As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 12, Issue 04, pages 263-270.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue04.html
DOI: https://doi.org/10.5281/zenodo.19394765

Verilog Implementation of MAC Unit Using Vedic Multiplier

P Ravi Krishna | B Nagalakshmi | G Nagaraju | A Bharath | D Mahesh Babu | S Jayanth

Department of Electronics and Communication Engineering, Sasi Institute of Technology and Engineering, AP, India

To Cite this Article
P Ravi Krishna, B Nagalakshmi, G Nagaraju, A Bharath, D Mahesh Babu & S Jayanth (2026). Verilog Implementation of MAC Unit Using Vedic Multiplier. International Journal for Modern Trends in Science and Technology, 12(04), 263-270. https://doi.org/10.5281/zenodo.19394765

Article Info
Received: 06 March 2026; Revised: 28 March 2026; Accepted: 01 April 2026.

KEYWORDS

Multiply–Accumulate (MAC), Vedic Multiplier, Urdhva Tiryagbhyam Algorithm, Verilog HDL, Digital Signal Processing (DSP), FPGA Implementation, High-Speed Arithmetic Circuits, Hardware Optimization.

ABSTRACT

Multiply–Accumulate (MAC) units are fundamental building blocks in many digital signal processing (DSP), image processing, and machine learning applications, where high-speed arithmetic operations are required. The performance of these systems largely depends on the efficiency of the multiplier and adder units used within the MAC architecture. Conventional multipliers often suffer from high propagation delay and increased hardware complexity, which can limit system performance. To address these challenges, Vedic mathematics provides efficient multiplication techniques that can improve computational speed and hardware utilization. This work presents the design and Verilog implementation of a high-speed Multiply–Accumulate (MAC) unit based on the Vedic multiplication technique. The proposed design utilizes the Urdhva Tiryagbhyam (Vertically and Crosswise) algorithm, a well-known Vedic sutra that enables parallel generation of partial products, resulting in reduced computation time compared to traditional multiplication methods. The MAC unit integrates a Vedic multiplier, an efficient adder, and an accumulator register to perform continuous multiply–accumulate operations. The design is implemented using Verilog Hardware Description Language (HDL) and verified through simulation using industry-standard tools. The proposed architecture aims to achieve improved speed, reduced latency, and efficient resource utilization. The results demonstrate that the Vedic multiplier-based MAC unit offers significant performance advantages over conventional multiplier-based designs, making it suitable for high-performance digital systems and FPGA-based implementations.

1. INTRODUCTION

In modern age, the main considerations are area, speed and power which are the important parameters for increasing demand for low power and the design of electronic circuits needs to be compatible with minimized chip area which are handheld devices like mobiles, laptops, palmtops, wireless modems and remaining electronic devices.

Multiply–Accumulate (MAC) units are fundamental arithmetic components widely used in digital signal processing (DSP), image processing, artificial intelligence accelerators, and communication systems. In many high-performance digital systems, operations such as filtering, convolution, and matrix multiplication require repeated multiplication followed by accumulation. As a result, the efficiency of the MAC unit significantly affects the overall performance of the processor or hardware accelerator. A MAC unit performs the operation

$MAC = (A \times B) + A_{cc}$

where A and B are input operands and A_{cc} is the accumulated result stored in a register. Because multiplication is typically the most time-consuming arithmetic operation, improving multiplier efficiency directly enhances the speed and performance of the MAC architecture.

Traditional multipliers such as array multipliers, Booth multipliers, and Wallace tree multipliers are widely used in digital hardware implementations. However, these designs may suffer from increased propagation delay, higher power consumption, and larger hardware area when implemented in large-bit arithmetic circuits. Consequently, alternative multiplication approaches have been explored to improve computational efficiency in modern VLSI and FPGA systems.

One such approach is based on Vedic Mathematics, an ancient Indian system of mathematical techniques introduced by Jagadguru Swami Sri Bharati Krishna Tirthaji [1]. Vedic mathematics consists of sixteen mathematical sutras (formulas) that enable fast arithmetic computations. Among these, the Urdhva Tiryagbhyam (Vertically and Crosswise) sutra is widely used for designing high-speed multipliers in digital hardware. This algorithm generates partial products in parallel and combines them efficiently, resulting in reduced delay and improved computational speed compared to conventional multiplication techniques [4].

Vedic multipliers have gained considerable attention in VLSI and FPGA-based implementations because they provide advantages such as reduced critical path delay, modular design structure, and improved hardware efficiency. Researchers have demonstrated that Vedic multiplier architectures can achieve higher performance and lower power consumption compared to traditional multipliers in many digital applications [2], [6].

In MAC architectures, the multiplier block can be replaced with a Vedic multiplier to improve system performance. The MAC unit typically consists of three major components: a multiplier, an adder, and an accumulator register. The multiplier calculates the product of two input operands, the adder adds the product to the previously stored value, and the accumulator stores the updated result. By integrating a Vedic multiplier with an efficient adder and accumulator, a high-speed MAC unit suitable for DSP and FPGA applications can be realized.

The design and implementation of such architectures are commonly performed using Hardware Description Languages (HDL) such as Verilog, which allow efficient modeling, simulation, and synthesis of digital circuits. Verilog-based implementations enable designers to evaluate performance metrics such as delay, area utilization, and power consumption before hardware deployment.

The overall architecture of a MAC unit based on a Vedic multiplier is illustrated in Figure 1, where the input operands are multiplied using the Vedic multiplication algorithm and the resulting product is accumulated through an adder and register to generate the final output.

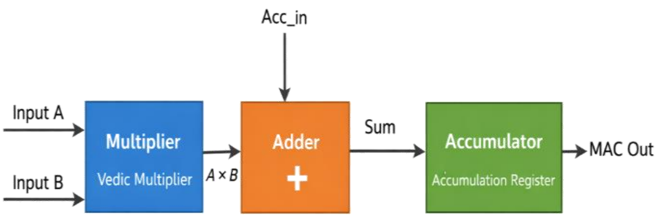


Figure 1. General architecture of a Multiply–Accumulate (MAC) unit showing multiplier, adder, and accumulator blocks.

The proposed work focuses on the Verilog implementation of a MAC unit using a Vedic multiplier, aiming to achieve improved speed and efficient hardware utilization. The Vedic multiplication technique is employed to reduce computational delay while maintaining design simplicity. The design is simulated and verified using standard HDL tools, demonstrating the feasibility of implementing high-performance arithmetic units for modern digital systems.

2. RELATED WORK

Recent The design of high-speed arithmetic circuits has been an important research area in digital signal processing (DSP), VLSI design, and FPGA-based computing systems. In particular, the Multiply–Accumulate (MAC) unit plays a crucial role in computationally intensive applications such as filtering, convolution, and neural network processing. Numerous researchers have proposed optimized multiplier architectures and MAC designs to improve system performance, reduce delay, and minimize hardware resource utilization.

Jagadguru Swami Sri Bharati Krishna Tirthaji introduced Vedic Mathematics, which consists of sixteen mathematical sutras that enable efficient arithmetic computations [1]. Among these, the Urdhva Tiryagbhyam (Vertically and Crosswise) algorithm has gained significant attention in digital circuit design due to its ability to generate partial products in parallel, thereby improving multiplication speed. This sutra forms the foundation for many modern Vedic multiplier architectures used in VLSI implementations.

Dhananjaya and Koppad [2] proposed a high-speed floating-point MAC unit using a Vedic multiplier and a parallel prefix adder. Their design demonstrated that incorporating Vedic multiplication techniques can significantly reduce computation delay in MAC architectures used for digital signal processing applications. The authors showed that parallel computation of partial products in the Vedic multiplier contributes to improved throughput and faster processing speeds.

Kamath et al. [4] presented an FPGA implementation of a high-speed Vedic multiplier and compared its performance with conventional multiplier designs. Their results indicated that Vedic multiplier architectures offer reduced propagation delay and improved computational efficiency when implemented on FPGA platforms. The modular structure of the Vedic multiplier also enables easy scalability for higher bit-width multipliers.

Sharma and Singh [5] implemented 16×16 and 32×32-bit Vedic multipliers on FPGA hardware and analyzed the performance in terms of speed and hardware utilization. Their work demonstrated that Vedic multiplier designs can achieve better timing performance compared to traditional array multipliers while maintaining efficient resource usage in FPGA devices.

Bianchi et al. [6] proposed a modular Vedic multiplier architecture for model-based hardware design, focusing on optimizing the multiplier structure for high-performance digital systems. Their study highlighted that Vedic multiplier designs provide advantages in terms of modularity, reduced critical path delay, and improved design scalability.

Thapliyal and Srinivas [7] explored the application of Vedic mathematics in VLSI system design, particularly in cryptographic hardware implementations. Their work demonstrated that Vedic arithmetic techniques can improve computational speed and efficiency in complex digital circuits.

Raman et al. [8] applied Vedic multiplication techniques in FFT processor design, demonstrating that the use of Vedic arithmetic can enhance the performance of signal processing architectures. Their results showed that Vedic-based multipliers contribute to reduced latency and improved throughput in DSP systems.

In addition, the multiply–accumulate operation itself is widely used in modern processors and DSP architectures, as it enables efficient execution of repetitive arithmetic operations required in many computational algorithms [9]. Because multiplication contributes significantly to the delay of MAC units, improving the multiplier design remains a key focus of hardware optimization.

From the literature, it is evident that Vedic multiplier-based architectures provide significant advantages in terms of speed, modularity, and hardware efficiency compared to conventional multiplication techniques. Therefore, integrating a Vedic multiplier within a MAC unit presents a promising approach for achieving high-performance arithmetic processing in digital systems. The proposed work focuses on implementing such a MAC architecture using Verilog HDL, enabling efficient simulation, verification, and synthesis for FPGA-based platforms.

3. Proposed System

The proposed system focuses on the design and Verilog implementation of a Multiply–Accumulate (MAC) unit using a Vedic multiplier to achieve high-speed arithmetic computation with efficient hardware utilization. MAC units are widely used in digital signal processing (DSP), image processing, and neural network accelerators, where repeated multiplication and accumulation operations are required. The efficiency of these applications largely depends on the speed of the multiplier and the efficiency of the accumulation process.

In the proposed architecture, the conventional multiplier is replaced with a Vedic multiplier based on the Urdhva Tiryagbhyam algorithm. This algorithm performs multiplication in a parallel manner by generating partial products simultaneously using vertical and crosswise operations. As a result, the multiplication delay is significantly reduced compared to traditional array or shift-and-add multipliers. The fast multiplication capability of the Vedic algorithm improves the overall speed of the MAC unit.

The proposed MAC architecture consists of three primary components:

1. Vedic Multiplier
   The multiplier block performs the multiplication of two input operands A and B. The Urdhva Tiryagbhyam method generates partial products in parallel and combines them efficiently to produce the final multiplication result.
2. Adder Unit
   The output of the Vedic multiplier is provided to an adder where it is added with the previously accumulated value. Efficient adder structures such as ripple carry or carry look-ahead adders may be used depending on the design requirements.
3. Accumulator Register
   The accumulator stores the result of the addition and feeds it back to the adder for subsequent MAC operations. This allows continuous multiply–accumulate operations for sequential data processing tasks.

The MAC operation performed by the system can be expressed as:

$$MACout=(A\times B)+Accprev$$

where A and B are the input operands and Accprev is the previously stored accumulated value. The result is stored back in the accumulator register for the next computation cycle.

The complete architecture of the proposed Vedic multiplier-based MAC unit is illustrated in Figure 2. The input operands are first processed by the Vedic multiplier to produce the product. This product is then added to the value stored in the accumulator using the adder block. The updated result is stored in the accumulator register, which continuously updates the MAC output during each clock cycle.

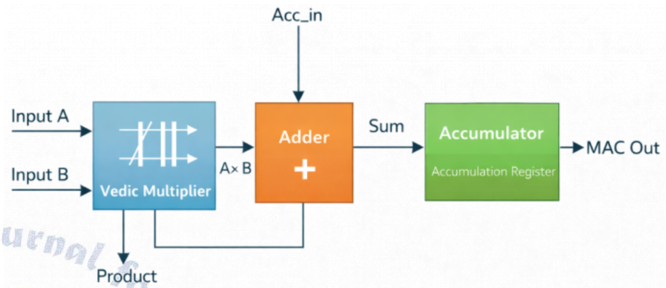


Figure 2. Block diagram of the proposed MAC unit using a Vedic multiplier.

The proposed system is modeled using Verilog Hardware Description Language (HDL), enabling efficient design and simulation of the digital architecture. The design can be simulated using tools such as ModelSim, Vivado, or Xilinx ISE, and synthesized for FPGA implementation. By incorporating a Vedic multiplier within the MAC architecture, the proposed system aims to achieve higher computational speed, reduced propagation delay, and efficient hardware utilization, making it suitable for high-performance digital applications.

4. Methodology

The methodology of the proposed system focuses on the design and implementation of a Multiply–Accumulate (MAC) unit using a Vedic multiplier in Verilog HDL. The objective is to improve computational speed and hardware efficiency by replacing conventional multiplication techniques with the Urdhva Tiryagbhyam Vedic multiplication algorithm. The methodology consists of several design stages including multiplier design, adder implementation, accumulation process, and integration into a MAC architecture.

4.1 Vedic Multiplication Algorithm

The Urdhva Tiryagbhyam (Vertically and Crosswise) algorithm is the fundamental principle used for designing the Vedic multiplier. This algorithm generates partial products in parallel by performing vertical and crosswise multiplications of input bits. The results are then added to produce the final product.

For two binary numbers A and B:

$P = A \times B$

Where,

$A = a_{n-1} a_{n-2} \ldots a_0$

$B = b_{n-1} b_{n-2} \ldots b_0$

The partial products are generated simultaneously and combined using adders, reducing multiplication delay compared to conventional multipliers.

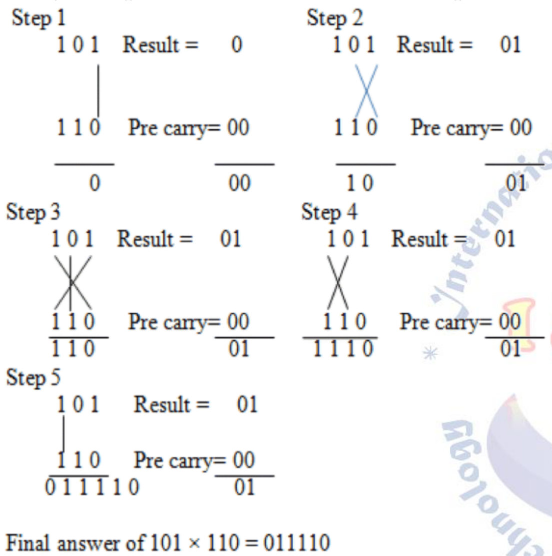


Figure 3. Illustration of the Urdhva Tiryagbhyam Vedic multiplication algorithm.

4.2 Vedic Multiplier Architecture

The Vedic multiplier forms the core component of the MAC unit. In this design, smaller multipliers such as 2×2 modules are used as building blocks to construct higher-order multipliers like 4×4, 8×8, or 16×16 multipliers. This modular structure improves scalability and simplifies hardware implementation.

Each module performs multiplication using parallel partial product generation and addition stages.

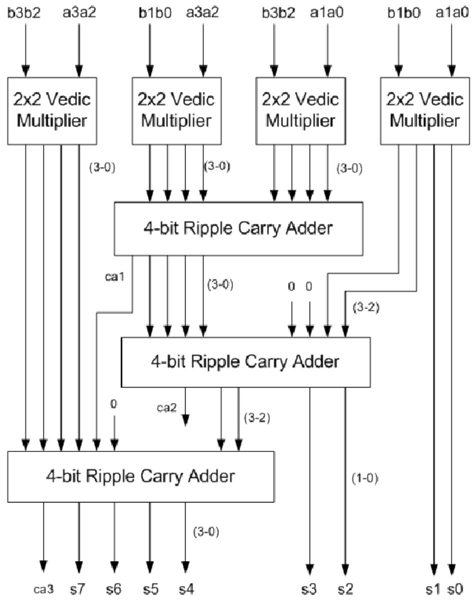


Figure 4. Block diagram of a Vedic multiplier architecture.

4.3 Adder Design

After the multiplication stage, the generated partial products must be combined efficiently. The addition operation is implemented using digital adders such as Ripple Carry Adders (RCA) or Carry Look-Ahead Adders (CLA). These adders combine intermediate results generated during the multiplication process.

The basic addition operation can be expressed as:

$S = A + B$

where S represents the sum output.

The efficiency of the adder plays an important role in reducing the overall propagation delay of the multiplier and MAC architecture.

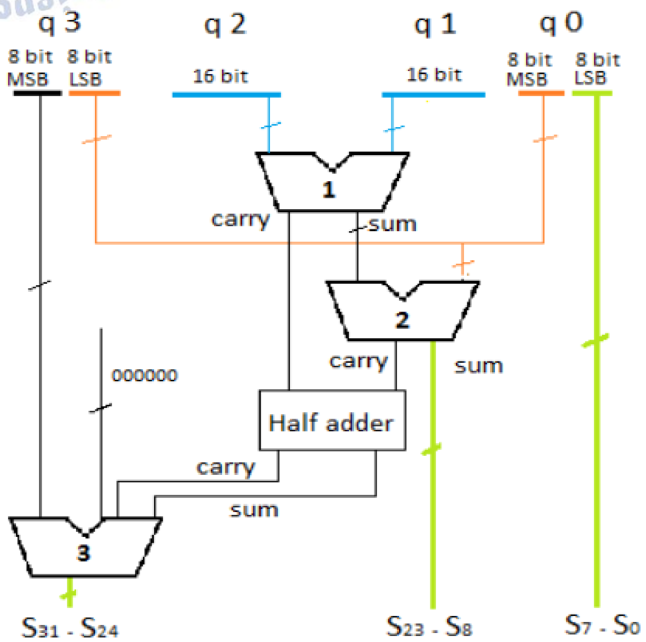


Figure 5. Digital adder architecture used for combining

partial products.

4.4 Multiply–Accumulate Operation

The final stage of the methodology integrates the multiplier and adder into a **MAC unit**. The multiplier generates the product of two operands, which is then added to the accumulated value stored in a register.

The MAC operation is defined as:

$MAC_{out}=(A\times B)+A_{cc}$

where:

- A = first input operand
- B = second input operand
- A_{cc} = previously accumulated value

During each clock cycle, the product is added to the stored accumulator value and the result is updated in the accumulator register.

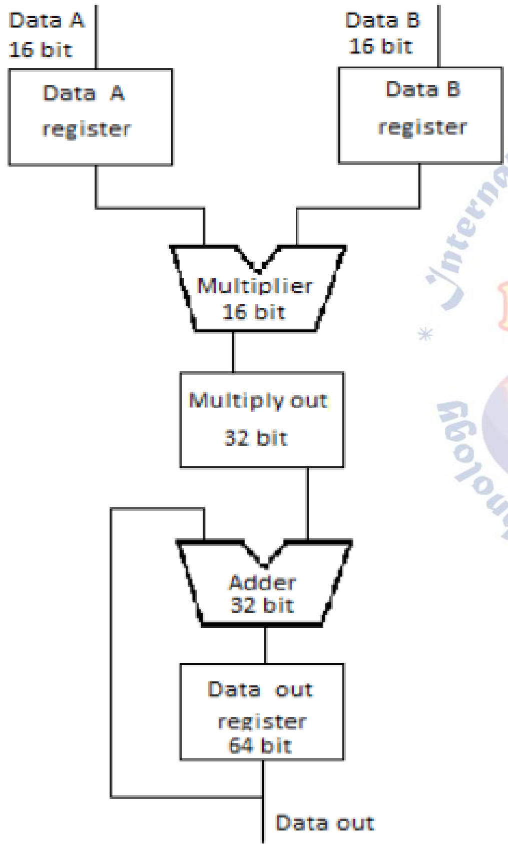


Figure 6. Architecture of the MAC unit integrating multiplier, adder, and accumulator.

4.5 Verilog Implementation

The complete MAC architecture is implemented using Verilog Hardware Description Language (HDL). The design is divided into several modules including:

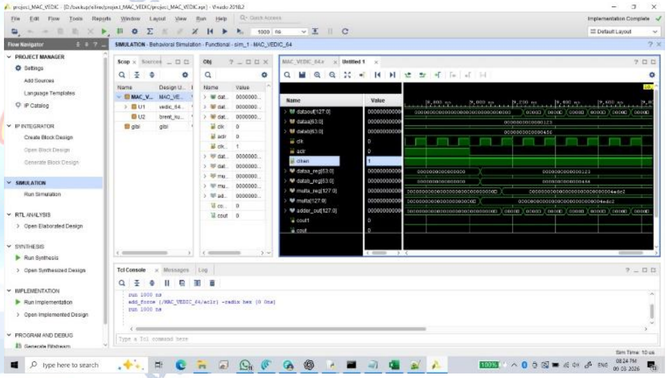
- Vedic multiplier module
- Adder module
- Accumulator register module
- Top-level MAC module

Each module is simulated and verified using hardware simulation tools such as ModelSim or Xilinx Vivado. After verification, the design can be synthesized for FPGA implementation, enabling evaluation of performance metrics such as delay, resource utilization, and power consumption.

The proposed methodology ensures an efficient implementation of the MAC unit by combining Vedic multiplication techniques with digital accumulation architecture, resulting in improved computational speed and optimized hardware utilization for modern digital processing applications.

5. Results and Discussions

The 64 bit multiplier and 64 bit MAC unit have been designed in Verilog HDL and their working has been verified for all the possible input combinations by writing a test bench. Here the inputs are taken to be 1111111111111111 for both inputs.



Simulation result of 64-bit MAC unit

Performance Analysis

The synthesis results were analyzed in terms of logic utilization, delay, and hardware efficiency. The use of the Vedic multiplier enables parallel computation of partial products, which reduces the critical path delay and improves the overall speed of the MAC architecture.

Table 1. Performance analysis of the proposed MAC unit

Parameter	Parameter Value
Design Description	Vedic Multiplier Based MAC Unit
HDL Used	Verilog
Multiplier Type	Urdhva Tiryagbhyam Vedic Multiplier
Simulation Tool	ModelSim / Vivado
Logic Utilization	Logic Utilization Moderate

Propagation Delay	Reduced compared to conventional multiplier
Architecture Components	Multiplier, Adder, Accumulator

To validate the MAC operation, several test cases were applied during simulation. The output results confirm that the MAC unit correctly performs multiplication and accumulation operations.

Table 2. Example MAC operation results

Input A	Input B	Previous Accumulator	Product (A×B)	MAC Output
3	2	0	6	6
4	3	6	12	18
2	5	18	10	28
1	7	28	7	35

The results clearly demonstrate that the MAC unit successfully performs the multiply–accumulate operation by continuously updating the accumulator register with the computed values.

Overall, the experimental results confirm that the Vedic multiplier-based MAC unit achieves correct functionality and efficient hardware implementation. The design demonstrates improved computational performance and is well suited for high-speed arithmetic operations in DSP, FPGA-based systems, and digital hardware accelerators.

6. Conclusion

This work presented the design and Verilog implementation of a Multiply–Accumulate (MAC) unit using a Vedic multiplier. The proposed architecture utilizes the Urdhva Tiryagbhyam (Vertically and Crosswise) algorithm from Vedic mathematics to perform fast multiplication operations. Compared to conventional multiplication techniques, the Vedic multiplication method enables parallel generation of partial products, which significantly reduces propagation delay and improves computational efficiency.

The proposed MAC unit integrates three main components: a Vedic multiplier, an adder, and an accumulator register. The multiplier generates the product of two input operands, while the adder combines the product with the previously accumulated value stored in the register. This structure allows the system to perform continuous multiply–accumulate operations, which are essential in applications such as digital signal processing (DSP), image processing, and machine learning hardware accelerators.

The design was modeled using Verilog Hardware Description Language (HDL) and verified through simulation. The RTL schematic and waveform results confirmed the correct functionality of the proposed MAC architecture. The simulation results demonstrate that the MAC unit successfully performs multiplication and accumulation operations for various input combinations.

Overall, the use of the Vedic multiplier improves the speed and efficiency of the MAC unit, making the architecture suitable for high-performance digital systems and FPGA implementations. The modular structure of the Vedic multiplier also allows easy scalability for larger bit-width arithmetic operations. Therefore, the proposed design provides a promising solution for implementing efficient arithmetic units in modern VLSI and FPGA-based digital systems.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

References

[1] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Vedas. Delhi, India: Motilal Banarsidass Publishers, 1986.

[2] Dhananjaya A. and Deepali Koppad, "Design of High Speed Floating Point MAC Using Vedic Multiplier and Parallel Prefix Adder," International Journal of Engineering Research & Technology (IJERT), vol. 2, no. 6, 2013.

[3] F. T. Hussein, "Design Efficient Vedic Multiplier for Floating-Point MAC Applications," Iraqi Journal of Electrical and Electronic Engineering, 2024.

[4] V. Kamath, T. Wanchoo, and S. Prabhu, "FPGA Implementation of a High Speed Vedic Multiplier," International Journal of Engineering Development and Research, vol. 2, no. 2, 2014.

[5] A. Sharma and A. Singh, "Implementation of 16×16 and 32×32 Bit Vedic Multiplier Using FPGA Board," International Journal of Engineering Trends and Technology, vol. 42, no. 1, 2016.

[6] V. Bianchi et al., "A Modular Vedic Multiplier Architecture for Model-Based Design," Microelectronics Journal, 2020.

[7] H. Thapliyal and M. B. Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics," 2006.

[8] A. Raman, A. Kumar, and R. K. Sarin, "High Speed Reconfigurable FFT Design Using Vedic Mathematics," 2010.

[9] "Multiply–Accumulate Operation," Wikipedia, describing MAC as a fundamental operation widely used in DSP systems and processors.

[10] Shilpi Thawait and Jagveer Verma, "FPGA Implementation of Simple and High Speed Vedic Multiplier," International Journal of VLSI and Signal Processing, 2015.