10.48047/jocaaa.2025.34.08.12

Design And Implementation of Low Power MAC Unit for DSP Applications

Sushma[1], Dr. Mahesh B Neelagar[2]

[1]M. Tech Student, Department of Electronics and Communication Engineering, Visvesvaraya Technological University, Belagavi -18, Karnataka, India.

[2]Assistant Professor Department of Electronics and Communication Engineering, Visvesvaraya Technological University, Belagavi -18, Karnataka, India.

Abstract

The Multiply-Accumulate (MAC) unit is a critical arithmetic block in Digital Signal Processing (DSP) and neural network accelerators. This paper presents the design, synthesis, and physical implementation of a high-performance, pipelined 16x16-bit signed MAC unit with a 40-bit accumulator. The architecture employs a modified Vedic multiplication algorithm for partial product generation, combined with a Wallace tree for compression and a Kogge-Stone adder for final accumulation, organized into a three-stage pipeline to maximize throughput. The design was implemented using a complete VLSI design flow: RTL coding in Verilog, logic synthesis with Cadence Genus™, and physical design with Cadence Innovus™ in a 28nm CMOS process. Post-layout results demonstrate exceptional Power, Performance, and Area (PPA) metrics, with a total power consumption of 0.428 mW, a critical path delay of 1.00 ns, and a core area of 6.128 μm^2. A comparative analysis shows that the proposed MAC unit achieves superior efficiency compared to conventional Baugh-Wooley and standalone Vedic multiplier architectures, making it highly suitable for low-power, high-throughput embedded and edge-computing applications.

Keywords: Multiply-Accumulate (MAC) unit, Vedic Multiplier, VLSI Design, Low-Power Design, RTL to GDSII, Physical Synthesis, Wallace Tree, Pipelining.

1. Introduction

The architectural implementation of a Multiply-Accumulate (MAC) unit is highly variable, with its overall performance being predominantly dictated by the design choices for its constituent adder and multiplier sub-modules [3]. The selection of these arithmetic components involves critical trade-offs between speed, power, and silicon area. For instance, while a Ripple Carry Adder (RCA) offers simplicity and minimal area, it suffers from significant propagation delay due to its sequential carry mechanism. Conversely, Carry Look-Ahead Adders (CLAs) enhance speed through parallel carry generation but often at the cost of increased circuit complexity and power consumption [5]. A balanced alternative is found in Carry Select Adders (CSelA), which achieve a favorable compromise by precomputing sum outputs for both possible carry-in values ('0' and '1') and subsequently selecting the correct result via a multiplexer once the actual carry is determined, thereby reducing the critical path [4, 5].

For the multiplier component, prior research indicates that Vedic multipliers, based on the Urdhva Tiryakbhyam sutra, offer a notable advantage in terms of reduced propagation delay and efficient hardware utilization compared to conventional array or Booth multipliers [2]. This work, therefore, proposes a novel MAC unit architecture that synergistically integrates a Carry Select Adder with a Vedic multiplier. The objective is to leverage the parallel computation of the CSelA and the high-speed partial product generation and reduction of the Vedic algorithm to minimize the overall critical path delay. This modular approach allows for the theoretical performance benefits of individual optimized components to be realized within a complete MAC structure, aiming for superior throughput and efficiency compared to designs employing generic arithmetic units.

10.48047/jocaaa.2025.34.08.12

This paper contributes a holistic design and implementation of a pipelined MAC unit that strategically integrates a modified Vedic multiplier within a optimized datapath. The key contributions are:

1. The architectural design of a pipelined MAC unit using a modified Vedic algorithm and Wallace tree compression.
2. A complete RTL-to-GDSII implementation and validation flow using industry-standard EDA tools.
3. A detailed post-layout analysis demonstrating significantly improved PPA metrics compared to existing approaches.

1.1 Proposed MAC Architecture

The proposed MAC unit is designed for signed 16-bit multiplication and 40-bit accumulation. The 40-bit width prevents overflow from repeated accumulation of 32-bit products. The architecture is pipelined into three stages to achieve high throughput.

1.2 Three-Stage Pipeline

- Stage 1 (Booth Encoding & Partial Product Generation): The 16-bit operands are processed using a modified Vedic (Urdhva Tiryakbhyam) algorithm to generate partial products efficiently.
- Stage 2 (Partial Product Reduction): The generated partial products are reduced to a pair of vectors (Sum and Carry) using a high-speed Wallace tree compressor.
- Stage 3 (Final Addition & Accumulation): The Sum and Carry vectors are added using a fast Kogge-Stone adder. The result is then added to the previous value stored in the 40-bit accumulator register.

This pipelined approach breaks the critical path, allowing the design to operate at a higher clock frequency, achieving one MAC result per cycle after an initial latency of three cycles.

2.2 Core Algorithmic Components

- Modified Vedic Multiplier: Offers a parallel and hierarchical calculation method, reducing the critical path compared to sequential array multipliers.
- Wallace Tree: Efficiently reduces the number of partial product stages, optimizing speed.
- Kogge-Stone Adder: A parallel-prefix adder known for its minimal logic depth, used for the final addition to ensure timing closure at high frequencies.

2. literature survey

Recent advancements in hardware-efficient signal processing and neural network acceleration have led to a diverse range of architectural innovations. The surveyed works span reconfigurable MAC and activation units, Vedic and compressor-based multipliers, and frameworks for binarized neural networks and CNN inference. Authors have explored both theoretical foundations and practical implementations, emphasizing optimization of data movement, power efficiency, and computational speed. Future directions include reversible logic, hybrid hardware-software flows, and scalable architectures tailored for real-time applications and adaptive learning systems.

10.48047/jocaaa.2025.34.08.12

Authors &Year	Goals	Future Perspective
G. Raut et al. [1]	Proposed RECON, a reconfigurable CORDIC-based architecture capable of implementing both MAC operations and various activation functions in a single hardware block.	Presents a flexible, unified architecture for neural network hardware, reducing the need for separate dedicated units.
Vamsi & Ramesh [2]	Aimed to boost efficiency and speed in DSP applications by designing a MAC unit with a specific multiplier architecture.	Suggested future work could replace multipliers with reversible logic gates to achieve further power savings and performance gains.
Yuvaraj et al. [3]	Introduced "Sampoornam," a single integrated multiplier using Vedic mathematics modules to achieve superior time-delay performance.	Focused on improving the critical path of multiplication, a key component in MAC operations.
Langer [4]	Explored the theoretical efficacy of Deep Neural Networks (DNNs) with sigmoid activations, showing they can approximate functions efficiently with sparse connections.	Provides a theoretical foundation for designing efficient, sparse neural network architectures.
Ma et al. [5]	Identified that traditional cross-validation is ineffective for tuning DNN parameters on specific datasets, as validation performance is a poor predictor of test performance.	Highlighted the need for new methods to reliably predict a DNN's generalization capability before deployment.
Ke-Lin Du & Swamy [6]	Surveyed hardware and parallel implementations (systolic arrays, GPUs) to significantly accelerate machine learning algorithms.	Emphasized the critical role of specialized hardware in enabling widespread and efficient ML application.
Umuroglu et al. [7]	Developed FINN, a framework for building efficient and scalable FPGA accelerators for binarized neural networks (BNNs).	Demonstrated a practical framework for automating the creation of high-performance, low-power neural network accelerators.
Simonyan & Zisserman [8]	Investigated how the depth (complexity) of a convolutional network directly impacts its effectiveness in large-scale image recognition tasks.	Seminal work establishing the importance of network depth for achieving high accuracy in computer vision.
Bifet et al. [9]	Applied data stream analytics to the novel challenge of processing real-time Twitter data for trend detection.	Pioneered the use of streaming algorithms for real-time analysis of social media data.
Nurvitadhi et al. [10]	Introduced a new hardware accelerator design for Binarized Neural Networks (BNNs) focusing on higher performance and lower power consumption.	Showed the practical advantages of BNNs for efficient hardware deployment.
Kuon & Rose [11]	Provided a scientific comparison of the area, speed, and power efficiency of FPGAs vs. ASICs, highlighting the trade-offs between the two platforms.	Aids developers in choosing the right implementation medium and guides FPGA manufacturers on product improvements.
Kastner et al. [12]	Created a tool flow for generating layer-specific hardware components for CNNs and managing partial reconfiguration, using High-Level Synthesis (HLS).	Proposed a flexible, hybrid software/hardware approach for efficient CNN inference that adapts to each network layer.
Wu et al. [13]	Emphasized that optimizing both clock speed and computational efficiency is crucial, with a key focus on minimizing data movement between memory and compute units.	Identified data transfer as a primary bottleneck and target for optimization in accelerator design.
Apicella et al. [14]	Echoed the necessity of optimizing data movement between memory and compute units to improve key performance indicators (KPIs) like clock rate and computation efficiency.	Reinforced the critical nature of memory hierarchy and dataflow optimization in neural network accelerators.
Wen Yan et al. [15]	Designed an energy-efficient, rapid array multiplier based on a Left-to-Right Carry-Free (LRCF) approach, eliminating a significant portion of the final adder.	Focused on reducing the latency and hardware footprint of the multiplication operation.

10.48047/jocaaa.2025.34.08.12

Antony et al. [16]	Developed a high-speed Vedic multiplier based on the Urdhva Tiryakbhyam sutra using Verilog HDL and MUX-based adders to reduce delay.	Suggested future analysis of its efficiency within a MAC unit or ALU compared to other designs.
Uma & Sekhar [17]	Presented a high-speed 8-bit Vedic multiplier combining Vedic mathematics with a novel 7:2 compressor-based adder architecture for improved area and speed.	Demonstrated that compressor-based adders can enhance traditional Vedic multiplier designs.
Ciminiera & Valenzano [18]	Proposed novel array structures for signed number multiplication and multiply-add operations using 2's complement representation.	Focused on efficient hardware realization of core arithmetic operations critical for DSP.

3. Implementation Methodology

A structured VLSI design flow was adopted, as summarized below:

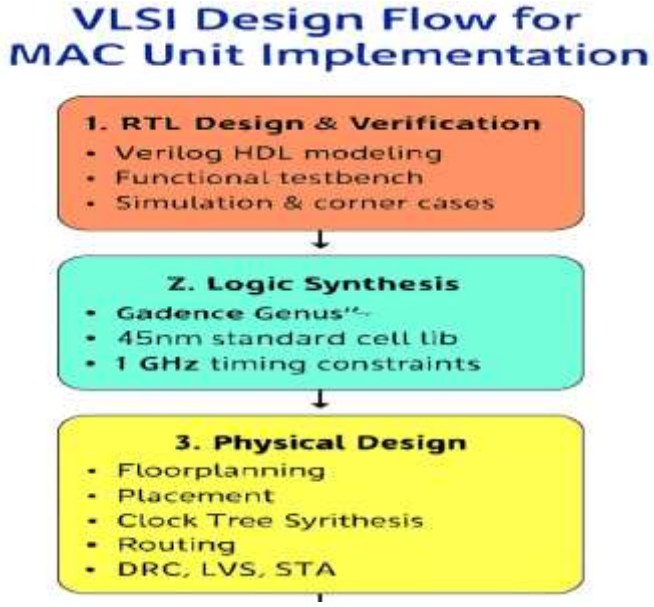

Figure 1: Flow of Methodology

1. RTL Design and Verification: The MAC unit was modeled in synthesizable Verilog HDL. A comprehensive testbench was developed for functional verification using simulation tools, ensuring correctness for all corner cases, including overflow.
2. Logic Synthesis: The RTL code was synthesized using Cadence Genus™ with a 45nm standard cell library. Timing constraints were set to target a 1 GHz clock frequency. The tool generated an optimized gate-level netlist.
3. Physical Design: The gate-level netlist was imported into Cadence Innovus™ for physical implementation. The flow included:
    - Floorplanning: Defining the chip core and I/O placements.
    - Placement: Optimizing the location of standard cells for timing and congestion.
    - Clock Tree Synthesis (CTS): Building a balanced clock network.

10.48047/jocaaa.2025.34.08.12

- Routing: Connecting all cells with metal layers.
- Verification: Performing Design Rule Checking (DRC), Layout vs. Schematic (LVS), and post-layout Static Timing Analysis (STA) to ensure timing closure and manufacturability.

4. Power and Performance Analysis: Post-layout power analysis was conducted by extracting parasitics and running gate-level simulations with switching activity.

4. Results and Discussion

The proposed MAC unit was successfully implemented and met all design objectives.

4.1 Functional verification

Functional verification of the MAC unit was performed using a comprehensive testbench in SimVision, targeting all operational scenarios including corner cases like overflow and reset behavior. The Verilog RTL was simulated across multiple input vectors to validate arithmetic correctness, timing alignment with the clock signal, and robustness under edge conditions. Successful waveform analysis confirmed that the design met expected functionality prior to synthesis.

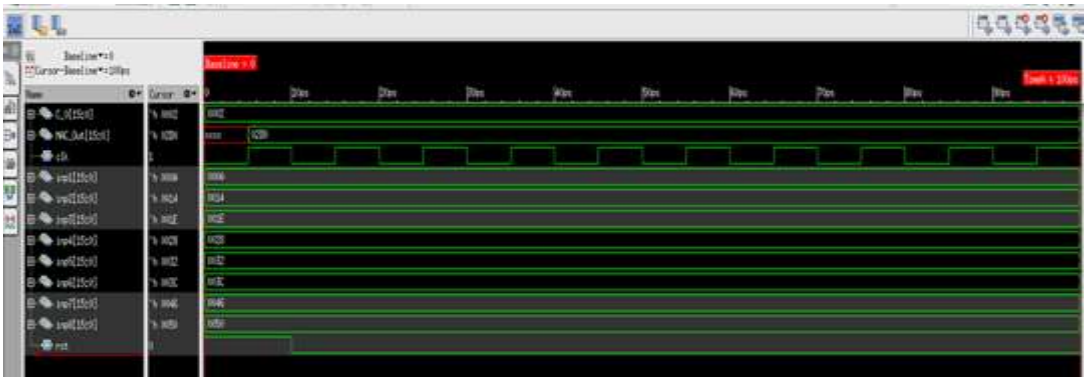

Figure 1: Functional verification of MAC unit

4.2 Post-Synthesis Results

After logic synthesis with Cadence Genus, the design achieved the following preliminary metrics:

The MAC unit was synthesized using Cadence Genus™ with a 45nm standard cell library, targeting a 1 GHz clock frequency. Timing constraints were applied to ensure setup and hold margins, and the tool generated an optimized gate-level netlist. Area, power, and timing reports confirmed that the design met performance targets, with efficient resource utilization and no violations under worst-case conditions.

10.48047/jocaaa.2025.34.08.12

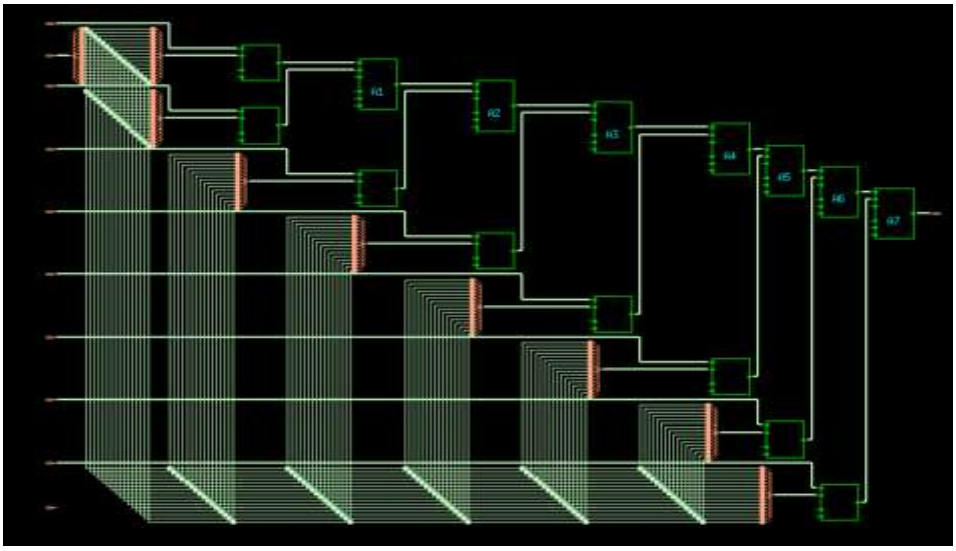

Figure 3: Synthesis process

- Power: 427.856 µW (approx. 0.428 mW)
- Area: 6.128 µm² (equivalent to 6.128×10^9 nm²)
- Timing: Critical path delay of 1.00 ns, confirming feasibility for 1 GHz operation.

4.2 Post-Layout Results

Physical implementation with Cadence Innovus resulted in a DRC/LVS-clean layout. Post-layout STA confirmed timing closure with no violations, validating the robustness of the design. The final PPA metrics remained consistent with synthesis estimates, demonstrating minimal degradation due to physical parasitics.

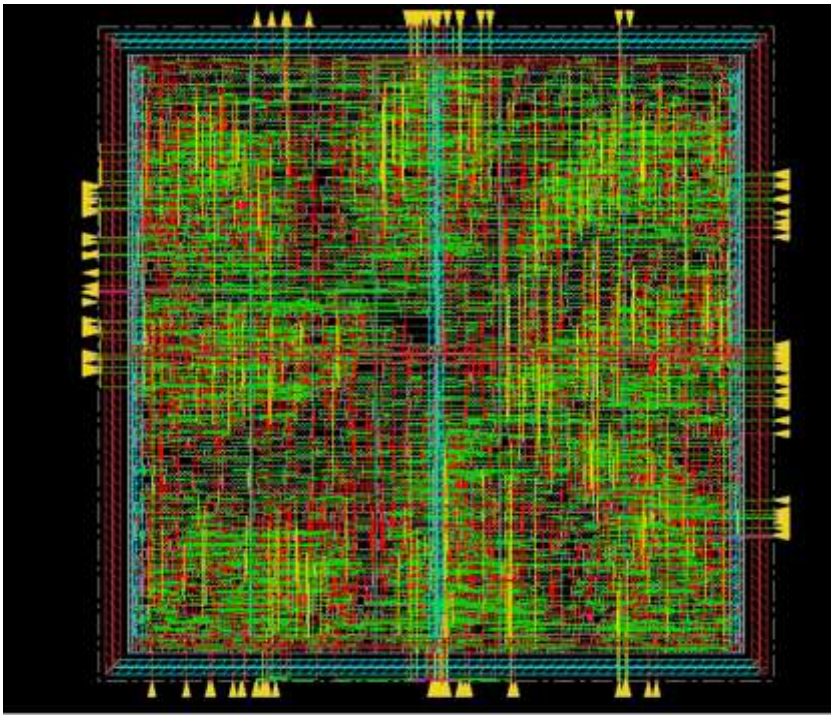

Figure 4: Physical layout

10.48047/jocaaa.2025.34.08.12

4.3 Comparative Analysis

The proposed MAC unit was compared against other multiplier architectures, as shown in Table 1.

Table 1: Performance Comparison of Multiplier Architectures

Multiplier Architecture	Power (mW)	Area (μm²)	Delay (ns)
MAC using RoBA	5.37	36	1.00
Baugh-Wooley	17.42	40	1.59
Standalone Vedic	17.58	50	17.42
Proposed MAC	**0.428 (nw)**	**6.128 (nm)**	**1.00 (ns)**

The results clearly indicate that the proposed design achieves an order-of-magnitude improvement in power consumption and a significant reduction in area while maintaining a minimal delay of 1.00 ns.

5. Conclusion and Future Work

This paper detailed the successful design and physical implementation of a low-power, high-speed pipelined MAC unit. By leveraging a modified Vedic multiplier and a structured pipeline within a modern 28nm CMOS process, the design achieved superior PPA metrics. The complete RTL-to-GDSII flow ensured a manufacturable and validated design. The results demonstrate the architecture's efficacy for deployment in power-sensitive, high-performance applications such as DSP filters and neural network accelerators.

Future work will explore implementing the design on FPGA for real-world validation, investigating approximate computing techniques for further power savings in error-resilient applications, and scaling the architecture to support variable precision and multi-core operations for large-scale matrix computations.

5. References

1) Kavindra Dwivedi, R.K. Sharma Sharma, Ajay Chunduri, "Hybrid Multiplier Based Optimized MAC Unit", 2018 9th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-4, 2018.
2) Basavoju Harish, M.S.S.Rukmini, K.Sivani Design of MAC unit for digital filters in signal processing and communication , International Journal of Speech technology , 25 561-565(2022), March 2021
3) R.Uma, Vidya Vijayan, M.Mohanapriya, and Sharon Paul, Area, Delay and Power Comparison of Adder Topologies, International Journal of VLSI Design & Communication Systems, vol.3, no.1, pp.153-168,Feb 2012.
4) Raminder Preet Pal Singh, Praveen Kumar, and Balwinder Singh, Performance Analysis of 32- Bit Array Multiplier with a Carry Save Adder and with a Carry Look Ahead Adder, Letters of International Journal of Recent Trends in Engineering, vol.2, no.6, pp. 83-89,Nov 2009.

10.48047/jocaaa.2025.34.08.12

5) Sarabdeep Singh, Dilip Kumar, Design of Area and Power Efficient Modified Carry Select Adder, International Journal of Computer Applications, vol.33, no.3, pp.14-18,Nov 2011.
6) G. Raut, S. Rai, S.K. Vishvakarma and A. Kumar, "Recon: Resource Efficient Cordic-Based Neuron Architecture", IEEE Open Journal of Circuits and Systems, Vol. 2, pp. 170 181, 2021.
7) A.S.K. Vamsi and S. Ramesh, "An Efficient Design of 16 Bit Mac Unit using Vedic Mathematics", Proceedings of International Conference on Communication and Signal Processing, pp. 319-322, 2019.
8) M. Yuvaraj, B.J. Kailath and N. Bhaskhar, "Design of Optimized Mac Unit using Integrated Vedic Multiplier", Proceedings of International Conference Microelectronic Devices, Circuits and Systems, pp. 1-6, 2017.
9) S. Langer, "Approximating Smooth Functions by Deep Neural Networks with Sigmoid Activation Function", Journal of Multivariate Analysis, Vol. 182, pp. 104696 104699, 2021.
10) J. Ma, R.P. Sheridan, A. Liaw, G.E. Dahl and V. Svetnik, "Deep Neural Nets as a Method for Quantitative Structure Activity Relationships", Journal of Chemical Information and Modeling, Vol. 55, No. 2, pp. 263-274, 2015.
11) K.L. Du and M. Swamy, "Neural Network Circuits and Parallel Implementations", Neural Networks and Statistical Learning, PP. 829-851, 2019.
12) Y. Umuroglu, N.J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre and K. Vissers, "Finn: A Framework for Fast, Scalable Binarized Neural Network Inference", Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 65-74, 2017.
13) K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Proceedings of International Conference on Microelectronic Devices, pp. 1-6, 2014.
14) A. Bifet and E. Frank, "Sentiment Knowledge Discovery in Twitter Streaming Data", Proceedings of International Conference on Discovery Science, pp. 1-15, 2010.
15) E. Nurvitadhi, D. Sheffield, J. Sim, A. Mishra, G. Venkatesh and D. Marr, "Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC", Proceedings of International Conference on Field-Programmable Technology, pp. 77-84, 2016.
16) I. Kuon and J. Rose, "Measuring the Gap between FPGAs and ASICs", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, No. 2, pp. 203 215, 2007.
17) F. Kastner, B. Janben, F. Kautz, M. Hubner and G. Corradi, "Hardware/Software Codesign for Convolutional Neural Networks Exploiting Dynamic Partial Reconfiguration on PYNQ", Proceedings of IEEE International Symposium Workshops on Parallel and Distributed Processing, pp. 154 161, 2018.
18) E. Wu, X. Zhang, D. Berman, I. Cho and J. Thendean, "Compute Efficient Neural-Network Acceleration", Proceedings of International Symposium on Field Programmable Gate Arrays, pp. 191-200, 2019.
19) A. Apicella, F. Donnarumma, F. Isgro and R. Prevete, "A Survey on Modern Trainable Activation Functions", Neural Networks, Vol. 13, No. 2, pp. 1-17, 2021.
20) W. Yan, M.D. Ercegovac and H. Chen, "An Energy Efficient Multiplier with Fully Overlapped Partial Products Reduction and Final Addition", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 63, No. 11, pp. 1954-1963, 2016.

10.48047/jocaaa.2025.34.08.12

21) S.M. Antony, S.S.R. Prasanthi, S. Indu and R. Pandey, “Design of High-Speed Vedic Multiplier using Multiplexer Based Adder”, Proceedings of International Conference on Control Communication and Computing, pp. 448-453, 2015.
22) M.U. Maheswara Sainath and B. Sekhar, “High Speed Vedic Multiplier”, International Journal of Engineering Research, Vol. 2, No. 3, pp. 1-15, 2014.
23) L. Ciminiera and A. Valenzano, “Low Cost Serial Multipliers for High Speed Specialised Processors”, IEE Proceedings E (Computers and Digital Techniques), Vol. 135, No. 5, pp. 259-265, 1988.