IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

FPGA-Based High-Performance Multiply–Accumulate Unit Using a Radix-4 Modified Booth Multiplier and Error-Correctable Carry Look-Ahead Adder

Vasikarla Renuka[1], N. Bhavana[2], Bandla Sandhya[3]
Assistant Professor, Dept. of ECE[1,2]
Dept. of ECE[3]
Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India.
vasikarlarenuka1@gmail.com, nbhavana755@gmail.com, sandhyabandla8@gmail.com

Abstract: *Multiply–Accumulate (MAC) units dominate the arithmetic workload of digital signal processing, communication, image-processing, and machine-learning datapaths. Conventional MAC architectures often realize multiplication and accumulation as distinct hardware stages, typically combining an array multiplier with a ripple-carry or other carry-propagating adder. The supplied design study identifies the resulting partial-product count, carry propagation, resource overhead, and power consumption as major limitations. This paper presents an FPGA-oriented MAC architecture that replaces the conventional multiplier with a Radix-4 Modified Booth Multiplier (MBM) and performs accumulation using an Error-Correctable Carry Look-Ahead Adder (EC-CLA). Radix-4 Booth recoding compresses the multiplier representation into coefficients drawn from {0, ±Y, ±2Y}, reducing the number of partial products. The EC-CLA precomputes generate and propagate signals and supplements the carry network with redundant/parity-based error checking, while a data-storage stage retains accumulated results and enables feedback for repeated MAC operations. The design is described in Verilog HDL and evaluated in Xilinx Vivado targeting a Zynq-7000 device. Reported results for N=8, 16, and 32 show consistent reductions in resource usage, total on-chip power, and delay. For N=32, LUT utilization decreases from 1505 to 1323, total on-chip power from 103.437 µW to 81.64 µW, and total delay from 75.487 ns to 23.705 ns. These correspond to reductions of approximately 12.1%, 21.1%, and 68.6%, respectively. The results indicate that the proposed MBM–EC-CLA combination is a practical architecture for high-throughput arithmetic kernels where power, latency, and FPGA resource utilization must be jointly controlled.*

Keywords: Multiply–Accumulate, MAC, Modified Booth Multiplier, Radix-4, Carry Look-Ahead Adder, error correction, FPGA, Verilog HDL, Xilinx Vivado, Zynq-7000

I. INTRODUCTION

The multiply–accumulate operation is one of the most frequently executed arithmetic primitives in modern digital hardware. A MAC computes a product and adds it to a running sum, making it the basic computational kernel in finite-impulse-response filters, convolution, correlation, matrix multiplication, transforms, neural-network inference, image processing, and communication systems. In VLSI implementations, the efficiency of the MAC directly influences throughput, energy consumption, and silicon or FPGA resource cost.[1-7].

A conventional MAC is typically organized as a multiplier, an adder, and an accumulator register. Although this decomposition is simple, its critical path can become long because multiplication generates many partial products and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

accumulation requires carry propagation. The supplied project uses an array multiplier and ripple-carry adder as the baseline. In that architecture, partial products are generated through AND-gate arrays and are added through a regular adder structure, while the ripple-carry network propagates carry information sequentially from lower to higher bit positions. The resulting design is accurate and structurally regular, but its area, power, and delay grow rapidly with operand width.[9-15].

The central design problem is therefore to reduce multiplication complexity and accumulation latency without sacrificing exact arithmetic. The proposed architecture addresses the multiplication stage through Radix-4 Modified Booth recoding, which reduces the number of partial-product rows, and addresses the accumulation stage through carry look-ahead generation. The adder is further described as error-correctable by using redundant generate/propagate signals and parity-based fault indication. A storage stage preserves the accumulated value and returns it to the adder for the next MAC cycle.[16-20].

1.1 Research Motivation and Contributions

A Radix-4 Modified Booth Multiplier is used in place of the array multiplier to reduce partial-product generation.

An Error-Correctable Carry Look-Ahead Adder is used to accelerate accumulation by precomputing carry information.

Redundant generate/propagate paths and parity comparison are incorporated conceptually to improve arithmetic reliability.

A storage/accumulator stage closes the MAC feedback loop and supports repeated accumulation.

The design is evaluated in Xilinx Vivado on a Zynq-7000 target and compared with the baseline using LUT count, power, logic delay, net delay, and total delay.

Scalability is examined at N=8, N=16, and N=32, with the 32-bit implementation providing the most substantial latency improvement.

II. RELATED WORK AND RESEARCH GAP

Recent MAC research spans approximate arithmetic, computing-in-memory, time-domain arithmetic, hybrid-logic implementations, FPGA-based accelerators, and application-specific multiplier/adder structures. Approximate MAC architectures can reduce power significantly when some numerical error is acceptable, while compute-in-memory designs reduce data movement at the cost of mixed-signal complexity. FPGA-oriented work has explored Vedic multipliers, Wallace trees, carry-select adders, pipelining, sparsity-aware arithmetic, and BIST-enhanced MAC blocks.

The literature included with the supplied study also reports several complementary directions. Static-segmentation MAC units reduce hardware through controlled approximation; hybrid-logic MACs target transistor-level power and area reductions; Vedic and Wallace-tree multipliers improve partial-product processing; and error-compensation techniques improve efficiency for CNN workloads. Other works use time-domain MAC operations, SRAM or RRAM compute-in-memory, and reconfigurable arithmetic to increase throughput per watt.

The research gap addressed here is narrower and hardware-oriented: an exact FPGA MAC is required that simultaneously reduces the number of multiplier partial products, shortens carry propagation in the accumulator, and retains a reliability mechanism without depending on analog computation or approximate arithmetic. The proposed Radix-4 MBM + EC-CLA architecture directly targets this gap and can be implemented using conventional RTL design and FPGA synthesis flows.

III. BASELINE MAC ARCHITECTURE

The baseline system consists of an array multiplier, a ripple-carry adder (RCA), and a data-storage device. For each multiplier bit, the array multiplier forms a shifted partial product and sums the set of partial products. The RCA then adds the multiplication result to the previous accumulated value. Because each carry in an RCA depends on the carry generated by the preceding bit, its delay increases with word length. In addition, the array multiplier instantiates a large and regular

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

network of partial-product generators and adders, causing resource consumption and switching activity to rise with operand width.

3.1 Limitations of the Baseline

- High partial-product count in the multiplier.
- Sequential carry propagation in the ripple-carry accumulation path.
- Higher FPGA resource consumption as bit width increases.
- Larger net delay because of extensive routing between partial-product and adder structures.
- Higher dynamic and total on-chip power under the supplied implementation conditions.

IV. PROPOSED MAC ARCHITECTURE

The proposed design combines a Radix-4 Modified Booth Multiplier, an Error-Correctable Carry Look-Ahead Adder, and an accumulator/data-storage stage. The multiplier receives operands A and B, performs Booth recoding and partial-product generation, and forwards the product terms to the EC-CLA. The EC-CLA combines the current product with the stored accumulated value. The updated sum is written back to the storage stage and is simultaneously available as the MAC output.

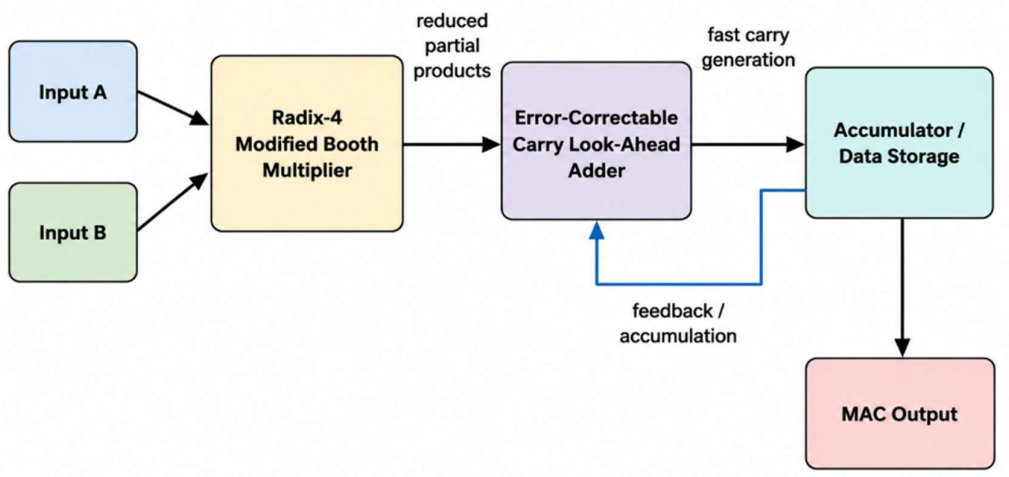

Figure 1. Proposed Radix-4 MBM + EC-CLA MAC architecture.

4.1 MAC Operation

For a sequence of input pairs, the MAC recurrence is represented as:

$$\boldsymbol{ACC(k) = A(k) \times B(k) + ACC(k-1)}$$

The storage stage retains ACC(k−1), allowing the next multiplication result to be accumulated without external memory access. This organization also creates a natural register boundary for pipelining.

4.2 Radix-4 Modified Booth Multiplier

The Modified Booth Multiplier reduces the number of partial products by recoding the multiplier in overlapping groups of three bits. Each group selects one of five effective multiples of the multiplicand: 0, +Y, +2Y, −Y, or −2Y. Compared with bit-by-bit partial-product generation, Radix-4 recoding approximately halves the number of partial-product rows, thereby reducing the amount of addition and routing required in the multiplier. Signed values are naturally represented through the positive and negative recoding terms.

DOI: 10.48175/IJARSCT-38272

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

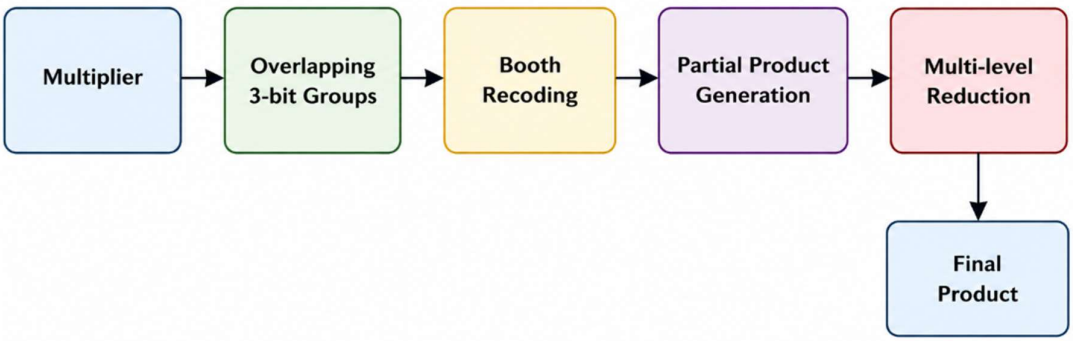

Figure 2. Radix-4 Modified Booth multiplication flow.

3-bit Booth group	Selected partial product
000	0
001	+Y
010	+Y
011	+2Y
100	−2Y
101	−Y
110	−Y
111	0

Table 1. Radix-4 Modified Booth encoding used by the proposed multiplier.

4.3 Error-Correctable Carry Look-Ahead Adder

The accumulation path uses a carry look-ahead structure so that carry values are derived from generate and propagate functions rather than waiting for a ripple chain. For bit position i, the supplied design defines the generate and propagate relations as:

$$G_i = A_i \cdot B_i$$
$$P_i = A_i + B_i$$
$$C_{i+1} = G_i + P_i \cdot C_i$$

In addition to the standard generate and propagate paths, the proposed concept introduces redundant generate/propagate logic. Parity comparisons between standard and redundant results provide an error indication, and corrected signals can then be selected before the look-ahead carry network is evaluated. This structure combines the latency advantage of carry look-ahead addition with a fault-detection mechanism suitable for reliability-oriented digital systems.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

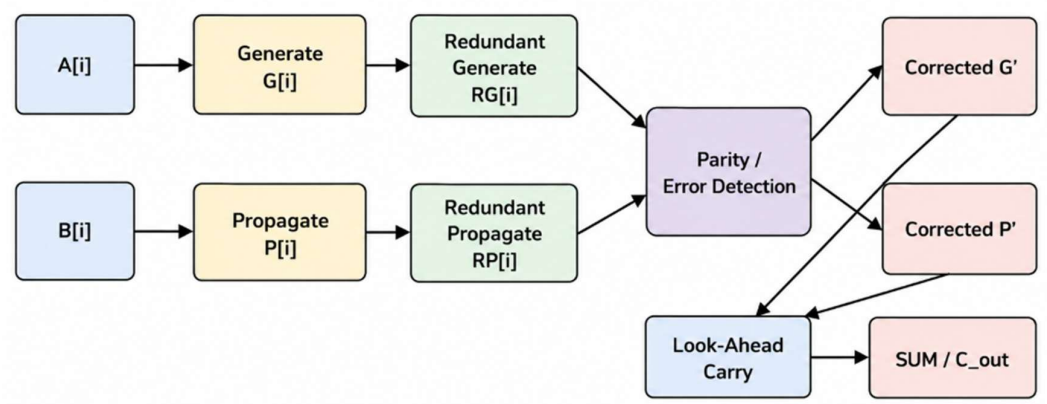

Figure 3. Conceptual EC-CLA dataflow with redundant/parity-based checking.

4.4 Data Storage, Feedback, and Pipelining

Registers or equivalent storage elements are used at the MAC output to preserve the accumulated result. The stored value is fed back to the EC-CLA and added to the next multiplication result. This organization reduces repeated external data access, keeps intermediate arithmetic local to the MAC block, and supports pipelined execution when register boundaries are inserted between multiplication, accumulation, and output stages.

V. RTL IMPLEMENTATION AND VERIFICATION METHODOLOGY

The architecture is implemented using Verilog HDL and evaluated in Xilinx Vivado. The supplied project targets the Zynq-7000 family, part number xc7z007sclg400-1, package clg400, speed grade −1. The RTL design is organized into multiplier, adder/error-correction, storage, and top-level MAC modules.

Functional verification is performed using simulation waveforms. The N=32 testbench applies operands a and b, an enable signal en, and a clock. For each enabled cycle, the multiplier forms the product, the adder combines it with the accumulated value, and the new value is stored for the next cycle. Correct operation is therefore verified by observing the repeated multiply-plus-feedback behavior across successive clock cycles.

After functional verification, synthesis and implementation reports are used to extract LUT utilization, flip-flop and I/O usage, power, and timing. The comparison is performed against the baseline array-multiplier/RCA MAC under the same general FPGA flow so that relative architectural effects can be observed.

5.1 Evaluation Metrics

Area: LUT count, flip-flop count, I/O count, and global-buffer use.

Power: dynamic components and total on-chip power reported by Vivado.

Timing: logic delay, net delay, total setup-path delay, and hold delay.

Scalability: comparison for N = 8, 16, and 32-bit operand widths.

VI. RESULTS AND DISCUSSION

6.1 N=32 FPGA Resource Utilization

For the proposed 32-bit design, the supplied Vivado report uses 1323 LUTs out of 133800 available LUTs (0.99%), 64 flip-flops out of 267600 (0.02%), 130 I/O resources out of 500 (26.0%), and one BUFG out of 32 (3.13%). The key comparison metric is the LUT count: the baseline uses 1505 LUTs while the proposed design uses 1323.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

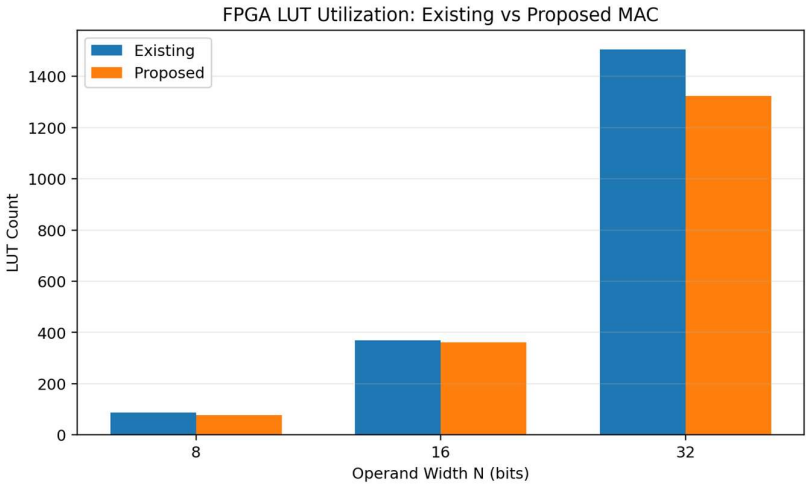

Figure 4. LUT-count comparison for N=8, 16, and 32.

6.2 Power Results

The 32-bit baseline consumes 103.437 µW total on-chip power. Its reported dynamic components include 17.083 µW signal power, 25.080 µW logic power, and 60.039 µW I/O power, with 1.235 µW PL static power. The proposed 32-bit implementation reduces total power to 81.64 µW. The proposed report includes 10.218 µW signal power, 10.090 µW logic power, 60.098 µW I/O power, and the same 1.235 µW PL static power. The reduction is therefore concentrated in switching and logic activity rather than static leakage.

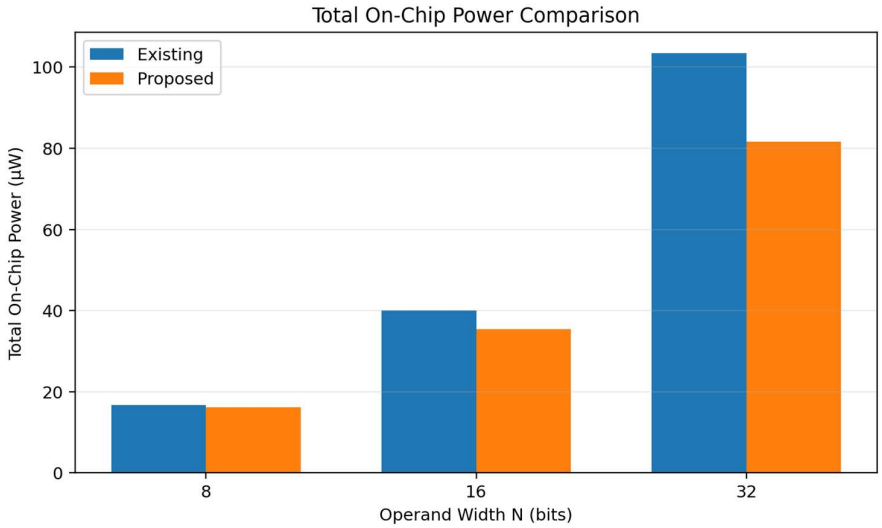

Figure 5. Total on-chip power comparison.

6.3 Timing Results

Timing is the strongest improvement reported by the proposed architecture. At N=32, total delay falls from 75.487 ns to 23.705 ns. The logic component decreases from 11.122 ns to 7.669 ns, while net delay decreases from 64.365 ns to 16.036 ns. The large reduction in net delay is consistent with a simpler partial-product structure and a shorter carry path. The reported proposed hold delay is 0.589 ns, with 0.345 ns logic and 0.244 ns net components.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

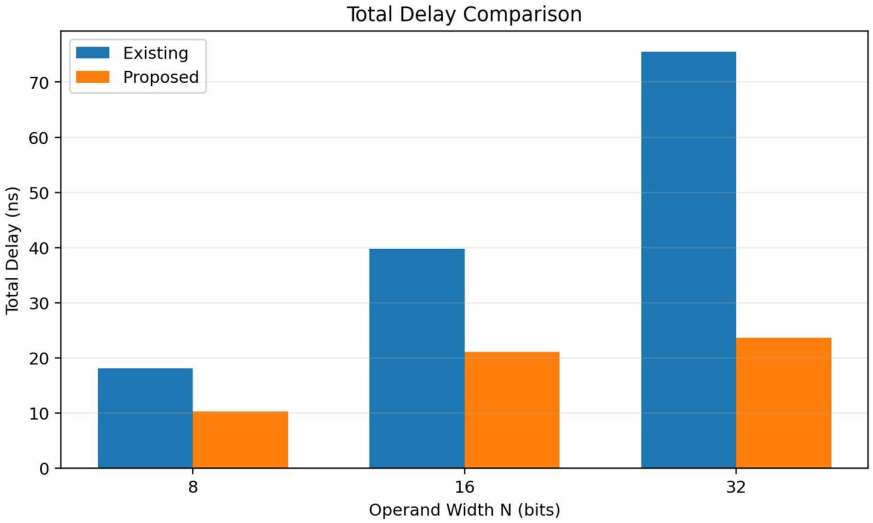

Figure 6. Total-delay comparison.

6.4 Cross-Width Comparison

N	Metric	Existing	Proposed	Reduction	Unit
8	LUTs	87	78	10.34%	count
16	LUTs	369	362	1.90%	count
32	LUTs	1505	1323	12.09%	count
8	Total power	16.763	16.178	3.49%	µW
16	Total power	40.005	35.462	11.36%	µW
32	Total power	103.437	81.640	21.07%	µW
8	Total delay	18.114	10.338	42.93%	ns
16	Total delay	39.752	21.123	46.86%	ns
32	Total delay	75.487	23.705	68.60%	ns

Table 2. Summary of reported scalability results.

At N=8, the proposed design reduces LUT usage by approximately 10.34%, total on-chip power by 3.49%, and total delay by 42.93%. Even at the smallest evaluated width, the carry and routing optimization has a clear timing benefit.

At N=16, LUT count changes only slightly (369 to 362), but total on-chip power falls by about 11.36% and total delay by about 46.86%. This indicates that the architectural benefit is not limited to area reduction; switching activity and the critical path also improve.

At N=32, the reduction becomes more pronounced. LUT count decreases by approximately 12.09%, total on-chip power by 21.07%, logic delay by 31.05%, total delay by 68.60%, and net delay by about 75.09%. These values suggest that the baseline array/RCA structure scales less favorably than the Radix-4 MBM + EC-CLA structure as operand width increases.

The power distribution also reveals an important trade-off. I/O power does not decrease materially in the N=32 report; the principal savings occur in signal and logic power. Therefore, further optimization of the complete system would require interface activity reduction, clock gating, or a more integrated data path rather than arithmetic changes alone.

6.5 Summary of N=32 Improvement

Metric	Existing	Proposed	Improvement
LUT count	1505	1323	12.09% reduction

DOI: 10.48175/IJARSCT-38272

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

Total on-chip power	103.437 µW	81.640 µW	21.07% reduction
Logic delay	11.122 ns	7.669 ns	31.05% reduction
Total delay	75.487 ns	23.705 ns	68.60% reduction
Net delay	64.365 ns	16.036 ns	75.09% reduction

Table 3. Key improvements for the 32-bit MAC.

VII. APPLICATIONS

Digital signal processing: FIR filters, convolution, correlation, transforms, and adaptive filtering repeatedly execute MAC operations.

Machine learning and AI: Convolution and matrix–vector multiplication contain dense MAC workloads and benefit from reduced latency and resource usage.

Image and medical processing: Filtering, enhancement, feature extraction, ultrasound, and MRI processing require high-throughput arithmetic.

Communication systems: Modulation, equalization, channel estimation, and digital filtering depend on multiply–accumulate operations.

Robotics and automotive systems: Sensor fusion, feedback control, radar processing, and perception algorithms require deterministic real-time computation.

FPGA accelerators: The modular RTL structure is suitable for integration into application-specific FPGA datapaths and systolic processing arrays.

VIII. LIMITATIONS AND DESIGN CONSIDERATIONS

The reported power numbers are tool estimates from a particular FPGA design and activity model; absolute values depend on switching assumptions, clocking, I/O activity, and implementation constraints.

The supplied document describes parity/redundancy and majority-voting concepts for the EC-CLA, but the degree of fault-injection validation is not documented in the extracted results. Reliability claims should therefore be experimentally validated in future work.

The comparison emphasizes FPGA implementation metrics and does not include post-layout ASIC area or transistor-level energy measurements.

At higher widths, routing and interface power can become dominant; arithmetic optimization alone may not minimize total system energy.

The present paper focuses on the consistently reported N=8, 16, and 32 results. Wider configurations should be re-verified under identical constraints before being used for quantitative claims.

IX. CONCLUSION

This paper presented an FPGA-oriented high-performance MAC architecture that combines Radix-4 Modified Booth multiplication with an Error-Correctable Carry Look-Ahead Adder and an accumulator/data-storage feedback stage. The multiplier reduces the number of partial-product rows, the carry look-ahead network shortens accumulation latency, and the redundancy/parity mechanism provides a basis for arithmetic error detection. Vivado results from the supplied design show consistent improvement as operand width increases. For the 32-bit case, the proposed MAC reduces LUT count from 1505 to 1323, total on-chip power from 103.437 µW to 81.64 µW, and total delay from 75.487 ns to 23.705 ns. The corresponding reductions of approximately 12.1%, 21.1%, and 68.6% demonstrate that the proposed architecture addresses the central VLSI objectives of area, power, and speed simultaneously. The design is therefore suitable as a reusable arithmetic core for DSP, communication, image-processing, and FPGA-based AI acceleration.

X. FUTURE WORK

Pipeline the MBM and EC-CLA stages to increase clock frequency and throughput.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

Add operand isolation and clock gating to reduce switching activity and I/O-related power.
Perform systematic fault injection to quantify EC-CLA error-detection coverage and correction capability.
Evaluate signed fixed-point saturation, rounding, and overflow handling for DSP and neural-network workloads.
Compare the architecture against Wallace-tree, Dadda-tree, Vedic, and DSP-slice-based FPGA MAC implementations under identical constraints.
Map the design to ASIC standard cells for post-layout power, timing, and area evaluation.
Integrate arrays of the proposed MAC into systolic or tensor-processing architectures for convolution and matrix multiplication.

REFERENCES

[1] G. Di Meo, G. Saggese, A. G. M. Strollo, and D. De Caro, "Approximate MAC unit using Static Segmentation," IEEE Transactions on Emerging Topics in Computing, 2023.
[2] M. Iliyas, F. Anjum, and M. Javeed, "High speed dynamic shift register for convolution encoding and viterbi decoding," International Journal of Advanced Science and Technology, vol. 29, no. 5s, pp. 613–619, 2020. Google Scholar
[3] R. S. Mishra, P. Gour, S. Dhariwal, G. Kumar, and A. Anand, "Design and Analysis of Low Power MAC for DSP Processor," Proc. ICAIA/ATCON-1, 2023.
[4] P. S. Locatelli, D. M. Colombo, and K. El-Sankary, "Time-Domain Multiply–Accumulate Unit," IEEE Transactions on VLSI Systems, 2023.
[5] X. Xu, Q. Wei, Y. Zhang, H. Cai, and B. Liu, "Error-Compensation-Based Energy-Efficient MAC Unit for CNNs," CASES, 2023.
[6] J. Ponraj, R. Jeyabharath, P. Veena, and T. Srihari, "High-performance multiply-accumulate unit by integrating binary carry select adder and counter-based modular Wallace tree multiplier for embedding system," Integration, vol. 93, 2023.
[7] R. Vinoth and M. V. R. Kasyap, "Design and Implementation of High Speed 32-bit MAC Unit," Journal of Physics: Conference Series, vol. 2571, no. 1, 2023.
[8] I. Grout, "BIST Design and Implementation for a Fixed-Point Arithmetic MAC Unit within a Systolic Array," iEECON, 2023.
[9] K. V. Gowreesrinivas, B. Usha Sri, S. Saideepak, G. Tarun, and I. Sathya Sagar, "Design and Implementation of Hybrid Full Adder Based 16-bit Multiplication Using FPGA," IEEE DevIC, 2023.
[10] Dr. R. Shrivastava, M. Javeed and M. Jain, "Image Segmentation Based on Edge Detection and Enhancement Based on EECS Algorithm," *International Journal of Advanced Science and Technology*, vol. 29, no. 3s, pp. 330–341, 2020. Google Scholar
[11] H. T. Tesfai, H. Saleh, M. Meribout, M. Al-Qutayri, and T. Stouraitis, "Efficient Mux-Based Multiplier for MAC Unit," International Conference on Microelectronics, 2023.
[12] W. Li, J. Read, H. Jiang, and S. Yu, "MAC-ECC: In-Situ Error Correction and Its Design Methodology for Reliable NVM-Based Compute-in-Memory Inference Engine," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 12, no. 4, 2022.
[13] R. Saravanan, S. Bavikadi, S. Rai, A. Kumar, and S. M. P. Dinakarrao, "Reconfigurable FET Approximate Computing-based Accelerator for Deep Learning Applications," IEEE ISCAS, 2023.
[14] R. Samanth and S. G. Nayak, "An Efficient Two-phase Clocked Sequential Multiply-Accumulator Unit for Image Blurring," International Journal of Electronics and Telecommunications, vol. 68, 2022.
[15] P. Anuradha, K. Rajkumar, R. Arabelli, and R. Shareena, "Implementation of high speed 64-bit MAC unit using FPGA," AIP Conference Proceedings, vol. 2418, 2022.
[16] M. Javeed and T. Srujana, "Multi-Modal Medical Image Fusion using Improved Guided Imaging Filter," Turkish Journal of Computer and Mathematics Education, vol. 10, no. 3, pp. 856–863, 2019. Google Scholar

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

[17] H. Wang et al., "A Charge Domain SRAM Compute-in-Memory Macro With C-2C Ladder-Based 8-Bit MAC Unit in 22-nm FinFET Process for Edge Inference," IEEE Journal of Solid-State Circuits, vol. 58, no. 4, 2023.
[18] M Senthil Kumar and Md Javeed. An efficient rice variety identification scheme using shape, harlick & color feature extraction and multiclass svm. Int J Eng Adv Technol, 8(6):3629–3632, 2019. Google Scholar
[19] S. Widmer et al., "Design of Time-Encoded Spiking Neural Networks in 7nm CMOS Technology," IEEE Transactions on Circuits and Systems II: Express Briefs, 2023.
[20] X. Mao et al., "An Energy-Efficient Mixed-Bitwidth Systolic Accelerator for NAS-Optimized Deep Neural Networks," IEEE Transactions on VLSI Systems, vol. 30, no. 12, pp. 1878–1890, 2022.