

VLSI COURSE PROJECT

Carry Look Ahead Adder

Sanjana Sheela

2023102027

IIIT Hyderabad

Abstract— This is the design of a 4-bit carry-lookahead adder (CLA), which is optimized to reduce the number of transistors and minimize delay compared to a ripple-carry adder. The CLA efficiently computes the sum of two 4-bit numbers.

I. INTRODUCTION

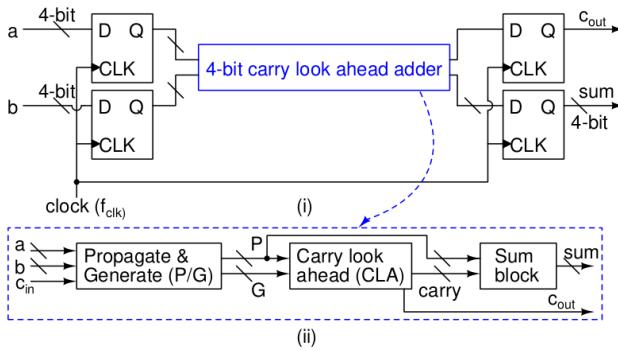


Fig. 1. Schematic of the 4-bit Carry Lookahead Adder

This is the conventional design of the carry look ahead adder. Instead of waiting for the previous carry to get generated we find all of the sums and the carries together. In a CLA adder we input the 4 bit numbers and a carry-in. In the first stage of the CLA we generate the propagate and generate signals for each bit of the input as follows.

$$p_i = a_i \oplus b_i$$

$$g_i = a_i \cdot b_i$$

Then we find the individual carries as follows

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

$$c_1 = p_1 c_0 + g_1$$

$$c_2 = p_2 c_1 + g_2$$

$$c_3 = p_3 c_2 + g_3$$

$$c_4 = p_4 c_3 + g_4$$

$$c_4 = p_4 p_3 p_2 p_1 c_0 + p_4 p_3 p_2 g_1 + p_3 g_2 + g_3$$

C_i	A_i	B_i	$A_i \oplus B_i$	$A_i + B_i$	$A_i \cdot B_i$	C_{i+1}
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	0	1	1	1
1	0	0	0	0	0	0
1	0	1	1	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	1	1

Doing this operation we can find the carries simultaneously, after these carries are generated we can find the sum as follows

$$sum_i = p_i \oplus c_i$$

There are flipflops on both of the sides of the combinational circuit which makes the circuit dynamic and the circuit works according to the clock pulses. The circuit is designed such that consider 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 like displayed in Fig. 2.

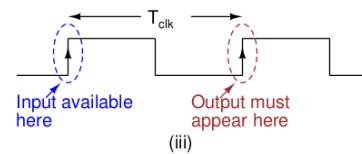


Fig. 2. Circuit behaviour based on clock

II. DESIGN DETAILS

In this design all the circuits are static. The sizing of the gates is done such that the worst case situations or paths of the circuits are equivalent to the sizes of the PUN and PDN of the inverter to match the inverters drive strength. The sizing is based on the inverter because CMOS inverters are used to determine the unit delay of a logic path. To match the drive strength we tend to match the resistance in the worst case path of that gate, based on the number of transistors the width changes. Delay is proportional to the number of transistors and to the resistance offered. The resistance is inversely proportional to the width of the transistors.

$$R \propto \frac{1}{W}$$

$$R_{eq} = R_1 + R_2 + \dots + R_n$$

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n}$$

$$Delay \propto \frac{C_{in}}{W}$$

$$\frac{C_{in,gate}}{W_{gate}} = \frac{C_{in,inverter}}{W_{inverter}}$$

A. AND gate

The AND gate is realized by static CMOS logic by cascading static CMOS NAND gate with the static CMOS not gate.

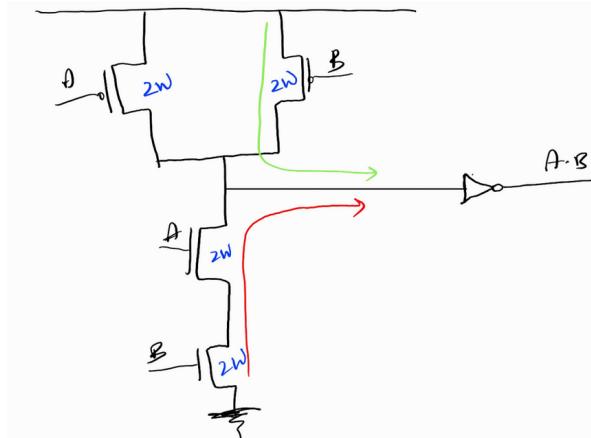


Fig. 3. Critical paths and Sizing of AND gates

B. XOR gate

In this design the implementation of the XOR gate is done by using 4 transistors.

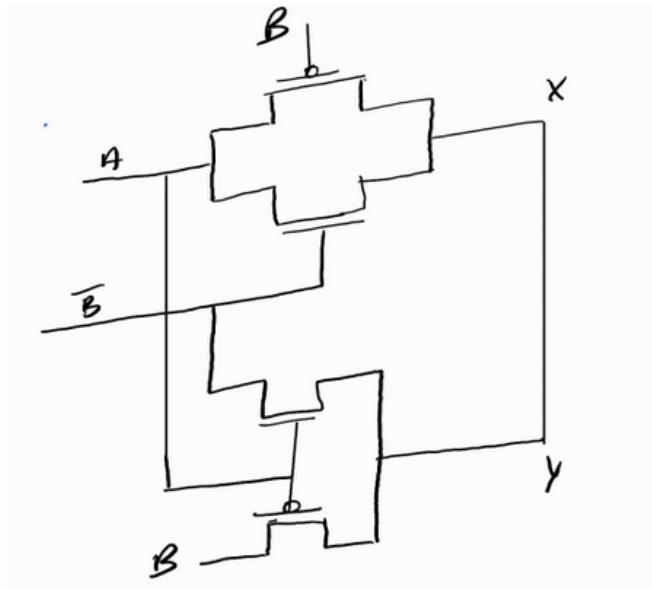


Fig. 4. XOR gate design

C. Carry Blocks

The carry blocks are made by CMOS logic by converting the equations into PDN and its complement into PUN.

1) C1block :

$$y = p_1 c_0 + g_1$$

$$\bar{y} = (\bar{p}_1 + \bar{c}_0)\bar{g}_1$$

The output will give \bar{c}_1 so we put and inverter to get c_1

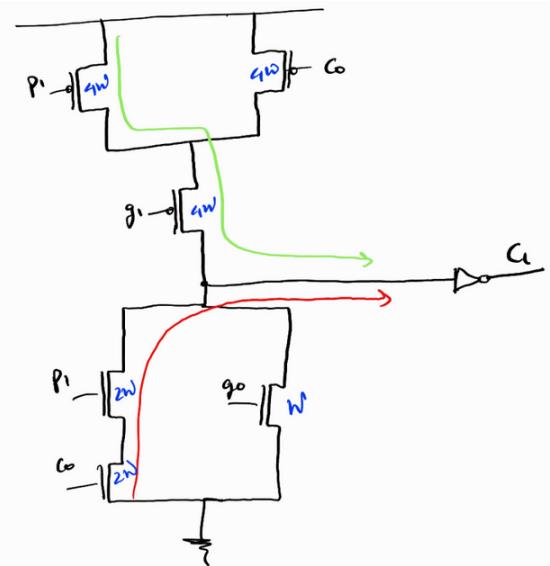


Fig. 5. Critical paths and Sizing of c1 circuit

2) C2block :

$$y = p_2 p_1 c_0 + p_2 g_1 + g_2$$

$$\bar{y} = (\bar{p}_2 + \bar{p}_1 + \bar{c}_0)(\bar{p}_2 + \bar{g}_1)(\bar{g}_2)$$

The output will give \bar{c}_2 so we put and inverter to get c_2

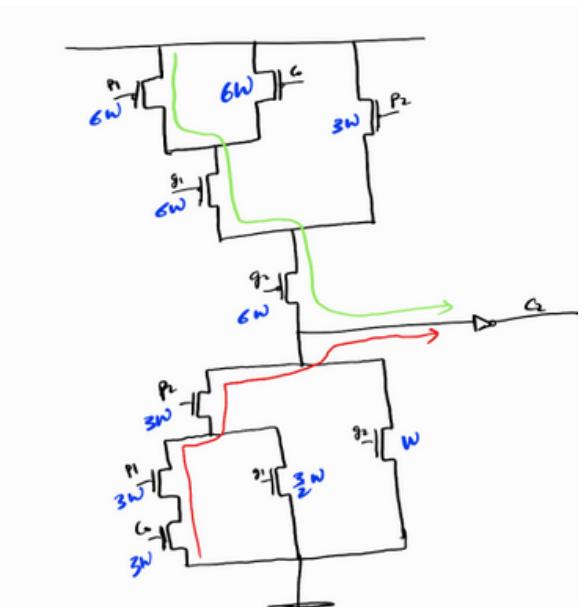


Fig. 6. Critical paths and Sizing of c2 circuit

3) C_3 block :

$$y = p_3 p_2 p_1 c_0 + p_3 p_2 g_1 + p_3 g_2 + g_3$$

$$\bar{y} = (\bar{p}_3 + \bar{p}_2 + \bar{p}_1 + \bar{c}_0)(\bar{p}_3 + \bar{p}_2 + \bar{g}_1)(\bar{p}_3 + \bar{g}_2)(\bar{g}_3)$$

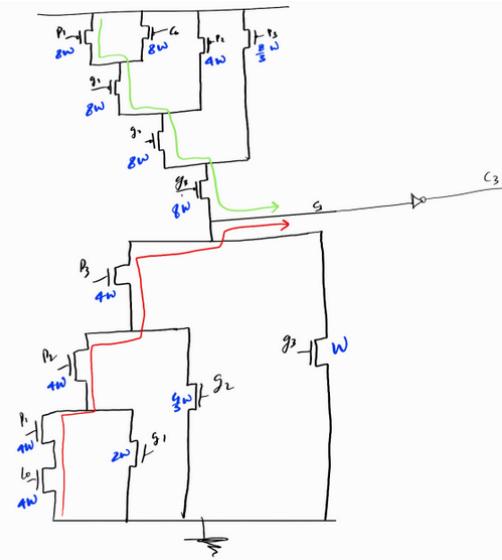


Fig. 7. Critical paths and Sizing of c_3 circuit

4) C_4 block :

$$y = p_4 p_3 p_2 p_1 c_0 + p_4 p_3 p_2 g_1 + p_3 g_2 + g_3$$

$$\bar{y} = (\bar{p}_3 + \bar{p}_2 + \bar{p}_1 + \bar{c}_0)(\bar{p}_3 + \bar{p}_2 + \bar{g}_1)(\bar{p}_3 + \bar{g}_2)(\bar{g}_3)$$

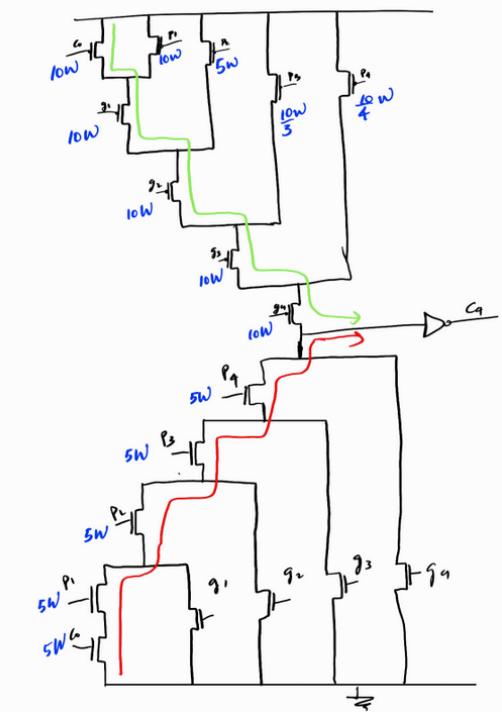


Fig. 8. Critical paths and Sizing of C_4

D. D-FlipFlop

In this design the D-FlipFlop used is TSPC with 12 transistors and two inverters at the end inorder to get a good signal. In total 16 transistors are used with NMOS of width W and PMOS of width 2W. It offers reduced power consumption due to single phase clocking, cause it ahhs fewer transistors it offers high speed operation adding to that it has $t_h = 0$.

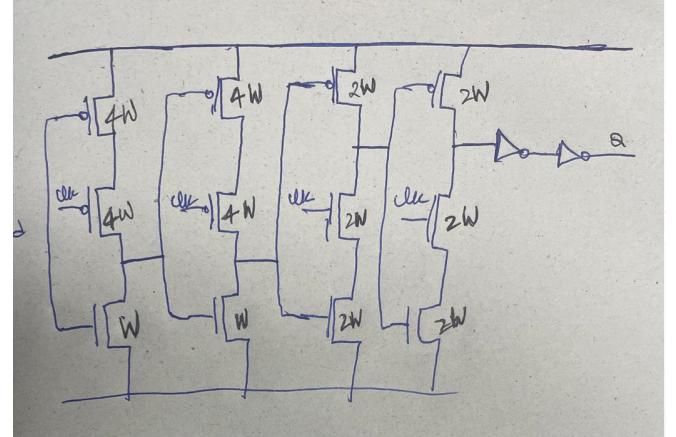


Fig. 9. Sizing of the D-FlipFlop

III. NGSPICE SIMULATIONS

Using NGSpice we can simulate the circuits and get the desired outputs.

A. Propagate and Generate block

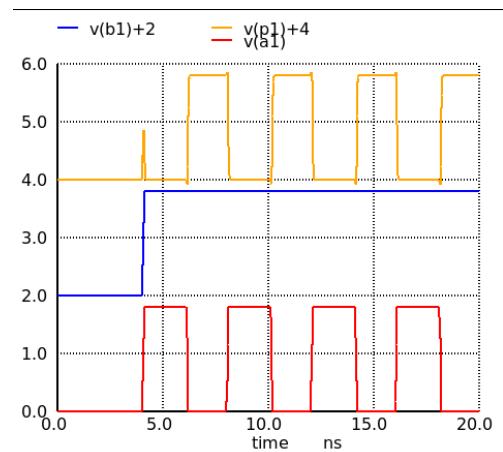


Fig. 10. $p_1 = a_1 \oplus b_1$

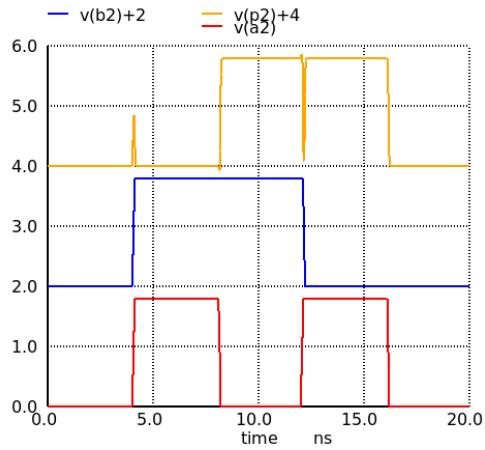


Fig. 11. $p_2 = a_2 \oplus b_2$

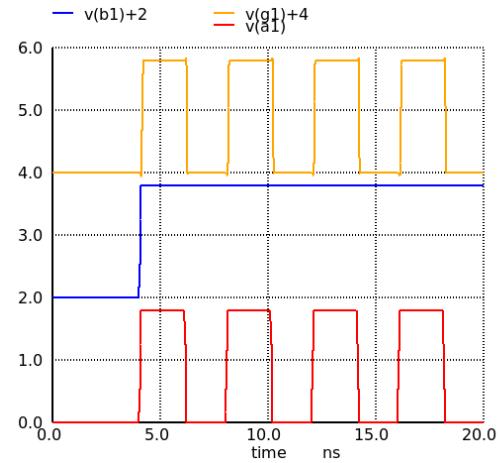


Fig. 14. $g_1 = a_1 \cdot b_1$

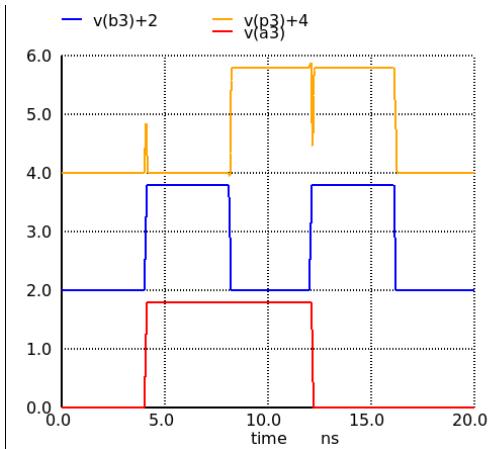


Fig. 12. $p_3 = a_3 \oplus b_3$

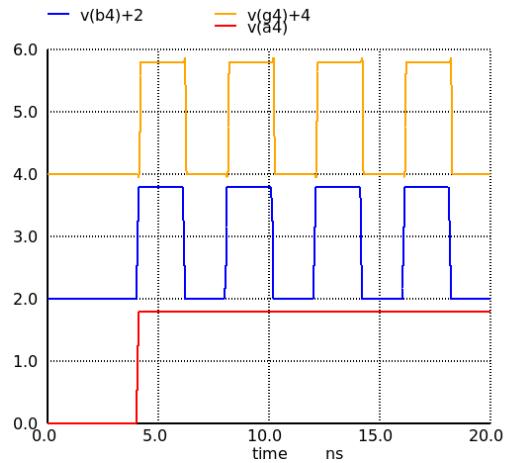


Fig. 15. $g_2 = a_2 \cdot b_2$

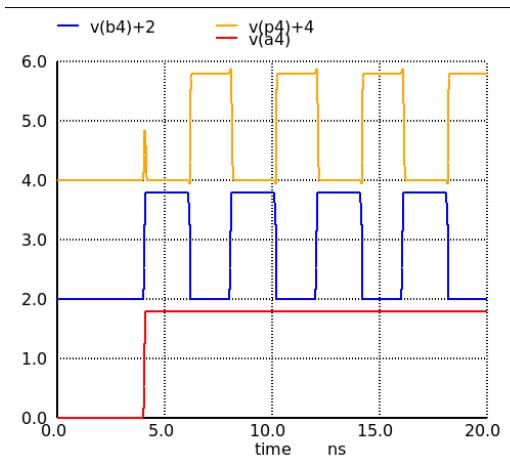


Fig. 13. $p_4 = a_4 \oplus b_4$

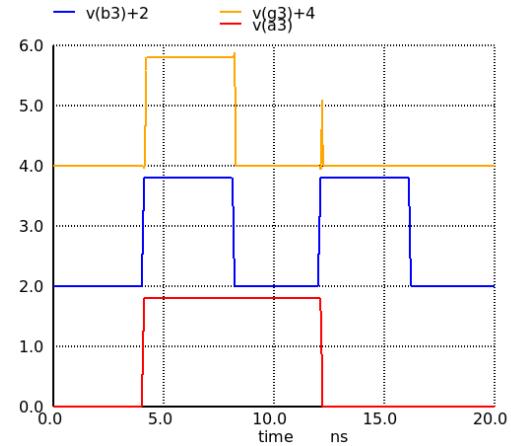


Fig. 16.
 $g_3 = a_3 \cdot b_3$

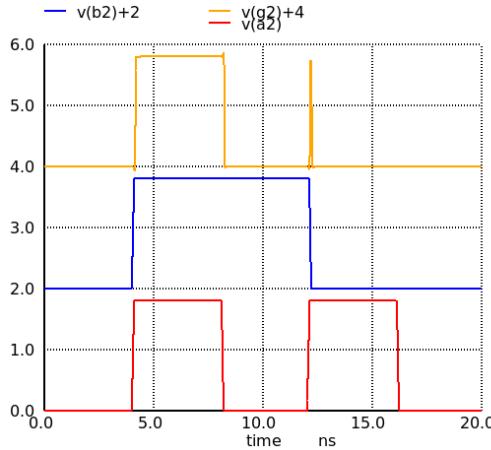


Fig. 17. $g_4 = a_4 \cdot b_4$

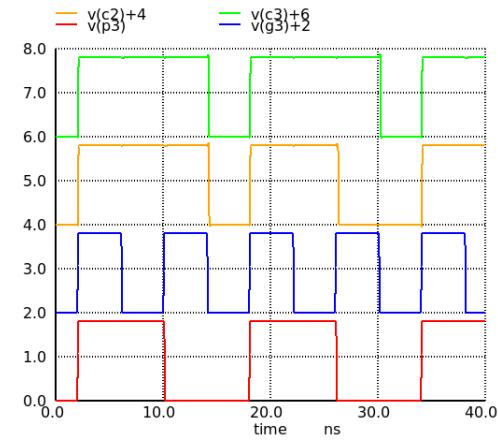


Fig. 20. $c_3 = p_3 c_2 + g_3$

B. Carry Blocks

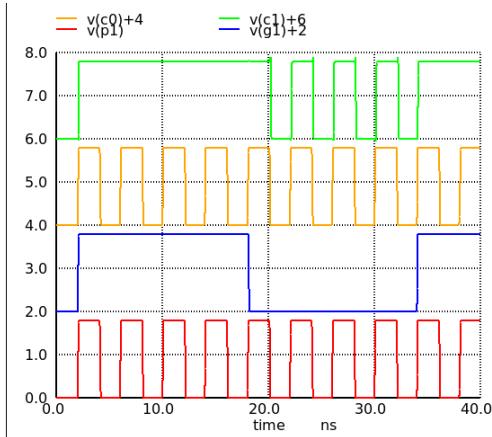


Fig. 18. $c_1 = p_1 c_0 + g_1$

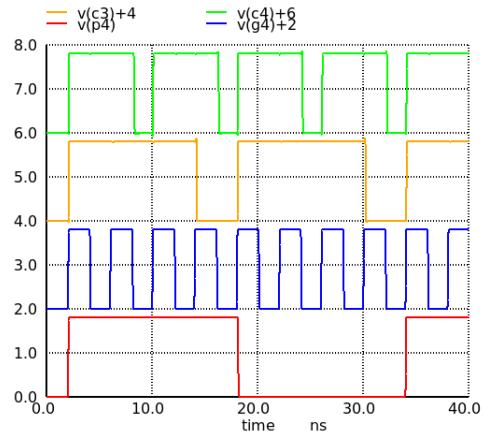


Fig. 21. $c_4 = p_4 c_3 + g_4$

C. Sum Block

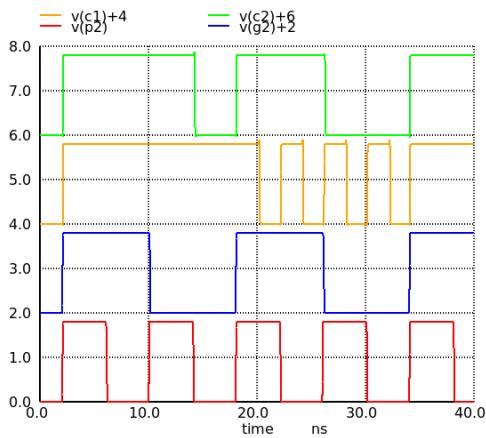


Fig. 19. $c_2 = p_2 c_1 + g_2$

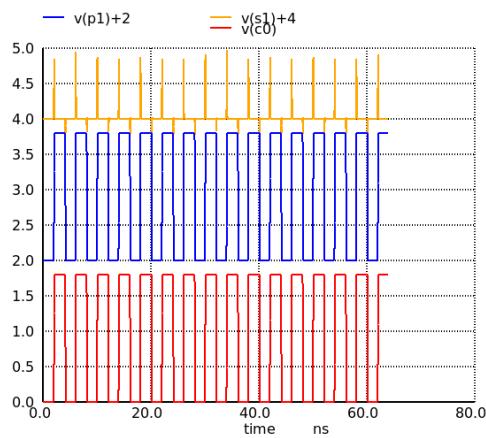


Fig. 22. $s_1 = p_1 \oplus c_0$

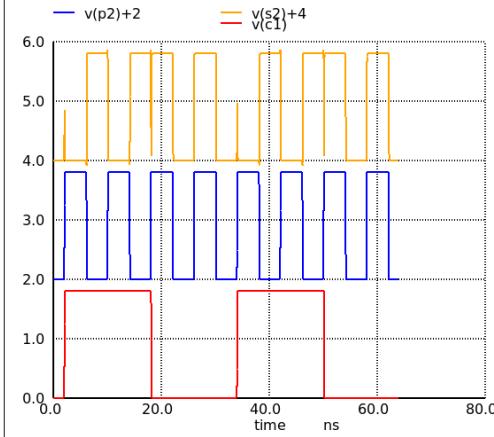


Fig. 23. $s_2 = p_2 \oplus c_1$

D. D-FlipFlop

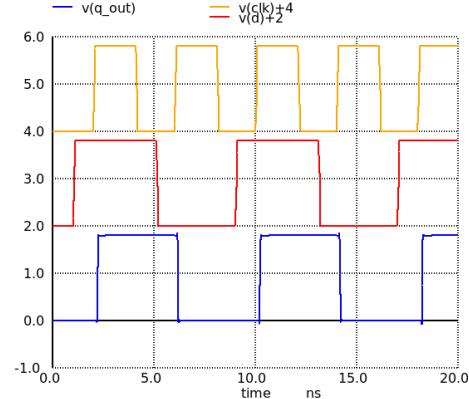


Fig. 26. Output of the TSPC D-flipflop

IV. SETUP TIME, HOLD TIME ,CLOCK TO Q DELAY

A. t_{su}

The setup time is the time before the clock edge when the input should be kept constant and not be changed. For the D-flipflop configuration i used the tpsc has set up time of 0.15ns .So the input should not change in that time before the clock edge comes.

B. t_h

Hold time is the time after the clock edge where the input to the D-flipflop should not change and keep it constant. The hold time of this tpsc is 0.004ns which is almost 0 so the hold time of the tpsc is 0.

C. t_{PCQ}

It is the delay between the clock and the Q. It is the delay of the tpsc. The t_{QLH} is 0.165ns and t_{QHL} is 0.133ns. 165 p is the t_{PCQmax} and 133p is the t_{PCQmin} of the tpsc.

V. STICK DIAGRAMS

A. NOT gate

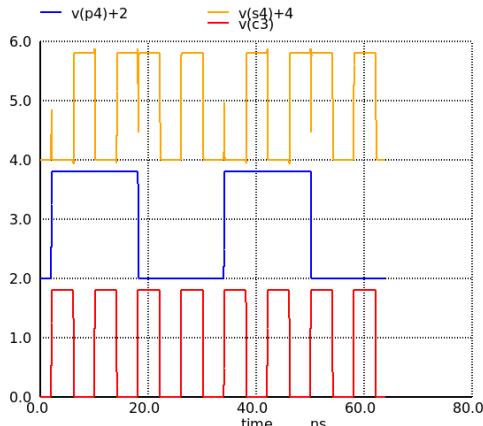


Fig. 25. $s_4 = p_4 \oplus c_3$

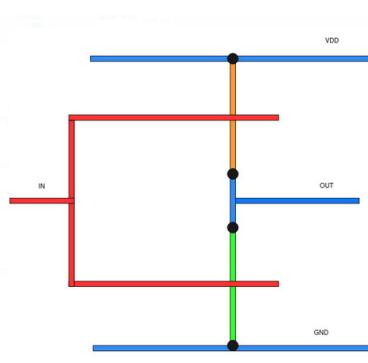


Fig. 27. Stick Diagram of NOT Gate

B. AND gate

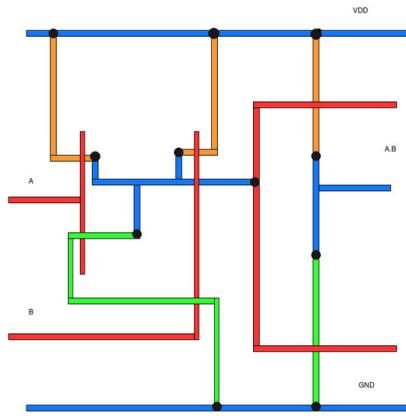


Fig. 28. Stick Diagram of AND Gate

C. XOR gate

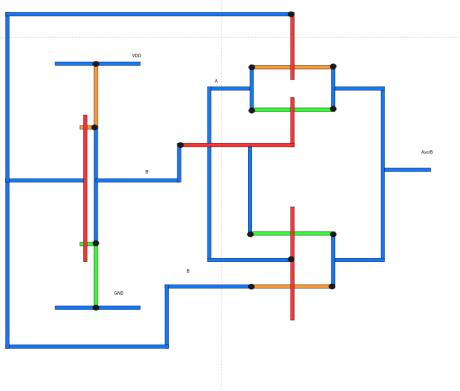


Fig. 29. Stick Diagram of XOR Gate

D. Carry Blocks

1) C_1 block : The stick diagram of C_1 block.

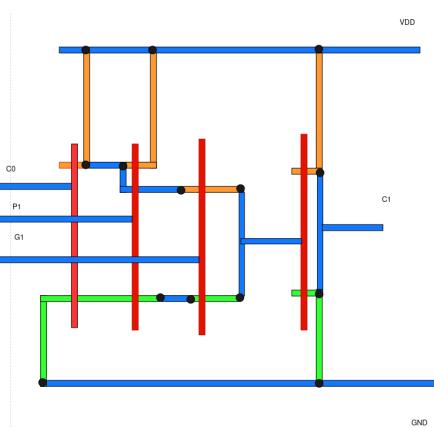


Fig. 30. Stick Diagram of C1 Block

2) C_2 block : The stick diagram of C2 block.

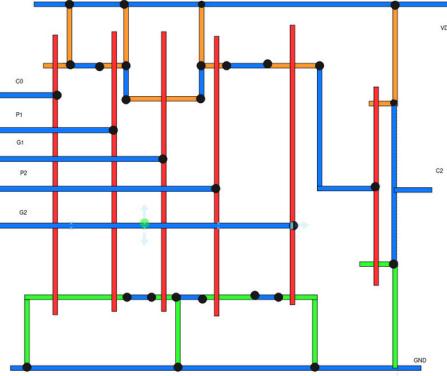


Fig. 31. Stick Diagram of C2 Block

3) C_3 block : The stick diagram of C_3 block.

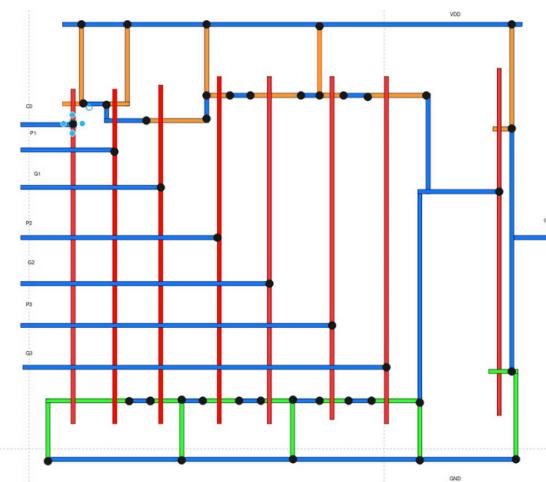


Fig. 32. Stick Diagram of C3 Block

4) C_4 block : The stick diagram of C_4 block.

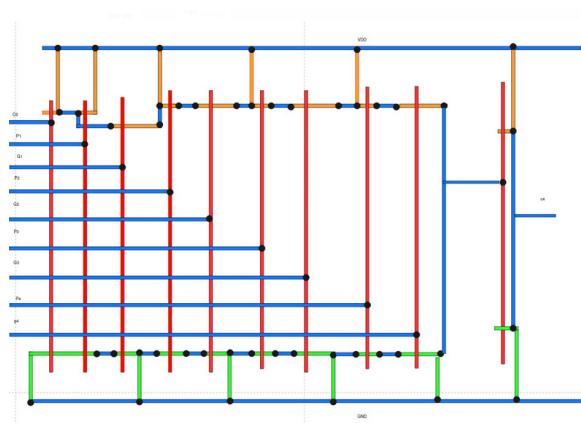


Fig. 33. Stick Diagram of C4 Block

VI. MAGIC LAYOUT

B. Carry Block

A. Propogate and Generate Block

1) AND gate: Magic layout of AND gate.

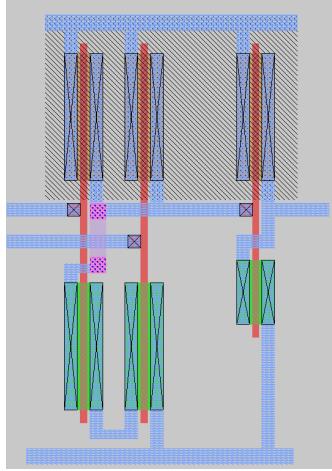


Fig. 34. AND gate Magic Layout

2) XOR gate: Magic layout of Xor gate.

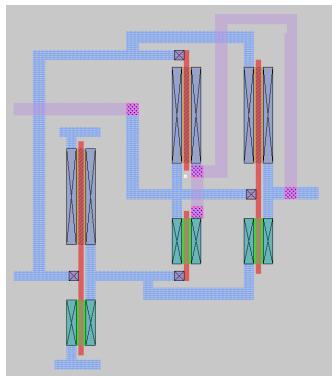


Fig. 35. XOR Magic Layout

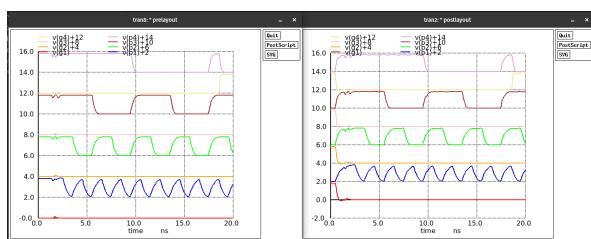


Fig. 36. Comparision of the Propogate and Generate Block

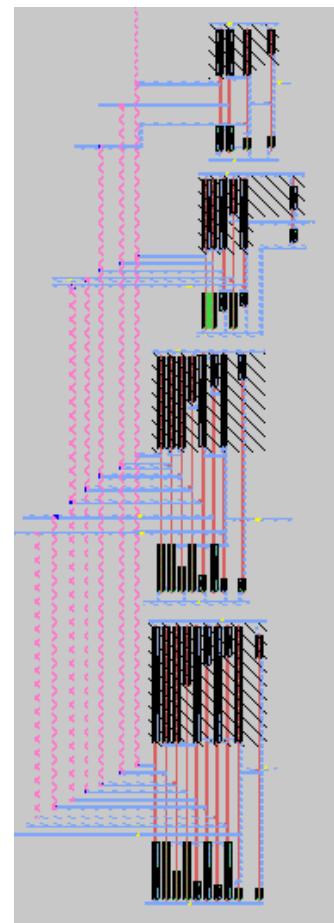


Fig. 37. Carry Block Magic Layout

1) C₁block : Magic layout of C1 block.

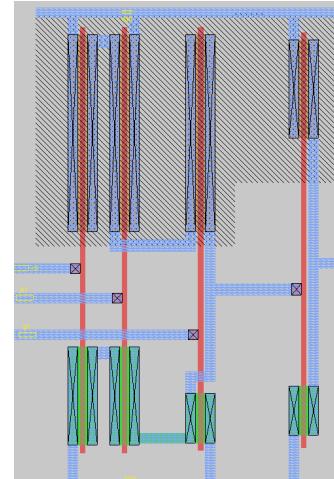


Fig. 38. C1 BLock Magic layout

2) C₂block: Magic layout of C2 block.

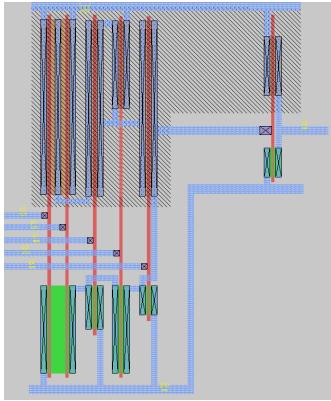


Fig. 39. C2 block Magic Layout

3) C_3 block : Magic layout of C_3 block.



Fig. 40. C3 Block Magic Layout

4) C_4 block : Magic layout of C_4 block.

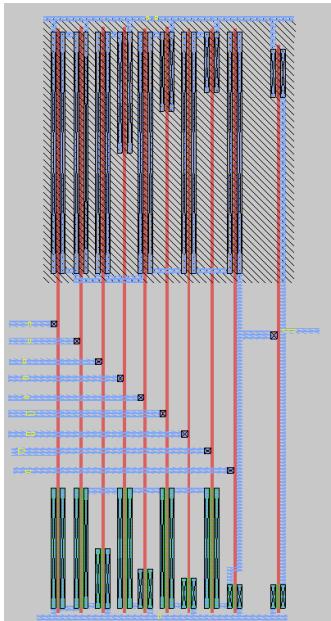


Fig. 41. C4 Block Magic Layout

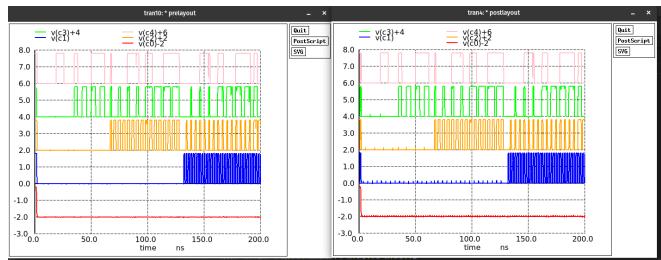


Fig. 42. Comparision of the Carry Block

C. Sum Block

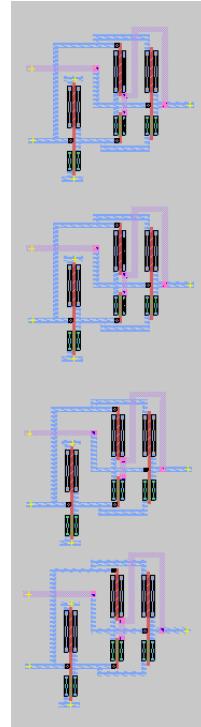


Fig. 43. Sum Block Magic Layout

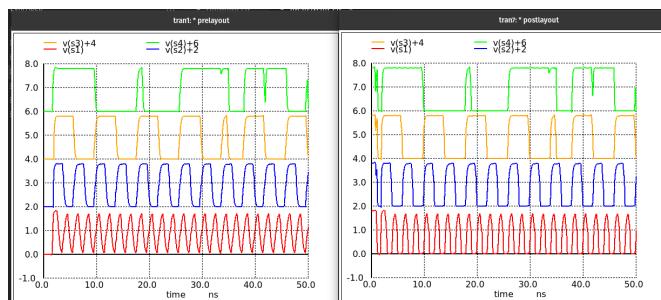


Fig. 44. Comparision of the Sum Block

D. D-FlipFlop

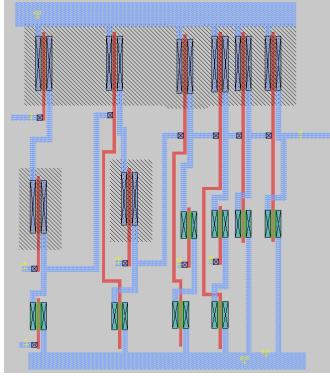


Fig. 45. D-FLipFlop Magic Layout

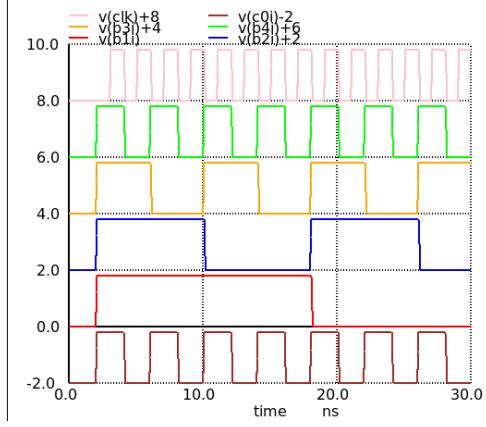


Fig. 48. Input b of the Prelayout

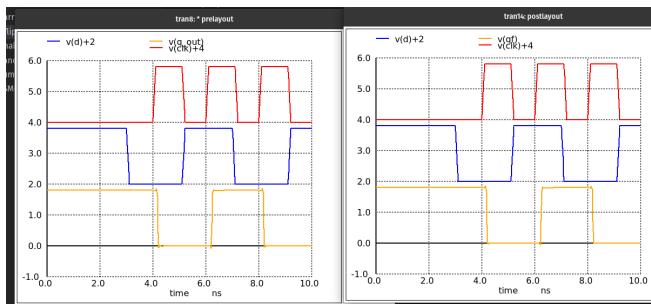


Fig. 46. Comparision of the tpc

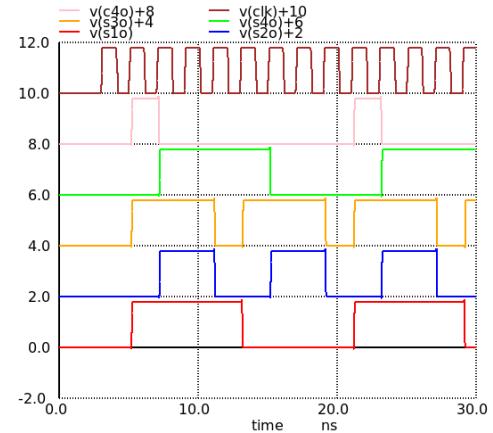


Fig. 49. Output of the Prelayout

416ps is the worst case delay of the Carry Look Ahead Adder of the Prelayout. the maximum frequency for which the circuit is working is 2GHz.

VII. PRE LAYOUT

VIII. FLOOR PLAN

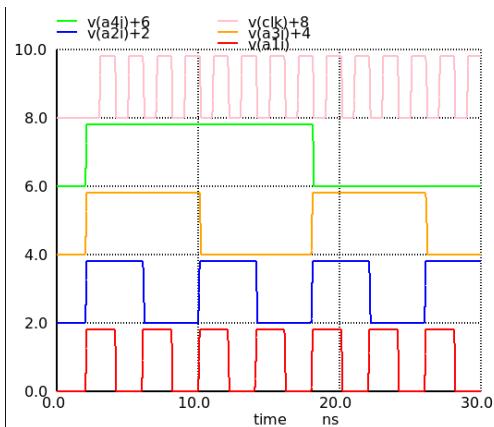


Fig. 47. Input a of the Prelayout

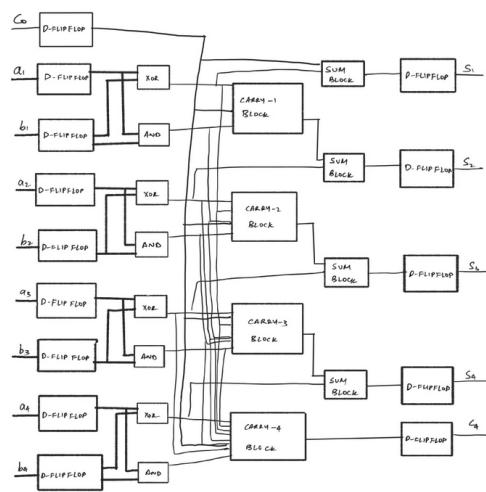


Fig. 50. Floor Plan of the full circuit

127.44um is the Horizontal Pitch and 185.13um is the Vertical Pitch.

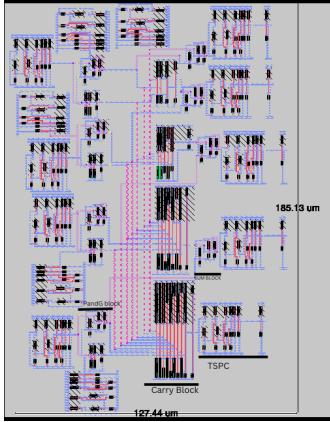


Fig. 51. Vertical and Horizontal Pitches

IX. POST LAYOUT

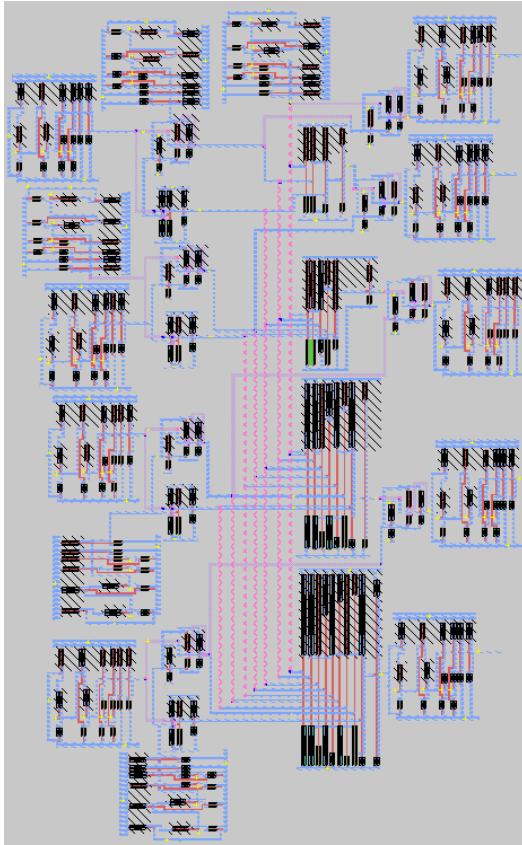


Fig. 52. Full Circuit Magic Layout

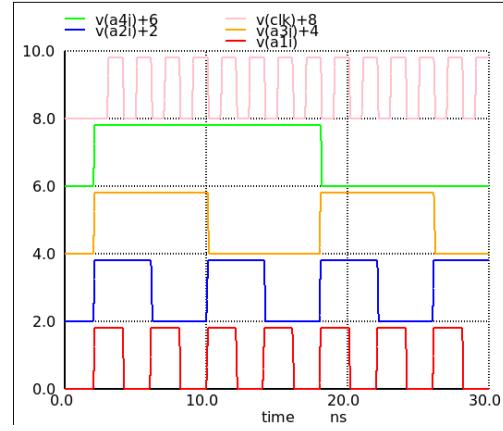


Fig. 53. Input a of the Postlayout

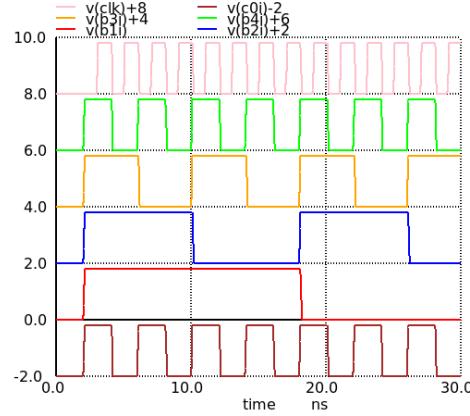


Fig. 54. Input b of the Postlayout

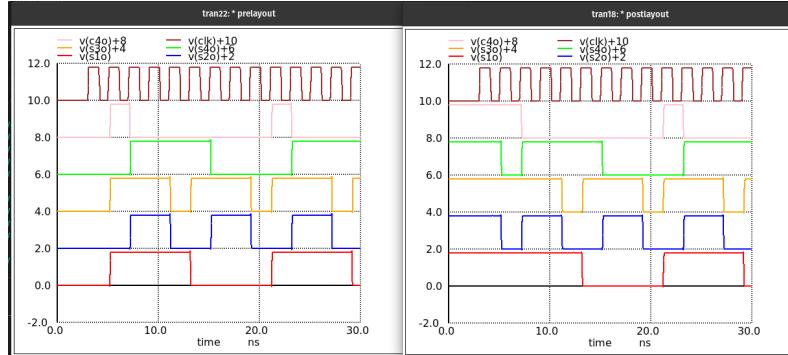


Fig. 55. Outputs of final circuit

	PreLayout	PostLayout
Setup Time	0.15ns	0.06ns
Hold Time	0.004	0.005ns
$T_{clk\min}$	0.57	0.8
f_{max}	1.75GHz	1.56GHz
$T_{pd\max}$	0.416ns	0.563ns

TABLE I
PRELAYOUT AND POSTLAYOUT SIMULATION RESULTS

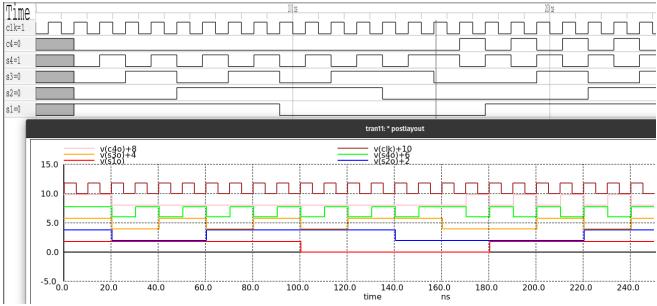


Fig. 58. GTKWave compare with the postlayout

The following are the outputs I got for the case A = 0101 B=1101

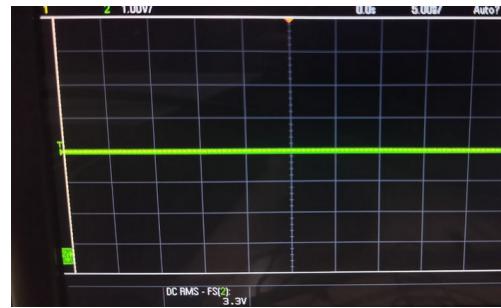


Fig. 60. C4 waveform

X. SPECIFICATIONS

The max delay of the combinational circuit is 563ps. The circuit works at a maximum frequency of 2GHz. The t_{QH} is 0.185ns and t_{QH} is 0.132ns of the postlayout. The t_{su} is 0.06ns of the Dflipflop of the post layout.

$$T_{clk} = T_{PCQmax} + T_{pdmax} + T_{su}$$

$$T_{clk} = 0.18ns + 0.563ns + 0.06ns$$

Frequency of the clock is 1.56GHz. Frequency at which the circuit works is almost 1GHz.

XI. VERILOG SIMULATION

```
Time = 285000 | clk = 1 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 0, b4 = 0 | c0 = 1 | s1 = 0, s2 = 1, s3 = 1, s4 = 0 | c4 = 1
Time = 290000 | clk = 0 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 0, b4 = 1 | c0 = 0 | s1 = 0, s2 = 1, s3 = 1, s4 = 0 | c4 = 1
Time = 295000 | clk = 1 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 0, b4 = 1 | c0 = 1 | s1 = 0, s2 = 0, s3 = 1, s4 = 1 | c4 = 0
Time = 300000 | clk = 0 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 1, b4 = 0 | c0 = 0 | s1 = 0, s2 = 0, s3 = 1, s4 = 1 | c4 = 0
Time = 305000 | clk = 1 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 1, b4 = 0 | c0 = 1 | s1 = 0, s2 = 0, s3 = 1, s4 = 0 | c4 = 1
Time = 310000 | clk = 0 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 1, b4 = 1 | c0 = 0 | s1 = 0, s2 = 0, s3 = 1, s4 = 0 | c4 = 1
Time = 315000 | clk = 1 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 1, b4 = 1 | c0 = 1 | s1 = 0, s2 = 0, s3 = 1, s4 = 0 | c4 = 1
Time = 320000 | clk = 0 | a1 = 0, a2 = 0, a3 = 0, a4 = 1 | b1 = 1, b2 = 1, b3 = 1, b4 = 1 | c0 = 0 | s1 = 0, s2 = 0, s3 = 1, s4 = 0 | c4 = 1
Time = 325000 | clk = 1 | a1 = 0, a2 = 0, a3 = 1, a4 = 0 | b1 = 0, b2 = 0, b3 = 0, b4 = 0 | c0 = 1 | s1 = 0, s2 = 0, s3 = 0, s4 = 1 | c4 = 0
Time = 330000 | clk = 0 | a1 = 0, a2 = 0, a3 = 1, a4 = 0 | b1 = 0, b2 = 0, b3 = 0, b4 = 0 | c0 = 0 | s1 = 0, s2 = 0, s3 = 0, s4 = 1 | c4 = 0
Time = 335000 | clk = 1 | a1 = 0, a2 = 0, a3 = 1, a4 = 0 | b1 = 0, b2 = 0, b3 = 0, b4 = 0 | c0 = 1 | s1 = 0, s2 = 0, s3 = 1, s4 = 0 | c4 = 0
Time = 340000 | clk = 0 | a1 = 0, a2 = 0, a3 = 1, a4 = 0 | b1 = 0, b2 = 0, b3 = 0, b4 = 0 | c0 = 0 | s1 = 0, s2 = 0, s3 = 1, s4 = 0 | c4 = 0
Time = 345000 | clk = 1 | a1 = 0, a2 = 0, a3 = 1, a4 = 0 | b1 = 0, b2 = 0, b3 = 1, b4 = 0 | c0 = 1 | s1 = 0, s2 = 0, s3 = 1, s4 = 1 | c4 = 0
Time = 350000 | clk = 0 | a1 = 0, a2 = 0, a3 = 1, a4 = 0 | b1 = 0, b2 = 0, b3 = 1, b4 = 0 | c0 = 0 | s1 = 0, s2 = 0, s3 = 1, s4 = 1 | c4 = 0
Time = 355000 | clk = 1 | a1 = 0, a2 = 0, a3 = 1, a4 = 1 | b1 = 0, b2 = 0, b3 = 1, b4 = 1 | c0 = 1 | s1 = 0, s2 = 0, s3 = 0, s4 = 1 | c4 = 1
```

Fig. 56. Verilog output of the final circuit

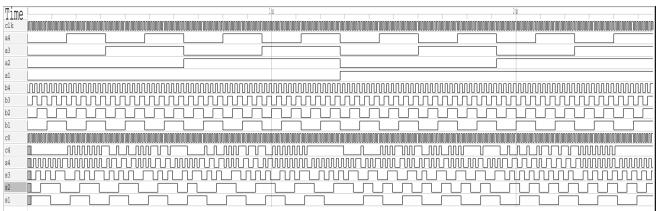


Fig. 57. GTKWave output of the final circuit

XII. FGPA WAVWFORMS

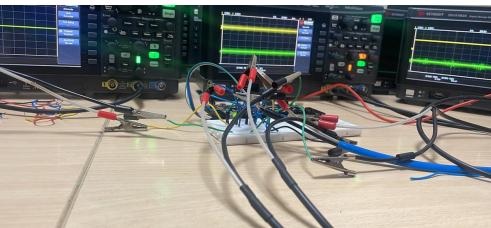


Fig. 59. FPGA board

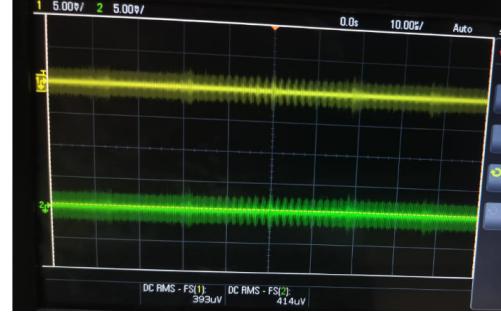


Fig. 61. S4 S3 waveforms

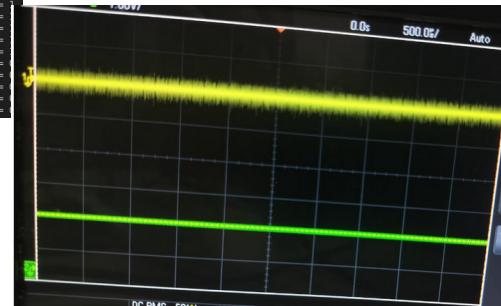


Fig. 62. S2 S1 waveforms