

DESIGN AND IMPLEMENTATION OF HIGH SPEED 32-BIT VEDIC MULTIPLIER USING VERILOG HDL

¹Mr. M.V.V. SATYA CHOWDARY, ²K. SAI KARTHIK REDDY, ³N. SAIKIRAN,
⁴M. PRAVALIKA, ⁵M. CHANDRAMOHAN REDDY

¹(ASSISTANT PROFESSOR), ECE, TEEGALA KRISHNA REDDY ENGINEERING COLLEGE

²³⁴⁵UG. SCHOLAR, ECE, TEEGALA KRISHNA REDDY ENGINEERING COLLEGE

ABSTRACT

The increasing demand for high-speed digital arithmetic operations in modern computing systems necessitates the development of efficient multiplication techniques. This work presents the design of a high-speed 32-bit binary Vedic multiplier based on the principles of ancient Vedic mathematics, particularly the Vedic Sutras. Vedic multiplication, renowned for its parallel processing capabilities, offers significant advantages over conventional multiplication techniques. The proposed design optimizes the addition and carry management stages, which are crucial for reducing the overall delay in the multiplication process, making it suitable for modern digital systems requiring high-speed

arithmetic operations. The architecture of the 32-bit Vedic multiplier is implemented using Verilog Hardware Description Language (HDL), ensuring flexibility and scalability. The multiplier was simulated and synthesized using AMD Vivado to validate its performance in a simulated environment. Simulation results show that the 32-bit Vedic multiplier significantly reduces time delay compared to conventional multiplier designs. Additionally, optimized resource utilization further enhances its performance. The design is scalable, allowing it to be extended to larger bit-widths for higher-order multiplications, which is particularly advantageous for applications in Digital Signal Processing (DSP), cryptography, and other computationally intensive tasks.

1.INTRODUCTION

1.1 Introduction

In modern digital systems, multiplication is one of the most crucial and frequently used arithmetic operations. From processors and signal processing units to embedded systems and artificial intelligence hardware, the efficiency of multiplication directly influences the system's overall performance, power consumption, and area utilization. As applications continue to demand high-speed, low-power, and area-efficient hardware, the need for advanced multiplier architectures has become more prominent than ever. Traditional binary multipliers, such as array and Wallace tree multipliers, have served their purpose effectively in numerous applications. However, these designs often involve a tradeoff between speed, complexity, and area. With the emergence of high-throughput computing systems and real-time applications, conventional

multiplication methods face significant limitations in terms of speed and resource efficiency, especially when scaled to higher bit-widths like 32-bit and beyond. To overcome these limitations, researchers have increasingly turned toward unconventional and more optimized mathematical methodologies, among which Vedic mathematics has gained considerable traction. Rooted in ancient Indian mathematics, Vedic sutras offer highly efficient methods for arithmetic computations. One such sutra, Urdhva Tiryakbhyam (meaning “Vertically and Crosswise”), forms the basis for the Vedic multiplier. This technique significantly reduces the number of partial products and intermediate steps, enabling faster computations and more compact hardware realization. The motivation behind this project stems from the desire to implement a high-speed, area-optimized 32-bit multiplier using Vedic mathematical principles and modern digital design practices.

Furthermore, the standard Vedic multiplier design is enhanced by introducing Carry Save Adders (CSAs) and hierarchical modular construction, aimed at reducing critical path delays and resource bottlenecks. The integration of CSAs ensures that intermediate sums are processed concurrently, thus improving the overall propagation delay compared to traditional Ripple Carry Adder-based designs. By designing a Modified High-Speed 32-bit Vedic Multiplier in Verilog HDL and testing it through simulation and synthesis tools such as AMD Vivado, this project aims to validate the theoretical speed and area benefits in a practical FPGA design environment. This multiplier design can serve as a foundational block in applications such as digital signal processors (DSPs), embedded control systems, and low-power VLSI designs. The combination of ancient arithmetic wisdom with contemporary hardware engineering not only showcases the timeless value of Vedic mathematics but

also provides a competitive edge in the development of high-performance arithmetic units for modern computing architectures.

1.2 IMPORTANCE OF HIGH-SPEED MULTIPLIERS

High-speed multipliers are indispensable components in modern computational systems. As digital systems evolve to handle increasingly complex tasks — from multimedia processing to real-time artificial intelligence and embedded control — the demand for efficient and high-speed arithmetic units, particularly multipliers, has grown exponentially. At the core of many digital systems, multiplication plays a pivotal role in algorithms for digital signal processing (DSP), image and video compression, cryptography, machine learning, and scientific computations. In many of these applications, the multiplication operation is a bottleneck, often being the most time-consuming arithmetic operation. Therefore, optimizing

the speed of multipliers has a direct and profound impact on overall system throughput and responsiveness. In processors and digital hardware, multiplication frequently occurs in tight loops or timecritical sections of code. For instance, in digital filters or transform computations (e.g., FFT, DCT), a single delay in a multiplication step can propagate throughout the entire system, degrading realtime performance. Consequently, high-speed multipliers contribute not only to faster execution times but also to lower power consumption, as operations can be completed more quickly, allowing circuits to enter low-power states sooner. Furthermore, in VLSI system design, performance metrics are often dictated by the critical path delay, which is the longest time required for data to propagate through combinational logic. Multipliers often lie on this critical path, especially in data paths involving matrix computations, convolution operations, or polynomial evaluations. Designing

multipliers with minimal propagation delay ensures that the overall clock speed of the chip can be increased, thereby improving performance without needing architectural overhauls. In recent years, parallelism and pipelining have been used to accelerate multiplication, but these techniques come at the cost of increased silicon area and power usage. A well-optimized multiplier — such as one based on Vedic mathematics — can reduce the number of required logic gates and interconnections while preserving or enhancing speed. This results in compact, scalable, and low-power implementations, which are particularly valuable in mobile, battery-powered, and embedded environments. Moreover, in hardware accelerators used for AI/ML inference (such as those in GPUs, TPUs, and custom ASICs), millions of multiplications are performed per second. Here, the efficiency of multipliers directly affects computational throughput and energy efficiency, two of the most critical factors in data center and edge

computing applications. In summary, high-speed multipliers are crucial for:

- Enhancing system speed and throughput
- Reducing power consumption and heat dissipation
- Improving area efficiency in VLSI implementations
- Enabling real-time performance in embedded systems
- Supporting scalability for high-performance applications.

The Modified Vedic Multiplier explored in this project directly addresses these demands, offering a balance of speed, area efficiency, and design simplicity, and serving as a high impact building block for next-generation digital systems.

1.3 OVERVIEW OF VEDIC MATHEMATICS

Vedic Mathematics is an ancient system of Indian mathematics derived from the Vedas — specifically the Atharva Veda. It was systematized in the early 20th century by Bharati Krishna Tirthaji, a Sanskrit scholar and mathematician, who reconstructed this system from ancient Hindu texts. He compiled 16 sutras (formulas) and 13 sub-sutras (sub formulas) that collectively offer quick, logical, and efficient methods for arithmetic computation.

2.LITERATURE SURVEY

2.1 Literature Survey

In the evolving landscape of digital system design, the efficiency of multiplication operations plays a critical role in enhancing the performance of processors, especially in applications like signal processing, encryption, and multimedia. Traditional multiplier architectures, such as Array, Wallace Tree, and Booth, often face challenges related to speed, power

consumption, and hardware complexity. To address these issues, researchers have increasingly turned to Vedic mathematics, an ancient system based on 16 sutras, which offers innovative and efficient multiplication techniques like Urdhva Tiryagbhyam and Nikhilam. These methods enable faster computation, better scalability, and improved area and power optimization, making them highly suitable for implementation in Verilog or VHDL and synthesis on FPGA platforms using tools such as Xilinx ISE and ModelSim. The research also emphasizes the importance of modular design for scalability, efficient adder structures for improved performance in cryptographic applications, and comparative studies to benchmark Vedic multipliers against conventional ones. While some included studies diverge into unrelated fields like MANETs, the core focus remains on optimizing arithmetic logic through modified high-speed multiplier designs grounded in Vedic principles.

2.2 MAIN CONCERNS AND REASONS

1. Need for High-Speed and Efficient Multiplication

- **Concern:** Modern digital systems, particularly ALUs and DSPs, demand extremely fast multiplication operations to keep up with real-time processing requirements

- **Reason:** Multiplication is a core operation in many computationally intensive tasks (e.g., image processing, encryption, signal processing), and traditional approaches (like array or Wallace tree multipliers) often result in increased propagation delay and hardware complexity. Vedic multipliers offer reduced delay through parallelism and fewer logic levels, which is why they are being explored extensively.

2. Scalability of Multiplier Architectures

- **Concern:** Designs must scale efficiently from smaller bit widths (4, 8, 16 bits) to

higher ones (32-bit and beyond) without a significant rise in delay or hardware usage.

- **Reason:** As systems evolve, the need for handling larger data sizes grows. The modularity in Vedic algorithms (especially Urdhva Tiryagbhyam) allows for seamless scalability using repetitive structural blocks, ensuring performance does not degrade sharply with increased operand sizes.

3. Optimization of Area and Power Consumption

- **Concern:** Multiplier circuits must consume less silicon area and power, especially in mobile, embedded, and cryptographic applications.

- **Reason:** Low-power designs are vital for battery-operated devices and high-density FPGAs. Vedic multipliers are preferred as they reduce the number of partial products and adders needed, leading to more compact and power-efficient implementations.

4. Hardware Implementation Suitability (Verilog/VHDL)

- **Concern:** Practical implementation requires that these algorithms be efficiently coded in hardware description languages (HDLs) like Verilog or VHDL and synthesized on FPGAs.

- **Reason:** To deploy these designs in real-world applications, simulation and synthesis are necessary. Papers focus on using tools like Xilinx ISE and ModelSim to validate timing, area, and functional correctness.

5. Comparison with Existing Multiplier Architectures

- **Concern:** Vedic multipliers must be benchmarked against other popular architectures such as Booth, Wallace Tree, and Array multipliers to justify their benefits.

- **Reason:** A comparative analysis helps identify performance trade-offs. Vedic

multipliers have been found to outperform in speed and resource utilization but might be less effective in certain fixed-width precision cases without further optimization.

3.PROPOSED SYSTEM

3.1 Modified 32-bit Vedic Multiplier Design

3.1.1 Design Improvements Over Conventional Design:

The Modified 32-bit Vedic Multiplier introduces significant architectural and performance improvements over traditional binary and even standard Vedic multiplication approaches. These enhancements are made to meet the demands of high speed arithmetic in modern digital systems such as embedded processors, DSPs, and real-time automotive control units. Below are the key improvements incorporated in the modified design:

1.Hierarchical and Modular Architecture:

One of the most critical advancements is the strictly hierarchical design, where the 32 bit multiplier is constructed from smaller Vedic modules: • 2-bit → 4-bit • 4-bit → 8-bit • 8-bit → 16-bit • 16-bit → 32-bit This recursive modular construction ensures that each module is thoroughly tested and optimized before being used in the larger design. It allows easier debugging, verification, and synthesis. Unlike conventional flat designs, this hierarchy maintains consistency, readability, and scalability.

2. Use of Carry Save Adders (CSAs):

In conventional designs, carry propagation is a major bottleneck in addition stages. The modified design smartly integrates Carry Save Adders (CSAs) in intermediate stages to postpone carry propagation until the final stage. Advantages of CSAs in this architecture:

- Enables parallel computation of partial sums and carries.
- Minimizes delay caused by serial carry propagation.
- Allows faster accumulation of partial products in the final summation stage.

3. Improved Partial Product

Management:

Instead of relying on a rigid shift-and-add mechanism, the modified design dynamically manages partial products by:

- Carefully aligning them using predetermined logic.
- Efficiently feeding them into dedicated adder modules. This strategy reduces the gate-level switching activity, which improves both speed and power consumption.

4. Optimized Adders for Final Stage

Summation:

The final stage uses high-performance adders (like Ripple Carry or Kogge-Stone in more aggressive variants) depending on the critical path requirements.

5. Parallelism and Pipeline Readiness:

Although not pipelined in its basic form, the modified 32-bit design is pipeline-ready. Each hierarchical module can be pipelined independently if timing closure demands it. The clear module boundaries make it ideal for pipelining in ASIC/FPGA implementations, which can drastically improve the throughput

3.2 FUNCTIONAL BLOCK DIAGRAM

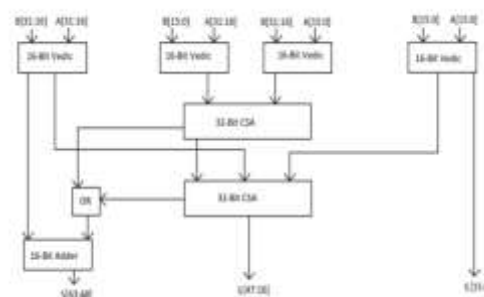


Fig 3.2.: Functional Block Diagram

3.3 SCHEMATIC DIAGRAM

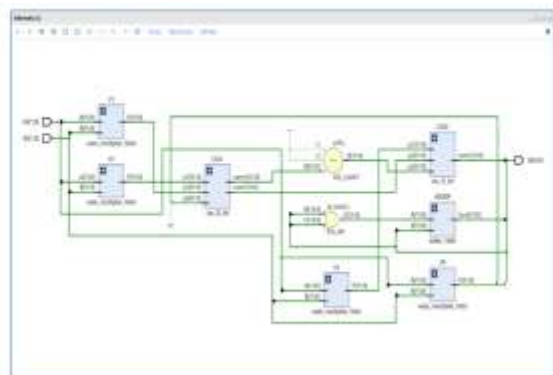


Fig 3.3: Schematic Diagram

The provided schematic represents a 32-bit Vedic multiplier implemented using a modular and hierarchical approach, designed to produce a 64-bit output. The circuit takes two 32-bit inputs, labeled $A[31:0]$ and $B[31:0]$, which are each divided into two 16-bit segments. The design employs four instances of a 16-bit Vedic multiplier module to compute the partial products: V0 handles the lower 16 bits of both inputs, V1 and V2 compute the cross-products between the higher and lower halves, and V3 multiplies the upper 16 bits of both inputs.

These four partial results are then efficiently combined using two 32-bit carry-save adders (CSA1 and CSA2), a logical OR unit, a left shifter, and a 16-bit adder module. The CSA modules play a critical role in adding the intermediate results without propagating carry immediately, which enhances speed. The output of V3 is shifted left by 32 bits using a left shift block to align it properly in the final 64-bit result. The carry and sum from CSA1, along with the shifted output of V3, are added in CSA2. Meanwhile, the result of V0 and a logical OR operation are summed using a 16-bit adder. Ultimately, all partial products and intermediate results are aggregated to produce the final 64-bit multiplication result, $S[63:0]$. This design showcases a high-speed multiplication architecture using Vedic principles, optimized through parallelism and modular design.

4.RESULT



Fig 4.1: Output Simulation for a=15, b=5

1.Signal Breakdown: a[31:0]: Input operand A (32-bit), b[31:0]: Input operand B (32-bit), prod[63:0]: Output product (64-bit, since $32\text{-bit} \times 32\text{-bit} = 64\text{-bit}$).

2.Time Points: At the time marker 33 ns, inputs and output are stable, and we can examine the values: Values at 33 ns: a = 15, b = 3, prod = 45. Explanation: $15 \times 3 = 45$, this confirms the multiplier is functioning correctly for this input.

Other Time Segments:

1. 10 ns: a = 12345678, b = 0, prod = 0, Correct: Anything multiplied by 0 yields 0.

2. 20 ns: a = 0, b = 98765432, prod = 0.

Again, multiplication by 0 is valid.

3. 40 ns: a = 4294967295 (Max 32-bit unsigned value = $2^{32} - 1$), b = 3, prod = 12884901885. Check:

$$4294967295 \times 3 = 12884901885$$

4. 50 ns: a = 2779096485, b = 1515870810, prod = 4212751239785102580

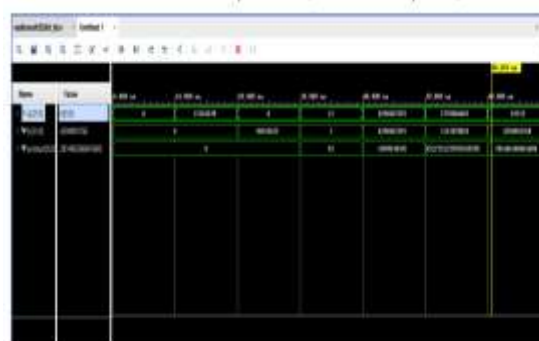


Fig 4.2: Output simulation for a=65535, b=4294901760

5. 60 ns: a = 65535, b = 4294901760, prod = 281466386841600. Again correct (this seems to be multiplying max 16-bit with a 32-bit high mask). This simulation confirms

that the 32-bit Vedic multiplier module is working correctly, producing accurate 64-bit results across a wide range of test cases including small numbers, zeros, and large unsigned integers. The appearance of negative values indicates the simulator may be interpreting the output as signed—which can be addressed by explicitly defining the product signal as unsigned if required

5.CONCLUSION

The development and implementation of the Modified 32-bit Vedic Multiplier reflect a significant advancement in the field of digital arithmetic design. This project successfully combines the elegance of Vedic mathematics, specifically the Urdhva Tiryakbhyam Sutra, with contemporary design principles such as hierarchical architecture, modular construction, and carry-save addition to produce a multiplier that is both computationally efficient and practically deployable in realtime systems.

Synthesis of Ancient Principles with Modern Technology:

At the heart of this work is the innovative application of the Urdhva Tiryakbhyam Sutra—a centuries-old Vedic technique for multiplication—adapted for binary arithmetic. Unlike conventional multiplication algorithms that depend heavily on sequential addition and shifting operations, the Vedic approach allows for parallel generation and accumulation of partial products, leading to substantial reductions in delay. By mapping this technique to digital logic, the design inherits the benefits of parallelism, regular structure, and simplicity, making it ideal for VLSI implementation. The result is a high-speed multiplier that outperforms traditional designs, both in simulation and hardware synthesis. Modular and Hierarchical Architecture One of the cornerstones of this project is the adoption of a modular, hierarchical structure. The 32-bit multiplier

is not built monolithically but rather constructed from proven, smaller components:

• **2-bit** → **4-bit** → **8-bit** → **16-bit** → **32-bit**

Each of these stages has been meticulously designed, simulated, and verified individually, ensuring accuracy, scalability, and reliability. This structure not only simplifies the verification process but also enables reuse of modules in other designs, thus promoting design efficiency and maintainability.

6. REFERENCES

[1] S. Akhter, "VHDL implementation of fast NxN multiplier based on Vedic mathematics," in Proc. 18th European Conference on Circuit Theory and Design, 2007, pp. 472-475.

[2] S. Nagaraj, Dr.G.M. Sreerama Reddy and Dr.S. Aruna Mastani; A Comparative Study on Different Multipliers-Survey Journal of Advanced Research in Dynamical

and Control Systems14739-7522018Institute of Advanced Scientific Research.

[3] M.Pushpa, S. Nagaraj, Design and Analysis of 8-bit Array, Carry Save Array, Braun,Wallace Tree and Vedic Multipliers, IEEE Sponsored International Conference On New Trends In Engineering & Technology(ICNTET 2018).

[4] Nagaraj, S; Thyagarajan, K; Srihari, D; Gopi, K; Design and Analysis of Wallace Tree Multiplier for CMOS and CPL Logic2018 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)006-0102018IEEE

[5] Josmin Thomas ; R. Pushpangadan ; S Jinesh Comparative study of performance Vedic multiplier on the Basis of Adders used 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE).

[6] S. Nagaraj, Dr.G.M. Sreerama Reddy and Dr.S. Aruna Mastani; A Survey on Adiabatic Logic International Conference on Communications, Signal Processing and VLSI(IC2SV2019), Springer Conference, National Institute of Technology, Warangal.

[7] S. Nagaraj, K. Venkataramana Reddy and P. Anil Kumar; Analysis of Vedic Multiplier for Conventional CMOS & Complementary Pass Transistor Logic (CPL) Logics SCOPUS Indexed Springer 8th International Conference on Innovations in Electronics and Communication Engineering, (ICIECE-2019).

[8] Au L.S. and Burgess N. (2002), "A (4:2) adder for unified GF(p) and GF(2ⁿ) Galois field Multipliers", Proceedings of 36th IEEE Asilomar Conference on Signals, Systems, and Computers, vol. 2, pp. 1619-1623.

[9] Chittibabu A., Sola V.K. and Raj C.P. (2006), "ASIC Implementation of New Architecture for constant coefficient Dadda

multiplier for High Speed DSP applications", Proceedings of the National Conference on Recent trends in Electrical, Electronics and Computer Engineering, JCECON, pp. 299–304.