International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Performance Evaluation of Ripple Carry and Carry Look-Ahead Adders: Insights from 4-bit Designs

[1]A. Pragathi
Dept of ECE
GMR Institute of Technology
Rajam, Andhra Pradesh, India
22341A0406@gmrit.edu.in

[2]B. Pravallika
Dept of ECE
GMR Institute of Technology
Rajam, Andhra Pradesh, India
22341A0413@gmrit.edu.in

[3]B. Bhanu Murthy
Dept of ECE
GMR Institute of Technology
Rajam, Andhra Pradesh, India
22341A0415@gmrit.edu.in

[4]B. Prem Kumar
Dept of ECE
GMR Institute of Technology
Rajam, Andhra Pradesh, India
22341A0420@gmrit.edu.in

[5]B. Bhavani
Dept of ECE
GMR Institute of Technology
Rajam, Andhra Pradesh, India
22341A0427@gmrit.edu.in

[6]Dr. Yogesh Mishra
Dept of ECE
GMR Institute of Technology
Rajam, Andhra Pradesh, India

Abstract: This research evaluates the performance of two common adder structures, a 4-bit Ripple Carry Adder (RCA) and 4-bit Carry Look-Ahead Adder (CLA). RCA features a small compact design that requires minimal hardware, but its sequential propagation delay increases linearly, which leads to inefficiencies at high-speed application rates. The CLA speeds up calculations through its Generate and Propagate functions, which predict carry values in advance; therefore, it works well in high-speed arithmetic operations. The supplementary logic circuits increase hardware complexity and generate additional power consumption in systems. The analysis evaluates the operation methods, hardware needs, performance speeds, and scalability aspects before investigating the possibilities of enhancing the performance through hybrid structure designs. This research will help to develop decisions regarding efficient adder selection and design for contemporary digital computing devices.

Keywords: Ripple Carry Adder (RCA), Carry Lookahead Adder (CLA), Propagation Delay, Digital Arithmetic Circuits, High-Speed Computing.

I. INTRODUCTION

The significance of adders in digital arithmetic operations becomes crucial for microprocessors and arithmetic logic units (ALUs) and digital signal processors (DSPs) and embedded systems. The ability of Adders to determine binary number sums becomes vital for whole operations since addition functions in floating-point calculations and addresses generation and control logic operations alongside complex signal processing algorithm implementations. The development of adders requires finding optimal solutions that reduce both their hardware complexity and operation timing needs when faster and more efficient systems arise. The two fundamental design choices for adders involve the implementation of Ripple Carry Adder (RCA) and Carry Look-Ahead Adder (CLA). The basic design of Ripple Carry Adder involves sequential adders arranged as a linear chain so its limited efficiency prevents such systems from supporting high-speed computations. The CLA operates with Generate (G) functions and Propagate (P) functions. The prediction of parallel carry values through the (G) and Propagate (P) functions reduces the overall time duration needed to execute the sum calculation. The Carry Look-Ahead adder achieves faster speed than the Ripple Carry Adder but demands more complicated hardware which increases power requirements as it needs additional logic elements for carry calculations. The evaluation study investigates the architectural standards and operational performance along with energy usage and system complexity and flexibility of RCA and CLA to select the best design for different practical applications. The

review examines design integration methods which unite RCA and CLA capabilities to produce effective solutions for systems requiring rapid operation together with optimization.

RIPPLE CARRY ADDER (RCA)

Basic digital circuits use Ripple Carry Adders as digital binary addition components. The design follows a straightforward sequence which conducts bit sum computation step by step while carry values move progressively from one stage to the next. Each stage of the RCA contains a full adder that utilizes two bits from the respective lower numbers alongside the carry-in bit to determine the output sum. Each full adder produces the sum bit for its stage before passing on the carry- out value to the following stage as the carry- in value. The n-bit RCA needs n full adders that perform the summation for one bit at each position of both input binary numbers.

The sum and carry of each bit are calculated using formulas: sum = $A_i \oplus B_i \oplus C_i$ and carry = Ci+1 =(Ai·Bi)+(Ci·(Ai+Bi))).The RCA is simple, easy to design, and requires fewer hardware requirements than more complex adder types like the Carry Look-Ahead Adder (CLA). However, it is not suitable for high-speed systems, where adders like the Carry Look-Ahead Adder (CLA) or Manchester Carry Chain Adders are preferred for faster computation.

A. ARCHITECTURE

An essential digital circuit component named Ripple Carry Adder (RCA arranges multiple full adders with a sequential order for performing binary addition. Full adders act as components that produce the addition and carry output for individual bits of two numbers entering the system. Multiple architectural elements form the whole system of the RCA including full adders and carry chains as well as input and output lines with logic components including AND and XOR gates and OR gates.

The truth table for a full adder is as follows:

A	B	C	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

In a Ripple Carry Adder (RCA) for an n-bit addition, there are n full adders connected in series. The carry from each adder propagates to the next, hence the name "ripple carry."

For example, for two 4-bit numbers A = A3 A2 A1 A0 and

B= B3 B2 B1 B0, where A3, A2, A1, A0, and B3, B3, B2, B1,

B0 represent the individual bits of the two binary numbers, the structure of the RCA for these 4-bit numbers is as follows.

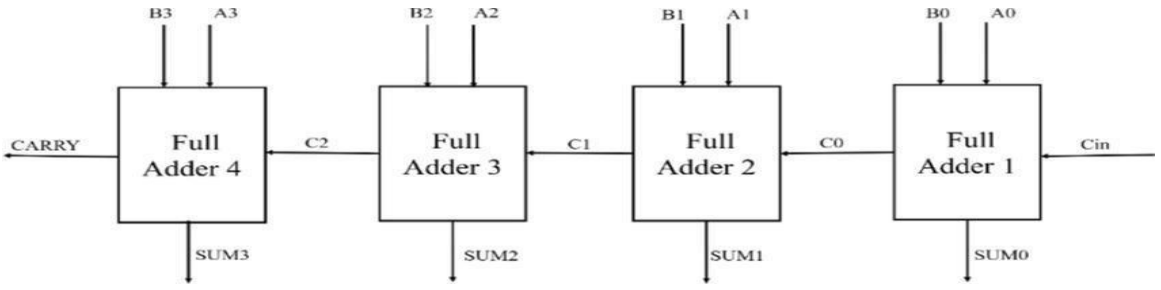


Fig 1 Block Diagram of 4 bit ripple carry adder

1. **First Stage (Full Adder 1):**

First full adder adds the least significant bits of A and B (A0 and B0), along with the initial carry (Cin = 0). It computes the sum (S0) and produces a carry-out (C1) that will be used as the carry-in for the next full adder.

2. **Second Stage (Full Adder 2):**

Second full adder adds the second least significant bits of A and B (A1 and B1) and the carry-in from the previous stage ((C1).
It computes the sum (S2) and generates a carry-out (C3).

3. **Third Stage (Full Adder 3):**

Third full adder performs a similar operation with bits A2 and B2 and the carry-in from the previous stage. It computes the sum (S2) and generates a carry-out (C3).

4. **Final Stage (Full Adder 4):**

The last full adder provides sum calculations for the most significant bits A3 and B3 and the carry-in value (C3) delivered from the previous stage. The delay in the RCA becomes apparent during the propagation time of the carry from its least to its most significant bit locations. Each full adder requires the input carry to determine its output carry because the dependability propagates from one full adder to the next leading to proportional delay growth with added bits. The RCA contains basic design features which use minimal hardware thus making it simple to deploy and understand. The time required for the delay increases along with the number of bits under Sequential Carry Propagation. The full adder scalability through broader bit-widths depends on additional implementation while the resulting delay increases with larger bit-byte counts. While the RCA maintains power-efficient operations during smaller bit operations it shows poor energy performance at higher bit ranges. The Ripple Carry Adder functions as a foundational digital circuit element for binary addition while delivering straightforward design yet experiences speed-related problems as the amount of bits enlarges specifically in time-sensitive applications.

B. WORKING PRINCIPLE

The Ripple Carry Adder (RCA) executes binary number addition by moving through corresponding bits in sequence while considering the carry effect that occurs at each stage. Core operations of the full adder determine the computation of sum and carry outputs for individual bits. A carry output from every successive full adder propagates the carry to the next full adder in a ripple-based system until the carry reaches the end of the stages. An RCA contains three main components which are its two initial binary inputs A and B together with the default Cin value which begins at zero. The circuit produces two outputs through its operation which includes both the sum output (S) and the carry output (Cout). The sum output produces the total addition result at the same time the carry-out expresses the carry value from the most significant bit. The basic component of RCA forms a Full Adder as a combinational circuit that combines two operands bits A and B while accepting a carry-in bit Cin. The circuit produces two outputs which include Sum (S) that represents the bitwise sum of its inputs. The Ripple Effect of Carry Propagation stands as the main core feature of the RCA configuration. A single full adder calculates bit sums while its carry output moves sequentially forward to the following full adder and continues through each one. The carry propagates from bit to bit through successive adders during this method of calculation. The winding propagation of carry across each bit gives the Ripple Carry Adder its descriptive name. In this situation of adding four bits using RCA arithmetic the sum calculation in the first bit depends on all four variables A_0 B_0 C_0 and start carry (C_0). The calculation of the second bit sum utilizes A_1, B_1 and the carry value from C_1. Ripple Carry Adder uses three main features including propagating carry values one after another together with individual bit operations and timing delays. The computational delay of an RCA grows continuously according to the number of bits since carries require complete traversal of every preceding stage before the final sum calculation completes.

The Ripple Carry Adder (RCA) consists of an architectural design which performs straightforward bit addition through full adders allowing carry propagation between stages. The carry propagation delay causes the RCA to experience poor

performance levels that limit its use in fast processing environments.

II. METHODOLOGY

The methodology applied to analyze 4-bit Ripple Carry Adder (RCA) and Carry Look-Ahead Adder (CLA) determines their performance metrics for speed, power usage, area occupation and complexity levels and scalability potential. The analysis includes three steps which are theoretical modeling together with simulation and performance metric-based comparison. A full adder forms the basis of the RCA through series connectivity to calculate sum and carry outputs for individual bits. The main emphasis assesses how the carry moves progressively through the bits while bringing increased delays as per the number of bits processed. The Carry Look-Ahead Adder (CLA) decreases carry propagation time through parallel calculation of all carry values using "Generate (G)" and "Propagate (P)" signals which are processed through a logic gate hierarchy in parallel. Theoretical evaluation of RCA and CLA determines total delay through calculations that track carry propagation starting from the LSB up to the MSB. The numbers of bits determine the duration of the delay because of direct proportionality thus creating an O(n) delay. The power consumption of RCA depends on both the total number of bits as well as the gate counts for each bit between AND, OR and XOR logic gates. The power consumption follows a direct relation between operational frequency and bit number. The power requirements of CLA become higher due to its additional "Generate" and "Propagate" signals together with the hierarchical carry calculation schemes. Area analysis includes both RCA and CLA assessment. The total area depends on the count of full adders along with the necessary gates per adder consisting of AND, OR and XOR gates. The larger area of this block originates from both the new "Generate" and "Propagate" signals and carry look-ahead logic gates. Design and simulation of the circuits occurs through Verilog or other digital hardware description language models which represent system behavior. The computational model combines RCA and CLA structures through simulation until the carry propagation performance can be analyzed between both methods. The simulation data allows a power analysis of the circuit and an area assessment through gate-level switching activity measurement. Decision-making about real-world digital system implementations becomes more effective through comparison of speed, power consumption, area, complexity and scalability results. Designers should use RCA when creating low-power applications with basic functionality and few bits since CLA provides superior speed and carry propagation delay reduction for high-speed designs.

I. CARRY LOOK-AHEAD ADDER(CLA)

An advanced adder design named the Carry Look-Ahead Adder (CLA) uses parallel carry generation to overcome the main carry propagation delay issues present in Ripple Carry Adders (RCA). The CLA performs parallel carry calculation for its bits which shortens the total addition time. Parallel carry generation and reduced propagation delay together with faster performance and increased hardware complexity are the fundamental characteristics of CLA. The main concepts supporting the CLA design include Generate (G) and Propagate (P) functionality. These parallel processing functions determine carry values across all positions which helps decrease propagation delay throughout the calculation process. A major benefit of CLA is its ability to operate fast with scalability benefits compared to other designs for larger bit-widths as well as parallel functionality. The parallel operation of CLA results in hardware complexity alongside power consumption and high implementation expenses. High-speed processors together with arithmetic logic units (ALUs) and multiplexers and digital signal processing systems implement CLA functionality. The processor and high-performance digital circuits benefit most from applying this method because of its critical significance for speed. Research about CLA systems will likely work to minimize hardware complexity without sacrificing its fast computing capabilities.

A. ARCHITECTURE

A fast adder design named Carry Look-Ahead Adder (CLA) reduces carry propagation delay to enhance the speed of addition operations. The CLA operates unlike the Ripple Carry Adder (RCA) since it evaluates carry values simultaneously instead of doing it incrementally. This parallel operation of carry calculation speeds up execution time in the CLA. CLA operates with two essential functions which are called Generate (G) and Propagate (P) Signals. The

Generate (G) function determines if a carry occurs while the Propagate (P) function shows if carry propagation happens during the next stage. The computation of G and P values for every bit uses A_i and B_i input bits as the basis. Parallel calculation of carries represents the main benefit for CLA processors. The Generate and Propagate values determine the carry calculation for each bit so that multiple bits obtain their carry values simultaneously before waiting for a sequential propagation process.

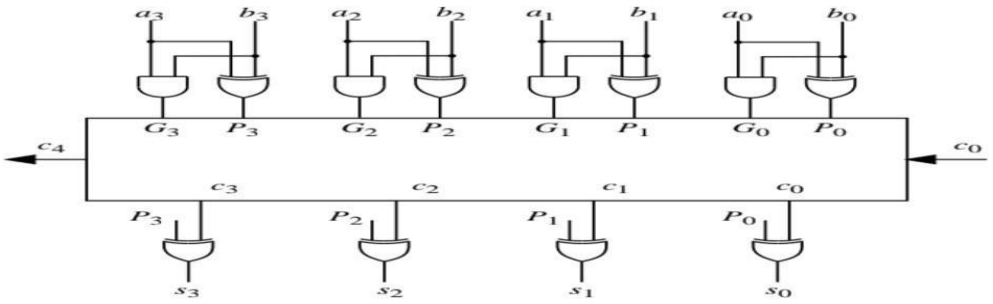


Fig .2 Block diagram of Carry Look Ahead Adder

The carry calculations are done using recursive equations, with the first carry determined using G_0 and P_0, and the carry values for each subsequent bit calculated using the generate and propagate signals.The sum calculation logic is the same for each bit, and the sum for each bit is computed after the carry for that bit is determined. The sum for each bit is given by S_i= A_i ⊕ B_i ⊕ C_i, where A_i and B_i are the input bits for addition and C_i is the carry for the bit, which was calculated in the previous step.The 4-bit Carry Look-Ahead Adder can be structured into three blocks: Generate and Propagate Blocks, Carry Calculation Block, and Sum Calculation Block. Generate and Propagate Blocks calculate the Generate(G) and Propagate (P) signals for each bit, while Carry Calculation Block computes the carry outputs using the Generate and Propagate signals. Sum Calculation Blocks calculate the sum for each bit using XOR gates with the corresponding bits of A_i, B_i, and the calculated carries.In conclusion, the Carry Look-Ahead Adder (CLA) is a fast adder design that significantly reduces the delay caused by carry propagation. By computing the carries in parallel using Generate (G) and Propagate (P) functions, the CLA can perform addition in a shorter time compared to traditional adders like the Ripple Carry Adder (RCA). However, this improvement comes at the cost of increased hardware complexity due to the additional logic required for parallel carry calculation.

B. WORKING PRINCIPLE

The Carry Look-Ahead Adder (CLA) exists as a hardware implementation that addresses speed limitations found in traditional Ripple Carry Adder (RCA) designs because of its carry propagation slowness. Each bit carry in an RCA propagates sequentially to the following bit position which accumulates overall delay. CLA enhances carry calculation through parallel operations of "generate (G)" and "propagate (P)" which significantly accelerates addition operations. The carry lookup array evaluates all positions of carry simultaneously through generate and propagate functions rather than waiting for sequential calculations. The CLA functioning procedure requires the following key process steps:

Generate (G) and Propagate (P) Functions:

For each bit of two numbers (A and B), two signals are computed:

1. **Generate (G):** Indicates whether a carry will be generated at the given bit position. If both bits are 1, then a carry will be generated at that position.
2. **Carry Propagation Logic:**

Using the "generate" and "propagate" functions, the carry for each bit position is determined in parallel without the need for sequential carry propagation. The carry-in for each bit is computed by considering whether the previous carries should propagate through or generate new carries.

3. **Sum Calculation:**

Once the carry bits are determined, the sum for each bit is computed using the "XOR" operation. The sum for each bit is determined as follows:

$Si=Ai \oplus Bi \oplus C$

This sum calculation is also done in parallel for all the bits.

The main advantage of CLA's working principle is its ability to compute the carry bits in parallel, rather than sequentially, which significantly reduces the delay, especially for large bit- widths.

1. **Efficient Carry Calculation:**

By using the "Generate" and "Propagate" signals, the CLA can calculate carries quickly and independently of the bit positions,enabling faster additions.

2. **Reduced Carry Propagation Delay:**

The CLA reduces the critical path delay, which is the key limitation in RCA. The carry propagation delay is logarithmic with the number of bits in CLA, while it is linear in RCA

In conclusion, the Carry Look-Ahead Adder's working principle revolves around the efficient parallel computation of carry bits using "generate" and "propagate" signals. By pre-calculating the carries, the CLA removes the need for sequential carry propagation, making it faster than traditional adders like the RCA. Although the CLA requires more hardware to implement, it offers a significant performance improvement for applications where speed is crucial.

III. PERFORMANCE COMPARISON

Property	**RCA**	**CLA**
Power (mW)	0.85672	0.924
Area (um^2)	3812.576	4256.892
Delay	319.89	230.06

Two prominent constructions of digital circuits include the Ripple Carry Adder (RCA) along with the Carry Look-Ahead Adder (CLA). The individual adder architectures present different performance capabilities that match different system needs which depend on speed demands and area coverage as well as power requirements and complexity needs. An RCA produces propagation delay that matches the number of bits that get processed so the total delay becomes O(n). The delay of 4-bit RCA grows long due to the propagation process which depends on sequential carry chains. The Generate function in CLA lets the device execute all carries at once in parallel processing. The (G) and Propagate (P) functions enable CLA to achieve an O(log n) propagation delay making its delay logarithmic relative to the number of bits. The RCA has a lower operation speed when compared to its RCA counterpart because it operates at serial full adder connections that need up to two XOR gates and two AND gates along with one OR gate. A smaller area is accessible for smaller bit-widths but CLA demands supplementary logic to produce Generate (G) signals and Propagate (P) signals and additional gates to generate parallel carry computations. The additional logic needed for CLA produces a larger hardware area than an RCA functions would. RCA demonstrates low energy use throughout its operations because it uses only straightforward full adders with no supplementary logic for carry prediction functions. When the bit-width elevates the lengthened carry propagation delay generates larger power consumption. The additional power demand of CLA occurs because it needs multiple gates to execute its parallel carry computations. Despite its speed improvement through parallel computing the CLA creates additional power consumption since it requires supplementary logic. Simple full adder connection in series makes RCA operation low in complexity. The generation of the carry requires no supplementary logic. CLA presents increased complexity because it demands supplementary logic to execute parallel calculations of the carry bits. The delay grows in direct proportion to the number of bits due to the fixed carry

computation mechanism that combines generation logic with propagation and carry operations at each position. Therefore large bit-widths make scalability impossible. High-speed operations involving broad bit sizes can be supported better with CLA than with RCA because its carry propagation delay shows log-based scaling patterns. The 4-bit sizes make RCA the preferred option when considering its simple design and lower power consumption together with minimal area requirements. An RCA operation experiences declining performance when working with growing bit-widths because both increased complexity and latency from the CLA affect its functionality.

IV. RESULTS

The outcomes include the simulation and schematic results of the Ripple Carry Adder (RCA) and Carry Look-Ahead Adder (CLA) respectively. The schematic results are observed using the Cadence synthesis tool.

A. Schematic Results:

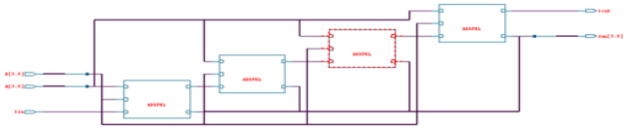


Fig.3 Schematic result of 4-bit Ripple Carry Adder

The design consists of four cascaded full adders, where the carry output of each stage serves as the carry input for the next.

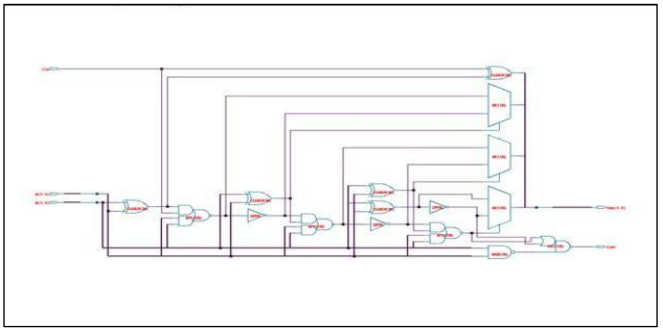


Fig.4 Schematic result of 4-bit Carry Look-Ahead Adder

The CLA structure emphasizes the optimized logic path, with carry outputs determined in advance, leading to faster summation.

B. Simulation Results:

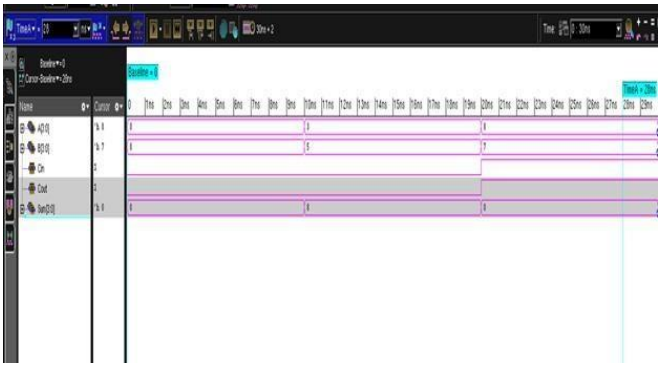


Fig. 5Simulation results of 4-bit Ripple Carry Adder

The simulation output showcases the sequential behavior of the RCA. Unlike the CLA, the sum bits stabilize progressively from the least significant bit (LSB) to the most significant bit (MSB), revealing the ripple effect. Intermediate glitches in the waveform signify the delay in carry propagation. This behaviour confirms the inherent latency in ripple-based designs.

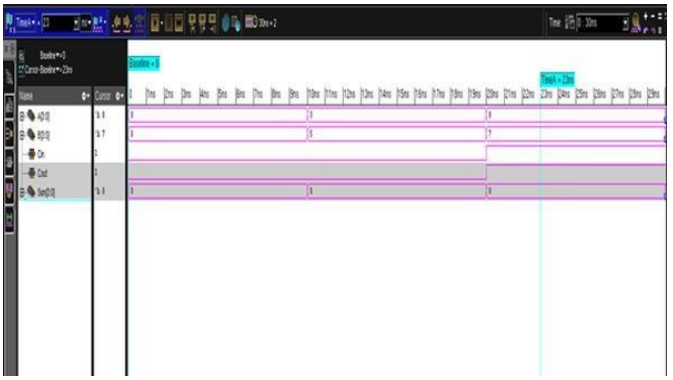
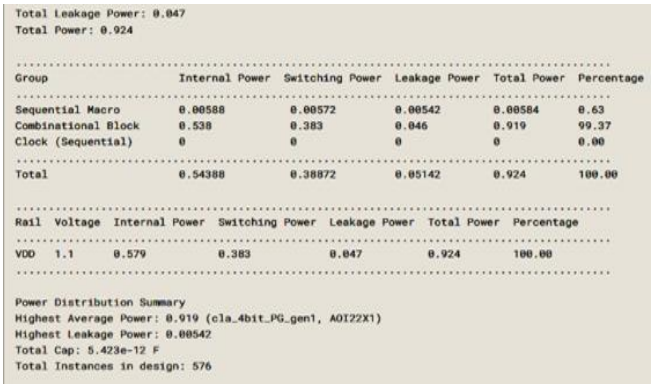


Fig.6 Simulation results of 4-bit Carry Look-Ahead Adder

The waveform illustrates that the sum bits are stabilized almost simultaneously, indicating minimal delay. This confirms that the CLA achieves parallel carry generation and significantly reduces total computation time compared to traditional ripple carry architectures.

C. Reports:

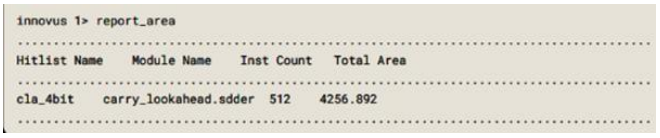
Using the synthesis tool, the timing reports of the two adders are also generated. The report confirms no timing violations (negative slack) in this maximum delay scenario.



```
Total Leakage Power: 0.047
Total Power: 0.924

Group                  Internal Power  Switching Power  Leakage Power  Total Power  Percentage
Sequential Macro       0.00588         0.00572          0.00542        0.00584      0.63
Combinational Block    0.538           0.383            0.046          0.919        99.37
Clock (Sequential)     0               0                0              0            0.00
Total                  0.54388         0.38872          0.05142        0.924        100.00

Rail  Voltage  Internal Power  Switching Power  Leakage Power  Total Power  Percentage
VDD   1.1      0.579          0.383            0.047          0.924        100.00

Power Distribution Summary
Highest Average Power: 0.919 (cla_4bit_PG_gen1, AOI22X1)
Highest Leakage Power: 0.00542
Total Cap: 5.423e-12 F
Total Instances in design: 576
```

Fig.7 Power Report of 4-bit Carry Look-ahead Adder

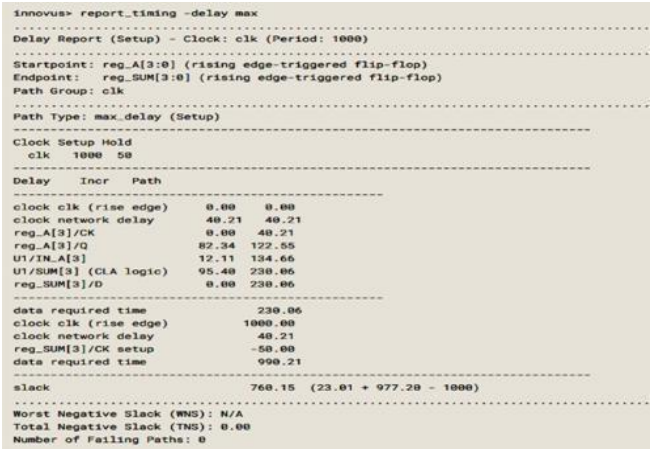
The power analysis reveals that the total power consumed by the CLA is 0.924 mW, with the majority attributed to the combinational logic blocks (0.919 mW). Leakage power is minimal at 0.047 mW.



```
innovus 1> report_area

Hitlist Name    Module Name    Inst Count   Total Area
cla_4bit    carry_lookahead.sdder  512    4256.892
```

Fig.8 Area Report of 4-bit Carry Look-ahead Adder

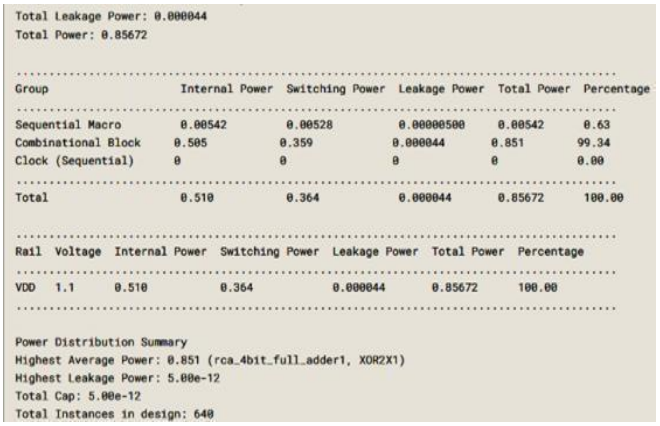
This report presents the physical area required for the CLA implementation, calculated at 4256.892 μm². The instance count of 512 reflects the higher component usage due to complex lookahead circuitry.



```
innovus> report_timing -delay max
Delay Report (Setup) - Clock: clk (Period: 1000)
Startpoint: reg_A[3:0] (rising edge-triggered flip-flop)
Endpoint:   reg_SUM[3:0] (rising edge-triggered flip-flop)
Path Group: clk
Path Type: max_delay (Setup)
Clock Setup Hold
  clk   1000  50
Delay    Incr   Path
clock clk (rise edge)      0.00    0.00
clock network delay       40.21   40.21
reg_A[3]/CK                0.00   40.21
reg_A[3]/Q                82.34  122.55
U1/IN_A[3]                12.11  134.66
U1/SUM[3] (CLA logic)     95.40  230.06
reg_SUM[3]/D               0.00  230.06
data required time                230.06
clock clk (rise edge)            1000.00
clock network delay                40.21
reg_SUM[3]/CK setup               -50.00
data required time                990.21
slack                             760.15  (23.01 + 977.20 - 1000)
Worst Negative Slack (WNS): N/A
Total Negative Slack (TNS): 0.00
Number of Failing Paths: 0
```

Fig.9 Timing Report of 4-bit Carry Look-ahead Adder

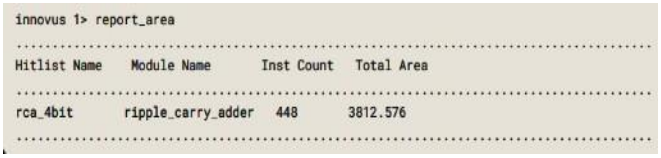
The timing analysis shows a worst-case signal propagation delay that result in a positive slack of 760.15 ps. This indicates that all timing constraints are met, and the CLA has a reliable timing margin.



```
Total Leakage Power: 0.000044
Total Power: 0.85672

.....................................................................................
Group                  Internal Power  Switching Power  Leakage Power  Total Power  Percentage
.....................................................................................
Sequential Macro       0.00542         0.00528          0.00000500     0.00542      0.63
Combinational Block    0.505           0.359            0.000044       0.851        99.34
Clock (Sequential)     0               0                0              0            0.00
.....................................................................................
Total                  0.510           0.364            0.000044       0.85672      100.00

.....................................................................................
Rail  Voltage  Internal Power  Switching Power  Leakage Power  Total Power  Percentage
.....................................................................................
VDD   1.1      0.510           0.364            0.000044       0.85672      100.00
.....................................................................................

Power Distribution Summary
Highest Average Power: 0.851 (rca_4bit_full_adder1, XOR2X1)
Highest Leakage Power: 5.00e-12
Total Cap: 5.00e-12
Total Instances in design: 640
```

Fig.10 Power Report of 4-bit Ripple Carry Adder

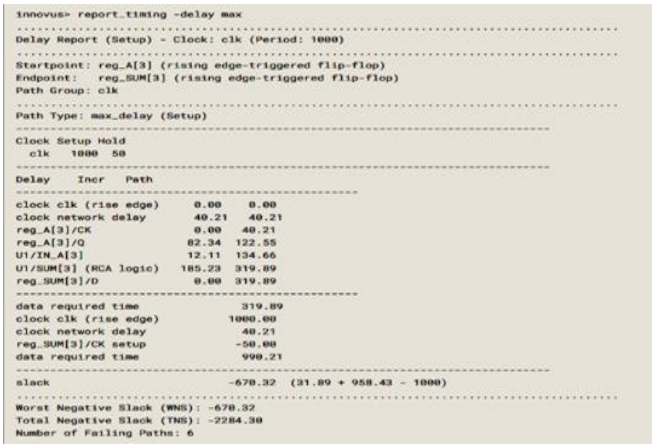
According to the power analysis, the RCA consumes 0.85672 mW, slightly lower than the CLA. The majority of this power is consumed by the combinational logic (0.851 mW), with negligible leakage and sequential power components.



```
innovus 1> report_area
.....................................................................................
Hitlist Name    Module Name       Inst Count   Total Area
.....................................................................................
rca_4bit        ripple_carry_adder   448       3812.576
.....................................................................................
```

Fig.11 Area Report of 4-bit Ripple Carry Adder

The area report indicates that the RCA occupies 3812.576 µm², which is smaller than that of the CLA. With an instance count of 448, this design uses fewer resources, making it more compact. The reduced area utilization supports its deployment in area-constrained systems.



```
innovus> report_timing -delay max
.....................................................................................
Delay Report (Setup) - Clock: clk (Period: 1000)
.....................................................................................
Startpoint: reg_A[3] (rising edge-triggered flip-flop)
Endpoint:   reg_SUM[3] (rising edge-triggered flip-flop)
Path Group: clk
.....................................................................................
Path Type: max_delay (Setup)
-------------------------------------------------------------------------------
Clock Setup Hold
  clk   1000  50
-------------------------------------------------------------------------------
Delay    Incr    Path
-----------------------------------------------
clock clk (rise edge)      0.00    0.00
clock network delay       40.21   40.21
reg_A[3]/CK                0.00   40.21
reg_A[3]/Q                82.34  122.55
U1/IN_A[3]                12.11  134.66
U1/SUM[3] (RCA logic)    185.23  319.89
reg_SUM[3]/D               0.00  319.89
-----------------------------------------------
data required time             319.89
clock clk (rise edge)         1000.00
clock network delay             40.21
reg_SUM[3]/CK setup            -50.00
data required time             990.21
-------------------------------------------------------------------------------
slack                         -670.32  (31.89 + 958.43 - 1000)
.....................................................................................
Worst Negative Slack (WNS): -670.32
Total Negative Slack (TNS): -2284.30
Number of Failing Paths: 6
```

Fig.12 Timing Report of 4-bit Ripple Carry Adder

The timing report reveals a critical issue in the RCA design, with a worst negative slack (WNS) of -670.32 ps and a total negative slack (TNS) of -2284.30 ps.

V. CONCLUSION

This evaluation shows that 4-bit Ripple Carry Adder (RCA) and 4-bit Carry Look-Ahead Adder (CLA) feature different architectural performance aspects and operational characteristics. The RCA operates at 0.85672 mW power while using a silicon area of 3812.576 µm² and performing with a 319.89 ps propagation delay.

319.89 ps. The CLA achieves faster operation due to its 230.06 ps delay but consumes more power (0.924 mW) and occupies larger silicon space (4256.892 µm²). The achieved results show the CLA delivers superior delay reduction for time-sensitive applications but the RCA presents itself as a suitable solution when speed demands are minimal. The selection between RCA and CLA requires consideration of design application factors including power efficiency and area specifications as well as timing benchmarks.

VI. REFERENCES

[1] Balasubramanian, P., et al. "Approximate ripple carry and carry lookahead adders—A comparative analysis." 2017 IEEE 30th International Conference on Microelectronics (MIEL). IEEE, 2017.

[2] Balasubramanian, Padmanabhan, and Douglas Maskell. "A new carry look-ahead adder architecture enabling improved speed and energy efficiency." 2024 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM). IEEE, 2024.

[3] Shekhawat, Kuldeep Singh, and Gajendra Sujediya. "Design and analysis of RCA and CLA using CMOS, GDI, TG and ECRL Technology." International Journal of Advanced Engineering Research and Science 4.11 (2017): 126-129.

[4] Mohithsaicharan, Karanam, and P. Jagadeesh. "Performance analysis of carry look ahead adder with ripple carry adder using VHDL language." AIP Conference Proceedings. Vol. 2871. No. 1. AIP Publishing, 2024.

[5] Keshavarz, Shahrzad, and Daniel Holcomb. "Privacy leakages in approximate adders." 2017 IEEE international symposium on circuits and systems (ISCAS). IEEE, 2017.

[6] Iqbal, Asma, and K. Manjunatha Chari. "A brief analysis of fault-tolerant ripple carry adders with a design for reliable approximate adders." Proceedings of Second International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2021. Singapore: Springer Nature Singapore, 2022.