International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

A 32-bit Ripple-Ling Hybrid Carry adder

Mr. V. Narayana Reddy[1], M. Srinidhi[2], P. Balaji[3], P. Sirisha[4], G. Mani Sankar Yadav[5], CH. Mahidhar[6]

[1]*Associate Professor, Dept of ECE, PBR VITS, Kavali, Andhra Pradesh, India.*

[23456] *Students of Department of Electronics and Communication Engineering, PBR Visvodaya Institute of Technology & Science, Kavali (Autonomous), SPSR Nellore (Dt.), Andhra Pradesh – 524201, India*

Abstract - This paper presents the design and implementation of a 32-bit Ripple Carry Hybrid Adder that combines the simplicity of the Ripple-Carry Adder (RCA) with the speed advantages of Multiplexer (MUX)-based adders. In modern digital systems, high-speed arithmetic operations are essential, especially in processors, signal processing, and cryptographic applications, where delay, power consumption, and area are critical design parameters. The conventional RCA suffers from significant propagation delay due to the sequential carry propagation across each full adder stage, limiting its performance in high-speed applications.

To address this limitation, the proposed hybrid architecture integrates a MUX-Based Adder (MBA) to reduce carry propagation delay and improve computational efficiency. The design achieves an effective trade-off between speed and hardware complexity by retaining the structural simplicity of RCA while enhancing performance through multiplexing techniques. The proposed adder is modeled using Verilog HDL. Performance evaluation is carried out based on key metrics such as delay, area utilization, and power consumption. Simulation and synthesis results demonstrate that the hybrid adder significantly reduces computation time compared to the conventional RCA, with optimized resource usage. The proposed design is therefore well-suited for high-speed and energy-efficient arithmetic units in modern digital processors and embedded systems.

Keywords: Ripple-Carry Adder, MUX-Based Adder, Hybrid Adder, Digital Arithmetic, FPGA, High-Speed Computing

---

1. INTRODUCTION

Arithmetic operations are very important in digital systems such as processors, signal processing, and communication systems. Among these, addition is the most basic and frequently used operation. The performance of an adder affects the overall speed, power, and efficiency of the system, so designing a fast and efficient adder is very important. The Ripple-Carry Adder (RCA) is commonly used because of its simple design and low hardware requirement. However, it has a major drawback of high delay, as the carry must pass through each stage one by one. This makes it slow for large bit operations. To reduce this delay, advanced adders are used, but they increase complexity, area, and power consumption. To solve this problem, this paper proposes a 32-bit Ripple Carry Hybrid Adder. This design combines RCA with Multiplexer (MUX)-based adders to improve speed while keeping the design simple. The proposed adder is implemented using Verilog HDL and tested on FPGA. The results show that it provides better speed with efficient area and power usage, making it suitable for high-speed digital applications.

2. LITERATURE SURVEY

Early designs mainly used the Ripple Carry Adder (RCA) due to its simple structure and ease of implementation. However, RCA suffers from high propagation delay because the carry signal must pass through each stage sequentially. This limitation makes it unsuitable for high-speed applications. To overcome this issue, several advanced adder architectures such as Carry Look-Ahead Adder (CLA), Carry Select Adder (CSLA), and Carry Skip Adder were developed. These adders reduce delay by generating carry signals in parallel, but they increase hardware complexity, power consumption, and area.

Further research focused on optimizing RCA designs using different logic styles and circuit techniques. Studies have shown that modifying full adder structures and using efficient logic gates can reduce power and area while maintaining acceptable performance.

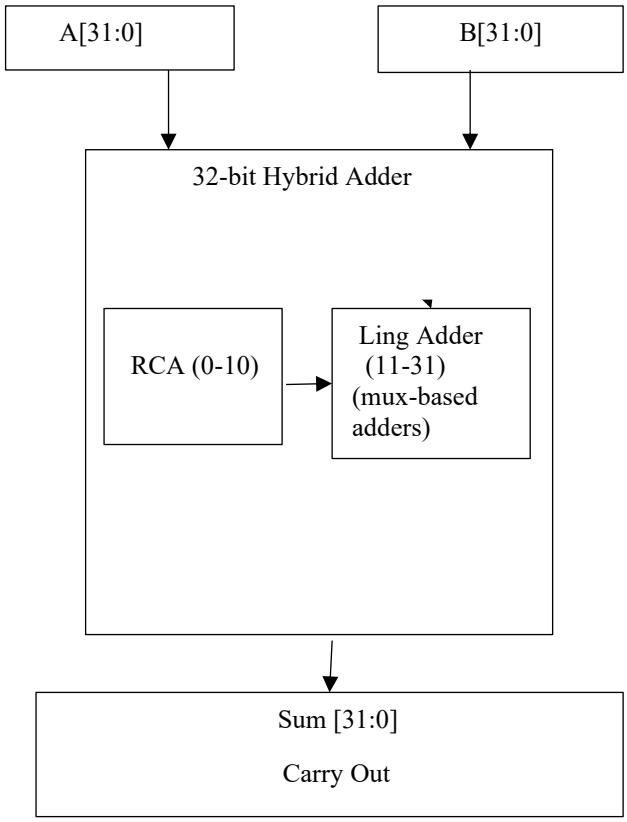

Fig. 1: Block Diagram of Proposed Method Recently,

Multiplexer (MUX)-based adders have gained attention due to their ability to reduce carry propagation delay and improve speed. These designs use multiplexers to select precomputed outputs, leading to better performance with lower delay and improved efficiency. Based on the existing literature, there is a need for an efficient adder design that provides high speed with low power and reduced complexity. This motivates the proposed 32-bit Ripple Carry Hybrid Adder, which combines RCA with MUX-based techniques to achieve improved performance

3. EXISTING SYSTEM

The existing method of the hybrid carry adder combines two techniques: the Ripple-Carry Adder (RCA) and the Ling-based parallel prefix adder. The design is divided into two parts based on bit positions: low-order bits and high-order bits. In the low-order section, a Ripple-Carry Adder is used. This part is simple and requires less hardware because the carry propagates sequentially from one stage to another. Since these bits are not on the critical path, the delay does not affect the overall performance significantly In the high-order section, a Ling-based parallel prefix method is used to reduce delay. This method improves speed by generating carry signals faster compared to RCA, but it increases design complexity. Overall, the existing method tries to balance speed and simplicity by using RCA for lower bits and a faster prefix adder for higher bits. However, it still involves higher complexity and area due to the use of parallel prefix logic.

Limitations: The existing hybrid carry adder, although it improves speed by using Ling-based parallel prefix logic, has several limitations. The design becomes more complex due to the use of advanced prefix structures, making it difficult to implement and understand. It also requires more hardware components, which increases the overall area of the circuit. In addition, the power consumption is higher compared to simpler adder designs like the Ripple-Carry Adder. While the delay is reduced, the increased complexity and resource usage lead to higher design cost. Therefore, the existing method does not provide an optimal balance between speed, area, and power efficiency.

4. PROPOSED SYSTEM

The proposed method shown in Fig.1, presents a 32-bit Ripple Carry Hybrid Adder that improves the performance of traditional adder designs. The architecture is divided into two parts: lower-order bits and higher-order bits.

The lower-order bits are implemented using a Ripple-Carry Adder (RCA) to keep the design simple and area-efficient. The higher-order bits use Multiplexer (MUX)-based adder logic, where outputs are precomputed and selected using multiplexers to reduce carry propagation delay. This combination helps in achieving faster computation compared to conventional RCA while avoiding the complexity of advanced parallel prefix adders. The design ensures a good balance between speed, power, and area. The proposed adder is modeled using Verilog HDL and implemented on FPGA to analyze its performance. The results indicate that the proposed method provides reduced delay and efficient resource utilization. Carry Adder in order to address the issue of the conventional Ling structure utilizing complex logics for the summation of low order bits, this paper presents a new structure that combines the Ling-carry structure for the high-order bits and the ripple carry structure for the low-order bits. By utilizing the low delay of the Ling structure on the critical path and the low cost of the ripple structure on the non-critical path, the proposed adder ensures high-speed computation while reducing hardware consumption.

5. IMPLEMENTATION AND RESULTS

The design has to be synthesized and implemented before it can be checked for correctness, by running functional simulation or downloaded onto the prototyping board. With the top-level Verilog file opened (can be done by double-clicking that file) in the HDL editor window in the right half of the Project Navigator, and the view of the project being in the Module view, the implement design option can be seen in the process view. Design entry utilities and Generate Programming File options can also be seen in the process view. To implement the design, double click the Implement design option in the Processes window. It will go through steps like Translate, Map and Place & Route. If any of these steps could not be done or done with errors, it will place a X mark in front of that, otherwise a tick mark will be placed after each of them to indicate the successful completion After synthesis right click on synthesis and click view text report in order to generate the report of our design.

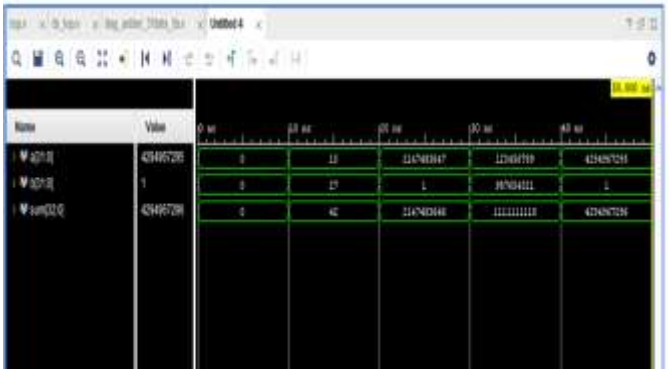

Fig.2: Simulation Report

Fig.2 shows a simulation waveform of a 32-bit Ripple-Ling Hybrid Carry Adder implemented in Verilog. The simulation confirms that the 32-bit Ripple–Ling Hybrid Carry Adder operates correctly for different input combinations. The hybrid approach improves speed compared to a traditional Ripple Carry Adder while maintaining lower complexity than a full Ling adder.

Name	Slice LUTs (134600)	Bonded IOB (400)
top	86	97

Fig. 3: Area Report

Fig. 3 shows the area report of the proposed 32-bit Ripple–Ling Hybrid Carry Adder is evaluated in terms of FPGA resource utilization. The design uses 86 Slice LUTs out of the available 134600 LUTs, indicating very low logic resource consumption. Additionally, it utilizes 97 Bonded I/O Blocks (IOBs) out of the total 400 available IOBs. The low number of LUTs used demonstrates that the proposed hybrid architecture is area-efficient. This is mainly due to the use of a Ripple Carry Adder (RCA) in the lower-order bits, which reduces hardware complexity. At the same time, the higher-order bits use optimized logic (Ling/MUX-based structure), ensuring improved speed without significantly increasing the area.

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay
Path 1	∞	11	12	8	b[13]	sum[31]	9.218
Path 2	∞	11	12	8	b[13]	sum[32]	8.983
Path 3	∞	11	12	8	b[13]	sum[30]	8.964
Path 4	∞	11	12	8	b[13]	sum[29]	8.955
Path 5	∞	11	12	8	b[13]	sum[28]	8.613
Path 6	∞	11	12	8	b[13]	sum[26]	8.605
Path 7	∞	11	12	8	b[13]	sum[27]	8.605
Path 8	∞	10	11	8	b[13]	sum[25]	8.161
Path 9	∞	10	11	6	a[0]	sum[22]	7.861
Path 10	∞	10	11	6	a[0]	sum[23]	7.861

Fig. 4: Delay Report

Fig. 4 shows the timing report of the proposed 32-bit Ripple–Ling Hybrid Carry Adder shows the critical path delays and signal propagation characteristics. The analysis includes parameters such as logic levels, routing paths, fanout, and total delay. From the report, the maximum delay observed is 9.218 ns, which corresponds to the critical path from input b[13] to output sum [31]. Other paths exhibit slightly lower delays, such as 8.983 ns, 8.964 ns, and 8.955 ns, indicating consistent performance across different output bits. Most of the critical paths have 11 logic levels and 12 routing stages, showing the depth of combinational logic involved. The high fanout value of 8 in several paths indicates that certain signals drive multiple outputs, contributing to delay. Some lower paths have reduced delay (7.861 ns) due to fewer logic levels and lower fanout.

The hybrid design helps in reducing delay compared to a conventional Ripple Carry Adder. The lower bits use RCA, while the higher bits use optimized Ling/MUX-based logic, which minimizes carry propagation delay and improves overall speed.

Fig.5 shows the power report of the adder By integrating the simplicity of ripple carry propagation with optimized hybrid carry generation techniques, the design effectively

reduces unnecessary switching activity, which directly contributes to lower dynamic power consumption. The architecture selectively accelerates carry computation in critical paths while

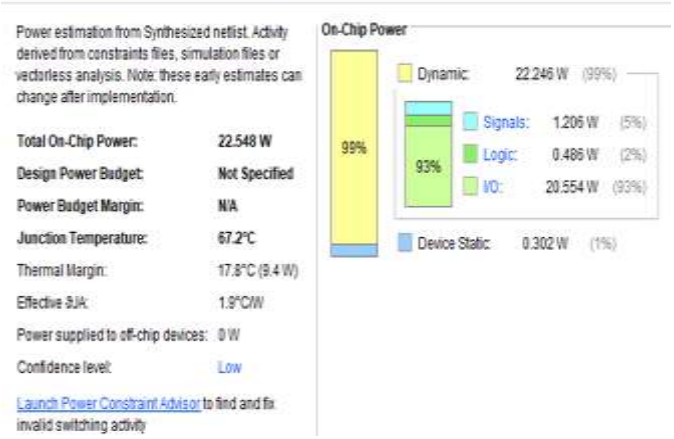

Fig. 5: Power Report

preserving a compact layout in non-critical sections, resulting in a balanced trade-off between delay and hardware complexity. Simulation results indicate that the proposed adder achieves reduced power dissipation compared to conventional ripple carry and standard hybrid adders, making it well-suited for low-power and energy-constrained applications such as portable and embedded systems.

6. DISCUSSION

The proposed hybrid adder correctly performs 32-bit addition and generates accurate sum and carry outputs. The design demonstrates improved efficiency in terms of propagation delay while keeping hardware complexity within acceptable limits. The results also indicate that the hybrid approach offers a good balance between speed, area, and power consumption. The performance analysis indicates that the proposed architecture offers improved propagation delay and better operational efficiency. It also maintains reasonable power consumption and circuit area, which are important factors in digital circuit design. These improvements make the hybrid adder suitable for use in high-speed digital systems such as microprocessors, arithmetic logic units (ALUs), digital signal processing systems, and embedded applications.

7. CONCLUSION

A high-order Ling and low-order ripple hybrid carry adder is proposed in this project. The low order 11 bits use a ripple-carry structure instead of the conventional lookahead carry method, while the high order 21 bits continue to use Ling carry structure, thereby simplifying the low-order sum circuit while maintaining the critical path of the entire adder. Moreover, new intermediate variables are used as the object for Shannon expansion in the implementation of the output sum circuit, and simplified custom logic circuits are used to optimize the design of each bit's sum circuit.

In conclusion, the project successfully demonstrates the design and analysis of an optimized 32-bit hybrid carry adder. The proposed architecture offers improved speed and efficient operation, making it a useful component for high-speed computing applications. This work also provides a foundation for further research and development in advanced adder architectures and high-performance digital circuit design.

ACKNOWLEDGEMENT

The authors sincerely thank Mr. V. Narayana Reddy (Associate Professor, ECE, PBR VITS Kavali) for his guidance, Dr. R. Sravanthi (Professor & HoD, ECE) for providing facilities, and Dr. V. Anil Kumar (Principal, PBR VITS Kavali) for the academic environment that enabled this work.

REFERENCES

[1] Y. He and C.-H. Chang, "A power-delay efficient hybrid carry look ahead/carry select based redundant binary to two's complement converter," IEEE Transistor Circuits Syst. I, Reg. Papers, vol. 55, no. 1, pp. 336–346, Feb. 2008.

[2] A. M. Shams, T. K. Darwish, and M. A. Bayumi, "Performance analysis of low power 1-bit CMOS full adder cells," IEEE Trans. Very Large Scale Integration. (VLSI) Syst., vol. 10, no. 1, pp. 20–29, Feb. 2002.

[3] R. Zlatanovici and B. Nikolic, "Power-performance optimal 64-bit carry look - a ahead adders," in Proc. 29th Eur. Solid-State Circuits Conf., Estoril, Portugal, 2003, pp. 321–324

[4] H. Naseri and S. Timarchi, "Low-power and fast full adder by exploring new the XOR and XNOR gates," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1481–1493, Aug. 2018.

[5] P. Bhattacharyya, B. Kundu, S. Ghosh, V. Kumar, and A. Dandapat, "Performance analysis of a low-power high-speed hybrid 1-bit full adder circuit," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 10, pp. 2001– 2008, Oct. 2015.

[6] S. Goel, A. Kumar, and M. A. Bayoumi, "Design of robust, energyefficient full adders for deep-submicrometer design using hybrid-CMOS logic style," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 12, pp. 1309–1321, Dec. 2006.

[7] K. Papachatzopoulos and V. Paliouras, "Static delay variation models for ripple- carry and borrow-save adders," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 7, pp. 2546–2559, Jul. 2019.

[8] N. Kaushik and S. Bodapati, "IMPLY-based high-speed conditional carry and carry select adders for in-memory computing," IEEE Trans. Nanotechnol., vol. 22, pp. 280–290, 2023.

[9] B. R. Zeydel, D. Baran, and V. G. Oklobdzija, [9] "Energy-efficient design methodologies: High-performance VLSI adders," IEEE J. Solid-State Circuits, vol. 45, no. 6, pp. 1220–1233, Jun. 2010.

[10] D. Esposito, D. De Caro, E. Napoli, N. Petra, and A. G. M. Strollo, "Variable latency speculative Han–Carlson adder," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 5, pp. 1353–1361, May 2015

[11] H. Ling, "High-speed binary adders," IBM J. R&D, vol. 25, nos. 2–3, pp. 156– 166, May 1981.

[12] C. Efstathiou, Z. Owda, and Y. Tsiatouhas, "New high-speed multioutput carry look-ahead adders," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 10, pp. 667–671, Oct. 2013.

[13] A. Mitra and A. Bakshi, "Design of a high speed adder," Int. J. Sci. Eng. Res., vol. 6, no. 4, pp. 918–921, Apr. 2015.

[14] G. Dimitrakopoulos and D. Nikolos, "High-speed parallel-prefix VLSI Ling adders," IEEE Trans. Comput., vol. 54, no. 2, pp. 225–231, Feb. 2005.

[15] N. T. Quach and M. J. Flynn, "High-speed addition in CMOS," IEEE Trans. Comput., vol. 41, no. 12, pp. 1612–1615, Dec. 1992.

[16] Y. Wang, C. Pai, and X. Song, "The design of hybrid carrylookahead/carry-select adders," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 49, no. 1, pp. 16–24, Jan. 2002.

[17] D. Esposito, D. De Caro, and A. G. M. Strollo, "Variable latency speculative parallel prefix adders for unsigned and signed operands," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 63, no. 8, pp. 1200–1209, Aug. 2016.

[18] Y. Choi and E. E. Swartzlander, "Speculative carry generation with prefix adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 3, pp. 321–326, Mar. 2008.

[19] J. Wang, G. Liu, Y. Chen, and S. Wang, "Construction and analysis of SHA-256 compression function based on chaos S-Box," IEEE Access, vol. 9, pp. 61768– 61777, 2021.

[20] B. K. Mohanty and S. K. Patel, "Area–delay–power efficient carryselect adder," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 6, pp. 418–422, Jun. 2014.