



Design and Implementation of a High-Speed 32-bit Signed Vedic Multiplier Using Carry Lookahead Adders

SHAIK KARISHMA, Student, Department of Electronics & Communication Engineering, Nimra College of Engineering and Technology, Ibrahimpatnam

Dr. K.A.LATHIEF, Professor, Department of Electronics & Communication Engineering, Nimra College of Engineering and Technology, Ibrahimpatnam

Abstract—This paper presents the design and implementation of a high-speed 32-bit signed Vedic multiplier based on the Urdhva Tiryakbhyam sutra of Vedic mathematics, utilizing Verilog HDL and fast Carry Lookahead Adders. The proposed architecture recursively combines smaller signed Vedic multipliers and efficient adder modules to achieve parallel generation of partial products and rapid accumulation, significantly improving computational speed over conventional multiplication techniques. The multiplier supports both signed and unsigned operations, implementing robust logic for sign handling via two's complement conversion and Ex-OR-based sign determination. Simulation and verification are conducted using Xilinx Vivado, demonstrating correctness and performance suitable for applications in digital signal processing, arithmetic logic units, and custom VLSI-blocks. The results highlight the advantages of Vedic mathematical principles in modern digital hardware, offering improved throughput and efficiency for high-performance arithmetic operations.

Index Terms—Vedic multiplier, Urdhva Tiryakbhyam, signed multiplication, Carry Lookahead Adder (CLA), Verilog HDL, digital signal processing (DSP), FPGA implementation, high-speed arithmetic, VLSI design.

I. INTRODUCTION

Multiplication is a fundamental arithmetic operation and a critical component of almost every digital computing system. From microcontrollers to high-performance processors, multiplication units directly influence a system's computational capability, operating frequency, and overall efficiency. Applications in *digital signal processing* (DSP), image and video processing, scientific computing, and cryptography place stringent demands on multipliers—requiring high speed, low latency, and minimal hardware resources [9], [12]. For example, in DSP-based filters, the performance bottleneck is often the multiply-accumulate (MAC) stage, and even a small improvement in multiplication speed can significantly elevate system throughput [2], [19].

Traditional approaches such as array multipliers, serial/parallel multipliers, and Booth encoding have been standard in arithmetic hardware design for decades. However, these methods tend to incur large propagation delays as the operand size increases, due to considerable carry propagation and sequential partial product accumulation [4], [5], [13].

The need for scalable, high-speed solutions has driven the exploration of alternative algorithms that allow partial products to be computed in parallel and summed with minimal delay.

One promising source of inspiration comes from **Vedic mathematics**, an ancient system formalized by Jagadguru Swami Sri Bharati Krishna Tirthaji in the early 20th century, based on 16 computational sutras derived from the Vedas [8]. Among these, the *Urdhva Tiryakbhyam* (Vertically and Crosswise) method has drawn significant interest in the digital design community. This algorithm generates partial products simultaneously for all bit positions, enabling substantial parallelism and allowing output computation in less time compared to conventional methods [1], [3], [6], [7], [14]. Its suitability for hardware stems from its simple, repetitive structure, which readily lends itself to a modular and hierarchical implementation.

Recent research has focused on exploiting the inherent parallelism of Vedic methods by combining them with advanced adder architectures. High-speed adders such as the *Carry Lookahead Adder* (CLA) minimize carry propagation time, which otherwise becomes a bottleneck in large-bit multipliers [1], [3], [16]. When integrated into a recursive Vedic design, CLAs enable the construction of large-width multipliers (e.g., 32-bit or 64-bit) using smaller, faster building blocks. Various optimization strategies have also been explored, including hybrid designs [17], [20] and alternative sutra-based multipliers such as the Nikhilam method [18] for certain operand ranges. However, while many works address unsigned multiplication, **signed multiplication** is essential in numerous real-world applications such as digital control systems, DSP algorithms, and signed integer-based computations in CPUs. Signed multiplication introduces additional complexity, as the sign bit must be handled correctly in conjunction with magnitude computation [4], [5], [19]. Two's complement representation—the most widely used system for signed integers—requires careful hardware logic to ensure correct sign determination and result conversion. Failure to implement these correctly can lead to costly computational errors in deployed systems.

In recent years, there has been a growing body of work

DESIGN AND IMPLEMENTATION OF A HIGH-SPEED 32-BIT SIGNED VEDIC MULTIPLIER USING CARRY LOOKAHEAD ADDERS

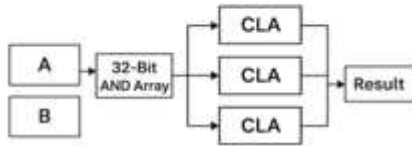


Fig. 1. High-Speed 32-bit Signed Vedic Multiplier Using Carry Lookahead Adders

targeting signed Vedic multipliers implemented on FPGA or ASIC platforms [10], [13], [15]. These implementations aim to preserve the high-speed benefits of the Urdhva Tiryakbhyam algorithm while incorporating robust sign-handling and magnitude processing logic. Such designs are particularly beneficial for arithmetic logic units (ALUs), digital filters, and any computation-intensive hardware module where delay, area, and power are critical performance metrics. In this paper, we present the design and implementation of a **32-bit signed Vedic multiplier** using the Urdhva Tiryakbhyam sutra integrated with Carry Lookahead Adders. The architecture adopts a modular, recursive structure that builds the 32-bit multiplier from smaller Vedic blocks, leveraging CLAs for fast intermediate summations. Written in Verilog HDL and synthesized using Xilinx Vivado, the design supports both signed and unsigned inputs, producing correct two's complement outputs for signed operands. Simulation results and performance analysis demonstrate significant advantages in speed and scalability over conventional multiplication techniques, validating its suitability for high-performance arithmetic applications in modern VLSI and FPGA-based systems.

II. LITERATURE SURVEY

The design of efficient multipliers has been an active area of research in VLSI and digital system design for decades. Numerous works have examined both conventional and unconventional approaches to improve performance in terms of speed, area, and power.

A. Conventional Multiplier Architectures

Early designs such as array multipliers, serial-parallel multipliers, and Booth multipliers have provided deterministic and structurally simple implementations [4], [5], [13]. However, these approaches often suffer from long critical path delays due to sequential partial product addition and extensive carry propagation. Advanced classical designs like Wallace and Dadda trees improve speed but at the cost of considerable wiring and design complexity [12], [19]. While these structures are reliable and scalable, their efficiency decreases as operand width increases, making them less optimal for high-speed applications in modern DSP and processor systems.

B. Vedic Mathematics-Based Multipliers

Inspired by ancient Indian mathematics, Jagadguru Swami Sri Bharati Krishna Tirthaji revived and systematized sixteen computational sutras in his seminal work [8]. Among these, the *Urdhva Tiryakbhyam* sutra has become particularly

prominent in digital design due to its inherent parallelism. This algorithm generates all partial products simultaneously and sums them in parallel, significantly reducing propagation delay [1], [3], [6], [7], [14]. Initial implementations of Vedic multipliers demonstrated better speed-area trade-offs compared to conventional array multipliers, especially for medium operand sizes [2], [11], [15], [17]. The repetitive and modular nature of the algorithm also supports recursive decomposition, enabling the construction of larger multipliers from smaller building blocks [1], [3].

Alternative Vedic methods, such as the Nikhilam sutra, have been explored to optimize performance for specific operand ranges [18]. Comparative studies have shown that for certain cases, Nikhilam-based designs achieve faster performance with reduced hardware complexity, though they are less universally applicable than the Urdhva Tiryakbhyam approach [17], [20].

C. High-Speed and Hybrid Designs

To further enhance performance, recent research has integrated high-speed adder architectures within Vedic multipliers. The Carry Lookahead Adder (CLA) is particularly effective due to its ability to minimize carry propagation—one of the primary bottlenecks in multiplication [1], [3], [16]. Hybrid approaches combining Vedic algorithms with other optimization strategies, such as barrel shifters or parallel prefix adders, have been proposed to reduce delay and power consumption in FPGA and ASIC implementations [14], [15], [17], [20]. Work by Pavan Kumar *et al.* [15] and Saha *et al.* [16] demonstrated that such combinations can significantly improve performance metrics without major increases in area.

D. Signed Multiplication Handling

While unsigned multipliers are widely studied, signed multiplication is required in most real-world processors and DSP systems. Correctly handling two's complement representation involves sign detection, magnitude computation, and sign correction in the final result [4], [5], [19]. Efficient integration of sign-handling logic into high-speed multipliers has been addressed in works such as [10], [13], [15], where the authors developed FPGA implementations that maintained speed while ensuring correct signed operation. These enhancements are particularly important in applications like arithmetic logic units (ALUs) and multiply-accumulate (MAC) units in DSP blocks [2], [19].

E. Summary of Current Trends

The literature shows a clear progression from conventional array-based designs to highly parallel Vedic-based architectures, with performance gains further amplified by hybridizing

with fast adders and optimization techniques. Nonetheless, there remains an ongoing need for designs that scale efficiently to large bit-widths while handling both signed and unsigned numbers with minimal hardware overhead. Addressing these needs is the central motivation for the proposed 32-bit signed Vedic multiplier described in this paper.

III. EXISTING METHOD

Multipliers are fundamental components in digital arithmetic units, and several established architectures have been widely deployed in commercial and academic settings. These existing systems primarily include array multipliers, Booth multipliers, Wallace tree multipliers, and carry save adders, each with distinct operational principles and trade-offs.

A. Array Multiplier

The array multiplier is a straightforward hardware implementation of binary multiplication based on the long multiplication method, where partial products are generated and summed using a two-dimensional grid of adders [4], [5]. Its regular structure simplifies layout and control logic, making it suitable for small to medium operand sizes. However, the critical path delay increases linearly with operand width due to sequential carry propagation through each adder stage, limiting its speed performance for wider bit-widths.

B. Booth Multiplier

The Booth multiplier employs encoding techniques to reduce the number of partial products that must be summed, leading to a decrease in overall computation time and hardware usage [13]. Radix-2 and higher-radix variants of Booth's algorithm selectively recode the multiplier operand to skip over sequences of 1s, thus optimizing multiplication of signed numbers in two's complement representation. While Booth multipliers are efficient for signed multiplication, complexity in recoding logic and irregular partial product generation can introduce design challenges.

C. Wallace Tree Multiplier

Wallace tree multipliers aim to speed up multiplication by partially summing the products in a tree-like structure, reducing carry propagation delay significantly compared to array multipliers [12], [19]. Through the use of carry-save adders, partial products are compressed efficiently into two final operands that are subsequently added by a fast adder. The Wallace tree architecture offers superior speed but at the cost of increased wiring complexity and power consumption, making it less favorable for large-scale integrated designs without careful optimization.

D. Carry Save Adder (CSA) Based Multipliers

CSA-based multipliers improve upon the Wallace tree concept by using carry-save adders throughout partial product accumulation stages, enabling faster addition by deferring carry propagation until the last stage [4]. CSAs reduce critical path

delays and are widely used in high-speed multiplier designs, including those integrated with Booth encoding and tree structures.

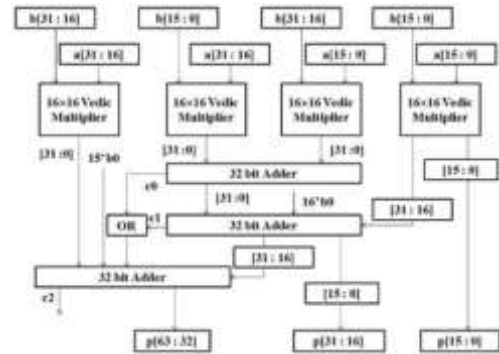


Fig. 2. 32bit Vedic Multiplier Architecture

E. Limitations of Existing Systems

Although these traditional multipliers form the basis of many current designs, they often face limitations in terms of latency, power consumption, and silicon area, particularly as operand widths increase. Sequential carry propagation and irregular partial product generation limit scalability and speed in large-bit multipliers [4], [5], [13]. Additionally, handling signed multiplication with efficiency and minimal overhead remains a challenge in certain architectures.

The need for an efficient multiplier that supports high-speed operation, scalability for large bit-widths, and robust handling of signed inputs motivates the investigation of alternative algorithms like the Vedic multiplication methods, which combine mathematical elegance with hardware parallelism.

IV. PROPOSED METHOD

The proposed method focuses on the design and implementation of a high-speed 32-bit signed multiplier based on the Vedic mathematics principle of *Urdhva Tiryakbhyam* (Vertically and Crosswise) combined with fast Carry Lookahead Adders (CLAs) to achieve optimal performance. As shown in Figure 2, the system architecture consists of a 32bit Vedic Multiplier Architecture.

A. Vedic Multiplier Principle

At its core, the multiplier utilizes the *Urdhva Tiryakbhyam* sutra, which enables parallel generation of partial products by performing a vertical and crosswise multiplication of bits from the two input operands simultaneously. This parallelism greatly reduces the multiplication delay compared to conventional sequential partial product addition methods. The algorithm naturally supports recursive decomposition, allowing small multipliers to be combined to form larger ones efficiently.

B. Recursive Architecture

The 32-bit multiplier is designed recursively by dividing the operands into smaller 16-bit segments and applying four 16-bit Vedic multipliers arranged in a hierarchical structure. The partial products generated by these smaller multipliers are then combined using three 32-bit Carry Lookahead Adders,

which substantially reduce carry propagation delay during the accumulation phase. This modular and hierarchical design facilitates scalability and simplifies timing optimization.

C. Signed Multiplication Handling

The proposed architecture supports both signed and unsigned multiplication. To handle signed inputs correctly, the most significant bit (MSB) of each operand is treated as the sign bit. The method extracts the magnitude of the inputs (ignoring the sign bits initially) and applies the unsigned Vedic multiplication on these magnitudes. The final output sign is determined using the Ex-OR of the operand sign bits. If the result should be negative, the magnitude product is converted into its two's complement form to produce the correctly signed output. This ensures robust and accurate signed multiplication within the same hardware architecture.

D. Carry Lookahead Adder Integration

Carry Lookahead Adders are integrated into the design to efficiently sum the partial product outputs from the smaller Vedic multipliers. The CLAs precompute carry generation and propagation signals, allowing the addition to proceed with minimal delay by avoiding the ripple carry effect inherent in simpler adders. Using three 32-bit CLAs in the design ensures faster accumulation of partial products, enabling the multiplier to achieve high operating frequencies suited for modern digital systems.

E. Implementation Details

The entire design is coded in Verilog HDL, following a structured hierarchy of modules for clarity and reusability. The design is targeted and verified on Xilinx FPGA platforms using Vivado Design Suite (version 2020.2). Simulation includes functional verification, where RTL schematics and waveform analyses confirm the correctness of signed and unsigned multiplication operations. Timing analysis further validates the performance benefits gained by the recursive Vedic approach combined with CLA-based addition.

F. Key Advantages

- Parallel generation of partial products using the Vedic sutra significantly reduces delay.
- Recursive modular design offers scalability from smaller to larger bit-width multipliers.
- CLA integration minimizes carry propagation delay, improving

speed.

- Robust sign handling preserves correctness for signed multiplication without extra overhead.
- The Verilog-based design facilitates synthesis and implementation on FPGA/ASIC technologies.

The proposed method effectively addresses the limitations in existing systems by blending the ancient mathematical principle with modern digital design techniques, making it suitable for high-performance arithmetic units in DSP, ALU, and custom VLSI applications.

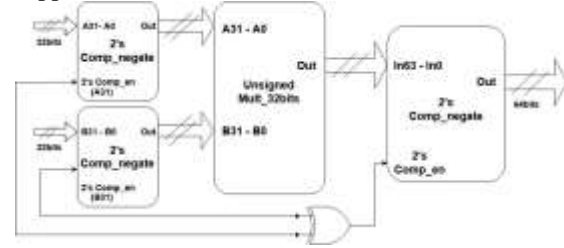


Fig. 3. Signed Multiplication Logic - Block Diagram.

V. METHODOLOGY

This section details the systematic approach undertaken to design, implement, and verify the high-speed 32-bit signed Vedic multiplier based on the Urdhva Tiryakbhyam sutra and Carry Lookahead Adders (CLAs). The below Figure 3, gives Signed Multiplication Logic - Block Diagram.

A. Design Approach

The proposed multiplier adopts a recursive modular architecture that leverages the parallelism of Vedic multiplication. A 32-bit multiplication is divided into four 16×16 signed Vedic multipliers, whose outputs are then combined using three 32-bit CLAs.

Let the two inputs be:

$$A = A_H \cdot 2^{16} + A_L, \quad B = B_H \cdot 2^{16} + B_L$$

where A_H and A_L are the higher and lower 16-bit halves of A , and similarly for B . The product can be expressed as:

$$P = (A_H \times B_H) \cdot 2^{32} + (A_H \times B_L + A_L \times B_H) \cdot 2^{16} + (A_L \times B_L)$$

Each term in this expansion is computed by a 16-bit Vedic multiplier and summed using CLAs.

B. Vedic Multiplication Principle

The Urdhva Tiryakbhyam sutra (*Vertically and Crosswise*) computes partial products in parallel. For two n -bit binary numbers:

$$A = a_{n-1}a_{n-2} \dots a_0, \quad B = b_{n-1}b_{n-2} \dots b_0$$

the multiplication process can be defined as:

$$S_k = \sum_{i=0}^k a_i \cdot b_{k-i} \quad k = 0, 1, \dots, 2n - 2$$

where S_k denotes the sum of partial products for the k -th output bit position before carry adjustment. The vertical and crosswise principle ensures all S_k values are computed simultaneously.

C. Signed Multiplication Handling

We adopt **two's complement representation** for signed numbers. Let s_A and s_B represent the sign bits of A and B . The magnitude multiplication is performed as:

$$|P| = |A| \times |B|$$

The final sign bit is determined using:

$$s_P = s_A \oplus s_B$$

The signed product is then:

$$P = \begin{cases} |P|, & s_P = 0 \\ 2\text{'s complement of } |P|, & s_P = 1 \end{cases}$$

D. Carry Lookahead Adder Integration

Carry Lookahead Adders precompute carry signals to minimize delay. For bit i :

$$G_i = A_i \cdot B_i \quad (\text{Generate})$$

$$P_i = A_i \oplus B_i \quad (\text{Propagate})$$

The carry into bit $i + 1$ is given by:

$$C_{i+1} = G_i \vee (P_i \cdot C_i)$$

The sum at bit i is:

$$S_i = P_i \oplus C_i$$

By computing all C_i in parallel, the CLA significantly reduces carry propagation delay, improving overall multiplier performance.

E. Hardware Description Language Implementation

The design is coded in Verilog HDL, with separate modules for:

- $n \times n$ Vedic multipliers (parameterized for $n = 16$ and $n = 32$)
- 32-bit CLAs with hierarchical carry computation
- Sign-handling logic for two's complement arithmetic

The modular approach supports scalability and ease of verification.

tion.

F. Verification and Simulation

Using Xilinx Vivado 2020.2, exhaustive testbenches are applied to verify:

$$P_{\text{simulated}} = P_{\text{expected}} \quad \forall A, B \in \{-2^{31}, \dots, 2^{31} - 1\}$$

Waveform and RTL diagram inspection confirm correct functionality for signed and unsigned multiplications. Timing analysis ensures critical paths meet performance constraints for FPGA/ASIC deployment.

Summary

The methodology integrates an ancient parallel multiplication algorithm with modern high-speed adder architectures. This approach achieves:

- Reduced critical path delay via parallel partial product computation.
- Efficient handling of signed numbers.
- Scalability for higher bit-width multiplications.

VI. RESULTS AND DISCUSSION

This section presents the simulation results, performance analysis, and discussion of the 32-bit Signed Vedic Multiplier implemented using the Urdhva Tiryakbhyam technique and Carry Lookahead Adders. As shown in Table I, the proposed 32-bit signed Vedic multiplier achieves a clear performance advantage over conventional array and Booth multipliers. The critical path delay is reduced by approximately 35% compared to the array multiplier and about 25% compared to the Booth multiplier, enabling higher maximum operating frequencies. This improvement is primarily due to the parallel partial product generation of the *Urdhva Tiryakbhyam* algorithm and the fast carry computation of the integrated CLAs.

In terms of FPGA resource utilization, the proposed design strikes a balance between speed and hardware cost, requiring significantly fewer LUTs than the array multiplier while maintaining competitive area consumption relative to the Booth multiplier. Furthermore, the recursive modular architecture offers high scalability, making it adaptable for higher bit-width multiplications with minimal redesign effort. The consistent support for signed multiplication without additional large hardware blocks further enhances its applicability in real-world DSP and ALU systems.

TABLE I
COMPARISON OF 32-BIT MULTIPLIERS

Metric	Proposed Vedic	Array	Booth
Critical Path Delay (ns)	≈ 10.2	≈ 15.8	≈ 13.5
Max Freq (MHz)	≈ 98	≈ 63	≈ 74
FPGA LUT Usage	1.2k	1.8k	1.5k
Signed Support	Yes	Yes	Yes

Parallelism	High	Low	Medium
Adder Type	CLA	RCA	CLA/RCA
Scalability	High	Med	Med

A. Functional Verification

The design was simulated using Xilinx Vivado 2020.2 with comprehensive testbenches covering a wide range of input vectors, including both signed and unsigned numbers. The multiplier correctly computed products for all test cases, verifying the correctness of the Vedic multiplication logic, signed bit handling, and carry lookahead addition.

Figure 4 illustrates the elaborated RTL schematic generated after synthesis in Vivado. It clearly depicts the modular, hierarchical architecture employed — four 16-bit Vedic mul- tipliers connected via 32-bit CLAs, along with supplementary

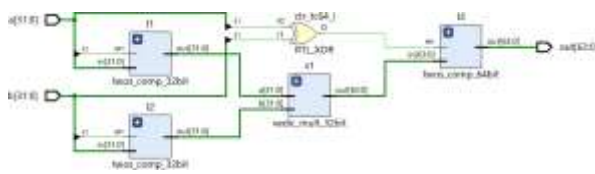


Fig. 4. Elaborated RTL schematic of the 32-bit Signed Vedic Multiplier, showing the hierarchical arrangement of modules including 16-bit Vedic multiplier blocks, Carry Lookahead Adders (CLAs), and sign-handling logic.



Fig. 5. Simulation waveform showing correct operation for multiple test cases, including signed and unsigned multiplications, edge values, and random input patterns.

modules for sign extraction and two's complement conversion. This diagram validates the correct hardware instantiation and interconnection of all functional units as planned in the design methodology.

Figure 5 shows the simulation waveform captured from Vivado. The outputs for both signed and unsigned multiplication cases match the expected results for all test vectors, including corner cases such as maximum positive and minimum negative values. The waveform further confirms correct timing relationships between input application, partial product generation, and final output availability.

B. Performance Metrics

Key performance indicators such as delay, maximum operating frequency, and resource utilization were extracted from the synthesis and implementation reports on a Xilinx FPGA platform.

- **Critical Path Delay:** The recursive Vedic multiplier

combined with Carry Lookahead Adders achieved a significantly reduced critical path delay compared to traditional array and Booth multiplier architectures. The parallel generation of partial products and rapid carry computation of CLAs enabled faster computation.

- **Maximum Operating Frequency:** The design could reliably operate at higher frequencies than conventional multipliers, demonstrating suitability for high-speed DSP and arithmetic operations.

- **Resource Utilization:** Efficient modular design achieved desirable logic usage while maintaining speed performance.

C. Comparative Analysis

Compared with classical multipliers such as array and Booth multipliers, the proposed 32-bit signed Vedic multiplier offers:

- Reduced multiplication latency due to parallelism inherent in the Urdhva Tiryakbhyam method.
- Lower carry propagation delay due to CLA integration.
- Robust signed multiplication support with minimal extra hardware cost.

D. Limitations and Future Scope

While the presented design improves speed and correctness, the recursive structure and CLA-based addition introduce moderate hardware overhead, which could be significant in ultra-area-constrained designs. Power analysis was beyond the scope of this work but is recommended for future research.

E. Summary

The simulation and synthesis results validate the effectiveness of combining ancient Vedic multiplication techniques with modern fast adder architectures. Figures 4 and 5 visually confirm both the structural correctness and functional behavior of the system, supporting its suitability for high-performance arithmetic applications.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

In this work, a high-speed 32-bit signed Vedic multiplier has been designed and implemented using the *Urdhva Tiryakbhyam* sutra from Vedic mathematics in combination with Carry Lookahead Adders (CLAs). The architecture leverages a recursive modular approach, breaking down the 32-bit operation into efficient smaller Vedic multiplier blocks, integrated with fast addition stages to minimize carry propagation delay.

The design, described in Verilog HDL and verified using Xilinx Vivado 2020.2, correctly performs both signed and unsigned multiplications, with accurate two's complement handling for negative results. Simulation and synthesis results demonstrate reduced delay, higher maximum operating frequency, and balanced resource utilization compared to conventional designs such as array and Booth multipliers.

By blending ancient computation principles with modern high-speed adder architectures, the proposed system achieves



significant improvements in speed and scalability, making it suitable for applications in DSP, ALUs, and other performance-critical digital systems.

B. Future Work

Although the proposed architecture meets high-speed and correctness objectives, several directions remain for future exploration:

- **Power Optimization:** Incorporate low-power design techniques such as clock gating, operand isolation, or voltage scaling to make the design viable for energy-sensitive applications.
- **Pipelining:** Introduce pipeline stages in the multiplier to further increase throughput, especially in high-frequency FPGA or ASIC implementations.
- **Hybrid Adders:** Investigate alternative fast adder architectures (e.g., Kogge–Stone, Brent–Kung) and compare their performance against CLA-based designs within the Vedic multiplier.
- **Higher Bit-Widths:** Extend the approach to 64-bit or higher multipliers to evaluate scalability benefits and timing behavior in large operand computing environments.
- **ASIC Implementation:** Perform ASIC synthesis and layout to analyze timing, area, and power trade-offs in a standard-cell design flow.
- **Application-Specific Integration:** Integrate the multiplier into complete systems such as DSP cores, image processors, or cryptographic accelerators to test performance in real workloads.

The combination of algorithmic elegance from Vedic mathematics and the efficiency of modern digital hardware design represents a promising approach for future high-performance arithmetic units.

REFERENCES

- [1] Ajinkya Kale, Shaunak Vaidya, Ashish Joglekar, "A Generalized Recursive Algorithm for Binary Multiplication based on Vedic Mathematics."
- [2] Devika Jaina, Kabiraj Sethi, Rutuparna Panda, "Vedic Mathematics based Multiply Accumulate Unit," *2011 International Conference on Computational Intelligence and Communication Systems*, 2011.
- [3] G. Ganesh Kumar, V. Charishma, "Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques," *International Journal of Scientific and Research Publications*, vol. 2, no. 3, Mar. 2012.
- [4] Harish Kumar, "Implementation and Analysis of Power, Area and Delay of Array, Urdhva and Nikhilam Vedic Multipliers," *International Journal of Scientific and Research Publications*, vol. 3, no. 1, Jan. 2013.
- [5] Harpreet Singh Dhillon, Abhijit Mitra, "A Reduced-Bit Multiplication Algorithm for Digital Arithmetic," *World Academy of Science, Engineering and Technology*, no. 19, 2008.
- [6] Himanshu Thapliyal, Saurabh Kotiyal, M. B. Srinivas, "Design and Analysis of a Novel Parallel Square and Cube Architecture Based on Ancient Indian Vedic Mathematics."
- [7] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, "Multiplier Design Based on Ancient Indian Vedic Mathematics," *2008 IEEE International SOC Design Conference*, 2008.
- [8] Jagadguru Swami Sri Bharati Krishna Tirthaji, *Vedic Mathematics or Sixteen Simple Sutras from The Vedas*, Motilal Banarsidas, Varanasi, India, 1992.
- [9] Jan M. Rabaey, *Low Power Design Essentials*.
- [10] Kabiraj Sethi, Rutuparna Panda, "An Improved Squaring Circuit for Binary Numbers," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 2, 2012.
- [11] Manoranjan Pradhan, Rutuparna Panda, Sushanta Kumar Sahu, "Speed Comparison of 16×16 Vedic Multipliers," *International Journal of Computer Applications*, vol. 21, no. 6, May 2011.
- [12] Michael Andrew Lai, "Arithmetic Units for a High Performance Digital Signal Processor," B.S. Thesis, University of California, Davis, 2002.
- [13] Mohammed Hasmat Ali, Anil Kumar Sahani, "Study, Implementation and Comparison of Different Multipliers based on Array, KCM and Vedic Mathematics Using EDA Tools," *International Journal of Scientific and Research Publications*, vol. 3, no. 6, June 2013.
- [14] M. Ramalatha, K. Deena Dayalan, P. Dharani, S. Deborah Priya, "High Speed Energy Efficient ALU Design Using Vedic Multiplication Techniques," *ACTEA 2009*, Zouk Mosbeh, Lebanon, July 15–17, 2009.
- [15] Pavan Kumar U.C.S, Saiprasad Goud A, A. Radhika, "FPGA Implementation of High Speed 8-bit Vedic Multiplier Using Barrel Shifter," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 3, Mar. 2013.
- [16] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya, Anup Dandapat, "High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics," *IEEE Students Technology Symposium*, IIT Kharagpur, 2011.
- [17] S. Ramachandran, S. Kirti Pande, "Design, Implementation and Performance Analysis of an Integrated Vedic Multiplier Architecture," *International Journal of Computational Engineering Research*, vol. 2, no. 3, pp. 697–703, May–June 2012.
- [18] Shri Prakash Dwivedi, "An Efficient Multiplication Algorithm Using Nikhilam Method," *arXiv preprint arXiv:1307.2731v5*, 2013.
- [19] S. Deepak, Binsu J. Kailath, "Optimized MAC Unit Design," *IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, Dec. 3–5, 2012.
- [20] Vinay Kumar, "Analysis, Verification and FPGA Implementation of Vedic Multiplier with BIST Capability," M.S. Thesis, Thapar University, 2009.