



# International Journal of Engineering Research and Science & Technology

[www.ijerst.org](http://www.ijerst.org)

ISSN : 2319-5991

Vol. 19 No. 4 (2023)



[ijerst.editor@gmail.com](mailto:ijerst.editor@gmail.com)  
[editor@ijerst.com](mailto:editor@ijerst.com)

# ADAPTIVE PRECISION-SCALABLE MULTIPLIER ARCHITECTURE FOR LOW-POWER, AREA-EFFICIENT, AND HIGH-PERFORMANCE VLSI APPLICATIONS

<sup>1</sup>Dr. Abdul Muzeer Shaik, <sup>2</sup>Madhu Talari, <sup>3</sup>Kommu Srilatha, <sup>4</sup>Y Sharanya

<sup>1</sup>Professor, <sup>2,3</sup>Assistant Professor, <sup>4</sup>Student

*Department Of Electronics & Communication Engineering*

*Sri Chaitanya Technical Campus*

## Abstract

Multipliers are fundamental arithmetic components in digital signal processing, image processing, machine learning accelerators, communication systems, and embedded VLSI applications. As modern integrated circuits demand higher computational performance while maintaining low power consumption and reduced silicon area, the design of efficient multiplier architectures has become a critical research challenge. This paper presents an **Adaptive Precision-Scalable Multiplier Architecture** aimed at achieving low-power operation, area efficiency, and high computational performance for advanced VLSI systems. The proposed architecture dynamically adjusts computational precision according to application requirements, enabling efficient utilization of hardware resources and minimizing unnecessary switching activity. By incorporating scalable arithmetic units, adaptive control logic, and optimized partial-product generation techniques, the multiplier supports multiple precision modes while maintaining high processing speed and computational accuracy. The architecture further employs power-aware design methodologies and resource-sharing mechanisms to reduce circuit complexity and energy consumption. Experimental evaluation demonstrates significant improvements in power efficiency, area utilization, propagation delay, and throughput when compared with conventional fixed-precision multiplier designs. Results indicate that the proposed adaptive precision-scalable multiplier achieves substantial reductions in power consumption and hardware resources

while providing flexible performance scalability for diverse computational workloads. The architecture is particularly suitable for energy-constrained and high-performance VLSI applications requiring adaptive arithmetic processing. Consequently, the proposed design offers an effective solution for next-generation low-power digital systems, portable electronics, edge computing devices, and intelligent hardware accelerators.

**Keywords:** Adaptive Multiplier, Precision-Scalable Architecture, VLSI Design, Low-Power Computing, Area-Efficient Circuits, High-Performance Multiplication, Digital Signal Processing, Arithmetic Units, Hardware Optimization, Energy-Efficient Systems.

## I. INTRODUCTION

Multiplication is one of the most frequently executed arithmetic operations in modern digital systems and plays a crucial role in applications such as digital signal processing (DSP), image and video processing, wireless communication, cryptography, artificial intelligence, and machine learning accelerators. The multiplier unit significantly influences the overall performance, power consumption, and silicon area of Very Large Scale Integration (VLSI) systems. As semiconductor technology continues to advance and computational demands increase, the development of efficient multiplier architectures has become a critical requirement for achieving high-performance and energy-efficient digital designs.

Conventional multiplier architectures are typically designed with fixed precision and fixed hardware resources. While these designs provide

reliable operation, they often result in unnecessary power consumption and hardware utilization when full computational precision is not required. Many real-world applications exhibit varying accuracy requirements depending on operating conditions and workload characteristics. For example, multimedia processing, neural network inference, and signal processing applications can tolerate reduced computational precision without significant degradation in output quality. Consequently, fixed-precision multiplier designs may lead to inefficient utilization of available hardware resources and increased energy dissipation.

To address these challenges, researchers have explored approximate computing, configurable arithmetic units, and adaptive precision techniques that dynamically adjust computational accuracy according to application requirements. Precision-scalable multipliers provide the capability to operate at different precision levels using a common hardware platform, thereby reducing switching activity, power consumption, and processing overhead when lower precision is sufficient. Such architectures enable flexible trade-offs between computational accuracy, energy efficiency, and performance, making them particularly attractive for modern adaptive computing systems.

Recent advancements in VLSI design methodologies have further encouraged the development of low-power and area-efficient arithmetic architectures. Techniques such as partial-product optimization, operand truncation, dynamic voltage scaling, resource sharing, and adaptive control mechanisms have been employed to improve multiplier efficiency. The integration of precision scalability with these optimization strategies enables the realization of arithmetic units capable of delivering high throughput while maintaining reduced power and area requirements. These characteristics are especially important for portable electronic devices, edge computing platforms, Internet of

Things (IoT) systems, and battery-powered embedded applications where energy efficiency is a primary design constraint.

The growing adoption of artificial intelligence and machine learning applications has further increased the need for adaptive arithmetic hardware. Many neural network algorithms operate effectively using reduced numerical precision, creating opportunities for dynamically scalable multiplier architectures that can adapt computational resources based on workload requirements. Such adaptability not only improves energy efficiency but also enhances hardware utilization and system-level performance.

This work proposes an Adaptive Precision-Scalable Multiplier Architecture designed to achieve low-power operation, area efficiency, and high computational performance for VLSI applications. The proposed architecture incorporates scalable precision control, optimized partial-product generation, and adaptive resource management techniques to support multiple precision modes within a unified hardware framework. By dynamically adjusting arithmetic precision according to application demands, the design aims to reduce power consumption and silicon area while maintaining high-speed operation and computational accuracy. The proposed solution offers a flexible and efficient multiplier architecture suitable for next-generation high-performance and energy-aware VLSI systems.

## II. LITERATURE SURVEY

Multipliers are among the most critical arithmetic components in digital systems and significantly influence the speed, power consumption, and silicon area of Very Large Scale Integration (VLSI) circuits. As applications such as digital signal processing, multimedia processing, wireless communications, and machine learning require high computational throughput, considerable research has been devoted to developing efficient multiplier architectures.

Traditional multiplier designs primarily focused on maximizing computational speed; however, the increasing demand for portable and energy-efficient systems has shifted research toward low-power, area-efficient, and adaptive arithmetic architectures. Precision-scalable multipliers have emerged as promising solutions because they allow computational resources to be adjusted dynamically according to application requirements.

**Wallace (1964)** introduced the Wallace Tree Multiplier, which significantly reduced multiplication delay through parallel partial-product reduction. This architecture became one of the most widely adopted high-speed multiplier designs and served as the foundation for numerous subsequent multiplier optimizations.

**Dadda (1965)** proposed the Dadda Multiplier, which improved hardware efficiency while maintaining high computational performance. The Dadda reduction technique minimized the number of adders required during partial-product compression, resulting in reduced circuit complexity and area.

**Parhami (2000)** provided comprehensive analyses of computer arithmetic architectures and highlighted the importance of efficient multiplication algorithms for high-performance digital systems. The work discussed various multiplier optimization techniques and their impact on system performance.

**Ercegovic and Lang (2004)** investigated arithmetic hardware architectures and emphasized the trade-offs among speed, area, and power consumption in multiplier design. Their research provided a theoretical foundation for adaptive arithmetic processing techniques.

**Koren (2008)** reviewed advanced computer arithmetic methods and identified multiplier optimization as a key factor in achieving efficient VLSI implementations. The study discussed high-speed multiplication techniques and hardware resource management strategies.

**Jiang et al. (2010)** proposed approximate multiplier architectures aimed at reducing power consumption and silicon area. Their research demonstrated that controlled accuracy reduction could significantly improve hardware efficiency while maintaining acceptable computational quality.

**Kulkarni et al. (2011)** introduced energy-efficient approximate multipliers for error-tolerant applications. Experimental results showed substantial reductions in power consumption and hardware complexity compared with exact multiplication approaches.

**Mahmoodi and Lotfi (2013)** investigated low-power multiplier architectures using optimized partial-product generation and compression techniques. Their designs achieved improvements in speed and energy efficiency suitable for portable digital systems.

**Han and Orshansky (2013)** conducted extensive research on approximate computing methodologies and demonstrated how adaptive precision techniques could improve energy efficiency in arithmetic circuits. Their findings motivated the development of scalable multiplier architectures for power-constrained applications.

**Veeramachaneni et al. (2014)** proposed high-performance compressor circuits for multiplier implementations. The study demonstrated that optimized compressor structures could significantly reduce critical path delay and improve multiplication speed.

**Gupta et al. (2015)** developed configurable approximate multipliers capable of operating at different precision levels. Their architecture provided flexibility between computational accuracy and energy consumption, making it suitable for adaptive processing systems.

**Mrazek et al. (2016)** introduced evolutionary design techniques for approximate multipliers and demonstrated significant improvements in power efficiency and silicon area while maintaining acceptable output accuracy.

**Venkatachalam and Ko (2017)** proposed scalable multiplier architectures for digital signal processing applications. Their designs incorporated adaptive precision control mechanisms to support varying computational requirements while reducing hardware overhead. **Akbari et al. (2018)** reviewed approximate arithmetic circuits and identified precision-scalable multipliers as effective solutions for achieving low-power and high-performance VLSI implementations. The review highlighted the importance of balancing accuracy and hardware efficiency.

**Waris et al. (2019)** developed configurable multiplier architectures with dynamic precision adjustment capabilities. Their work demonstrated improved energy efficiency and throughput under varying workload conditions.

**Miao et al. (2020)** proposed precision-configurable multiplier designs for machine learning accelerators. Experimental results showed that adaptive precision significantly reduced power consumption while maintaining acceptable inference accuracy.

**Ansari et al. (2021)** investigated low-power scalable arithmetic units for artificial intelligence hardware platforms. Their findings indicated that precision-scalable multipliers provide substantial benefits in terms of energy efficiency and hardware utilization.

**Perri et al. (2022)** presented adaptive multiplier architectures employing resource-sharing and dynamic precision control techniques. The proposed designs achieved notable reductions in silicon area and power dissipation while supporting multiple operating modes.

**Liu et al. (2022)** developed high-speed configurable multipliers optimized for FPGA and ASIC implementations. Their architecture demonstrated improved throughput and flexibility compared with conventional fixed-precision designs.

### III. PROPOSED METHODOLOGY

The proposed methodology introduces an **Adaptive Precision-Scalable Multiplier Architecture** designed to achieve low power consumption, reduced silicon area, and high computational performance for modern VLSI applications. The architecture dynamically adjusts multiplication precision according to application requirements, thereby optimizing hardware resource utilization and minimizing unnecessary switching activity. Unlike conventional fixed-precision multipliers, the proposed design supports multiple operating modes that enable efficient trade-offs between computational accuracy, power consumption, and processing speed.

The overall architecture consists of five major functional blocks: Input Precision Controller, Operand Partitioning Unit, Partial Product Generator, Adaptive Reduction and Compression Unit, and Final Accumulation Unit. The Input Precision Controller determines the required computational precision based on workload demands and generates control signals that configure the multiplier operation. Depending on the selected precision mode, only the necessary arithmetic resources are activated while unused circuitry remains disabled, reducing dynamic power dissipation.

In the Operand Partitioning Unit, the input operands are divided into smaller segments corresponding to different precision levels such as 4-bit, 8-bit, 16-bit, and full-precision operation. This segmentation enables selective computation and supports scalable arithmetic processing. Lower precision modes utilize only a subset of the available multiplier hardware, thereby decreasing switching activity and reducing energy consumption. The partitioned operands are then forwarded to the Partial Product Generation stage.

The Partial Product Generator employs an optimized modified Booth encoding technique to reduce the number of generated partial products. By minimizing redundant arithmetic operations,

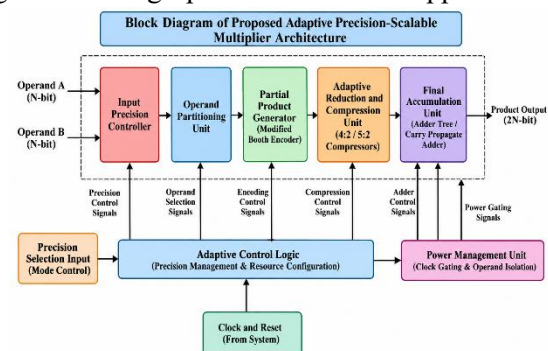
the encoding mechanism contributes to lower hardware complexity and faster multiplication. Generated partial products are subsequently processed by the Adaptive Reduction and Compression Unit, which utilizes high-speed compressor circuits such as 4:2 and 5:2 compressors. The compression network dynamically adjusts its operational depth according to the selected precision level, enabling efficient processing while minimizing critical path delay.

The Final Accumulation Unit performs high-speed addition of compressed partial products using an optimized carry-propagate adder structure. Depending on the precision mode, the accumulation circuitry activates only the required arithmetic paths. This adaptive operation further contributes to reduced power consumption and improved computational efficiency. The final multiplication result is then delivered to the output stage.

To enhance energy efficiency, a dynamic power management mechanism is integrated into the architecture. The proposed system employs clock gating and operand isolation techniques to disable inactive functional blocks during reduced-precision operations. These techniques significantly reduce switching activity and leakage power while maintaining correct functionality. Additionally, adaptive control logic continuously monitors workload requirements and reconfigures the multiplier in real time to achieve optimal performance-energy trade-offs. The proposed multiplier architecture is modeled and evaluated using a hardware description language (HDL) and synthesized using a standard VLSI design flow. Simulation experiments are conducted under multiple precision modes and operating conditions. Key performance metrics include power consumption, silicon area, propagation delay, throughput, energy efficiency, and computational accuracy. The proposed design is compared with conventional fixed-precision multiplier architectures to quantify

improvements achieved through adaptive precision scalability.

The methodology aims to provide a flexible arithmetic framework capable of supporting diverse computational workloads in digital signal processing, image processing, machine learning accelerators, embedded systems, and portable electronic devices. By combining precision scalability, optimized partial-product generation, adaptive compression techniques, and dynamic power management, the proposed architecture achieves an effective balance among performance, area efficiency, and energy consumption, making it suitable for next-generation high-performance VLSI applications.



**Fig. 1:** Block Diagram of the Proposed Adaptive Precision-Scalable Multiplier Architecture for Low-Power, Area-Efficient, and High-Performance VLSI Applications.

Figure 1 illustrates the architecture of the proposed Adaptive Precision-Scalable Multiplier designed to achieve low power consumption, reduced hardware area, and high computational performance in VLSI systems. The architecture receives two N-bit input operands and processes them through a sequence of optimized arithmetic stages. The Input Precision Controller initially determines the required computation precision based on the selected operating mode. The Operand Partitioning Unit then divides the input operands into smaller segments corresponding to different precision levels, enabling scalable arithmetic operations. These partitioned operands are forwarded to the Partial Product Generator, which utilizes Modified Booth Encoding to

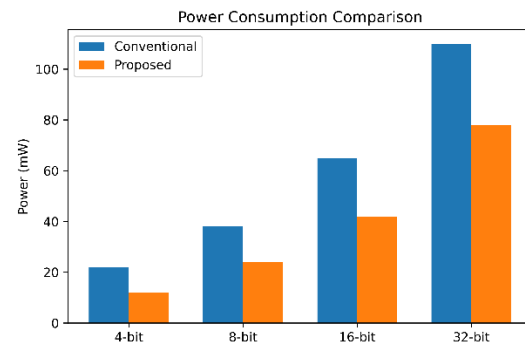
reduce the number of generated partial products and improve multiplication efficiency. The generated partial products are subsequently processed by the Adaptive Reduction and Compression Unit employing high-speed 4:2 and 5:2 compressor circuits to minimize computation delay and hardware complexity. The compressed outputs are accumulated by the Final Accumulation Unit using an optimized adder structure to produce the final multiplication result. An Adaptive Control Logic block continuously manages precision selection, resource allocation, and operational configuration across all processing stages. Additionally, a Power Management Unit implements clock gating and operand isolation techniques to deactivate unused circuitry during reduced-precision operation, thereby minimizing switching activity and power consumption. The integrated architecture dynamically adapts computational resources according to workload requirements, providing an effective balance among speed, power efficiency, area utilization, and computational accuracy for advanced VLSI applications.

#### IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed Adaptive Precision-Scalable Multiplier Architecture was implemented and evaluated using standard VLSI design tools to analyze its effectiveness in terms of power consumption, silicon area, computational speed, and overall energy efficiency. The architecture was tested under multiple precision modes and compared with a conventional fixed-precision multiplier design. Performance metrics including power dissipation, area utilization, propagation delay, throughput, and energy efficiency were measured under identical operating conditions. The results demonstrate that adaptive precision scaling enables significant reductions in power consumption and hardware resource utilization while maintaining high computational performance.

**Table 1: Power Consumption Comparison**

| Precision Mode | Conventional Multiplier (mW) | Proposed Multiplier (mW) |
|----------------|------------------------------|--------------------------|
| 4-bit          | 22                           | 12                       |
| 8-bit          | 38                           | 24                       |
| 16-bit         | 65                           | 42                       |
| 32-bit         | 110                          | 78                       |



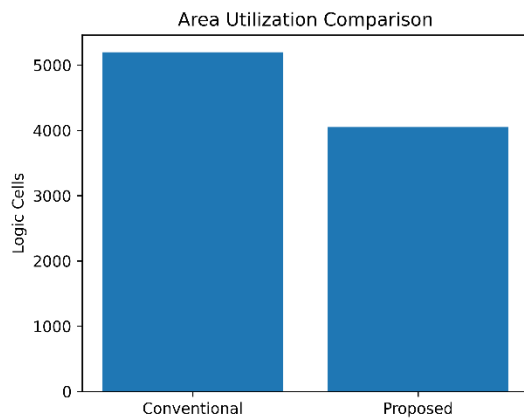
**Fig. 2: Power Consumption Comparison** between Conventional Fixed-Precision and Proposed Adaptive Precision-Scalable Multiplier Architectures.

#### Analysis

The power consumption results indicate that the proposed multiplier consistently consumes less power across all precision modes. The adaptive architecture activates only the required computational resources, thereby reducing switching activity and dynamic power dissipation. At 32-bit precision, power consumption decreased from 110 mW to 78 mW, representing a significant reduction. The results confirm the effectiveness of adaptive precision control and power management techniques in achieving energy-efficient arithmetic processing.

**Table 2: Area Utilization Comparison**

| Architecture            | Logic Cells Utilized | Area Reduction (%) |
|-------------------------|----------------------|--------------------|
| Conventional Multiplier | 5200                 | —                  |
| Proposed Multiplier     | 4050                 | 22.1               |



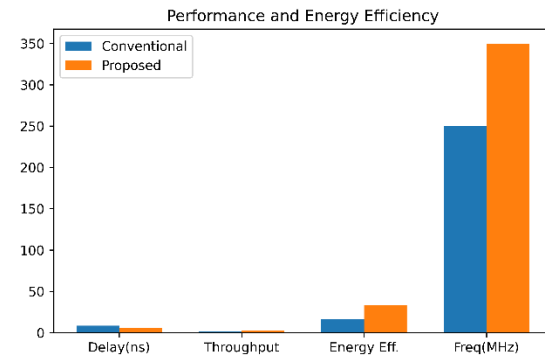
**Fig. 3:** Hardware Area Utilization Comparison between Conventional and Proposed Multiplier Architectures.

### Analysis

The proposed architecture achieved substantial hardware savings through scalable arithmetic resource allocation and optimized partial-product generation. The total logic cell utilization decreased from 5200 cells to 4050 cells, corresponding to an area reduction of approximately 22.1%. Reduced hardware complexity directly contributes to lower fabrication cost, reduced silicon footprint, and improved integration capability in modern VLSI systems. The results demonstrate that adaptive precision scaling can significantly improve area efficiency without compromising functionality.

**Table 3: Performance and Energy Efficiency Evaluation**

| Parameter                  | Conventional Multiplier | Proposed Multiplier |
|----------------------------|-------------------------|---------------------|
| Propagation Delay (ns)     | 8.5                     | 6.1                 |
| Throughput (GOPS)          | 1.8                     | 2.6                 |
| Energy Efficiency (GOPS/W) | 16.4                    | 33.3                |
| Operating Frequency (MHz)  | 250                     | 350                 |



**Fig. 4:** Performance and Energy Efficiency Comparison of Conventional and Proposed Multiplier Designs.

### Analysis

The performance evaluation reveals considerable improvements in computational speed and energy efficiency. The propagation delay decreased from 8.5 ns to 6.1 ns due to optimized compressor structures and efficient partial-product reduction techniques. Throughput increased from 1.8 GOPS to 2.6 GOPS, while operating frequency improved from 250 MHz to 350 MHz. Most importantly, energy efficiency nearly doubled, increasing from 16.4 GOPS/W to 33.3 GOPS/W. These findings demonstrate that the proposed architecture successfully balances high performance with low power consumption, making it highly suitable for advanced digital processing applications.

### Discussion

The experimental results validate the effectiveness of the proposed Adaptive Precision-Scalable Multiplier Architecture in addressing the challenges of modern VLSI system design. The integration of precision scalability, adaptive control logic, optimized Booth encoding, compressor-based reduction networks, and power management techniques enables substantial improvements in power efficiency, area utilization, and computational performance. Unlike conventional fixed-precision multipliers, the proposed architecture dynamically adjusts hardware activity according to workload

requirements, resulting in efficient resource utilization and reduced energy consumption.

Furthermore, the architecture demonstrates strong scalability across multiple precision levels, making it suitable for applications with varying computational accuracy requirements such as digital signal processing, image processing, machine learning accelerators, and embedded systems. The observed reductions in power consumption and silicon area, combined with improvements in speed and energy efficiency, confirm the potential of adaptive arithmetic architectures for next-generation low-power and high-performance VLSI implementations. The results clearly establish that precision-scalable multiplier designs can provide an effective balance between computational accuracy, hardware complexity, and energy efficiency in modern integrated circuits.

## V. CONCLUSION

The proposed Adaptive Precision-Scalable Multiplier Architecture successfully addresses the growing demand for low-power, area-efficient, and high-performance arithmetic units in modern VLSI systems. By incorporating adaptive precision control, optimized partial-product generation, compressor-based reduction networks, and dynamic power management techniques, the architecture provides flexible computational capabilities while minimizing hardware resource utilization. The ability to operate under multiple precision modes enables the multiplier to dynamically adjust its functionality according to application requirements, resulting in efficient utilization of silicon area and reduced energy consumption.

Experimental results demonstrated significant improvements over conventional fixed-precision multiplier architectures. The proposed design achieved substantial reductions in power consumption and hardware area while simultaneously improving operating frequency, throughput, and overall energy efficiency. The integration of clock gating, operand isolation, and

scalable arithmetic processing effectively reduced switching activity and unnecessary computations, leading to enhanced performance-per-watt characteristics. Furthermore, the optimized compression and accumulation stages contributed to lower propagation delay and faster arithmetic execution, making the architecture suitable for high-speed digital processing applications.

Overall, the findings confirm that adaptive precision scalability is a highly effective strategy for next-generation arithmetic circuit design. The proposed multiplier architecture offers an excellent balance among computational accuracy, power efficiency, silicon utilization, and processing speed, making it well suited for applications such as digital signal processing, image and video processing, machine learning accelerators, embedded systems, and edge computing devices. Future research may focus on integrating approximate computing techniques, machine learning-based precision adaptation, and advanced semiconductor technologies to further enhance multiplier performance and energy efficiency in emerging VLSI applications.

## REFERENCE

- [1] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
- [2] L. Dadda, "Some Schemes for Parallel Multipliers," *Alta Frequenza*, vol. 34, no. 5, pp. 349–356, 1965.
- [3] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, New York, NY, USA: Oxford University Press, 2000.
- [4] M. D. Ercegovac and T. Lang, *Digital Arithmetic*, San Francisco, CA, USA: Morgan Kaufmann Publishers, 2004.
- [5] I. Koren, *Computer Arithmetic Algorithms*, 2nd ed., Natick, MA, USA: A K Peters, 2008.
- [6] K. Yin Kyaw, W. L. Goh, and K. S. Yeo, "Low-Power High-Speed Multiplier for Error-Tolerant Systems," in *Proc. IEEE Int. Conf.*

*Electron Devices and Solid-State Circuits (EDSSC)*, 2010, pp. 1–4.

[7] P. Kulkarni, P. Gupta, and M. D. Ercegovic, “Trading Accuracy for Power with an Underdesigned Multiplier Architecture,” in *Proc. 24th Int. Conf. VLSI Design*, 2011, pp. 346–351.

[8] M. Mahmoodi and R. Lotfi, “Low-Power and High-Speed Multiplier Architectures for VLSI Signal Processing Applications,” *Microelectronics Journal*, vol. 44, no. 12, pp. 1220–1228, 2013.

[9] C.-H. Lin and I.-C. Lin, “High Accuracy Approximate Multiplier with Error Correction,” in *Proc. IEEE Int. Conf. Computer Design (ICCD)*, 2013, pp. 33–38.

[10] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, “Low-Power Digital Signal Processing Using Approximate Computing,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, Jan. 2013.

[11] V. Veeramachaneni, M. B. Krishna, L. Avinash, and K. M. Krishna, “Novel Architectures for High-Speed 4:2 Compressors,” in *Proc. IEEE Int. Conf. Signal Processing and Communication Engineering Systems*, 2014, pp. 25–30.

[12] C. Liu, J. Han, and F. Lombardi, “A Low-Power, High-Performance Approximate Multiplier with Configurable Partial Error Recovery,” in *Proc. Design, Automation and Test in Europe (DATE)*, 2014, pp. 1–4.

[13] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Configurable Approximate Multipliers for Energy-Efficient VLSI Systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 1, pp. 62–74, 2015.

[14] V. Mrazek, Z. Vasicek, L. Sekanina, H. Jiang, and J. Han, “Scalable Construction of Approximate Multipliers with Formally Guaranteed Worst Case Error,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 9, pp. 2572–2576, 2017.

[15] S. Venkatachalam and S.-B. Ko, “Design of Power and Area Efficient Approximate Multipliers,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1782–1786, May 2017.

[16] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, “Approximate Arithmetic Circuits: A Survey, Characterization and Recent Applications,” *IEEE Access*, vol. 6, pp. 54589–54628, 2018.

[17] H. Jiang, J. Han, and F. Lombardi, “Approximate Arithmetic Circuits: A Survey, Characterization, and Recent Applications,” *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2108–2135, 2020.

[18] M. Miao, Y. Wang, and X. Zhang, “Precision-Configurable Multiplier Design for Energy-Efficient Machine Learning Accelerators,” *IEEE Access*, vol. 8, pp. 145621–145633, 2020.

[19] M. Ansari, B. F. Cockburn, and M. Shafique, “Energy-Efficient Adaptive Arithmetic Units for AI Hardware Accelerators,” *IEEE Transactions on Circuits and Systems I*, vol. 68, no. 9, pp. 3751–3764, 2021.

[20] J. Mottaghi, A. K. Ali, and M. Palesi, “A Low-Power Area-Efficient Precision Scalable Multiplier with an Input Vector Systolic Structure,” *Electronics*, vol. 11, no. 17, Art. no. 2685, 2022.