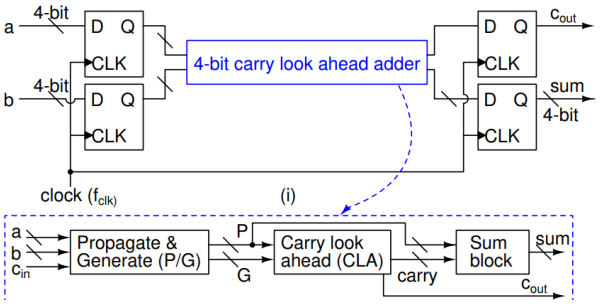
4-BIT CARRY LOOK AHEAD ADDER

Dataarnoor Singh Oberoi

IIIT Hyderabad

1. Introduction

This project involves designing a 4-bit **Carry Look-Ahead Adder (CLA)**, an advanced digital circuit for performing fast binary addition. The CLA adder is based on the concept of precomputing the carry signals for each bit, reducing the delay associated with ripple carry adders.



In Fig. 1 part (ii), we need to create the three blocks. First we generate the p_i and g_i (Propagate and Generate) for CLA block.

$$p_i = a_i \oplus b_i$$

$$g_i = a_i \cdot b_i$$

Then we calculate all carries in CLA block by following logic,

$$c_{i+1} = p_i \cdot c_i + g_i$$

From this, we can calculate all carries $c_i's$ from the propagates and generates ($p_i's$ and $g_i's$).

$$c_1 = (p_0 \cdot c_0) + g_0$$

$$c_2 = (p_1 \cdot p_0 \cdot c_0) + (p_1 \cdot g_0) + g_1$$

$$c_3 = (p_2 \cdot p_1 \cdot p_0 \cdot c_0) + (p_2 \cdot p_1 \cdot g_0) + (p_2 \cdot g_1) + g_2$$

$$c_4 = (p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_0) + (p_3 \cdot p_2 \cdot p_1 \cdot g_0) + (p_3 \cdot p_2 \cdot g_1) + (p_3 \cdot g_2) + g_3$$

Finally we can get the final sum by following logic,

$$S_i = p_i \oplus c_i$$

2. CIRCUIT DIAGRAM AND STRUCTURE

Now we will implement the circuit of the CLA adder using the equations of pi,gi,ci . We will get the carry lookahead generator and then using it we will get the full circuit of the CLA adder as shown.

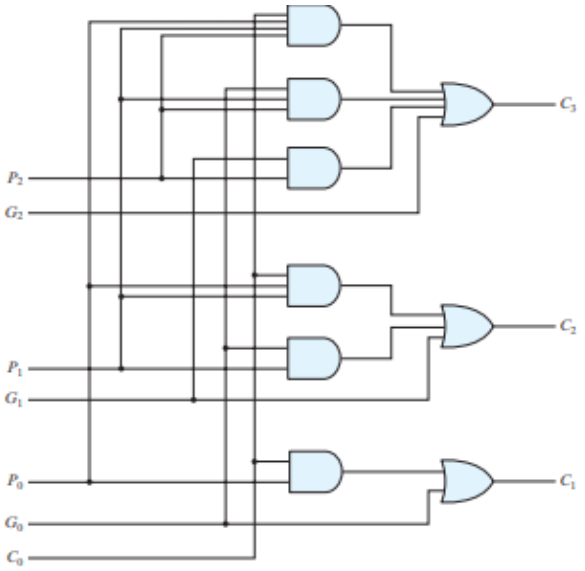


FIGURE 4.11
Logic diagram of carry lookahead generator

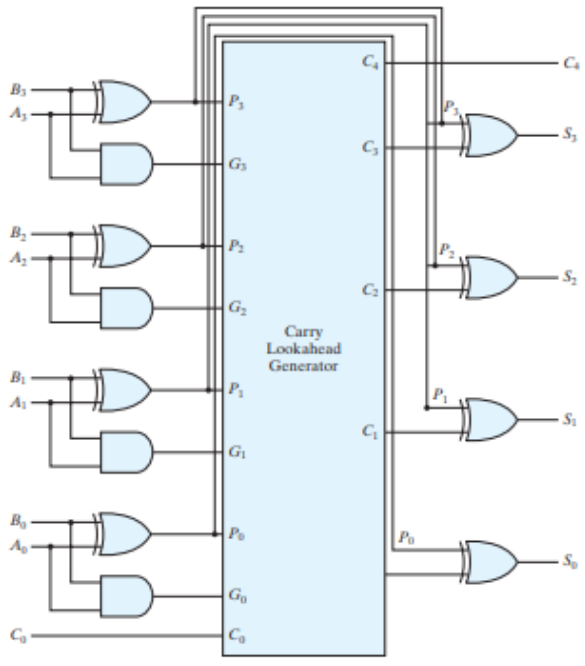
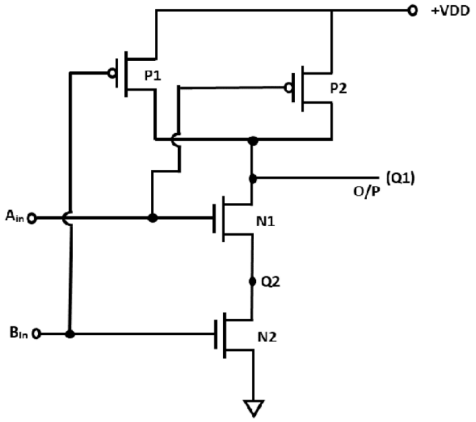
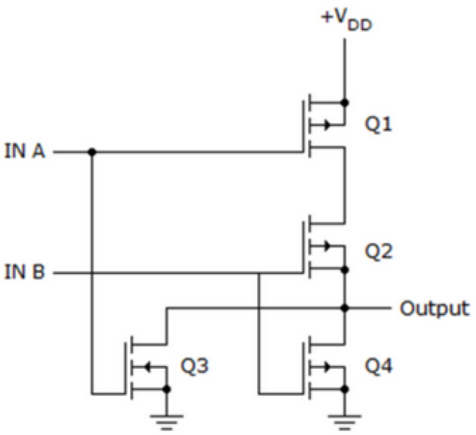


FIGURE 4.12
Four-bit adder with carry lookahead

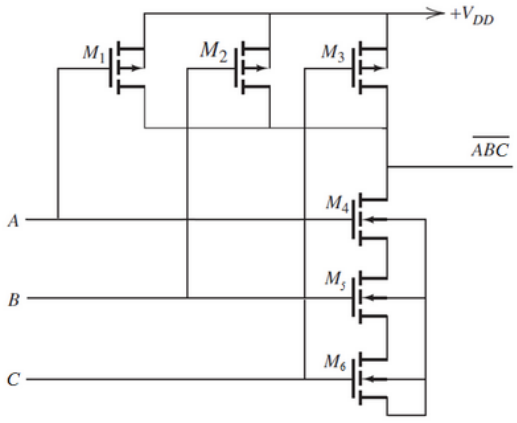
2 INPUT NAND GATE



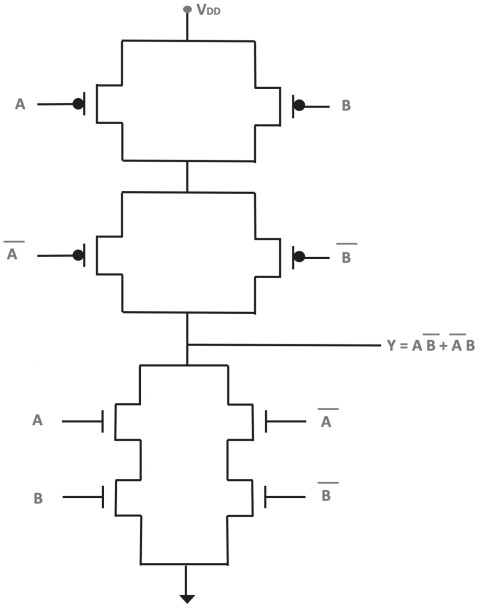
2 INPUT NOT GATE



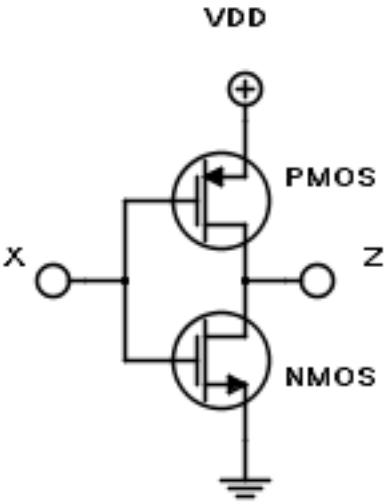
3 INPUT NAND GATE



2 INPUT XOR GATE

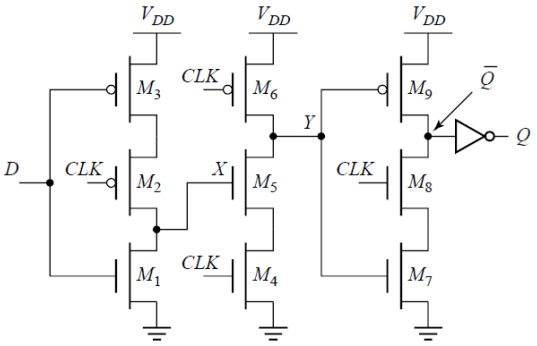


INVERTER



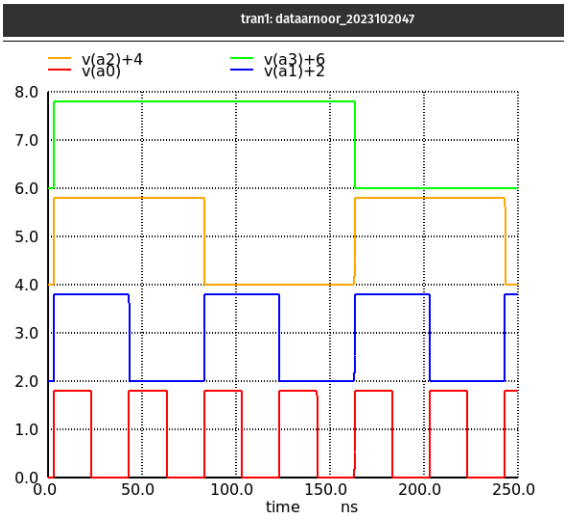
CMOS static logic has been used for implementing all the logic gates and the sizing is PMOS and NMOS width 20λ and 10λ. Where λ=0.09u.

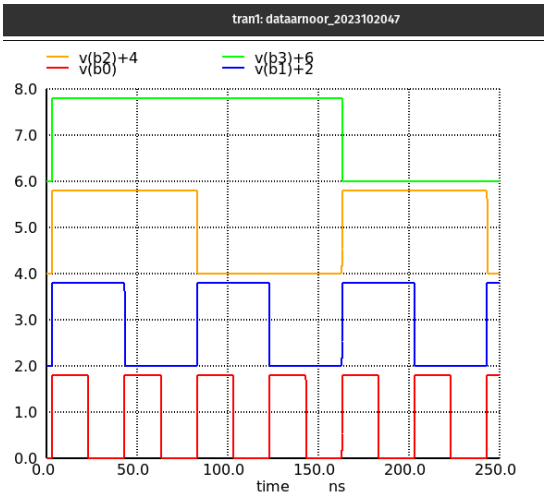
FLIP FLOP (TSPC LOGIC)

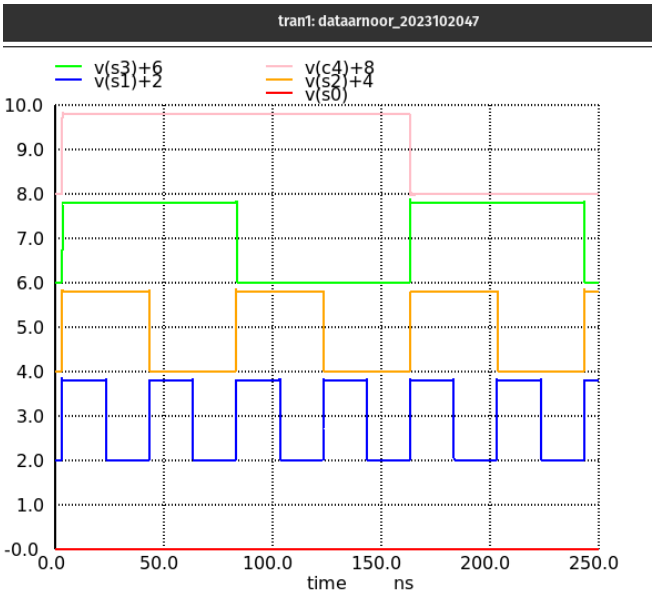


3. SIMULATIONS

CLA ADDER BLOCK:

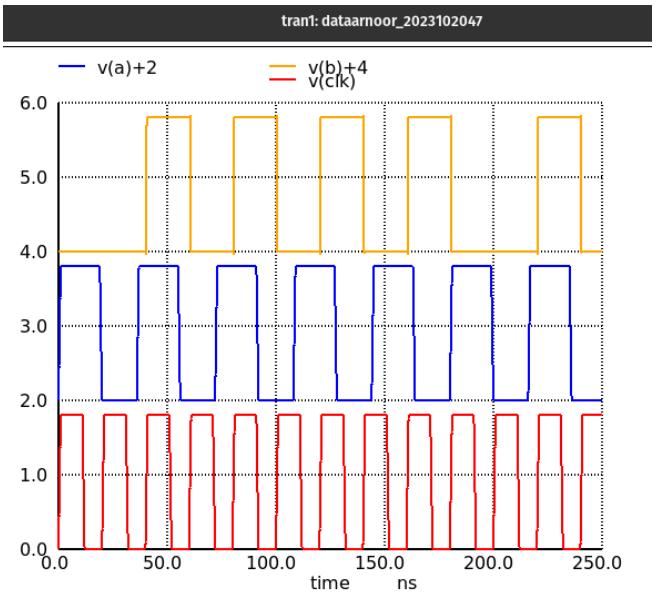






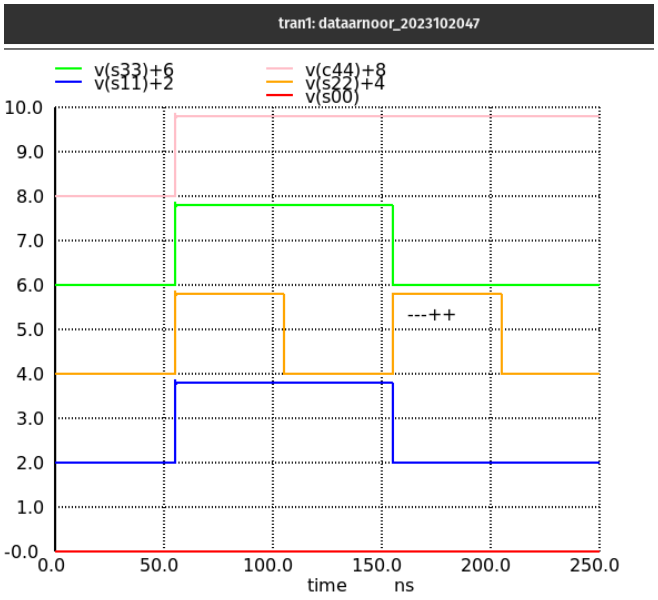
In these simulations a0,a1,a2,a3, and b0,b1,b2,b3 are this inputs bits (given as a pulse signal) and s0,s1,s2,s3,are output sum bits and c4 is the final carry of the cla adder block.

FLIP-FLOP:

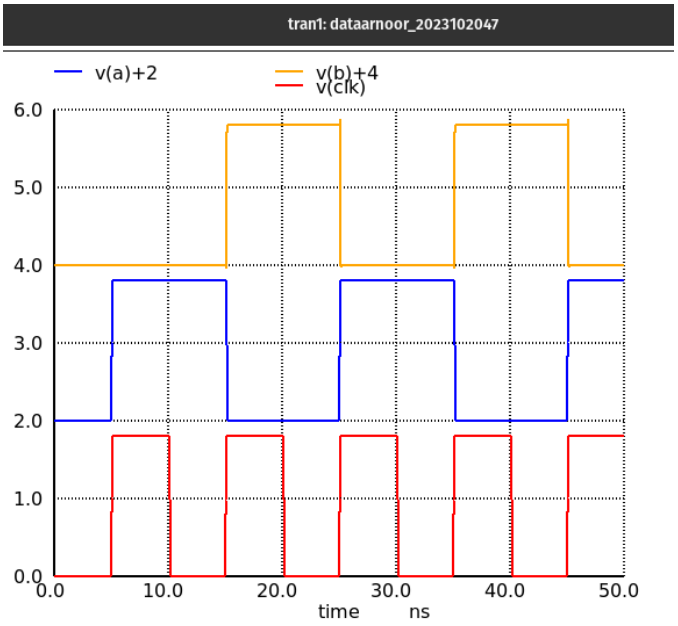


In the flip flop module's simulation, a is the input (D value) and b is the output (Q value) and both are given as pulse signals and clk signal is also a pulse.

COMPLETE CLA ADDER:



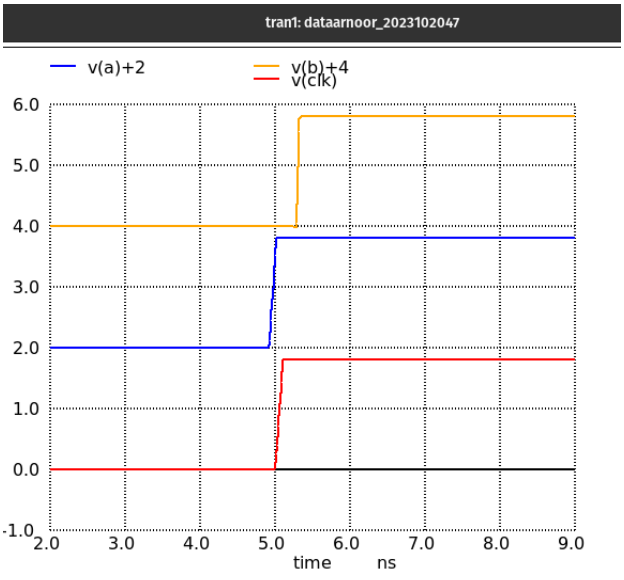
In this we have the same inputs that we had in cla adder block and a clock signal.



We will see that hold time is coming 0 ns for TSPC flip flop

4. SETUP, HOLD AND CLOCK TO Q DELAY:

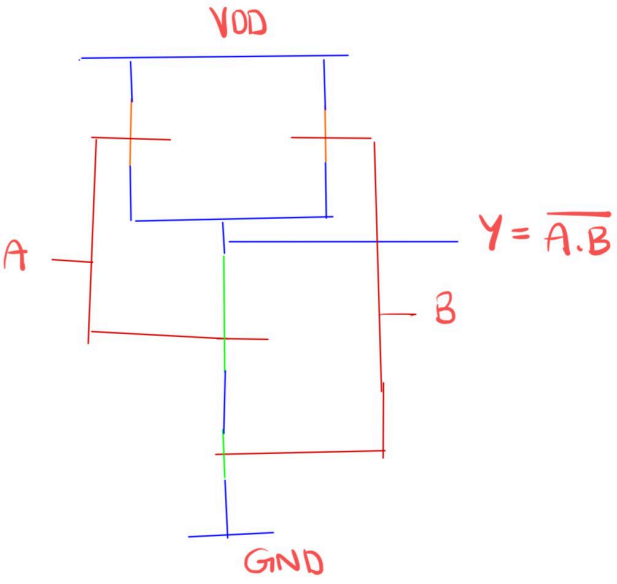
```
tpcq_min          =  6.281086e-11

tpcq_max          =  8.583450e-11
```



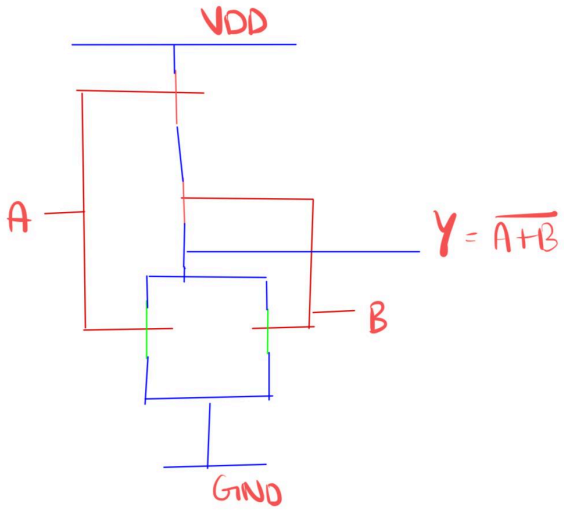
The setup time for this flip-flop comes near about 0.085ns by adjusting the values accordingly.

5. STICK DIAGRAMS

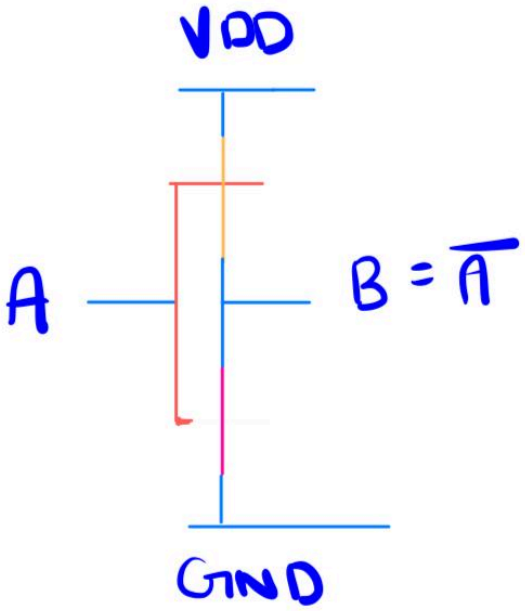
2 INPUT NAND GATE:



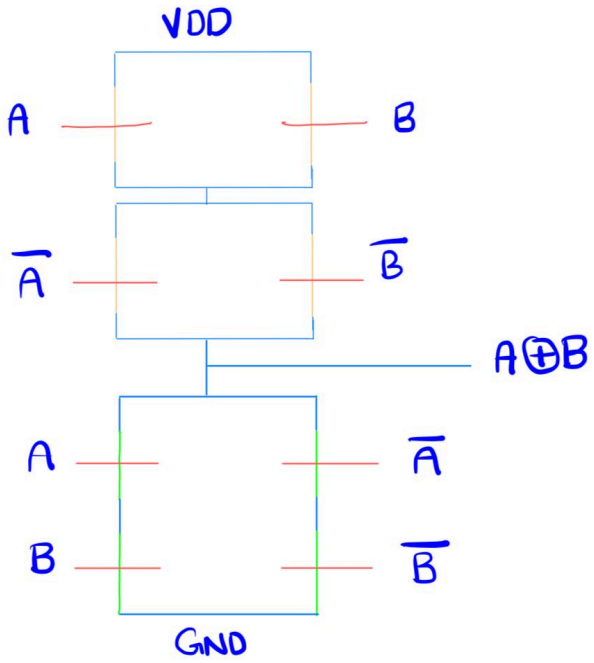
2 INPUT NOR GATE:



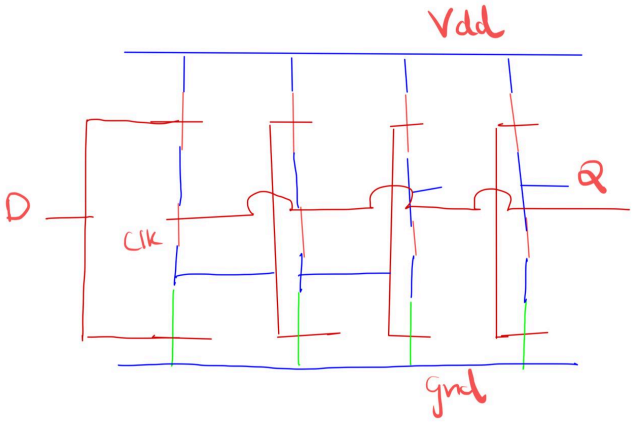
INVERTER:



2 INPUT XOR:

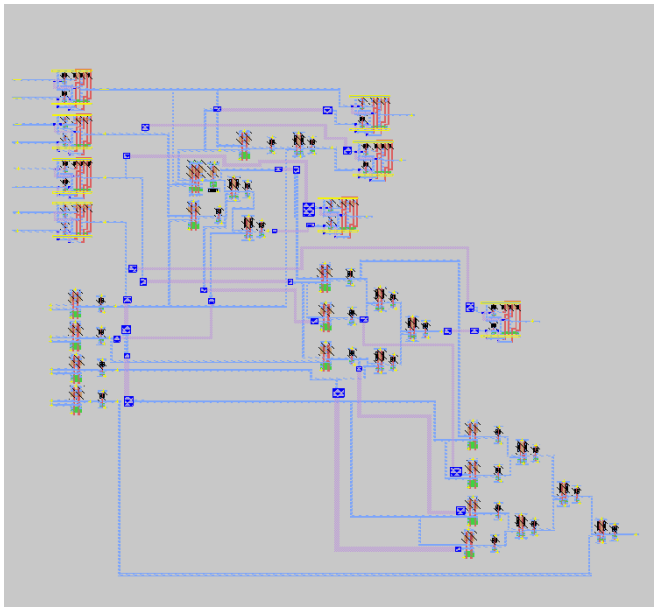


TSPC FLIP FLOP:

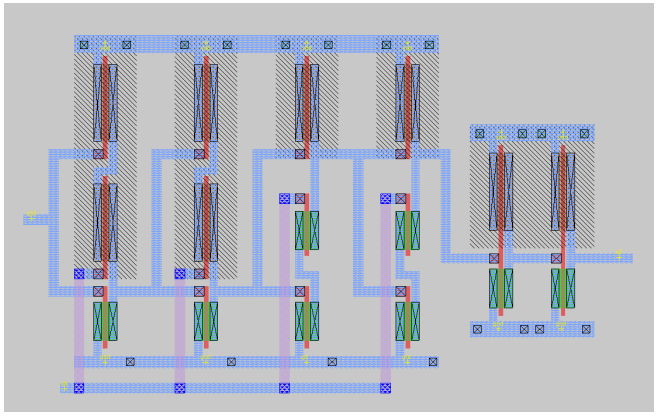


6. POST LAYOUT

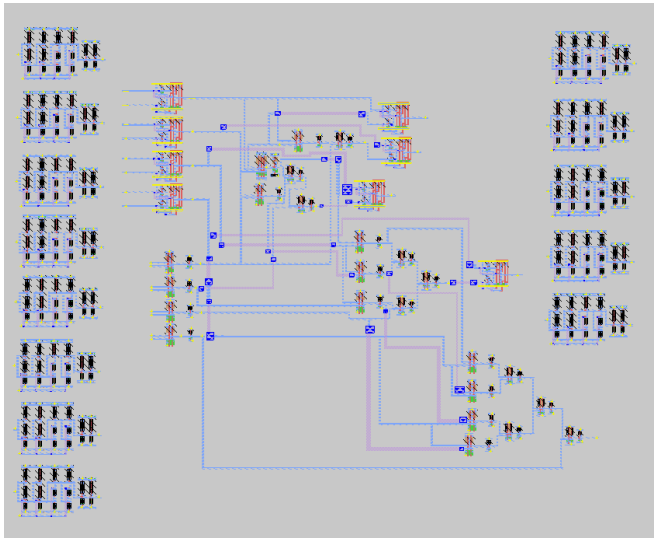
CLA MAGIC LAYOUT:



FLIP FLOP LAYOUT:

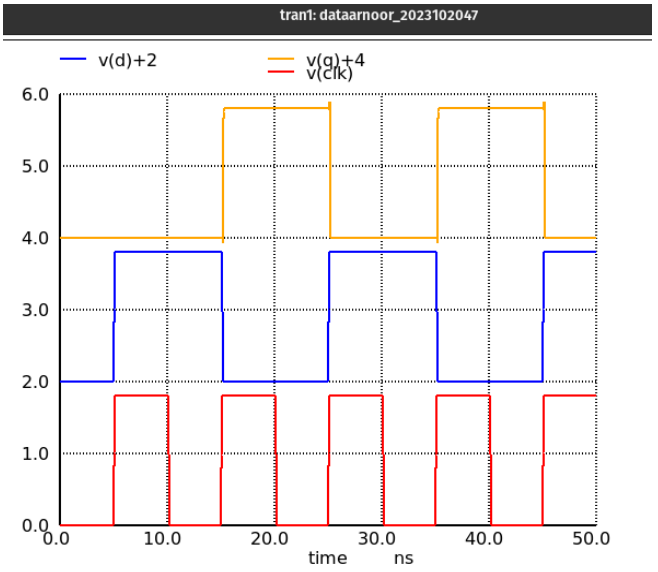


COMPLETE CLA ADDER:

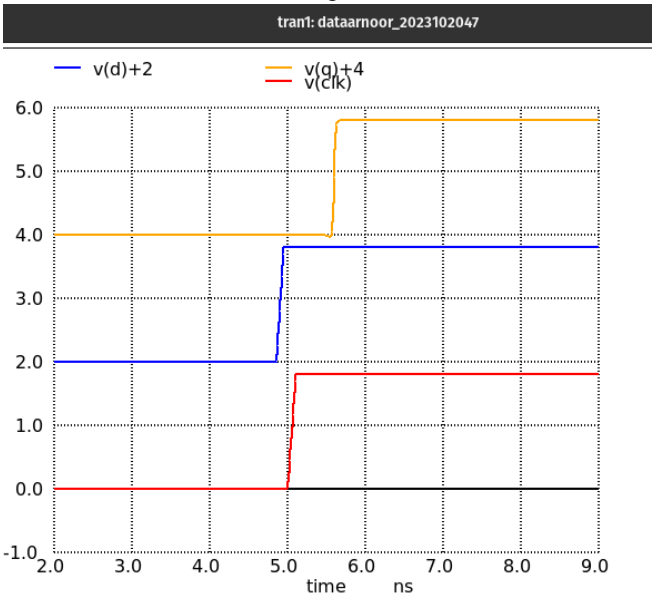


Post layout flip flop values and simulations:

```
tpcq_max          =  1.852222e-10

tpcq_min          =  1.338519e-10
```



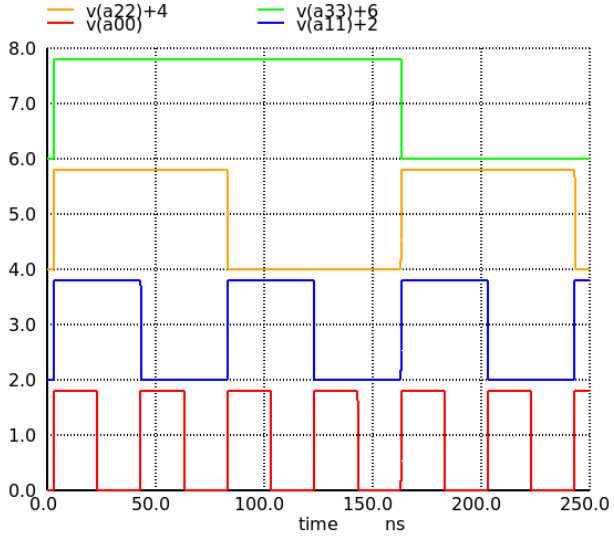
For setup time:

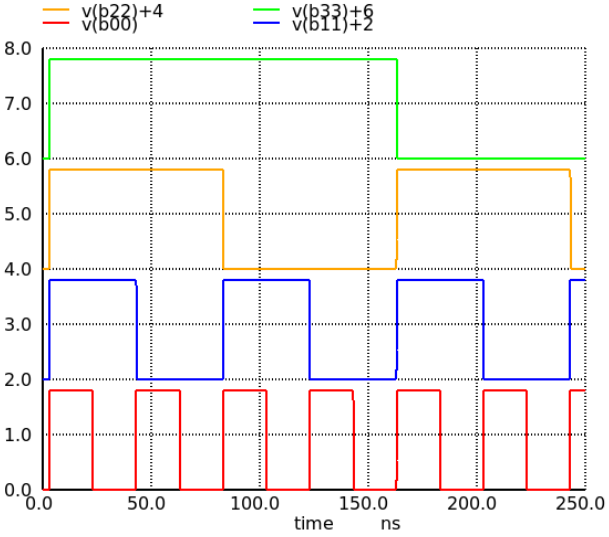


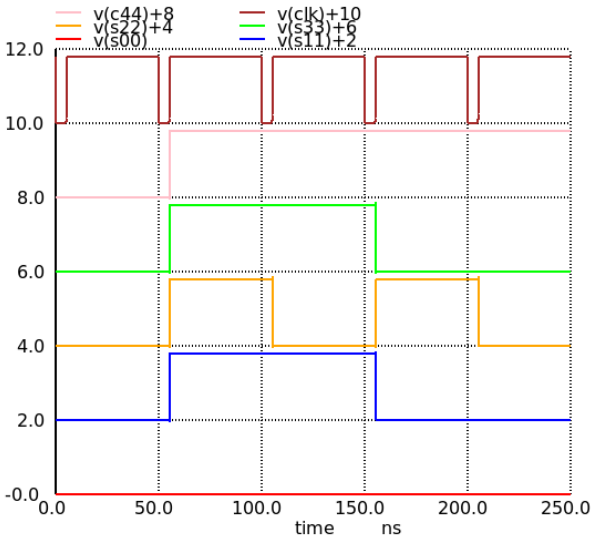
The setup time comes out to be 0.149ns

And hold time for post layout also comes out to be 0ns

POST LAYOUT CLA RESULTS:







	PRE LAYOUT	POST LAYOUT
setup time	0.085ns	0.149ns
hold time	0	0
tpcq_min	0.062ns	0.133ns
tpcq_max	0.083ns	0.185ns

7. WORST CASE DELAY AND MAXIMUM CLOCK FREQUENCY:

Worst case delay implies that the delay for that input values comes out to be maximum and worst delay comes out for inputs 1111 and 1111 and the value pf t_pd_max comes out as 0.65ns

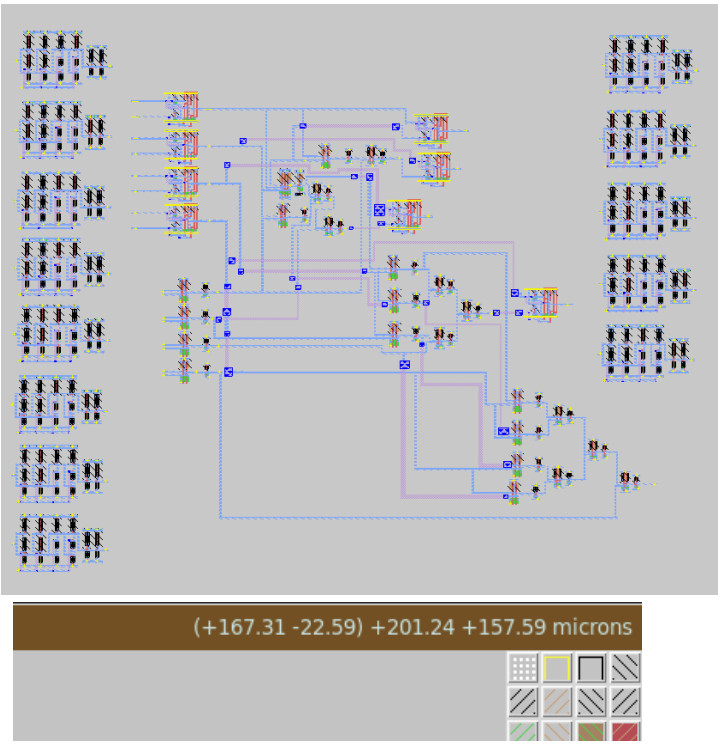
$$T_{\text{clk,min}} = t_{\text{setup}} + t_{\text{propagation, max}} + t_{\text{pcq, max}}$$

t_clk_min=0.083+0.65ns+0.083ns

Therefore, t_clk_min=0.816 ns

and fmax=1/t_clk_min=1.22GHz

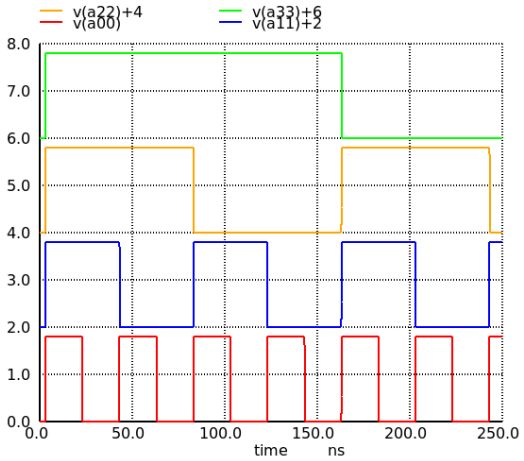
8. FLOOR PLAN

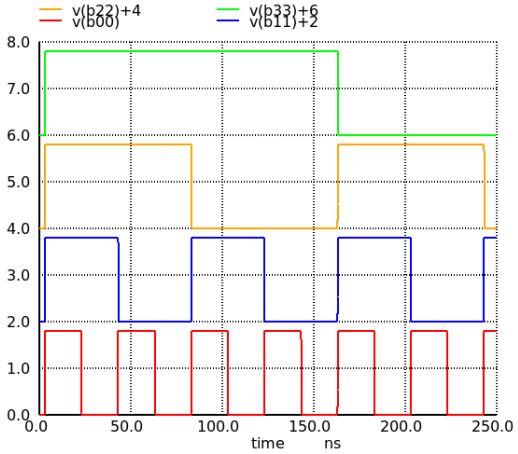


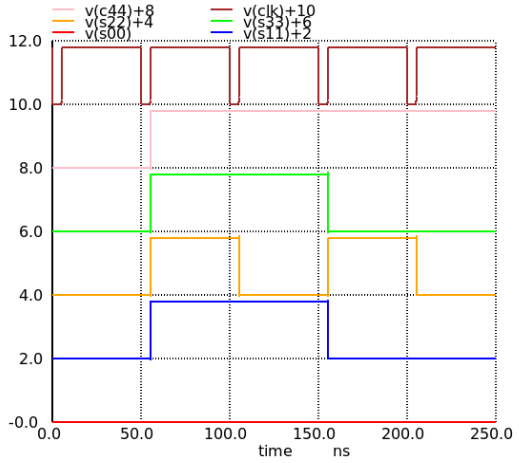
HORIZONTAL PITCH= 201.24 um

VERTICAL PITCH= 157.59 um

9. SIMULATIONS OF THE COMPLETE CIRCUIT:







To find t_pd_maxwe will consider the case of 1111 and 1111 and find the t_pd_max for post layout in the same way we did it for pre layout. t_pd_max comes out to be 0.55 ns

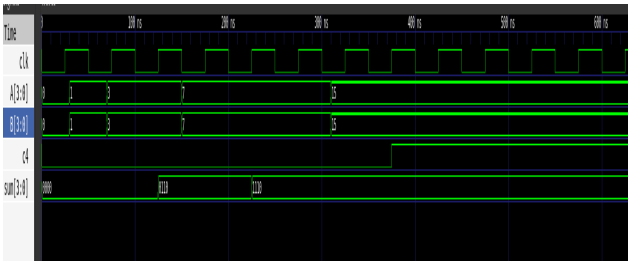
t_clk_min= 0.185+ 0.55+0.149=1.5175 ns

	pre layout	post layout
t_pd_max	0.65ns	0.552ns
t_clk_min	0.816ns	1.5175ns

10. MAXIMUM CLOCK FREQUENCY

fmax=1/t_clk_min=00.65GHz

11. IMPLEMENTATION USING VERILOG HDL



The circuit integrates a 4-bit CLA with D FLIP FLOP for synchronised operation. Inputs a[3:0], b[3:0], and cin are synchronized using DFFs, producing outputs labeled as aff, bff, and cinff. The CLA logic generates propagate (p[i]) and generate (g[i]) signals, which are used to compute the carry signals (c[1] to c[3] and cout) through hierarchical parallel logic. The sum (sum[i]) is calculatedas p[i] xor c[i]. Finally, DFFs at the outputs store the results as sumff and coutff, ensuring synchronized outputs for pipelined sequential operation.

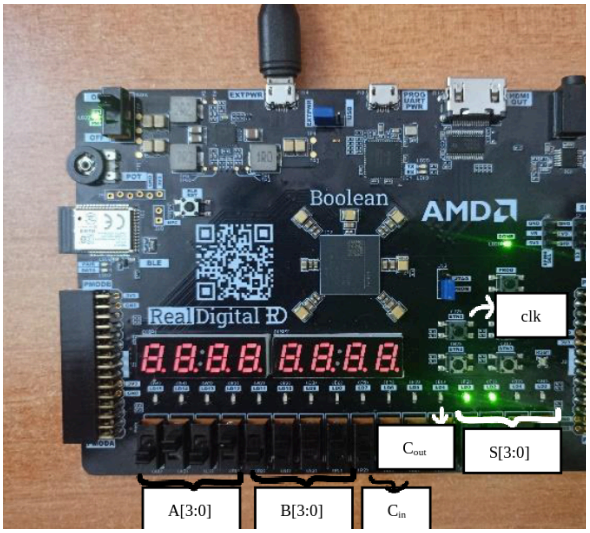
Example 1 : a=’1111’ and b=’1111’.

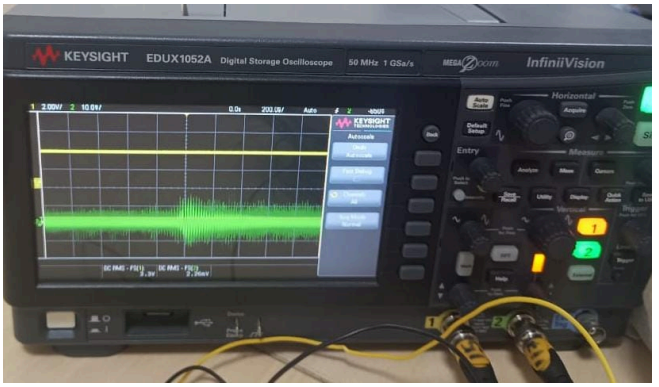
As we can see output carry is 1 and S[0:3] is E (1110).

Thus our answer is ’11110’ which is correct.

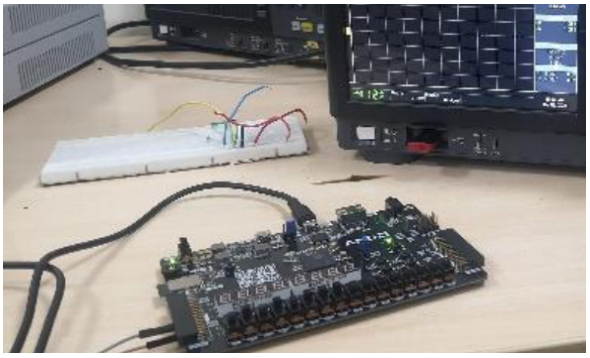
Example 2 : a=’1100’ and b=’0011’.

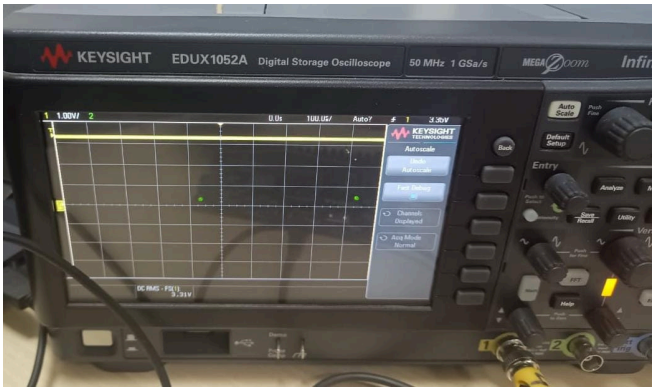
12. IMPLEMENTATION ON A FPGA BOARD:



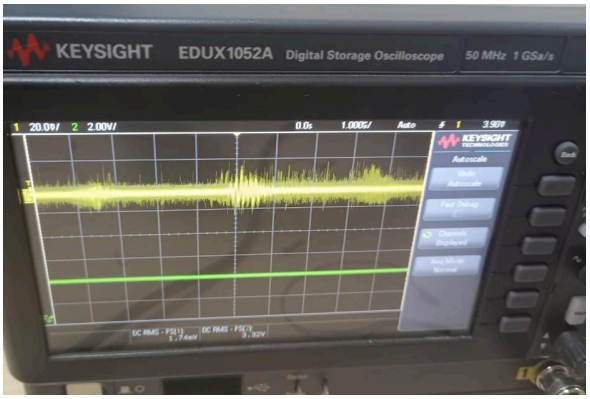


CIRCUIT:





TO TAKE OUTPUT ON A DSO, AN EXAMPLE IS TAKEN SUCH THAT A=7 AND B =22 AND THE OUTPUT RESULTS ARE:



REFERENCES

[1] Digital Logic and Computer Design by Morris Mano.
[2] Computer Architecture and Organization by John P. Hayes.
[3] CMOS VLSI Design (fourth edition) by Weste and Harris.
[4] Fundamentals of Digital Logic with Verilog Design by Brown and Vranesic.