*Grenze International Journal of Engineering and Technology, January Issue*

# Performance Analysis of Multi-Operand 4-Bit Adders

Abhay Chopde[1], Vaishnavi Kangralkar[2], Varsha Hulmani[3] and Tushar Nasery[4]
[1-4]Vishwakarma Institute of Technology,Pune, India
Email: abhay.chopde@vit.edu, s.vaishnavi21@vit.edu , ravindra.varsha21@vit.edu, tushar.nasery21@vit.edu

***Abstract*—In order to meet the growing requirement for computing systems that are efficient, a study of the problem of creating a multi-operand adder that can process many operands is required. The work involves a thorough examination of multiple adder architectures, closely examining their underlying properties such as area efficiency, power consumption, and latency. This paper presents a new strategy that uses a Carry Save Adder (CSA) to divide the addition process into discrete steps, which optimizes the process and reduces the drawbacks of using traditional techniques such as the Ripple Carry Adder (RCA). In order to shed light on the trade-offs between these approaches, the suggested analysis includes a detailed examination of the design's performance in Verilog at both the behavioral and Register-Transfer Level (RTL) abstraction levels.**

***Index Terms*— Adder, Carry-Save, Ripple-Carry, 4-bit, Multi-Operand, Area, Delay and Power Consumption.**

## I. INTRODUCTION

Adders are essential components in digital circuits, widely used in a variety of applications, including arithmetic, logic, and microprocessors. The efficiency of an adder, determined by factors such as area, delay, and power consumption, plays a crucial role in the overall performance of a digital system. In this paper, we analyze the performance of multi-operand 4-bit adders utilizing Carry-Save and Ripple-Carry architectures.

Multi-operand adders are designed to perform addition on more than two operands, extensively utilized in digital signal processing, arithmetic logic units (ALUs), and numerous other applications where multiple additions are performed simultaneously. The efficiency and speed of an adder are paramount in high-performance computing systems, especially in applications requiring complex arithmetic operations. These include signal processing, encryption, and graphics processing, among others. As technology advances and the demand for higher computational performance increases, it becomes crucial to explore and compare different adder architectures to optimize their performance.

The motivation behind this study lies in comparing the performance metrics of 3-operand, 4-operand, and 8-operand 4-bit adders using the Carry-Save and Ripple-Carry architectures. The objective is to determine which architecture is more efficient in terms of area, delay, and power consumption. By conducting a thorough analysis, we aim to provide valuable insights into the practical implementation of these adders. This analysis is expected to contribute significantly to the design and optimization of digital circuits, leading to improved performance and efficiency.

This paper is organized as follows: Section 2 describes the architecture and operation of Carry-Save and Ripple-Carry adders. Section 3 presents the methodology used for the comparative analysis. Section 4 discusses the results of the analysis, and Section 5 provides conclusions drawn from the study. Through this structured

*Grenze ID: 01.GIJET.11.1.31_1*

approach, we aim to offer a comprehensive understanding of the performance of multi-operand 4-bit adders, thus aiding in making informed decisions regarding their implementation.

## II. LITERATURE SURVEY

The paper "Modified CSA-CIA for Reducing Propagation Delay" explores the design and performance of adder circuits in Very Large-Scale Integration (VLSI) systems. It compares the Carry Save Adder (CSA) and Carry Increment Adder (CIA) and highlights their impact on system performance. The authors introduce a hybrid adder circuit that combines the strengths of both to minimize propagation delay while maintaining lower circuit complexity. The hybrid adder offers reduced gate count and improved propagation delay, making it a promising solution for VLSI system design[1].The research paper introduces a novel approach to approximate matrix multiplication, aiming to improve speed and reduce resource consumption in digital systems. It uses Approximate Computing, a trade-off between accuracy and efficiency, to enhance performance in scenarios like computer vision, machine learning, and data mining. The proposed architecture divides computation into two parts, using an OR gate approximation for summation and Dadda compression and Carry-Save Ahead adder for calculations[2].The paper discusses the design of a high-speed Carry Save Adder (CSA) for digital signal processing applications like Fast Fourier Transform and digital filters.It suggests incorporating a Carry Lookahead Adder (CLA) in the final stage instead of the conventional Ripple Carry Adder (RCA), which significantly improves the CSA's speed by 27.5%. The CLA-based CSA is found to be approximately 27.5% faster than the RCA-based CSA, meeting timing constraints and enhancing performance in DSP and FFT applications[3].The paper "Modified CSA-CIA for Reducing Propagation Delay" explores the design and performance of adder circuits in Very Large-Scale Integration (VLSI) systems. It compares the Carry Save Adder (CSA) and Carry Increment Adder (CIA) and highlights their impact on system performance. The authors introduce a hybrid adder circuit that combines the strengths of both to minimize propagation delay while maintaining lower circuit complexity. The hybrid adder offers reduced gate count and improved propagation delay, making it a promising solution for VLSI system design[4].The paper "Fault Tolerant Carry Save Adders – A NMR Configuration Approach" explores the design and implementation of fault-tolerant carry-save adders (CSAs) using hardware redundancy configurations. The authors explore various redundancy schemes to detect and potentially correct single event upset errors. The paper focuses on adding four 4-bit numbers using CSAs with different final stages, such as carry-lookahead adders (CLA) and carry-select adders (CSeA), along with a ripple-carry adder (RCA) stage. The findings suggest that fault-tolerant CSA designs using redundancy configurations can enhance the reliability of digital systems, making them valuable for researchers and engineers in Very-Large-Scale Integration (VLSI) design[5]. The authors present a 32-bit energy-efficient Wallace Tree Carry Save Adder (CSA) architecture for modern electronic systems. They propose a 1-bit hybrid full adder design that improves power consumption, transistor count, and delay compared to existing CMOS designs. This design offers shorter delay, lower power consumption, and requires fewer transistors, making it cost-effective for real-time applications. Simulation results show improved performance, including reduced power consumption, lower delay, and better power-delay products[6].The paper presents a 4-bit Vedic multiplier design using Quantum-dot Cellular Automata (QCA) technology, aiming to improve area, speed, and power consumption. The design uses the Urdhva Tiryagbhyam sutra for concurrent partial product generation and employs carry-save techniques. Simulation results show a 30% reduction in cell count, 60% reduction in area, and a 50% decrease in delay compared to a 4x4 Wallace and Dadda multiplier. The design offers significant improvements in efficiency and size, making it a promising solution for digital systems[7].The paper "Implementation and Comparison of VLSI Architectures of 16 Bit Carry Select Adder Using Brent Kung Adder" explores the design and comparison of different VLSI architectures for 16-bit carry select adders (CSLAs). The authors propose three architectures: Linear Brent Kung Carry Select Adder (LBKCSLA), Square Root Brent Kung Carry Select Adder (SQRTBKCSLA), and Modified Square Root Brent Kung Carry Select Adder (MSQRTBKCSLA). They use BKA and Binary to Excess-1 Converter (BEC) to optimise chip area and delay. The paper concludes that MSQRTBKCSLA offers a significant reduction in chip area[8].The paper discusses the design of a high-speed multiplier based on Vedic mathematics principles, aiming to reduce delay and area. The authors use the Urdhva-Tiryakbhyam algorithm for multiplication and introduce a new low-area high-speed adder. The design offers reduced delay and lower area compared to the Vedic multiplier with the regular carry select adder[9].The paper introduces a high-speed approximate multiplier design using compressors and carry-look-ahead (CLA) adders to accelerate computational speed. The design reduces the number of full adders and speeds up computations. Key innovations include approximate 4-2 compressors, which minimise error, and the replacement of the carry propagate adder with the faster CLA adder. Simulation and synthesis results show improved speed, fewer Look-

Up Tables, and acceptable error for multimedia and image processing applications[10].This paper presents a fast Finite Impulse Response (FIR) filter for ECG signal denoising, combining Vedic Mathematics principles and the UrdhvaTiryagbhyam sutra to create an 8-bit multiplier. The architecture includes four 4x4 Vedic multipliers, carry skip techniques, and Ripple Carry Adders for partial product addition. The paper also discusses the design of three adders for the 8-bit multiplier, evaluates FPGA and ASIC implementations for performance, and presents experimental results for both noisy and denoised ECG signals[11].This paper presents a design for a high-speed constant-breadth Adder Plantlet (AP) using a modified full adder. The structure reduces delays and compensates for truncation error using shortened input data. The modified full adder replaces a conventional XOR gate with a multiplexer, minimising path delay and power consumption. The modified full adder is integrated into a larger array multiplier, enhancing efficiency. This design is useful in digital signal processing and matrix vector multiplication within VLSI systems[12]..The paper reviews the performance of various Parallel Prefix Adders (PPAs) in VLSI design, focusing on their importance in digital systems. PPAs, also known as Carry Tree Adders, are fast and efficient for binary addition in digital applications. The Kogge Stone Adder (KSA) is known for its high-speed performance but increases circuit area. The Brent Kung Adder (BKA) is power-efficient with low node count and low area, offering a balance between speed and complexity. The Han Carlson Adder (HCA) is a combination of the two, offering a balance between speed and complexity [13]. The paper analyzes subthreshold leakage in a Carry Save Adder (CSA) in digital circuits. It aims to optimize a 4-bit CSA design using Gate Diffusion Input (GDI) methodology to minimize power consumption. Key results include a 4-bit CSA design with seven full adders and one half adder, simulation results, analysis of DC characteristics, optimization of process parameters, utilization of GDI cells, and a 37% optimized area reduction in average power consumption [14]. The paper presents a VLSI architecture based on Binary Vedic multiplication using a Carry Save Adder (CS3A), demonstrating its efficiency in terms of delay. Simulation results show the adder saves over 12% of energy and reduces area-delay-product by over 5%. Comparing it with existing architectures, it shows improved performance [15].

## III. METHODOLOGY

After reviewing the above papers, we decided to proceed with the implementation of a Carry Save Adder (CSA) for our VLSI system design. The CSA offers several advantages that make it an ideal choice for high-speed and efficient digital circuits. Firstly, it significantly reduces the propagation delay by enabling parallel addition of multiple operands, which is crucial for applications requiring fast computation such as digital signal processing and computer arithmetic. Secondly, the CSA architecture simplifies the carry propagation process, thereby improving overall system speed and performance. Additionally, the CSA is well-suited for fault-tolerant designs, as it can be easily integrated with various redundancy schemes to enhance reliability. These benefits, along with the potential for reduced power consumption and gate count, make the Carry Save Adder a promising solution for modern VLSI system design.

The proposed system designs a multi-operand adder that accepts four-bit as input. Ripple Carry Adder (RCA) is a typical adder used in this design. However, RCA has the disadvantage of carry propagation delay, where the carry-out of each bit position ripples through to become the carry-in of the next bit position. This leads to increased propagation delay and hardware requirements. To overcome these downsides, alternative adder types are explored. After considering various options, the design is implemented using the Carry Save Adder (CSA) technique. In CSA, the addition process is divided into two stages: generating partial sum and carry bits, and the final addition stage. This approach separates the carry generation and summation processes, reducing propagation delay. The multi-operand adder is designed using Verilog Hardware Description Language (HDL). To code in Verilog, the input and output sizes must be determined.

A Multi-operand adder (MOA) is designed and implemented using Verilog in Xilinx Vivado. It allows us to describe the behavior and structure of digital systems, which can then be synthesized into actual hardware. The design is targeted to the Spartan-7 FPGA board. The MOA is designed using the behavioral and RTL (Register Transfer Level) or Data Flow abstraction levels in Verilog. The functionality of the design is then tested using a testbench code. Further steps involve synthesis using Vivado's synthesis tools, and the final step is to analyze the power consumption and propagation delay of the design using Vivado's power analysis and timing analysis tools. Fig. 1 illustrates the typical design flow for digital systems, which involves describing the system behavior/structure in RTL code, verifying functionality through testbench simulation, synthesizing the RTL code into a gate-level netlist, and finally analyzing the synthesized design for power, area, and timing characteristics.

For a 3-operand 4-bit adder, the output size will be 5 bits including the carry-out, calculated using the equation size_of_final_sum = N + ⌈log2(k)⌉, where N is the bit width (4) and k is the number of operands (3).

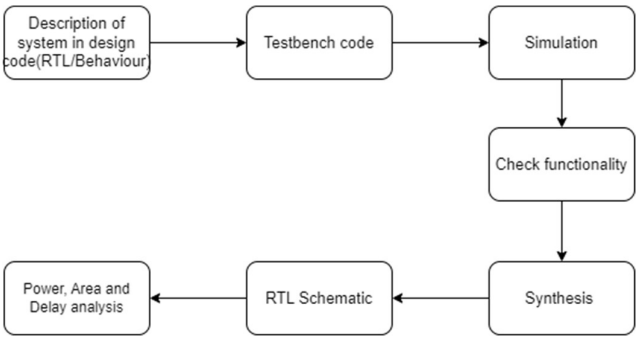


Fig.1. Workflow of design of study

The carry-save adder (CSA) technique separates addition into two stages. In the first stage, the first two operands are passed to a CSA to generate partial sums and carries. In the second stage, the partial sum is added to the third operand using a ripple carry adder (RCA) or another CSA, considering the partial carry as carry-in. The design is implemented in Verilog with modules for full adders, CSAs, and RCAs. Functionality is tested via a testbench, synthesized for the target FPGA, and power/delay are analyzed.

For a 4-operand 4-bit adder, the output size is 6 bits. Following the CSA technique, the first three operands are passed to a CSA in the first stage to generate partial sums and carries. In the second stage, the partial sum is added to the fourth operand using an RCA or CSA with the partial carry as carry-in. The Verilog implementation, testing, synthesis, and analysis steps are similar to the 3-operand case.

An 8-operand 4-bit adder has an output size of 7 bits. Due to the larger number of operands, multiple CSA stages are required. In the first stage, operands are grouped into pairs, and each pair is passed to a CSA to generate partial sums and carries. In the second stage, the partial sums and carries from the first stage are grouped into pairs and passed to another set of CSAs. This process repeats until only two operands (partial sum and carry) remain. In the final stage, these two operands are added using an RCA or CSA. The design is implemented in Verilog with full adder, CSA, and RCA modules, followed by testbench verification, synthesis, and power/delay analysis. The design complexity increases with more operands.

## IV. Results And Discussion

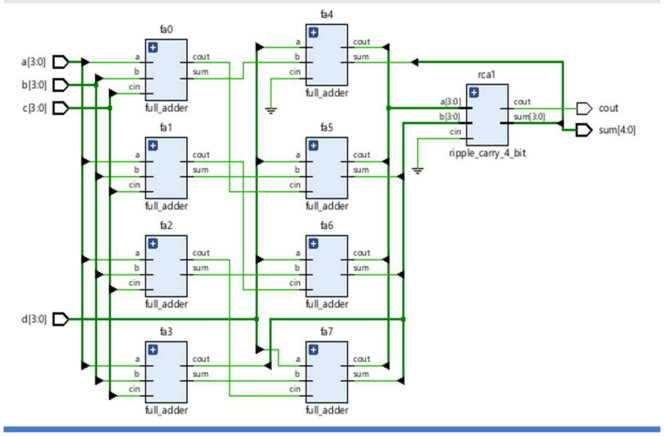


Fig. 2. RTL schematic of analysis for 4-bit 4-operand

Figures 2 and 3 show us the schematic for the 4-bit 4-operand adder circuit and the timing diagram for its output. In the first stage, the first three operands are added and the last operand is added in the last stage. The results of this can be seen in the simulation results which can be verified manually.

Figures 4 and 5 give us the utilization report and the power consumption summary for the 4-bit 4-operand adder.

Figure 6 gives us the register transfer layer schematic of the full adder used as blocks for the entire adder circuit. This is made using a half adder block and logic gates.

Figures 7 and 8 show us the simulation results and the utilization report for the 4-bit 3-operand adder circuit.

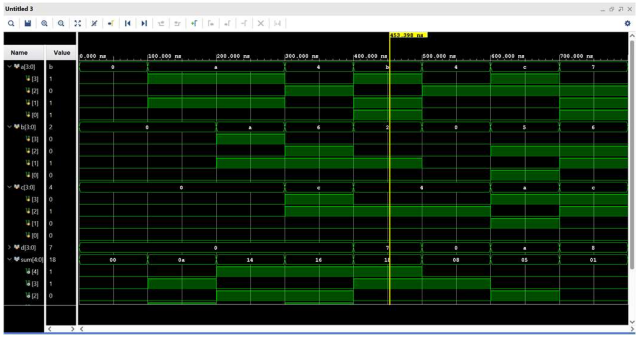

Fig. 3. Simulation Results of adder

| Name | Slice LUTs (64000) | Slice (16000) | LUT as Logic (64000) | Bonded IOB (400) |
|---|---|---|---|---|
| carry_save_adder | 16 | 6 | 16 | 22 |

Fig. 4. Utilization Report

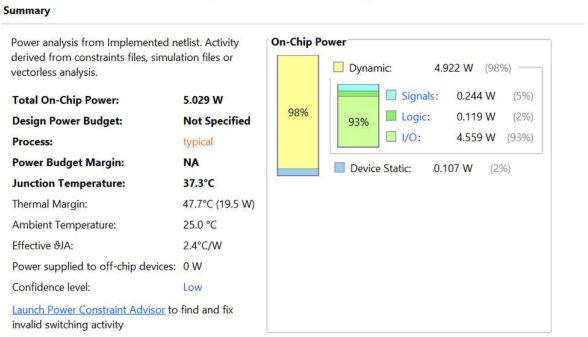


Fig. 5. Power Summary for 4-operand

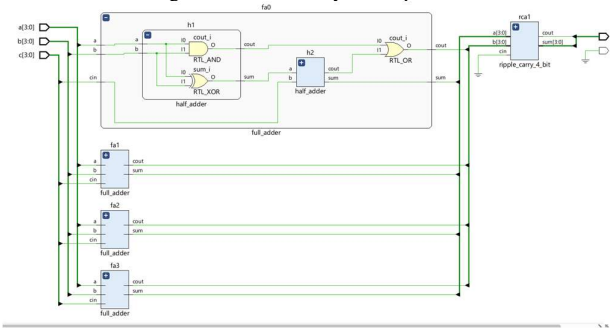


Fig. 6. RTL schematic of analysis for 4-bit 3-operand

Fig. 7 Simulation Results

| Name | Slice LUTs (64000) | Slice (16000) | LUT as Logic (64000) | Bonded IOB (400) |
|---|---|---|---|---|
| carry_save_adder_3op | 7 | 2 | 7 | 18 |

Fig. 8. Utilization Report for 4-bit 3-operand

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

| | |
|---|---|
| **Total On-Chip Power:** | **2.507 W** |
| **Design Power Budget:** | **Not Specified** |
| **Process:** | typical |
| **Power Budget Margin:** | **NA** |
| **Junction Temperature:** | **31.1°C** |
| Thermal Margin: | 53.9°C (22.0 W) |
| Ambient Temperature: | 25.0 °C |
| Effective θJA: | 2.4°C/W |
| Power supplied to off-chip devices: | 0 W |
| Confidence level: | Low |

Launch Power Constraint Advisor to find and fix invalid switching activity

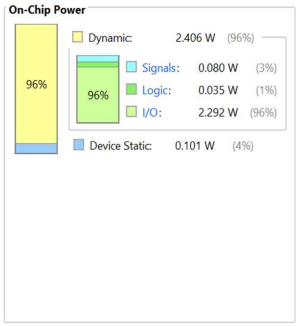


Fig. 9. Power Summary for 3-operand

Figure 9 shows the power usage report for the 4-bit 3-operand adder. As observed, this is almost half the power used by the 4-operand adder due to one less layer.

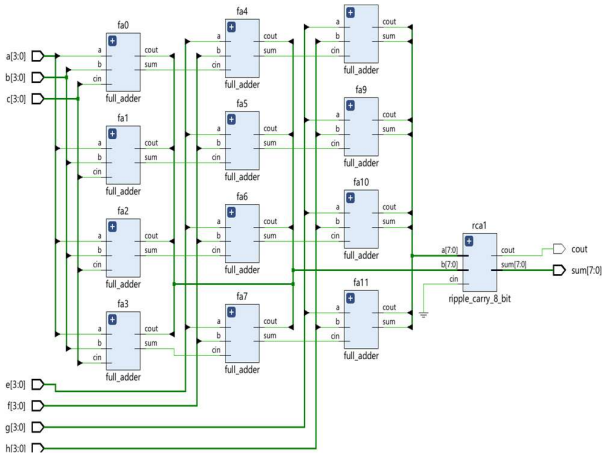


Fig. 10. RTL schematic of analysis for 4-bit 8-operand

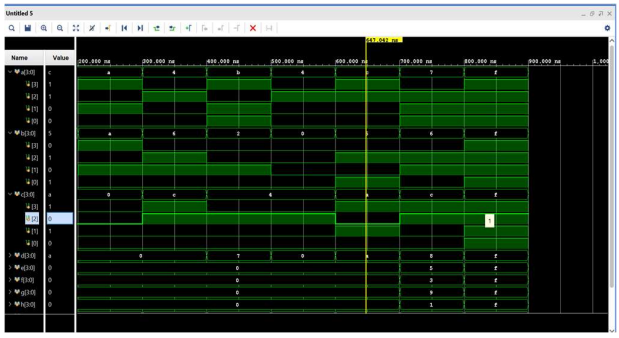

Fig. 11. Simulation Results

Figure 10 shows the schematic of the 4-bit 8-operand adder circuit. To accommodate the extra operands, another layer has been added to the circuit. Figure 11 shows the simulation results for this adder.
Figures 12 and 13 show us the power usage report and the utilization report for the 4-bit 8-operand adder. Since there is an additional layer in this circuit, the power used is higher than the 3 and 4-operand adders.

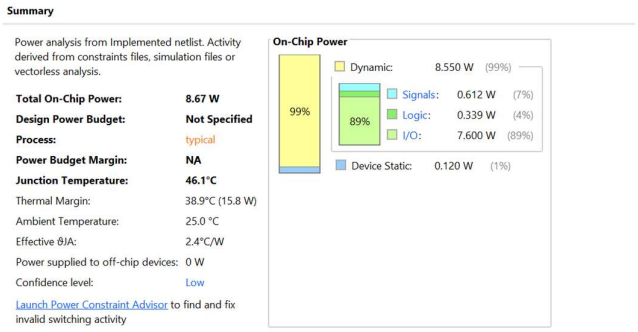


Fig. 12. Power Summary for 8-operand

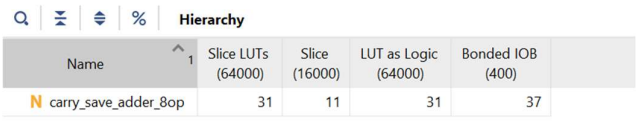


Fig. 13. Utilization Report for 4-bit 8-operand

## V. CONCLUSIONS

The study presents an efficient design and implementation of multi-operand adders using the Carry Save Adder (CSA) technique to overcome the limitations of traditional Ripple Carry Adders (RCAs). The proposed approach separates the addition process into stages, reducing propagation delay and optimizing hardware resources. The design, realized using Verilog HDL and Xilinx Vivado for the Spartan-7 FPGA, encompasses behavioral and RTL abstraction levels, testbench verification, synthesis, and power/delay analysis. The study demonstrates the effectiveness of the CSA technique and provides a scalable solution for multi-operand addition operations, contributing to the advancement of high-performance and energy-efficient digital systems.

## REFERENCES

[1] Sarkar, Shubham, Sujan Sarkar, and Jishan Mehedi. "Modified CSA-CIA for Reducing propagation delay." In 2018 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-5. IEEE, 2018.

[2] Kumar, Arun, Asif Ansari, Ayush Srivastava, and Kriti Suneja. "Fast Approximate Matrix Multiplier based on Dadda Reduction and Carry Save Ahead Adder." In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 1058-1061. IEEE, 2021

[3] Javali, Ravikumar A., Ramanath J. Nayak, Ashish M. Mhetar, and Manjunath C. Lakkannavar. "Design of high speed carry save adder using carry lookahead adder." In International Conference on Circuits, Communication, Control and Computing, pp. 33-36. IEEE, 2014

[4] Chandrashekara, M. N., and S. Rohith. "Design of 8 bit Vedic multiplier using Urdhva Tiryagbhyam sutra with modified carry save adder." In 2019 4th international conference on recent trends on electronics, information, communication & technology (RTEICT), pp. 116-120. IEEE, 2019

[5] Radhakrishnan, S., T. Nirmalraj, S. Ashwin, V. Elamaran, and Rakesh Kumar Karn. "Fault Tolerant Carry Save Adders-A NMR Configuration Approach." In 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), pp. 210-215. IEEE, 2018.

[6] Pandey, Jai Gopal, and Gaurav Dhiman. "An architecture for 32-bit energy-efficient wallace tree carry save adder." In 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), pp. 1-4. IEEE, 2017

[7] Chudasama, Ashvin, and Trailokya Nath Sasamal. "Implementation of 4× 4 vedic multiplier using carry save adder in quantum-dot cellular automata." In 2016 International conference on communication and signal processing (ICCSP), pp. 1260-1264. IEEE, 2016.

[8] Kumar, N. Udaya, K. Bala Sindhuri, K. Durga Teja, and D. Sai Satish. "Implementation and comparison of VLSI architectures of 16 bit carry select adder using Brent Kung adder." In 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1-7. IEEE, 2017.

[9] Sunil Kumar, K., and M. Swathi. "128-bit multiplier with low-area high-speed adder based on vedic mathematics." In Proceedings of 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications: ICMEET 2016, pp. 163-172. Springer Singapore, 2018.

[10] Jeevan, B., and K. Sivani. "A new high-speed multiplier based on a carry-look-ahead adder and compressor." In VLSI Design: Circuits, Systems and Applications: Select Proceedings of ICNETS2, Volume V, pp. 69-78. Springer Singapore, 2018.
[11] Padmavathy, T. V., S. Saravanan, and M. N. Vimalkumar. "Partial product addition in Vedic design-ripple carry adder design fir filter architecture for electrocardiogram (ECG) signal de-noising application." Microprocessors and Microsystems 76 (2020): 103113.
[12] Pushpalatha, B., T. Pavani, A. Ushasree, Ch Usha Kumari, and G. L. Sumalatha. "High speed constant-breadth adder-plantlet design using modified full adder." Materials Today: Proceedings 45 (2021): 1575-1582.
[13] Rakesh, S., and KS Vijula Grace. "A comprehensive review on the VLSI design performance of different Parallel Prefix Adders." Materials Today: Proceedings 11 (2019): 1001-1009.
[14] Ojha, Sunil Kumar, O. P. Singh, and G. R. Mishra. "Analysis of the Effect of Subthreshold Leakage on Carry Save Adder."
[15] Priyanga, A., and C. N. Marimuthu. "Binary Vedic Multiplication Using Carry Save Adder CS3A based on Modified Approximate Three Operand Adder.