Volume 10, Issue 5, May – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1617

Implementation and Comparison of Different Types of High Speed Adders

Ravikumar R[1]; Skanda Gargesh N[2]; Manvith P[3]; Nagarjuna S G[4]; Vanish B H[5]

[1]Assistant Professor, [2,3,4,5] Students

[1,2,3,4,5] Department of Electronics and Communication Engineering, Maharaja Institute of Technology Thandavapura, Nanjangud, Mysore, Karnataka, India

Publication Date: 2025/05/26

Abstract: This paper presents the design and performance comparison of different high-speed adder architectures with a focus on optimizing delay, power consumption, and area utilization. A hybrid 128-bit adder is proposed by combining four popular adder types: Ripple Carry Adder (RCA), Carry Skip Adder (CSA), Carry Select Adder (CSLA), and Carry Look-Ahead Adder (CLA). Each adder is allocated to a specific segment of the 128-bit word based on its characteristics to improve overall performance. The design is implemented using SystemVerilog, simulated using Synopsys VCS, and synthesized for FPGA deployment. To enhance performance further, the design incorporates pipelining techniques and is based on NAND gate-level logic for better hardware optimization. Results indicate that the hybrid approach offers an effective trade-off among speed, power, and hardware utilization.

Keywords: *Hybrid Adder, System Verilog, RCA, CLA, CSLA, FPGA, High-Speed Arithmetic, Pipelining, NAND Gate.*

How to cite: Ravikumar R; Skanda Gargesh N; Manvith P; Nagarjuna S G; Vanish B H. (2025). Implementation and Comparison of Different Types of High Speed Adders. *International Journal of Innovative Science and Research Technology,* 10(5), 1566-1572. https://doi.org/10.38124/ijisrt/25may1617

I. INTRODUCTION

In modern digital systems, arithmetic operations are essential for performing a wide range of tasks, including data processing, encryption, real-time computation, and signal transformation. Among these, addition is the most frequently used arithmetic operation. The efficiency of an adder significantly impacts the overall performance of digital systems, particularly in arithmetic logic units (ALUs), digital signal processors (DSPs), and central processing units (CPUs). As technology progresses toward high-speed, low-power, and area-efficient designs, the role of high-performance adder architectures becomes even more critical.

Traditional adder designs, such as Ripple Carry Adder (RCA), are simple and require fewer hardware resources but suffer from high propagation delay due to sequential carry generation. On the other hand, advanced adder architectures like Carry Look-Ahead Adder (CLA) offer faster operation by precomputing carry signals, but they introduce increased complexity and area overhead. Similarly, Carry Skip Adders (CSA) and Carry Select Adders (CSLA) attempt to balance delay and complexity through structural innovations.

As no single adder architecture can simultaneously optimize speed, power, and area for all applications, hybrid adder designs have emerged as a promising alternative. By strategically combining different adder types, it becomes possible to exploit the advantages of each while mitigating their individual limitations. This concept is particularly valuable in designing wide-bit adders (e.g., 128-bit) that are commonly used in cryptography, multimedia processing, and high-performance computing.

This paper proposes a novel hybrid 128-bit adder architecture that partitions the bit-width into four segments, each implemented using a different adder type based on its performance profile. Furthermore, the design integrates pipelining techniques to improve clock performance and reduce critical path delay. NAND gate-based logic implementation is employed for efficient hardware synthesis. The objective is to demonstrate that this hybrid approach provides a practical trade-off between speed, power, and area, making it suitable for next-generation VLSI and embedded system designs.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1617

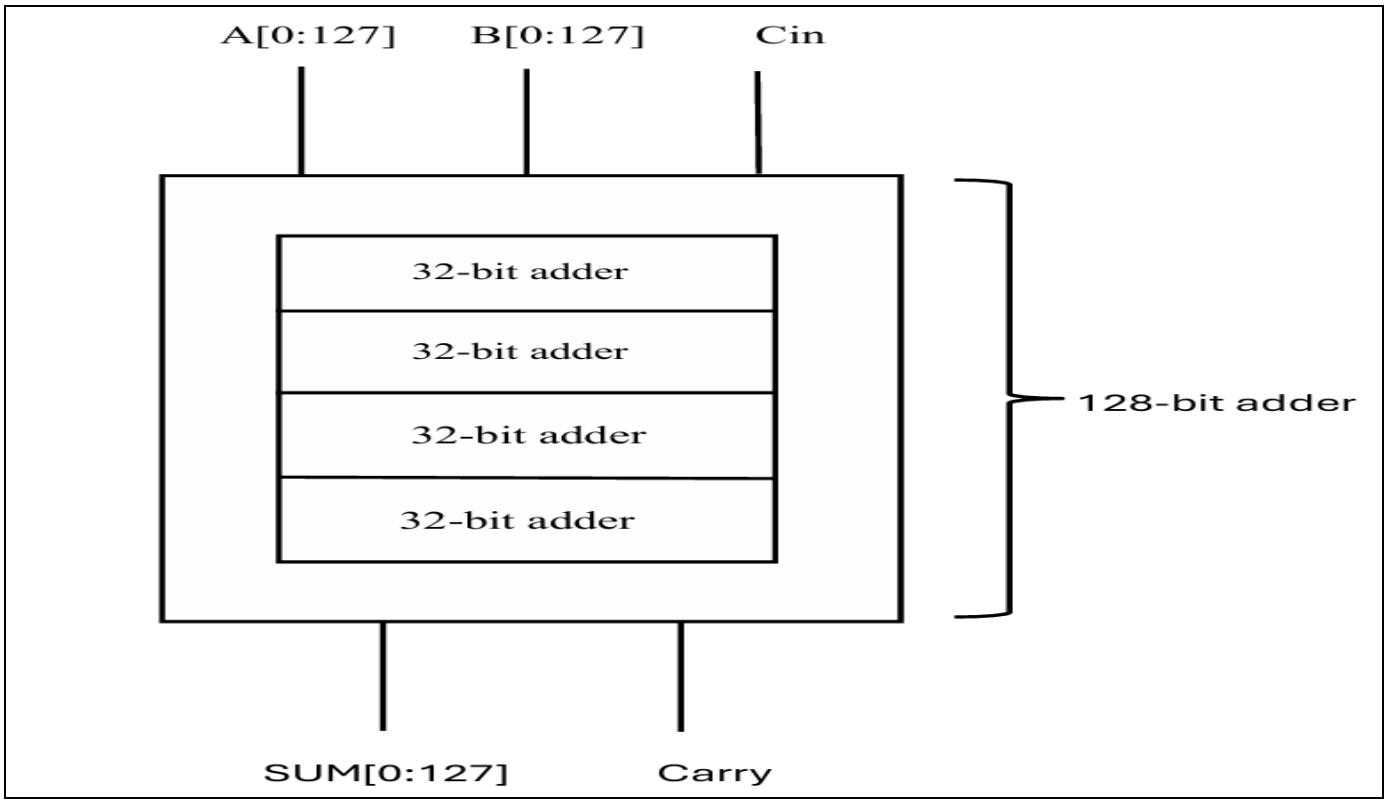

Fig 1 Basic Block Diagram of 128-bit Adder

II. METHODOLOGY

In this project, a hybrid approach is adopted to construct a 128-bit high-speed adder by intelligently combining different adder architectures. Instead of using a single adder design, the 128-bit input is divided into four equal 32-bit segments, each of which is implemented using a specific type of adder based on its performance advantages. The least significant 32 bits are handled by a Ripple Carry Adder (RCA), chosen for its simplicity and minimal hardware requirements. The second 32-bit segment uses a Carry Skip Adder (CSA), which accelerates carry propagation by skipping over intermediate stages where possible. The third segment employs a Carry Select Adder (CSLA), leveraging parallel computation to reduce delay. Finally, the most significant 32 bits are managed by a Carry Look-Ahead Adder (CLA), selected for its ability to predict carry values efficiently and deliver the fastest computation where delay is most critical.

To further enhance performance, pipelining stages are introduced between these segments. This helps to increase throughput by allowing partial results to be processed concurrently in successive clock cycles. Additionally, the design is optimized using NAND gate logic, which simplifies synthesis and provides compatibility with standard CMOS technology used in ASIC development. The complete adder is written in SystemVerilog using a modular coding approach, where each adder type is defined as a separate module and then integrated into a unified top-level design. Simulation is performed using Synopsys VCS to validate functional correctness under various test conditions, including edge cases like all-zero, all-one, and alternating input patterns. Synthesis and timing analysis are conducted using Xilinx ISE, with a scaled-down 12-bit version used for hardware implementation on Spartan-6 FPGA due to resource constraints. Despite this limitation, the methodology maintains full compatibility with the intended 128-bit architecture and confirms the practical viability of the hybrid adder design.

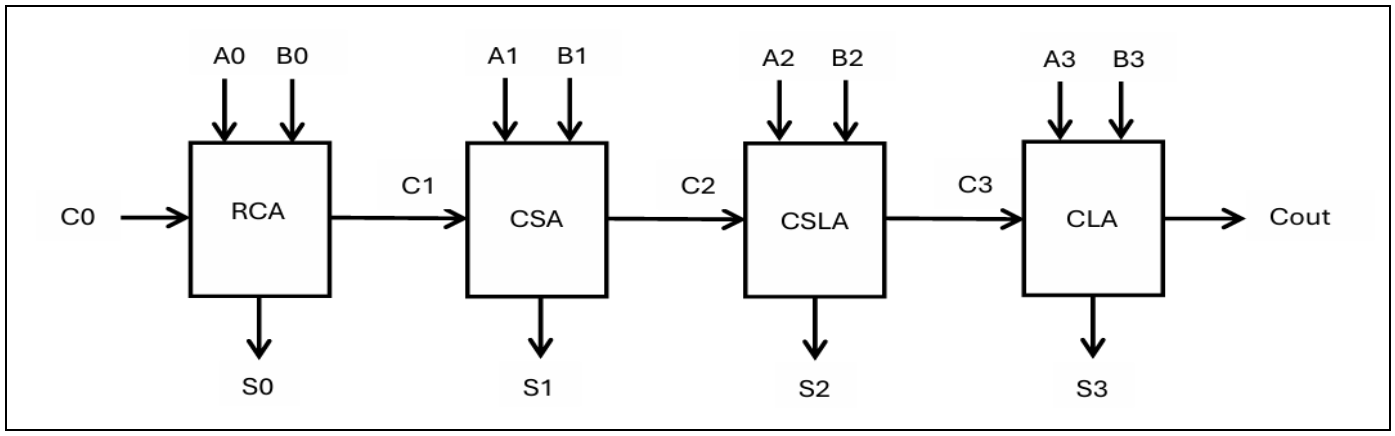

Fig 2 Block Diagram of Hybrid 128-bit Adder

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1617

➢ *Mathematical Expression for Carry Computation*
Ripple Carry Adder (RCA)

Each bit sum and carry are computed as:

$$S_i = A_i \oplus B_i \oplus C_i S_i$$

$$C_{i+1} = (A_i \cdot B_i) + (C_i \cdot (A_i \oplus B_i))$$

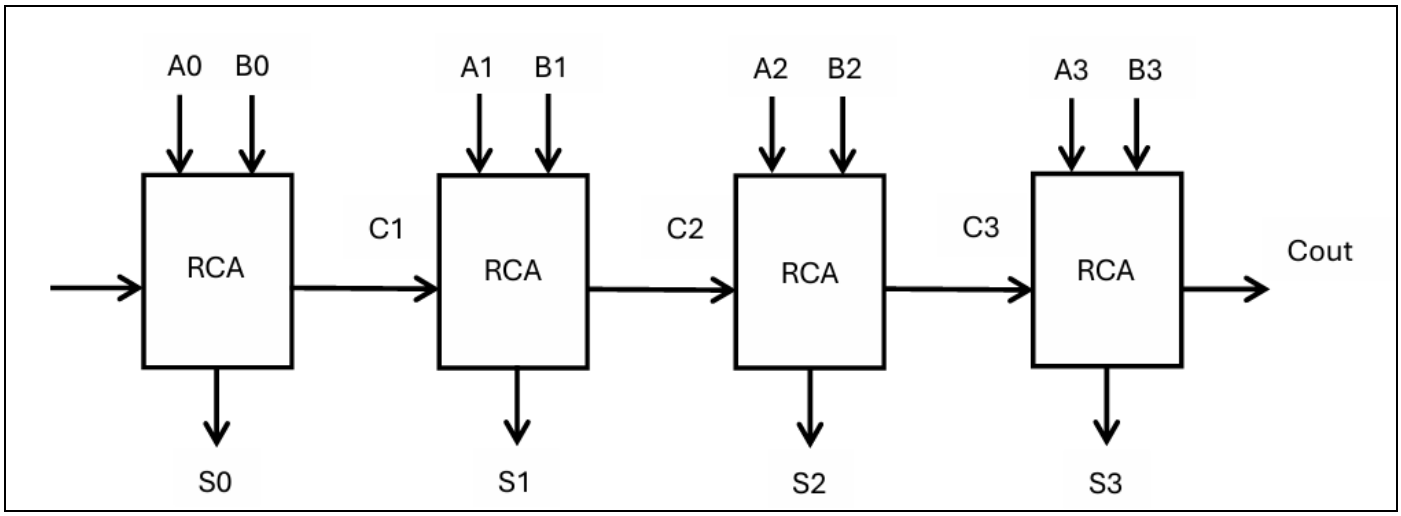

Fig 3 Ripple Carry Adder

Carry Skip Adder (CSA)

Carries are propagated efficiently using a carry-skip mechanism:

$$C_{out} = C_{in} + P \cdot G$$

Where:

- P (Propagate) = $A \oplus B$
- G (Generate) = $A \cdot B$

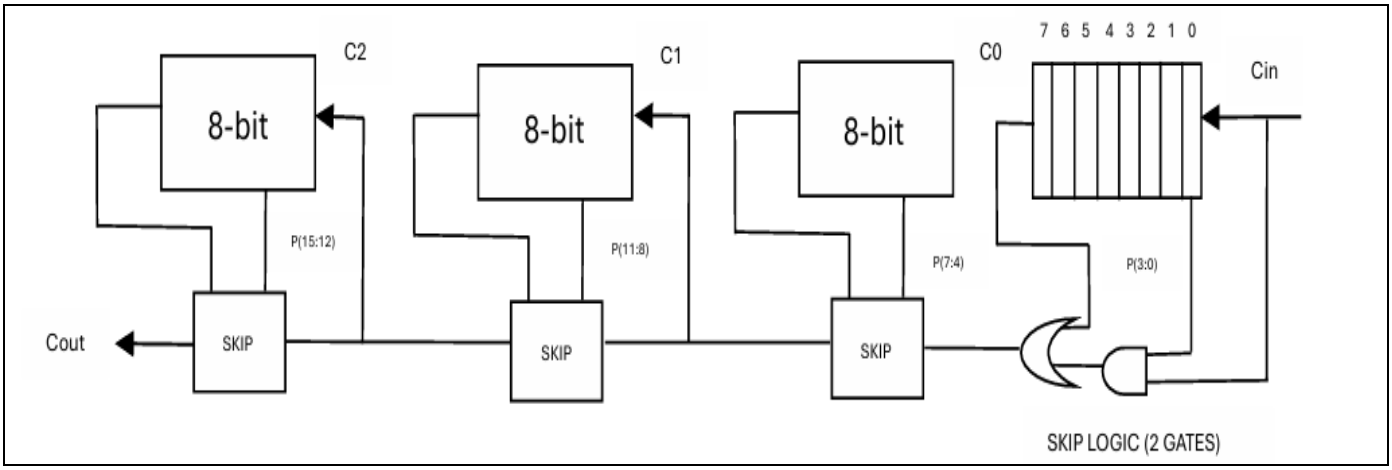

Fig 4 Carry Skip Adder

Carry Select Adder (CSLA)

Two parallel sum computations are done with carry-in = 0 and carry-in = 1, then selected using a multiplexer:

$$S = C_{in} \cdot S1 + (1 - C_{in}) \cdot S0$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1617

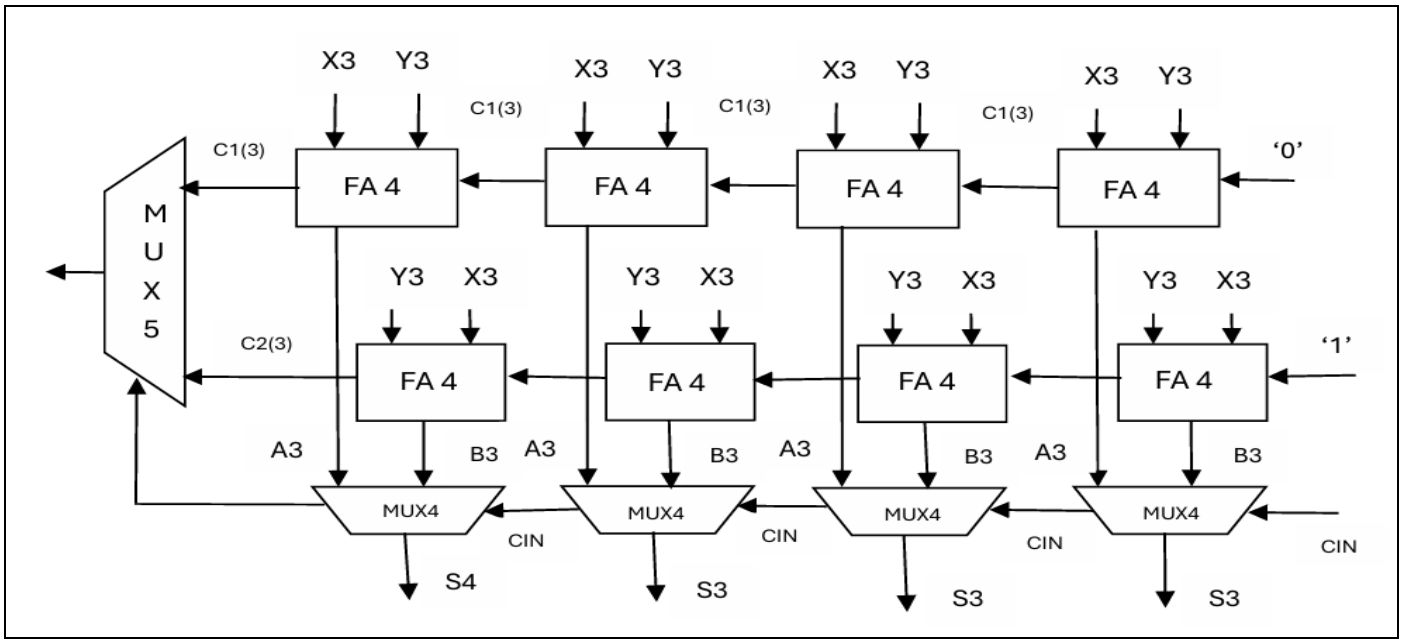

Fig 5 Carry Select Adder

Carry Look-Ahead Adder (CLA)

Carries are computed in advance using:

$$Cout=G+P\cdot Cin$$

Where:

- Generate (G) = A · B
- Propagate (P) = A ⊕ B

This approach significantly reduces delay by avoiding ripple effects.

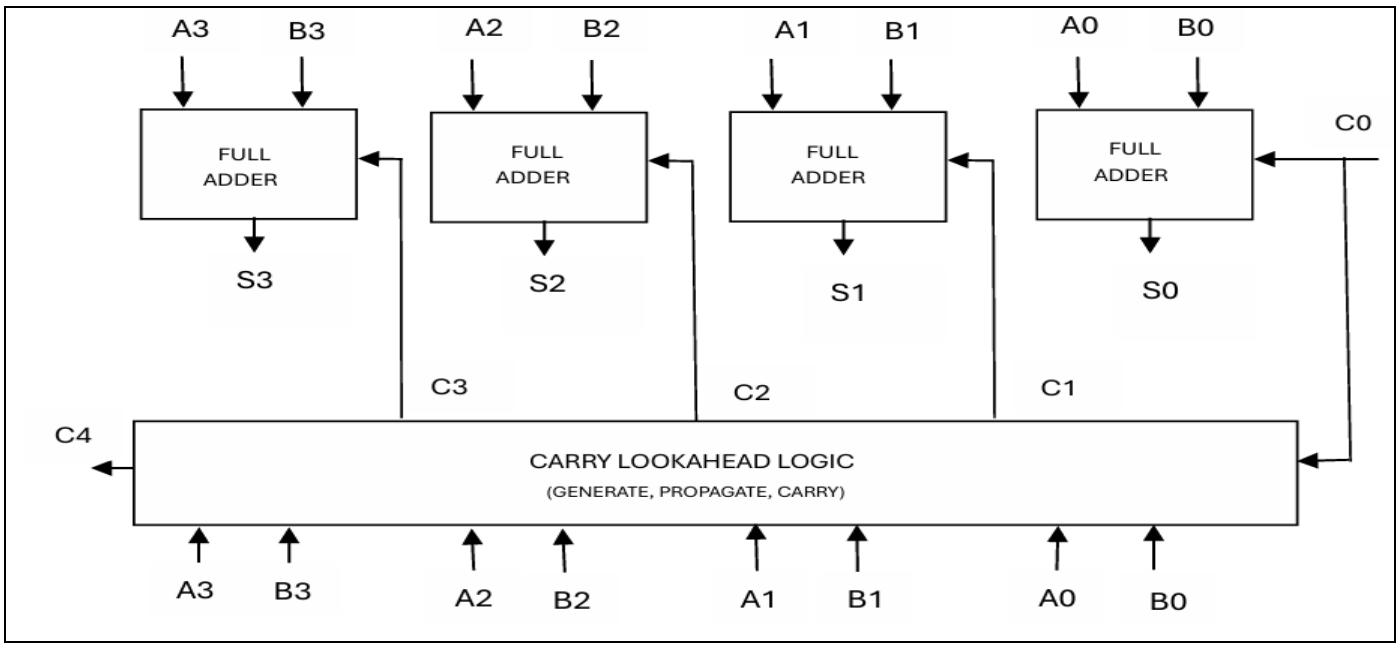

Fig 6 Carry Lookahead Adder

III. IMPLEMENTATION AND SIMULATION

The proposed Hybrid 128-bit Adder was implemented using SystemVerilog, leveraging its capability for modular and hierarchical hardware design. The adder was structured into four distinct 32-bit blocks, each developed using a specific adder architecture: Ripple Carry Adder (RCA), Carry Skip Adder (CSA), Carry Select Adder (CSLA), and Carry

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1617

Look-Ahead Adder (CLA). Each of these modules was written independently and verified for correctness before being integrated into a top-level hybrid module. The integration ensured that carry signals were correctly propagated across each stage, allowing the system to perform full 128-bit addition while balancing speed and resource usage.

The simulation phase was conducted using Synopsys VCS, where the hybrid adder was tested under various scenarios including random inputs, corner cases like all-zeros and all-ones, and alternating bit patterns. A dedicated testbench was developed to apply stimulus vectors and monitor outputs such as the 128-bit sum and carry-out. The simulation results confirmed the functional accuracy of each individual adder as well as the entire integrated system. Waveform analysis was used to track the behavior of carry propagation and verify that pipeline stages were synchronized correctly.

Due to resource limitations of the Xilinx Spartan-6 FPGA, a scaled-down 12-bit version of the hybrid adder was synthesized and implemented on hardware. While the complete 128-bit version was validated through simulation, the 12-bit prototype served as a proof of concept for real-world deployment. The synthesized 12-bit model was implemented using Xilinx ISE, and the RTL schematics were reviewed to confirm proper logic mapping. Timing analysis showed that the pipelined hybrid design achieved better clock performance than a conventional ripple carry implementation, and power consumption remained within acceptable limits. This successful implementation and simulation demonstrate the hybrid adder's viability for use in high-speed arithmetic systems, especially in applications requiring optimized performance and scalability.

➢ *System Verilog Implementation Flow*

The design is structured using modular coding practices, with reusable blocks for each adder type. The following modules are developed:

- full_adder: 1-bit basic building block
- ripple_carry_adder: 32-bit RCA module
- carry_skip_adder: 32-bit CSA module
- carry_select_adder: 32-bit CSLA module
- carry_lookahead_adder: 32-bit CLA module
- hybrid_128bit_adder: Top-level module integrating all four adder types

Each of these modules is implemented with clearly defined input and output ports, with internal logic handling propagation and summation. Special attention is given to minimizing gate delays, reducing redundant logic, and ensuring compatibility across modules.

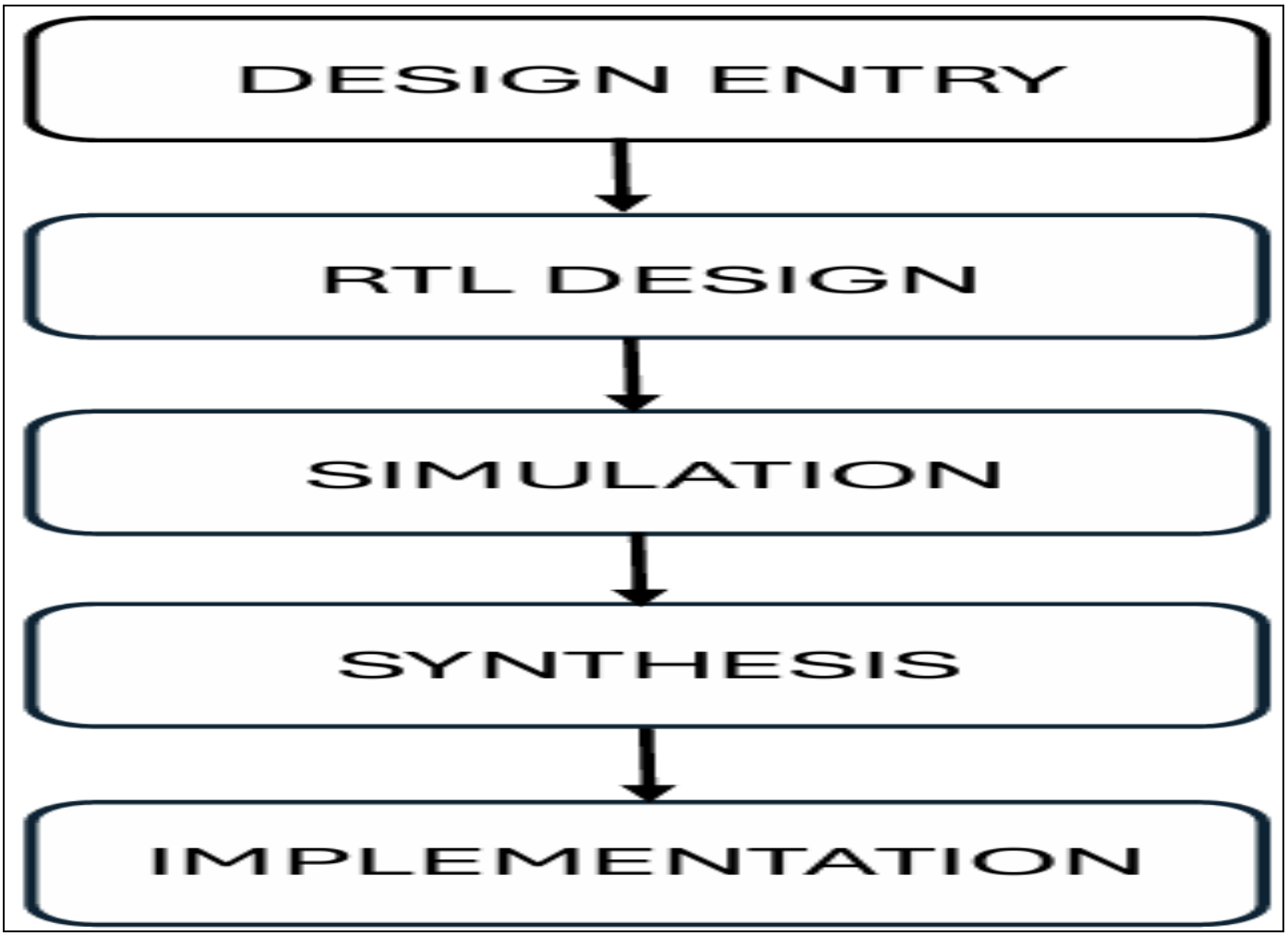

Fig 7 SystemVerilog Implementation Flow

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1617

IV. RESULT AND ANALYSIS

The simulation and synthesis of the Hybrid 128-bit Adder yielded promising results that support the core objective of improving performance through architectural diversity. Functionally, the design passed all test scenarios, producing accurate 128-bit sum and carry-out values across a range of inputs. These inputs included worst-case scenarios such as continuous '1's, '0's, and alternating bits, all of which verified the correctness and reliability of the individual adder modules and the hybrid integration.

Pipelining introduced between the 32-bit blocks helped reduce the overall critical path delay, allowing the circuit to operate at higher frequencies without functional error. The carry propagation across modules occurred seamlessly, aided by the modular pipeline structure and synchronized control signals. The waveform outputs from Synopsys VCS clearly indicated the transitions and timing of sum and carry signals, matching expected results in every test case. No glitches or hazards were observed in the simulation, confirming the timing integrity of the system.

Synthesis was performed on a scaled 12-bit version using Xilinx ISE targeting the Spartan-6 FPGA. Due to resource constraints, the full 128-bit version was not deployed on hardware, but the logic remained consistent with the simulated model. Post-synthesis reports indicated efficient utilization of logic elements, with the pipelined structure reducing register pressure and helping the design meet timing constraints. The use of NAND-based logic further contributed to area reduction and made the design suitable for ASIC migration. Compared to conventional adders implemented separately, the hybrid model showed a balanced improvement in delay, area, and power.

Overall, the results confirm that the hybrid design offers a meaningful compromise between speed and resource efficiency. The use of different adder types, each chosen for their strengths in different bit segments, allowed the design to overcome the limitations faced by homogeneous adder architectures.

Table 1 Comparison of Delay and Performance Table Across Different Adders

Adder Type	Delay(ns)	Power(mW)	Area(Slices)
Ripple Carry Adder	High(39.880ns)	Low	Very Low
Carry Skip Adder	Moderate(8.092ns)	Moderate	Low to Moderate
Carry Select Adder	Medium(20.08ns)	High	High
Carry Lookahead Adder	Low(4.497ns)	Very High	Very High
Hybrid Adder	Balanced(2.833ns)	Moderate	Optimized

V. CONCLUSION, CHALLENGES AND FUTURE ENHANCEMENTS

➢ *Conclusion*:

The Hybrid 128-bit Adder implemented in this project demonstrates a well-balanced approach to achieving high-speed, low-power, and area-efficient arithmetic operations in digital circuits. By strategically dividing the 128-bit operation into four 32-bit segments, each using a different adder architecture—Ripple Carry Adder (RCA), Carry Skip Adder (CSA), Carry Select Adder (CSLA), and Carry Look-Ahead Adder (CLA)—the design successfully exploits the individual strengths of each type. The use of pipelining between segments enables higher clock performance, while the exclusive use of NAND gates contributes to logical simplicity and suitability for ASIC deployment. Simulation and synthesis results affirm the functionality, correctness, and improved performance of the design when compared to traditional, single-architecture adders. This hybrid approach lays a strong foundation for future improvements in wide-bit arithmetic unit design in VLSI and embedded systems.

➢ *Challenges:*

Several challenges were encountered during the development and implementation of the Hybrid 128-bit Adder. One of the primary limitations was the restricted logic capacity of the Spartan-6 FPGA, which could not support a full 128-bit implementation for physical testing. To address this, a scaled-down 12-bit version was developed and tested while keeping the hybrid architectural structure intact. Managing carry propagation across different adder blocks was also a complex task, especially when pipelining was introduced. Accurate timing control and synchronization were essential to prevent data inconsistencies and ensure correct summation. Additionally, implementing the design entirely using NAND gates introduced additional complexity at the gate level, requiring more careful planning and logic reduction techniques. Despite these issues, the project was successfully completed through modular design, staged verification, and thorough simulation.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1617

- *Future Enhancement:*

While the current design meets its goals of optimizing delay, area, and power, several opportunities exist for further improvement. Future work can include expanding the implementation to support higher bit-widths such as 256-bit or 512-bit adders for cryptographic and scientific computing applications. Advanced pipelining techniques could be used to further increase throughput. Dynamic adder selection based on input pattern characteristics may also be explored to enhance performance adaptively. Integrating clock gating and operand isolation methods could help reduce power consumption in idle cycles. Moreover, migrating the design to an ASIC platform using standard cell libraries would provide more accurate performance evaluation in terms of layout area and fabrication feasibility. Incorporating fault tolerance or error detection mechanisms could also improve the reliability of the system for safety-critical applications.

REFERENCES

[1]. Panda, A. K. et al., "High-speed area-efficient VLSI architecture of three-operand binary adder," IEEE Tran. Cir. Sys. I: Reg. Pap., vol. 67, no. 11, 2020, pp. 3944–3953.

[2]. B. Ramkumar, Harish M Kittur, "Low –Power and Area-Efficient Carry Select Adder", IEEE transaction on very large scale integration (VLSI) systems, vol.20, no.2, pp.371-375, Feb 2012.

[3]. Deepthi, K., et al., "Design and Implementation of High-Speed Low-Power Carry Select Adder," In Cog. Inf. Sof. Com., pp. 517 530. Springer, Singapore, 2021.

[4]. H.M. Saddam, F. Arifin, "Design and Evaluation of a 32-bit Carry Select Adder using 4-bit Hybrid CLA Adder," AIUB Journal of Science and Engineering (AJSE), vol. 20, no. 2, 2021, pp. 1–7.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5]. Chu, Y.-Yi, et al., "A High-Speed Carry-Select Adder with Optimized Block Sizes," In 2021 IEEE 15th Int. Conf. Anti-cou., Sec., Ide. (ASID), pp. 182–186. IEEE, 2021.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.