IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Survey of Efficient Multiplier Architectures Using Optimized Reduction and Fast Addition Techniques

Mrs. Amrapali Nilesh Nirmal[1], Dr. Hemant T. Inale[2], Dr. Vijay D Chaudhari[3], Hemraj V Dhande[4], Prof. S. K. Chaudhari[5]

[1]PG Student : 0009-0004-1040-5383,

[2]Professor, [3,4] Asso. Prof. E&TC Engg dept, GFGCOE,Jalgaon,India,Pin- 425003

[2] :0009-0003-2174-5669, [3] 0009-0007-9192-6907,

[5]Asso.Prof. E&TC Engg dept,Govt CoE Jalgaon-425001, MS, India

amrapali.nirmal@gmail.com[1]

Abstract *–This paper proposes an improved Dadda multiplier architecture aimed at achieving high speed and area efficiency in digital processing systems. Since multipliers consume a significant portion of hardware resources, optimizing their performance is critical in VLSI design. The proposed approach enhances the conventional Dadda multiplier by incorporating 4:2 compressors in the partial product reduction stage, which helps reduce critical path delay. Additionally, parallel prefix adders are employed in the final summation stage to further improve computational speed.*

Four different implementations are developed using Sklansky, Kogge–Stone, Brent–Kung, and Ladner–Fischer adders. The designs are simulated using Xilinx ISE 14.7 and compared with the traditional Dadda multiplier in terms of area and delay. Results demonstrate that the proposed architecture effectively reduces latency while maintaining low area complexity, making it suitable for high-performance VLSI and DSP applications.

Keywords- *Dadda Multiplier, 4:2 Compressor, Parallel Prefix Adders, VLSI Design, High-Speed Arithmetic, Area Efficiency, Propagation Delay, LUT Utilization, DSP Applications*

INTRODUCTION

Multiplication is one of the most important operations in Digital Signal Processing (DSP) systems, used in applications like filtering, FFT, image processing, and machine learning. In VLSI design, the efficiency of a multiplier directly affects system speed, power consumption, and hardware area. Since multiplication depends heavily on addition, the choice of adder architecture—such as Ripple Carry, Carry Look-Ahead, Carry Save, and Carry Select—plays a key role in overall performance.

Over time, various multiplier designs like array, Booth, Wallace tree, Dadda, and Vedic multipliers have been developed, each offering different advantages in terms of speed and complexity. More recent approaches, such as approximate multipliers and carry-save techniques, focus on reducing power and delay while maintaining acceptable accuracy.

Overall, modern research aims to design multipliers that are faster, more energy-efficient, and suitable for high-performance DSP applications, especially in FPGA and VLSI-based systems.

LITERATURE REVIEW

Efficient arithmetic units play a crucial role in determining the performance of digital systems, particularly in processors and digital signal processing (DSP) applications where addition and multiplication operations dominate computation. Several researchers have explored optimized adder and multiplier architectures to improve speed, reduce hardware area, and enhance overall system efficiency.

1.Maroju SaiKumar and Dr. P. Samundiswary provides a clear and practical comparison of seven widely used adder architectures—RCA, CSkA, CIA, CLA, CSA, CSlA, and CBA—implemented using Verilog HDL on a Virtex-6 FPGA platform. The authors designed, simulated, and synthesized each adder using Xilinx ISE 13.2, focusing on two important performance factors: hardware area (in terms of LUTs and slices) and propagation delay.

From the results, it is observed that RCA, CSkA, CIA, and CSlA consume less area, making them suitable for designs where hardware resources are limited. In contrast, the Carry Save Adder (CSA) offers the fastest performance because it avoids carry propagation, though it is mainly useful in multi-operand addition rather than final computation stages. Among all the architectures, the Carry Increment Adder (CIA) emerges as the most balanced option, providing both good speed and efficient area usage.

Overall, the study emphasizes the importance of choosing the right carry-handling technique, as it directly affects performance. Although power consumption was not analyzed, the findings give useful direction for selecting suitable adders in VLSI and DSP system design [1].

2. This paper by C. Karthi focuses on improving the speed and reliability of multiplication in Digital Signal Processing (DSP) applications such as convolution and Fast Fourier Transforms (FFT). It begins by reviewing conventional multipliers like Array, Booth, Carry Save Array, and Wallace tree, pointing out their limitations in terms of speed and efficiency. To address this, the study introduces Vedic multipliers based on ancient Indian mathematical techniques, particularly the Urdhva Tiryakbhyam Sutra for general multiplication and the Nikhilam Sutra for large numbers.

The main idea is to use a Vedic multiplier architecture to achieve faster computation and then apply the Digital Root (DR) method to verify the correctness of input and output signals after DSP operations. The results show that the Vedic multiplier outperforms traditional multipliers in terms of speed. It is also observed that Urdhva Tiryakbhyam works better for smaller bit sizes, while Nikhilam becomes more efficient as the number size increases. Additionally, Pass Transistor Logic (PTL) provides better performance compared to CMOS and CPL implementations.

Overall, the study highlights that combining fast Vedic multiplication with a simple DR-based verification method can significantly enhance both performance and reliability in DSP systems [2].

3.This paper by Sunkireddy Sahithi, Sahithi Thallapally, Ganesh Avula, and B. Shivani Gayathri Devi focuses on improving multiplication performance, which is a key operation in applications like image processing and digital signal processing. Traditional multipliers often become slower and consume more power as the size of the data increases. To address this, the authors propose the use of the Dadda Multiplier, which uses an optimized tree-like structure to reduce the number of partial products and speed up computation.

The implementation involves three main steps: generating partial products, reducing them in stages using adders based on specific height limits, and finally summing the reduced outputs. The design was implemented on an FPGA and evaluated based on area, delay, and power consumption. The results for 8×8, 16×16, and 32×32 multipliers show a gradual increase in resource usage and delay, but still maintain efficient performance.

Overall, the study finds that the Dadda Multiplier performs better than traditional multipliers like the Array Multiplier in terms of speed and efficiency. This makes it a strong candidate for high-performance and power-efficient digital systems [3].

4.This paper by V. Kiran Kumar Reddy and K. Sravan Kumar explores the design of high-performance and area-efficient DSP architectures using Vedic multipliers. The study compares traditional multiplication techniques like Booth and Array multipliers with Vedic methods derived from ancient Indian mathematics. The main goal is to achieve faster computation, reduced power consumption, and efficient hardware utilization in VLSI systems.

The authors focus on two important Vedic Sutras: Urdhva Tiryakbhyam, which enables parallel generation of partial products for faster operation, and Nikhilam, which simplifies multiplication of large numbers near a base. These techniques are implemented using Verilog and synthesized on a Xilinx Spartan-3 FPGA. The design uses modular structures and efficient adders like Carry Save Adders to enhance performance.

The results clearly show that Vedic multipliers outperform traditional methods. While Booth and Array multipliers exhibit higher delays, Vedic approaches significantly reduce computation time, with the combined Nikhilam and Urdhva method achieving the fastest performance.

Overall, the study concludes that Vedic multipliers provide better speed, scalability, and area efficiency, making them highly suitable for modern DSP and ALU applications, despite minor limitations for very large bit sizes.[4]

5.This paper by V. Kiran Kumar Reddy and K. Sravan Kumar presents an efficient DSP architecture aimed at improving speed, area, and power performance. The core idea is the use of a Flexible Computational Unit (FCU) that operates directly on carry-save (CS) data. By avoiding frequent conversion to binary form, the design reduces computation time and improves overall efficiency.

The FCU is supported by predefined operation templates that allow it to perform complex DSP tasks involving both addition and multiplication. A key strength of the design is the use of carry-save arithmetic and compressor trees, which help minimize carry propagation delays and reduce latency during multi-operand operations.

The paper compares two multiplier techniques: Modified Booth (MB) and the proposed Dadda multiplier. While MB reduces partial products, it introduces higher complexity. In contrast, the Dadda multiplier efficiently compresses partial products using fewer reduction stages, leading to faster and more optimized performance.

The results, obtained using a Virtex-6 FPGA, show that the Dadda-based design reduces area by 15.6%, power consumption by 13.8%, and delay by 15.5% compared to the MB approach. Overall, the study demonstrates that the proposed architecture significantly enhances DSP performance.[5]

6.This paper, "FPGA-Based Multiplier with a New Approximate Full Adder for Error-Resilient Applications," presents an innovative approach to improving the efficiency of multipliers used in DSP and image processing systems. Instead of relying on exact computations, the study adopts approximate computing, where small errors are acceptable in exchange for significant gains in speed, power, and area.

At the core of the design is a simplified approximate full adder, which uses minimal logic—generating the Sum with an OR gate and taking Carry-out directly from an input. Although this introduces minor inaccuracies, the error remains very low and does not significantly affect overall system performance. This adder is then used to build an 8×8 approximate multiplier with optimized reduction stages and efficient counter structures.

The implementation on a Xilinx Spartan-3 FPGA shows impressive improvements, including reduced hardware usage, lower power consumption, and faster operation. The design achieves significant reductions in power and power-delay product while maintaining good output quality, as indicated by high PSNR and SSIM values.

Overall, the work demonstrates that approximate designs can effectively balance performance and accuracy, making them suitable for energy-efficient applications like image processing and edge AI systems.[6]

. 7.This paper, "Very-Large-Scale Integration (VLSI) Implementation and Performance Comparison of Multiplier Topologies for Fixed- and Floating-Point Numbers," presents a detailed comparison of three common multiplier architectures: array, Wallace tree, and radix-4 Booth multipliers. The study evaluates these designs across different word lengths and focuses on their application in floating-point operations used in modern computing systems.

The multipliers are designed using a complete VLSI flow with the Alliance open-source toolchain, ensuring a realistic comparison. The array multiplier is simple and easy to implement but suffers from higher delay and area as size increases. In contrast, the Wallace tree multiplier uses parallel reduction of partial products, making it significantly faster, especially for small to medium word lengths. The radix-4 Booth multiplier improves efficiency by reducing the number of partial products, making it suitable for signed operations.

Results show that the Wallace tree multiplier offers the best speed for smaller bit sizes, while for larger word lengths, both Wallace tree and Booth multipliers perform similarly. The array multiplier consumes the most area and is less efficient for high-precision tasks.

Overall, the study concludes that Wallace tree and Booth multipliers are better suited for high-performance DSP and floating-point applications.[7]

8.This paper by Shubham Hemant Marode and Neetesh Raghuwanshi focuses on designing an efficient IEEE-754 floating-point multiplier to improve speed and energy efficiency in modern processors. Floating-point operations are essential in applications like DSP, robotics, and scientific computing, where both accuracy and performance are critical. The study considers both single-precision and double-precision formats, explaining their structure in terms of sign, exponent, and mantissa.

The multiplication process is carried out in four steps: determining the sign using XOR, adding exponents with bias adjustment, multiplying significands (including the hidden bit), and normalizing the result. To enhance performance, the authors introduce architectural optimizations, particularly using a modified Carry Select Adder (CSA) combined with a Binary to Excess-1 Converter (BEC). This approach reduces hardware complexity and improves speed compared to traditional designs.

Additionally, a partitioned multiplier technique is used to handle large bit-width operations efficiently. The design was implemented in VHDL and tested using Xilinx ISE. Results show significant improvements in both area and delay compared to earlier methods.

Overall, the study demonstrates that optimized adder structures and partitioning techniques can greatly enhance floating-point multiplier performance.[8]

3. DISCUSSION

S. No.	Title & Authors	Technique / Architecture Used	Objective of the Work	Key Methodology	Performance Metrics Considered	Major Findings / Contributions	Limitations / Research Gap
1	Design and Performance Analysis of Various Adders using Verilog Maroju Sai Kumar, Dr. P. Samundiswary	Ripple Carry Adder, Carry Look-Ahead Adder, Carry Skip Adder, Carry Select Adder	To analyze and compare different adder architectures for digital systems	Verilog-based implementation and simulation of multiple adder designs	Delay, Area, Power	Carry Look-Ahead and Carry Select Adders offer reduced delay compared to Ripple Carry Adder	Study limited to adders only; impact on multiplier or DSP systems not explored

2	Digital Roots (DR) Method Verification for DSP Convolution in Vedic Multiplier I/O C. Karthi	Vedic Multiplier with Digital Root Verification	To verify correctness of convolution operations in DSP using digital root technique	Application of digital root method for validating Vedic multiplier input/output	Accuracy, Speed of Verification	Digital root technique provides fast and efficient verification of DSP convolution results	Focuses on verification only; hardware optimization and power analysis not addressed
3	Implementing and Analysing the Performance of Dadda Multiplier on FPGA Sahithi Sunkireddy et al.	Dadda Multiplier	To implement and analyze Dadda multiplier performance on FPGA	FPGA implementation using Verilog/VHDL and synthesis	Delay, LUT utilization, Power	Dadda multiplier shows reduced delay and efficient resource utilization	Comparison with approximate or hybrid multipliers is missing
4	High Performance and Area Efficient DSP Architecture using Dadda Multiplier V. Kiran Kumar Reddy, K. Sravan Kumar	Dadda Multiplier-based DSP Architecture	To improve DSP performance using optimized multiplier architecture	Integration of Dadda multiplier in DSP processing blocks	Speed, Area Efficiency	Dadda multiplier significantly improves throughput and reduces hardware area	Power optimization and error tolerance not considered
5	Analysis of Vedic Multiplier Akshata R.	Vedic Multiplier (Urdhva Tiryagbhyam Sutra)	To study the efficiency of Vedic multiplication techniques	Architectural analysis and simulation	Speed, Area	Vedic multiplier provides faster multiplication compared to conventional methods	Scalability and FPGA/ASIC implementation not deeply analyzed
6	FPGA-Based Multiplier with a New Approximate Full Adder for Error-Resilient Applications	Approximate Full Adder-based Multiplier	To reduce power and delay in error-tolerant applications	Simplified FA logic integrated into multiplier design	Power, Delay, Error Distance	Significant reduction in power and delay with acceptable error levels	Not suitable for accuracy-critical applications like financial computing
7	VLSI Implementation and Performance Comparison of Multiplier Topologies for Fixed- and Floating-Point Numbers	Array, Wallace, Dadda, Floating-Point Multipliers	To compare different multiplier architectures for fixed and floating-point arithmetic	ASIC/VLSI implementation and simulation	Area, Power, Delay	Dadda multiplier performs better for fixed-point; floating-point multipliers incur higher complexity	Hybrid and approximate techniques not explored
8	VLSI Design and Analysis of IEEE 754 Floating Point Multiplier Marode Shubham Hemant, Dr. Neetesh Raghuwanshi	IEEE 754 Floating Point Multiplier	To design and analyze a compliant floating-point multiplier	VLSI-based design and synthesis	Accuracy, Area, Delay	Accurate floating-point multiplication achieved as per IEEE standards	High area and power overhead compared to fixed-point designs

- **Dadda multipliers** are preferred for **high-speed and area-efficient DSP applications**.
- **Vedic multipliers** offer simplicity and speed but require optimization for large bit-widths.
- **Approximate multipliers and adders** are gaining importance in **error-resilient applications** such as image and signal processing.
- **Floating-point multipliers** ensure accuracy but incur higher area and power costs.

Recent research emphasizes **low latency, reduced switching activity, and energy efficiency**

CONCLUSION

The Dadda multiplier emerges from the literature as one of the most efficient architectures for high-performance VLSI and DSP applications. Its key strength lies in the optimized partial product reduction strategy, which minimizes the number of reduction stages compared to conventional array and Wallace multipliers. By carefully controlling the height of the partial product matrix at each stage, the Dadda multiplier achieves lower critical path delay while maintaining reduced hardware complexity. FPGA- and ASIC-based implementations consistently demonstrate improvements in speed and area efficiency, making it well suited for real-time signal processing, inner product computation, and high-throughput arithmetic units. When integrated into DSP architectures, the Dadda multiplier enhances overall system performance without significantly increasing resource utilization. However, power optimization and scalability to very high bit-widths still present challenges, particularly in deep submicron technologies. Future research can further strengthen the Dadda multiplier by combining it with low-power techniques, approximate adders, or hybrid reduction schemes to meet the growing demands of energy-efficient and error-resilient computing systems.

REFERENCES

1. Maroju, SaiKumar, and P. Samundiswary. Design and Performance Analysis of Various Adders Using Verilog. n.d.
2. Karthi, C. “Digital Roots (DR) Method Verification for Digital Signal Processors Convolution in Vedic Multiplier I/O.” Proceedings of the International Conference on Digital Signal Processing, n.d.
3. Sunkireddy, Sahithi, Sahithi Thallapally, Ganesh Avula, and B. Shivani Gayathri Devi. Implementing and Analysing the Performance of Dadda Multiplier on FPGA. n.d.
4. Reddy, V. Kiran Kumar, and K. Sravan Kumar. “High Performance and Area Efficient DSP Architecture Using Dadda Multiplier.” International Journal of Engineering Research and Technology, n.d.
5. Akshata, R. Analysis of Vedic Multiplier. n.d.
6. FPGA-Based Multiplier with a New Approximate Full Adder for Error-Resilient Applications. n.d.
7. Very-Large-Scale Integration (VLSI) Implementation and Performance Comparison of Multiplier Topologies for Fixed- and Floating-Point Numbers. n.d.
8. Marode, Shubham Hemant, and Neetesh Raghuwanshi. “VLSI Design and Analysis of IEEE 754 Floating Point Multiplier.” International Journal of Engineering and Advanced Technology, n.d.
9. Usman, Muhammad, Miloš D. Ercegovac, and Jeong-A Lee. “Low-Latency Online Multiplier with Reduced Activities and Minimized Interconnect for Inner Product Arrays.” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, n.d.
10. Patil, Pragnya. Complex Multiplier Architectures on VLSI. n.d.