International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 03 (March)2026
Received: 10-1-2026 Accepted: 17-2-2026 Published: 30-3-2026

An Analysis of Parallel Prefix Adders in Radix-2 and Multi-Value Logic: A Review

Subhashini Kodur,
Dept. of Electronics and Communication Engineering,
Annamacharya Institute of Technology and Sciences Kadapa, India.
kodursubha@gmail.com

T. Naga sushma,
Dept. of Electronics and Communication Engineering,
Annamacharya Institute of Technology and Sciences Kadapa, India.
sushmatelugu25@gmail.com

U. Naga Mallika,
Dept. of Electronics and Communication Engineering,
Annamacharya Institute of Technology and Sciences Kadapa, India.
upparenagamallika@gmail.com

L. Arun Kumar,
Dept. of Electronics and Communication Engineering,
Annamacharya Institute of Technology and Sciences Kadapa, India.
arunkumarlakkam66@gmail.com

P. Kalyani,
Dept. of Electronics and Communication Engineering,
Annamacharya Institute of Technology and Sciences Kadapa, India.
kalyanipuduru96@gmail.com

***Abstract*—Adders are crucial parts of digital systems that impact area, power and speed. Parallel Prefix Adders (PPAs), which are renowned for their quick carry calculations employing parallel structures, are reviewed in this study. Performance measures, are reviewed in this study. Performance measures are used to compare architectures like Kogge-Stone, Brent-Kung, Sklansky, Ladner-Fischer, and Han-Carlson. The study also looks into ternary logic, which, in contrast to binary logic, increases data density and decreases interconnections. Both binary and ternary PPA design approaches are covered. Ternary PPAs offer, possible benefits in speed and compactness with more complexity, but binary PPAs give dependable performance. In general, ternary PPAs exhibit potential for high-performance VLSI systems in the future.**

Keywords*—Parallel Prefix Adder (PPA), Binary Logic, Ternary Logic, Kogge–Stone Adder, Brent–Kung Adder, VLSI Design, CNFET, High-Speed Arithmetic.

I. INTRODUCTION

Adders are frequently employed in arithmetic and logic operations and are crucial parts of digital systems. Microprocessors, DSPs, image processing units, and VLSI designs frequently use them. The efficiency of the adder circuits in terms of speed, power consumption, and space has a major impact on these systems' performance. Designing quick and energy-efficient adders has grown in significance as deep submicron technology has advanced [1-2], [9]. Digital circuits frequently employ conventional adder designs like Carry Look-Ahead Adders (CLA) and Ripple Carry Adders (RCA), although they have certain drawbacks. RCA is straightforward and takes up less space, but because the carry must spread gradually, it is sluggish. Because it generates carries in parallel, CLA is faster; nonetheless, it increases fan-in, circuit complexity, and power consumption. More sophisticated adder designs are required as a result of these restrictions [3], [10].

Because they can calculate carry signals in logarithmic time, Parallel Prefix Adders (PPAs) are among the quickest adder designs. To effectively process, create, and propagate signals across stages, they employ a structure resembling a tree. Popular varieties include the Skalsky Adder, which strikes a compromise between fan-out and delay, the Brent-Kung Adder, which is more area-efficient with less wire, and the Kogge-Stone Adder, which provides extremely high speed with low delay. However, particularly for large bit sizes, PPAs can result in increased power consumption and wiring complexity [4-5], [11], [12]. Multivalued logic, particularly ternary logic, has become more popular as a substitute for conventional binary logic in recent years. Ternary logic allows for the representation of more data with fewer connections by using three levels (0, 1, and 2). Overall efficiency and circuit density are enhanced as a result. Additionally, it can lessen switching activity, which in VLSI circuits reduces dynamic power consumption [6], [13].

Ternary logic systems are now simpler to implement thanks to advances in nanotechnology. The shortcomings of conventional CMOS technology are partially addressed by devices such as Carbon Nanotube Field Effect Transistors (CNFETs). CNFETs are appropriate for multivalued logic systems because they have advantages including high carrier mobility, near-ballistic transport, and changeable threshold voltage by altering the nanotube diameter. Ternary logic circuits perform better thanks to these characteristics [7-8], and [14]. In this regard, a viable method of achieving high-speed and low-power arithmetic units is to combine Parallel Prefix Adder (PPA) designs with ternary logic. But there are obstacles that must be properly addressed, such as fabrication problems, maintaining stable logic levels, and increased circuit complexity [15].

II. OVERVIEW OF PARALLEL PREFIX ADDER

In contemporary digital systems, parallel prefix adders (PPAs) are regarded as one of the fastest adder designs [16]. By computing carry signals in parallel, they are intended to solve the delay problems of conventional adders. PPAs use a tree-like structure to cut down on computation time, in contrast to Ripple Carry Adders, where the carry goes step by step [17]. For every bit, PPAs produce (G) and propagate (P) signals. Pre-processing (calculating G and P), prefix computation (using a tree structure to generate carry signals), and post-processing (calculating the final sum) are the three phases of the operation. PPAs are frequently utilized in microprocessors, DSPs, and VLSI systems due to their quick performance and effective carry calculation [18]. They may, however, need more wiring and be more complicated. All things considered, PPAs are crucial for accomplishing fast arithmetic operations in digital circuits.

A. *Types of parallel prefix adder:*

There are five types of parallel prefix adders. They are

1. Kogge–Stone Adder (KSA)
2. Brent–Kung Adder (BKA)
3. Sklansky Adder
4. Ladner–Fischer Adder
5. Han–Carlson Adder

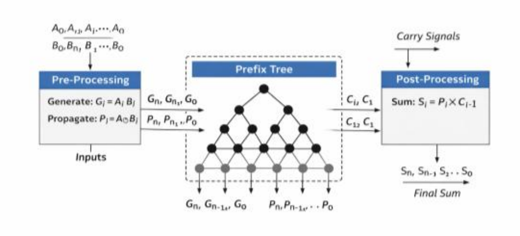

Fig.1. Architecture of Parallel Prefix Adder [16]

III. BINARY LOGIC-BASED PARALLEL PREFIX ADDERS

Digital systems are built on binary logic, which uses two values to represent data: 0 and 1, which stand for OFF/ON or FALSE/TRUE states. Logic gates like AND, OR, and NOT, which carry out fundamental operations on binary inputs, are used in its implementation. Digital circuit design and simplification are made possible by binary logic, which is based on Boolean algebra created by George Boole. It is the foundation of contemporary computer and VLSI architecture and is extensively employed in arithmetic operations, data processing, and control systems [10], [15].

Design Methodology:

By computing carry signals in parallel rather than sequentially, binary logic-based Parallel Prefix Adders (PPAs) are intended to do high-speed arithmetic. PPAs are among the fastest adder architectures in digital systems because of this parallelism, which drastically lowers delay [18], [19]. The Generate (G) and Propagate (P) signals that are generated from input bits are the foundation of PPA operation. These signals serve as the foundation for effective carry computation by determining whether a carry is formed or propagated to the subsequent step [20].

There are three primary phases of PPA design:

A. *Pre-processing Stage:*

Using basic logic operations like AND and XOR, signals are generated and propagated for each bit in this stage. In this step, inputs are prepared for parallel carry computation with the least amount of latency [20].

B. *Prefix Computation Stage:*

A prefix tree structure is used in this crucial stage to create carry signals. The prefix operator efficiently computes carries by combining (G, P) pairs. Compared to traditional adders, this greatly improves performance by enabling carry computation in logarithmic time [18], [21]. Different prefix structures, including Brent-Kung and Kogge-Stone, offer different trade-offs in complexity, area, and latency [19], [21].

C. *Post-processing Stage:*

Using the propagate signals and previously generated carry values, the sum bits are computed in this last stage. This stage adds very little latency to the overall design and usually employs XOR operations [20].

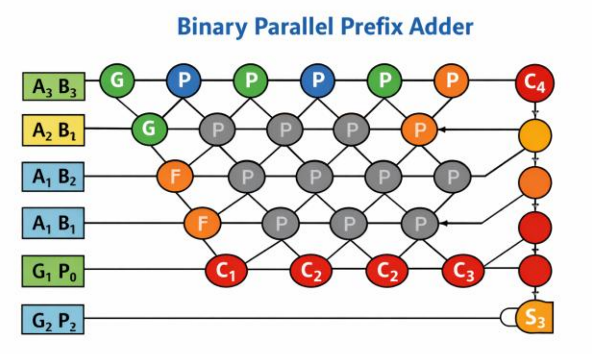

Fig.2. 4-bit Binary Parallel Prefix Adder [18]

IV. TERNARY LOGIC-BASED PARALLEL PREFIX ADDERS

In contrast to binary systems, which typically employ two logic levels, ternary logic employs three. In order to store and analyze more information in a single digit, these levels are commonly represented as 0, 1, and 2. This lowers the number of interconnections needed in digital circuits and increases data efficiency. It is utilized in sophisticated computing applications and is based on multi-valued logic techniques first developed by Stephen Cole Kleene. Despite being more difficult to implement than binary logic, ternary logic has advantages in VLSI and nanoelectronics because of its compact architecture and increased computing capability [6], [13].

Design Methodology:

While the construction of ternary logic-based PPAs is similar to that of binary PPAs, multi-valued logic operations are incorporated. Depending on the logic encoding, ternary systems frequently employ extended signal representations like generate, propagate, and intermediate states in place of basic create and propagate signals [22], [23].

Three primary phases comprise the design:

A. *The pre-processing phase*

Ternary equivalents of carry-generation signals are created by processing input trits. These signals indicate whether a carry is partially transferred, created, or propagated [22].

B. *Stage of Prefix Computation:*

Carry signals are computed in parallel using a ternary prefix tree structure. Due to the various logic levels, the prefix operation is more complicated than binary, although logarithmic delay performance is still possible. Ternary signals can be effectively combined using modified prefix operators [22-23].

C. *Stage of post-processing:*

The generated carry values and input signals are used to calculate final sum trits. For base-3 addition, the arithmetic rules are modified [23].

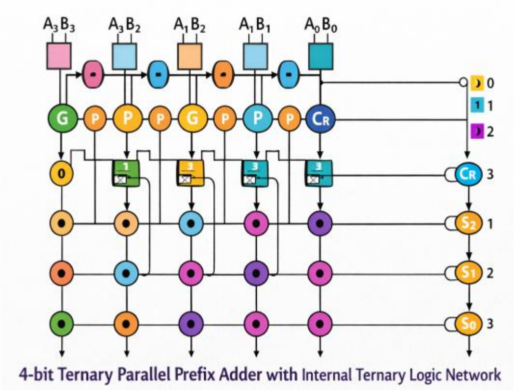

Fig. 3. 4-bit Ternary Parallel Prefix Adder with Internal Ternary Logic Network [22]

V. RESULT AND DISCUSSIONS

Parallel Binary Prefix Delay, area, power, and fan-out are used to compare adders (PPAs). Although it uses a lot of space and power, the Kogge-Stone Adder (KSA) offers the fastest performance with the least amount of delay. Although it has a moderate delay, the Brent-Kung Adder (BKA) is more energy and space efficient. Although it has a minimal delay, the Sklansky Adder has a significant fan-out. A balanced trade-off between all factors is offered by the Ladner-Fischer Adder. Because of its hybrid construction, the Han-Carlson Adder strikes a solid balance between power, speed, and area. In general, design specifications like efficiency or speed influence the adder selection [18].

TABLE. I. Performance Comparison of Binary Parallel Prefix Adders [18]

Adder Type	**Delay**	**Area**	**Power Consumption**	**Fan-out**	**Key Feature**
Kogge–Stone (KSA)	Very Low	Very High	High	Low	Fastest adder with minimum delay
Brent–Kung (BKA)	Moderate	Low	Low	Low	Area-efficient design
Sklansky	Low	Medium	Medium	High	Minimum logic depth
Ladner–Fischer	Moderate Low	Medium	Medium	Moderate	Balanced performance
Han–Carlson	Low	Medium	Medium	Low–Moderate	Hybrid (speed + area optimization)

TABLE. II. Performance Comparison of Ternary Parallel Prefix Adders [22]

Adder Type	**Delay**	**Area**	**Power Consumption**	**Fan-out**	**Key Feature**
Kogge –Stone	Very Low	Very High	High	Low	Fastest ternary adder with parallel carry
Brent–Kung	Moderate	Low	Moderate	Low	Area-efficient ternary design
Sklansky	Low	Medium	Medium	High	Minimum computation stages
Ladner–Fischer	Moderate–Low	Medium	Medium	Moderate	Balanced speed and complexity
Han–Carlson	Low	Medium	Medium–High	Low–Moderate	Hybrid structure for optimized

The table contrasts various ternary Parallel Prefix Adders (PPAs) according to fan-out, power consumption, area, and delay. Because of its extremely parallel structure, the Ternary Kogge-Stone Adder has the lowest delay, but it uses a lot of space and power. With a moderate delay and power consumption, the Ternary Brent-Kung Adder is more area-efficient. The Ternary Sklansky Adder has a high fan-out but low delay. A fair trade-off between speed, area, and complexity is provided by the Ternary Ladner–Fischer Adder. Because it is a hybrid design, the Ternary Han-Carlson Adder offers an excellent compromise between resource utilization and performance [22].

TABLE. III. Performance Comparison of Binary vs Ternary Parallel Prefix Adders [6]

Parameter	**Binary PPAs**	**Ternary PPAs**
Radix / Logic Levels	2 (0, 1)	3 (0, 1, 2)
Number of Digits	Higher	Lower (more information per digit)
Computational Stages	More stages	Fewer stages (due to higher radix)
Delay	Low	Potentially lower (theoretical)

Circuit Complexity	Moderate	High (multi-valued logic design)
Power Consumption	Moderate	Higher (complex gate structures)
Wiring Complexity	High	Reduced (fewer interconnections)
Noise Margin	High (stable)	Lower (more sensitive to noise)
Implementation Ease	Easy (well-established CMOS technology)	Difficult (limited practical support)
Overall Efficiency	Balanced	Trade-off between compactness and complexity

Parallel Prefix in Binary and Ternary The primary differences between adders (PPAs) are in their logic representation and performance attributes. Ternary PPAs employ three logic levels (0, 1, 2), enabling more information density and fewer digits, whereas binary PPAs only use two logic levels (0 and 1). As a result, ternary PPAs may achieve reduced delay and require fewer computational processes. In contrast to binary PPAs, ternary PPAs have a more complicated circuit design, which results in increased power consumption and smaller noise margins. Ternary PPAs have practical difficulties because of limited hardware support, whereas binary PPAs enjoy the advantages of well-established CMOS technology and simpler implementation. While ternary PPAs offer compact representation and some speed improvements at the expense of greater complexity, binary PPAs often give a balanced and dependable solution [6].

VI. CONCLUSION

This study compared Parallel Prefix Adders (PPAs) based on binary and ternary logic. Binary PPAs offer a dependable and tried-and-true solution with balanced speed, area, and power performance. Ternary PPAs, on the other hand, have benefits including increased data density, fewer interconnections, and possible delay reduction. Nevertheless, they have problems with implementation, higher power consumption, and more complex circuits. Overall, ternary PPAs have great potential for high-performance and compact VLSI applications in the future, especially with developments in nanotechnology like CNFETs, even though binary PPAs are still the preferred option for modern digital systems.

REFERENCES

[1] N. L. V. Calazans and F. G. Moraes, “Integrating the teaching of computer organization and architecture with digital hardware design early in undergraduate courses,” *IEEE Transactions on Education*, vol. 45, no. 2, pp. 109–116, May 2002.

[2] Y. M. Aneesh and B. Bindu, “A physics-based single event transient pulse width model for CMOS VLSI circuits,” *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 2, pp. 304–311, Jun. 2020.

[3] R. P. Brent and H. T. Kung, “A regular layout for parallel adders,” *IEEE Transactions on Computers*, vol. C-31, no. 3, pp. 260–264, Mar. 1982.

[4] P. M. Kogge, “Parallel solution of recurrence problems,” *IBM Journal of Research and Development*, vol. 18, no. 2, pp. 138–148, Mar. 1974.

[5] J. Sklansky, “Conditional-sum addition logic,” *IRE Transactions on Electronic Computers*, vol. EC-9, no. 2, pp. 226–231, Jun. 1960.

[6] K. C. Smith, “The prospects for multivalued logic: A technology and applications view,” *IEEE Transactions on Computers*, vol. C-30, no. 9, pp. 619–634, Sep. 1981.

[7] A. Doostaregan and A. Abrishamifar, “Evaluating a methodology for designing CNFET-based ternary circuits,” *Circuits, Systems, and Signal Processing*, vol. 39, no. 6, pp. 3030–3048, Jun. 2020.

[8] S. Jogad, S. A. Loan, and N. Afzal, “Implementation of CNFET based negative gain digitally programmable current conveyor,” in *Proc. 5th Int. Conf. on Devices, Circuits and Systems (ICDCS)*, 2022, pp. 1–6.

[9] S. Borkar, “Design challenges of technology scaling,” *IEEE Micro*, vol. 22, no. 4, pp. 23–29, Jul./Aug. 2002.

[10] S. Harris and D. Harris, *Digital Design and Computer Architecture: RISC-V Edition*. Cambridge, MA, USA: Morgan Kaufmann, 2021.

[11] A. K. Panda, R. Palisetty, and K. C. Ray, “High-speed area-efficient VLSI architecture of three-operand binary adder,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 11, pp. 2553–2557, Nov. 2020.

[12] A. S. Shastri, “VLSI circuits and approximate computing,” in *Optimization Methods in VLSI Design: A Machine Learning Perspective*. Cham, Switzerland: Springer, 2025, pp. 1–20.

[13] M. D. Miller and M. A. Thornton, “Multiple-valued logic applications,” in *Multiple Valued Logic: Concepts and Representations*. Cham, Switzerland: Springer, 2022, pp. 321–360.

[14] P. Avouris, J. Appenzeller, R. Martel, and S. J. Wind, “Carbon nanotube electronics,” *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1772–1784, Nov. 2003.

[15] F. Merchant, “Computer arithmetic,” in *Handbook of Computer Architecture*. Cham, Switzerland: Springer, 2024, pp. 1–25.

[16] P. M. Kogge and H. S. Stone, “A parallel algorithm for the efficient solution of a general class of recurrence equations,” *IEEE Transactions on Computers*, vol. C-22, no. 8, pp. 786–793, Aug. 1973.

[17] J. Jiang, C. Chen, Y. Zhou, W. Han, L. Liu, and Q. Pei, “A parallel attention-based framework for multi-step multivariate runoff forecasting in mountainous watersheds: Wuyuan case study,” *Journal of Hydrology*, Elsevier, 2026.

[18] D. Harris, “A taxonomy of parallel prefix networks,” in *Proc. 37th Asilomar Conf. on Signals, Systems and Computers*, 2003, pp. 2213–2217.

[19] A. Raju, R. Patnaik, and R. K. Babu, “Parallel prefix adders—A comparative study for fastest response,” in *Proc. Int. Conf. on Communication and Signal Processing (ICCSP)*, 2016, pp. 2041–2045.

[20] S. Butchibabu and S. K. Bab, “Design and implementation of efficient parallel prefix adders on FPGA,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 6, pp. 1–5, Jun. 2014.

[21] S. Rakesh and K. S. V. Grace, “A comprehensive review on the VLSI design performance of different parallel prefix adders,” *Materials Today: Proceedings*, vol. 18, pp. 4590–4596, 2019.

[22] S. V. Yamani, H. K. R. V. Kudulla, B. Ganesh, *et al*., “Design of an energy efficient ternary parallel prefix carry/sum propagate adders using 32-nm CNTFET,” *Journal Name*, 2025.

[23] S. V. Yamani, H. K. R. V. Kudulla, B. Ganesh, *et al*., “Design of an energy efficient ternary parallel prefix carry/sum propagate adders using 32-nm CNTFET,” *Microelectronics Journal*, Elsevier, 2025.