Mac Unit Design For AI Accelerator

Sudhamalla Mallesh[1], Yelgapuri Shivani[2], Nareddy Laxmikanth[3], Desireddy Srinivas[4], Mansuju Umesh Chandra[5]

[1,2,3,4,5]Department of ECE, CMR Technical Campus, Hyderabad ,Telangana, India
Corresponding mail:shivaniyelgapuri9@gmail.com

ABSTRACT—Efficient arithmetic hardware is essential for modern computing systems, especially in artificial intelligence and digital signal processing. This paper presents the design and functional simulation of a 2×2 Multiply–Accumulate (MAC) array developed using Verilog HDL. The architecture is organized to execute multiplication and accumulation operations in parallel, improving processing efficiency while maintaining a simple and modular structure.

The proposed design is verified through simulation in Xilinx Vivado 2016.2 using a dedicated testbench. The obtained results demonstrate correct functionality for the applied input values and validate the reliability of the design. The developed MAC array serves as a basic building block that can be expanded for larger hardware accelerators and FPGA-based applications in the future.

Keywords—Multiply–Accumulate (MAC), Verilog HDL, Vivado, Digital Design, FPGA, Parallel Processing.

I. INTRODUCTION

The rapid advancement of Artificial Intelligence (AI) and Machine Learning (ML) has increased the demand for high-speed and efficient computing hardware. One of the key components of AI accelerators is the Multiply–Accumulate (MAC) unit, which performs multiplication and accumulation operations that are essential for neural networks, digital signal processing, and image processing. In this work, a 2×2 MAC array is designed using Verilog HDL and verified through simulation in Xilinx Vivado. The proposed architecture enables parallel processing, improving computational efficiency while maintaining a simple and scalable design. The simulation results confirm the correct functionality of the MAC array, making it suitable for future FPGA- and ASIC-based AI hardware applications.

ABBREVIATIONS AND ACRONYMS

AI,ML,MAC,HDL,FPGAASIC,VHDL

II. LITERATURE SURVEY

Recent research focuses on improving AI and digital signal processing performance through efficient matrix multiplication and parallel computing. The Multiply–Accumulate (MAC) unit is a key component used to accelerate these operations. Researchers have proposed systolic array architectures and optimized MAC designs to improve speed, reduce power consumption, and enhance hardware efficiency. These developments support the design of scalable and high-performance AI accelerator systems[1].

The paper presents EncodingNet, an efficient MAC architecture for deep neural networks that replaces conventional multiplication with a logic-based encoding technique. This approach reduces hardware complexity by using simple logic gates and encoded bit representations instead of traditional multipliers. Experimental results demonstrate significant reductions in power consumption and chip area while maintaining comparable computational accuracy. The proposed architecture offers a low-power, high-speed, and area-efficient solution, making it well suited for AI accelerator and neural network hardware applications [2].

This work presents an optimized Multiply–Accumulate (MAC) unit for machine learning accelerators using a bit decomposition Dadda Tree multiplier combined with Carry Look-Ahead (CLA) and Brent–Kung adders. The proposed architecture improves computation speed through parallel processing while reducing delay, power consumption, and hardware area. The design is implemented and verified in Vivado,where simulation results demonstrate better performance than conventional MAC units. The proposed approach provides an efficient and reliable solution for high-speed, low-power AI accelerator applications [3].

This work presents a low-power, high-performance Multiply–Accumulate (MAC) unit designed for digital signal processing and neural network applications. The proposed architecture combines a modified Vedic multiplier, Wallace tree, and Kogge–Stone adder with a three-stage pipelined design to improve speed and throughput. Implemented using Verilog HDL and validated through VLSI design tools, the MAC unit achieves low power consumption, reduced delay, and efficient area utilization. The architecture is well suited for high-speed DSP, AI accelerator, and embedded system applications [4].

This work proposes an automated MAC design framework using a Python-based Verilog generator. It employs a GD multiplier and carry-skip adder to improve speed and reduce power consumption. The design achieves better efficiency and scalability, making it suitable for AI and DSP applications [5].

This work presents an optimized Multiply–Accumulate (MAC) unit using a Counter-Based Modular Wallace Tree Multiplier and a High-Speed Binary Carry Select Adder. The proposed pipelined architecture improves speed while reducing power consumption and hardware area. Implemented in Xilinx Vivado, the design demonstrates better performance than existing MAC architectures [6].

This work presents a precision-configurable Multiply–Accumulate (MAC) unit for neural network accelerators that supports multiple bit-widths and approximate computing. The proposed architecture reduces hardware area and power consumption while maintaining high accuracy. Its flexible and scalable design makes it suitable for energy-efficient AI and edge computing applications [7].

This work proposes an energy-efficient memristive Multiply–Accumulate (MAC) unit for neural network applications using in-memory computing. The design performs multiplication and accumulation in analog form, reducing computation delay and energy

consumption while maintaining good accuracy. The proposed architecture is highly suitable for low-power AI accelerators and edge computing systems.

Furthermore, the proposed architecture can be evaluated based on power consumption, processing delay, area utilization, and computational accuracy. Although analog computing improves energy efficiency, factors such as noise and device variability must be considered. By optimizing the design, the memristive MAC unit aims to provide faster and more efficient neural network computations. This approach offers a promising solution for next-generation AI accelerators and low-power computing systems.[8].

III. PROPOSED SYSTEM

The proposed system implements a 2×2 Multiply–Accumulate (MAC) array using Verilog HDL and verifies its functionality through Xilinx Vivado simulation. The architecture performs multiplication and accumulation in parallel, improving processing speed and reducing execution time. A modular design is adopted to simplify hardware implementation and support future expansion to larger MAC arrays. Compared to conventional approaches, the proposed system provides better computational efficiency, lower latency, and improved scalability, making it suitable for AI accelerator and digital signal processing applications.

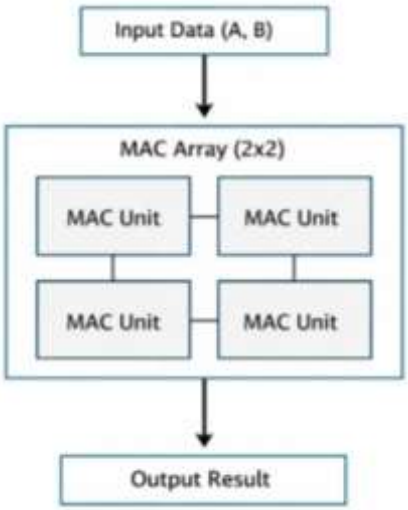


FIGURE1: PROPOSED MODEL

IV. EXISTING SYSTEM

Conventional computing systems mainly rely on general-purpose processors to perform arithmetic operations required by AI applications. Since these operations are executed one after another, the overall computation becomes slower for large workloads. Existing MAC implementations also have limited parallel processing capability, leading to increased execution time and higher power consumption. Furthermore, their scalability is restricted, making them less efficient for modern AI applications that require fast and continuous matrix computations.

V. RESULTS

Figure 2 presents the initial simulation of the MAC unit array. At the beginning, some signals appear in the 'X' (unknown) and 'Z' (high-impedance) states because the input signals are not initialized. Once the clock and reset signals are applied, the signals gradually become stable, confirming that the MAC unit array is initialized and operates correctly.

Figure 2:MAC Array initial simulation output

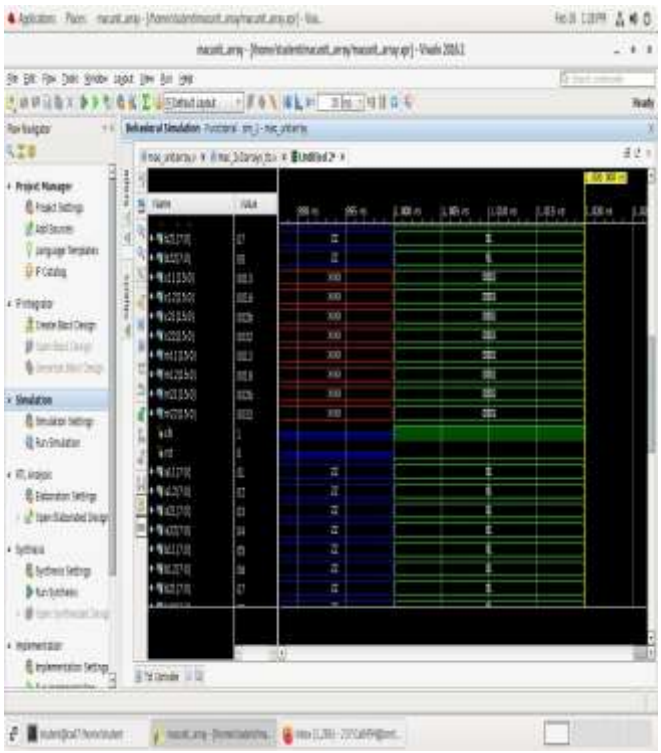
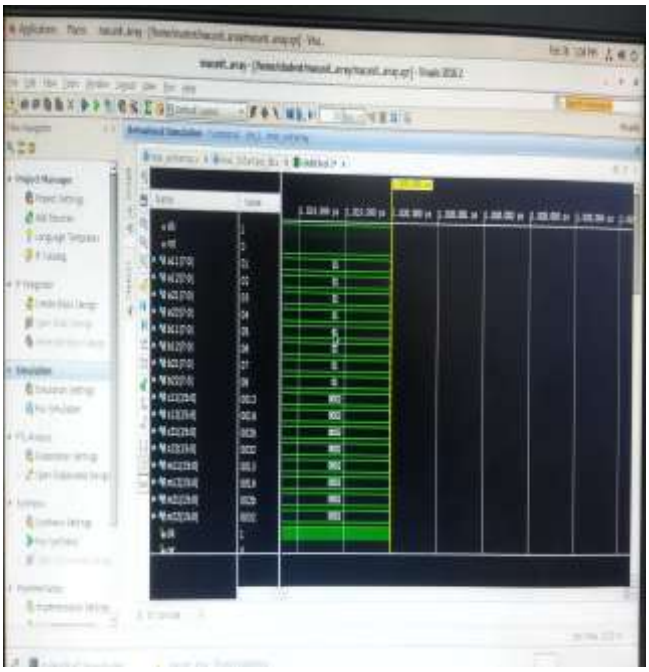


Figure 3: Simulation output



The fig 3, shows the final simulation output where all signals have valid values. The MAC units correctly perform multiplication and accumulation, and outputs (c11, c12, c21, c22) are stable, verifying correct functionality of the design.

The MAC (Multiply-Accumulate) unit array was successfully designed and simulated using Xilinx Vivado to verify its functional correctness. During the initial stage of simulation, the waveform showed unknown ('X') and high impedance ('Z') states, which usually occur when inputs and registers are not initialized at the beginning of the simulation. After applying proper clock and reset signals, the system initialized correctly and all signal values became stable. This indicates that the reset logic and clock synchronization of the MAC array are functioning properly and the system starts operation in a controlled manner.

The simulation results confirmed that each MAC unit in the array correctly performs multiplication followed by accumulation

operations. In each clock cycle, the input values are multiplied and then added to the previously stored accumulated value. The outputs obtained from the MAC array, namely c11, c12, c21, and c22, match the expected mathematical results, which proves that the MAC units are performing accurate computations. The waveform also shows that the outputs are updated according to the clock signal, indicating proper synchronous operation and correct data flow between different MAC units in the array.

Overall, the design has been successfully verified at the simulation level and demonstrates reliable performance. The MAC unit array shows parallel processing capability, which is very important in applications such as digital signal processing, image processing, matrix multiplication, and AI hardware accelerators. In future work, the design can be implemented on FPGA hardware and further optimized to improve speed, reduce hardware area, and minimize power consumption. This would make the MAC unit array more efficient and suitable for real-time AI accelerator applications.

VI. REFERENCES

[1] Umadevi, S., Penumaka, P., Ram, C. K. & Devi, T. K. High performance MAC unit design with grouping and decompositionmultiplier and 18 T gate diffusion Input-Transmission gate adder. Circuits Syst. Signal. Process. 44 (4), 2830–2854 (2025).

[2] Niknia, F. et al. A Configurable Floating-Point Fused Multiply-Add Design with Mixed Precision for AI Accelerators. IEEETransactions on Circuits and Systems for Artificial Intelligence. (2025).

[3] Dabbagh, A., Karamimanesh, M., Hassanli, K. & Abiri, E. CD-MAC: mixed signal binary/ternary in-memory computingaccelerator for power-constrained MAC processing. J. Supercomputing. 81 (1), 190 (2025).

[4] R. Sun, Y. Ni, X. He, J. Zhao, and A. Zou, ONE-SA: Enabling nonlinear operations in systolic arrays for efficient and flexibleneural network inference, in Proc. 2024 Design, Automation & Test in Europe Conf. & Exhibition, Valencia, Spain, 2024, pp. 1–6.

[5] K. Anderson, Hardware for machine learning inference: CPUs, GPUs, TPUs, https://telnyx.com/resources/ hardware-for-machine-learning-inference-cpus gpus-tpus, 2023.

[6] M.Jiang,J.Wang,A.Eldebiky,X.Yin,C.Zhuo,I.C.Lin,andG.L.Zhang, "aware pruning for efficient neural networks, "in Design,Automation and Testin Europe(DATE),2024.

[7] J.Wang, B.Li, and G.L.Zhang, "Early exit with class exclusion for efficient in reference of neural networks," in InternationalConference on Artificial Intelligence Circuits and Systems (AICAS),2024.

[8] Kashif Inayat, Fahad Bin Muslim, Javed Iqbal, Syed Agha Hassnain Mohsan, Hend Khalid Alkahtani and Samih M. Mostafa:"Power-Intent Systolic Array Using Modified Parallel Multiplier for Machine Learning Acceleration. Sensors"2023.

[9] Inayat, K.; Chung, J. "Hybrid Accumulator Factored Systolic Array for Machine Learning Acceleration". IEEE Trans. VeryLarge-Scale Integration. (VLSI) Syst. 2022.

[10] Jeyakumar Ponraj, R. Jeya bharath, P. Veena, Tharumar Srihari, "High performance multiply accumulate unit by integratingbinary carry select adder and counter-based modular wall ace tree multiplier for embedding system", the VLSI Journal 23(2023).