

Implementation and Performance Analysis of MAC Architecture using Fastest Multipliers for DSP Applications

Dr. Madhavi Mallam¹, Rachana M², Preetham Gowda N M³, Keerthana R⁴, Samrud H S⁵

^{1,2,3,4,5}Dept. of Electronics and Communication Engineering, Global Academy of Technology, Bengaluru, India

Abstract— In this paper, an approach is made towards the design and implementation of the Multiplier and Accumulator (MAC) unit using various fastest multipliers which serves as a Digital Signal Processing (DSP) core with a capability of energy-optimized architecture and with superior accuracy. The central element in most systems of signal processing is the multiplier; it is the fundamental arithmetic element that significantly affects the performance as a whole. In this work we proposed an energy efficient design of one such multiplier named as new Rounding-Based Approximate (ROBA) multiplier for the MAC unit. The ROBA multiplier operates by rounding the input operands to the nearest power of two, allowing the multiplication execution to be based on the employment of relatively few adders and shifters. This technique greatly simplifies implementation complexity while elevating energy efficiency by a substantial amount. To identify its efficiency, we carried out a comparative simulation using several multipliers, i.e., Wallace, and Vedic multipliers. On careful investigation of the performance with respective to area, power dissipation, and gate complexity, the best-performing multiplier is chosen based on the simulation result, and a Multiply-Accumulate (MAC) unit is designed and incorporated using the Cadence software. Finally, the design and implementation of the MAC unit with the chosen multiplier were accomplished and finalized, with the chip layout using Cadence. The ROBA multiplier-based MAC unit shows 78% of power saving and 79% of area saving when compared to Vedic and Wallace based Architectures.

Keyword— MAC Unit, ROBA Multiplier, Vedic, Wallace Multiplier, RCA, Accumulator.

I. INTRODUCTION

Very Large-Scale Integrated Circuits (VLSI) involve the

design of detailed integrated circuits (IC). The integrated circuit technology transformed chip design by enabling thousands and millions of transistors in chip [1]. Owing to the increasing complexity in digital systems, the use of HDL in the design of Very Large Scale Integrated (VLSI) based subsystem has increased in importance.

Recent computing architectures exhibit a wide number of multipliers which are achieved by (a) smaller number of partial products and their implementation complexity [5], (b) using exact compressor for row accumulation of the partial products in order to minimize latency and power [6], (c) employing approximate adder in the final step for adding the last two rows output by the compressor [1], and (d) dividing the operand into two parts and the computing accurate and approximate multiplication using accurate and approximate logic respectively [6]. Using the last technique, an Error Tolerant Multiplier (ETM) that divides the operand in half into two sections (significant-segment has very few Most-Significant Bits (MSBs) while non-sign segment containing the remaining Least Significant Bits (LSBs) is presented in [6]. The ETM carries out partial product with less multiplication for the non-significant segment and accurate multiplication for the significant segment. An improved approximate multiplier where non-significant segment is computed by using bitwise AND-OR gates to reduce computational error and it is given in reference [5].

II. LITERATURE REVIEW

The synthesis and implementation of an 8-bit Multiply-Accumulate (MAC) Unit with some adder structures, i.e., Kogge-Stone, Ladner-Fischer, Carry

Look-Ahead, and Ripple Carry Adders, in Verilog HDL and simulation programs like Xilinx ISE and Model Sim, enable high-speed arithmetic function optimized in the DSP application field [1]. The architecture can be incorporated in the project in order to perform efficient calculation of energy-based information, such as, for instance, power fluctuations or sensor inputs. In supplementing that, an approximate MAC unprovided on dynamically configurable 4-2 compressors and partial product truncation provides a trade-off in between accuracy and power efficiency, making it suitable in the context of low-power, error-tolerant environments [2]. The incorporation of the dynamic approach of approximation in the present system can enable optimization in energy management decisions in the scenario where a certain loss of accuracy is tolerable [1].

Multippliers are fundamental building blocks for VLSI-based digital signal processing systems, for which power efficiency, area optimization, and high-speed execution are critical for overall system performance [3]. A very accurate multiplier integrated with a Multiply-Accumulate (MAC) unit consisting of an adder, multiplier, and accumulator—boosts processing speed and cuts resources. Multiplication techniques, namely the Vedic, Booth, and Egyptian approaches, are investigated and compared for the best delay, area, and power performance. The design for the proposed MAC is simulated and explored with Xilinx 14.5, with an eye to performance optimization [3]. To obtain further energy efficiency enhancement for signal processing cores, novel rounding-based approximate (RBA) multipliers were recently introduced [4]. The multipliers reduce computation with the help of input value, rounding to the next power of two simple adders and shifters, greatly simplifying hardware. The 8-bit model for the newly proposed RBA0, for instance, achieves nearly 60% area reduction and more than 50% delay improvement compared to standard architecture. A further runtime reconfigurable version (RRBA) enables accuracy versus performance trade-offs at runtime, demonstrating practical potential for real-world applications including Gaussian filtering with significant energy saving [4].

III. DESIGN AND IMPLEMENTATION

To contrast performance on key parameters—power consumption, area, and gate count—the several multipliers is implemented and tested, i.e., Vedic, Wallace, and ROBA architectures. Upon detailed comparison, we found that the ROBA multiplier always outperformed the others in every respect. Specifically, it required 17 fewer gates than Vedic, and Wallace multipliers, and saved area by 79%. Based on these results, we selected the ROBA multiplier to be the basis for the design of an energy-efficient MAC (Multiply-Accumulate) unit.

A. ROBA Based MAC Unit

The main intention of this design is to develop a hardware unit that fulfills the Multiply-Accumulate operation ($\text{Result} = \text{Previous Result} + A \times B$) with low power consumption and minimal hardware area (gate count) while optimizing it. This optimization occurs through deliberately sacrificing certain accuracy in the multiplication step through the usage of approximation methods. The design comprises a top-level MAC module which employs three central sub-components: a user-defined Approximate Multiplier, a regular RCA (Ripple Carry Adder), and a Register (Accumulator). Top-Level MAC module manages overall functioning. Inputs: It is supplied with an 8-bit multiplicand (A), 8-bit multiplier (B), four single-bit truncation control signals (Trunc0 to Trunc3), a clock signal (clk), as well as a reset signal (rst). It gives a 16-bit accumulated result. The Approximate Multiplier module (inst1) accepts A, B, as well as the Truncation signals so as to calculate a 16-bit approximate Product. The RCA module (inst2) accepts such a 16-bit Product as well as adding it to the result that was accumulated earlier (in this case, the then-existing value of Mac out is fed back as part of the adder's B input). The carry-in is 0. The output is a 16-bit addend result (Add out). The Register module (inst3) latches up the value of Add out on clk's falling edge. If the rst signal is asserted (high within this particular register code), then register output Dout (which is also connected with Mac out) is set to zero; otherwise, it inherits that of Add out. The value thus stored is the new Mac out. The basic operation considers this equation:

$$\text{Macout}[t+1] = (\text{rst})? 0: (\text{Macout}[t] + (A \times B)\text{approx.})$$

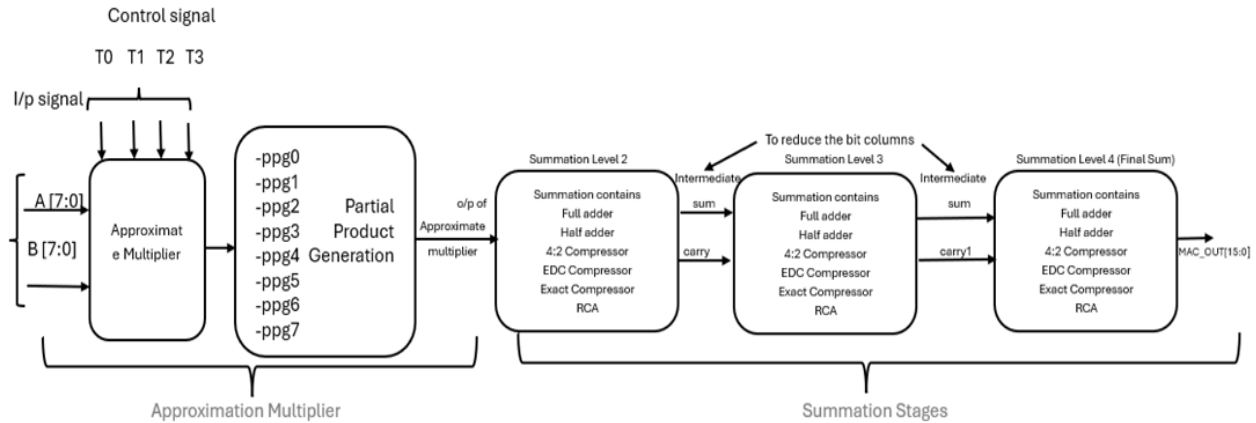


Figure 1. Block Diagram of Approximation based MAC unit

The figure1 illustrate the block diagram of approximation-based MAC unit and it contains two part one is Approximation Multiplier and another one is Summation stages as shown in figure 1 and each one is explained in detail in below figure 2 and figure 3 respectively.

1) Approximation Multiplier:

This module is the essence of the novelty of the design, applying the approximate multiplication. Regular multipliers produce a matrix of partial products (each being $A_i \wedge B_j$) and add them in a complicated adder tree. This design alters both stages of generation and summation.

i) Partial Product Generation Stage:

In this stage, each partial product generation module calculates the individual partial products ($P_{ij} = A_i \wedge B_j$).

In Approximation Method we use a Dynamic Partial Product Truncation method. The main innovation here is to use of the Truncation0 to Truncation3 input signals. Within the pp_generation and ppg modules, the associated Truncation signal is initially inverted with a not gate. This inverted signal is subsequently ANDed with the multiplier bit. Then this is ANDed with the multiplicand bit(s) to generate the partial product. The correct logic to produce a partial product PPD driven by a Truncation signal is

$$PPD_{ij} = (A_i \wedge B_j) \wedge (\neg \text{Truncation}_k)$$

When a particular Truncation signal (e.g., Truncation0) is asserted (set to logic '1'), the inverted signal is '0'. This directly makes the output (PPD) of all pp_generation or ppg instances controlled by that

Truncation signal equal to '0', which truncates or eliminates those partial products from the computation. This dynamically lowers the count of bits to sum in the subsequent stage, reducing hardware. The particular mappings as shown in Figure1, which indicate which Truncation signal drives which collection of partial products, probably ordered from least significant (Truncation0) to more significant (Truncation3).

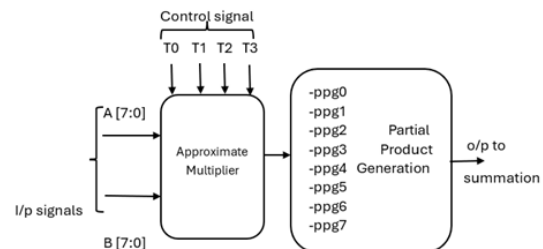


Figure 2. Block Diagram of Approximate Multiplier.

Where T0, T1, T2 and T4 are the four truncation control signal as shown in above Figure2 and ppg0 to ppg7 are the eight partial product generated signal as shown in above Figure2 and some of ppg's are made zero based which truncation signal is logic high.

2) Summation Stage (Compression):

Sum of all the produced partial products (including those that were zeroed out) to form the ultimate 16-bit Product. The sum is done through various levels (Level 2, Level 3, Level 4) with a mix of various adder and compressor blocks as shown in figure 3. In Approximation Technique, we use Approximate Compressors, This design includes the use of specially designed approximate compressors rather than just

using precise adders. Compressors are adder modules that accept several input bits (say, 4 bits) and "compress" them into less output bits (say, a sum and two carries having the same overall value).

Approximate Compressors (4:2 Compressor): This module accepts 4 input bits (x_1 to x_4) and generates a sum and a carry output. Its internal logic employs a

mixture of AND (&) and OR (|) gates, which is less complex (fewer transistors) than the XOR/AND logic in exact compressors but that leads to computational errors. The approximate logic can be as follows:

Intermediate signals:

$$w_1 = x_1 \& x_2, w_2 = x_1 | x_2, w_3 = x_3 \& x_4, w_4 = x_3 | x_4, \\ w_5 = x_1 | x_3, w_6 = x_2 \& x_4$$

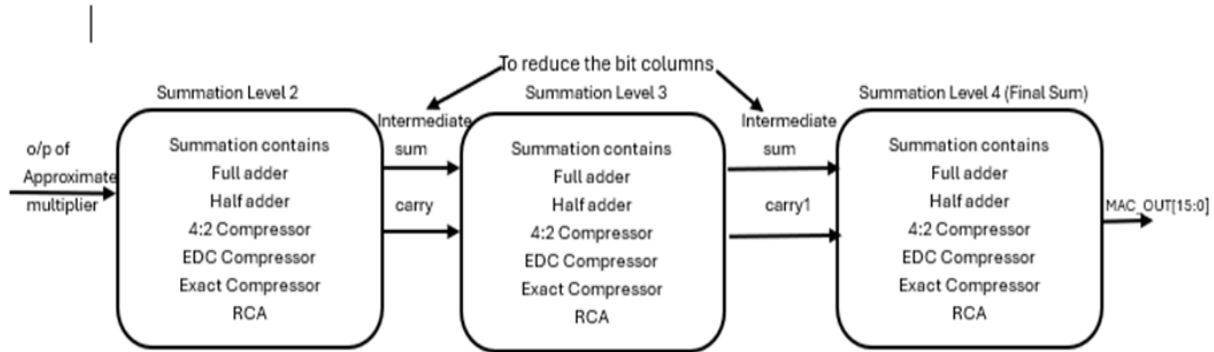


Figure 3. Block Diagram of Summation Levels

EDC_compressor: Yet another 4-input approximate compressor architecture. It employs another logic structure using XOR and multiplexer-like functionality as a function of the first two inputs' XOR to choose between ORing or ANDing the last two inputs for the SUM output. The CARRY is just the OR of the first two inputs. This probably presents another error characteristic than approximate 4:2 compressor.

$$\text{SUM} = (A_1 \oplus A_2) ? (A_3 \wedge A_4) : (A_3 \vee A_4) \\ \text{CARRY} = (A_1 \vee A_2)$$

The sum stage also utilizes standard accurate building blocks:

Half Adder: Adds two bits, generating a sum and a carry.

$$\text{Sum} = (a \oplus b), \text{Carry} = (a \wedge b)$$

Full Adder: Adds three bits (a , b , carry-in), outputting a sum and a carry-out.

$$\text{Sum} = (a \oplus b \oplus c), \text{Carry} = (a \wedge b) \vee (b \wedge c) \vee (c \wedge a)$$

Exact Compressor (cp): This most probably instantiates an exact 4:2 compressor, generally employing two full adders internally, adding four input bits plus a carry-in to generate a sum, a carry, and a carry-out.

Multi-Level Reduction: These compressors and adders accept the partial products in Level 2. The carry and

sum outputs of Level 2 are fed to modules in Level 3, and the same from Level 3 to Level 4. There is a hierarchical arrangement that successively decreases the number of bits until the final 16-bit Product is obtained. The locations of approximate compressors are presumably aimed at the lower significant bit positions where errors contribute less.

Ripple Carry Adder (RCA): Instantiates a 16-bit Ripple Carry Adder by wiring 16 generic full_adder instances in chain. The carry-out of a given adder (c_1 , c_2 , etc.) is the carry-in for the following one. This design is easy in terms of layout but slow for wide bit sizes because the carry signal has to "ripple" through all stages before the end sum bits are ready.

Register (Accumulator): Instantiates a plain 16-bit synchronous register with an asynchronous reset. On the increasing clock edge, when reset is high, the output is 0; otherwise, the output latches the input value (D_{in}).

$$D_{out}[t+1] = (\text{rst}) ? 0 : D_{in}[t].$$

B. Physical Design Implementation of MAC Unit

After successful logic synthesis, the physical design phase converts the gate-level netlist of the MAC unit into a geometric layout that can be manufactured. It starts with the importing of the synthesized netlist (.v),

the physical library information (.lef that specifies cell sizes), and the timing constraints (.sdc). Floor planning is the initial step, where the total chip area is specified, usually with a target core utilization (e.g., 70%) to maintain room for wiring, and the I/O pins are allocated. Then, power planning constructs the VDD and ground grid structure (add_rings) on the chip to supply power safely to all the components. Finally, placement (place_opt_design) accurately positions all basic cells (flip-flops and logic gates) in the netlist on the floorplan with initial estimates of timing and wire length optimized.

The second major step is routing (route_opt_design), where the tool meticulously places the metal

interconnects (wires) on different layers to connect the pins of the placed cells according to the netlist logic, optimizing to meet the timing constraints set earlier. Post-routing, post-route optimizations and verification checks such as DRC (Design Rule Check) (verify_drc) and Connectivity checks (verify_connectivity) are conducted to verify manufacturability and electrical correctness. The final product is the GDSII file (write_gds), which is the final geometric drawing of the MAC unit chip to be manufactured. Final area, and power reports (report_area, report_timing, report_power) based on the real layout are also produced at this point.

IV. RESULTS

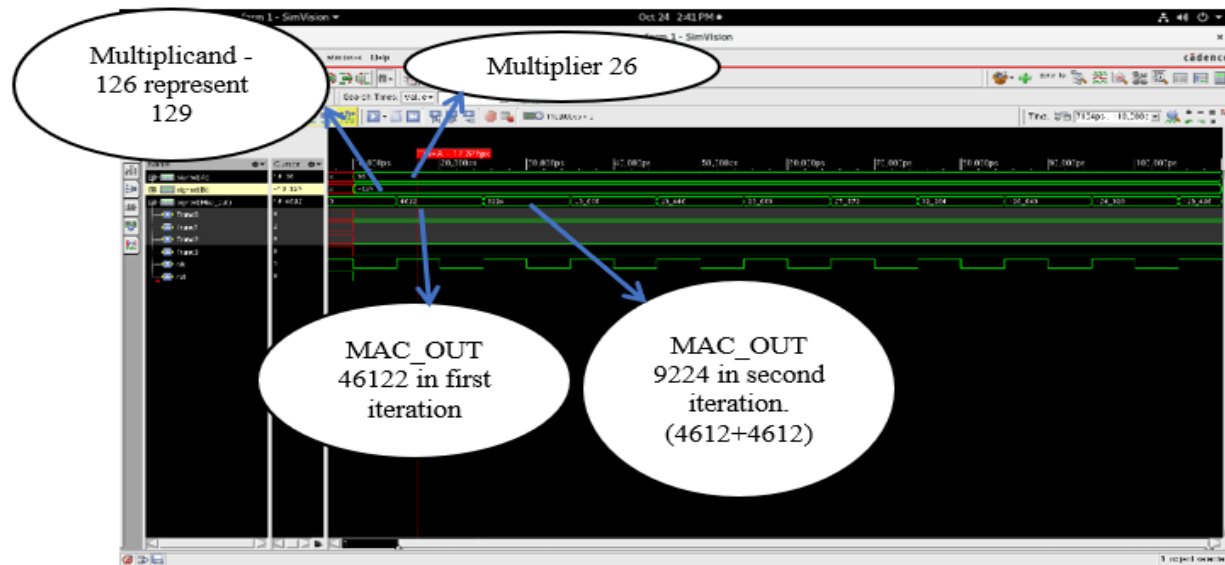


Figure 4. Waveform of MAC Unit Using Approximate Multiplier

The above Figure 4 illustrates the simulation waveform results for the ROBA based MAC unit. Initially, the accumulator output (Mac_out) is shown as being reset to zero. Subsequent to the reset, the approximate multiplier operates on the input values provided for a (decimal 36) and b (decimal 129, represents -127 in 8-bit two's complement).

During the first active clock cycle after reset time, the near multiplier delivers the product of these inputs as 4612. This product value is captured and stored in the accumulator register (Mac_out) at the subsequent clock edge.

The second clock cycle has the multiplier take the same input values (36 and -127) and calculate the same

approximate product of 4612. The resulting product is added to the content currently residing in the accumulator (4612). Therefore, the accumulator register (Mac_out) is now updated with the new value of 9224 (which equals $4612 + 4612$). This is repeated during subsequent clock cycles: the multiplier produces the approximation of the inputs, and this approximation is accumulated into the running total held in the accumulator. The waveform clearly shows this time-domain cumulative multiply-accumulate process as shown in Figure3. The synthesis result of (schematic of) MAC unit and physical design of (chip layout of) MAC unit is shown in figure5 and figure6 respectively.

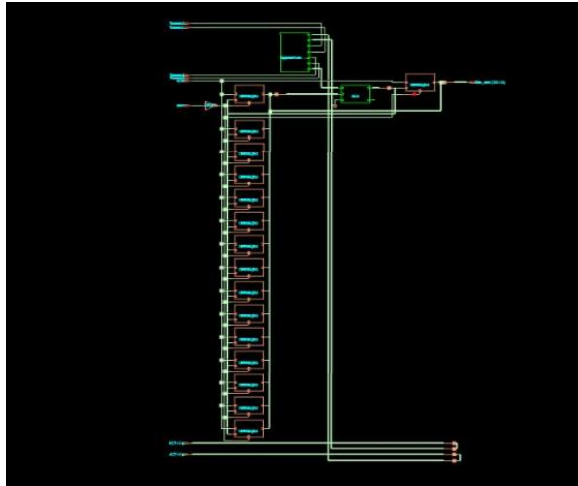


Figure 5. Schematic of MAC Unit Using Approximate Multiplier

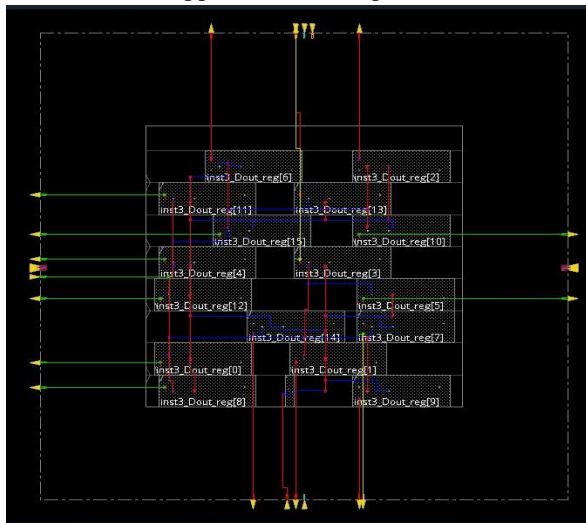


Figure 6. Chip Layout of MAC Unit Using ROBA Multiplier.

Table 1, illustrates the comparison of area, power and gate count and its graphical comparison as shown in figure 7, figure 8 and figure 9.

Type of Multiplier	Area in μm^2	Area saving in %	Power in mW	Power saving in %	Gate Count	Gate Count saving in %
MAC based Vedic	1696.21	100	22.32	100	122	23.75
Wallace	1353.34	20.21	13.34	40.23	160	100
MAC based RoBA	341.36	79.88	4.85	78.27	17	89.38

Table 1. Comparison of Design parameters

The above Table 1 illustrates the comparison of power, area, and gate count of different types of MACS based

Multiplier. The ROBA based MAC saving an area of 79.88% compared to Vedic based MAC unit and ROBA based MAC unit saving a power of 78.27% compared to Vedic based MAC unit is illustrated in the above Table 1.

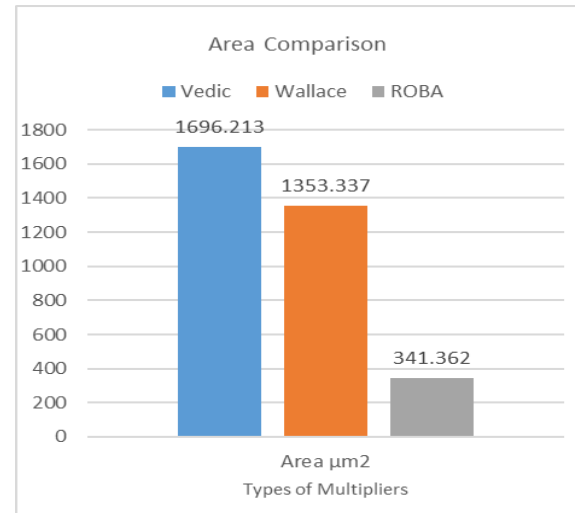


Figure 7. Comparison of Multipliers Area.

Figure 7 illustrates the graphical representation of area comparison. From figure 7 we conclude that the Wallace Multiplier saving the area of 20.21% and ROBA Multiplier saving the area of 79.88% when compared to Vedic Multiplier area.

Figure 8 illustrates the graphical representation of Gate count comparison. From figure 8 we conclude that the Vedic Multiplier saving the Gate count of 23.75% and ROBA Multiplier saving the Gate count of 89.38% when compared to Wallace Multiplier Gount count. The proposed ROBA Multiplier saved more gates compared to Vedic/Wallace Multiplier i.e. 17 as shown in figure 8.

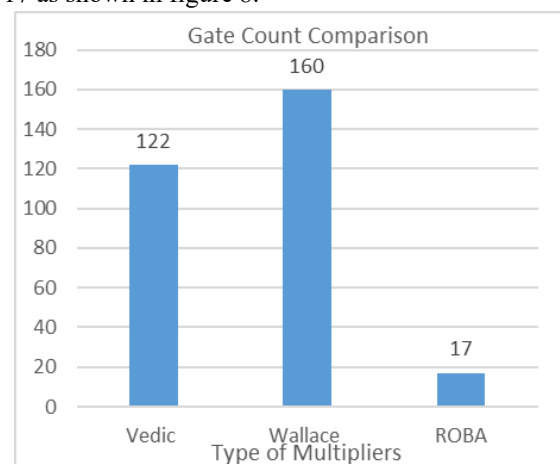


Figure 8. Comparison of Multipliers Gate Count.

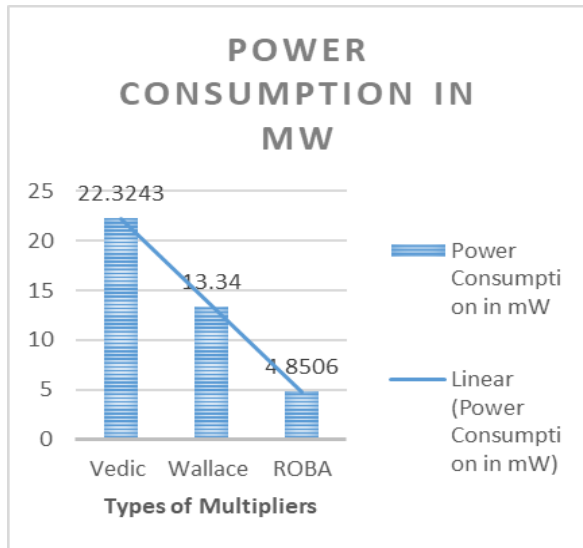


Figure 9. Comparison of Multipliers Power.

Figure 9 illustrates the graphical representation of power comparison. From the figure 9 we conclude that the Wallace Multiplier saving the power of 40.23% and ROBA Multiplier saving the area of 78.27% when compared to Vedic Multiplier power.

V. CONCLUSION

This design efficiently deploys a ROBA based MAC unit which works based on dynamic partial product truncation and approximate compressors to attain good reduction in area and power. The extent of the approximation (and therefore error) can be regulated through the Truncation inputs. Although the component of the multiplier may experience increased speed, the performance of the whole MAC unit would be limited due to the Ripple Carry Adder that is utilized for accumulation. The design's applicability relies on whether the application of interest tolerates computational error with a high priority as compared to its requirement for minimal power as well as area. The ROBA based MAC unit shows significant reduction in area up to 79.88% and power up to 78.27% when compared with Vedic based MAC units. The obtained power consumption of 4.85 mW represents a 9.68% saving compared to the 5.37 mW reported in the reference paper.

REFERENCES

[1] Bushra Siddiqui ,Sudhanya P, "Performance Analysis of MAC unit with Various Parallel Adders" , 2025, IEEE International Students'

Conference on Electrical, Electronics and Computer Science (SCEECS) © 2025 IEEE | DOI: 10.1109/SCEECS64059.2025.10940882.

- [2] K.Prathyusha, M.Balaraju, K.Jamal, Manchalla.O.V.P.Kumar, M.Suneetha, B.Akash "Designing a Mac Unit Using ApproximateMultiplier", 2024, 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) | 979-8-3503-7519-0/24/\$31.00 ©2024 IEEE | DOI: 10.1109/ICAAIC60222.2024.10575181.
- [3] Shaik Nannu Shaida, Shajahan Patan , "A Navel Rounding Based Approximate Multiplier for High Speed Mac Unit Design", December 2024 | IJIRT | Volume 11 Issue 7 | ISSN: 2349- 6002.
- [4] Abinav Balachandar, Aniket Patel, Ramesh S, "Design of a Vedic Multiplier based 64-bit Multiplier Accumulator Unit" 2024, 5th International Conference on Innovative Trends in Information Technology (ICITIIT) | 979-8-3503-8681- 3/24/\$31.00 ©2024 IEEE | DOI: 10.1109/ICITIIT61487.2024.10580179.
- [5] S.Saranya,G. Sudha , Sankari Subbiah, Surya. S, Rajalakshmi.B, Devi priya. P "DESIGN OF AN EFFICIENT MAC UNIT FOR DSP APPLICATIONS" 2023, Intelligent Computing and Control for Engineering and Business Systems (ICCEBS) | 979-8-3503-9458-0/23/\$31.00 ©2023 IEEE | DOI: 10.1109/ICCEBS58601.2023.10448585.
- [6] Bharat Garg, Sujit Patel, "Reconfigurable Rounding Based Approximate Multiplier for Energy Efcient Multimedia Applications",under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021.
- [7] Neelima K, Satyam "High Performance Variable Precision Multiplier and Accumulator Unit for Digital Filter Applications" 2021, IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER) | 978-1-6654-1244-5/21/\$31.00 ©2021 IEEE | DOI: 10.1109/DISCOVER52564.2021.966366.
- [8] G. Umamaheswara Reddy, K.Rajesh , "FPGA Implementation of Multiplier- Accumulator Unit using Vedic multiplier and Reversible gates"International Conference on Inventive

Systems and Control (ICISC 2019) IEEE Xplore
Part Number: CFP19J06-ART; ISBN: 978- 1-
5386-3950-4.

- [9] Joseph Anthony Prathap, Sarvigari Akhila Reddy, Vattam Shravya, Madugula Aparna, Moghal Raval Cheruvu Nagma "FPGA Based Schonhage Strassen Integer Multiplication Algorithm" Jour of Adv Research in Dynamical & Control Systems", Vol. 10, 03-Special Issue, 2018.
- [10] Reza Zendegani, Mehdi Kamal, Milad Bahadori, Ali Afzali-Kusha, and Massoud Pedram, "RoBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing", IEEE Transaction on very largescale integration (VLSI) systems | DOI:1063-8210 © 2016 IEEE.ss.
- [11] G.Indira and B.Srinivas, "Design and Implementation of Multiplier Accumulator (MAC) unit using Rounding Based Approximate (ROBA)", Journal of Information and Computational Science | ISSN:1548-7741, volume 10.
- [12] M Naveen Kumar Reddy and E Balakrishna, "Design of High Speed Rounding Technique Based Approximate Multiplier", International Journal of Research Publication and Reviews | ISSN 2582-7421, volume 4
- [13] M.Srinivasachary, P.Ashok and Dr.P.Bala murali Krishna, "Design of an Efficient Rounding-Based Approximate Multiplier(ROBA)", International Journal of Management, Technology and Engineering | ISSN NO:2249-7455 , volume IX.
- [14] M Sagar and K Gouthanmi, "DSP Applications Based Rounding based Approximate Multiplier for High Speed Yet Energy Efficient", Journal of Nonlinear Analysis and Optimization", | ISSN:1906-9685, volume 14.