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

Analysis of Low Power High Speed Parallel Prefix Adder Using Cadence Encounter Platform

Ravi L S [a]*, Sinchana H S [b*], Thejaswini A R [c*], Tulasi C S [d*] Vinutha N A[e*]
[a] Assistant Professor, Department of Electronics and Communication, Rajeev Institute of Technology, Hassan-573201,India
[b,c,d,e] Students, Department of Electronics and Communication, Rajeev Institute of Technology, Hassan-573201,India

I. ABSTRACT

High performance in the age of lightning-fast computing and energy-efficient electrical devices depends on the creation of optimised arithmetic units. The efficiency of adders, which are crucial components of the majority of digital systems, has a direct effect on the system's overall speed and power usage. In this study, a low-power, high-speed parallel prefix adder (PPA) architecture leveraging Cadence nclaunch is designed and implemented. In terms of gates, power, and area, prefixes topologies such as Brent-Kung and Kogge-Stone are examined and contrasted. The suggested architecture uses effective gate-level implementations and logic-level optimisations to lower power consumption without sacrificing speed. To confirm functionality and performance, the design is synthesised and simulated using the cadence tool. When compared to traditional adder architectures, the findings show a notable gain in both power efficiency and area, demonstrating the usefulness of the suggested integration strategy in contemporary low-power, high-speed VLSI systems.

Keywords: PPA, EDP, SoC, FPGA.

II. INTRODUCTION

Arithmetic operations, especially addition, are essential and common in contemporary high-performance digital systems. The entire system performance is greatly impacted by an adder's space and power efficiency, particularly in applications like communication systems, microprocessors, digital signal processing (DSP), and image processing. Parallel prefix adders (PPAs), out of all the adder kinds, have become the favoured architecture because of their capacity to provide fast operation by use of effective carry computation. But as portable and battery-powered devices have advanced, power consumption along with speed and space has emerged as a crucial design limitation. Low-power yet fast adder architectures that may satisfy the requirements of contemporary VLSI design are therefore becoming more and more necessary.

The Kogge-Stone and Brent-Kung designs are examples of parallel prefix adders that use tree topologies to optimise the carry propagation process and achieve logarithmic latency. These architectures offer excellent performance, but because of their intricate interconnects and logic levels, they also come with trade-offs in terms of area and power. Thus, it is crucial to design a parallel prefix adder that strikes a compromise between area, gates, and power efficiency.

In order to develop and build a low-power, high-speed parallel prefix adder, this work investigates power-aware design techniques, logic-level improvements, and architectural optimisations. The suggested design is ideal for next-generation VLSI systems since it seeks to lower dynamic power consumption while preserving or enhancing delay performance. To verify the efficacy of the suggested design, a comparison with traditional prefix adders is conducted in terms of power, gates, and area using typical CMOS technology nodes.

Optimising the architecture of a low-power, high-speed PPA is necessary to reduce power consumption. This is essential in modern digital systems, where heat dissipation in high-performance computers and battery life in portable devices are directly impacted by power efficiency. Reducing the amount of logic levels in the carry computation, minimising switching activity, and balancing the trade-offs between power and time are the

main obstacles to accomplishing these objectives. To overcome these obstacles, methods like clock gating, transistor-level power saving techniques and logic gate optimisation are frequently used.

III. LITERATURE SURVEY

A new design for a parallel prefix adder that combines fast speed and low power consumption is presented by Y. Choi et al. [1]. In order to maximise the adder's performance, he suggests a hybrid logic method that combines the advantages of several logic types. CMOS and pass-transistor logic are combined in the architecture to minimise power consumption without sacrificing speed.

K. C. Shilpa , et. a1.,[2] focuses on a comprehensive study on the design and implementation of energy-efficient parallel prefix adders for Field-Programmable Gate Arrays (FPGAs). The authors aim to reduce the energy consumption of parallel prefix adders, which are critical components in many digital signal processing systems, while maintaining their high-performance capabilities. To achieve this goal, the authors explore various design techniques, including the use of different logic styles, such as Look-Up Table (LUT)-based and Digital Signal Processing (DSP)-based implementations, as well as the application of voltage scaling and clock gating techniques.

A. Raju et al. [3] present a novel approach to high-speed, low-power arithmetic circuits. In order to achieve low power consumption and high-speed operation, the authors suggest an optimised design that combines the advantages of CMOS logic and parallel prefix processing. A parallel prefix structure is used in the design to reduce latency and increase throughput, while the CMOS technology implementation offers high-speed operation and low power consumption. The efficiency of the design is demonstrated by simulation results, which show notable savings in power consumption and delay when compared to typical adders. The suggested architecture is optimised to save power usage while preserving high speed.

G. Thakur, et al., [4] with the goal of tackling the difficulties of attaining low-power and high-speed operation in arithmetic circuits. In order to minimise delay and lower power consumption, the authors suggest an optimised architecture that combines the advantages of parallel prefix computation with logic optimisation. A parallel prefix structure, logic reduction, and transistor scaling optimisation are some of the architectural and logical strategies used in the design to maximise the adder's latency. The architecture and function of the adder are thoroughly examined in the study, together with simulation results that show how well the design works to provide quick and low-power operation.

A new method for creating parallel prefix adders that strikes a balance between high performance and low power consumption is presented by N. Banerjee et al,.[5] By splitting the adder into two halves that run at separate voltage levels, the authors suggest a dual-voltage supply strategy. While the non-critical path of the adder runs at a lower voltage, the critical path, which controls the overall speed, runs at a greater voltage.

IV. PROBLEM FORMULATION

A.EXISTING METHOD

The high power consumption of current parallel prefix adders restricts their use in applications with limited energy resources. The inability of current designs to reach high operating frequencies limits their applicability for high-speed applications. Large silicon areas are occupied by current architectures, which raise prices and decreases integration. The performance of the adder is limited by long propagation delays. The accuracy and dependability of current designs are impacted by process variables.

B. PROPSOED METHOD

i. Minimize Power Consumption: Reduce the power consumption of the adder while maintaining its performance.
ii. Maximize gates: Achieve high-speed operation, typically measured in terms of clock frequency or propagation delay.
iii. Optimize Area: Minimize the area occupied by the adder on the chip, reducing the overall cost and increasing the yield.

V. METHODS

i) Power Optimization Techniques:

To minimize power consumption, the following techniques were employed:

- Hierarchical design: The design is divided into multiple stages, each computing a subset of the prefix values. This reduces the number of gates and wires required, resulting in lower power consumption.
- Minimized logic: The logic gates used in each stage are minimized to reduce power consumption.
- Reduced signal switching: The design is optimized to minimize signal switching, which reduces power consumption.

ii) Performance Optimization Techniques:

To maximize speed, the following techniques were employed:

- Pipelining: The design is pipelined, allowing each stage to compute its output independently. This increases the throughput of the design.
- Minimized delay: The delay through each stage is minimized to increase the overall speed of the design.
- Optimized gate sizing: The size of each gate is optimized to minimize delay and maximize speed.

A. Parallel prefix adder

The parallel prefix adder consists of three stages. In the pre-processing stage, the Propagate (Pi) and Generate (Gi) signals are generated, as illustrated in Figure 1. Upon receiving inputs A and B, the generate and propagate values are computed according to equation (1) and equation (2).

$$Gi = Ai \text{ AND } Bi \quad (1)$$
$$Pi = Ai \text{ XOR } Bi \quad (2)$$

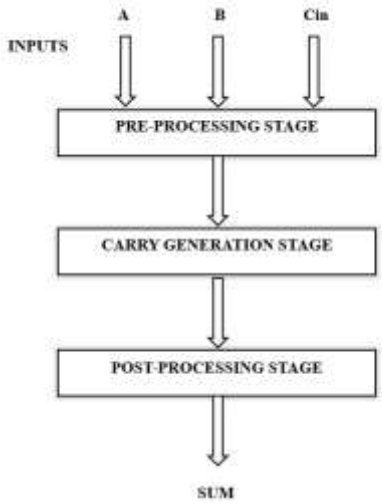

Fig. 1- Flow chart of parallel prefix adder

The prefix graph-based tree structure is employed during the PPA carry generation phase. Pairs of generate and propagate signals (Gx, Px) and (Gy, Py) are input into the second stage, known as the carry generation stage.

This results in the calculation of group generate and group propagate signals (Gx:y, Px:y), as illustrated in Figure 2. The final cell in each bit functions is to provide the carry bit. The last carry bit is enabled in a simultaneous summation of the next bit until the last bit. The carry generate and carry propagate are given in the equation (3) & equation (4).

$$Gm: n = Gm + (Pm.\ Gn) \quad (3)$$

$$Pm: n = Pm.\ Pn \quad (4)$$

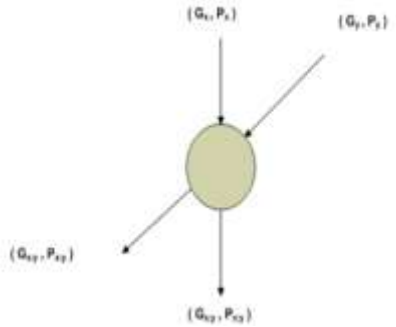

Fig. 2- Calculation of carry bit in a prefix graph

In post processing stage, the sum is calculated as given in the equation (5).

$$Si = Pi\ XOR\ Gi{-}1{:}cin \quad (5)$$

The critical route in parallel adders is defined by the carry transition from the least significant bit adder to the most significant bit adder. Therefore, it is essential to minimise the critical path for the carry to propagate to the most significant bits (MSB).

B. Brent - kung adder

The Brent-Kung adder represents a highly efficient and extensively utilised parallel prefix adder, serving as a fundamental component in digital circuit design for many years. In 1982, Richard Brent and H.T. Kung

developed an adder that significantly transformed the execution of arithmetic operations within digital systems. The Brent-Kung adder employs a recursive prefix computation to generate carry signals, which effectively minimises the critical path delay, thereby facilitating rapid and efficient arithmetic operations. This characteristic renders it suitable for applications necessitating high-speed arithmetic, including digital signal processing, scientific computing, and cryptography.

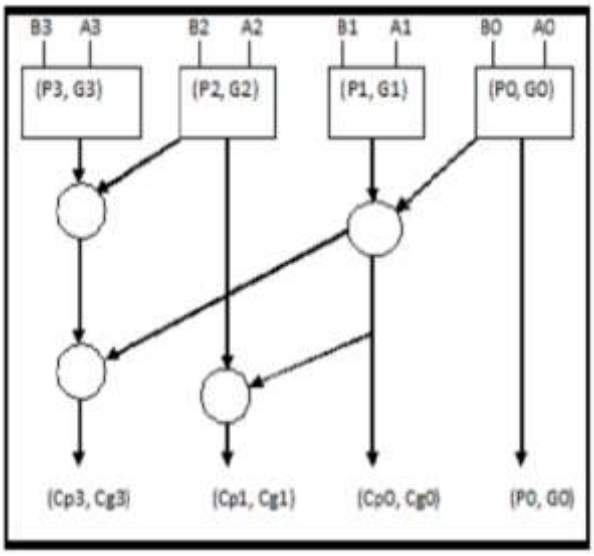

Fig. 3- Architecture of brent kung adder

C. Kogge – stone adder

When it comes to the design of digital circuits, the Kogge-Stone adder is a form of parallel prefix adder that is commonly utilised due to the fact that it is both fast and efficient in its architecture. This adder, which was developed by Peter Kogge and Harold Stone in 1973, generates the carry signals through the use of a recursive prefix computation. This allows for a large reduction in the critical path delay. The Kogge-Stone adder is very helpful in applications that demand high-speed arithmetic operations, such as digital signal processing, scientific computing, and encryption. These are all examples of applications that can benefit from its usefulness. The fact that it is designed to be modular and scalable also makes it appropriate for implementation in a broad variety of digital technologies, ranging from FPGA to ASIC implementations. Generally speaking, the Kogge-Stone adder is an arithmetic circuit that is extremely effective and frequently utilised, and it is an essential component in a great number of contemporary digital systems.

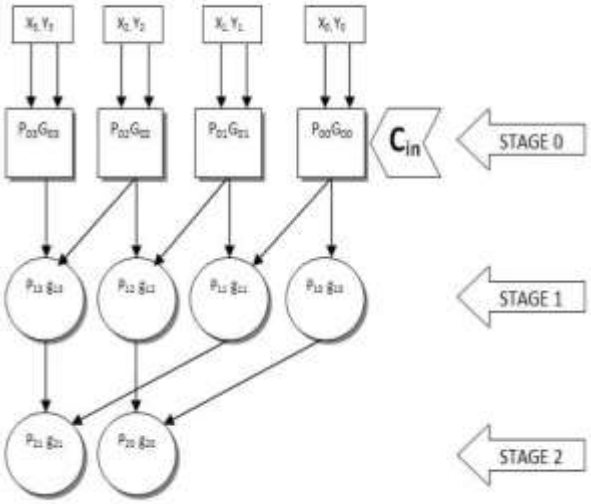

Fig. 4- Architecture of Kogge stone adder

VI. RESULTS & DISCUSSION

In order to illustrate the differences and similarities between the Kogge-Stone and Brent-Kung parallel prefix adders, we shall examine their performance, runtime, and area using an example. There is a difference between these adders in terms of fanout, the number of calculation stages, and the complexity of the logic. The inputs A=1011, B=1101 and Cin=0 have been taken into consideration, as can be seen in the image below. The results of the simulation showed that the sum was 1000, the carryout was 1, the carry propagation was 0110, and the carry generation was 1001.

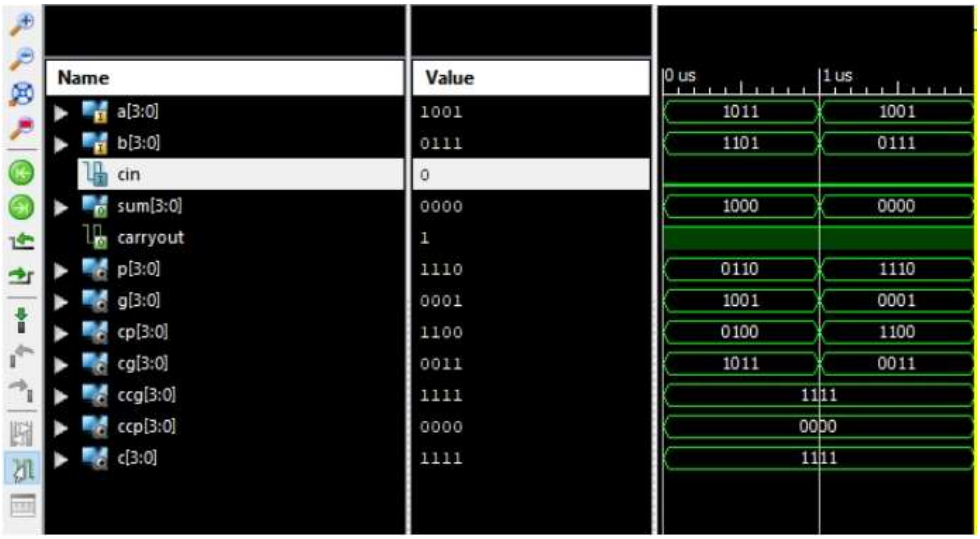

Fig. 5- Simulation output for Kogge Stone adder

The register-transfer level (RTL) schematic of the Kogge-Stone adder is a representation of the digital circuitry that is present in the adder. This schematic is made up of a number of different components, such as registers, wires, logic gates, generate (G) and propagate (P) blocks, and sum blocks. The G and P blocks, which are responsible for the generation and propagation of carry signals, receive the input bits A and B as their input. Following that, the sum blocks make use of these signals in order to compute the sum of the bits that were input. It is the sum and carry signals that are responsible for the generation of the output bits S and

C. A comprehensive comprehension of the Kogge-Stone adder's operation and usefulness is made possible by the RTL schematic, which offers a representation of the digital circuitry of the adder that is both clear and detailed.

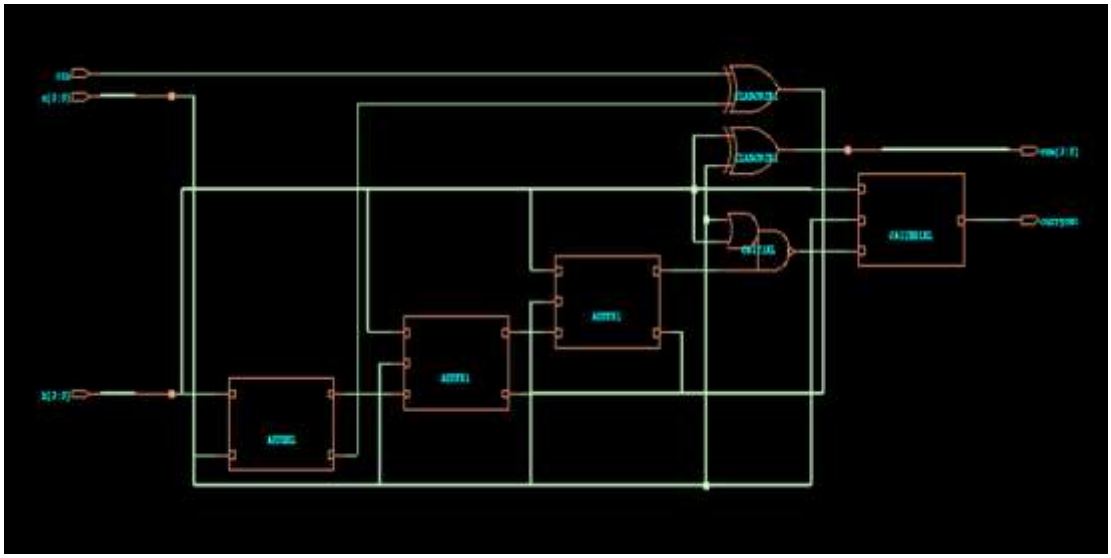

Fig 6- Snapshot of RTL Schematic of Kogge-stone adder

Performance analysis of the Brent – kung parallel prefix adder. As shown in the below figure, have taken the input has, A=1001, B=1101 and Cin=0. And after simulation we obtained sum=1001, carryout=1, carry propagation=0100, carry generation=1001.

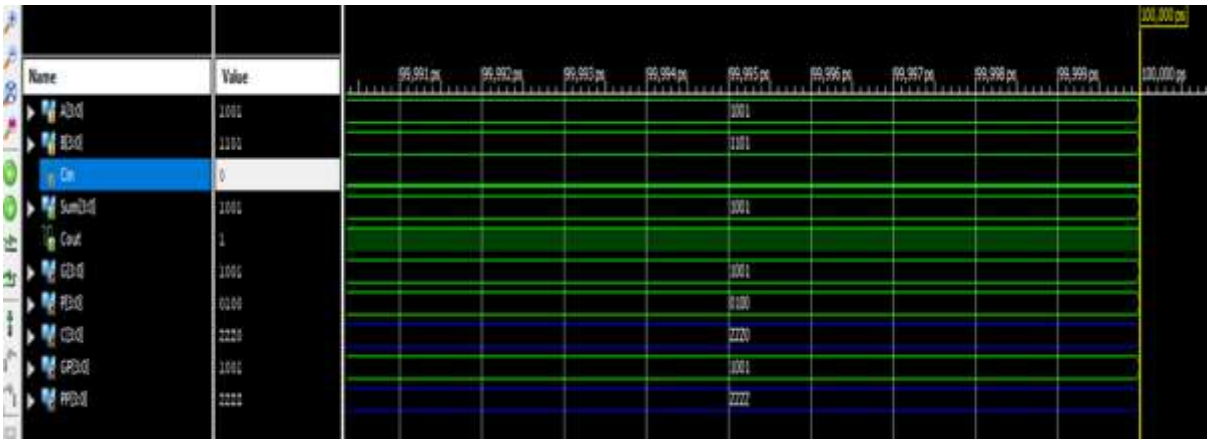

Fig 7- Simulation output for Brent-Kung adder

The Brent-Kung adder is a digital circuit that permits high-speed arithmetic operations. The RTL schematic, which stands for register-transfer level, is a representation of this circuit. The schematic is made up of input registers that store the binary integers A and B that are entered, as well as a prefix calculation logic that computes the prefix sums of the bits that are input by utilising a combination of AND, OR, and XOR gates. Carry computation logic then uses the prefix sums and carry-in signals to compute the carry-out signals for each bit position. This process is repeated for each bit position. Last but not least, the logic that performs the sum computation computes the final sum bits by making use of the prefix sums and the carry-out signals. The results of this computation are then saved in the output registers. Because of its parallel prefix computing scheme and optimised logic gates, the Brent-Kung adder is able to achieve both low latency and high speed, which makes it an appealing option for applications that require high-speed arithmetic. A comprehensive

representation of the digital circuit is provided by the RTL schematic of the Brent-Kung adder. This schematic highlights the most important components of the circuit as well as the flow of signals.

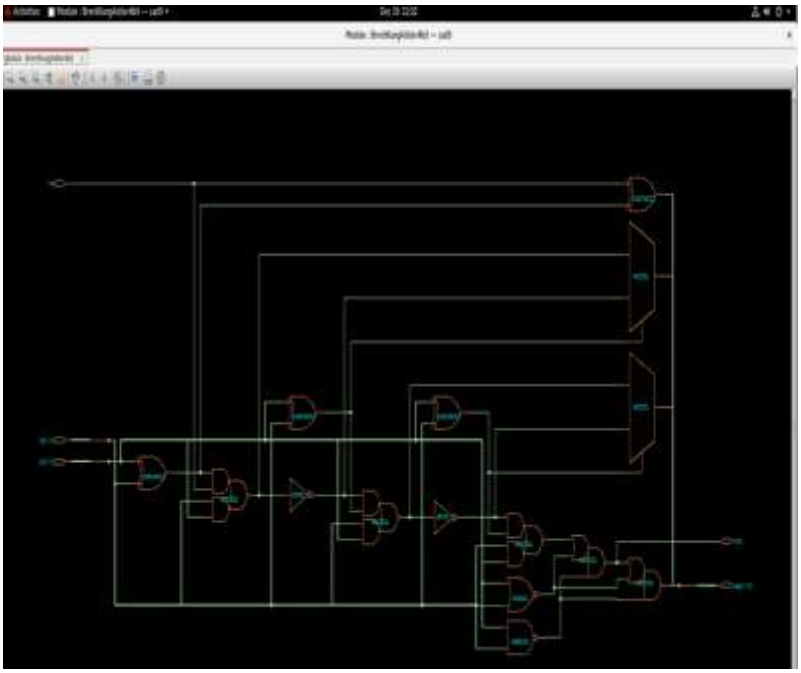

Fig 8- Snapshot of RTL Schematic of Brent-Kung adder

VII. PERFORMANCE ANALYSIS

As a result of the architectural and optimisation benefits that it possesses, the Kogge-Stone adder is significantly faster than the Brent-Kung adder. To be more specific, the Kogge-Stone adder has a smaller logic depth, which decreases the delay caused by signal propagation. Additionally, it uses a more parallel way to compute the prefix sums, which enables more concurrent computing and reduces the overall latency. In addition, the Kogge-Stone adder has a reduced fan-out, which reduces the capacitive load and enhances speed. Additionally, it utilises an enhanced carry chain structure, which minimises delay and also reduces the amount of power that is consumed. With all of these benefits taken into consideration, the Kogge-Stone adder is able to achieve faster and more efficient results than the Brent-Kung adder. Performance analysis of the Kogge-Stone parallel prefix adder in the below table.

Table1- *Comparison of Kogge-Stone adder and Brent-Kung adder*

Kogge-stone adder			
Techniques	**No. of Gates**	**area**	**Power in watt**
Existing (180nm)	12	92.357	4.84394e-06
Proposed(45nm)	7	77.961	2.27808e-06
Brent-Kung adder			
Techniques	**No. of Gates**	**area**	**Power in watt**
Existing (180nm)	17	98.547	4.73478e-06
Proposed(45nm)	5	83.259	2.77313e-06

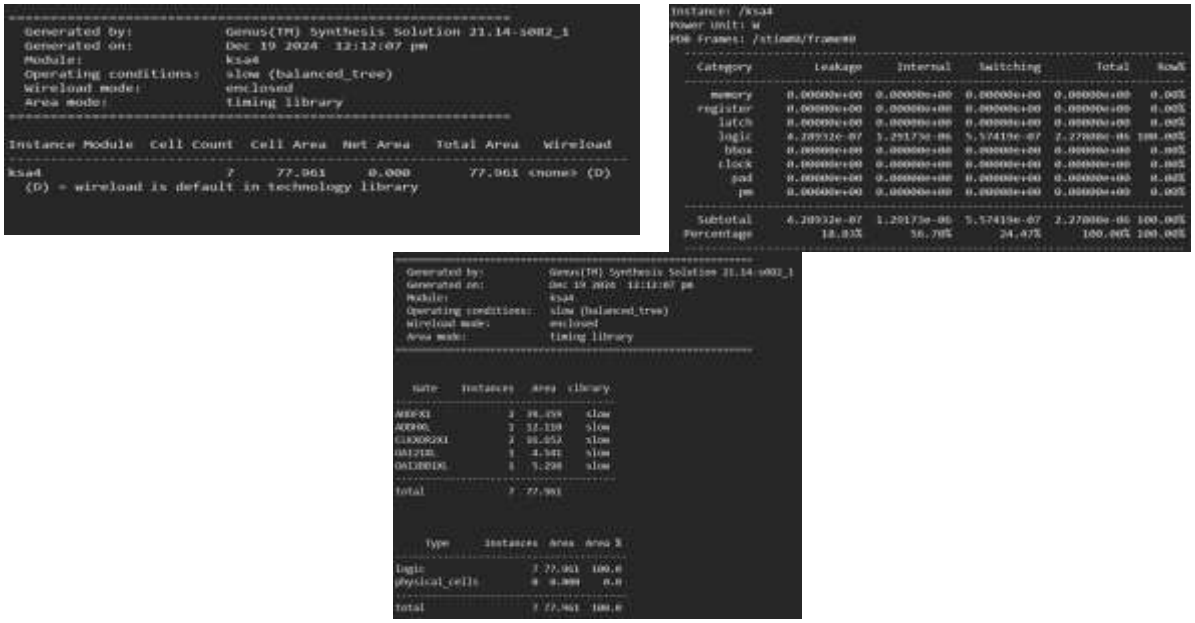

Fig. 9- Report of gates, power & area for Kogge-stone adder using cadence

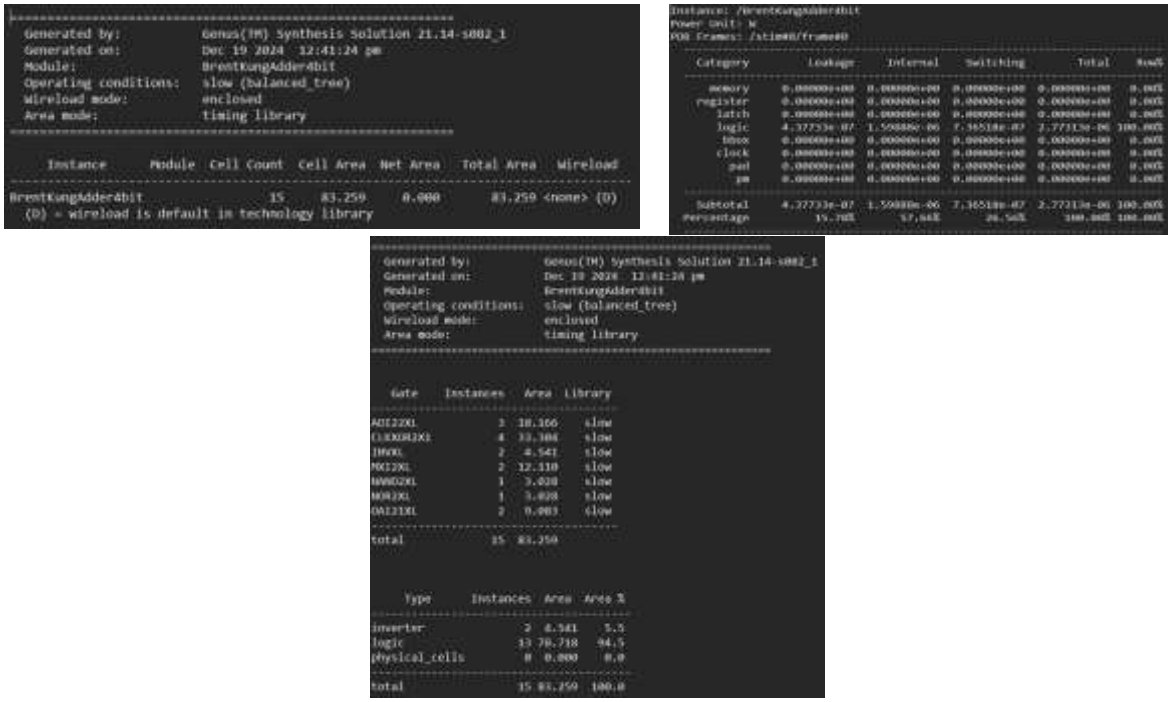

Fig. 10- Report of gates, power & area for Brent-Kung adder using cadence

VIII. CONCLUSION

The design and simulation of low power high speed parallel prefix adders have been successfully studied in this study, providing promising results that contribute significantly to the development of digital systems that are designed to be both energy efficient and high-performing. Based on the findings of the simulation, it has been demonstrated that the suggested design is capable of achieving substantial reductions in power consumption while still maintaining high-speed operation. This makes it appropriate for a broad variety of applications, ranging from low-power Internet of Things devices to high-performance

computing systems. In addition, the optimisation of power consumption and area has been thoroughly investigated, and the findings of the simulation have been validated by employing a variety of simulation tools and techniques. In general, this research has made a substantial addition to the field of low power VLSI design. Furthermore, the findings of this study can be utilised to direct the development of future digital systems that demand both high performance and low power consumption.

IX. REFERENCES

[1].Y. Choi and E. E. Swartzlander, "Parallel prefix adder design with matrix representation," 17th IEEE Symposium on Computer Arithmetic (ARITH'05), 2005, pp. 90-98, Doi: 10.1109/ARITH.2005.35.
[2].K. C. Shilpa, M. Shwetha, B. C. Geetha, D. M. Lohitha, and N. V. Pramod, "Performance analysis of parallel prefix adder for data path VLSI design," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1552–1555, 2018, Doi: 10.1109/ICICCT.2018.8473087.
[3].A. Raju and S. K. Sa, "Design and performance analysis of multipliers using Kogge Stone Adder," 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2017, pp. 94–99, Doi: 10.1109/ICATCCT.2017.8389113.
[4].G. Thakur, H. Sohal, and S. Jain, "A novel parallel prefix adder for optimized Radix-2 FFT processor," Multi-dim Stys Sign Process, vol. 32, pp. 1041–1063, 2021, Doi: 10.1007/s11045-021-00772-1.
[5].N. Banerjee, C. Augustine, and K. Roy, "Fault-Tolerance with Graceful Degradation in Quality: A Design Methodology and its Application to Digital Signal Processing Systems," IEEE International Symposium on, Defect and Fault Tolerance of VLSI Systems, pp. 323–331, 2008, Doi: 10.1109/DFT.2008.43.
[6].K. S. Pandey, D. K. B, N. Goel, and H. Shri Mali, "An ultra-fast parallel prefix adder," 2019 IEEE 26th Symposium on Computer Arithmetic (ARITH), pp. 125-134, 2019, Doi: 10.1109/ARITH.2019.00034.
[7].R. A. Patel, M. Benaissa, and S. Boussakta, "Fast parallel-prefix architectures for modulo 2n-1 addition with a single representation of zero," IEEE Transactions on Computers, vol. 56, pp. 1484–1492, 2007, Doi: 10.1109/TC.2007.70750.
[8].S. Roy, "Towards optimal performance-area trade-off in adders by synthesis of parallel prefix structures," IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems, vol. 33, pp. 1517–1530, 2014, Doi: 10.1145/2463209.2488793.
[9].S. Banerjee and W. Rao, "A general design framework for sparse parallel prefix adders," 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 231-236, 2017, Doi: 10.1109/ISVLSI.2017.48.
[10]. B. Harish, K. Sivani, and M. S. S. Rukmini, "Design and performance comparison among various types of adder topologies," 2019 3rd Inter- national Conference on Computing Methodologies and Communication (ICCMC), pp. 725–730, 2019, Doi: 10.1109/ICCMC.2019.8819779.
[11]. M. Bhavani, M. S. Kumar, and K. S. Rao, "Delay comparison for 16x16 Vedic multiplier using RCA and CLA," International Journal of Electrical and Computer Engineering (IJECE), vol. 6, no. 3, p. 1205, 2016, Doi: 10.11591/ijece. v6i3.pp1205-1212.
[12]. B. Harish and M. S. S. Rukmini, "Ultra high-speed full adder for biomedical applications," International Journal of Reconfigurable and Embedded Systems (IJRES), vol. 10, no. 1, p. 25, 2021, Doi: 10.11591/ijres. v10.i1. pp 25-31.
[13]. Y.-T. Pai and Y.-K. Chen, "The fastest carry lookahead adder," Proceedings. DELTA 2004. Second IEEE International Workshop on Electronic Design, Test and Applications, pp. 434-436, 2004, Doi: 10.1109/DELTA.2004.10071.
[14]. C.-J. Fang, C.-H. Huang, J.-S. Wang, and C.-W. Yeh, "Fast and compact dynamic ripple carry adder design," Proceedings. IEEE Asia-Pacific Conference on ASIC, 2002, pp. 25-28, Doi: 10.1109/APASIC.2002.1031523.
[15]. R. W. Doran, "Variants of an improved carry look-ahead adder," in IEEE Transactions on Computers, vol. 37, no. 9, pp. 1110-1113, 1988, Doi: 10.1109/12.2261.
[16]. R. A. Javali, R. J. Nayak, A. M. Mhetar, and M. C. Lakkannavar, "Design of high-speed carry save adder using carry lookahead adder," International Conference on Circuits, Communication, Control and Computing, pp. 33-36, 2014, Doi: 10.1109/CIMCA.2014.7057751.
[17]. A. Naik, D. Deka, and D. Pal, "ASIC implementation of high-speed adaptive recursive karatsuba multiplier with square-root-carry select-adder," 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS), pp. 1-4, 2020, Doi: 10.1109/LASCAS45839.2020.9068973.
[18]. G. C. Ram, D. S. Rani, R. Balasaikesava, and K. B. Sindhuri, "Design of delay efficient modified 16-bit Wallace multiplier," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016, pp. 1887-1891, Doi: 10.1109/RTEICT.2016.7808163.
[19]. Y. D. Ykuntam, K. Pavani, and K. Saladi, "Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, Doi: 10.1109/ICCCNT49239.2020.9225404.