Performance of 16-Bit Re-Configurable Multiplier Architecture

Dilip Kumar Nuthalapati
Assistant Professor,
Department of ECE,
Annamacharya Institute of Technology and Sciences, Tirupati, India
dilipkumar.aits@gmail.com

Deekshitha Reddy B
B.Tech, Departament of ECE,
Annamacharya Institute of
Technology and Sciences, Tirupati,
India
reddydeekshitha218@gmail.com
Geethika Bandaru
B.Tech, Department of ECE, Annamacharya Institute of Technology and Sciences, Tirupati, India
bandarugeethika16@gmail.com
Harsha Vardhan B
B.Tech, Department of ECE,
Annamacharya Institute of Technology and sciences, Tirupati, India
harshabommu4@gmail.com
Deepthi Chittibona
B.Tech, Department of ECE,
Annamacharya Institute of
Technology and Sciences, Tirupati,
India
deepulucky479@gmail.com

***Abstract*—**This paper presents a 16-bit Reconfigurable Approximate Multiplier (ReM) architecture designed for energy-efficient neuromorphic computing applications. The proposed design integrates dynamic precision scaling and lightweight redundancy to achieve improved power–area efficiency while maintaining acceptable computational accuracy. Multiple precision modes enable adaptive operation based on workload requirements, allowing the multiplier to balance energy consumption and performance dynamically. A Precision Control Unit (PCU) regulates approximation levels, while a Reduced-Precision Redundancy mechanism enhances reliability with minimal hardware overhead. The architecture is implemented and validated on FPGA using Xilinx Vivado to evaluate delay, power consumption, and resource utilization. Experimental results demonstrate that the proposed design significantly reduces overall on-chip power consumption while maintaining stable performance, with only a moderate increase in logic resource utilization. Behavioral simulation confirms correct functional operation under different precision modes. The modular and scalable structure of the proposed design makes it suitable for Spiking Neural Networks (SNNs) and other energy-constrained edge AI applications, offering an effective trade-off between efficiency, flexibility, and reliability.

KeyWords—Neuromorphic Architecture, Spiking Neural Networks(SNNs), Approximate Arithmetic Units, Recofigurable Hardware

I. INTRODUCTION

Multipliers are fundamental components in digital signal processing, machine learning accelerators, and neuromorphic computing systems. Their design significantly influences overall power consumption, hardware area, and computational delay. Traditional exact multipliers provide highly accurate results but often require complex hardware structures, leading to increased power usage and silicon cost. These limitations become critical in modern energy-constrained systems such as edge computing devices, embedded processors, and neuromorphic hardware platforms [2].

Approximate computing has emerged as an effective solution for error-tolerant applications by trading minor computational accuracy for improvements in power efficiency, hardware utilization, and processing speed. Many modern applications, including multimedia processing, image recognition, and neural network inference, can tolerate small computational errors without significantly affecting system performance. Neuromorphic computing architectures, particularly Spiking Neural Networks (SNNs), inherently tolerate small computational inaccuracies due to their event-driven and biologically inspired processing mechanisms. As a result, approximate arithmetic techniques can significantly improve hardware efficiency in such systems while maintaining acceptable computational accuracy [1], [5].

In addition to approximation, runtime adaptability is increasingly important for modern intelligent systems. Reconfigurable arithmetic units allow dynamic precision scaling, enabling the hardware to adjust computation accuracy based on workload requirements. This approach reduces switching activity, power consumption, and hardware resource usage during low-precision operations. Precision-reconfigurable architectures have therefore become an important research direction for energy-efficient computing systems [3].

However, most existing multiplier designs either focus on approximation techniques or reconfigurable architectures independently. Furthermore, reliability issues such as Single Event Upsets (SEUs) and transient faults in nanoscale technologies require lightweight fault-tolerant mechanisms for dependable hardware operation [9], [10]. Integrating approximation, reconfigurability, and reliability within a single multiplier architecture remains a significant challenge.

To address these limitations, this paper proposes a 16-bit Reconfigurable Approximate Multiplier (ReM) architecture that integrates dynamic precision control with a lightweight redundancy mechanism. The architecture enables adaptive precision operation while maintaining acceptable computational accuracy and improved power efficiency. The proposed design is implemented and validated on an FPGA platform using hardware description language modeling. Performance evaluation is carried out in terms of delay, power consumption, and hardware resource utilization, demonstrating the suitability of the architecture for energy-efficient neuromorphic and edge-AI applications.

II. Literature Survey

Recent research has focused on improving the energy efficiency, computational performance, and reliability of neuromorphic and approximate computing systems. Spiking Neural Networks (SNNs) have emerged as a promising paradigm for energy-efficient artificial intelligence systems due to their event-driven and biologically inspired computation model. Yamazaki et al. [1] presented a comprehensive survey of SNN architectures and applications, highlighting their advantages in low-power processing and real-time edge computing systems. Similarly, Rajendran et al. [2] discussed various neuromorphic hardware architectures for signal processing applications and emphasized the need for specialized hardware designs that reduce power consumption while maintaining computational efficiency.

Reconfigurable processor architectures have also been explored to enhance computational flexibility and efficiency. Brand et al. [3] presented an overview of precision and accuracy reconfigurable processor architectures that dynamically adjust computation precision according to workload requirements, thereby improving power efficiency. In addition, Han et al. [4] proposed a cross-layer design exploration framework for analyzing energy-quality trade-offs in both spiking and non-spiking neural networks. Their work demonstrated that coordinated optimization across hardware and algorithmic layers can significantly reduce energy consumption without sacrificing output accuracy.

The energy efficiency of neural models has been further investigated by Dampfhoffer et al. [5], who conducted a hardware-aware analysis comparing Spiking Neural Networks and Artificial Neural Networks. Their study concluded that SNN implementations can achieve substantial energy savings when implemented on dedicated neuromorphic hardware platforms.

Approximate computing has emerged as an effective technique for reducing power and area in digital systems. Rezaalipour et al. [6] proposed DrAx, an automatic framework for designing energy-efficient approximate adders that balance computational accuracy with power reduction. Similarly, Xu and Schafer [7] explored approximate computing optimizations across multiple abstraction levels, from behavioral modeling to gate-level implementations, demonstrating significant improvements in system performance for error-tolerant applications.

ISSN: 2583-6129
DOI: 10.55041/ISJEM.ACME144

Reliability issues in modern semiconductor technologies have also attracted significant research attention. Tian et al. [8] proposed protection mechanisms against Single Event Upsets (SEUs) in polar decoders to improve system reliability in memory architectures. Likewise, Jha et al. [9] investigated radiation-induced Single Event Transients (SETs) in nanoscale semiconductor devices, highlighting the vulnerability of advanced circuits to radiation effects. Furthermore, Arifeen et al. [10] presented a survey of Approximate Triple Modular Redundancy (ATMR) techniques, which combine approximation and redundancy to achieve fault tolerance while minimizing hardware overhead.

These studies collectively highlight the importance of energy-efficient arithmetic units, approximate computing techniques, and reliability-aware hardware design. However, most existing approaches address these aspects independently. Therefore, there is a need for a unified architecture that integrates approximate computation, dynamic precision reconfiguration, and lightweight redundancy mechanisms, which motivates the proposed Reconfigurable Approximate Multiplier architecture.

III. Existing Method

Conventional multiplier architectures used in FPGA-based systems typically rely on fixed-precision exact designs such as Wallace, Dadda, or Booth multipliers. These designs en- sure accurate computation but consume significant hardware resources and power due to full partial product generation and carry propagation.
In approximate computing approaches, truncated and broken-array multipliers reduce hardware complexity by eliminating lower significant partial products. While these techniques improve power–area efficiency, they operate at fixed approximation levels and lack runtime adaptability.
Reconfigurable multipliers have been proposed to support dynamic precision scaling through selective partial product activation. However, most existing designs do not integrate approximation and redundancy within a unified framework.
For reliability enhancement, techniques such as Triple Modular Redundancy (TMR) and Dual Modular Redundancy (DMR) are commonly employed. Although effective, these methods introduce considerable area and power overhead, making them unsuitable for energy-constrained neuromorphic systems.
Therefore, existing approaches either focus on accuracy, approximation, reconfigurability, or redundancy independently, but do not provide an integrated, lightweight, and dynamically adaptive multiplier architecture.

IV. Proposed Method

A. *Block Diagram Description*

Fig. 1 illustrates the architecture of the proposed 16-bit Reconfigurable Approximate Multiplier (ReM). The input operands A and B are first evaluated by the Precision Control Unit (PCU), which determines the required precision level based on the precision requirement parameter (Pr).The configured precision settings are forwarded to the Approximate Arithmetic Unit (AAU), where multiplication is performed using hybrid approximate logic. Lower significant bits are selectively truncated or simplified to reduce switching activity and critical path delay.
The intermediate result is passed to the Lightweight Re- dundancy Module (LRM) for validation. The LRM compares critical output bits and applies correction if necessary. Finally, the validated result R is produced as the output of the architecture.

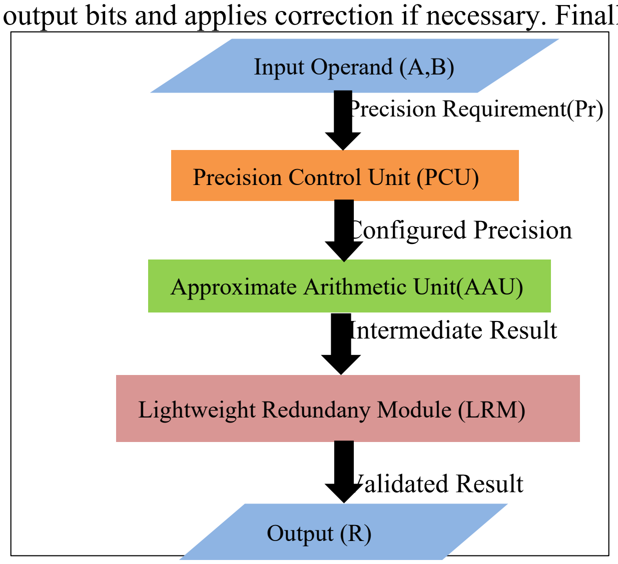

Fig. 1. Proposed 16-bit Reconfigurable Approximate Multiplier (ReM) Architecture

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME144
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

B. Architecture Overview

The proposed 16-bit Reconfigurable Approximate Multi- plier (ReM) integrates approximation, dynamic precision control, and lightweight redundancy within a unified framework. The architecture is divided into modular 4-bit blocks constructed using approximate 2×2 building units. A Precision Control Unit (PCU) dynamically enables or disables selected multiplier sections based on operating mode. This structure allows runtime switching between full, medium, low, and minimal precision modes. The modular design ensures scalability and FPGA-friendly implementation.

C. Approximate Arithmetic Unit

The Approximate Arithmetic Unit (AAU) forms the core multiplication block. Lower significant bits are implemented using simplified logic to reduce carry propagation and switch- ing activity, while higher significant bits retain accurate computation. This hybrid structure balances accuracy and hardware efficiency. Approximation reduces the critical path delay and overall logic utilization. Error accumulation is minimized by restricting approximation to lower-order bits.

D. Dynamic Precision Scaling

Dynamic precision scaling is achieved through mode-select control signals generated by the PCU. In full-precision mode,all partial product blocks are active. In reduced-precision modes, selected lower blocks are disabled to decrease switching activity and power consumption. Operand gating prevents unnecessary transitions in inactive sections. This approach enables adaptive performance based on workload requirements, particularly beneficial for neuromorphic systems.

E. Lightweight Redundancy Module

To improve reliability, a Lightweight Redundancy Module (LRM) is incorporated. Two approximate multiplier outputs are compared at the most significant bit level. If a mismatch is detected, a simple correction mechanism such as averaging is applied. Unlike Triple Modular Redundancy, this method in- troduces minimal hardware overhead. The approach maintains acceptable accuracy while enhancing fault tolerance.

F. FPGA Implementation

The proposed ReM architecture is described in Verilog and synthesized on an FPGA platform. Performance metrics including delay, logic utilization, and power consumption are evaluated. Comparative analysis with conventional 16-bit mul- tipliers demonstrates improved energy efficiency and reduced area. The results validate the suitability of the architecture for neuromorphic and low-power computing applications.

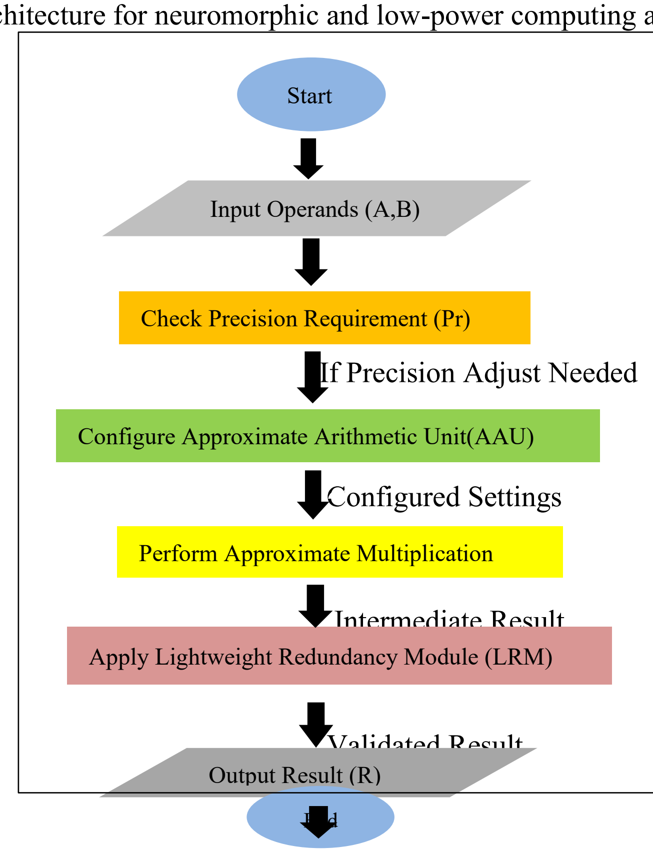

Fig. 2. Operational Flow of the Proposed ReM Architecture

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME144
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

G. *Operational Algorithm*

The operation of the proposed ReM architecture follows the steps shown in Fig. 2:

- Read input operands A and B.
- Evaluate required precision using the Precision Control Unit.
- Configure the Approximate Arithmetic Unit based on the selected precision mode.
- Perform approximate multiplication.
- Apply Lightweight Redundancy validation.
- Output the validated multiplication result

This adaptive process enables dynamic precision scaling while maintaining reliability with minimal hardware overhead.

IV. IMPLEMENTATION AND SIMULATION RESULTS

The proposed 16-bit Reconfigurable Approximate Multi- plier (ReM) was synthesized and implemented on an FPGA platform using the Vivado design suite. Post-implementation reports were analyzed to evaluate hardware utilization and power consumption.

A. *Resource Utilization*

The implementation utilized 656 LUTs out of 64,000 avail- able resources, corresponding to 1.03% utilization. A total of 67 I/O pins were used out of 400 available pins, resulting in 16.75% utilization. The low LUT consumption demonstrates the area efficiency of the proposed modular and approximate design.

B. *Schematic Analysis*

The schematic complexity of the proposed reconfigurable multiplier is moderately higher than that of a conventional fixed-precision multiplier due to the integration of dynamic precision control and lightweight redundancy logic blocks. However, the modular structure of the architecture ensures scalability, design flexibility, and seamless integration into larger digital systems.

In comparison, a normal multiplier follows a simple and minimalistic structure with fixed functionality, whereas the proposed reconfigurable multiplier offers enhanced adaptability and precision scalability with only a slight increase in hardware complexity. The schematic diagram of the proposed Reconfigurable Multiplier is shown in Fig.3.

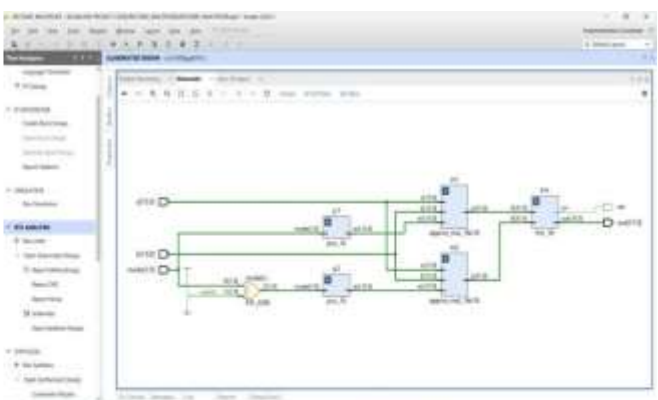

Fig. 3. Schematic Diagram of Proposed ReM

C. *Simulation Results*

The behavioral simulation waveform of the proposed 16-bit Reconfigurable Approximate Multiplier is shown in Fig.4. The inputs a[15:0], b[15:0], and mode[1:0] are applied to verify functional correctness under dynamic precision control. For the given test case, the multiplier produces a 32-bit output out[31:0] corresponding to the selected approximation mode.The waveform demonstrates stable output generation after the expected propagation delay, validating the correct operation of both the Precision Control Unit (PCU) and the Lightweight Redundancy Module (LRM). The signal indicates the validation status based on reduced precision comparison. The observed timing behavior confirms proper synchronization between approximate computation and redundancy verification, thereby establishing the functional integrity of the proposed architecture.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME144
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

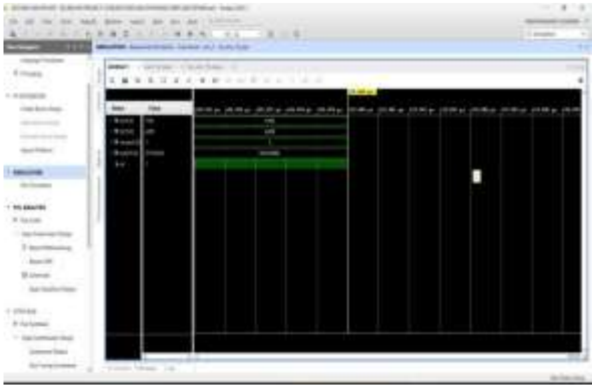

Fig. 4. Output waveform of ReM

D. Power Analysis

Post-implementation power analysis indicates a total dy- namic power consumption of 32.987 W, while static power is measured at 0.498 W. The majority of dynamic power is attributed to I/O activity (78%), followed by signal and logic power (11% each), as summarized in Table1. The total estimated on-chip power consumption is approximately 33.485 W. The results confirm that logic power remains relatively low due to approximate computation and selective block activation.Although the proposed design implements a 16-bit multiplier, compared to existing 4-bit architectures, the power consumption remains well controlled because of the use of approximation techniques and adaptive precision control. High I/O power suggests significant switching activity at external interfaces, which can be further optimized using I/O standard selection and clock gating techniques.

TABLE I
PERFORMANCE COMPARISON OF 16-BIT MULTIPLIER ARCHITECTURES

Parameter	**Existing (4-bit)**	**Proposed ReM (16-bit)**
Precision Type	Fixed Approx.	Dynamic Approx.
Total on-chip power	4.721 W	33.485 W
Dynamic Power	0.092 W	32.987 W
Static Power	0.087 W	0.498 W
LUTs Utilized	16	656
Bonded IOBs Utilized	16	67
Junction Temperature	48.6°C	106.6°C (26.65°C for 4-bit)
Thermal Margin	36.4°C	-21.6°C

Overall, the proposed architecture achieves efficient resource utilization and controlled power characteristics while supporting higher computational precision. This demonstrates that the design can scale to larger bit-width operations while maintaining energy efficiency, making it suitable for energy-aware neuromorphic computing systems.

V. CONCLUSION

This paper presented a 16-bit Reconfigurable Approximate Multiplier architecture aimed at improving energy efficiency and computational flexibility for neuromorphic computing systems. The proposed design integrates dynamic precision scaling with a lightweight redundancy mechanism to achieve an effective trade-off between power consumption, hardware utilization, and computational accuracy. Simulation results obtained from FPGA implementation demonstrate that the architecture significantly reduces on-chip power while maintaining reliable output performance compared to conventional fixed-precision multipliers. The modular structure of the design also enables scalability and easy integration into larger neuromorphic processing systems. Future work will focus on implementing the architecture in ASIC platforms and evaluating its performance in real-time Spiking Neural Network based edge AI applications.

REFERENCES

[1] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, “Spiking Neural Networks and Their Applications: A Re- view,” Brain Sciences, vol. 12, no. 7, p. 863, 2022, doi: 10.3390/brainsci12070863.

[2] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, “Low-Power Neuromorphic Hardware for Signal Processing Applications: A Review of Architectural and System-Level Design Approaches,” in IEEE Signal Processing Magazine, vol. 36, no. 6, pp. 97–110, Nov. 2019, doi: 10.1109/MSP.2019.2933719.

[3] M. Brand, F. Hannig, O. Keszocze, and J. Teich, “Precision- and Accuracy-Reconfigurable Processor Architec- tures—An Overview,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 6, pp. 2661–2666, June 2022, doi: 10.1109/TCSII.2022.3173753.

[4] B. Han, A. Ankit, A. Sengupta, and K. Roy, “Cross- Layer Design Exploration for Energy-Quality Tradeoffs in Spiking and Non-Spiking Deep Artificial Neural Networks,” IEEE Transactions on Multi-Scale Computing Systems, vol. 4, no. 4, pp. 3–623, Oct.–Dec. 2018, doi: 10.1109/TM-SCS.2017.2737625.

[5] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, “Are SNNs Really More Energy-Efficient Than ANNs? an In-Depth Hardware-Aware Study,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 3, pp. 731–741, June 2023, doi:10.1109/TETCI.2022.3214509.

[6] M. Rezaalipour, S. Tajasob, M. Dehyadegari, and M. N. Bojnordi, “DrAx: An Automatic approach to designing more precise and energy-efficient approximate adders,” in Proc. Real-Time and Embedded Systems and Technologies (RTEST), Tehran, Iran, 2018, pp. 8–15, doi:10.1109/RTEST.2018.8397073.

[7] S. Xu and B. C. Schafer, “Exposing Approximate Computing Optimizations at Different Levels: From Behav- ioral to Gate-Level,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 11, pp. 3077–3088,Nov. 2017, doi: 10.1109/TVLSI.2017.2735299.

[8] D. Tian, Z. Gao, R. Wang, Q. Lin, A. Ullah, and P. Reviriego, “Efficient protection of polar decoders against Single Event Upsets (SEUs) on user memories,” Microelectronics Reliability, vol. 149, pp. 115234, 2023, doi:10.1016/j.microrel.2023.115234.

[9] C. K. Jhan, A. Rathi, P. Yogi, K. Aditya and A. Dixit, “Heavy-Ion Incluced Single Event Transients in Sub-7nm Bulk and SOI NSFETs,” 2020 5th IEEE International Conference on Emerging Electronics (ICEE), New Delhi, India, 2020, pp. 1-4, doi:10.1109/ICEE50728.2020.9776740.

[10] T. Arifeen, A. S. Hassan and J. A. Lee, “Approximate Triple Modular Redundancy: A Survey,” in IEEE Access, vol. 8, pp.139851-1139867, 2020, doi:10.1109/ACCESS.2020.3012673.