Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 10
October 2025

Design And Performance Comparison Among Various Types Of Adder Topologies

Dr. Chetana[1], Nthya S[2], Latha S[3], Dr. Anitha P[4]

[1,2,3]Asst Prof, [4]Assoc Prof., Department of ECE, SJB Institute of Technology, Bangalore, India

Abstract

Introduction: In modern digital systems, arithmetic operations are critical for high-performance computations, with addition being a fundamental operation in microprocessors, digital signal processing (DSP), and application-specific integrated circuits (ASICs). The performance, power consumption, and area utilization of these systems are significantly influenced by the choice of adder architecture.

Objectives: This work provides valuable insights for circuit designers in selecting the most appropriate adder topology for diverse computational applications.

Methods: This project explores the design and performance comparison of various adder topologies, including Ripple Carry Adder (RCA), Carry Look-Ahead Adder (CLA), Carry Select Adder (CSLA), Carry Skip Adder (CSKA) and Kogge-Stone Adder (KSA), using the Xilinx Vivado platform. The study evaluates key metrics such as propagation delay, power consumption, and area utilization to determine the suitability of each topology for specific application requirements.

Results: The result is measured by evaluating three parameters Delay, Power Consumption and Area Usage. The present delay values for each adder topologies is determined and the results are illustrated. Similarly the Power Consumption for each design is reported, highlighting most energy-efficient adders. Along with the above two parameters the Area Usage is measured by including the number of logic elements or gates used which further helps in identifying the designs that balance area efficiency with performance.

Conclusions: The analysis highlights the Carry Look-Ahead Adder (CLA) as the most efficient topology overall due to its balance of speed, power efficiency, and minimal resource usage, making it an optimal choice for modern digital systems prioritizing both performance and resource optimization.

Keywords: Ripple Carry Adder (RCA), Carry Look-Ahead Adder (CLA), Carry Select Adder (CSLA), Carry Skip Adder (CSKA) and Kogge-Stone Adder (KSA)

INTRODUCTION

From basic arithmetic operations to the complexities of modern computational units, digital adders are the cornerstone of digital systems. Adders form the backbone of various critical subsystems, such as Arithmetic Logic Units (ALUs), processors, memory access units, and digital signal processing (DSP) systems. The increasing demand for high-performance computational systems has intensified the need for efficient adder designs that achieve an optimal balance between speed, power consumption, and hardware utilization. Digital circuits, particularly in applications involving Field Programmable Gate Arrays (FPGAs), often face stringent constraints related to processing speed, area occupancy, and energy efficiency. As digital systems scale in size and complexity, the trade-offs between these parameters become more critical. Choosing the appropriate adder architecture is vital to addressing these trade-offs, especially in performance-critical systems such as microprocessors, signal processing hardware, and real-time embedded systems.

This project explores and compares five well-known adder architectures: Ripple Carry Adder (RCA), Carry Skip Adder (CSkA), Carry Select Adder (CSlA), Carry Look-Ahead Adder (CLAA), and Kogge-Stone Adder (KSA). Each of these adders represents a unique design philosophy, with varying performance characteristics in terms of delay, power consumption, and area utilization. For instance, the Ripple Carry Adder, while simple

and compact, suffers from significant delay due to its sequential carry propagation. In contrast, the Kogge-Stone Adder, often referred to as a parallel prefix adder, offers minimal delay at the cost of increased area and power. The performance analysis of these architectures is conducted across multiple bit lengths to evaluate scalability and efficiency under varying computational demands.

Using industry-standard tools like Xilinx Vivado and hardware description language Verilog, simulations are performed to measure critical performance parameters such as delay, power consumption, and area utilization. These metrics serve as benchmarks for determining the suitability of each adder topology in specific applications.

In modern computing systems, adders play an indispensable role in arithmetic calculations, including addition, subtraction, multiplication, and division. Beyond traditional applications, adders are critical components in advanced domains such as cryptography, artificial intelligence, neural networks, real- time autonomous systems, and graphics processing units (GPUs). Optimizing adder performance, therefore, directly impacts the overall efficiency of these systems. This study seeks to contribute to the ongoing efforts to enhance digital circuit design by identifying the strengths and weaknesses of various adder topologies. The analysis aims to provide insight into the trade-offs between speed, power, and area, thereby guiding designers to select the most suitable adder architecture for their specific requirements. By bridging the gap between theory and practical implementation, this work will enable the development of high-performance, low-power digital systems tailored to real-world applications.

LITERATURE SURVEY

A thorough review of existing literature is essential to understand the advancements, challenges, and research trends in adder design and performance analysis. The literature survey examines various adder architectures and their applications, providing a foundation for this project's objectives

From paper [1] the study evaluates various adder architectures with a focus on parameters such as delay, resource usage, and power consumption. Among the tested designs, the Kogge Stone Adder demonstrates superior performance with the lowest delay, minimal resource utilization, and the least power consumption, establishing it as the most efficient architecture in this comparative analysis. In [2], it highlights the comparative analysis of different adder designs based on area, power, and delay metrics. The Brent-Kung Adder (BKA) emerges as the fastest adder in terms of delay, emphasizing its suitability for high-speed applications in FPGA-based systems. [3] presents a comprehensive evaluation of adder architectures, focusing on delay, area, and power consumption. The findings underscore that the Brent-Kung Adder (BKA) has notable improvements in delay efficiency, making it a favorable choice for performance-critical systems. Paper [4] explores the modifications to traditional adder designs enhancing their speed and reducing power consumption. The authors have modified the adders for betterment of speedy devices with low power consumptions. The proposed improvements enable these adders to be more suitable for modern VLSI applications requiring fast and energy-efficient computations. In paper [5] the authors provide a comparative study of seven different 4-bit adder topologies. The research delves into their performance characteristics and explores their trade-offs, setting a foundation for selecting adders based on specific design requirements. In [6] the authors evaluate various adder architectures using Verilog HDL in Xilinx ISE 13.2 for the Virtex-6 FPGA family. The Carry Increment Adder is identified as the best performer in terms of area and delay, emphasizing its potential for efficient hardware implementation in digital circuits. In [11] the author compares with ddifferent adders considering the area, power and delay as the main parameters.

The comparative studies across various papers emphasize the evaluation of adder architectures based on parameters like delay, area, and power consumption. Key findings include: • Kogge Stone Adder consistently outperforms other designs in terms of delay, resource usage, and power consumption, making it ideal for high-performance applications.

• Brent-Kung Adder (BKA) shows improved delay efficiency, highlighting its suitability for systems prioritizing speed.

• Carry Increment Adder demonstrates a balance between area and delay, standing out in FPGA- based implementations.

• Modified adder designs aim to enhance speed and reduce power consumption, aligning with the needs of modern VLSI applications.

• Studies on 4-bit adders and broader analyses provide insights into design trade-offs, aiding in the selection of architectures based on specific design goals. These findings guide the design of efficient adders tailored to application-specific requirements in digital systems.

This literature survey provides an overview of comparative studies and performance analyses of

different adder topologies, helping to identify the best-suited designs for specific requirements based on delay, area, power consumption, and overall efficiency.

PROPOSED METHODOLOGY

The proposed methodology focuses on a structured and systematic approach to design, implement, and evaluate various adder topologies using the Xilinx Vivado platform. The aim is to provide a detailed comparative analysis of the performance metrics of these topologies to aid in selecting the most efficient design for specific application scenarios.

Technology Selection: The Simulation Tool used is Xilinx Vivado 2024.2 which is a state-of-the-art FPGA design suite. It allows for comprehensive design, simulation, and synthesis of digital circuits, providing support for timing analysis, power estimation, and area evaluation, which are critical for your project's goals.

Relevance:

• The tool will be used to evaluate and compare the performance of adder architectures.
• It supports hardware verification through FPGA implementation, ensuring that theoretical results align with practical outcomes.

To ensure efficient design, implementation, and evaluation of adder topologies, Xilinx Vivado 2024.2 is proposed for use over the older Xilinx ISE. Below is a detailed explanation of why Vivado 2024.2 is the better choice.

➢ Vivado offers an integrated design environment for High-Level Synthesis (HLS), simulation, synthesis, and implementation. This eliminates the need for separate tools for various stages, unlike ISE, which requires external simulation tools.

➢ Vivado provides a block diagram interface for designing complex systems visually, simplifying integration for large-scale designs, a feature not available in ISE.

➢ Vivado supports cutting-edge FPGA families like Ultra Scale+, Versal ACAP, and other high-performance devices, whereas ISE is limited to older FPGA series such as Spartan and Virtex.

➢ Vivado's architecture is optimized for multi-threaded processing, significantly speeding up synthesis and implementation compared to the single-threaded processing of ISE.

➢ By leveraging Xilinx Vivado 2024.2, designers can take advantage of its advanced toolchain, superior performance, modern FPGA support, and comprehensive debugging capabilities. This makes it the ideal choice for implementing, testing, and optimizing adder topologies compared to the outdated and less efficient Xilinx ISE.

METHODOLOGY

1. Selection of Adder Topologies: -Identify and select a diverse set of adder topologies for comprehensive comparison.
2. Adder Design and Modeling: -Develop HDL (Hardware Description Language) models for each adder topology within the Xilinx Vivado environment. Design the selected adder topologies using VHDL or Verilog hardware description languages.
3. Synthesis and Implementation: -Translate HDL designs into gate-level representations and implement them on the target FPGA platform. Execute the synthesis process for each adder design, Run the implementation phase, which includes mapping, placing, and routing the synthesized netlist onto the FPGA.
4. Performance Metrics Evaluation: -Assess each adder topology based on key performance indicators: propagation delay, power consumption, and area utilization

- Timing Analysis: Extract and record the critical path delays from the timing reports for each adder.
- Power Estimation: Use power analysis tools to estimate dynamic and static power consumption.
- Area Measurement: Document the number and types of FPGA resources utilized by each adder topology.

5. Comparative Analysis: -Systematically compare the performance of different adder topologies based on the collected data.

- Optimal Selection Criteria: Establish criteria for selecting the most suitable adder topology based on specific application requirements.

6. Documentation and Reporting: -Compile the findings into a comprehensive report detailing the design processes, performance evaluations, and comparative insight. This methodology provides a structured framework to design, implement, simulate, and evaluate various adder topologies using Xilinx Vivado.

CONCEPTUAL DESIGN

Framework: The design consists of the following stages:

a) Input Specifications
Define adder input bit-widths: 16-bit.

Input signals: Binary operands AAA and BBB, and optional carry-in Cin C{in}Cin.

b) Verilog Code Development
Write Verilog code for each adder topology: Ripple Carry Adder (RCA), Carry Skip Adder (CSkA), Carry Look-Ahead Adder (CLAA), Carry Select Adder (CSlA), Kogge Stone Adder (KSA). Example (Carry Look Ahead Adder):

c) Simulation Environment
Software: Xilinx Vivado for synthesis and performance analysis.
Language: Verilog for describing adder behavior.

d) Performance Metrics Measurement:
Power Consumption.
Area Usage.
Delay (Latency).

e) Comparison and Analysis:
Evaluate metrics across different architectures. Identify trade-offs between speed, power, and area.

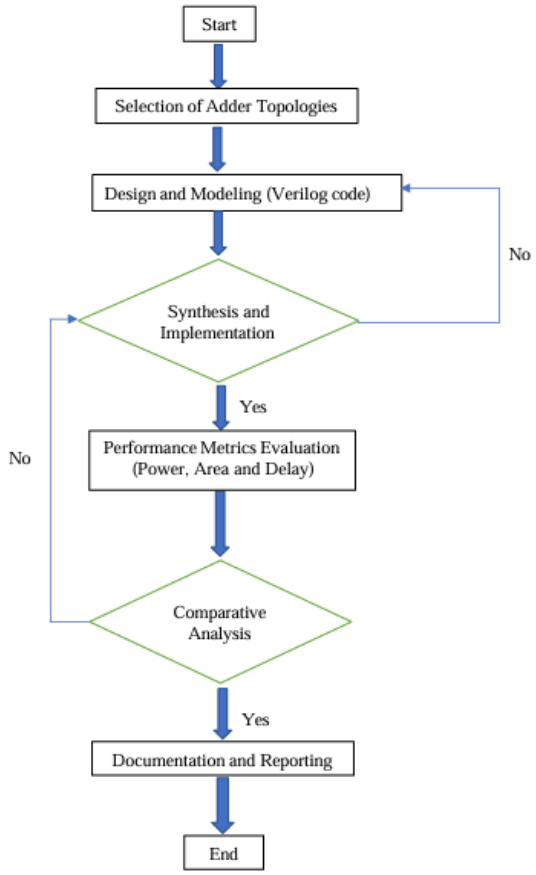

Fig 1: Flow Chart of Methodology.

SOFTWARE REQUIREMENTS

The Xilinx Vivado 2024.2 ML Edition introduces cutting-edge improvements tailored for FPGA and SoC designs, with a strong focus on Machine Learning (ML) capabilities to enhance efficiency and performance.

Enhanced Machine Learning (ML) Features:

- ML-Powered Optimization:
- Machine learning algorithms are integrated to optimize Place and-Route (P&R) processes.
- Provides faster convergence to better Quality of Results (QoR) in terms of timing, power, and area.

1. **Advanced Versal Device Flow:**

Partition-Based Placement:

➢ Improves routability by dividing the design into manageable partitions.

➢ Reduces congestion and enhances the placement process.

Parallel Place-and-Route: Utilizes multi-threading to perform P&R tasks concurrently.

Significantly reduces runtime for large designs.

2. **Top-Level RTL Flow:**

➢ Enables direct configuration of hard IP components (e.g., NoC, transceivers) from top- level RTL.

➢ Simplifies the design process and ensures seamless integration with programmable logic (PL).

3. **Segmented Configuration for Fast Boot:**

Processing System First Boot:

➢ Allows the PS (Processing System) in Versal devices to boot first, bringing up the OS faster.

➢ Defers configuration of PL until required, optimizing boot sequences.

4. **Expanded Device Support:**

Fully supports the latest Xilinx devices, including:

➢ Versal Adaptive SoCs.

➢ Spartan and Artix families for cost-sensitive applications.

➢ Zynq UltraScale+ MPSoCs for embedded systems.

5. Timing and Power Improvements:

Integrated Timing Reports: Enhanced timing analysis tools to identify critical paths and minimize slack.

Power Estimation and Reduction: Provides detailed reports on dynamic and static power. Suggests design adjustments to minimize energy consumption.

6. Visualization and Debugging Enhancements :

Waveform Viewer Improvements:

➢ Displays simulation results with better resolution and interactivity.

➢ Supports advanced signal filtering and grouping. 8. Workflow Automation:

➢ Enhanced Tcl scripting environment for automating repetitive tasks.

➢ Allows users to customize flows based on specific design needs.

➢ Supports running multiple designs or simulations in parallel, improving efficiency.

7. Software and System Requirements:

Operating System:

➢ Supports Windows 10/11 (64-bit) and popular Linux distributions like Ubuntu.

Hardware:

➢ Minimum 8-core CPU, 16 GB RAM (32 GB recommended for large designs).

➢ At least 100 GB free disk space for installation and project files.

8. Benefits of Vivado 2024.2 ML:

- ➢ Faster Design Closure: ML algorithms reduce runtime and improve convergence on design goals.
- ➢ Improved QoR: Enhances performance, reduces power consumption, and optimizes resource usage.
- ➢ Scalability: Handles small FPGA designs to complex SoC systems efficiently.
- ➢ Ease of Use: Streamlined GUI and workflow improvements simplify design processes for engineers.

RESULTS

Performance Metrics

1. Delay:
   ➢ Present delay values for each adder topologies.
   ➢ Use a table or graph to illustrate the delay trends.
2. Power Consumption:
   ➢ Report the power consumption for each design.
   ➢ Highlight which adders are most energy-efficient
3. Area Usage:
   ➢ Include the number of logic elements or gates used.
   ➢ Identify designs that balance area efficiency with performance.

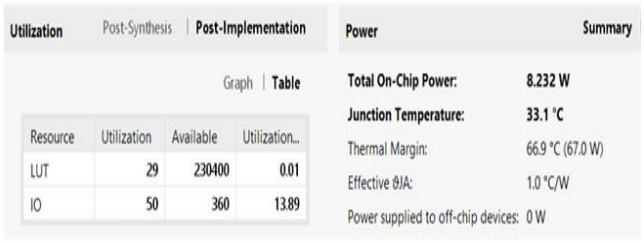

Fig 2: Area Utilization and Power Analysis of Ripple Carry Adder.

Fig 3: Area Utilization and Power Analysis of Carry Select Adder.

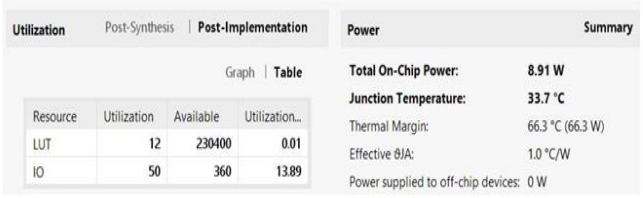

Fig 4: Area Utilization and Power Analysis of Carry Look Ahead Adder.

Fig 5: Area Utilization and Power Analysis of Carry Skip Adder.

Fig 6: Area Utilization and Power Analysis of Kogge Stone Adder.

1. Power Analysis:

• Ripple Carry Adder (RCA): Power consumption is relatively low but increases with input size.

• Carry Look Ahead Adder (CLA): Consumed the least power, making it the most power-efficient topology.

• Carry Select Adder (CSLA): Slightly higher power usage than CLA but efficient under larger input conditions.

• Carry Skip Adder: Moderate power consumption, slightly higher than CSLA and CLA.

• Kogge-Stone Adder (KSA): Exhibited the highest power consumption due to its complexity, even though it has better speed

2. Area Utilization:

• CLA: Consumed the least FPGA resources, making it the most area-efficient.

• KSA: Consumed maximum area due to its parallel prefix computation logic.

• CSLA: Moderate area usage, better optimized compared to KSA.

• RCA: Low area usage but inefficient due to higher delays. • Carry Skip Adder: Moderate resource utilization, striking a balance.

3. Delay (Speed Performance):

• CLA: Fastest computation with the lowest delay among all topologies.

• KSA: Very high speed due to parallel prefix computation but slightly slower than CLA.

• CSLA: Balanced speed, better than RCA but slower than KSA and CLA.

• Carry Skip Adder: Moderate delay.

• RCA: The highest delay due to its sequential nature, making it the slowest.

➢ The implementation of various adder topologies—Ripple Carry Adder (RCA), Carry Look- Ahead Adder (CLA), Carry Select Adder (CSLA), and Kogge-Stone Adder (KSA)—on the Xilinx Zynq UltraScale+ ZCU106 platform revealed significant performance differences.

➢ The Carry Look-Ahead Adder (CLA) emerged as the most efficient overall, offering the fastest computation, minimal power consumption, and optimal area utilization. While the Kogge-Stone Adder (KSA) demonstrated superior speed, its higher power and resource requirements limit its applicability for power-sensitive designs. The results underscore the importance of selecting an adder topology tailored to specific application priorities, whether it be speed, power efficiency, or resource conservation.

CONCLUSION

This project provides a comprehensive analysis of various adder topologies, comparing their design and performance metrics using the Xilinx Vivado platform. The study highlights the strengths and limitations of each topology, offering valuable insights for selecting the most suitable adder design based on specific application requirements.

Key Findings

1. Carry Look-Ahead Adder (CLA):

➢ Emerged as the most efficient topology overall, excelling in speed, power consumption, and area efficiency.

➢ Recommended for applications requiring high performance and balanced resource usage.

2. Kogge-Stone Adder (KSA):

➢ Achieves the fastest computation speeds but at the cost of higher power and resource utilization.

➢ Best suited for speed-critical applications where power and area are secondary concerns.

3. Ripple Carry Adder (RCA):

➢ Simple and resource-efficient but suffers from higher propagation delays.

➢ Suitable for low-cost, low-power systems with minimal performance demands.

4. Carry Select Adder (CSLA) and Carry Skip Adder:

➢ Suitable for low-cost, low-power systems with minimal performance demands.

Here's the conclusion with a comprehensive comparison table summarizing the performance of all five adders:

TABLE 1: Comparison among Various types of Adder Topologies.

Adders	Delay (ns)	Area (LUT, IOB)	Temperature (C)	Power (w)
Ripple Carry Adder	4.6697	29(0.01%), 50(13.89%)	33.1	8.232
[1]	12.578	21(0.05%), 49(16.33%)	78.5	10.7
Carry Select Adder	4.4362	25(0.01%), 50(13.89%)	34.4	9.616
[1]	10.212	30(0.14%), 49(46.23%)	81	11.204
Carry Skip Adder	4.6341	23(0.01%), 50(13.89%)	34.1	9.244
[1]	15.834	21(0.10%), 49(46.23%)	82.6	11.512
Carry Look Ahead Adder	4.0385	12(0.02%), 50(13.89%)	33.7	8.91
[1]	12.519	21(0.01%), 49(46.23%)	79	10.802
Kogge Stone Adder	4.5314	40(0.02%), 49(13.61%)	34.2	9.353
[1]	8.641	41(0.197%), 49(46.23%)	80.3	11.057

FUTURE WORK

1. Exploration of Advanced Adder Topologies: Investigate newer or less conventional adder architectures that may offer advantages in emerging technologies, such as quantum computing or neuromorphic systems.
2. Machine Learning for Design and Evaluation: Employ machine learning algorithms to predict optimal adder designs for specific applications based on given performance constraints.
3. Hardware Platform Expansion: Test the designs on various FPGA platforms or ASICs to generalize findings and identify platform-specific optimizations.
4. Energy-Efficient Designs for IoT: Investigate ultra-low-power adder designs tailored for Internet of Things (IoT) applications where power consumption is a critical factor.
5. Software Tool Comparisons: Extend the study to include other design and simulation tools, comparing their efficiency and ease of use with Xilinx Vivado.
6. Scaling and Integration: Scale the designs to larger input sizes and assess their performance. Explore integration with more complex systems, such as multipliers or entire arithmetic logic units.
Figures and Tables

Final Remark

When comparing all the parameters, the Carry Look Ahead Adder (CLA) emerges as the best overall choice in terms of:

- Fastest computation (lowest delay)
- Most area-efficient (least resource usage)
- Most power-efficient (least power consumption)

While the Kogge-Stone Adder (KSA) has a faster computation speed compared to others, its higher resource usage and power consumption make it less ideal for applications that prioritize power and area efficiency.

Thus, CLA would be the most suitable choice if the goal is to optimize for both speed and resource usage. By systematically addressing these challenges, the project successfully provided a detailed performance comparison and practical insights for selecting the optimal adder topology for various applications..

REFERENCES

[1] Tammana M, Vardhan M, Geethu R S, Abhimanyu S, Anujith A, "Efficiency Analysis of Adder Architectures: A Comparative Study Across Various Bit Lengths," International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), pp. 1-6, June 2024

[2] Ajit A, Arathi P V, Haridas K, Nambiar N M, Devi S, "FPGA based performance comparison of different basic adder topologies with parallel processing adder," 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 87-92, IEEE, 2019

[3] Harish B, Sivani K, Rukmini M S S, " Design and performance comparison among various types of adder topologies," 3rd international conference on computing methodologies and communication (ICCMC), pp. 725-730, IEEE, 2019.

[4] Sarkar S, Sarkar S, Mehedi Jm, "Comparison of various adders and their VLSI implementation," International Conference on Computer Communication and Informatics (ICCCI), pp. 1-9, IEEE, 2018

[5] Koyada B, Meghana N, Jaleel M O, Jeripotula P R, " A comparative study on adders," International conference on wireless communications, signal processing and networking (WiSPNET), pp. 2226-2230, IEEE, 2017.

[6] Maroju SaiKumar, Dr P. Samundiswary. "Design and performance analysis of various adders using

verilog, " International journal of computer science and mobile computing 2, pp: 128-138, 2013.

[7] S. Akhter, V. Saini and J. Saini, "Analysis of vedic multiplier using various adder topologies," 4th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 173-176, 2017

[8] Mohanapriya, D, Dr N. Saravanakumar, "A comparative analysis of different 32-bit adder topologies with multiplexer based full adder," Int J Eng Sci 1, pp. 4850-4854, 2016.

[9] Tripathy S, Omprakash L B, Patro B S, Mandal S K, “A comparative analysis of different 8-bit adder topologies at 45 nm technology,” International journal of engineering research and technology, Oct;2(10), 2013.

[10] Goel, Sumeer, Ashok Kumar, and Magdy A. Bayoumi. “Design of robust energy-efficient full adders for deep-submicrometer design using hybrid-CMOS logic style,” IEEE Transactions on Very Large-Scale Integration (VLSI) Systems 14, no. 12, pp: 1309-1321, 2006.

[11] Ananthakrishnan, Anaswar Ajit, Arathi P V, Kiran Haridas, “FPGA Based Performance Comparison of Different Basic Adder Topologies with Parallel Processing Adder,” 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), : 2019