

Design and Implementation of a 4-bit Carry Look-Ahead (CLA) Adder in 180nm Technology

Monsoon 2024 VLSI Design Course Project, IIIT Hyderabad (Instructor: Prof. Abhishek Srivastava)

Krish Pandya
2023102026

*Department of Electronics and Communication Engineering
IIIT Hyderabad
Hyderabad, India*

Contents

I Introduction	2	IV-C9 XOR Gate Layout	6
II Design Overview	2	IV-C10 D Flip-Flop Layout	6
III Design Specifications and Methodology	2	IV-D Post-Layout Simulation	6
III-A Adder Architecture	2	IV-D1 AND Gate Post-Layout Simulation	6
III-B D Flip-Flop Design	2	IV-D2 AND3 Gate Post-Layout Simulation	7
III-C MAGIC Layout	2	IV-D3 AND4 Gate Post-Layout Simulation	7
III-D Gate and Module Structure	2	IV-D4 AND5 Gate Post-Layout Simulation	7
III-D1 Novel AND and XOR Gate Architecture	3	IV-D5 OR Gate Post-Layout Simulation	7
III-D2 Novel D Flip-Flop Architecture	3	IV-D6 OR3 Gate Post-Layout Simulation	7
III-D3 Static CMOS Logic Gates	3	IV-D7 OR4 Gate Post-Layout Simulation	7
IV-D8 OR5 Gate Post-Layout Simulation	7	IV-D9 XOR Gate Post-Layout Simulation	7
IV-D10 D Flip-Flop Post-Layout Simulation	7	IV-E Floor Planning and Stick Diagrams	8
IV Simulation Results	3	V-1 Vertical and Horizontal Pitches	8
IV-A NGSPICE Simulations	3	V-2 Stick Diagrams	8
IV-A1 AND Gate	3	V-3 Floor Plan	9
IV-A2 AND3 Gate	3	VI Final Circuit Simulation and Delay Analysis	9
IV-A3 AND4 Gate	3	VI-1 Simulation Video	10
IV-A4 AND5 Gate	3	VI-2 MAGIC Layout of CLA with DFF	10
IV-A5 OR Gate	3	VI-3 MAGIC Layout of CLA with Load Capacitance	10
IV-A6 OR3 Gate	3	VI-4 Delay Results	10
IV-A7 OR4 Gate	4	VI-5 Carry Look-Ahead Adder Simulation Results	10
IV-A8 OR5 Gate	4	VII Simulation Results Comparison	11
IV-A9 XOR Gate	4	VIII Delay and Maximum Clock Frequency	11
IV-A10 D Flip Flop	4	IX Verilog Implementation	11
IV-B D Flip-Flop Timing Analysis	4	IX-1 iVerilog Output	12
IV-C MAGIC Layouts	5	IX-2 GTWAVE Output	12
IV-C1 AND Gate Layout	5	IX-3 FPGA Outputs	12
IV-C2 AND3 Gate Layout	5	IX-4 Oscilloscope Waveforms	12
IV-C3 AND4 Gate Layout	5	X Conclusion	12
IV-C4 AND5 Gate Layout	5	XI Acknowledgements	12
IV-C5 OR Gate Layout	6		
IV-C6 OR3 Gate Layout	6		
IV-C7 OR4 Gate Layout	6		
IV-C8 OR5 Gate Layout	6		

This project is completed under the guidance of Prof. Abhishek Srivastava and supported by IIIT Hyderabad's Department of Electronics and Communication Engineering.

Abstract—This report presents the design, simulation, and FPGA implementation of a 4-bit Carry Look-Ahead (CLA) adder using 180nm technology. The design leverages static CMOS logic with optimized sizing of NMOS and PMOS transistors, implemented on MAGIC layout and verified with NGSPICE simulations. Post-layout simulations are conducted to evaluate the performance metrics, such as delay and maximum clock speed. The circuit functionality is further validated with Verilog HDL and oscilloscope waveforms from FPGA testing.

Index Terms—Carry Look-Ahead Adder, VLSI, MAGIC Layout, NGSPICE, Verilog, FPGA

I. INTRODUCTION

Carry Look-Ahead (CLA) adders provide faster addition by pre-computing carry signals for each bit position. This project entails the complete design flow of a 4-bit CLA adder, including schematic design, layout, simulations, and FPGA implementation.

II. DESIGN OVERVIEW

The proposed 4-bit CLA adder is based on static CMOS logic for larger gates and a novel architecture for AND and XOR gates. The design comprises propagate-generate logic and sum blocks. Transistor sizing is optimized to balance performance and area, with the ratio $W_p/W_n = 20\lambda/10\lambda$, where $\lambda = 0.09 \mu\text{m}$. Each sum output is designed to drive an inverter, sized according to project specifications. The design captures input bits just before the rising edge of the clock and outputs the computed result at the following rising edge.

III. DESIGN SPECIFICATIONS AND METHODOLOGY

A. Adder Architecture

The 4-bit carry look-ahead (CLA) adder is structured with separate Propagate and Generate (P/G) blocks and a carry look-ahead (CLA) unit. Each sum bit is computed using XOR logic with the calculated carry bit from the CLA. This adder architecture optimizes speed and area based on improved propagation delay from novel AND and XOR gate designs, as discussed in [1].

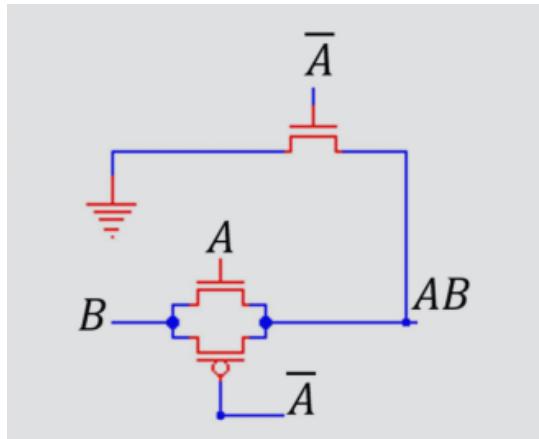


Fig. 1: AND Novel Structure

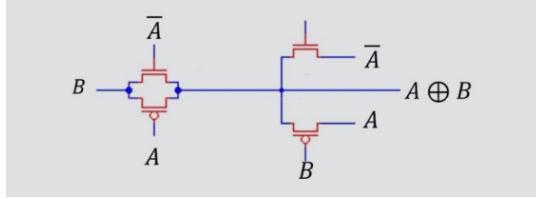


Fig. 2: XOR Novel Structure

B. D Flip-Flop Design

The D flip-flop (DFF) in this design is based on True Single-Phase Clock (TSPC) logic, utilizing only 11 transistors. This positive-edge triggered TSPC DFF is inspired by Razavi's work on TSPC flip-flops [3]. The minimized transistor count optimizes power efficiency while maintaining reliable performance, making it a suitable choice for high-speed applications in the CLA adder. The design which is in the final circuit is from [4].

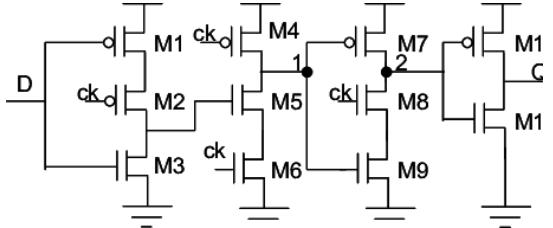


Fig. 3: D Flip Flop using 11-T

C. MAGIC Layout

The MAGIC layout editor is used to create and verify layouts for each circuit block. Post-layout extraction includes parasitic capacitances, enhancing accuracy between simulated and real-world performance. Layouts undergo DRC and LVS checks to ensure design consistency and correctness.

D. Gate and Module Structure

Each gate design leverages specific optimizations to improve modularity and performance. The project directory includes files for AND, OR, XOR, and DFF gates, as well as the complete CLA adder circuit. The AND and XOR gates use modified designs for better speed and reduced area based on insights from [1].

- **AND Gate Variants:** and3.cir, and3modular.cir, and4.cir, and4modular.cir, and5.cir, and5modular.cir, and.cir, andmodular.cir, and_with_abar.cir
- **OR Gate Variants:** or3.cir, or3modular.cir, or4.cir, or4modular.cir, or5.cir, or5modular.cir, or.cir, ormodular.cir
- **XOR Gate Variants:** xor.cir, xormodular.cir, xor_with_abar.cir
- **CLA and Final Circuit:** cla.cir, clamodular.cir, cla_withabar.cir, finalcir.cir

- D **Flip-Flop:** dffmodular.cir, dfilpflop.cir, dflipflopusingnands.cir

1) *Novel AND and XOR Gate Architecture:* In this architecture, I used the AND and XOR Gate designs from [1], which employ a mix of CPTL and CMOS Static logic to improve delay and area. The novel structures of the AND and XOR gates are illustrated in Figures 1 and 2, respectively.

The sizing is followed by using W_n equal to 20λ and W_p as 40λ while minimum transistor length is $2 * \lambda$.

They are a mix of PTL and CMOS Static style logic.

2) *Novel D Flip-Flop Architecture:* Using designs from [2] and [3], this architecture is a TSPC-based 12-transistor positive edge-triggered flip-flop. The specific D flip-flop structure with 11 transistors, as shown in Figure 3, optimizes power efficiency while maintaining reliable performance.

3) *Static CMOS Logic Gates:* The remaining gates in this design are based on static CMOS logic. Static CMOS ensures low leakage currents and reliable operation, making it a suitable choice for OR and basic logic gates in the CLA structure.

They are based on the basis of static CMOS Inverter with $\frac{2}{1}$ ratio. So the sizing is done accordingly to match the equal rise and fall time.

IV. SIMULATION RESULTS

A. NGSPICE Simulations

Individual blocks and the integrated CLA adder are verified using NGSPICE. Timing metrics, including setup time, hold time, and clock-to-Q delay of the flip-flop, are documented. Below are simulation results for each gate:

- 1) *AND Gate:*

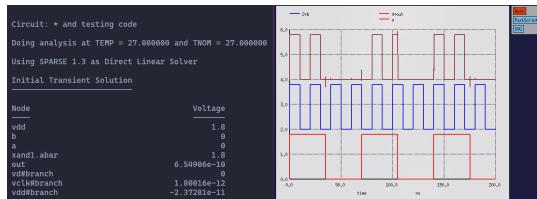


Fig. 4: AND Gate Simulation Results

Power Consumption: 42.67 pW

- 2) *AND3 Gate:*



Fig. 5: AND3 Gate Simulation Results

Power Consumption: 48.42 pW

- 3) *AND4 Gate:*

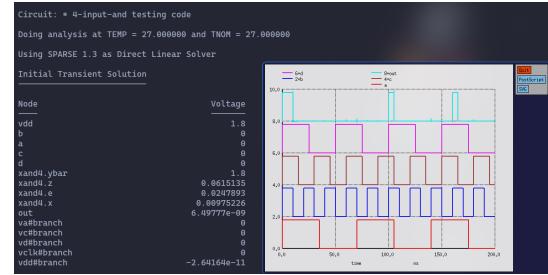


Fig. 6: AND4 Gate Simulation Results

Power Consumption: 47.52 pW

- 4) *AND5 Gate:*

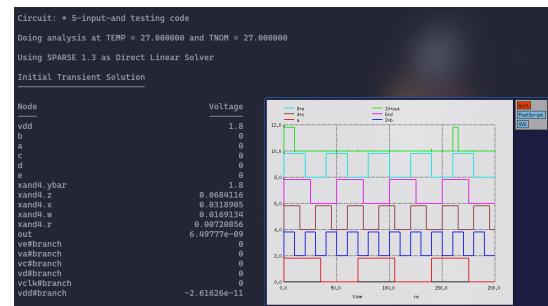


Fig. 7: AND5 Gate Simulation Results

Power Consumption: 46.98 pW

- 5) *OR Gate:*

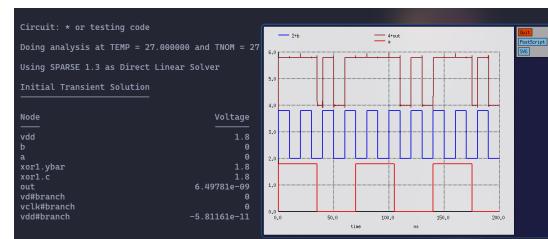


Fig. 8: OR Gate Simulation Results

Power Consumption: 104.58 pW

- 6) *OR3 Gate:*



Fig. 9: OR3 Gate Simulation Results

Power Consumption: 140.76 pW

7) OR4 Gate:

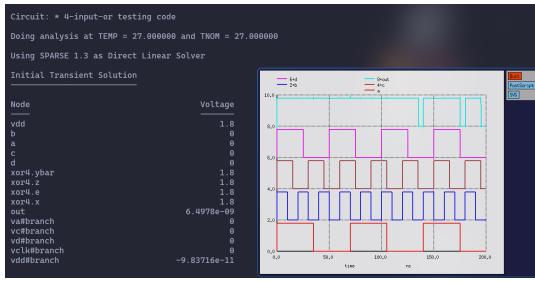


Fig. 10: OR4 Gate Simulation Results

Power Consumption: 176.94 pW

8) OR5 Gate:

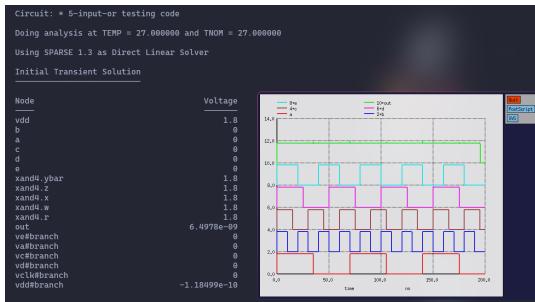


Fig. 11: OR5 Gate Simulation Results

Power Consumption: 212.24 pW

9) XOR Gate:

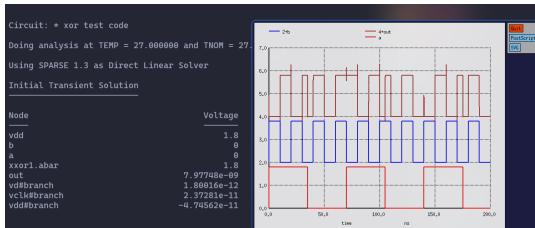


Fig. 12: XOR Gate Simulation Results

Power Consumption: 85.32 pW

10) D Flip Flop:

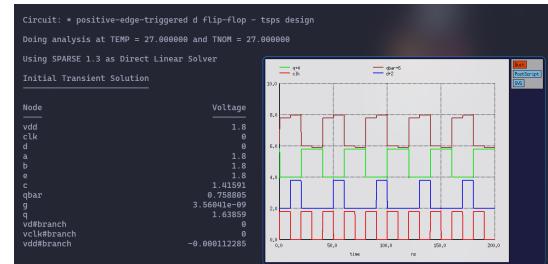


Fig. 13: DFF Gate Simulation Results

Power Consumption: 202.113 uW

B. D Flip-Flop Timing Analysis

The timing characteristics of the designed 11-transistor TSPC D flip-flop were analyzed using NGSPICE simulations. The following parameters were extracted:

- **Setup Time (t_{su}):** The minimum time the data input (D) must be stable before the rising clock edge for reliable data capture. Our simulations determined a setup time of **35 ps**. This value ensures robust data latching even with variations in process, voltage, and temperature. The simulation waveform demonstrating this is shown in Figure 14. After this around 40-50ps we observe that changes are getting reflected on the output, giving us setup time around 35ps.
- **Hold Time (t_h):** The minimum time the data input (D) must remain stable after the rising clock edge to prevent metastability. Our simulations yielded a hold time of **0.5 ps**. This value ensures the flip-flop's stable operation without data corruption. The simulation waveform illustrating this is presented in Figure 15. We know that TSPC are excellent at having considerably low setup and hold time, as we see in 13 that even after getting low in the non-existent time of 0.5 ps , it still comes back , this is due to how the logic in TSPC is structured.
- **Clock-to-Q Delay (t_{CQ}):** The delay between the rising clock edge and the stable output (Q). Our NGSPICE simulations measured a clock-to-Q delay of **0.5 ps**. This parameter is crucial for determining the maximum operational frequency of the flip-flop and the overall CLA adder. The simulation confirming this is depicted in Figure 16. Calculated using:

```
.measure tran tclktoq TRIG v(clk)
+VAL='SUPPLY/2' RISE=1 TARG v(q)
+VAL='SUPPLY/2' RISE=1
```

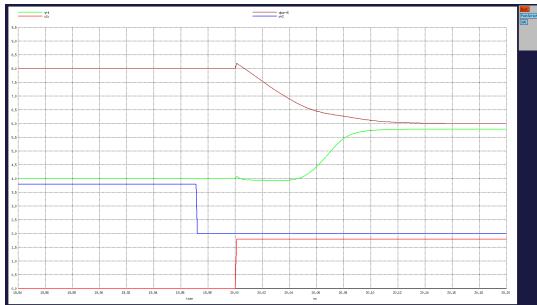


Fig. 14: NGSPICE simulation illustrating the setup time of the D flip-flop.

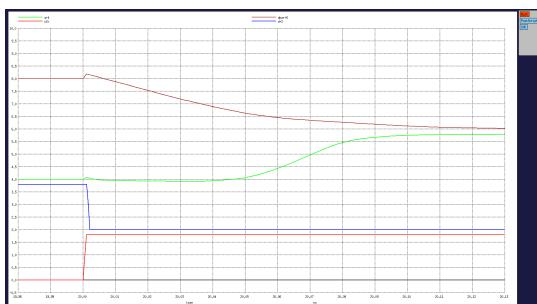


Fig. 15: NGSPICE simulation illustrating the hold time of the D flip-flop.

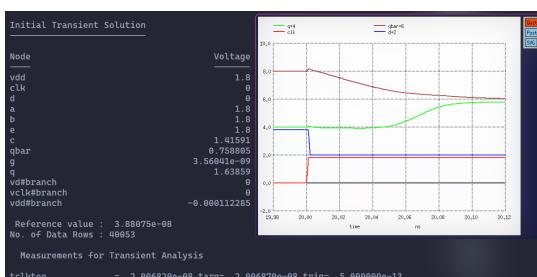


Fig. 16: NGSPICE simulation illustrating the clock-to-Q delay of the D flip-flop.

C. MAGIC Layouts

The MAGIC layout editor is used to design and verify the layout of each circuit block. The layout files are verified with DRC and LVS checks to ensure design consistency and correctness. The layout files are provided in the project directory ‘./magic’ for each block and the complete CLA adder.

1) AND Gate Layout:

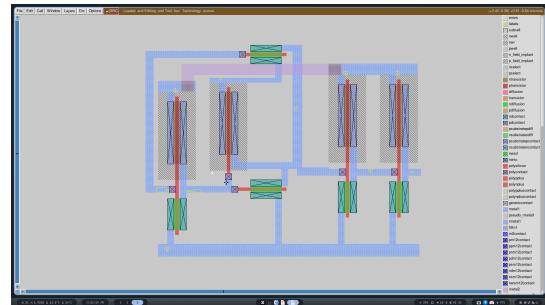


Fig. 17: AND Gate MAGIC Layout

2) AND3 Gate Layout:

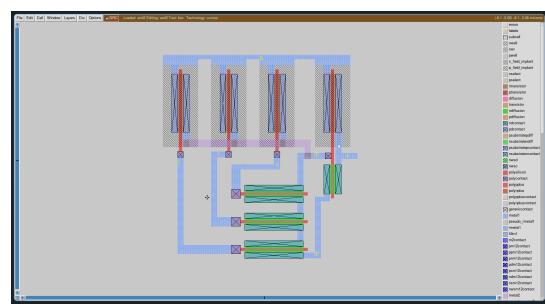


Fig. 18: AND3 Gate MAGIC Layout

3) AND4 Gate Layout:

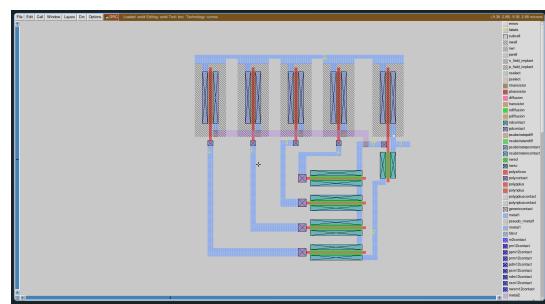


Fig. 19: AND4 Gate MAGIC Layout

4) AND5 Gate Layout:

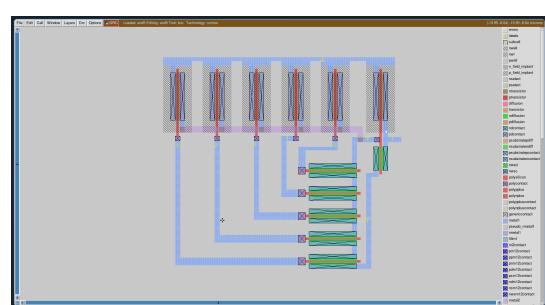


Fig. 20: AND5 Gate MAGIC Layout

5) OR Gate Layout:

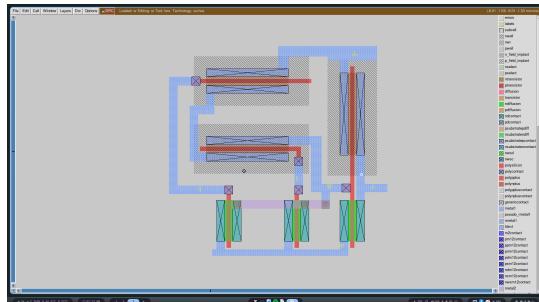


Fig. 21: OR Gate MAGIC Layout

6) OR3 Gate Layout:

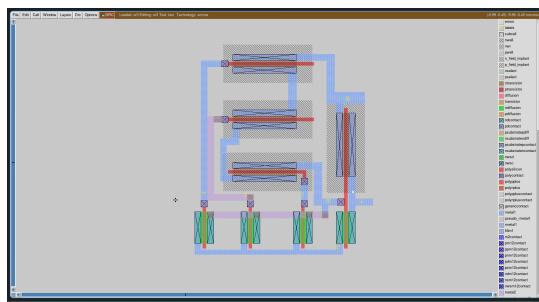


Fig. 22: OR3 Gate MAGIC Layout

7) OR4 Gate Layout:

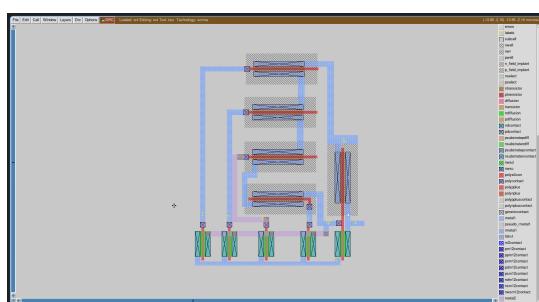


Fig. 23: OR4 Gate MAGIC Layout

8) OR5 Gate Layout:

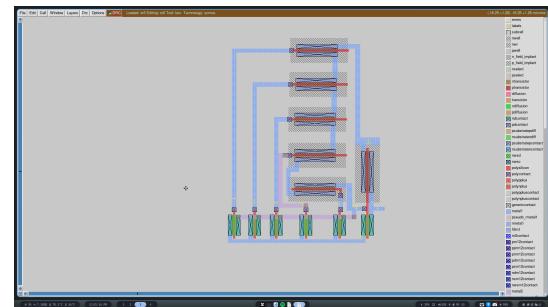


Fig. 24: OR5 Gate MAGIC Layout

9) XOR Gate Layout:

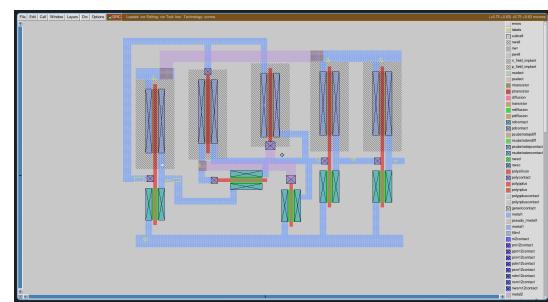


Fig. 25: XOR Gate MAGIC Layout

10) D Flip-Flop Layout:

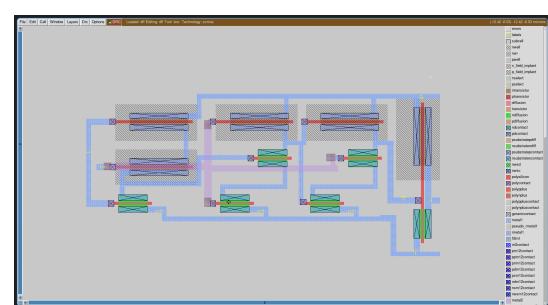


Fig. 26: D Flip-Flop MAGIC Layout

D. Post-Layout Simulation

Post-layout simulations with extracted netlists show the different gates functionality and power consumption.

1) AND Gate Post-Layout Simulation:

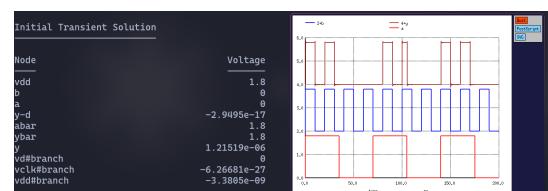


Fig. 27: AND Gate Post-Layout Simulation

Power Consumption: 6.084 nW

2) AND3 Gate Post-Layout Simulation:



Fig. 28: AND3 Gate Post-Layout Simulation

Power Consumption: 6.012 nW

3) AND4 Gate Post-Layout Simulation:



Fig. 29: AND4 Gate Post-Layout Simulation

Power Consumption: 6.013 nW

4) AND5 Gate Post-Layout Simulation:

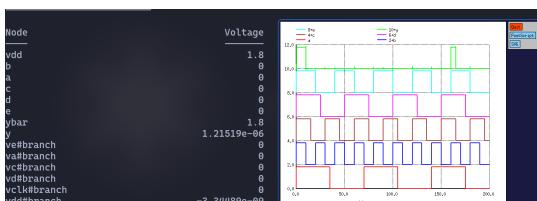


Fig. 30: AND5 Gate Post-Layout Simulation

Power Consumption: 6.019 nW

5) OR Gate Post-Layout Simulation:

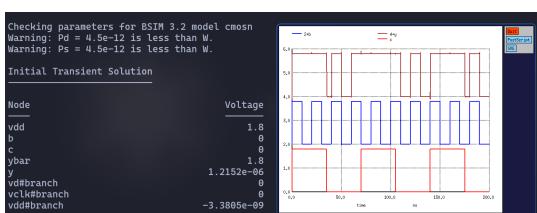


Fig. 31: OR Gate Post-Layout Simulation

Power Consumption: 6.084 nW

6) OR3 Gate Post-Layout Simulation:

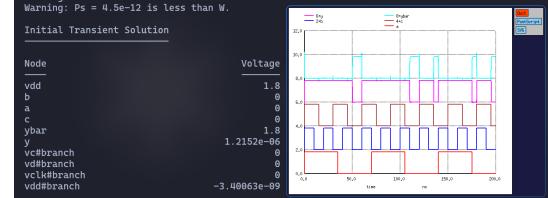


Fig. 32: OR3 Gate Post-Layout Simulation

Power Consumption: 6.12 nW

7) OR4 Gate Post-Layout Simulation:

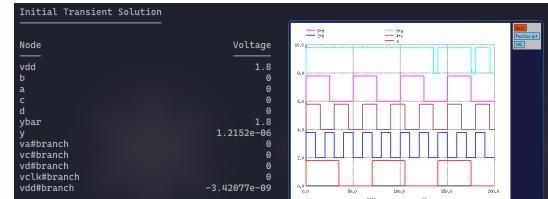


Fig. 33: OR4 Gate Post-Layout Simulation

Power Consumption: 6.15 nW

8) OR5 Gate Post-Layout Simulation:

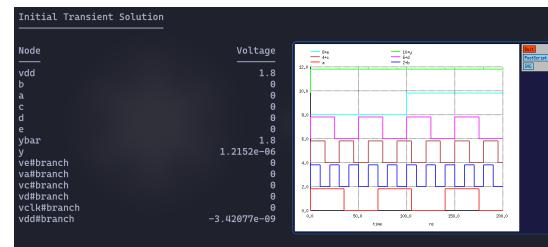


Fig. 34: OR5 Gate Post-Layout Simulation

Power Consumption: 6.16 nW

9) XOR Gate Post-Layout Simulation:

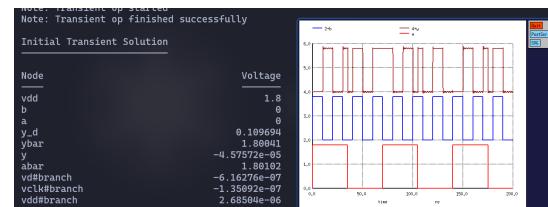


Fig. 35: XOR Gate Post-Layout Simulation

Power Consumption: 6.23 nW

10) D Flip-Flop Post-Layout Simulation:

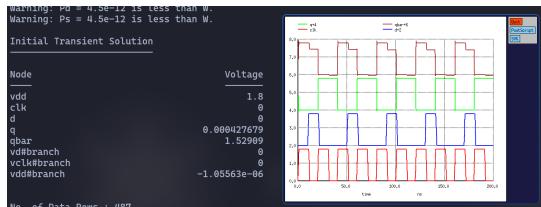


Fig. 36: D Flip-Flop Post-Layout Simulation

Power Consumption: 1.89 μW

V. FLOOR PLANNING AND STICK DIAGRAMS

1) *Vertical and Horizontal Pitches:* As we can see from the following picture that it's Horizontal pitch is 127.5 μm and Vertical pitch is 109 μm .

It is on the top right corner of the picture.

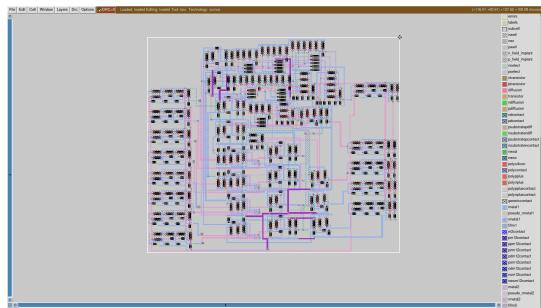


Fig. 37: Vertical and Horizontal Pitches

2) *Stick Diagrams:*

a) *AND Gate Stick Diagram:*

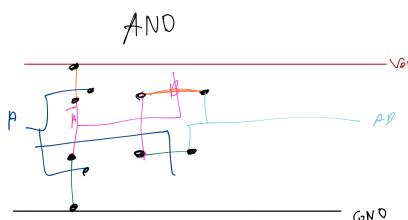


Fig. 38: AND Gate Stick Diagram

b) *AND3 Gate Stick Diagram:*

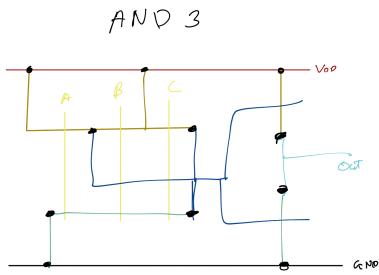


Fig. 39: AND3 Gate Stick Diagram

c) *AND4 Gate Stick Diagram:*

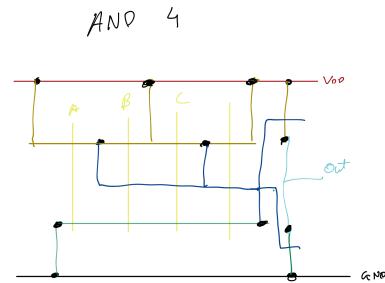


Fig. 40: AND4 Gate Stick Diagram

d) *AND5 Gate Stick Diagram:*

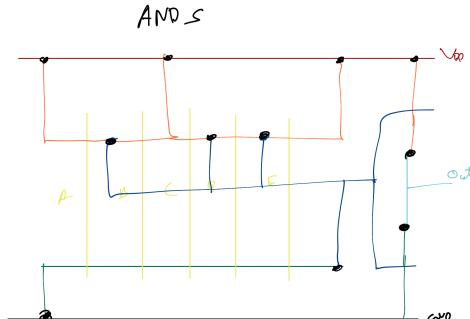


Fig. 41: AND5 Gate Stick Diagram

e) *OR Gate Stick Diagram:*

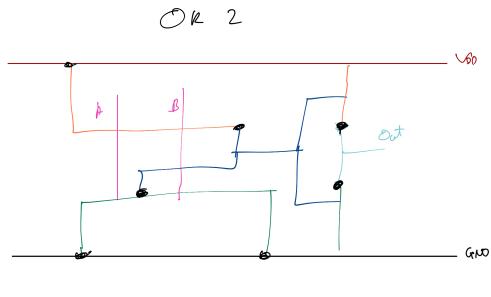


Fig. 42: OR Gate Stick Diagram

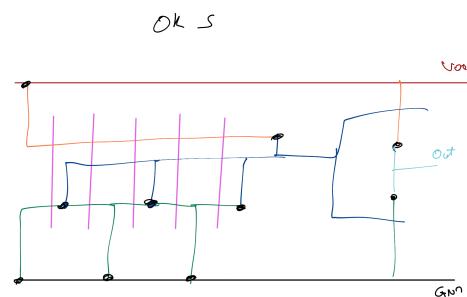


Fig. 45: OR5 Gate Stick Diagram

f) OR3 Gate Stick Diagram:

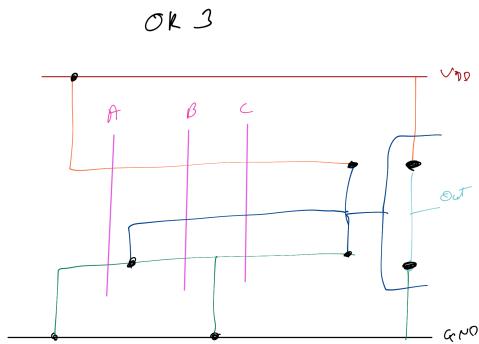


Fig. 43: OR3 Gate Stick Diagram

i) XOR Gate Stick Diagram:

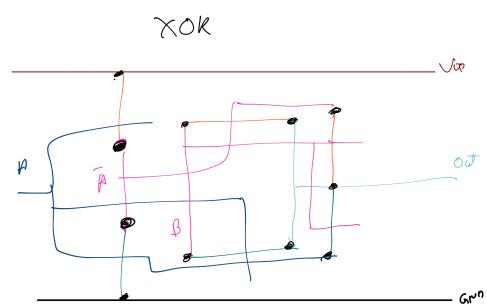


Fig. 46: XOR Gate Stick Diagram

g) OR4 Gate Stick Diagram:

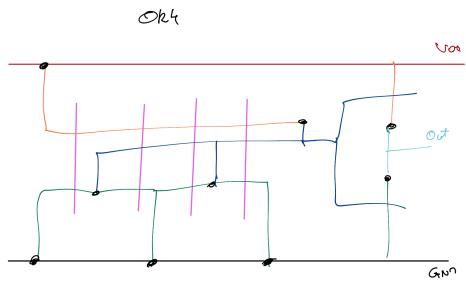


Fig. 44: OR4 Gate Stick Diagram

3) Floor Plan:

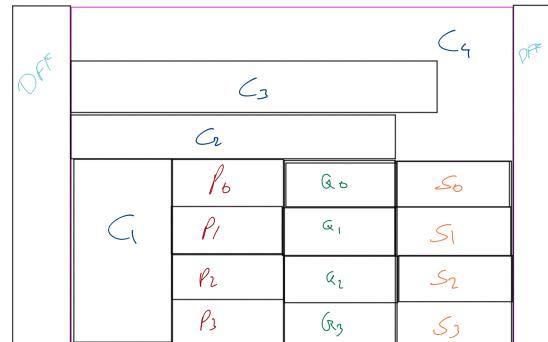


Fig. 47: Floor Plan

h) OR5 Gate Stick Diagram:

VI. FINAL CIRCUIT SIMULATION AND DELAY ANALYSIS

The final CLA adder circuit is simulated to evaluate the overall performance metrics, including delay and power consumption. The simulation results are analyzed to determine the maximum clock speed and overall efficiency of the 4-bit CLA adder.

The following simulations pictures will be comparing the final circuit in the pre-layout and post-layout simulations.

The delay analysis is done by measuring when input changes from 0000 1111 and Cin becomes from 0 to 1 for the rise time as this is the case where maximum transistors are switching and for the fall time when input changes from 1111 0000 and Cin becomes from 1 to 0. The delay is calculated by measuring the time difference between the input and output. The delay is calculated using the following command:

```
.measure tran trise TRIG v(in)
+VAL='SUPPLY/2' RISE=1 TARG
+v(out) VAL='SUPPLY/2' RISE=1
.measure tran tfall TRIG v(in)
+VAL='SUPPLY/2' FALL=1 TARG
+v(out) VAL='SUPPLY/2' FALL=1
```

1) Simulation Video: This whole synthesis from Extracting magic to running it on ngspice has been recorded and uploaded on the link [here](#). It shows that the circuit has 0 DRC and no warnings and the circuit functionality is correct.

2) MAGIC Layout of CLA with DFF:

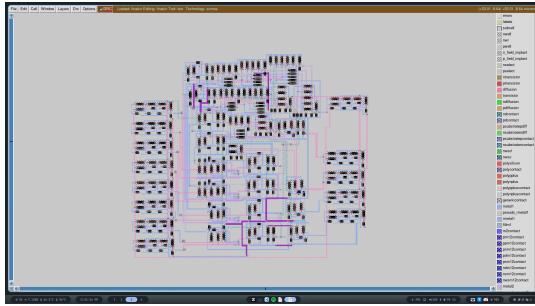


Fig. 48: CLA Adder MAGIC Layout

3) MAGIC Layout of CLA with Load Capacitance:

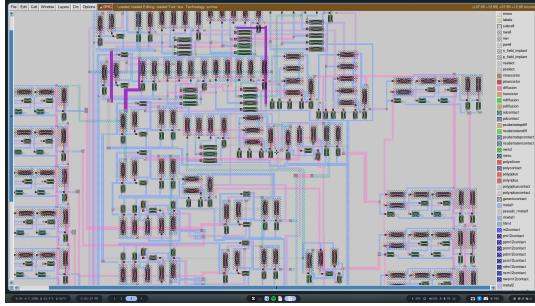


Fig. 49: Load Capacitance with CLA

4) Delay Results:

```
Measurements for Transient Analysis
delay_c4_fall = 1.736968e-10 targ= 1.018870e-08 trig= 1.001500e-08
delay_c4_rise = 2.234143e-10 targ= 2.284143e-10 trig= 5.000000e-12
delay_s0_fall = 9.184277e-12 targ= 1.002419e-08 trig= 1.001500e-08
delay_s0_rise = 7.782191e-12 targ= 1.278219e-11 trig= 5.000000e-12
delay_s1_fall = 3.195133e-10 targ= 1.033099e-08 trig= 1.001500e-08
delay_s1_rise = 3.098190e-10 targ= 3.075190e-08 trig= 5.000000e-12
delay_s2_fall = 5.069666e-10 targ= 1.021569e-08 trig= 1.001500e-08
delay_s2_rise = 2.561268e-10 targ= 2.631268e-10 trig= 5.000000e-12
delay_s3_fall = 2.396742e-10 targ= 1.025462e-08 trig= 1.001500e-08
delay_s3_rise = 2.627300e-10 targ= 2.677300e-10 trig= 5.000000e-12
```

Fig. 50: Delay Pre-Layout Simulation

```
Measurements for Transient Analysis
delay_c4_fall = 3.585798e-10 targ= 1.036558e-08 trig= 1.001500e-08
delay_c4_rise = 1.832705e-10 targ= 1.882705e-10 trig= 5.000000e-12
delay_s0_fall = 9.971784e-11 targ= 1.011472e-08 trig= 1.001500e-08
delay_s0_rise = 6.321581e-11 targ= 6.821581e-11 trig= 5.000000e-12
delay_s1_fall = 2.740186e-10 targ= 1.028902e-08 trig= 1.001500e-08
delay_s1_rise = 2.420385e-10 targ= 2.470385e-10 trig= 5.000000e-12
delay_s2_fall = 5.181022e-11 targ= 1.029431e-08 trig= 1.001500e-08
delay_s2_rise = 2.518102e-11 targ= 2.568132e-10 trig= 5.000000e-12
delay_s3_fall = 2.793138e-10 targ= 1.029431e-08 trig= 1.001500e-08
delay_s3_rise = 1.729257e-10 targ= 1.779257e-10 trig= 5.000000e-12
```

Fig. 51: Delay Post-Layout Simulation

5) Carry Look-Ahead Adder Simulation Results:

a) Generate:

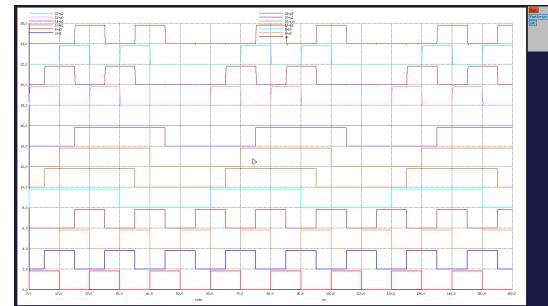


Fig. 52: Generate Pre-Layout Results

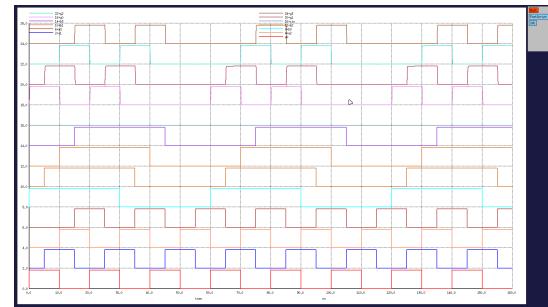


Fig. 53: Generate Post-Layout Results

b) Carry:

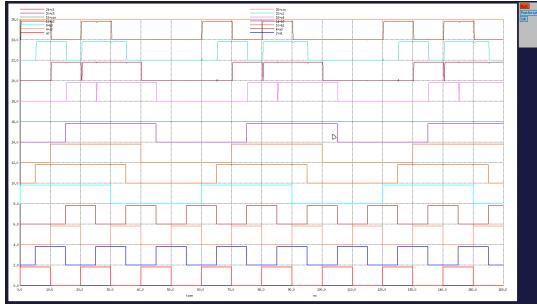


Fig. 54: Carry Pre-Layout Results

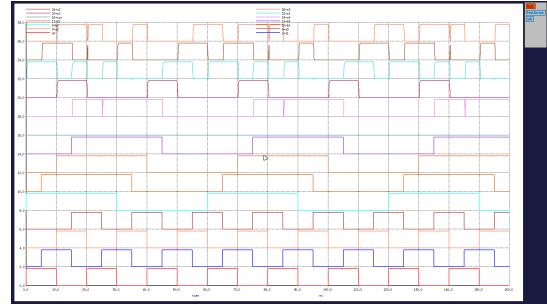


Fig. 58: Sum Generate Pre-Layout Results

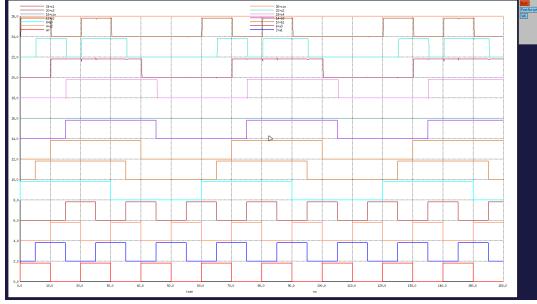


Fig. 55: Carry Post-Layout Results

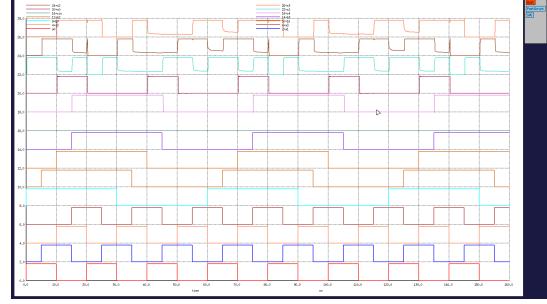


Fig. 59: Sum Generate Post-Layout Results

c) Propagate:

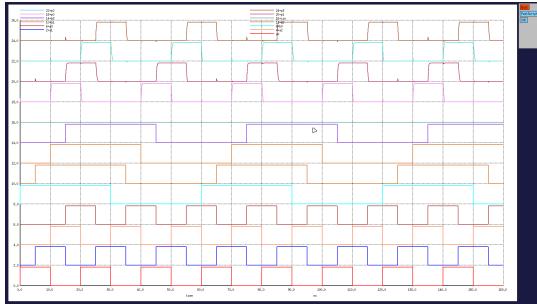


Fig. 56: Propagate Pre-Layout Results

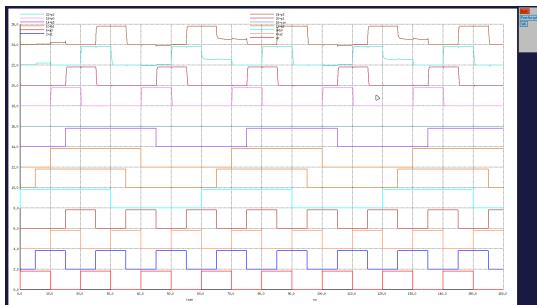


Fig. 57: Propagate Post-Layout Results

d) Sum:

VII. SIMULATION RESULTS COMPARISON

TABLE I: Comparison of Schematic and Post-Layout Simulation Results

Measurement	Schematic (s)	Post-Layout (s)
delay_c4_fall	1.723277e-10	3.505798e-10
delay_c4_rise	2.215140e-10	1.832705e-10
delay_s0_fall	9.229427e-12	9.971784e-11
delay_s0_rise	7.818817e-12	6.321581e-11
delay_s1_fall	3.304259e-10	2.740186e-10
delay_s1_rise	2.993190e-10	2.420385e-10
delay_s2_fall	2.503602e-10	3.136325e-10
delay_s2_rise	2.550430e-10	2.518182e-10
delay_s3_fall	2.385159e-10	2.793138e-10
delay_s3_rise	2.600989e-10	1.729257e-10

VIII. DELAY AND MAXIMUM CLOCK FREQUENCY

The delay of the CLA-adder is determined by the maximum delay observed in the post-layout simulation results. The maximum delay is:

- **delay_c4_fall:** 3.505798e-10 s

The maximum clock frequency at which the circuit operates reliably can be calculated as the inverse of the maximum delay:

$$f_{max} = \frac{1}{3.505798 \times 10^{-10}} \approx 2.85 \text{ GHz}$$

Thus, the maximum clock frequency is approximately 2.85 GHz.

IX. VERILOG IMPLEMENTATION

The Verilog HDL description is synthesized for FPGA, and functionality is verified through oscilloscope waveforms. The hardware results validate the design and confirm the correct

operation of the CLA adder on FPGA. These are corrected and checked and can be viewed on [this link](#).

1) iVerilog Output:

```
Time = 0, clk = 0, A_in = 0000, B_in = 0000, Cin = 0, S_out = xxxx, Clk_out = x
Time = 5, clk = 1, A_in = 0000, B_in = 0000, Cin = 0, S_out = xxxx, Clk_out = x
Time = 10, clk = 0, A_in = 0011, B_in = 0101, Cin = 0, S_out = xxxx, Clk_out = x
Time = 15, clk = 1, A_in = 0011, B_in = 0101, Cin = 0, S_out = 0000, Clk_out = 0
Time = 20, clk = 0, A_in = 1111, B_in = 0001, Cin = 1, S_out = 0000, Clk_out = 0
Time = 25, clk = 1, A_in = 1111, B_in = 0001, Cin = 1, S_out = 1001, Clk_out = 0
Time = 30, clk = 0, A_in = 1010, B_in = 0101, Cin = 0, S_out = 0000, Clk_out = 0
Time = 35, clk = 1, A_in = 1010, B_in = 0101, Cin = 0, S_out = 0000, Clk_out = 1
Time = 40, clk = 0, A_in = 0100, B_in = 1001, Cin = 1, S_out = 0000, Clk_out = 1
Time = 45, clk = 1, A_in = 0100, B_in = 1001, Cin = 1, S_out = 0000, Clk_out = 1
Time = 50, clk = 0, A_in = 0001, B_in = 0010, Cin = 0, S_out = 0000, Clk_out = 1
Time = 55, clk = 1, A_in = 0001, B_in = 0010, Cin = 0, S_out = 1111, Clk_out = 0
Time = 60, clk = 0, A_in = 0010, B_in = 0011, Cin = 1, S_out = 1111, Clk_out = 0
Time = 65, clk = 1, A_in = 0010, B_in = 0011, Cin = 1, S_out = 0100, Clk_out = 0
Time = 70, clk = 0, A_in = 0100, B_in = 0100, Cin = 0, S_out = 0100, Clk_out = 0
Time = 75, clk = 1, A_in = 0100, B_in = 0100, Cin = 0, S_out = 0101, Clk_out = 0
Time = 80, clk = 0, A_in = 0101, B_in = 0101, Cin = 1, S_out = 0101, Clk_out = 0
Time = 85, clk = 1, A_in = 0101, B_in = 0101, Cin = 1, S_out = 1000, Clk_out = 0
Time = 90, clk = 0, A_in = 1111, B_in = 0000, Cin = 0, S_out = 1000, Clk_out = 0
Time = 95, clk = 1, A_in = 0111, B_in = 0000, Cin = 0, S_out = 1010, Clk_out = 0
Time = 100, clk = 0, A_in = 1000, B_in = 0011, Cin = 1, S_out = 1010, Clk_out = 0
Time = 105, clk = 1, A_in = 1000, B_in = 0011, Cin = 1, S_out = 1110, Clk_out = 0
Time = 110, clk = 0, A_in = 0001, B_in = 1001, Cin = 0, S_out = 1110, Clk_out = 0
Time = 115, clk = 1, A_in = 0001, B_in = 1000, Cin = 0, S_out = 1111, Clk_out = 0
Time = 120, clk = 0, A_in = 0011, B_in = 1001, Cin = 1, S_out = 1111, Clk_out = 0
Time = 125, clk = 1, A_in = 0011, B_in = 1001, Cin = 1, S_out = 0010, Clk_out = 1
Time = 130, clk = 0, A_in = 0100, B_in = 1100, Cin = 0, S_out = 0010, Clk_out = 1
Time = 135, clk = 0, A_in = 0100, B_in = 1100, Cin = 0, S_out = 0000, Clk_out = 1
Time = 140, clk = 1, A_in = 0101, B_in = 1011, Cin = 1, S_out = 0100, Clk_out = 1
Time = 145, clk = 0, A_in = 1101, B_in = 1011, Cin = 1, S_out = 0111, Clk_out = 1
Time = 150, clk = 0, A_in = 1100, B_in = 1100, Cin = 0, S_out = 0111, Clk_out = 1
Time = 155, clk = 1, A_in = 1100, B_in = 1100, Cin = 0, S_out = 1000, Clk_out = 1
Time = 160, clk = 0, A_in = 1101, B_in = 1001, Cin = 1, S_out = 1000, Clk_out = 1
Time = 165, clk = 1, A_in = 1101, B_in = 1101, Cin = 1, S_out = 1011, Clk_out = 1
Time = 170, clk = 0, A_in = 0000, B_in = 1110, Cin = 0, S_out = 1011, Clk_out = 1
Time = 175, clk = 1, A_in = 0000, B_in = 1110, Cin = 0, S_out = 1000, Clk_out = 1
Time = 180, clk = 0, A_in = 0001, B_in = 1111, Cin = 0, S_out = 1000, Clk_out = 1
Time = 185, clk = 1, A_in = 0001, B_in = 1111, Cin = 1, S_out = 1111, Clk_out = 0
Time = 190, clk = 0, A_in = 0010, B_in = 0000, Cin = 0, S_out = 0000, Clk_out = 0
Time = 195, clk = 1, A_in = 0010, B_in = 0000, Cin = 0, S_out = 0000, Clk_out = 1
Time = 200, clk = 0, A_in = 0011, B_in = 0001, Cin = 1, S_out = 0000, Clk_out = 1
Time = 205, clk = 1, A_in = 0011, B_in = 0001, Cin = 1, S_out = 0011, Clk_out = 1
Time = 210, clk = 0, A_in = 0000, B_in = 0000, Cin = 0, S_out = 0011, Clk_out = 0
Time = 215, clk = 1, A_in = 0000, B_in = 0000, Cin = 0, S_out = 0100, Clk_out = 0
```

Fig. 60: Verilog Output for CLA Adder with DFF

2) GTWAVE Output:

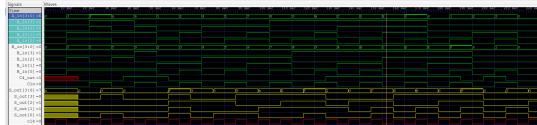


Fig. 61: GTWAVE Output for CLA Adder with DFF

3) *FPGA Outputs:* The following picture is about explaining where are input and output are connected to the FPGA board as constraints.



Fig. 62: FPGA Board Connections

Example of the output of the FPGA board is shown below.



Fig. 63: FPGA Board Output

4) *Oscilloscope Waveforms:* All of these are explained in detail in the video [here](#). We connected arduino to the analog pins and used serial plotter to view the waveforms. Some screenshots from the video of waveforms are shown below.

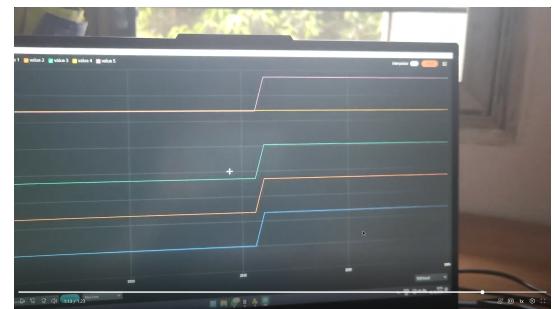


Fig. 64: Waveform 1

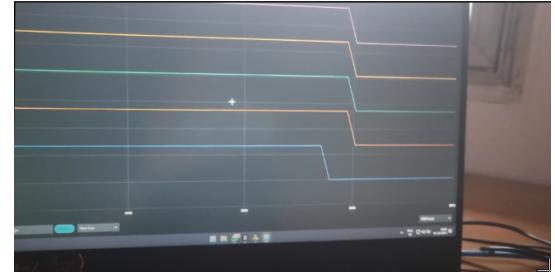


Fig. 65: Waveform 2

X. CONCLUSION

This project successfully demonstrates a 4-bit CLA adder design, verified with NGSPICE simulation, MAGIC Post Layout simulations and FPGA. Future work could involve exploring dynamic logic styles or scaling the design to higher bit widths.

XI. ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof. Abhishek Srivastava for his guidance and support throughout this project. I would also like to thank the TAs for their assistance and feedback during the project.

REFERENCES

- [1] M. Hasan, M. Islam, and H. M. Hasan, "Performance Improvement of 4-Bit Static CMOS Carry Look-Ahead Adder Using Modified Circuits for Carry Propagate and Generate Terms," *Science Journal of Circuits Systems and Signal Processing*, vol. 8, no. 2, pp. 32-38, Dec. 2019. [Online]. Available: https://www.researchgate.net/publication/338214910_Performance_Improvement_of_4-Bit_Static_CMOS_Carry_Look-Ahead_Adder_Using_Modified_Circuits_for_Carry_Propagate_and_Generate_Terms
- [2] S. Kumar and B. Singh, "A New Low Power 4T-Based Three Input Universal Gate for High Speed Applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 3, pp. 1450-1457, June 2018. [Online]. Available: https://www.researchgate.net/publication/325386038_A_New_Low_Power_4T-Based_Three_Input_Universal_Gate_for_High_Speed_Applications
- [3] B. Razavi, "A Circuit for All Seasons," in *Principles of Data Conversion System Design*, TSPC Logic, John Wiley & Sons, 2018.
- [4] M. Agarwal, B. Paul, M. Zhang, and S. Mitra, "NBTI-aware flip-flop characterization and design," in *Proceedings of the 45th Annual Design Automation Conference (DAC)*, 2008, pp. 560-563. DOI: 10.1145/1366110.1366121. [Online]. Available: <https://www.researchgate.net/publication/220904032>