DOI: 10.21917/ijct.2025.0534

A LOW POWER DYNAMIC BITWIDTH-ADAPTIVE MULTIPLY ACCUMULATE UNIT FOR TINYML ACCELERATORS

Shyam Perika[1], Boddu Ajay[2] and Sumanto Kar[3]

[1,2]Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies, India
[3]FOSSEE, Indian Institute of Technology Bombay, India

Abstract

With the increasing demand for the deployment of machine learning models on energy-efficient and low-latency devices, TinyML stands out as an efficient solution for enabling intelligence on edge-constrained devices. TinyML workloads often need energy efficient hardware resources for reliable deployment of Machine Learning models. Existing hardware often lacks efficient hardware resources and is unable to perform efficient computations. The Multiply Accumulate Unit (MAC) plays a key role in defining the energy efficiency of the edge-constrained TinyML hardware. To bridge the gap, this work presents a novel architecture: a low power dynamic bit width-adaptive multiply accumulate unit (8-bit) for TinyML Accelerators. This architecture introduces a dynamic, multi-precision, bit width adaptive computational capability, supporting mixed-precision modes such as 2 × 2, 2 × 4, 2 × 8, 4 × 4, 4 × 8 and 8 × 8 with signed × unsigned support, making it highly scalable for TinyML accelerators. In addition, zero-aware gating and clock gating are implemented by employing a shift-and-add-based multiplier enabling partial product elimination and hybrid carry lookahead adder (CLA) based accumulator enabling dynamic segment-wise activation targeting energy efficiency in TinyML Accelerators. Proposed architecture is simulated and verified on eSim EDA tool and synthesized on the technology node of 130 nm using Google SkyWater's SKY130 PDK and the open-source EDA toolchain OpenLANE. The proposed Multiply Accumulate Unit reduces power by 59.36%, 68.78%, 74% and 80% when compared to PS4MAC, state-of-the-art (SotA) mixed precision MAC, Synopsys Design Ware MAC (DW) and approximate MAC unit respectively. Compared to prior works, this work stands out as an efficient architecture leading to the growth of energy-efficient TinyML Accelerators.

Keywords:

TinyML Accelerators, Ultra-low Power, Dynamic bit width adaptive, MAC Architecture

1. INTRODUCTION

With the rapid evolution of Artificial Intelligence and Advanced Machine Learning models, there is a huge demand for intelligent devices especially in the field of Internet of Things (IoT) and embedded MCUs. This evolution resulted in the development of specialized hardware tailored for edge-constrained ML models. This specialized hardware are termed as TinyML (Tiny Machine Learning), which focuses on deploying lightweight ML models on energy-efficient hardware. On these specialized TinyML-aware hardware, the Machine Learning models are locally hosted, eliminating the need for cloud-based processing while targeting on-board computations.

While prioritizing energy efficiency, TinyML hardware needs to sacrifice performance but should remain within an acceptable operating frequency. Most of the TinyML-dedicated hardware operates in a range of 1 to 200 MHz. Prioritizing energy efficiency, there is a need for optimization at the level of the core building blocks. Around 99% of the computations in TinyML accelerators are performed by Multiply Accumulate (MAC) Units. For energy-efficient TinyML-dedicated hardware, the Multiply Accumulate Unit can be a bottleneck component.

Existing MAC units are well adapted with mixed-precision scalability with signed and unsigned support optimized for computational efficiency, but they need to be more efficient in terms of power consumption. A power-aware variable-precision MAC unit employing a Baugh-Wooley array multiplier [19] offers an efficient architecture addressing the need for battery-powered wireless sensors with a power reduction of 43% compared to a conventional power-aware scalable pipelined MAC unit. MAC units employing different types of adder/multiplier architectures such as Carry Save Adder and Array Multiplier (0.238 mW and 14.0829 ns) [2], Carry Select Adder and Vedic Multiplier (0.256 mW and 14.4358 ns) [2], and Carry Save Adder and Multiplier (0.253 mW and 14.1497 ns) [2] have succeeded in the design of low-latency and low-power hardware.

Low power approximate Multiply Accumulate units [6] have also resulted in a reduction of power consumption and area by 42.6% and 46.1%, respectively. As approximate computing is gaining attraction for its efficient architecture, it results in reduced computational accuracy. In addition to this, mixed-precision quantization techniques [9] have been widely considered for balancing computational efficiency and flexibility in quantized TinyML Neural Networks. Compared to conventional architectures, a mixed-precision scalable approach has proven to be efficient and is widely employed in the field of TinyML Accelerators. Although these architectures often fail when handling computationally heavy tasks, which often require an energy-efficient architecture. To address this issue, the proposed work proves to be efficient in handling such computationally heavy tasks with an energy-efficient architecture. By employing different Register Transfer Level (RTL) techniques tailored for ultra-low-power design, the proposed architecture results in a huge reduction in power consumption as well as leading to an energy-efficient design. The novel contributions towards the proposed work are summarized as below:

- A novel mixed-precision scalable, zero-aware, clock-gated, signed-aware, bit-width-adaptive MAC architecture is devised with dynamic precision width detection allowing efficient computation with minimal power consumption leading to ultra-low-power design and low-latency inference for TinyML workloads.
- The entire architecture is developed using Verilog hardware description language, simulated and verified through a mixed-signal approach [22] using the open-source EDA tool eSim, developed by FOSSEE (Free/Libre and Open Source Software for Education), IIT Bombay.

- The design is synthesized on a technology node of 130 nm (Google's SkyWater SKY130A PDK) using the open-source EDA toolchain OpenLANE. Experimental results show the ultra-low-power capabilities of the proposed architecture.

2. LOW POWER MAC ARCHITECTURE FOR TINYML WORKLOADS

The proposed work employs different Low Power Register Transfer Level (RTL) techniques in order to gain an efficient reduction in power consumption when compared to conventional MAC architecture.

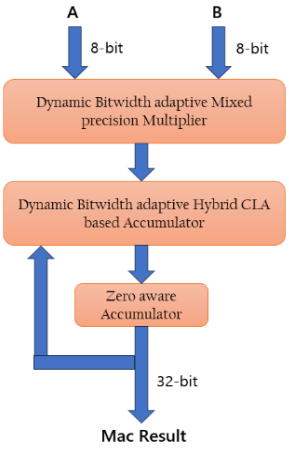

Fig.1. Proposed Multiply Accumulate Unit Architecture

The Fig.1. represents the proposed MAC architecture aligned with TinyML constraints. The proposed MAC unit operates as a precision-adaptive computational block, dynamically adjusting its behaviour based on the bit width of incoming operands. This flexibility is governed by a runtime-detected precision mode, defined as

$$M = f(B_I, B_W), \quad B_I, B_W \in \{2,4,8\} \tag{1}$$

Such dynamic reconfiguration enables the unit to support multiple operand pairings. At the system level, the core computation performed by the MAC unit in each cycle follows:

$$Y = \sum_{i=1}^{N} (I_i \cdot W_i) + A_{i-1} \tag{2}$$

where A_i and W_i are the activations and weights at cycle i, and A_{i-1} is the previously accumulated value. By detecting operand widths and activating only the required datapath segments, the architecture achieves substantial power savings. In further sections, we will explore various features of the proposed MAC architecture in relevance with the TinyML workloads.

2.1 DYNAMIC BIT-WIDTH ADAPTIVE MIXED PRECISION MULTIPLIER

The proposed MAC unit consists of a multiplier unit implemented using a shift-add based partial product accumulation algorithm. This avoids the use of conventional multipliers, leading to area and power reduction ideal for TinyML accelerators. The multiplier is designed in a structured reusable architecture consisting of a Sign detector, dynamic precision mode detector along with 2's complement transformation logic.

In the first stage of the data path, the signed inputs are detected and converted to unsigned format using 2's complement transformation logic. Let inputs A and B be 8-bit signed integers. Their unsigned equivalents A′, B′ are derived as:

$$A' = \begin{cases} \sim A + 1, & \text{if } A[7] = 1 \\ A, & \text{otherwise} \end{cases} \tag{3}$$

$$B' = \begin{cases} \sim B + 1, & \text{if } B[7] = 1 \\ B, & \text{otherwise} \end{cases} \tag{4}$$

The output sign $sign_p$ bit is computed separately using XOR of the sign bits.

$$\text{sign}_P = A[7] \oplus B[7] \tag{5}$$

The design supports true mixed-precision multiplication, including asymmetric combinations such as 2 × 4, 4 × 8, 2 × 8, 2 × 2, 4 × 4 and 8 × 8 by dynamically detecting operand widths and activating only the required logic. Bitwidth detection is performed after sign conversion. The effective mode of each operand is detected by checking upper zero bits. For operand $X \in \{A', B'\}$:

$$\text{mode}_X = \begin{cases} 2, & \text{if } X[7:2] = 6'b000000@4 \\ 4, & \text{if } X[7:4] = 4'b0000@8 \\ 8, & \text{otherwise} \end{cases} \tag{6}$$

To simplify computations, the multiplier handles the operations in unsigned format only, avoiding redundant hardware and simplifying integration. Moreover, based on the dynamic precision mode detector output, the multiplier adapts to operand widths (2-bit, 4-bit, or 8-bit) during runtime, allowing efficient computation with minimal switching. Logical slicing of operands into 2-bit, 4-bit, and 8-bit sections enables a highly structured and reusable design ideal for TinyML inference with varying precision needs.

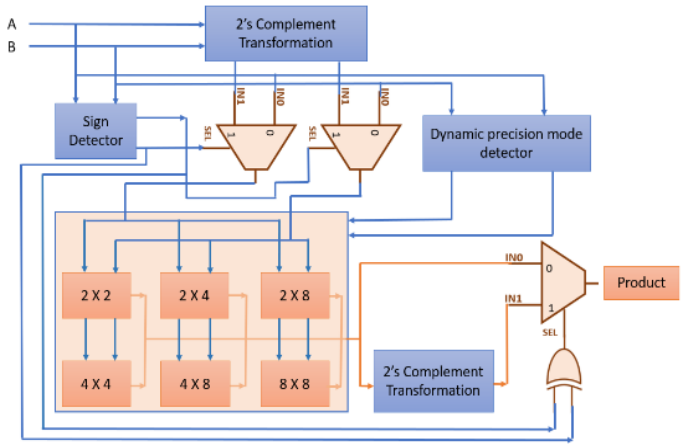

Fig.2. Dynamic Bitwidth adaptive Mixed precision Multiplier Architecture

The Fig.2 represents the proposed Multiplier architecture tailored for Ultra-low power and low latency applications. This scalable architecture enables the multiplier to handle heavy computational tasks, with efficient power consumption relevant to TinyML workloads. Along with these features, partial products are generated for active bits of input, eliminating unnecessary

computations. To facilitate hardware-level partial product generation, the operand A' is expanded into its binary representation, which acts as the control input for partial product generation. The operand is mathematically expressed as:

$$A' = \sum_{i=0}^{B_A-1} a_i \cdot 2^i \tag{7}$$

where $B_A \in \{2,4,8\}$ is the effective bit width of A', and a_i is the i^{th} bit of A'. Each bit a_i determines whether a shifted version of the multiplicand B' is included in the accumulation. The partial products are conditionally generated as:

$$PP_i = \begin{cases} B' \ll i, & \text{if } a_i = 1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The unsigned result is computed by summing all such partial products:

$$P_{\text{unsigned}} = \sum_{i=0}^{B_A-1} PP_i \tag{9}$$

This formulation reveals the explicit control dependency on A' where the bit pattern of A' determines which instances of $B' \ll i$ are activated and accumulated. This mechanism is efficiently implemented in the data path using Verilog. This partial product elimination plays a key role in power reduction especially when input sparsity is high. Furthermore, this multiplier architecture is zero-aware, enabling us to eliminate unnecessary switching when input is zero.

$$A' = 0 \quad \text{or} \quad B' = 0 \Rightarrow P = 0 \tag{10}$$

Finally, sign correction is performed on the accumulated unsigned result:

$$P = \begin{cases} \sim P_{\text{unsigned}} + 1, & \text{if sign}_P = 1 \\ P_{\text{unsigned}}, & \text{otherwise} \end{cases} \tag{11}$$

This architecture effectively eliminates unnecessary switching, supports operand sparsity, and adapts to runtime precision, making it highly energy-efficient and scalable. This novel scalable architecture, combining different low-power RTL techniques, makes it highly efficient for TinyML aware applications.

2.2 ADAPTIVE ZERO AWARE CLA BASED ACCUMULATOR

In conventional MAC units, several adder architectures are employed like Carry Save Adder, Carry Select Adder, Carry skip Adder [2] etc., and have gained an efficient power reduction. However, with the new era of intelligent hardware systems, there is a need for more energy-efficient architecture to be developed to gain an efficient trade-off between power and delay. In the proposed work, A power optimized, dynamic bit width adaptive, Hybrid CLA based Adder-Accumulator is integrated, with several Carry lookahead adders connected in a ripple carry configuration. Since the output of the multiplier is 16-bit, adder-accumulator block is designed with a 32-bit configuration to prevent overflow and ensure safe accumulation. Prior to accumulation, the 16-bit signed multiplier output M is sign-extended to 32 bits to preserve signed arithmetic during accumulation:

$$\text{Mult}_{\text{out}} = \begin{cases} \{16'b1111\ldots1, M\}, & \text{if } M[15] = 1 \\ \{16'b0000\ldots0, M\}, & \text{if } M[15] = 0 \end{cases} \tag{12}$$

The Fig.3 represents the proposed architecture for Adder accumulator in which each CLA block is of 4-bit configuration.

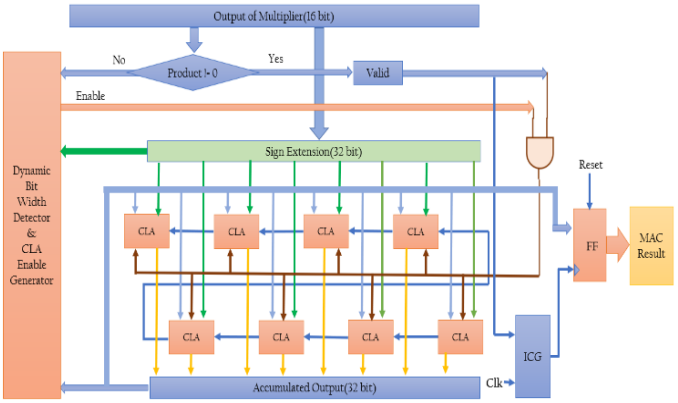

Fig.3. Dynamic bit-width adaptive Hybrid CLA based Accumulator Architecture

Since adder-accumulator is of 32-bit configuration, total 8 CLA blocks are connected in ripple carry configuration. The bit-widths of the current accumulator value A_{i-1} and the sign-extended multiplier output are compared to derive the required operational width W:

$$W = \max(W_A, W_M) \tag{12}$$

where W_A and W_M are operational widths of Accumulated output and Multiplier output respectively. Based on the highest precision mode(4-bit/8-bit/12-bit/16-bit/20-bit/24-bit/28-bit/32-bit) detected, enable signals for eight CLA blocks are generated to activate necessary CLA blocks, eliminating the unnecessary switching leading to reduction in dynamic power consumption.

$$\text{CLA}_{\text{enable}} = \begin{cases} 8'b00000001, & W = 4 \\ 8'b00000011, & W = 8 \\ 8'b00000111, & W = 12 \\ \quad\ldots & \\ 8'b11111111, & W = 32 \end{cases} \tag{13}$$

Each CLA block CLK_k receives an enable signal $e_k \in \{1,0\}$. If disabled, the block output is either all 1s or 0 s based on the sign of the result. The sign of the accumulation is computed from the MSB of the last active CLA block:

$$\text{sign} = S_{k_{\max}}[3] \tag{14}$$

where e_k=1 and k_{max}=max(k).

Each CLA block computes an output sum_k based on e_k and sign of the accumulated output as shown below:

$$\text{sum}_k = \begin{cases} \text{CLA}(A_k, M_k), & \text{if } e_k = 1 \\ 4'b0000, & \text{if } e_k = 0 \,\&\, \text{sign} = 0 \\ 4'b1111, & \text{if } e_k = 0 \,\&\, \text{sign} = 1 \end{cases} \tag{15}$$

where A_k and M_k represent the k^{th} 4-bit segments of the accumulator and multiplier output respectively, e_k is the corresponding enable signal, and sign represents the MSB of the highest active CLA output segment. Each CLA block internally

incorporates the carry propagate and carry generate logic as shown below:

$$p_i = a_i \oplus b_i, \quad g_i = a_i \cdot b_i \tag{16}$$

where a_i and b_i are the input bits. These Carry propagate (p_i) and generate (g_i) terms are used for the calculation of sum and carry results as shown below:

$$\text{sum}_i = p_i \oplus c_i, \quad C_{i-1} = g_i + p_i \cdot c_i \tag{17}$$

where s_i is the sum bit and $C_{i\text{-}1}$ is the carry-out to the previous stage.

In addition, zero-aware adaptive signed accumulation is implemented, enabling accumulation, only when non-zero inputs are fed. If any one of the inputs to the accumulator is zero, it results in zero output eliminating unnecessary switching efficient for low power design.

$$A_i = \begin{cases} \text{Sum}, & \text{If } M \neq 0 \\ A_{i-1}, & \text{otherwise} \end{cases} \tag{18}$$

where, A_i is the current accumulated output, M is the multiplier output, and $A_{i\text{-}1}$ is the previous accumulated output. In order to make the proposed architecture, an Ultra-low power design, Integrated Clock gating (ICG) cell is integrated, to disable the clock signal whenever the product output is zero. All these RTL low power techniques are integrated in order to improve the computational capability and energy efficiency of the MAC unit tailored for TinyML edge constrained devices.

2.3 ZERO AWARE CLOCK GATING

While implementing a low-power design, conventional approaches often result in a huge reduction in power consumption.

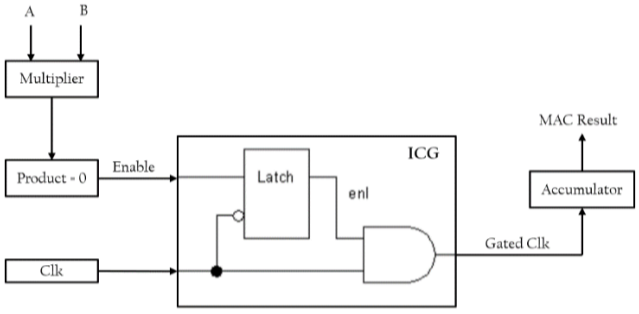

Fig.4. Integrated Clock Gated Cell

One of the exceptional techniques leading to Ultra-low power design is Clock Gating. Clock gating refers to deactivation of clock signal which is considered as a critical signal in low power designs, for prevention of unnecessary transitions, where active computation is not requires, dramatically, results in reduction of dynamic power consumption. The Fig.4 represents a clock-gated cell used to implement clock gated architecture, in order to eliminate metastability issues and signal glitches. In the proposed MAC architecture, clock signal is gated whenever the multiplier output is zero. At that time, as accumulation is not required i.e., is not an active computation, it prevents the unnecessary toggling of the clock signal resulting in reduction of dynamic power consumption. The gated clock maintains synchronism with the input clock signal eliminating signal glitches.

2.4 TINYML CENTRIC HARDWARE DESIGN

TinyML hardware is often powered by batteries, so energy efficiency is very important. The proposed architecture uses an Adaptive Shift-Add logic-based Multiplier and an Adaptive Segment-wise Activated Hybrid CLA-based Accumulator. This combination plays a key role in developing hardware for TinyML. Thanks to advancements in technology, this architecture stands out for its ultra-low power consumption and high energy efficiency. By integrating multiple techniques into a single MAC (Multiply-Accumulate) unit, the design can handle complex computational tasks typically found in TinyML applications. To further enhance suitability for edge deployment, the architecture emphasizes minimal silicon footprint and low-leakage design strategies. This makes it highly compatible with compact, resource-constrained embedded systems.

3. ARCHITECTURAL EVALUATION

3.1 MIXED SIGNAL SIMULATION USING ESIM

After the architecture level development, the proposed architecture is simulated using eSim, through a mixed signal approach. eSim is an open-source tool tailored for electronic design simulation and verification with features such as circuit design, PCB design, Mixed signal simulations, analog and digital simulations, Register Transfer level (RTL) design simulations using Hardware description languages (HDL) like Verilog, System Verilog, VHDL and TL-Verilog and Device modelling.

For simulation, the RTL code is converted into a NgVeri (NgSpice + Verilator) model using NgVeri model editor of eSim. Model editor is capable of converting the Hardware description language into a digital Model that can be integrated into a mixed signal circuit. The NgVeri model is interfaced with ADC and DAC bridges available in eSim component library in order to design a mixed signal circuit. The circuit is designed using KiCad, a tool responsible for circuit design in eSim.

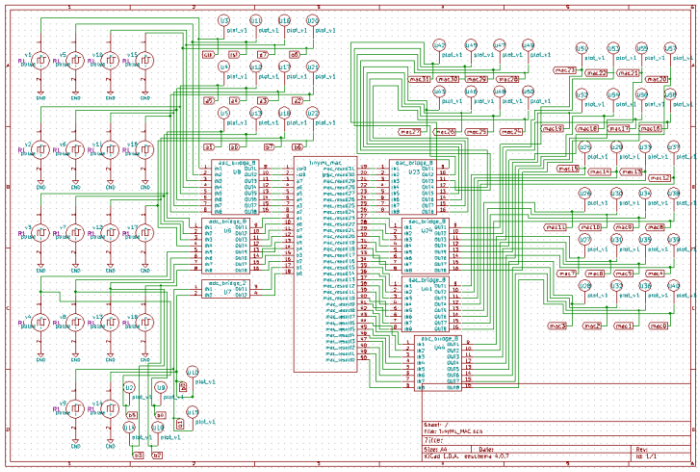

Fig.5. Schematic design of mixed signal circuit for MAC Unit in eSim

The Fig.5 represents the Schematic design in eSim in order to simulate and verify the NgVeri model of MAC unit. Further steps include conversion of KiCad file into Spice file through KiCad to Spice converter in eSim. Fig.6 shows the simulation results of

NgSpice, through which the proposed MAC architecture is successfully verified. The use of eSim seamlessly verifies the functional behaviour of the MAC architecture, which also lets the proposed work contribute towards the open-source development.

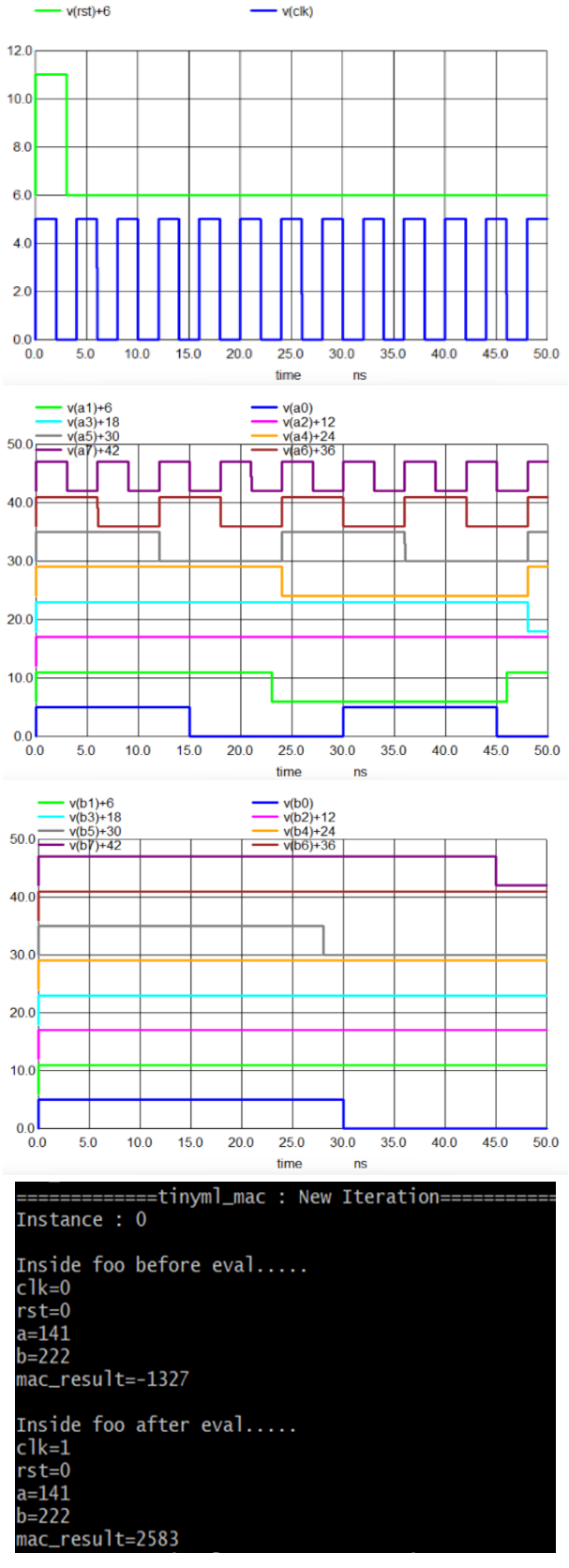

Fig.6. eSim Simulation Results for MAC Unit

This not only validates the design methodology but also demonstrates the capability of open-source tools in enabling sophisticated digital and mixed-signal hardware design, reducing dependency on costly commercial software, and fostering collaborative innovation. Building upon the successful functional verification using eSim, the proposed work progresses towards hardware realization using an ASIC design flow to demonstrate real-time operability of the proposed MAC unit.

3.2 HARDWARE EVALUATION USING OPENLANE EDA TOOLCHAIN

The proposed architecture is fully synthesized using the Open-source EDA toolchain OpenLANE, based on the 130nm technology node from Google's SkyWater SKY130A Process Design Kit. The OpenLANE toolchain includes several EDA tools, each serving different purposes in the design process. Additionally, the design is simulated using Icarus Verilog, an open-source RTL simulation tool.

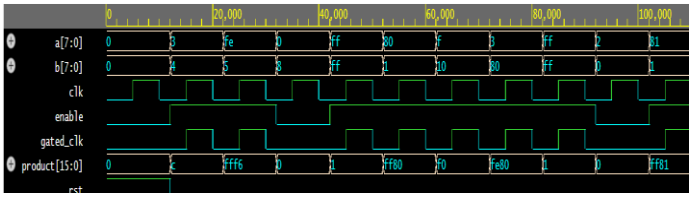

Fig.7. Simulation results for MAC Unit from Icarus Verilog

The Fig.7 shows the simulation results obtained using Icarus Verilog. The design is fully synthesized using Yosys, an open-source synthesis tool integrated within the OpenLANE toolchain. For the experimental results, synthesis uses the sky130_fd_sc_hd library, which is a high-density standard cell library. The synthesis strategy is set to 'DELAY 2', a high-effort in OpenLANE focused on delay optimization. This helps achieve optimal delay, making the design suitable for low-latency devices. The experimental results for timing and power analysis are summarized in the table below.

Table.1. Power ad Timing analysis results for MAC Unit

Parameter Description	Value	Unit
Internal Power (54.4%)	45.7	µW
Switching Power (40.3%)	33.9	µW
Leakage Power (5.2%)	4.41	µW
Total Power (100%)	84	µW
Worst Negative Slack (WNS)	0	ns
Total Negative Slack (TNS)	0	ns
Energy per Operation	3.5	pJ
Energy Efficiency	0.286	TOPS/W
Throughput	24	MOPS

Note: Above results are taken at an operating frequency of 24MHz.

The Table.1. shows results of post-synthesis power and timing analysis. In addition to this, Fig.8 shows the results for Combinational and Sequential power analysis performed using Yosys. These results show that due to the complexity of the architecture in order to design a computational efficient hardware,

resulted in a more combinational power, completely opposite to that of the conventional cases, in which sequential power always dominates the combinational power. The proposed work shows a throughput and energy efficiency of 24 MOPS and 286 GOPS/W respectively, which is very efficient in terms of performance and energy efficiency compared to conventional MAC units tailored for TinyML accelerators. From these results, we can conclude that, the proposed Multiply Accumulate Unit is an efficient Ultra Low power architecture tailored for TinyML workloads.

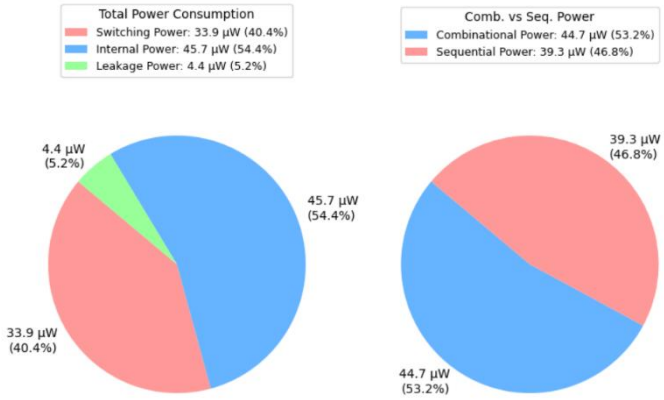

Fig.8. OpenSTA power analysis results for MAC Unit

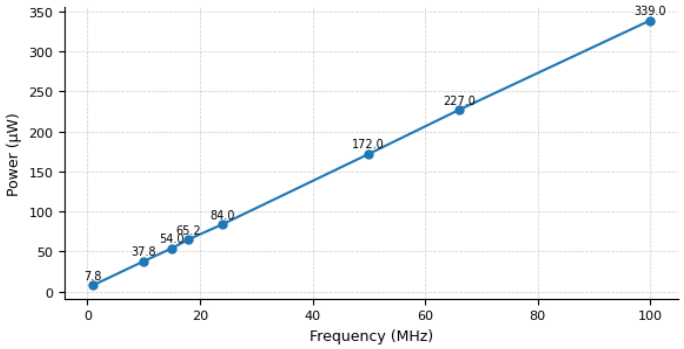

Fig.9. Frequency (MHz) vs Power (µW) analysis for MAC unit

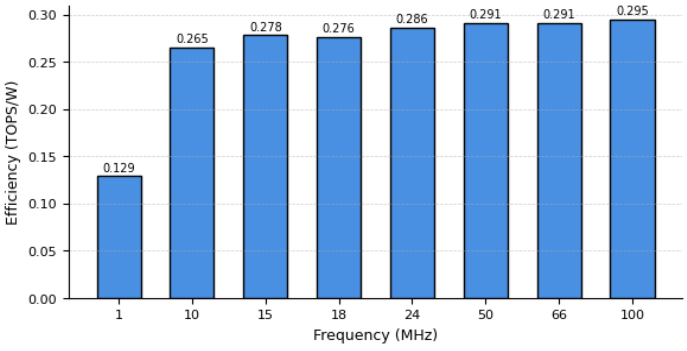

Fig.10. Frequency (MHz) vs Efficiency (TOPS/W) analysis for MAC unit

The Fig.9 and Fig.10 demonstrates the power and energy efficiency analysis of the proposed TinyML MAC unit across a range of operating frequencies. As TinyML accelerators mostly operates over a range of 1 to 100 MHz, the analysis is performed over this range to show the trade-off between power efficiency and a range of operating frequencies. Meanwhile, energy efficiency (TOPS/W) improves with frequency up to around 50 MHz, after which it saturates, indicating optimal operating regions for low-power inference workloads. These results validate the MAC unit's suitability for energy-constrained TinyML applications.

At lower frequencies (<10 MHz), static power dominates, reducing overall energy efficiency, while at higher frequencies (>70 MHz), dynamic switching and clock tree power become significant contributors. The flat energy efficiency curve beyond 50 MHz suggests that further increasing the frequency yields diminishing returns in performance-per-watt. This insight is particularly important for always-on and battery-operated inference systems, where energy proportionality plays a key role. Hence, the proposed MAC unit achieves a favorable balance between latency, power, and energy efficiency across typical TinyML operating conditions.

4. COMPARATIVE ANALYSIS

Numerous MAC architectures have been proposed and implemented in existing TinyML accelerators. A comparative analysis of these architectures in terms of power and energy efficiency is essential for informed design choices. Prior work in this domain often incorporates techniques such as approximate computing, mixed-precision scalability, and zero aware adaptiveness. While these approaches are well-suited for TinyML applications, due to their resource efficiency, they often lack in delivering optimal computational energy efficiency.

The Table.2. demonstrates the comparative analysis between the proposed work and other MAC architectures [6], [9], [20] in terms of power and energy efficiency. In comparison, the proposed work stands out as an optimized architecture with a power consumption and energy efficiency of 84uW and 286 GOPS/W respectively, making it efficient in powering TinyML accelerators

Table.2. Power Consumption of various MAC Architectures

Approximate [6]	DW [20]	SOTA [9]	PS4MAC [9]	Proposed
420	325	269	206.7	84

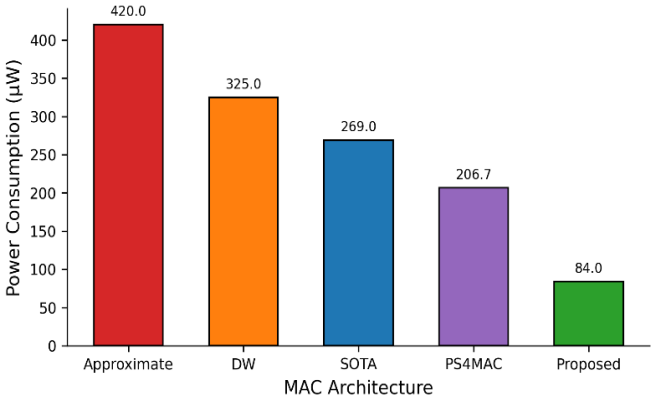

Fig.11. Comparative analysis between various MAC architectures in terms of Power Consumption

5. CONCLUSION

Aiming at the TinyML workloads, this paper proposes a power optimized mixed precision scalable, Zero-aware, clock gated, signed aware, bit-width adaptive MAC architecture devised with dynamic precision width detection allowing efficient computations with minimal power consumption leading to ultra-low power design and low latency inference for TinyML workloads. The proposed work achieves a power reduction of 68.78% when compared to State-of-the-art mixed precision MAC units, making it suitable for ideal for energy efficient TinyML Accelerators. With an energy efficiency and throughput of 0.286 TOPS/W and 24 MOPS respectively, the proposed work stands out as an exceptional choice for energy efficient hardware accelerators. The proposed work is evaluated and verified using open source tools eSim, and Open LANE EDA toolchain as a contribution towards Open source community. And finally, a comparative analysis between the proposed work and existing architectures is performed, from which we can conclude that the proposed MAC architecture is highly efficient for TinyML workloads.

ACKNOWLEDGEMENT

Grateful acknowledgment is extended to the FOSSEE project at IIT Bombay for providing the eSim tool used for circuit design and simulation in this study. This work also utilized the OpenLANE open-source digital ASIC design flow for synthesis. The design was implemented using the SKY130 process design kit (PDK), made available through the Open source initiative by Google in collaboration with SkyWater Technology.

REFERENCES

[1] Zhaoteng Meng, Lin Shu, Zhan Li, Kailin Lv, Chang Lin, Long Xiao and Jie Hao, "Optimizing Area and Power of MAC Arrays in DNN Accelerators via Overflow-Aware Partial Sum Management", *IEEE International Symposium on Circuits and Systems*, pp. 1-5, 2025.

[2] L.D. Teja, K. Shoneeth, C.H. Siri and T. Abhishek, "Design and Performance Analysis of Different Low-Power Multiply-Accumulate (MAC) Units", *Proceedings of International Conference on Communications and Information Technologies*, pp. 1-6, 2024.

[3] H. Zhang, D. Chen and S.B. Ko, "New Flexible Multiple-Precision Multiply-Accumulate Unit for Deep Neural Network Training and Inference", *IEEE Transactions on Computers*, Vol. 69, No. 1, pp. 26-38, 2020.

[4] U. Cini and O. Kurt, "A High Performance Multiply-Accumulate Unit with Double Carry-Save Scheme for 6-Input LUT based Reconfigurable Systems", *Proceedings of International Conference on Electrical and Electronics Engineering*, pp. 940-944, 2015.

[5] V. Camus, C. Enz and M. Verhelst, "Survey of Precision-Scalable Multiply-Accumulate Units for Neural-Network Processing", *Proceedings of International Conference on Artificial Intelligence Circuits and Systems*, pp. 57-61, 2019.

[6] T. Yang, T. Sato and T. Ukezono, "An Approximate Multiply-Accumulate Unit with Low Power and Reduced Area", *Proceedings of International Symposium on VLSI*, pp. 385-390, 2019.

[7] N. Arya, A.K. Rajput and M. Pattanaik, "Energy and Area Efficient 16-Bit Approximate Multiply-Accumulate (EAMAC) Architecture for Error Tolerant Applications", *Proceedings of International Conference on Intelligent Signal Processing and Effective Communication Technologies*, pp. 1-6, 2024.

[8] M.E. Sriram and S.R. Ramesh, "Pipelined and Low-Power MAC Unit with Sparsity-Aware Encoding for Neural Network Applications", *Proceedings of International Conference on Communication Systems and Network Technologies*, pp. 1210-1215, 2025.

[9] X. Hu, X. Geng, Z. Mao, J. Han and H. Jiang, "A Low-Power Mixed Precision Integrated Multiply-Accumulate Architecture for Quantized Deep Neural Networks", *Proceedings of International Conference on Design, Automation and Test*, pp. 1-7, 2025.

[10] S.J. Jou, C.Y. Chen, E.C. Yang and C.C. Su, "A Pipelined Multiplier-Accumulator using a High Speed, Low Power Static and Dynamic Full Adder Design", *IEEE Custom Integrated Circuit Conference*, pp. 593-596, 1995.

[11] A.P. Chandrakasan, S. Sheng and R.W. Brodersen, "Low-Power CMOS Digital Design", *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 4, pp. 473-483, 1992.

[12] N.H.E. Weste and D. Harris, "*CMOS VLSI Design: A Circuits and Systems Perspective*", Addison-Wesley, 2005.

[13] S.J. Jou, C.Y. Chen, E.C. Yang and C.C. Su, "A Pipelined Multiplier-Accumulator using a High Speed, Low Power Static and Dynamic Full Adder Design", *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 1, pp. 114-118, 1997.

[14] M. Suzuki, N. Ohkubo, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki and Y. Nakagome, "A 1.5ns 32-Bit CMOS ALU in Double Pass-Transistor Logic", *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 11, pp. 1145-1151, 1993.

[15] F. Lu and H. Samulei, "A 200-MHz CMOS Pipelined Multiplier-Accumulator using a Quasi-Domino Dynamic Full Adder Cell Design", *IEEE Journal of Solid-State Circuits*, Vol. 28, pp. 123-132, 1993.

[16] P.C. Anantha, S. Samuel and R.W. Borderson, "Low Power CMOS Digital Design", *IEEE Journal of Solid-State Circuits*, Vol. 27, pp. 473-483, 1992.

[17] Anu, P. Chaudhary and P.K. Dahiya, "Techniques for the Design of High Speed and Low Power MAC Unit: A State-of-the-Art Review", *International Journal of Computer Applications*, Vol. 148, No. 13, pp. 22-25, 2016.

[18] S. Shanthala and S.Y. Kulkarni, "VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique", *European Journal of Scientific Research*, Vol. 30, No. 4, pp. 620-630, 2009.

[19] K.S.N. Yengade and P.R. Indurkar, "Review on Design of Low Power Multiply and Accumulate Unit using Baugh-Wooley based Multiplier", *International Research Journal of Engineering and Technology*, Vol. 4, No. 2, pp. 638-642, 2017.

[20] Synopsys, "DesignWare DW02_Multiplier", Available at https://www.synopsys.com/dw/ipdir.php?c=DW02_mult, Accessed in 2024.

[21] R. Paknikar, S. Bansode, G. Nandihal, M.P. Desai, K.M. Moudgalya and A. Jha, "eSim: An Open Source EDA Tool for Mixed-Signal and Microcontroller Simulations", *Proceedings of International Conference on Circuits, Systems and Simulation*, pp. 212-217, 2024.

[22] S. Kar, R. Paknikar, D. Singh, K.M. Moudgalya, S. Hoover, S. Nalinkumar, S. Charaan, R.V. Rohinth Ram, M.H. Kadam and K. Ghosh, "Mixed Signal Simulation Marathon for Education and Employment", *Proceedings of International Conference on Electrical, Computer, Communications and Mechatronics Engineering*, pp. 1-6, 2022.