

Comparative Analysis of Energy-Efficient Approximate Multiplier and Wallace Tree Multiplier for Error Tolerant Applications

Mr.B.Manoj Kumar

Assistant Professor
Dept of ECE

Annamacharya Institute of Technology
and Sciences
Tirupati, India

Manojkumar.bollini@gmail.com

Dharmalingam Neesanthi

B tech Student
Dept of ECE

Annamacharya Institute of Technology
and Sciences
Tirupati, India

nr4732832@gmail.com

Shaik Mubeena

B tech Student
Dept of ECE

Annamacharya Institute of Technology
and Sciences
Tirupati, India

honeyq373425@gmail.com

Meegada Madhavi

B tech Student
Dept of ECE

Annamacharya Institute of Technology
and Sciences
Tirupati, India

Madhavireddy0611@gmail.com

Sathipati Lakshmi Narayana Chethan

B tech Student
Dept of ECE

Annamacharya Institute of Technology
and Sciences
Tirupati, India

Slakshminarayanachethan123@gmail.com

Abstract— Multiplication is a fundamental operation in digital signal processing, artificial intelligence, and multimedia applications, where performance and energy efficiency are critical design requirements. Conventional architectures such as the Wallace tree multiplier provide high-speed and accurate computation through parallel partial product reduction, but they incur considerable hardware complexity and switching activity. In contrast, energy-efficient approximate multipliers intentionally simplify lower significant bit computations to reduce power consumption while maintaining acceptable accuracy for error-tolerant applications. This paper presents a comparative analysis of an energy-efficient approximate multiplier and a Wallace tree multiplier in terms of computational efficiency, error characteristics, and suitability for tolerant systems. The proposed study evaluates performance trade-offs using relevant design metrics and demonstrates that approximate multipliers offer significant improvements in energy savings with minimal impact on output quality, making them suitable for image processing, machine learning, and embedded applications where perfect accuracy is not mandatory.

Keywords: Approximate multiplier, Wallace tree multiplier, energy efficiency, error-tolerant applications, low-power VLSI design, switching activity, truncation technique.

1. INTRODUCTION

Multiplication is one of the most critical arithmetic operations in modern digital systems, particularly in digital signal processing (DSP), artificial intelligence (AI), image processing, and embedded computing platforms. In such systems, multipliers dominate the overall hardware complexity, power consumption, and delay of arithmetic units, especially in Multiply-Accumulate (MAC) architectures. As technology scales and portable devices demand longer battery life, energy-efficient circuit design has become a primary objective in Very Large-Scale Integration (VLSI) systems. The Wallace tree multiplier is a high-speed multiplier architecture that reduces partial products using a tree-based structure of carry-save adders. By compressing partial products in parallel stages, it

minimizes propagation delay compared to conventional array multipliers. However, the extensive parallelism and large number of adders increase switching activity and hardware resource utilization, leading to higher dynamic power consumption.

In contrast, approximate multipliers are designed based on the principle of error-resilient computation. These architectures intentionally relax exact accuracy by simplifying computations in the least significant bits (LSBs) or truncating partial products, thereby reducing logic complexity and switching transitions. Since many real-world applications such as image processing, machine learning inference, and multimedia processing can tolerate small computational errors without significant degradation in output quality, approximate computing offers a promising trade-off between accuracy and energy efficiency.

This work presents a comparative analysis of an energy-efficient approximate multiplier and the Wallace tree multiplier for error-tolerant applications. The study focuses on evaluating architectural differences, computational efficiency, error characteristics, and suitability for low-power systems. By analyzing design trade-offs between accuracy, delay, and energy consumption, the proposed comparison aims to identify an optimized multiplier architecture suitable for next-generation energy-constrained digital applications.

Partial Product Generation: In the Wallace tree multiplier, all $N \times N$ partial products are generated accurately using AND gates, increasing hardware usage and switching activity. In approximate multipliers, LSB partial products are simplified or removed to reduce logic complexity, area, and dynamic power, introducing small and controlled computational errors.

Reduction Stage: The reduction stage uses Carry Save Adders arranged in a tree structure to compress multiple partial product rows into two rows. Carry bits are saved and forwarded to the next level without propagation. This parallel reduction reduces delay significantly but requires a large number of adders and increases area.

Final Adder Stage: After reduction, two rows of bits are added using a fast carry propagation adder such as a ripple carry or carry look-ahead adder. Full carry propagation ensures accurate final output. However, this stage contributes to increased critical path delay and higher power consumption.

MAC Integration Stage: In Multiply–Accumulate units, the Wallace tree multiplier performs accurate multiplication followed by precise accumulation. This ensures high computational accuracy in DSP and scientific applications. However, continuous switching and full carry propagation increase overall energy consumption and hardware utilization in large-scale digital systems.

2. LITERATURE REVIEW

Multipliers are indispensable arithmetic units in digital signal processing (DSP), neural network accelerators, embedded systems, and multimedia applications. Traditional multiplier architectures, such as the Wallace Tree Multiplier, have been extensively studied for their speed and accuracy. The Wallace tree structure compresses generated partial products using carry-save adders (CSAs) in a tree configuration, followed by a final carry propagate adder. This approach significantly reduces propagation delay compared to simple array multipliers, making it suitable for high-performance computing. However, the inherently high level of parallelism and large number of adders result in increased switching activity, larger silicon area, and higher power dissipation—factors that are critical constraints in energy-limited and battery-operated systems.

Recent research (2020–2024) has focused on improving multiplier architectures for energy-constrained and error-tolerant systems such as AI accelerators, edge devices, and multimedia processors. Modern implementations of the Wallace tree multiplier emphasize optimized compressor structures (4:2 and 5:2 compressors), low-power carry-save adder (CSA) networks, and hybrid final adders to reduce critical path delay and switching activity. Several studies between 2020 and 2022 reported improved Wallace tree implementations on FPGA and ASIC platforms, achieving reduced delay while maintaining exact computation, making them suitable for high-speed DSP and cryptographic applications.

Between 2021 and 2024, significant advancements have been made in energy-efficient approximate multipliers targeting neural network inference and image processing applications. Researchers proposed segmented and truncated multiplier architectures where the least significant bits (LSBs) are approximated while the most significant bits (MSBs) remain accurate. These works demonstrate power reductions ranging from 30% to 60% with minimal degradation in application-level performance metrics such as classification accuracy and image PSNR. Recent designs also incorporate approximate 4:2 compressors and error-configurable logic blocks to dynamically adjust accuracy based on workload requirements.

A. Research Gap

Although recent studies have improved both energy-efficient approximate multipliers and optimized Wallace tree multiplier designs, a comprehensive comparative evaluation remains limited. Most research focuses on either exact high-speed architectures or standalone approximate models without detailed stage-level analysis of partial product generation, reduction, truncation, and MAC integration. Additionally, multi-metric evaluation considering power, delay, switching activity, error metrics, and real application performance (such as neural network accuracy or image quality) is insufficient. Scalability to higher bit-width designs and adaptive approximation techniques are also underexplored. Therefore, a systematic structural and performance comparison is required for error-tolerant applications.

3. EXISTING SYSTEM

The existing system primarily relies on conventional exact multipliers such as array multipliers and Wallace tree multipliers to perform arithmetic operations. These multipliers are designed to provide highly accurate results, making them suitable for critical computations. However, they suffer from drawbacks like high power consumption, large chip area, and increased computational delay. In applications such as image processing, machine learning, and multimedia systems, perfect accuracy is not always required. Despite this, traditional systems do not take advantage of error tolerance, leading to inefficient performance. As a result, existing systems are not well-suited for modern applications that demand high speed and energy efficiency.

The main problem in the existing system is that it uses exact multipliers, which always produce precise results but are not optimized for efficiency. These systems consume high power, making them unsuitable for battery-operated and portable devices. They also require large hardware area, increasing cost and complexity. Additionally, exact multipliers introduce higher delay, reducing overall system speed. In many applications like image processing, a small error is acceptable, but existing systems do not utilize this advantage. Hence, the major issues are power inefficiency, large area, and slower performance, which limit their use in modern, high-speed, and energy-efficient applications.

4. PROPOSED METHOD

The proposed work focuses on the design and comparative analysis of an energy-efficient approximate multiplier and a Wallace Tree multiplier for error-tolerant applications. Initially, both multipliers are modeled using Verilog Hardware Description Language (HDL) and implemented in Xilinx Vivado. In the approximate multiplier, truncation and approximate adders are employed in the partial product reduction stage to reduce hardware complexity, power consumption, and delay. In contrast, the Wallace Tree multiplier utilizes exact compressors and carry-save adders to achieve high accuracy and speed. Both designs are simulated using identical input test vectors to ensure fair comparison. Performance metrics such as power, delay, area (LUT utilization), and error parameters including Mean Error Distance (MED), Mean Relative Error Distance (MRED), and Normalized Error Distance (NED) are evaluated. Based on the obtained results, a detailed comparison is carried out to identify

the most suitable multiplier architecture for error-tolerant applications.

Approximate Compressor (CAC): In architecture, with a higher count of negative errors, a compensator circuit is typically required to manage data flow between the

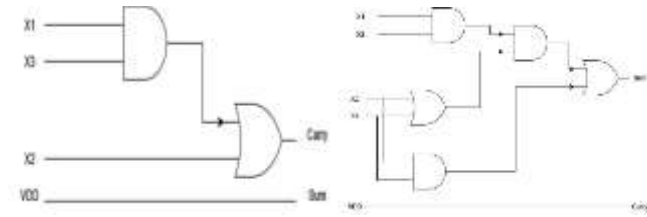


Fig. 1(a) : ACMLC

Fig. 1(b) : CAC

5. METHODOLOGY

A. APPROXIMATE MULTIPLIER

The system overview describes the overall structure and working process of the proposed multiplier comparison system. In this project, two types of multipliers, namely the Wallace Tree Multiplier and the Energy-Efficient Approximate Multiplier, are designed and implemented using Verilog Hardware Description Language (HDL). The purpose of the system is to analyze and compare the performance of these multipliers in terms of power consumption, area utilization, delay, and accuracy.

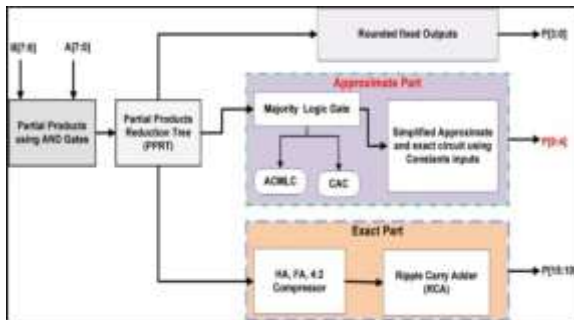


Fig. 2. Design flow of 8-bit multiplier

The above figure illustrates a hybrid multiplier architecture consisting of both approximate and exact computation parts to achieve a balance between energy efficiency and accuracy.

In the approximate part, the most significant bits are processed using Partial Product Reduction Tree (PPRT) followed by majority logic and simplified approximate circuits. Modules such as Approximate Carry Merge Logic Cell (ACMLC) and Carry Approximation Cell (CAC) are used to reduce hardware complexity and power consumption. These components simplify carry propagation, resulting in faster computation with acceptable error. The approximate section produces an output $P[n-1]P[n-1]P[n-1]$, which is a rounded fixed output.

In the exact part, the least significant bits are processed using accurate arithmetic units such as half adders (HA), full adders (FA), and 4:2 compressors. The final summation is carried out using a ripple carry adder to maintain computational accuracy. This section produces the output $P[15:0]P[15:0]P[15:0]$, ensuring correctness in critical bits.

Overall, the architecture combines approximate and exact computing techniques, where the approximate section improves energy efficiency and speed, while the exact section preserves accuracy. This hybrid approach is well suited for error-tolerant applications such as image processing and digital signal

processing, where minor inaccuracies can be tolerated in exchange for reduced power and area.

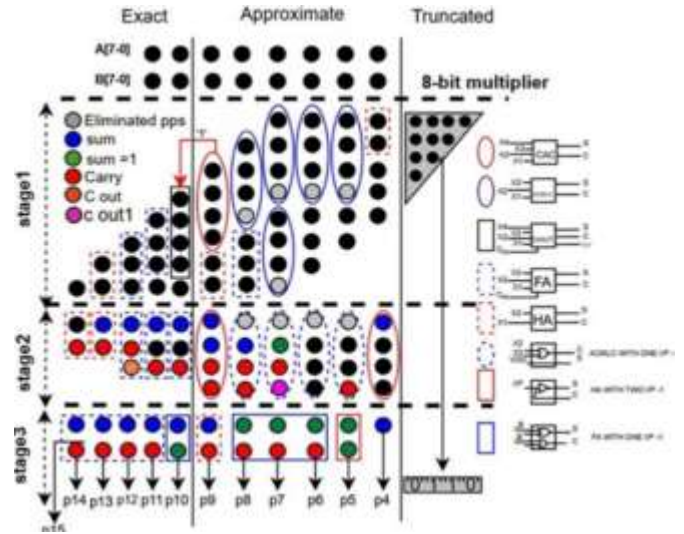


Fig. 3. Dot diagram of Proposed Approximate multiplier.

The figure illustrates an 8-bit approximate multiplier architecture that combines exact, approximate, and truncated regions to achieve energy-efficient computation. The inputs $A[7:0]A[7:0]A[7:0]$ and $B[7:0]B[7:0]B[7:0]$ generate partial products, which are arranged in a matrix form.

The partial products are divided into three regions: exact, approximate, and truncated. In the exact region (most significant bits), accurate computation is performed using full adders (FA) and half adders (HA) to maintain precision in critical outputs. In the approximate region, simplified arithmetic units such as approximate full adders and compressors are used. These components reduce carry propagation and hardware complexity, leading to lower power consumption and faster operation, while introducing minimal error. The truncated region (least significant bits) eliminates some partial products entirely, thereby reducing the number of computations and saving area and energy. This truncation introduces a small error but has limited impact on overall output accuracy.

The reduction process is carried out in multiple stages (stage 1 to stage 3), where partial products are progressively summed using adders and compressors. Finally, the outputs from all regions are combined to produce the result bits $p_{15}p_{14}p_{13}p_{12}p_{11}p_{10}p_9p_8p_7p_6p_5p_4p_3p_2p_1p_0$.

Overall, this architecture achieves a trade-off between accuracy and efficiency by preserving exact computation in significant bits while applying approximation and truncation in less critical regions, making it suitable for error-tolerant applications such as image and signal processing.

B. Wallace Tree Multiplier

The figure illustrates the step-by-step reduction process of partial products in a Wallace Tree multiplier. Initially, the partial products are generated and arranged in a matrix form based on the multiplication of input bits. These bits are grouped column-wise according to their significance.

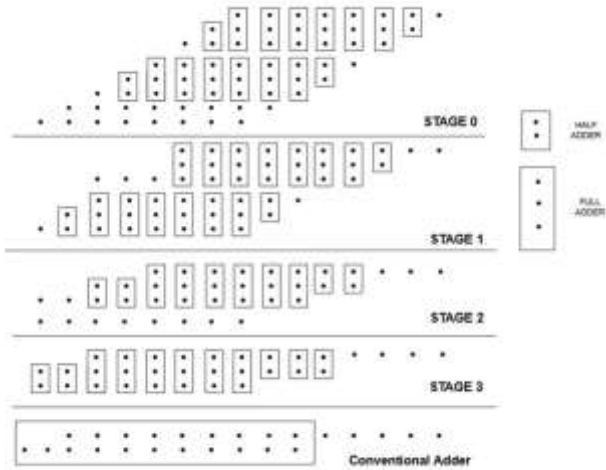


Fig. 4. Wallace Tree Multiplier for 8X8 bits.

In Stage 0, the partial products are reduced using half adders (HA) and full adders (FA). A full adder takes three input bits and produces a sum and carry, while a half adder processes two bits. This stage reduces the number of rows by converting multiple bits into fewer outputs.

In Stage 1, the outputs from the previous stage are further reduced using the same principle. The carries generated are forwarded to the next higher column, ensuring proper alignment of bit significance.

In Stage 2 and Stage 3, the reduction process continues iteratively. At each stage, the number of partial product rows decreases significantly due to parallel reduction, which improves computational speed.

Finally, when only two rows of partial products remain, a conventional adder (such as a Ripple Carry Adder or Carry Look-Ahead Adder) is used to produce the final multiplication result.

Overall, the Wallace Tree structure minimizes delay by performing parallel reduction of partial products, making it faster than traditional sequential multiplier.

6. RESULT ANALYSIS

Simulation and result analysis are important stages in the FPGA design flow of Xilinx Vivado. These stages help designers verify the functionality and performance of the digital design before implementing it on hardware. Simulation allows the designer to test the behavior of the design using testbench inputs without physically programming the FPGA device.



Fig. 5. RTL simulation for Approximate Multiplier.



Fig. 6. Power for Approximate Multiplier

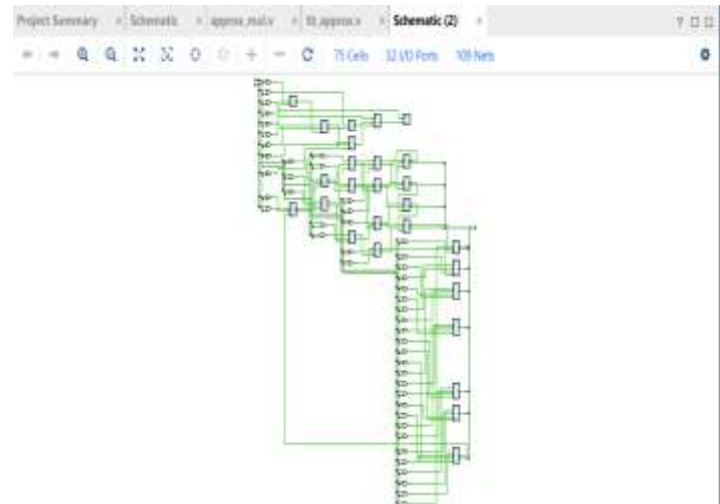


Fig. 7. RTL Schematic of Approximate Multiplier

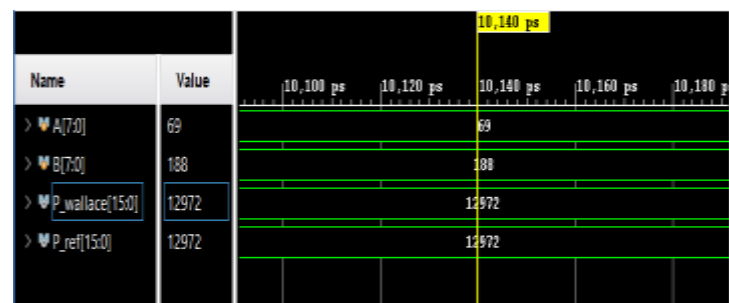


Fig. 8. RTL simulation for Wallace Tree Multiplier



Fig. 9. Power for Approximate Multiplier

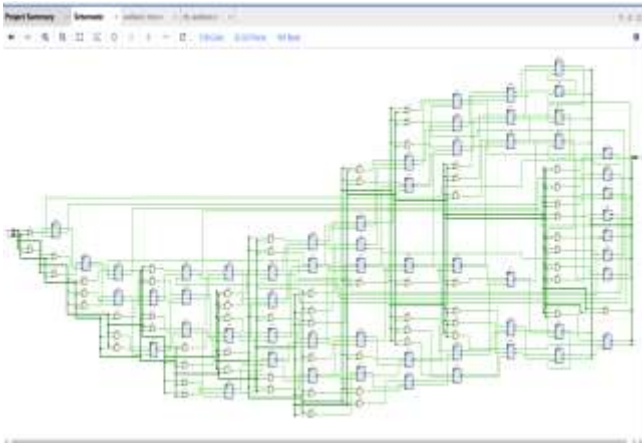


Fig. 10 .RTL Schematic of Wallace Tree Multiplier

TABLE II. COMPARISON BETWEEN APPROXIMATE AND WALLACE TREE MULTIPLIERS

	LUT	Delay (nS)	Power (W)
Approximate multiplier	31	6.620	6.421
Wallace multiplier	92	9.309	14.235

TABLE IV. POWER CALCULATIONS AT DIFFERENT TOGGLE RATES

Toggle rate(%)	Power of approximate multiplier(W)	Delay(ns)
12.5 (default)	6.421	6.620
25	8.38	6.623
50	9.966	6.623
100	11.164	6.623

6. CONCLUSION

The study concludes that energy-efficient approximate multipliers play a vital role in next-generation technologies by enabling low power consumption, reduced hardware complexity, and high speed. Despite minor errors, they are suitable for modern applications like AI, image processing, and IoT, where efficiency is more important than exact accuracy.

7. FUTURE SCOPE

This project includes developing hybrid multiplier architectures that balance accuracy and energy efficiency. Further work can focus on reducing error rates using adaptive or configurable approximation techniques. Implementation on advanced FPGA and ASIC technologies can improve real-time performance. These designs can be extended to applications such as AI, machine learning, image processing, and IoT, where low power and high speed are essential for next-generation systems.

8. REFERENCES

- [1] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-Power Digital Signal Processing Using Approximate Adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, 2013.
- [2] S. Hashemi, R. Bahar, and S. Reda, "DRUM: A Dynamic Range Unbiased Multiplier for Approximate Applications," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2015.
- [3] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed., Prentice Hall, 2003.
- [4] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 1, pp. 14–17, 1964.
- [5] P. Kulkarni, P. Gupta, and M. Ercegovic, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *Proceedings of the International Conference on VLSI Design*, 2011.
- [6] R. Zimmermann and D. Q. Tran, "Optimized Wallace Tree Multiplier Architectures," *IEEE International Conference on Computer Design (ICCD)*, 1998.
- [7] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and Area-Efficient Approximate Wallace Tree Multiplier for Error-Resilient Systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019.
- [8] B. Morasa, Y. B. Anwaraly, J. R. Chennuru, C. Devara, M. Gavvalla, "Smart Monitoring System for Waste Management Using IoT," In: *Advancements in Interdisciplinary Research*. AIR., vol 1738, 2022..
- [9] M. Balaji, N. Padmaja, P. S. A. Gitanjali, Shaik, Siva Kumar and Sai Geetesh, "Design of FIR filter with Fast Adders and Fast Multipliers using RNS Algorithm," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-6, doi: 10.1109/INCET57972.2023.10170321.
- [10] M. Balaji, N. Padmaja, A Low Power Transistor Level FIR Filter Implementation using CMOS 45nm Technology, *International Journal of Nanotechnology*, Vol. 20, No. 1-4, pp. 390-409, 2023.
- [11] M. Balaji, and N. Padmaja. "Area and delay efficient RNS-based FIR filter design using fast multipliers." *Measurement: Sensors*, vol. 31, 101014, 2024.
- [12] M. Sakthimohan and J. Deny, "An Efficient Design of 8 * 8 Wallace Tree Multiplier Using 2 and 3-Bit Adders" *Proceedings of International Conference on Sustainable Expert Systems, Lecture Notes in Networks and Systems* 176, https://doi.org/10.1007/978-981-33-4355-9_3.