Carry Look-Ahead Adder (CLA) Optimized for Speed and Energy

Razan Abdelrahman
Electrical and Computer Engineering Department
Birzeit University
Ramallah, Palestine
1200531@student.birzeit.edu

Masa Itmaizi
Electrical and Computer Engineering Department
Birzeit University
Ramallah, Palestine
1200814@student.birzeit.edu

Katya Kobari
Electrical and Computer Engineering Department
Birzeit University
Ramallah, Palestine
1201478@student.birzeit.edu

***Abstract*— This project explores the design and implementation of a 16-bit Carry Look-Ahead Adder (CLA) optimized for speed and energy efficiency. By employing Electric EDA and leveraging NMOS and PMOS transistors, the design enhances computational performance while minimizing power consumption and delay. Inspired by state-of-the-art research, the architecture uses uniform-size CLA modules and advanced process technologies to achieve a scalable, high-speed solution for modern digital systems. Simulations confirm significant improvements in speed and energy metrics, offering a promising direction for efficient arithmetic circuits in processors and embedded systems.**

***Keywords*— Carry Look-Ahead Adder, NMOS, PMOS, speed optimization, energy efficiency, Electric EDA.**

I. Introduction

In today's fast-changing world, having accurate and efficient digital systems is more important than ever. One of the key components in improving system performance is the Carry Look-Ahead Adder (CLA), which helps processors perform calculations faster. The CLA is designed to reduce delays caused by carry propagation, making it crucial for many applications such as image processing and machine learning. As technology advances, it is important to make the CLA more efficient by reducing power consumption and increasing processing speed. Various methods have been developed to achieve this, such as using advanced technologies and uniform modules. These improvements help reduce power use, speed up calculations, and make the system more efficient, leading to faster and more energy-saving digital systems [1].

II. Design Methodology.

A. *Circuit Description.*

1) *Overview of the CLA Circuit.*

The Carry Look-Ahead Adder (CLA) is a high-speed adder circuit designed to address the delay limitations inherent in Ripple-Carry Adders (RCA). Unlike RCAs, where carry propagation occurs sequentially, the CLA computes carry signals in parallel, significantly reducing propagation delay. This parallelism enables the CLA to handle large-bit-width additions with superior efficiency, making it an indispensable component in modern high-performance digital systems [2].

The CLA leverages Boolean logic to predict carry signals in advance, ensuring that carry computations do not cascade across bit positions. This approach optimizes addition time and provides faster operations, critical in applications requiring real-time data processingFinal Presentation.

2) *Logic and Operation.*

The CLA operation is based on two primary logic signals: Generate (G) and Propagate (P), defined as follows:

- Generate (G): Indicates when a carry is generated for a given bit.

$$G_i = A_i . B_i \quad (1)$$

- Propagate (P): Indicates when a carry is propagated to the next bit.

$$P_i = A_i + B_i \quad (2)$$

Using these signals, the carry-out C_{i+1} can be expressed as:

$$C_{i+1} = G_i + (P_i . C_i) \quad (3)$$

The sum S_i is computed using the XOR operation:

$$S_i = P_i \oplus C_i \quad (4)$$

This logical arrangement allows all carry signals to be determined simultaneously, eliminating the cascading delay of conventional adders and enabling high-speed operation.

3) *Use of NMOS/PMOS Transistors.*

The CLA is implemented using CMOS technology, which combines NMOS and PMOS transistors to construct logic gates such as AND, OR, and XOR. The key advantages of *CMOS technology include:*

- **Power Efficiency:** Complementary NMOS and PMOS transistors minimize static power dissipation.

- **High Speed:** The fast switching characteristics of CMOS ensure minimal propagation delays.
- **Reliability:** CMOS circuits exhibit high noise immunity and robust operation across varying environmental conditions.

Each subcomponent of the CLA, including the NAND gate, XOR gate, and full adder, is meticulously designed using CMOS principles, ensuring a balance between speed and power efficiency [3].

B. Schematic and Layout Design.

1) Block Diagram of the CLA.

The CLA is organized into key functional blocks:

- **Input Logic:** Captures the input operands and initial carry.
- **Generate (G) and Propagate (P) Generators:** Computes the generated and propagated signals for all bits.
- **Carry Generator:** Computes the carry signals in parallel for all stages, leveraging the G and P signals.
- **Sum Logic:** Determines the sum for each bit based on the propagate signal and the respective carry.

2) Subcomponents Used in Design.

The CLA circuit incorporates several critical subcomponents, each meticulously designed for functionality and efficiency:

- **CMOS Inverter:** A fundamental component used to generate the complement of input signals, essential in various logic functions. The CMOS inverter consists of a PMOS transistor connected to VDD and an NMOS transistor connected to GND, with their gates tied together to form the input. This complementary configuration offers: Power Efficiency, Fast Switching and Robustness.

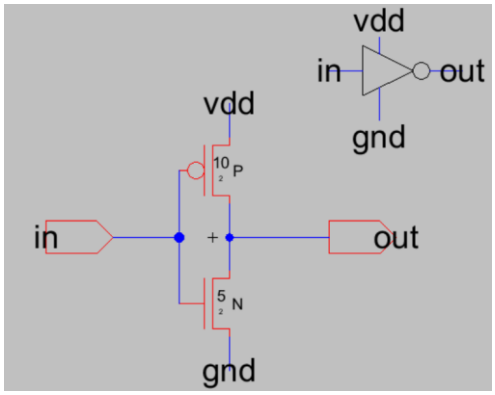

Figure 1 Inverter schematic.

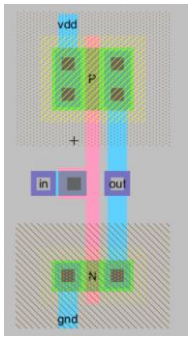

Figure 2 Inverter layout.

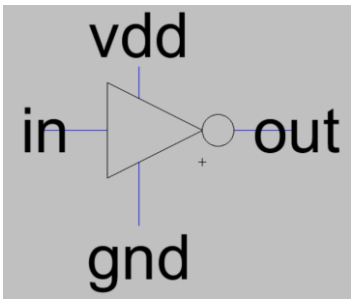

Figure 3 Inverter icon.

- **NAND Gate:** A versatile universal gate used extensively to implement AND and OR operations efficiently. It forms the basis of more complex logic circuits in the CLA design.

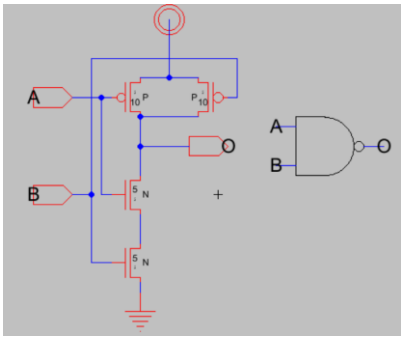

Figure 4 Nand gate schematic.

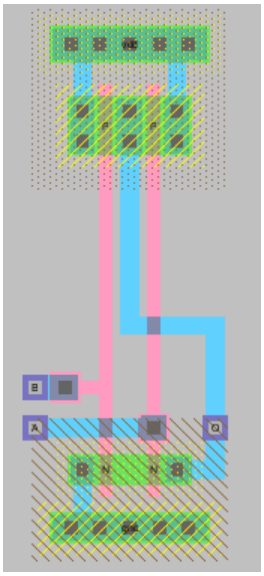

Figure 5 Nand gate layout.

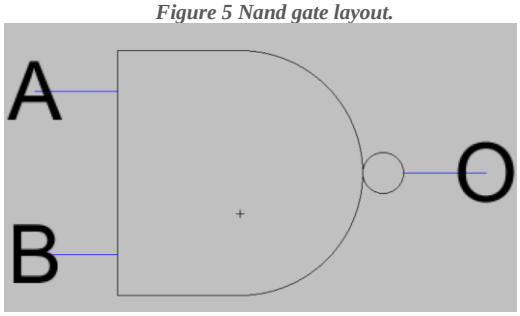

Figure 6 Nand icon.

- **XOR Gate:** A critical component used to compute the sum in the full adder. The XOR gate is implemented using CMOS technology, combining PMOS and NMOS transistors to achieve high-speed operation.

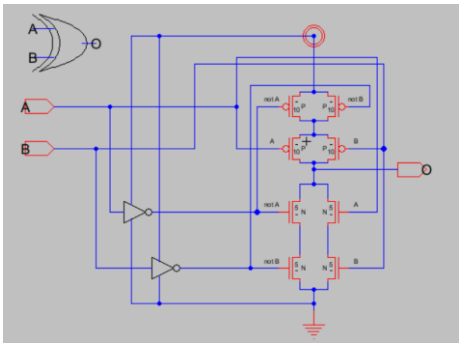

Figure 7 XOR schematic.

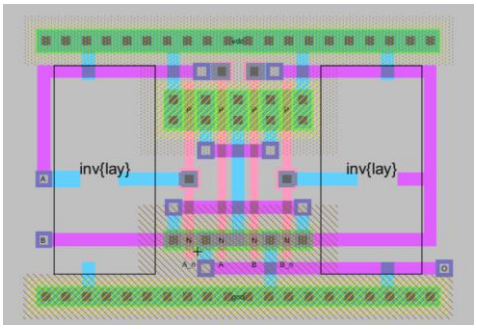

Figure 8 XOR layout.

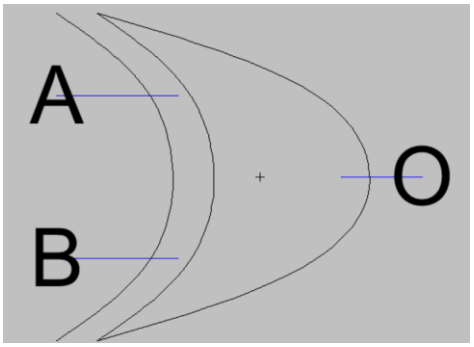

Figure 9 XOR icon.

- ***Full Adder:*** Each full adder in the CLA is designed using two XOR gates, NAND gates, and a CMOS inverter to compute the sum and carry for a single bit. It integrates the propagate (P), generate (G) and carry (C) logic seamlessly, forming the foundation for scalable CLA designs.

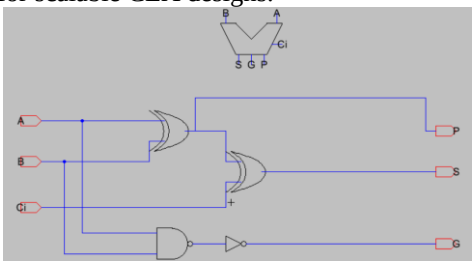

Figure 10 Full adder schematic.

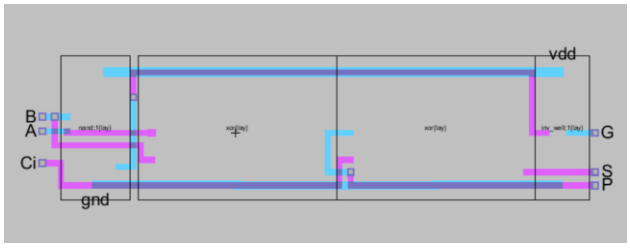

Figure 11 Full adder layout.

3) **Full Schematic and Layout.**

This section presents the hierarchical design progression of the CLA, starting with the basic 1-bit CLA and scaling up to the 16-bit CLA. Each stage includes the schematic, layout,

and icon representation for better visualization and understanding.

- **1-Bit CLA.**

  The 1-bit Carry Look-Ahead Adder (CLA) is the fundamental building block of larger CLA designs. It is designed to calculate the carry-out (C_1) for a single bit using the propagate (P_0) and generate (G_0) signals, along with the carry input (C_{an}). This allows for high-speed carry computation, reducing delays compared to traditional ripple-carry adders.

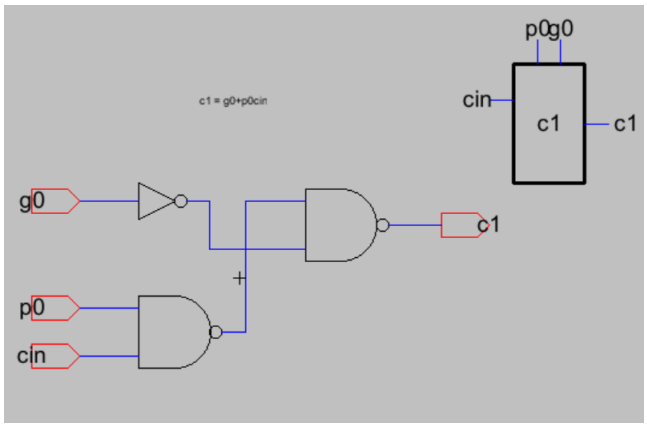

Figure 12 1-Bit CLA schamtic.

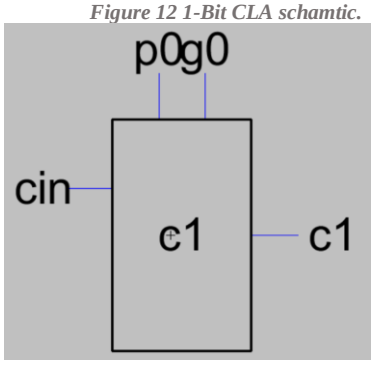

Figure 13 1-Bit CLA icon.

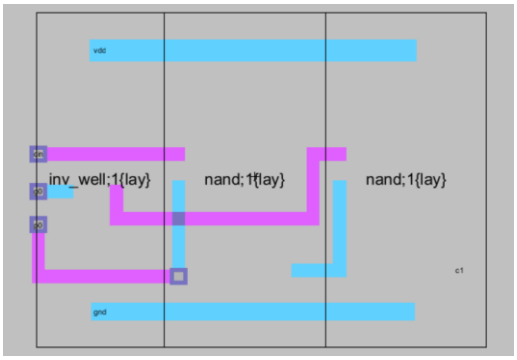

Figure 14 1-Bit CLA layout.

- **2-Bit CLA.**

  The 2-bit Carry Look-Ahead Adder (CLA) extends the functionality of the 1-bit CLA by handling carry computation for two bits simultaneously. This design reduces delay by computing both carries (C_1) and (C_2) in parallel.

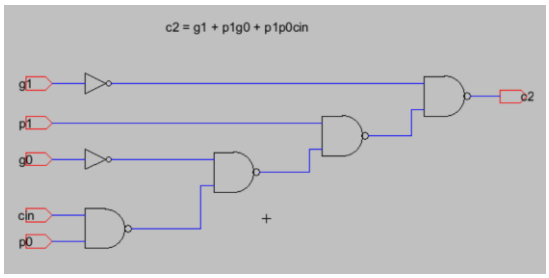

Figure 15 2-Bit CLA schematic.

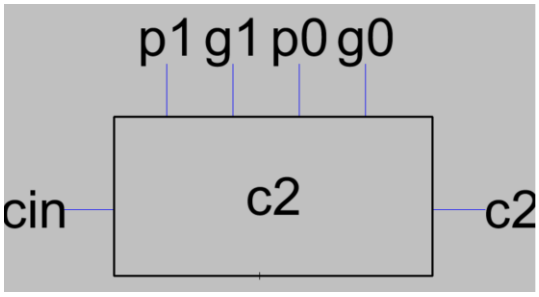

Figure 16 2-Bit CLA icon.

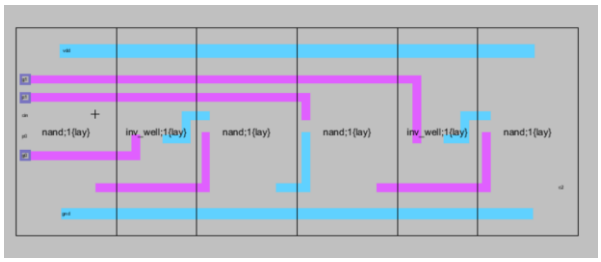

Figure 17 2-Bit CLA layout.

- **3-Bit CLA.**

  The 3-bit Carry Look-Ahead Adder (CLA) extends the carry computation logic to three bits, improving efficiency by parallelizing the carry generation for all bits. This design ensures that carry propagation delay is minimized.

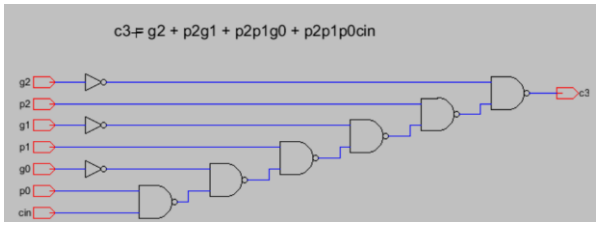

Figure 18 3-Bit CLA schematic.

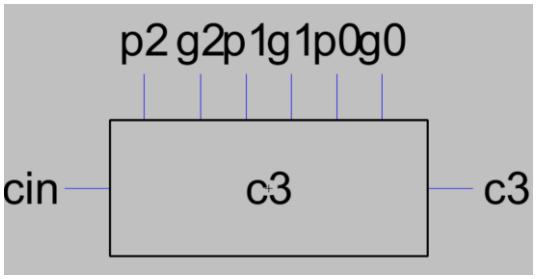

Figure 19 3-Bit CLA icon.

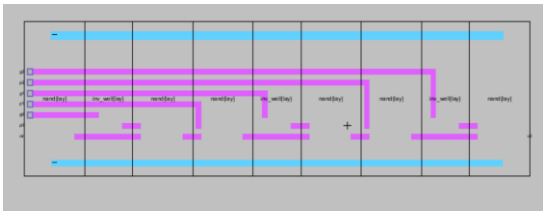

Figure 20 3-Bit CLA layout.

- **4-Bit CLA.**

  The 4-bit Carry Look-Ahead Adder (CLA) is a scalable design that calculates carries for four bits in parallel, significantly improving addition speed. It uses hierarchical carry generation logic and combines 1-bit, 2-bit, and 3-bit CLA blocks to compute carries efficiently.

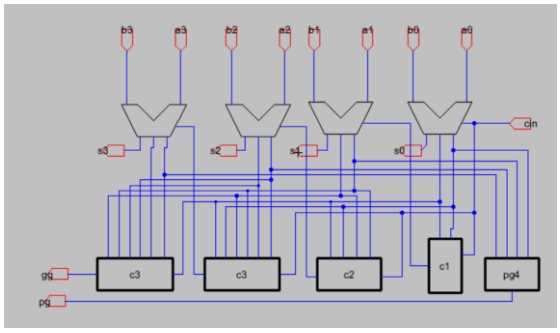

Figure 21 4-Bit CLA schematic.

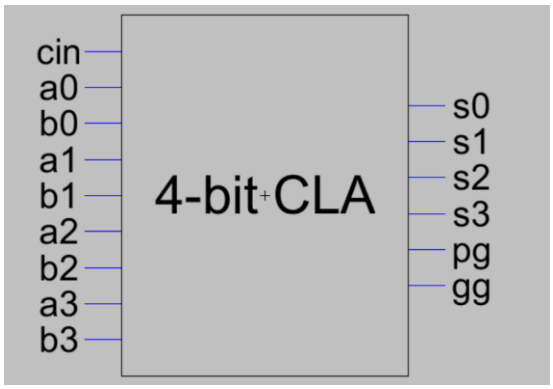

Figure 22 4-Bit CLA icon.

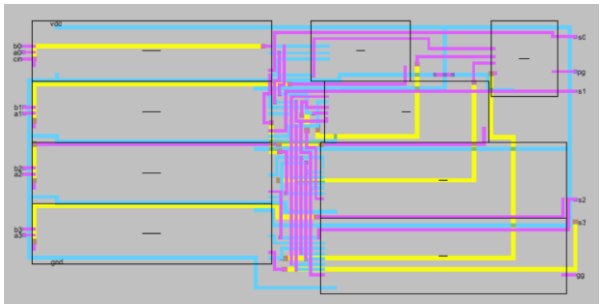

Figure 23 4-Bit CLA layout.

- **16-Bit CLA.**

  The 16-bit Carry Look-Ahead Adder (CLA) is a large-scale design constructed hierarchically using four 4-bit CLA blocks. It efficiently computes carries for 16 bits in parallel, leveraging the modularity of the 4-bit CLA units propagate higher-level carries across the 4-bit blocks.

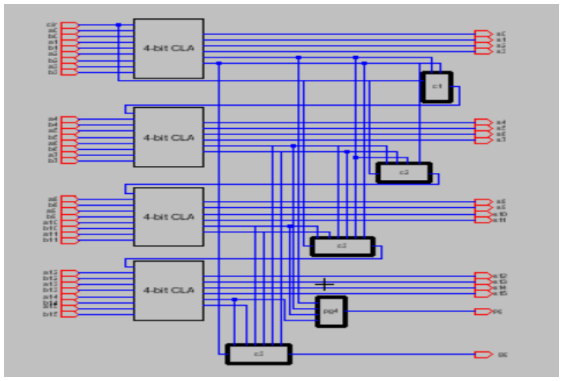

Figure 24 4-Bit CLA schematic.

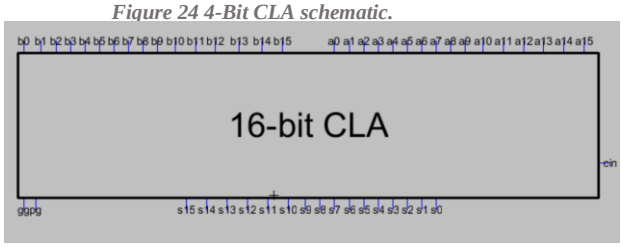

Figure 25 4-Bit CLA icon.

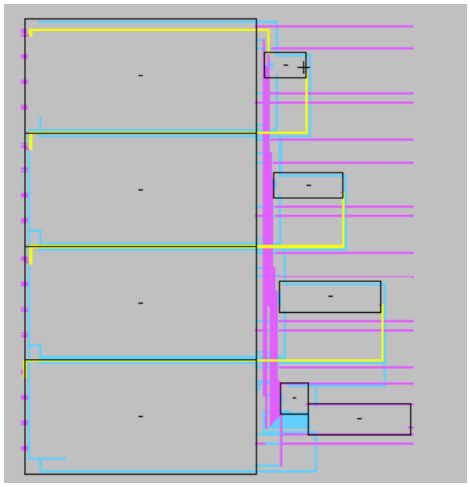

Figure 26 4-Bit CLA layout.

III. Simulation and Results

The simulation and results demonstrate that the implemented 16-bit Carry Look-Ahead Adder (CLA) effectively performs addition operations on 16-bit binary numbers with high efficiency. By leveraging fundamental logic gates such as NAND, NOT, and XOR, constructed using NMOS and PMOS transistors, the core components of the adder were successfully designed. The development of full adders and generate-propagate (pg4) circuits significantly enhanced the precision and performance of the overall system.

These components were systematically organized into a hierarchical structure, starting with smaller modules. The 1-bit CLA was validated as the foundational unit, which was scaled to create 2-bit, 3-bit, and subsequently 4-bit CLAs. The 4-bit CLA then served as a building block for the final 16-bit CLA. This modular approach optimized the design's scalability and performance.

Each stage of the design, including the 1-bit, 2-bit, 3-bit, 4-bit, and 16-bit CLA modules, underwent rigorous testing using both Verilog testbenches and LTSpice simulations. This comprehensive testing ensured accurate carry signal computation, minimized delays, and verified the robustness of the addition circuit. The result was a high-performance, reliable adder capable of fast and precise binary arithmetic operations.

A. *1-Bit Carry Look-Ahead Adder (CLA).*

The 1-bit Carry Look-Ahead Adder (CLA) was thoroughly tested to ensure its correctness and reliability. Simulation results confirmed that the circuit accurately computed the carry-out (C_1) for all possible input combinations, adhering to the equation:

$$C_1 = G_0 + P_0 * C_{in} \quad (5)$$

This ensured reliable carry propagation and demonstrated the circuit's capability in handling binary arithmetic operations. The design was evaluated through two approaches: transistor-level simulation in LTSpice, which showcased the dynamic behavior and verified the accurate computation of carry signals, and behavioral simulation in Verilog, which validated logical functionality using a comprehensive test bench. Both methods produced consistent and precise results, confirming the efficiency, robustness, and correctness of the 1-bit CLA. These tests demonstrated the circuit's efficiency and robustness in handling binary arithmetic operations.

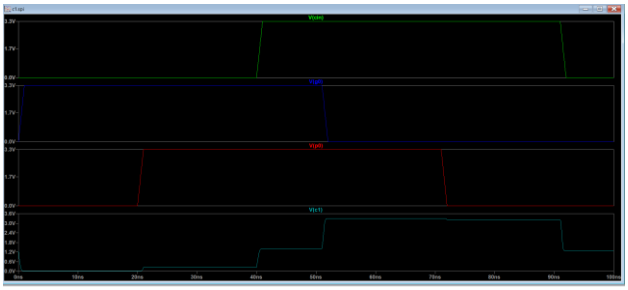

Figure 27 1-Bit CLA LTspice output.

Table 1 1-bit CLA test bench output.

Time	C_{in}	G_0	P_0	C_1
0	0	0	0	0
10	1	0	0	0
20	0	1	0	1
30	1	1	0	1
40	0	0	1	0
50	1	0	1	1
60	0	1	1	1
70	1	1	1	1

B. *2-Bit Carry Look-Ahead Adder (CLA).*

The 2-bit Carry Look-Ahead Adder (CLA) was simulated to verify its functionality and correctness. Verilog simulations validated the proper propagation of generate (g) and propagate (p) signals, as well as the accurate calculation of the carry-out (c2).

Additionally, LTSpice simulations provided detailed waveforms, illustrating the circuit's dynamic response to varying input signals (cin , g0 , g1 , p0 , p1). These simulations confirmed the theoretical design, highlighting accurate and reliable operation at both logical and transistor levels. The results demonstrated the circuit's efficiency in calculating carry signals while ensuring its robustness and correctness for multi-bit addition tasks.

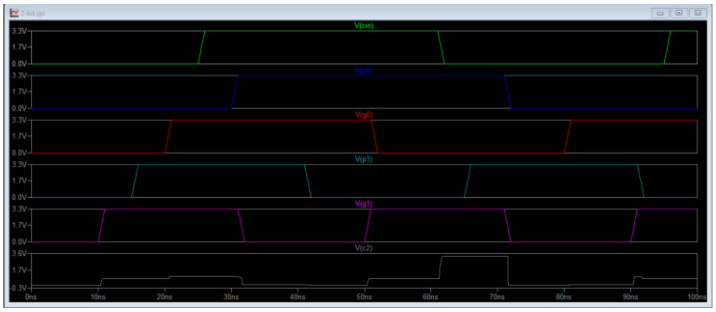

Figure 28 2-Bit CLA LTspice output.

Table 2 2-Bit CLA testbench output.

Time	Cin	G_0	G_1	P_0	P_1	C_2
0	0	0	0	0	0	0
10	1	0	0	0	0	0
20	0	1	0	0	0	0
30	0	0	1	0	0	1
40	1	0	0	1	0	0
50	0	0	1	0	1	1
60	1	1	1	1	1	1

C. *3-Bit Carry Look-Ahead Adder(CLA).*

The 3-bit Carry Look-Ahead Adder (CLA) computes the carry-out (c3) based on the equation:

$$C_3 = g_2 + p_2 * g_1 + p_2 * p_1 * g_0 + p_2 * p_1 * p_0 * C_{in} \quad (6)$$

LTSpice simulations generated detailed waveforms showcasing the correct transitions of generate (g), propagate (p), and carry signals (c3). These results confirmed the circuit's reliability and accuracy in handling three input bits efficiently, further validating its role in scalable addition designs.

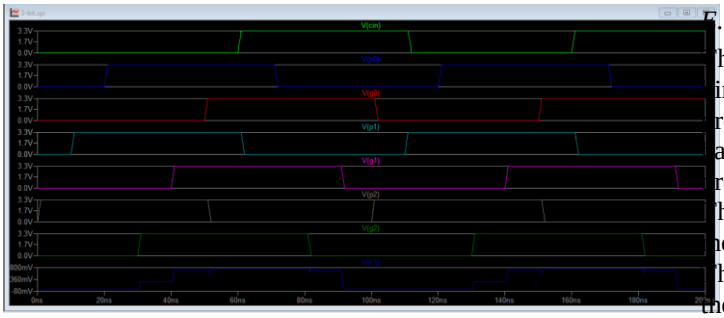

Figure 29 3-Bit CLA LTspice output.

D. 4-Bit Carry Look-Ahead Adder (CLA).

For the 4-bit carry-lookahead adder (CLA). In Verilog, a testbench was used to verify the correctness of the sum outputs (S3 to S0) and the carry-out (Cout) for various input combinations of A, B, and Cin. In LTSpice, we focused on analyzing the generate (GG) and propagate (PG) signals, which play a crucial role in carry computation. The inputs A (a0, a1, a2, a3), B (b0, b1, b2, b3), and Cin were plotted alongside the resulting GG and PG signals to verify their behavior. The GG signal indicates when a carry is directly generated within a block, while the PG signal shows when a carry can propagate through the block. These simulations confirm the correct operation of the 4-bit CLA and its ability to efficiently handle carry signals in multi-bit addition.

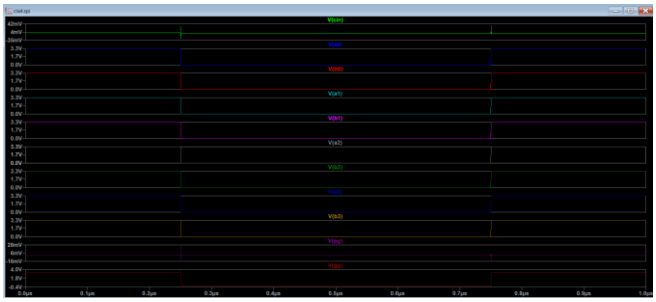

Figure 30 4-Bit CLA LTspice output.

Table 3 4-Bit CLA testbench output.

	A				B				Sum				Carry
Cin	A3	A2	A1	A0	B3	B2	B1	B0	S3	S2	S1	S0	Cout
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	1	1	0	1	1	0	0
0	0	1	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	1	0	1	0	1	1	0	1	0	0
0	0	1	1	0	0	1	1	0	1	1	0	0	0
0	0	1	1	1	0	1	1	1	1	1	1	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	1	0	0	1	0	0	1	0	1
0	1	0	1	0	1	0	1	0	0	1	0	0	1
0	1	0	1	1	1	0	1	1	0	1	1	0	1
0	1	1	0	0	1	1	0	0	1	0	0	0	1
0	1	1	0	1	1	1	0	1	1	0	1	0	1
0	1	1	1	0	1	1	1	0	1	1	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

E. 16-bit Carry Look-Ahead Adder (CLA).

The 16-bit Carry Look-Ahead Adder (CLA) adds two 16-bit binary numbers (A and B) and an initial carry-in (Cin). The process divides the inputs into four 4-bit segments, each handled by a 4-bit CLA module. Within each module, propagate (PG) and generate (GG) signals are computed. The PG signal indicates if a carry can propagate through the module, while GG shows if a carry is generated within it. These signals are used to calculate carries efficiently across the 16 bits, minimizing delay.

The final outputs include the 16-bit sum, PG, and GG signals for the entire adder. PG represents whether a carry can propagate through all 16 bits, and GG indicates if a carry is generated for the operation.

We verified the 16-bit CLA using both Verilog and LTSpice. Verilog simulations confirmed the logical accuracy of the sum, PG, and GG outputs for all input combinations, while LTSpice provided detailed electrical behavior, showing the correct waveforms for inputs and outputs. Both methods validated the design's correctness and efficiency.

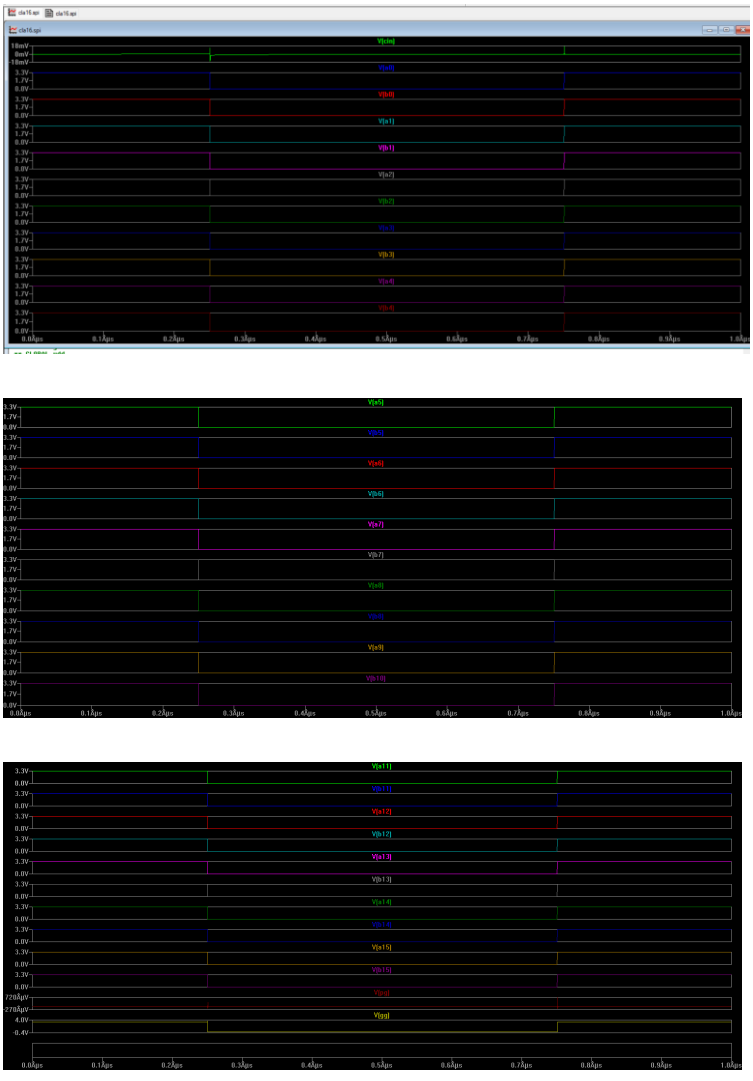

Figure 31 16-Bit CLA LTspice output.

Table 4 16-Bit CLA testbench output.

Truth Table for Inputs A, B, Cin, and Outputs Sum, GG, PG

Time (ns)	A	B	Cin	Sum	GG	PG
0	0000000000000000	0000000000000000	0	0000000000000000	0	0
10000	0000000000000001	0000000000000001	0	0000000000000010	0	0
20000	1111111111111111	0000000000000001	0	0000000000000000	1	0
30000	1111111111111111	1111111111111111	0	1111111111111110	1	0
40000	1000000000000000	0111111111111111	1	0000000000000000	0	1
50000	1010101010101010	0101010101010101	0	1111111111111111	0	1
60000	0001001000110100	0101011001111000	1	0110100010101101	0	0
70000	0000000000000000	0000000000000000	1	0000000000000001	0	0

F. Calculations for All CLA Models.

Table 5 The power, delay and area for CLA models.

Model	Power (mW)	Delay (ps)	Area (µm²)
1-bit CLA	0.95	55	283.18
2-bit CLA	1.52	110	1132.70
3-bit CLA	1.89	75	845.35
4-bit CLA	3.78	150	1237.37
16-bit CLA	15.14	780	4949.49

The table presents the power, delay, and area metrics for various Carry Look-Ahead Adder (CLA) models, ranging from a 1-bit CLA to a 16-bit CLA. These results were derived through a combination of transistor-level simulations and analytical calculations. The power dissipation was calculated using the formula P = VDD × IDD, where VDD is the supply voltage and IDD is the current drawn by the circuit.

The delay was determined by evaluating the critical path in each model, which accounts for the propagation time through the logic gates. The delay is proportional to the number of stages in the critical path and the individual gate delays, and can be expressed as t_delay = Σ (t_stage), where t_stage represents the delay of each gate in the critical path. The area was computed by measuring the layout dimensions of each CLA model, specifically the product of the width and height of the implemented circuit, A = W × H, where W is the width and H is the height of the circuit layout.

As the number of bits increases, the power consumption and area required for the Carry Look-Ahead (CLA) grow roughly in direct proportion to the bit-width. However, the delay, which is affected by the critical path, increases with complexity. Notably, for the 16-bit CLA, there is a considerable increase in power dissipation and area compared to smaller bit-widths, such as the 1-bit CLA.

Despite these increases, the delay only rises moderately due to the hierarchical structure of the CLA, which optimizes carry propagation. This shows that, while power and area increase with the bit-width, the CLA architecture is designed to manage the trade-offs between performance and resource consumption effectively, even for larger bit-widths like 16x16.

IV. Conclusion.

The presented 16-bit Carry Look-Ahead Adder (CLA) design demonstrates a modular approach, combining four 4-bit CLA modules to effectively tackle the issues of propagation delay and scalability. By leveraging the hierarchical structure, the design achieves faster carry computation compared to traditional ripple-carry adders. The use of intermediate carry blocks (c1, c2, c3) further enhances the speed by localizing and optimizing carry propagation.

Additionally, when implemented with advanced transistor technologies, such as NMOS and PMOS in a 22nm process, this design can significantly improve both energy efficiency and computational speed. While exact performance metrics depend on the hardware setup, this architecture is well-suited for applications requiring high-speed and energy-efficient arithmetic operations, such as signal processing and cryptographic systems.

V. References

[1] P. B. a. D. L. Maskell, "A New Carry Look-Ahead Adder Architecture Optimized for Speed and Energy," *Electronics*, vol. vol. 13, no. no. 18, p. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0289569.

[2] G. f. Geeks, "Carry Look-Ahead Adder," [Online]. Available: https://www.geeksforgeeks.org/carry-look-ahead-adder/. [Accessed 7 January 2025].

[3] Academia, "High Performance Design of a 4-Bit Carry Look-Ahead Adder in Static CMOS Logic," [Online]. Available: https://www.academia.edu/49268772/High_Performance_Design_of_a_4_Bit_Carry_Look_Ahead_Adder_in_Static_CMOS_Logic. [Accessed 7 January 2025].