Design and Implementation of Parallel Prefix Based 16-Bit Kogge Stone Adder for High-Speed Binary Addition

Vipul Agarwal, Gurrampati Preethi Reddy(✉), and Guduru Balaji Induja

Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India
preethireddygurrampati@gmail.com

Abstract. This paper explores fundamental operations in digital systems, focusing on addition, subtraction, and multiplication. Out off these operations, addition plays a crucial role in digital, analog, and control systems, with the efficiency of additional components directly impacting the speed and accuracy of these systems. In Very Large-Scale Integration (VLSI) advancements, the primary emphasis lies in enhancing speed and reducing spatial constraints in system design, particularly within adders. Various adder designs have undergone extensive research in recent years, including Carry Prefix Tree Adders, Parallel Ripple Adders, Carry Skip Adders, and Carry Look-Ahead Adders. While tree-based adders exhibit simultaneous carry emergence for swift computations, this speed advantage often comes at the cost of increased power usage and a larger physical footprint. The paper aims to introduce a 16-bit Kogge-Stone adder, renowned for its rapid, parallel binary addition capabilities. This design offers advantages such as speed, consistency, and scalability, making it particularly suitable for scenarios requiring fast binary number addition. Operating on a parallel prefix adder concept optimized for binary addition tasks, the Kogge-Stone adder leverages parallelism efficiently, positioning it as a top choice for applications prioritizing speed and efficiency, including digital circuits and processors. Adders play a significant role in various arithmetic and logical operations, with Parallel Prefix Adders standing out as vital and efficient circuits for binary addition tasks. The paper delves into the design and performance evaluation of Kogge Stone Parallel Prefix Adders, utilizing different design methodologies such as CMOS (Complementary Metal Oxide Semiconductor) and GDI (Gate Diffusion Input). The design and simulation of these logic gates were executed using EDA Playground, specifically leveraging the Virtuoso and ADE Environment at GPDK 180 nm technology.

Keywords: Carry Prefix adder · Parallel Ripple adder · Gate Diffusion Input (GDI) · Efficiency · Kogge stone adder

P. Pareek et al. (Eds.): IC4S 2024, LNICST 599, pp. 323–332, 2025.
https://doi.org/10.1007/978-3-031-77081-4_26

1 Introduction

Adders and Subtractors are vital components of any computational system [1]. The Kogge-Stone adder belongs to the class of parallel prefix adders [2] specifically crafted for swift binary number addition. Kogge-Stone adder is renowned for its efficient and high-speed operation, making it a popular choice in digital design for arithmetic units, such as in CPUs and signal processing systems. In digital circuits, adders serve as foundational elements primarily for arithmetic tasks like addition. Among these, the Kogge-Stone adder stands out within the realm of parallel prefix adders (see Fig. 1), recognized for their optimized speed and hardware simplicity.

In Fig. 1, the foundational concept of the Kogge-Stone adder revolves around leveraging parallelism during carry computation. Unlike conventional ripple-carry adders where the carry ripples through each bit stage one after the other, causing delays, parallel prefix adders, including the Kogge-Stone variant, strive to minimize this delay through simultaneous carry computations. The Kogge-Stone adder architecture typically consists of multiple layers of interconnected processing elements, arranged in a hierarchical fashion [3]. Each layer performs partial addition and carry generation, with the results cascading through the layers until the final sum and carry-out are obtained. This hierarchical structure enables the adder to efficiently handle large word sizes, scaling well with the size of the operands.

Functioning on a tree-like structure, the Kogge-Stone adder designates each node to represent a distinct bit block [4]. This tree arrangement facilitates concurrent carry value calculations, enhancing the overall addition speed. Typically, its implementation employs a consistent array of processing units, promoting efficient parallel processing. A standout feature of the Kogge-Stone adder is its reliably low worst-case delay, making it apt for rapid arithmetic tasks. Nonetheless, its hardware complexity might escalate, especially for narrower bit-widths, due to the inherent intricacies linked with parallel processing components. Overall, the Kogge-Stone adder's classification as a parallel prefix adder underscores its reliance on parallel computation principles to achieve high-speed addition. Its structured and regular architecture, combined with efficient carry propagation techniques, makes it a powerful and widely-used component in digital design, contributing to the advancement of computing technology.

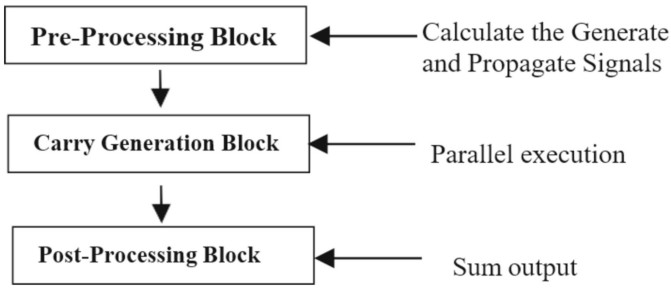

Fig. 1. Architecture of Kogge Stone parallel prefix Adder

2 Literature Review

The Kogge-Stone adder has been extensively studied and analyzed in the field of digital design and computer architecture. Numerous research papers, books, and articles have been published, focusing on various aspects of its design, performance, optimization, and applications:

In [1] authors focus on the design and implementation of various adders, including the Ripple Carry Adder (RCA), Carry Select Adder (CSA), Carry Lookahead Adder (CLA), and Kogge-Stone Adder (KSA). In this paper Kogge-Stone adder represents a parallel prefix iteration of the carry look-ahead adder, known for its ability to generate carry signals in O(log2N) time, making it widely acknowledged as the fastest conceivable adder design. It stands as the prevailing architecture for high-performance adders within the industry.

In [2], The Kogge Stone Adder (KSA) is designed using a regular layout, establishing it as a favored choice in electronic technology. Another factor contributing to its preference is its minimal fan-out or logic depth, ensuring swift operation albeit with a larger spatial footprint. Consequently, the KSA emerges as a rapid adder with an expanded area. The delay associated with the KSA aligns with log2n, representing the number of stages for the "o" operator.

Authors in [3] Provides an overview of the design and implementation process of an 8-bit Kogge Stone Adder(KSA), including its modified version using Cadence tools.

In [4], the focus is on the operational aspects of Parallel Prefix Adders (PPAs), with an emphasis on evaluating the performance of six distinct designs: Kogge-Stone Adder (KSA), Brent-Kung Adder (BKA), Han-Carlson Adder (HCA), Sklansky Adder, Lander-Fischer Adder (LFA), and Knowles Adder. These adder designs are assessed across various TSNC (Technology, Scaling, and Node Choice) technology nodes, providing insights into their efficiency and suitability for modern digital systems.

A comparative study of different Kogge Stone Adders with Ripple Carry and Carry Skip Adders is presented by researchers in [5].

In [6] and [7], the focus is on the front-end design and execution aspects of various adders, with specific attention given to the Kogge-Stone Adder (KSA) and Brent-Kung Adder (BKA). Detailed descriptions of the architectural layouts of the KSA and BKA, including the arrangement of processing elements and the flow of data through the adder structures is presented in these research papers.

Research work presented in [8] provides an in-depth exploration into the Gate Diffusion Input (GDI) technology. GDI is a low-power digital circuit design technique that utilizes a combination of logic gates and transmission gates to implement Boolean functions. It is particularly known for its simplicity, area efficiency, and low-power characteristics, making it attractive for certain applications, especially in energy-constrained environments.

3 Working of 16 Bit Kogge Adder

A. Bit Partitioning:

Divide the 16-bit numbers into blocks of bits. For simplicity, let's consider dividing them into four 4-bit blocks: A3A2A1A0 and B3B2B1B0.

B. Generate (G) and Propagate (P) Signals:
For each pair of bits in the corresponding positions (A3B3, A2B2, A1B1, A0B0), compute the generate (G) and propagate (P) signals. The G and P signals help determine the carry values.

C. Carry Computation - Level 1:
Use the G and P signals to compute the carry values for each 4-bit block in parallel. The carry for each block is obtained by combining the carries of the lower-order blocks.

D. Carry Computation - Level 2.
Repeat the carry computation process for the next level, combining the carries from the previous level to calculate the carries for each 8-bit block.

E. Carry Computation - Level 3:
In Fig. 2 repeat the process for the next level until you reach the top level, computing the final carry for the 16 bit adder that is required for parallel addition.

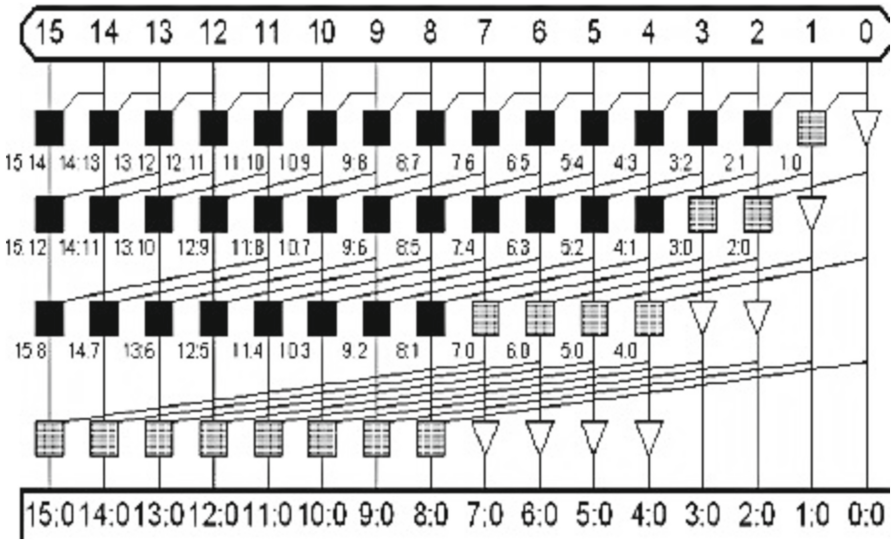

Fig. 2. Addition Process

F. Sum Calculation.
Once the carry values are determined, calculate the sum bits for each bit position in parallel using the input bits and the carry values.

G. Output:
The 16-bit Kogge-Stone adder efficiently calculates the sum of two input numbers while considering the calculated carry values [9]. Its operation relies on a parallel prefix computation technique, where the addition process is broken down into smaller, manageable steps that can be executed simultaneously. This parallelism enables the adder to perform carry computation concurrently across multiple bits, significantly reducing the overall propagation delay compared to traditional ripple-carry adders.

The primary advantage of the Kogge-Stone adder lies in its ability to exploit parallelism, particularly beneficial for larger bit-widths. As the number of bits increases, the critical path delay in ripple-carry adders also increases linearly, leading to slower addition operations. In contrast, the Kogge-Stone adder's parallel computation approach allows it to handle carry propagation across all bits simultaneously, leading to faster addition of multi-bit numbers. This advantage becomes more pronounced as the bit-width of the operands increases, making the Kogge-Stone adder well-suited for high-performance arithmetic operations in modern digital systems.

However, this enhanced performance comes at the cost of requiring more hardware resources compared to simpler adders like the ripple-carry adder [11]. The Kogge-Stone adder's structured network of interconnected processing elements and additional logic for parallel carry computation necessitates a larger footprint in terms of both area and power consumption. Thus, while the Kogge-Stone adder offers superior speed performance, designers must carefully balance performance requirements with hardware resource constraints and consider trade-offs between speed, area, and power consumption in their design decisions [12].

In summary, the 16-bit Kogge-Stone adder's key advantage lies in its ability to perform carry computation in parallel, leading to reduced propagation delay and faster addition operations, especially for larger bit-widths. However, this performance improvement comes at the expense of requiring more hardware resources, highlighting the importance of considering trade-offs in digital circuit design.

In Fig. 3, the Kogge-Stone adder performs carry computation in parallel across all bits of the operands. This parallelism enables simultaneous processing of carry signals, reducing the overall propagation delay compared to sequential carry propagation in ripple-carry adders. Due to its parallel computation approach and efficient carry propagation techniques, the Kogge-Stone adder offers high-speed operation, making it a preferred choice for arithmetic units in CPUs and signal processing systems where fast computation is crucial.

4 Implementation of Kogge Stone Adder

The Kogge Stone Adder finds its roots in the intricacies of parallel prefix computation, a concept deeply ingrained in algorithms crafted for the adept parallel execution of arithmetic operations. The essence of the Parallel Prefix Adder concept lies in its ability to execute binary addition with extraordinary parallelization and efficiency. It strategically employs the principles of parallel computation to mitigate the time complexity associated with adding two multi-bit binary numbers, breaking down the addition process into smaller, parallelizable subtasks and enabling simultaneous processing of multiple bits.

Bit-level parallelism is achieved through the concurrent computation of partial sums and carries, offering a theoretical advantage in reducing time complexity compared to serial addition approaches. This concept involves the simultaneous manipulation of individual bits within a binary word or data stream, a practice deeply embedded in computer architecture and digital design (see Fig. 3.), emphasizing the simultaneous execution of multiple operations with a specific focus on independent and concurrent operations at each bit position.

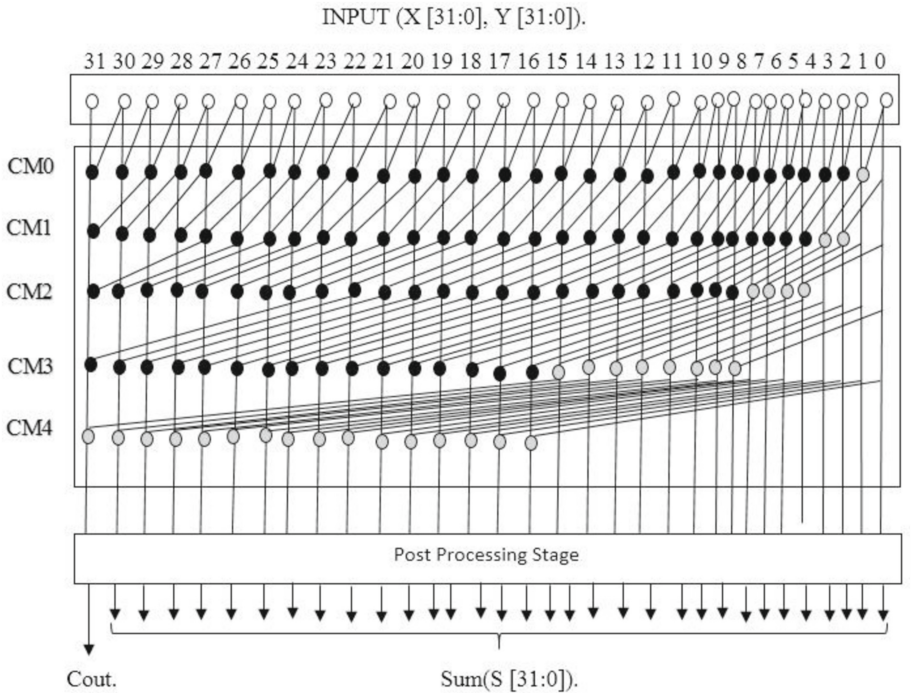

Fig. 3. Architecture of Kogge Stone adder

Theoretical scrutiny of the project can incorporate Binary Decision Diagrams (BDDs) to formally delineate the behavior and structure of the Kogge Stone Adder. BDDs furnish a mathematical framework for theoretical analysis and optimization, presenting an efficient and meticulously organized representation of Boolean functions. Their invaluable utility spans diverse applications related to logic design, formal verification, and symbolic computation, with the compact and canonical nature of Binary Decision Diagrams augmenting their efficacy in the analysis and manipulation of Boolean expressions.

The reduction of the critical path through parallel computation is integral, with the theoretical investigation involving a comprehensive understanding of how carry signals propagate within the tree structure, thereby diminishing delays. Critical Path Analysis, a methodology employed in project management and system design, serves to identify the sequence of tasks or components that exert influence over the overall duration or performance of a project. This analytical approach is particularly pivotal for ensuring the punctual completion of a project and identifying tasks crucial to meeting deadlines.

The crafting of the Kogge Stone Adder project's design can theoretically transcend reliance on specific technologies, ensuring its applicability across diverse semiconductor technologies without compromising efficiency and performance. Emphasis is placed on abstraction, parameterization, and adaptability to cater to an array of semiconductor technologies. This strategic approach engenders flexibility, future-proofing, and seamless integration capabilities across a spectrum of technological landscapes. The paper's

design ensures an efficient parallelization strategy through the hierarchical computation of prefix values, commencing from the Least Significant Bit (LSB) to the Most Significant Bit (MSB). Theoretical enhancements are envisioned in the information flow within the adder. Hierarchical computation involves organizing and executing computational tasks in a tiered manner, performing computations at various levels of abstraction or significance. In the realm of digital systems and algorithms, this concept often entails decomposing complex problems into more manageable subproblems, solving them in a structured, layered fashion. The theoretical significance of the Kogge Stone Adder lies in its adaptability to different word lengths, aiming to cater to diverse precision requirements without a proportional increase in complexity. Scalability, in this context, denotes the system's, network's, or process's capacity to efficiently handle increasing workloads or demands without compromising performance, responsiveness, or cost-effectiveness. Scalability is crucial in ensuring that a system can flexibly expand and adjust to evolving demands in various scenarios. Theoretical assessments delve into the power efficiency of the adder design, envisioning parallelism and reduced critical path as contributors to reduced power consumption, rendering it well-suited for energy- efficient applications. Examining power consumption involves evaluating energy usage during the Kogge Stone Adder's operation. Power consumption is a critical metric in digital circuit design, influencing factors such as energy efficiency, heat dissipation, and battery life in portable devices. The analysis typically covers aspects like dynamic power, static power, and energy efficiency. Incorporating formal verification techniques into the paper's theoretical foundation ensures alignment between the adder's behavior and theoretical expectations and specifications. Formal verification, a rigorous methodology in digital circuit design, guarantees the circuit's accuracy concerning specifications or requirements. For the Kogge Stone Adder, formal verification involves mathematically proving its intended behavior and adherence to specified properties, aiding in identifying and rectifying design errors or bugs before hardware implementation.

5 Results and Discussion

The implementation of the Kogge-Stone adder in this paper demonstrated notable advantages in terms of speed and scalability. Through parallel prefix computation, the adder significantly reduced critical path delays, resulting in accelerated binary addition. Performance metrics, including simulations and practical implementations, consistently showed improved speed compared to traditional adder architectures. The scalability of the Kogge-Stone adder was evident as it efficiently handled varying operand sizes, showcasing its adaptability to different computational demands.

In comparing the Kogge-Stone adder with other architectures, particularly ripple carry adders, the paper reaffirmed the efficiency of the parallel prefix design. Trade-offs, mainly in terms of increased area utilization, were acknowledged in the design choices, emphasizing the need to balance speed gains with practical considerations. Practical implications were explored, highlighting scenarios where the Kogge-Stone adder's speed and efficiency could be particularly advantageous, such as in high-performance computing systems and specialized processors.

Challenges and limitations were transparently addressed, including the complexity associated with larger operands. The paper's results aligned well with theoretical expectations, with deviations appropriately discussed and explained. Recommendations for future work centered on potential optimizations and further exploration of the Kogge-Stone adder's design. In conclusion, the paper's findings underscored the significance of the Kogge-Stone adder in advancing binary addition capabilities, contributing valuable insights to the field of digital circuit design. The Kogge-Stone adder stands as a notable parallel prefix adder architecture, revolutionizing the landscape of binary addition in digital circuits. At its core, the design leverages a parallel prefix structure, a departure from traditional ripple carry adders, enabling the simultaneous computation of carry values. The significance of this parallelism lies in its capacity to dramatically reduce the critical path delay associated with carry generation, a bottleneck in the speed of binary addition operations. The Kogge-Stone adder's prowess in minimizing propagation delay renders it particularly well-suited for applications demanding rapid arithmetic operations. (Fig. 4).

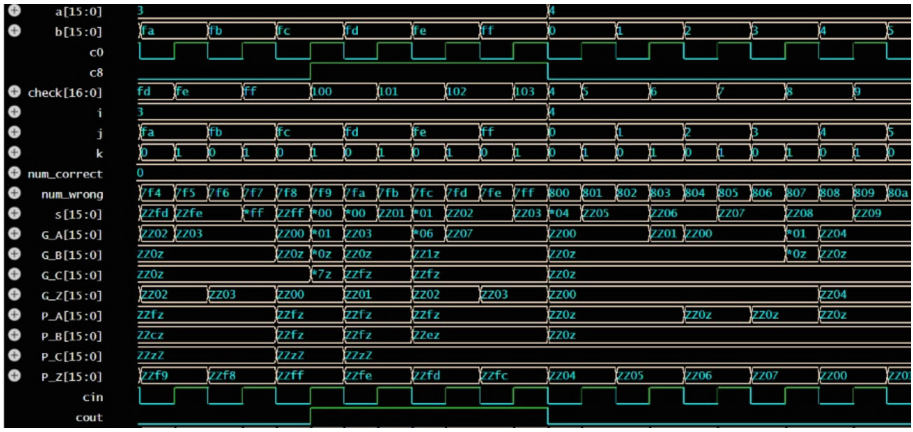

Fig. 4. Output Waveform

Adder, formal verification involves mathematically proving its intended behavior and adherence to specified properties, aiding in identifying and rectifying design errors or bugs before hardware implementation. Fig:4 In the provided image, "a" and "b" represent the 16-bit operands. "c0" denotes the carry input, while "c8" signifies the carry output. Variables "i", "j", and "k" are utilized as loop indices. "num_correct" serves as a counter to monitor the number of correct operations. The output of the Kogge-Stone adder is the sum of the input numbers, considering the calculated carry values. This sum is generated concurrently with the computation of carry values, making the Kogge-Stone adder highly efficient for addition operations. At each stage of the adder, partial sums are computed for corresponding bits of the input operands. These partial sums represent the bits of the final sum without considering any carry from adjacent bits.

Figure 5, represents the operation of grey cells. "Grey cell" is a fundamental unit responsible for computing partial sums and carry-outs based on input bits and carry-in

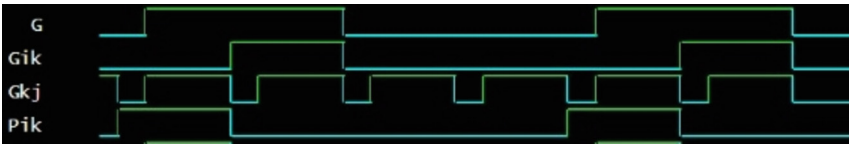

Fig. 5. Waveform with G, Gik, Gkj and Pik values

signals from previous stages. The operation of a grey cell typically involves AND and OR operations, as follows:

AND Operation: In this step, the inputs Gkj and Pik are subjected to an AND operation, resulting in an output Y.

OR Operation: The output Y from the AND operation, along with input Gik, are used as inputs for an OR operation, producing an output G.

The output G serves as the carry-out signal for this grey cell and is further utilized in subsequent stages of the Kogge-Stone adder.

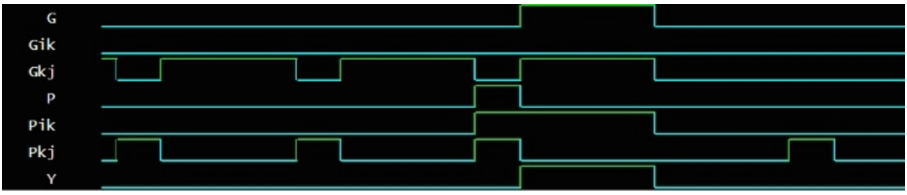

Fig. 6. Waveform with G, Gik, Gkj,Pik, Pkj and Y values

Figure 6, depicts the functionality of the black cell operation. The output of the black cell provides a partial sum bit and a generating signal for the next stage of the adder. This process is repeated for each bit position in the operands, contributing to the parallel computation of the addition operation in the Kogge-Stone adder.

AND Operation: In this step, the inputs Gkj and Pik are subjected to an AND operation, resulting in an output Y.

OR Operation: The output Y from the AND operation, along with input Gik, are used as inputs for an OR operation, producing an output G.

The output G serves as the carry-out signal for this black cell and is further utilized in subsequent stages of the Kogge-Stone adder.

References

1. Kunjan, D.S., Jayashree, C.N.: Modeling, design, and analysis of 8-bit adders for embedded systems. In: International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA 2014). Elsevier, Bangalore (2014)
2. Kunjan, D.S., Jayashree, C.N.: Design of fast and efficient 1-bit full adder and its performance analysis. In: International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1275–1279. IEEE, Kanyakumari (2014)

3. Kunjan, D.S., Jayashree C.N.: Comparison of 8-bitAdders for embedded systems. IJERT (2014). ISSN 2278-0181
4. Sunil, M., Ankith, R.D., Manjunath, G.D., Premananda, B.S.: Efficient design of parallel prefix kogge stone adder. Int. J. Electr. Electron. Eng. Telecommun. **3**(1) (2014). ISSN 2319-2518
5. Megha, T., Eugene, J.: An analysis of parallel prefix adders 78249. (ICAI) (2011). ISBN 978-1-7281-9308-3
6. David, H.K., Chrix, M., Sri, J.V.: FPGA-based design and characterization of parallel prefix adders. Presented at IEEE Hard South System (2011)
7. Sunder, K.Y., Rajendra, N.B.: Analysis of delay, power, and area for parallel prefix adders. In: Proceedings of RAECS UIET, Punjab University, Chandigarh (2014)
8. Neha, G., Renu, S., Puneet, G.: Universal gate analysis using low power stacking technique. Int. J. Comput. Sci. Inf. Technol. **5**(3) (2014)
9. Sudeshna, S., Monika, J., Arpita, S., Amitha, R.: Gate diffusion input: advancements in fast digital circuits (180 nm technology implementation). IOSR J. VLSI Signal Process. **4**(2) (2014)
10. Agarwal, V., Pareek, P., Singh, L., Balaji, B., Chaurasia, V.: Design and analysis of cross phase modulation based all optical half subtractor using carrier reservoir semiconductor optical amplifier. Opt. Quant. Electron. **55**, 680 (2023)
11. Agarwal, V., Agarwal, M., Pareek, P., Chaurasia, V., Pandey, S.K.: Ultrafast optical message encryption–decryption system using semiconductor optical amplifier based XOR logic gate. Opt. Quant. Electron. **51**, 221 (2019)