iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** VII **Month of publication:** July 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.84283

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue VII July 2026- Available at www.ijraset.com*

# An Efficient Approximate Compressor Architecture for Balanced Error Accumulation in MAC Units

Tirupathi Yaswanth[1], Dr. Suresh Babu Potladurty[2]
[1]*PG Scholar, Dept. of ECE, Sri Venkateshwara College of Engineering, Tirupathi, India*
[2]*Associate Professor, Dept. of ECE, Sri Venkateshwara College of Engineering, Tirupathi, India*

***Abstract:*** ***Designing energy-efficient and high-performance arithmetic hardware remains a major challenge for modern computing systems, particularly for the multipliers and compressors that dominate multiply–accumulate (MAC) workloads. This paper presents an efficient approximate compressor architecture designed to achieve balanced error accumulation in MAC units used for error-tolerant applications such as image processing and machine learning (ML). The proposed architecture selectively simplifies the compressor stages that reduce partial products of least numerical significance, while preserving exact computation for the bits that dominate accuracy, so that approximation error accumulates in a controlled and balanced manner rather than growing unpredictably across operand widths. Building on this compressor, a configurable multiply–accumulate (MAC) unit for ML hardware is developed, capable of switching between an exact mode and 2-bit and 4-bit approximate compression modes at runtime according to instantaneous power and accuracy requirements. A modified, error-compensated activation function is further proposed to offset the systematic bias introduced by the approximate compressor at the neuron output. The proposed architecture is implemented in SystemVerilog and synthesized with Synopsys Design Compiler for UMC 90 nm CMOS technology, with functional validation on FPGA platforms. A Gaussian blur filter is used to evaluate the compressor in an image-processing pipeline, while a multilayer perceptron (MLP) trained on standard ML datasets evaluates the configurable MAC unit and the compensated activation function. Experimental results show that the proposed approximate compressor architecture achieves substantial reductions in power, area, and energy per operation relative to an exact Wallace-tree baseline, while keeping error metrics such as NMED, MRED, and ER low and well balanced across configurations, making it a strong candidate for energy-constrained, error-tolerant computing systems.***

***Keywords:*** ***Approximate Compressor, MAC Unit, Multiplier, Neural Network, Balanced Error Accumulation, Approximate Computing.***

## I. INTRODUCTION

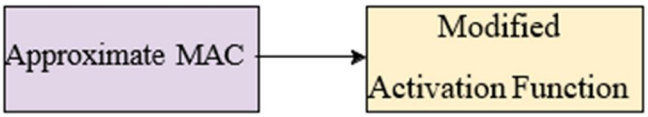
The growing complexity of arithmetic hardware continues to challenge the design of energy-efficient, high-performance computing systems. This challenge is amplified in platforms that operate under variable power budgets, such as energy-harvesting and edge-AI devices, where available power can fluctuate by two or more orders of magnitude across space and time. Prior work on approximate arithmetic has largely targeted fixed, application-specific circuit modifications, with limited attention to compressor structures that can adapt their approximation level to the operating conditions of the system while keeping the resulting error well behaved. Regardless of the operating regime, a computing device must still make forward progress and deliver adequate throughput, motivating architectures that combine approximate computation, machine-learning-aware design, and configurability to meet power constraints without unacceptable loss of output quality. Approximate computing addresses this problem by replacing exact arithmetic components with simplified counterparts that trade a controlled amount of numerical accuracy for reductions in power, delay, and area. Because many target applications are inherently error-resilient — for example, image processing, pattern recognition, and ML inference — small, well-managed inaccuracies are acceptable if they translate into meaningful efficiency gains. Realizing this trade-off in practice requires attention to both algorithmic resilience and the perceptual resilience of the application to error. In recent years, ML accelerators and edge-intelligence platforms have placed increasing pressure on arithmetic units, since MAC operations dominate the computation in convolutional and fully connected layers. Even modest reductions in the energy and delay of the compressors inside multipliers can therefore translate into significant system-level savings, particularly under tight power envelopes. This motivates the need for configurable, compressor-level building blocks that can dynamically adjust their precision and energy consumption to match the instantaneous quality requirements of the running application.

Approximation can be introduced at any level of the design stack, from circuits and logic through to algorithms and software. Within the circuit and logic levels, the compressor trees inside multipliers are a particularly attractive target: multipliers already possess a complex logic structure due to their central role in data processing, and compute-intensive applications rely heavily on large-scale multiplication to produce their results. Consequently, approximate compressor design — simplifying the 4-2, 5-2, and higher-order compressors used to reduce partial products — has become an active and productive area of study, and is the focus of the architecture proposed in this work.

## II. PROPOSED APPROXIMATE COMPRESSOR ARCHITECTURE

To evaluate the effectiveness of the proposed approximate compressor architecture, an adaptive neuron design incorporating a configurable MAC unit (exact and 2-bit/4-bit approximate compression modes) together with a compensated activation function is implemented using synthesizable SystemVerilog. The resulting designs are characterized for power, area, latency, and energy using Xilinx Vivado Design Suite and Synopsys Design Compiler targeting UMC 90 nm CMOS technology, along with a comprehensive error analysis. Based on model-driven accuracy–power trade-offs, the architecture selects between approximate and exact compressor configurations, allowing the processing capability to adapt to fluctuating power or energy budgets at runtime. In addition, a modified activation function that can be tuned to reduce the resulting prediction error is proposed, as illustrated below.




### *A. Proposed Approximate Compressor Design*

The design flow of the proposed compressor architecture begins by ranking the significance of each partial-product position generated in an 8×8 multiplication array, and selectively simplifying the compressor stages that contribute least to final numerical precision. In the exact mode, all partial products are reduced through a conventional Wallace-tree compressor network built from full and half adders, preserving full accuracy. In the approximate compression modes, subsets of low-significance partial products are pruned or merged using simplified compressor cells, reducing the number of active adders and the switching activity in the reduction tree — while the balance of exact and approximate compressor stages is chosen so that the resulting error accumulates predictably rather than growing sharply with operand width. A single configuration control signal selects between the exact and approximate compressor paths, allowing the same hardware to operate across different energy–accuracy trade-offs without requiring separate multiplier blocks for each mode.

Using the exact-mode compressor area as the reference, the proposed configurable structure is sized to accommodate the largest configuration among its d-bit approximate compressor modes; a Wallace-tree reduction network is used as the baseline example in this study, though the same approximate compressor strategy is applicable to other multi-level reduction trees such as Dadda trees. The approximate configurations are realized by rewiring the same adder cells used in the exact configuration rather than instantiating additional hardware, so the proposed 8×8 configurable compressor architecture supports both exact and approximate multiplication, as shown in Fig. 1. In this diagram, each circle represents a partial-product bit; rectangles spanning pairs of bits denote half adders, and boxes spanning three-bit rows denote full adders. Configuration (a) shows the exact Wallace-tree reduction using the maximum number of half- and full-adder cells; configuration (b) shows a compact 4-bit approximate compressor stage; and configuration (c) shows the proposed unified architecture, which combines the exact path with the 2-bit and 4-bit approximate compressor paths while adding only a small number of extra adders beyond what the exact design already requires.

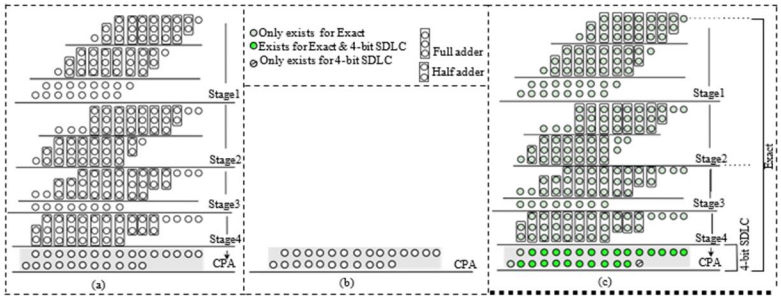



Fig. 1. Reduction steps of an 8×8 Wallace-tree multiplier showing the proposed approximate compressor configurations at two precision levels.

Fig. 2 shows the resulting neuron module, which accepts pairs of weights and inputs, multiplies them using the proposed configurable compressor-based multiplier, and accumulates the results sequentially using a carry-lookahead adder (CLA). The activation function is then applied to the accumulated total to produce the neuron output. The compression level of the MAC unit — i.e., which approximate compressor mode is active — is selected using an additional 2:1 multiplexer (MUX). Reusing the same adder cells and partial-product routing across modes is important from a hardware perspective, since it avoids the area and leakage overhead of instantiating a full duplicate multiplier for every operating mode. Instead, the proposed architecture exploits the structural similarity between the exact and approximate compressor trees, sharing most of the logic and adding only a small number of multiplexers and control lines. This makes the configurable compressor architecture particularly well suited to resource-constrained ML accelerators, where both silicon footprint and standby power are critical design metrics.


Fig. 2. Neuron datapath built around the proposed configurable approximate-compressor MAC unit.

The proposed configurable compressor and MAC architectures were synthesized in UMC 90 nm CMOS technology using Synopsys Design Compiler and evaluated under typical process–voltage–temperature conditions. For each configuration, gate-level netlists were generated and annotated with switching activity obtained from functional simulations in Vivado and MATLAB, enabling consistent estimation of area, dynamic power, leakage power, critical-path delay, and energy per operation across the exact and approximate compressor modes.

The baseline used for comparison is a conventional 8×8 Wallace-tree multiplier followed by an exact MAC implemented with standard carry-lookahead adders. Experimental results show that the 4-bit approximate compressor configuration achieves substantial reductions in power and area relative to the exact baseline, at the cost of a modest, well-balanced increase in error metrics such as MRED and NMED. When the proposed compressor is integrated into the MAC unit and evaluated on image-processing and neural-network workloads, these compressor-level savings translate into lower energy per inference and reduced silicon utilization, while preserving acceptable output quality and classification accuracy. The measured power–accuracy trade-off across the different approximate compressor modes confirms that the architecture can be tuned to meet a range of application requirements.

### *B. Configurable MAC Unit and Error-Balanced Compressor Behavior*

For the Gaussian-blur experiment, images processed with the approximate-compressor MAC exhibited only minor visual degradation relative to the exact implementation, with quality metrics such as peak signal-to-noise ratio remaining within a narrow margin. Similarly, in the multilayer-perceptron case study, the classification-accuracy loss introduced by the approximate compressor configuration remained within a small percentage of the exact baseline, even under aggressive energy constraints. These results indicate that the proposed approximate compressor architecture, together with the configurable MAC unit and compensated activation function, is well suited to error-tolerant inference tasks, where a small, balanced loss in numerical precision is an acceptable trade-off for significant reductions in energy and hardware cost.

This hypothesis forms the basis of the proposed error-compensation approach. A neural network Ne with a set of weights Me can be built using activation function AFe. NeMe, AFe denotes this configuration. Functionally, an equivalent or near-equivalent activation function AFa can be found for network Na such that Na = NaMa. This relationship is expressed as:

$$Na\ \{Ma, AFa\} \approx Ne\ \{Me, AFe\}$$

Here, the precise activation function (AFe) and the exact compressor-based multiplier for weights and inputs (Me) are the components of an exact neuron (Na). The compensation factor CM acts as a bias term that shifts the activation function to counterbalance the systematic error introduced by the approximate compressor configuration. Because the approximate compressor tends to under- or over-estimate the true MAC output by a nearly constant proportion, CM can be tuned using the normalized mean error distance (NMED) so that the average deviation at the neuron output is minimized. This approach is attractive because it requires no modification to the approximate compressor hardware itself; instead, it leverages the flexibility of the activation-function block, which is typically implemented with adders, shifters, or small lookup tables. While metrics such as error rate (ER), NMED, and mean relative error distance (MRED) are useful for evaluating output quality, error distance (ED) alone does not provide comparability across compressor configurations of different sizes; ER, MRED, and NMED are therefore used together when assessing the accuracy of the proposed approximate compressor for error-tolerant systems.

In practice, NMED-driven tuning of CM is performed offline during design-space exploration: for each candidate approximate compressor configuration, exhaustive or Monte-Carlo simulations estimate NMED over the relevant operand range, and the resulting value is mapped directly to a corresponding CM setting. During normal operation, the neuron simply selects the CM value associated with the active compressor mode, allowing the network to switch seamlessly between high-accuracy and energy-saving configurations while keeping the overall prediction error within acceptable, balanced limits.

Table 1. NMED of the proposed approximate compressor for each d-bit configuration of an 8×8 multiplier.

| d-bit↓ | NMED (%) |
|---|---|
| 2-bit | 0.0035 |
| 3-bit | 0.0101 |
| 4-bit | 0.0327 |

Table 2. Error metrics — ED, ER, MRED, and NMED — used to evaluate the proposed compressor configurations.

| Metrics | Error Metric | Approximate circuits | Bit SDLC | Comparing multipliers of different sizes |
|---|---|---|---|---|
| ED | [✓] | [×] | [×] | [×] |
| ER | [✓] | [✓] | [×] | [×] |
| MRED | [✓] | [✓] | [✓] | [×] |
| NMED | [✓] | [✓] | [✓] | [✓] |

As shown above, NMED is the most informative error measure across compressor configurations of different sizes and forms the basis of the proposed neuron system, which combines the configurable approximate-compressor MAC unit with the revised activation function.

An extensive set of simulations was carried out in Vivado and MATLAB with the proposed compressor-based MAC unit modelled functionally, evaluating the response of the approximate compressor across all operand combinations for the 4-bit configuration. To minimize the prediction error introduced by the 4-bit approximate compressor mode, CM is set equal to the corresponding NMED value from Table 1, as expressed in the compensation equation below. As expected, higher-order approximate compressor modes trade lower output quality for reduced energy consumption (e.g., the 4-bit configuration within an 8×8 compressor), following the same general principle used for standard ReLU-style activation functions. The figure below shows the datapath of a standard activation model alongside the proposed modified activation model for a single neuron, both including the CM term and the CLA adder, and highlights the small additional silicon area contributed by the extra adder and registers.

$$CM = NMED$$

$$sig(x) = \left(\frac{1}{1+e^{-x}}\right) + CM$$

where the standard sigmoid function maps the MAC output x into the range (0, 1), and CM is a compensation constant derived from the NMED of the active approximate compressor configuration. The additional CM term shifts this curve vertically to counteract the average error introduced by the approximate compressor, so that the mean deviation between the approximate and exact neuron outputs is minimized.

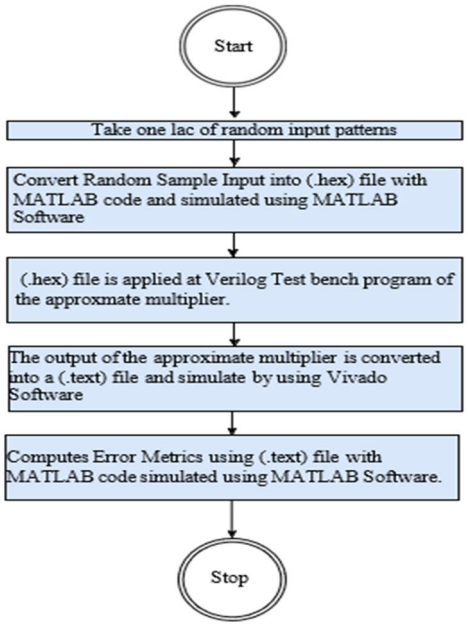
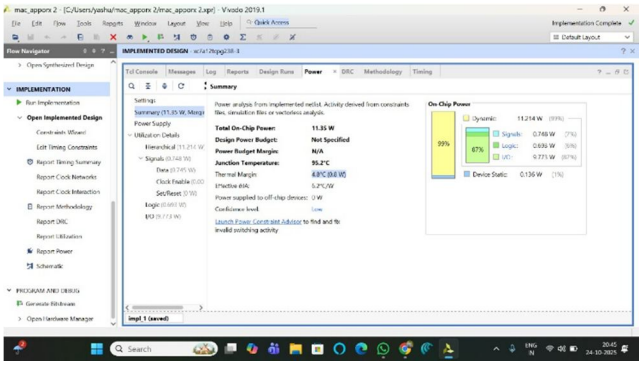



Fig. 3. Procedure used to simulate and compute the error metrics of the proposed approximate compressor architecture.

## III. RESULTS AND DISCUSSION



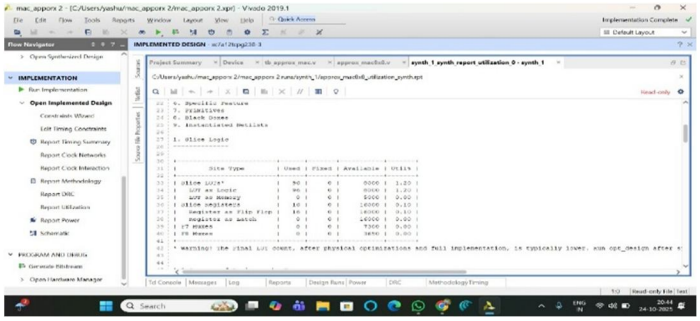


Fig. 4–6. Power, error, and energy trends of the proposed approximate compressor architecture across exact, 2-bit, and 4-bit configurations.

Table 3. Qualitative comparison of prior approximate designs and the proposed approximate compressor architecture.

| Method / Reference | Power Saving | Limitation |
|---|---|---|
| Approximate multiplier [1] | High | Accuracy loss |
| Approximate compressor [3], [4] | Medium | Large error |
| Proposed approximate compressor architecture | High | Minimal, balanced error |

As summarized in Table 3, conventional approximate multipliers provide substantial power savings but often incur noticeable accuracy loss when used directly in MAC-intensive workloads. Approximate compressors tailored for partial-product reduction achieve moderate power reduction but may introduce relatively large error magnitudes, degrading output quality in error-sensitive applications. In contrast, the proposed approximate compressor architecture combines configurable compression with a modified, error-compensated activation function to maintain high power savings while keeping the effective error at the neuron output minimal and, critically, balanced across configurations rather than concentrated in the most significant bits. This combination allows the architecture to exploit the benefits of approximate arithmetic without sacrificing application-level quality, making it more suitable for energy-constrained ML systems than prior fixed-approximation designs.

## IV. CONCLUSION

This work has presented an efficient approximate compressor architecture for balanced error accumulation in MAC units, targeting energy-constrained, error-tolerant applications. Approximate computing offers design techniques at many abstraction levels for building on-chip systems that are efficient in both energy and performance, and approximate compressors — a key subfield of approximate arithmetic — have drawn considerable attention in the literature. The central contribution of this work is a configurable compressor structure whose approximation level can be tuned on the fly through logic-compression control, so that a single set of adder cells serves the exact mode as well as multiple approximate compression modes. An energy-aware configuration strategy selects the appropriate compressor mode based on available energy, and the results show that the design executes reliably at runtime even under highly variable energy conditions, while reusing the same adder cells across configurations saves silicon area and leakage energy.

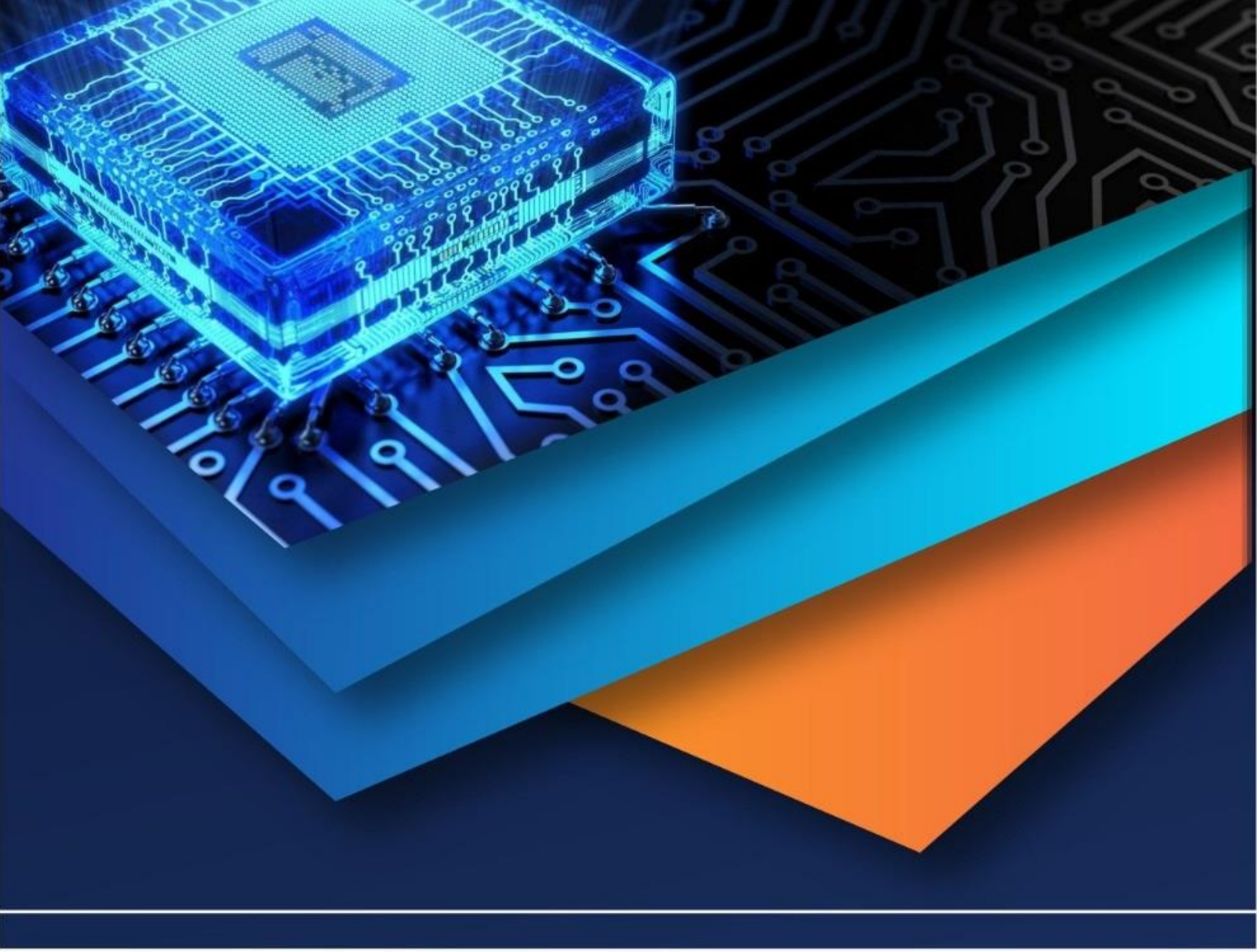
Building on the proposed compressor, a configurable MAC unit was designed for power-adaptive ML hardware, and its configuration is optimized at runtime using the same energy-aware strategy. The resulting MAC unit was deployed within neuron modules of an ML system, and a model-driven, power-adaptive neuron architecture was constructed around it. Despite significant reductions in energy use and area, the resulting drop in output quality was minimal. Finally, a customizable, error-compensated activation function was proposed to further reduce the impact of the approximate compressor on prediction accuracy. Together, these three contributions — the approximate compressor, the configurable MAC unit, and the compensated activation function — allow energy-adaptive neural-network hardware to realize substantial power and area savings with negligible degradation in output quality.

Future work will extend the proposed approximate compressor architecture to larger operand sizes, such as 16×16 and 32×32 multipliers, and integrate the resulting MAC units into more complex deep-learning accelerators. Another promising direction is to co-optimize the approximate compressor hardware with quantization-aware training so that the neural network can explicitly adapt its weights to the characteristics of the underlying arithmetic units. Finally, implementing and validating the architecture on more advanced technology nodes and additional FPGA platforms will provide further insight into its scalability and practical deployment in real-world edge-AI systems.

## REFERENCES

[1] Venkatachalam, S., & Ko, S.-B. (2017). Design of power and area efficient approximate multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(5), 1782–1786.
[2] Liang, J., Han, J., & Lombardi, F. (2013). New metrics for the reliability of approximate and probabilistic adders. IEEE Transactions on Computers, 62(9), 1760–1771.
[3] Chang, C., Gu, J., & Zhang, M. (2004). Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 51(10), 1985–1997.
[4] Pei, H., Yi, X., Zhou, H., & He, Y. (2021). Design of ultra-low power consumption approximate 4-2 compressors based on the compensation characteristic. IEEE Transactions on Circuits and Systems II: Express Briefs, 68(1), 461–465.
[5] Jiang, H. L., Liu, C., Liu, L. B., Lombardi, F., & Han, J. (2017). A review, classification, and comparative evaluation of approximate arithmetic circuits. ACM Journal on Emerging Technologies in Computing Systems, 13(4), Article 60.
[6] Nambi, S., Kumar, U. A., Radhakrishnan, K., Venkatesan, M., & DeBAM, A. S. (2021). Decoder-based approximate multiplier for low power applications. IEEE Embedded Systems Letters, 13, 174–177.
[7] Yang, Z., Han, J., & Lombardi, F. (2015). Approximate compressors for error resilient multiplier design. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS) (pp. 183–186).
[8] Lau, M. S. K., Ling, K. V., & Chu, Y. C. (2009). Energy-aware probabilistic multiplier: Design and analysis. In Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (pp. 281–290). Grenoble, France.
[9] Di Meo, G., Saggese, G., Strollo, A. G. M., & De Caro, D. (2023). Design of generalized enhanced static segment multiplier with minimum mean square error for uniform and nonuniform input distributions. Electronics, 12, 1–21. https://doi.org/10.3390/electronics12010234
[10] Hwang, S., Seok, H., & Kim, Y. (2024). Design of an approximate 4-2 compressor with error recovery for efficient approximate multiplication. Journal of Semiconductor Technology and Science, 24(4), 305–315.
[11] Han, J., & Orshansky, M. (2013). Approximate computing: An emerging paradigm for energy-efficient design. In Proceedings of the 18th IEEE European Test Symposium (ETS). IEEE.
[12] Pashaeifar, M., et al. (2018). Approximate reverse carry propagate adder for energy-efficient DSP applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 26(11), 2530–2541.
[13] Ye, R., et al. (2013). On reconfiguration-oriented approximate adder design and its application. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE.
[14] Di Meo, G., et al. (2023). Approximate MAC unit using static segmentation. IEEE Transactions on Emerging Topics in Computing, 12(4), 968–979.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY