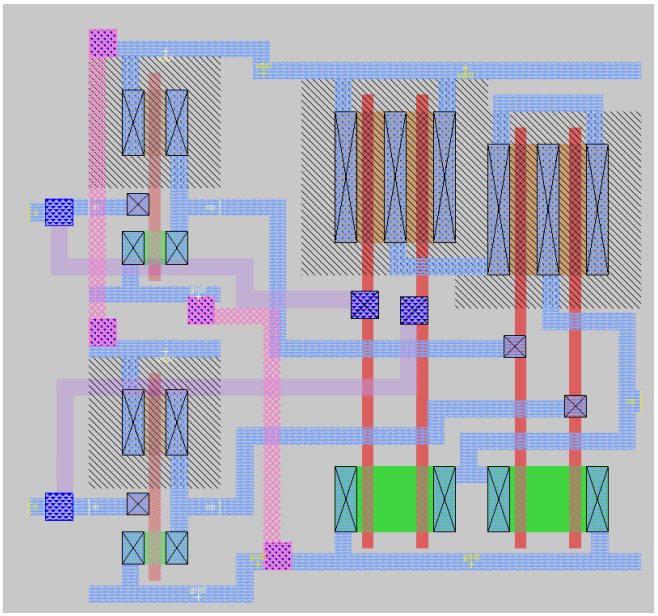


Sum-

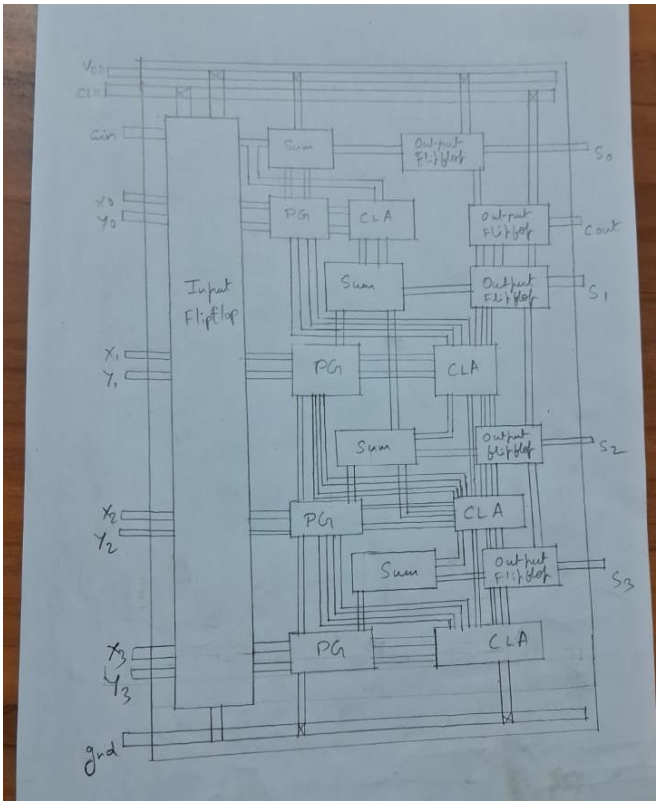
NOR-



XOR-



B. Floor plan of layout-



C. Horizontal and vertical pitches

```
Root cell box:
          width x height  (   llx,  lly  ), (   urx,  ury  )  area (units^2)

microns:   97.92 x 115.56  ( -1.80,  34.83), ( 96.12,  150.39)  11315.64
lambda:     1088 x 1284    (   -20,  387  ), (  1068,  1671 )  1396992
%
```

D. Comparison-

Parameters	Pre-layout	Post-layout
CLA	2.296ns	2.26ns
D flip flop Tpcq	0.25ns	0.38ns
Full circuit	9.91 x 10^-8	2.09 x 10^-7s
Hold time	0	2 x 10^-21s

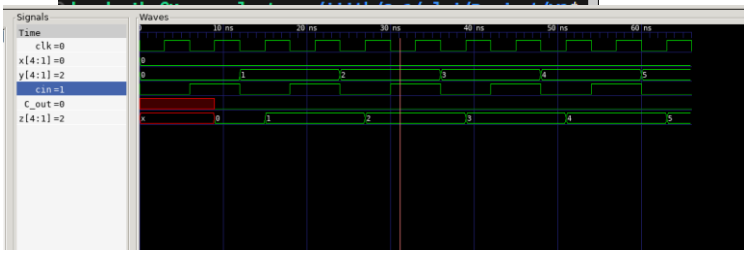
Maximum clock frequency- 1 / minimum Tclk

Pre layout-
Maximum clock speed / freq = 1/ 2.7ns = 3.7 x 10^8 Hz
Post layout-
Max clk freq = 1/ 2.8ns = 3.58 x 10^8 Hz

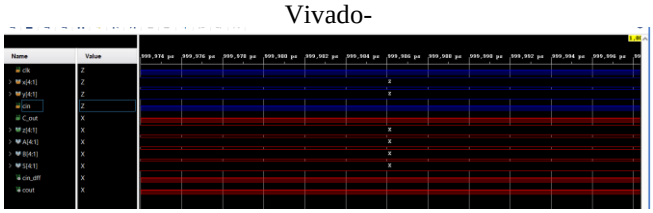
XIII. VERILOG

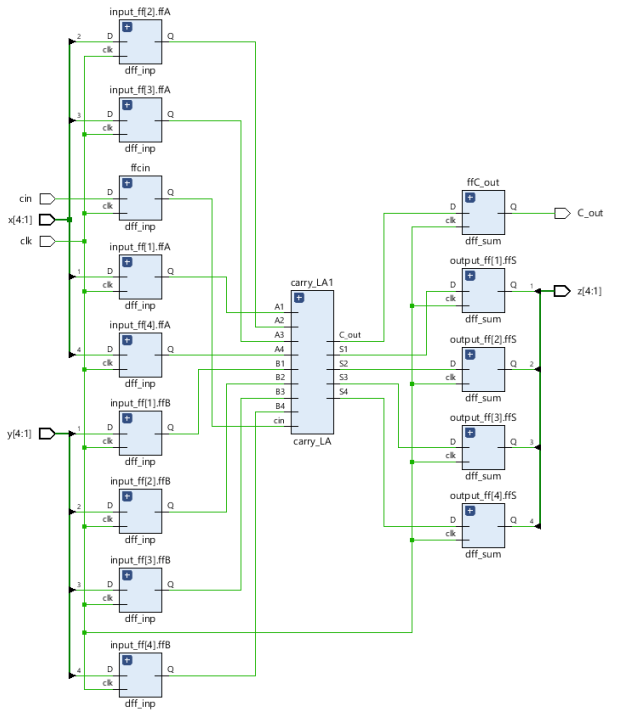
We can observe the results from the gtk waveforms of the structural description of the circuit in verilog. When the inputs are available at the positive edge of the clock, the sum appears at the next positive edge which is visible below. Any change in the input now will be affecting the next clock.

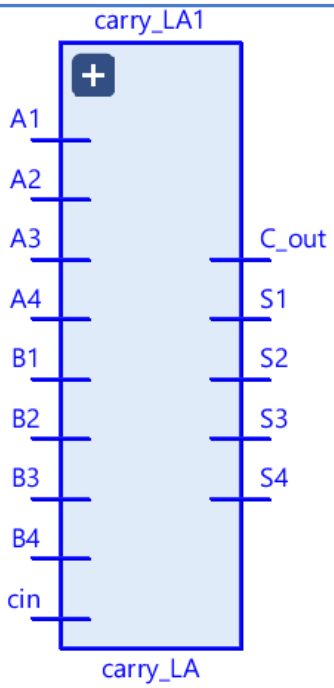
```
kaadasika@Kaamya-laptop:~/iiith/2-1/vlsi/Project/V1$ 1
kaadasika@Kaamya-laptop:~/iiith/2-1/vlsi/Project/V1$ v
VCD info: dumpfile 2023102034_gtkw.vcd opened for outp
clk=0 x=0000 y=0000 cin=0 C_out=x z=xxxx
clk=1 x=0000 y=0000 cin=0 C_out=x z=xxxx
clk=0 x=0000 y=0000 cin=1 C_out=x z=xxxx
clk=1 x=0000 y=0000 cin=1 C_out=0 z=0000
clk=0 x=0000 y=0001 cin=0 C_out=0 z=0000
clk=1 x=0000 y=0001 cin=0 C_out=0 z=0001
clk=0 x=0000 y=0001 cin=1 C_out=0 z=0001
clk=1 x=0000 y=0001 cin=1 C_out=0 z=0001
clk=0 x=0000 y=0010 cin=0 C_out=0 z=0001
clk=1 x=0000 y=0010 cin=0 C_out=0 z=0010
clk=0 x=0000 y=0010 cin=1 C_out=0 z=0010
clk=1 x=0000 y=0010 cin=1 C_out=0 z=0010
clk=0 x=0000 y=0011 cin=0 C_out=0 z=0010
clk=1 x=0000 y=0011 cin=0 C_out=0 z=0011
clk=0 x=0000 y=0011 cin=1 C_out=0 z=0011
clk=1 x=0000 y=0011 cin=1 C_out=0 z=0011
clk=0 x=0000 y=0100 cin=0 C_out=0 z=0011
clk=1 x=0000 y=0100 cin=0 C_out=0 z=0100
clk=0 x=0000 y=0100 cin=1 C_out=0 z=0100
clk=1 x=0000 y=0100 cin=1 C_out=0 z=0100
clk=0 x=0000 y=0101 cin=0 C_out=0 z=0100
clk=1 x=0000 y=0101 cin=0 C_out=0 z=0101
clk=0 x=0000 y=0101 cin=0 C_out=0 z=0101
```



Vivado-

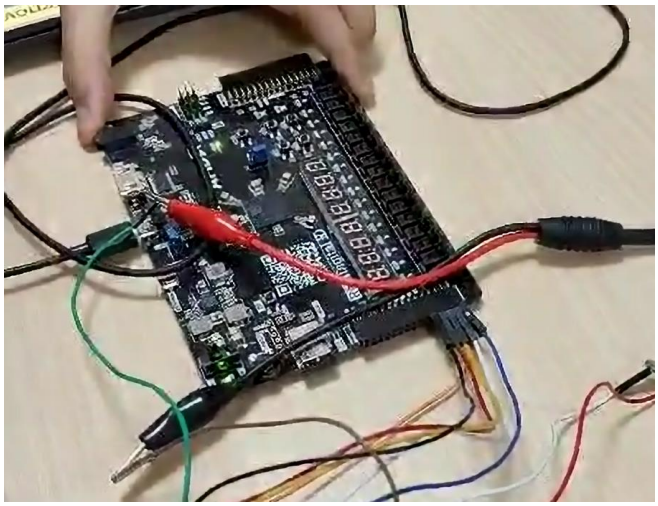






XIV.FPGA

The following drive link displays the working of FPGA and the oscilloscopes for the circuit. We use 3 oscilloscopes for 5 output bits. On giving an input combination, we observe the change in levels(VDD or GND) or the LED on FPGA for the next positive edge of clock.



https://drive.google.com/drive/folders/18tlj9y4bYujJHjAKJuKM1SPlawF1IIGF?usp=sharing

ACKNOWLEDGMENT

I thank the professor and the Tas for constant support throughout the project with claryfying doubts timely.

REFERENCES

Carry Look-Ahead Adder - GeeksforGeeks

A_Proposed_Design_of_Conventional_4-Bit_Carry_Look-Ahead_Adder_Improving_Performance[1].pdf

High_Speed_and_Ultra_Low_Power_Design_of_Carry-Out_Bit_of_4-Bit_Carry_Look-Ahead_Adder[1].pdf