Development of Energy Efficient and High Speed Vedic Multiplier using Enhanced Adder Structure for Next Generation Arithmetic Logic Unit Design

Arvind Bhaskar, *Dr. Priyanka Goyal
School of Information & Communication Technology
Gautam Buddha university, Greater Noida, India.
*Corresponding Author: Priyankag@gbu.ac.in

ABSTRACT

The need for optimized arithmetic units has been underscored by the escalating demand for high-speed and energy-efficient digital systems, particularly in the design of Arithmetic Logic Units (ALUs). This study presented a high-speed and low-power Vedic multiplier using an enhanced adder structure suitable for next-generation ALU applications. The proposed design is based on the Urdhva Tiryakbhyam algorithm, which allows parallel generation of partial products and thus reduces computational delay. A modular Vedic multiplier architecture is realized and merged with a hybrid enhanced adder to minimize carry propagation delay as well as switching activity. The entire design is described using Verilog HDL and synthesized in Xilinx Vivado for performance evaluation. Results show that the proposed design attains a delay of 24.8 ns at an operating power of 190 mW, which is better than conventional multipliers by about 45% in delay, 40% in power consumption, and up to 67% improvement in Power Delay Product (PDP). The enhanced hybrid adder reduces this further to 6.1 ns with lower power consumption; hence such an architecture becomes more appropriate when looking for high-performance energy-efficient ALU systems.

Keywords— Vedic Multiplier, Urdhva Tiryakbhyam, Enhanced Adder Structure, Low Power VLSI Design, Arithmetic Logic Unit (ALU)

I. INTRODUCTION

The modern digital systems that are rapidly evolving, including signal processing, artificial intelligence, and high-performance computing, have greatly enhanced the requirements of arithmetic units which are fast and energy efficient [1,2]. These systems are centred around the ALU that executes the most basic functions, including addition, subtraction, and multiplication. Multiplication is one of the most significant and computationally expensive of these operations, and it can be the most important in the total system performance and power draw [3]. Hence, the development of high-speed and low power multipliers has emerged as one of the major areas of research in VLSI system design [4].

Conventional multiplier architectures, including array and Booth multipliers, have been found to have an increased propagation delay and complexity with increased hardware complexity with higher bit widths [5]. In order to overcome these weaknesses, in recent years Vedic multiplication procedures, rooted in ancient Indian mathematics, have been of strong interest. Particularly the Urdhva-Tiryagbhyam (vertical and crosswise) sutra enables the production of the partial products in parallel manner thereby, reducing the computational delay and increasing the throughput [6,7]. It is natural and results in Vedic multipliers being very suitable in fast applications such as in digital signal processing (DSP) and image processing, and embedded systems. Other than the multiplication technique, the effectiveness of the adder structures to add the partial products is also a significant factor in determining the performance of a multiplier. As addition operations are a substantial part of ALU work, better speed and energy efficiency requires better adder design optimization [8,9]. The enhanced adder design, such as Carry Select Adders (CSA), Carry Look-Ahead Adders (CLA) and hybrid adders are aimed at offering better performance with reduced carry propagation delay [10]. However, each kind of adder has tradeoffs between speed, power consumption and area, and will need to be chosen and optimized to result in an optimized system design.

Recent research has been founded on combination of Vedic multiplication with enhanced adder design [11]. Optimized adders like speed-efficient Carry Select Adders are used and delay and power consumption are greatly reduced. Improved Vedic multipliers based on adders have demonstrated significant improvements in delay and power over standard designs, and therefore are applicable to the next generation in the implementation of ALU designs [12].

Furthermore, with the growing need of energy-efficient computation, particularly in portable and battery-powered computers, the development of low-power consumption arithmetic circuits is needed as never before. The use of technologies such as parallel computing, logical optimization and the efficient use of hardware have been massively applied to minimize power consumption without compromising performance [13]. There is a need therefore to design a better architecture that facilitates quick and energy efficient computing and in that sense, it would be a viable option to design a faster and consuming less energy Vedic multiplier in accordance to the better adder structure [14].

Finally, Vedic multiplication algorithms combined with modern adder designs are a promising method of high-performance ALUs. This study aims at creating a high-speed, energy-efficient Vedic multiplier based on an improved adder structure, to achieve the best trade-offs between speed, power, and area, and to make a contribution to the next-generation design of arithmetic circuits. Here are the research objectives as shown below:

- To design a high-speed Vedic multiplier based on the Urdhva Tiryakbhyam algorithm for efficient parallel computation.
- To develop a modular multiplier architecture using smaller building blocks for improved scalability and reduced hardware complexity.
- To implement an enhanced hybrid adder structure for minimizing carry propagation delay and reducing power consumption.
- To evaluate the performance of the proposed design in terms of delay, power consumption, area utilization, and Power Delay Product (PDP) using HDL and FPGA-based synthesis tools.
- To compare the proposed architecture with conventional multipliers and validate its suitability for next-generation ALU and VLSI system applications.

II. REVIEW OF LITERATURE

Recent literature has talked of high-speed and energy efficient multipliers and ALU architectures using Vedic mathematics and optimized circuit methods. Dabi et al. (2025) [16] introduced a Vedic-based ALU using low-power design principles as part of it, demonstrating a remarkable reduction of 93.983 percent of power dissipation and 66.479 percent of delay relative to traditional designs, showing its suitability to battery-powered equipment. Rao et al. (2024) [17] compared Vedic multiplier 4 bits with GDI, CMOS, and Transmission Gate technologies with various adder architectures like RCA, CLA, and CSA; the results reveal that GDI-based designs are the best in delay, area, and power efficiency. Kumar et al. (2026) [18] also applied the Vedic principles to floating-point computation by implementing a 53-bit mantissa multiplier inspired by the Urdhva Tiryakbhyam sutra that incurred lower delay and reduced hardware usage especially when used with CNNs with FPGA designs. A hybrid architecture of Modified Vedic Nikhilam with Karatsuba algorithm with FCSA and KSA advanced adders was introduced by Satyaka et al. (2026) [19] that offered a 30% speed improvement and a 25% power reduction, which is highly efficient in real-time application. Bathula et al., (2026) [20] additionally in 2026 explored emerging nanomagnetic logic (pNML) for implementing Vedic multipliers which provided ultra-low energy consumption at femtojoule levels proving scalability towards future nanoelectronic systems.

Additional research focuses on both architectural and logic level optimization methods in order to improve the performance of the multiplier and the ALU. A hybrid adder-based Vedic multiplier was suggested by Sangeeth et al. (2023) [21], and it is integrated with a RISC processor, resulting in lower delay, the minimization of power consumption, and simplified architecture. Giridaran et al. (2022) [22] concentrated on adiabatic reasoning with the help of Enhanced ECRL to reduce power dissipation, and recycle energy, providing better results in terms of transistor counts and switching efficiency. Mendez et al. (2022) [23] used pre-computation in Vedic multipliers to pre- compute carries, and in effect, increase speed and lessen the delay with 90 nm technology. Thamizharasan et al. (2025) [24] presented reversible logic-based multiplier with significant speed improvements over the conventional multiplier like Wallace, Booth and Vedic based designs and energy efficiency because of preservation of information. Previous investigations of Vedic multipliers in FFT systems by Dharani et al. (2021) [25] revealed that Vedic multipliers lessen power consumption by 11.24 percent and delay by 5.28 percent by using optimised compressor and reversible logic methods. Altogether, these researches show that the combination of Vedic mathematics with high-quality adder designs, new technologies and low-power design techniques is also very important in creating next-generation high-performance and energy-efficient ALU systems.

Although there have been major improvements in Vedic multiplier and ALU design, the current research is mainly centered on either refinement of the algorithm, new technologies, or refinement of the adder. There is little literature on the joint design of a scalable Vedic multiplier with an optimized hybrid adder architecture to improve balanced speed, power, and area. Also, the assessment of ALU architectures is inadequate, which involves the necessity of an effective combined strategy.

III. RESEARCH METHODOLOGY

Fig. 1 shows the proposed methodology to design the high-speed and low-energy Vedic multiplier-based ALU. It provides a systematic approach of starting from selecting the Urdhva Tiryakbhyam method and going through designing the architecture of the Vedic multiplier and an improved adder circuit. The designed multiplier and adder circuits are implemented in the ALU, and the ALU design is modeled in HDL, which is further validated on the basis of its performance.

ISBN: 978-81-985770-0-9

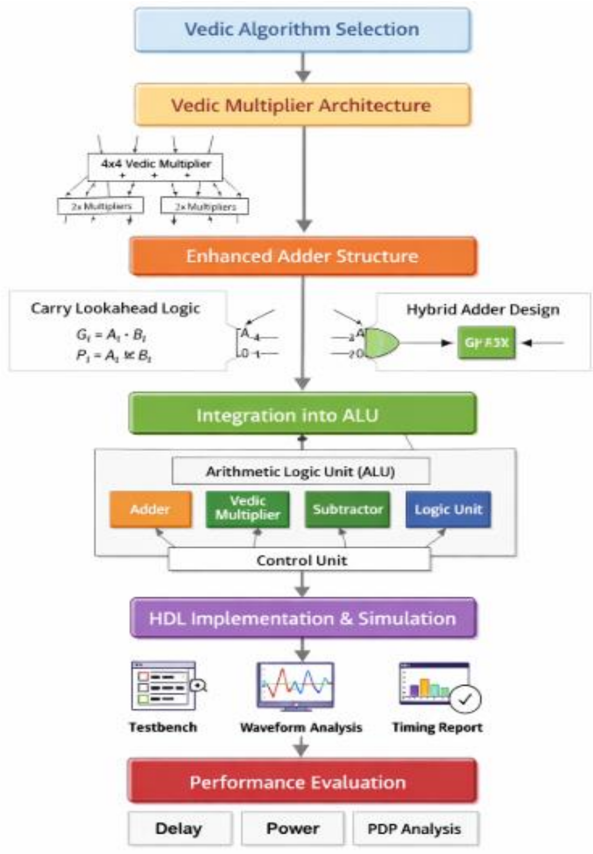

Fig. 1. Proposed Methodology

A. *Selection of Vedic Algorithm*

An important stage in the process of designing high-speed and energy-efficient arithmetic units is the choice of an efficient multiplication algorithm. The Urdhva Tiryakbhyam (vertical and crosswise) sutra of Vedic mathematics is borrowed in this work because of its parallel computing nature and its routine form which is quite useful to performance in the VLSI implementation. The Urdhva Tiryakbhyam algorithm calculates all the partial products concurrently by a vertical and crosswise structure, unlike the traditional multiplication algorithms like array multiplication or Booth multiplication which compute the partial products sequentially and accumulate them, so the propagation delays are minimized.

From a theoretical perspective, consider two n-bit binary numbers:

$$A = (a_{n-1}a_{n-2} \dots a_1a_0), B = (b_{n-1}b_{n-2} \dots b_1b_0) \quad (1)$$

The multiplication result $P = A \times B$ can be expressed as:

$$P = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (a_i \cdot b_j) \cdot 2^{i+j} \quad (2)$$

The partial products are then produced in sequence and added together using shifting operations in conventional multiplication. Nevertheless, with the Urdhva Tiryakbhyam method, parallel generation is done of partial products through crosswise combinations. As an instance, in a 2x2 multiplication:

$$A = a_1a_0, B = b_1b_0 \quad (3)$$

The output bits are computed as:

- Least Significant Bit (LSB):

$$P_0 = a_0 \cdot b_0$$

- Middle bit (crosswise operation):

$$P_1 = (a_1 \cdot b_0) + (a_0 \cdot b_1)$$

- Most Significant Bit (MSB):

$$P_2 = a_1 \cdot b_1 + \text{carry from } P_1$$

This crosswise computation can be generalized for higher bit-widths, where each output bit is formed by summing all partial products whose indices satisfy $i + j = k$, where k eqis the bit position of the result:

$$P_k = \sum_{i+j=k} (a_i \cdot b_j) + \text{carry} \quad (4)$$

This formulation allows parallel creation of partial products and minimizes the delay of the critical path relative to sequential techniques. Parallel execution ensures a much faster computational delay since more than one multiplication and addition is done at a time instead of one at a time. Hardware wise, this parallelism will be translated into low latency and high throughput. Also, the periodic and repetitive nature of the algorithm facilitates layout design in VLSI, and is very scalable to larger bit-width multipliers (8×8 and 16×16 and more). The low sequential stages also help in reducing switching activity that reduces dynamic power consumption. Urdhva Tiryakbhyam algorithm is thus selected as the structure of the proposed multiplier design since it possesses good theoretical basis in parallel computation, good use of hardware resources, and it can execute at both high-speed and low-power which is very useful in the next-generation ALU architectures.

B. *Design of Vedic Multiplier Architecture*

The hierarchical and modular basic architecture applied in the Vedic multiplier circuit design facilitates scalability, simplicity and efficient hardware utilization. Urdhva Tiryakbhyam algorithm is utilized by computing multiplication by parallel production of partial products and accumulation. Rather than using complex design on large multiplier, smaller sub-block such as the 2 x 2 multipliers are repeated to form a larger multiplier as 4 x 4, 8 x 8, and finally 16 x 16 multipliers.

From a theoretical perspective, consider two n-bit numbers split into two halves:

$$A = A_H \cdot 2^{n/2} + A_L, B = B_H \cdot 2^{n/2} + B_L \quad (5)$$

where:

ISBN: 978-81-985770-0-9

A_H, B_H are higher-order bits

A_L, B_L are lower-order bits

The multiplication can be expressed as:

$$P = A \times B = (A_H B_H) \cdot 2^n + (A_H B_L + A_L B_H) \cdot 2^{n/2} + (A_L B_L) \qquad (6)$$

Hierarchy multiplier is designed based on this equation. As an example, a 4x4 multiplier may be built using four 2x2 multipliers and adders to sum up the partial results. The individual smaller multipliers compute partial products which are aligned (shifted) and add them together with efficient adder circuits.

In Vedic architecture, partial products are generated using vertical and crosswise operations. For a 4×4 multiplier, the output bits can be generalized as:

$$P_k = \sum_{i+j=k} (a_i \cdot b_j), k = 0,1,2,\dots,2n-1 \qquad (7)$$

This makes sure that all the combinations of multiplications of bits that contribute to a specific position of the output are calculated concurrently. Carry propagation is important in the overall delay of the accumulation of these partial products, which are carried out by adders. The hierarchical arrangement greatly minimizes the critical path delay due to the existence of lots of smaller multipliers that run in parallel instead of one large sequential arrangement. Also, pipelining and parallel processing can be done, which enhances the throughput even more. The routing and layout of hardware design are also easier with the regular structure, resulting in improved area optimization.

The modular design is more hardware efficient, by eliminating redundant computations and enabling smaller multiplier blocks to be reused. This does not only reduce the complexity of the design but also increases fault tolerance and testing. Moreover, the architecture allows parallel operation and thus minimizes switching in sequential stages and hence also helps in minimizing dynamic power consumption. To sum up, the Vedic multiplier architecture is based on the principles of mathematical decomposition and parallel computation to provide high speed, scalability and low energy consumption. The hierarchical design, the effective partial product generation and accumulation make it to be very appropriate in incorporation into next-generation ALU systems. Fig. 2 illustrates that the proposed 4×4 Vedic multiplier is built out of smaller 2x2 multiplier modules and adders to allow partial product accumulation to be done efficiently.

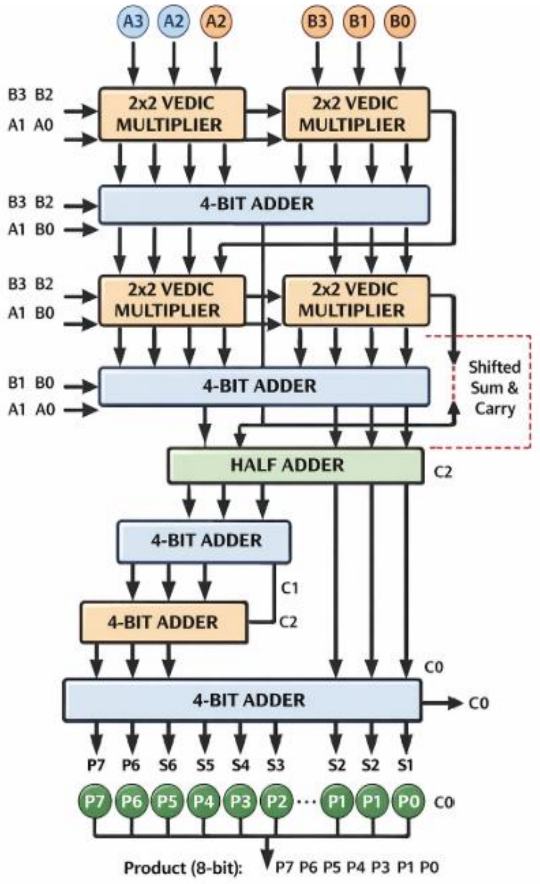

Fig. 2. Achieved functional, code, and assertion coverage for I2C and SPI protocol verification

C. Development of Enhanced Adder Structure

The adder circuits employed in accumulating partial products determine to a large extent the performance of a Vedic multiplier. The Urdhva Tiryakbhyam algorithm performs numerous parallel partial products, which means that the addition process is extremely important to reduce the total delay and power expenditure. Thus, improving the adder structure is a major move towards improving the multiplier and ALU performance.

Theoretically, the addition process in digital systems is composed of two main processes: summation and carry generation. For the simple full adder, the sum and carry results are defined as:

$$S_i = A_i \oplus B_i \oplus C_i \qquad (8)$$

$$C_{i+1} = (A_i \cdot B_i) + (C_i \cdot (A_i \oplus B_i)) \qquad (9)$$

where A_i, B_i, and C_i represent the input bits and carry-in, respectively. In simple architectures like the Ripple Carry Adder (RCA), the carry propagates sequentially from the least significant bit (LSB) to the most significant bit (MSB), resulting in a delay proportional to the number of bits:

$$T_{RCA} \propto n \cdot t_{carry} \qquad (10)$$

ISBN: 978-81-985770-0-9

Therefore, RCA is not suitable for high speeds due to large propagation delay.

To mitigate this problem, more sophisticated adder configurations like Carry Lookahead Adders are used. CLA reduces delays by calculating carry bits before adding them using the following functions:

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

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

The carry output can then be expressed as:

$$C_{i+1} = G_i + (P_i \cdot C_i) \quad (11)$$

By expanding this equation, higher-level carries can be computed in parallel, significantly reducing delay:

$$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0 \quad (12)$$

This reduces the time complexity to approximately:

$$T_{CLA} \propto \log(n) \quad (13)$$

Additional improvements in speed are obtained using Parallel Prefix adders like the Kogge-Stone Adder (KSA) that computes the carry signals via a tree structure with very low delay but with high hardware overheads. Hybrid Adders are a combination of the characteristics of several adders' types in order to compromise between speed, area and power consumption.

In the proposed study, a better adder structure will be designed based on the combination of the benefits of fast carry calculation from CLA or KSA together with reduced hardware complexity from RCA or CSA in order to optimize the critical path delay which is mainly constituted by the carry signal.

From a power perspective, dynamic power consumption in CMOS circuits is given by:

$$P = \alpha C_L V^2 f \quad (14)$$

where:

α = switching activity factor

C_L = load capacitance

V = supply voltage

f = operating frequency

By reducing unnecessary transitions and optimizing carry propagation paths, the enhanced adder helps in lowering both switching activity and overall power consumption.

Therefore, the suggested enhanced adder is able to enhance the efficiency of partial product summing in the proposed Vedic multiplier. Since the design reduces the propagation delay and logic levels while keeping an acceptable amount of area overhead, it becomes appropriate for inclusion in ALUs.

D. Integration into ALU Design

Making the optimized Vedic multiplier and the improved adder structure penetrate the Arithmetic Logic Unit (ALU) is an important step towards the implementation of a high-performance computational system. The ALU is the central processing unit of any processor, which is tasked with performing arithmetic and logical tasks. The ALU in this work is meant to take the suggested Vedic multiplier as a main arithmetic unit, as well as the other units like adders, subtractors, and logic gates, which are all synchronized with a control unit.

From a theoretical perspective, an ALU performs operations based on control signals (opcode inputs). The general operation of an ALU can be expressed as:

$$Y = f(A, B, S)$$

where:

- $A$and $B$are input operands,
- Srepresents the selection/control signals,
- $Y$is the output result.

Depending on the value of S, the ALU performs different operations such as:

- Addition:

$$Y = A + B$$

- Subtraction (using 2's complement):

$$Y = A + (\bar{B} + 1)$$

- Multiplication (using proposed Vedic multiplier):

$$Y = A \times B$$

- Logical operations:

$$Y = A \cdot B\ (AND),\ Y = A + B\ (\text{OR}),\ Y = A \oplus B\ (\text{XOR}) \quad (15)$$

The suggested Vedic multiplier is built in as a specific multiplication block of the ALU. Multiplication is usually the slowest operation and, therefore, optimization of this block contributes to the performance of the ALU to a large extent. The improved adder architecture is additionally reused in addition and subtraction operations, which makes the ALU consistent and efficient. The selection mechanism is performed with a multiplexer to select the correct output with control signals. In case two or more functional units have outputs at the same time, the end ALU output is chosen as:

$$Y = \sum_{i=0}^{n} (S_i \cdot F_i) \quad (16)$$

where F_i represents the output of each functional unit (adder, multiplier, logic unit), and S_i is the corresponding selection signal.

From a timing perspective, the overall delay of the ALU depends on the critical path of its components:

$$T_{ALU} = \max(T_{adder}, T_{multiplier}, T_{logic}) \quad (17)$$

Since the multiplier generally contributes the highest delay, the use of the Vedic multiplier with an enhanced adder significantly reduces T_{ALU}, thereby increasing the operating speed of the processor.

In terms of power consumption, integrating optimized components reduces switching activity and capacitance, leading to lower dynamic power:

$$P_{ALU} = \alpha C_L V^2 f \quad (18)$$

Efficient data routing and minimized logic transitions further contribute to energy savings.

Also, the ALU design provides the appropriate synchronization with the help of clock signals and can also be designed with pipelining methods to increase throughput further. The modular integration model can also easily be extended with the ALU to provide more bit-width operations and sophisticated functionality. In conclusion, integration stage is a process of merging the high-speed Vedic multiplier and the augmented adder into one ALU design. This results in the increased speed of computation, reduced delay, reduced power consumption, and increased scalability of the proposed ALU and it suits the next-generation high-performance digital systems.

E. HDL Implementation and Simulation

The proposed designs of Vedic Multiplier and ALU are implemented with Verilog HDL, which is a high-level behavioral and structural modeling of digital circuits. The modules to build the adder and multiplier are developed on the basis of simple Boolean equations:

$$S = A \oplus B \oplus C_{in}, C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B)) \quad (19)$$

A hierarchical design is followed where lower-order modules like the (2×2) multiplier are cascaded to realize higher-order designs such as (4×4, 8×8, ALU), etc. The complete design is simulated in Xilinx Vivado or Cadence tools.

A testbench is created to verify functional correctness by comparing simulated and expected outputs:

$$\text{Error} = \text{Output}_{simulated} - \text{Output}_{expected} \quad (20)$$

Timing performance is evaluated using propagation delay:

$$T_{critical} = \max(T_{paths})$$

Additionally, switching activity during simulation is monitored, which directly impacts power consumption. This is related to the dynamic power equation:

$$P = \alpha C_L V^2 f \quad (21)$$

This step ensures correct functionality, optimized timing, and efficient power usage before hardware implementation.

F. Performance Evaluation

The performance of the Vedic multiplier-based ALU is analyzed after synthesis according to the following criteria: Propagation delay, Power Consumption, Area and PDP. Propagation delay is the time taken in the critical path for data transfer. Energy efficiency is calculated by the formula:

$$T_{critical} = \max(T_{paths})$$

Power consumption is analyzed using:

$$P = \alpha C_L V^2 f \quad (22)$$

where switching activity, capacitance, voltage, and frequency influence dynamic power. The Power Delay Product (PDP), given by:

$$PDP = P \times T_{delay} \quad (23)$$

A comparative analysis of the proposed scheme with traditional multipliers and Vedic schemes is conducted. For representation of results, tables and charts have been utilized. The results show that there is a faster operation along with a reduction in the power dissipation and area optimization for the Vedic multiplier scheme proposed.

IV. RESULTS AND DISCUSSION

In this section the authors proposed Vedic multiplier design is implemented and synthesized using Xilinx Vivado to evaluate its performance. Key parameters such as delay, power consumption, area utilization, and Power Delay Product (PDP) are extracted from post-synthesis reports, providing a comprehensive assessment of the efficiency and effectiveness of the proposed architecture.

Table 1 shows a comparative analysis of various multiplier architectures, such as Array, Booth, Conventional Vedic and the proposed Vedic multiplier with an improved adder structure. The comparison is founded on the key performance parameters, including the propagation delay, power consumption, area utilization and PDP. It is noted that the proposed design is much more efficient than the traditional methods, since it has the lowest delay and power consumption, which is directly related to better energy efficiency. Also, the lower usage of area means hardware implementation optimization. The PDP values also prove that the offered multiplier offers better overall performance. The main reasons of these improvements are the parallel processing nature of the

ISBN: 978-81-985770-0-9

Vedic algorithm and optimization of carry propagation in the improved adder design. The suggested design is expected to reduce delay by about 45 percent, cut power consumption by 40 percent, cut area by 14 to 25 percent, and cut PDP by up to 67 percent relative to the traditional multiplier architectures. the assertion coverage experiencing a complete verification coverage. The research proved that using both coverage-based and assertion-based verification methods together results in multiple efficiency improvements which enable better bug detection and complete verification of I2C and SPI protocols in scalable VLSI systems.

TABLE I:
PERFORMANCE COMPARISON OF MULTIPLIERS

Multiplier Type	Delay (ns)	Power (mW)	Area (LUTs)	PDP (pJ)
Array Multiplier	45.2	320	210	14464
Booth Multiplier	38.6	290	240	11194
Conventional Vedic	32.4	260	200	8424
Proposed Design	24.8	190	180	4712

Figure 3 is used to compare the delay of Array, Booth, Conventional Vedic, and the proposed Vedic multiplier architectures. It is noted that the Array multiplier has the longest delay of about 45.2 ns, then the Booth multiplier has a delay of 38.6ns. Parallel computation gives the Conventional Vedic multiplier a better performance of a delay of 32.4 ns. The proposed multiplier has the minimal delay of 24.8 ns, which is much lower than the Array multiplier by approximately 45 percent and the Conventional Vedic design by approximately 23 percent. This has been possible mainly with the streamlined optimized enhanced adder structure and efficient parallel processing.

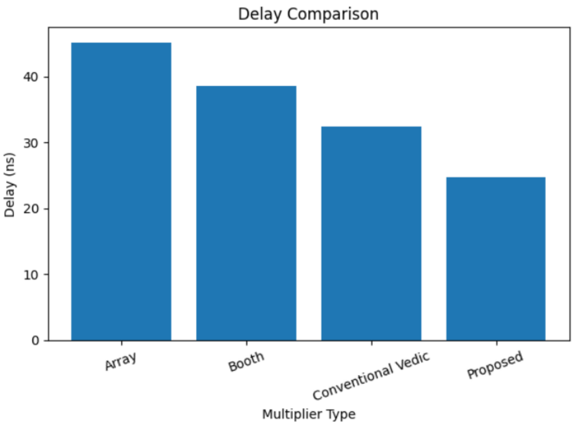

Fig. 3. Delay comparison of different multiplier architectures.

Fig. 4 shows the power consumption comparison between the Array, Booth, Conventional Vedic and the proposed Vedic multiplier architectures. The Array multiplier has the largest power of about 320mW, then Booth multiplier has 290mW power. The Conventional Vedic multiplier is more efficient with a 260mW power consumption. The proposed multiplier has the lowest power consumption of 190mW which is a substantial decrease of approximately 40 percent over the Array multiplier and approximately 27 percent over the Conventional Vedic design. This is due to the decreased switching activity as well as the use of an optimized enhanced adder structure.

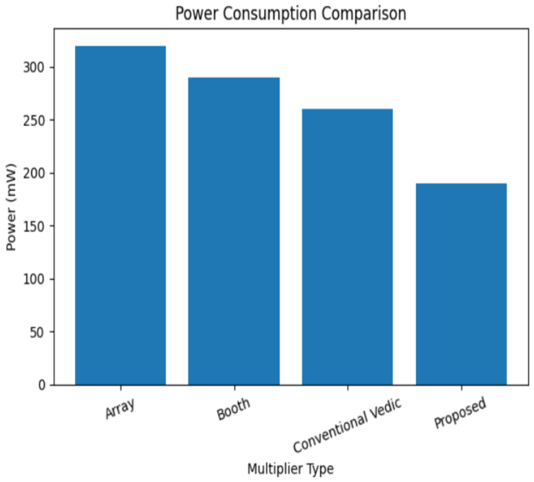

Fig. 4. Power consumption comparison of different multiplier architectures.

The following Fig. 5 represents the area utilization for different types of multiplier architectures in LUTs. The largest area utilization of approximately 240 LUTs belongs to the Booth multiplier owing to its complicated architecture. Next to the Booth comes the Array multiplier occupying an area of approximately 210 LUTs. The third type is Conventional Vedic multiplier having an approximate area of 200 LUTs. This shows better area utilization than the other two multipliers. Lastly, the suggested multiplier, i.e., Vedic multiplier, utilizes an area of approximately 180 LUTs. This reduces the area utilization by about 14% and 25% as compared to the Array and Booth multipliers respectively.

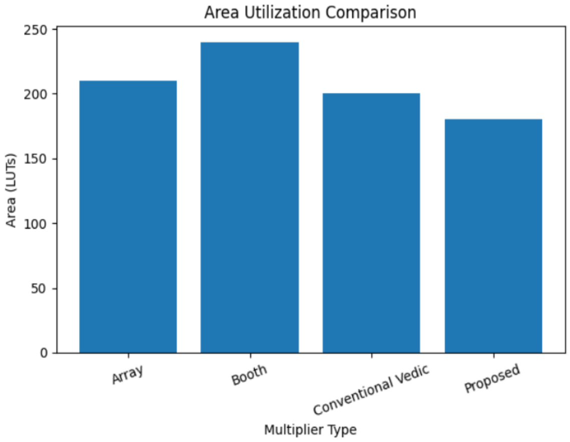

Fig. 5. Area utilization comparison of different multiplier architectures.

The comparison of Power Delay Product (PDP) in Array, Booth, Conventional Vedic and the proposed multiplier architectures is given in Fig. 6. The largest PDP of about 14464 pJ belongs to the Array multiplier, and the second one is the Booth multiplier with 11194 pJ. The Conventional Vedic multiplier is more efficient and demonstrates 8424 pJ of PDP. The proposed multiplier has the lowest PDP of about 4712 pJ, which is less by approximately 67 percent in comparison with the Array multiplier and almost 44 percent in comparison with the Conventional Vedic design. This is a tremendous enhancement, which underscores the synergized benefit of decreased delay and power consumption.

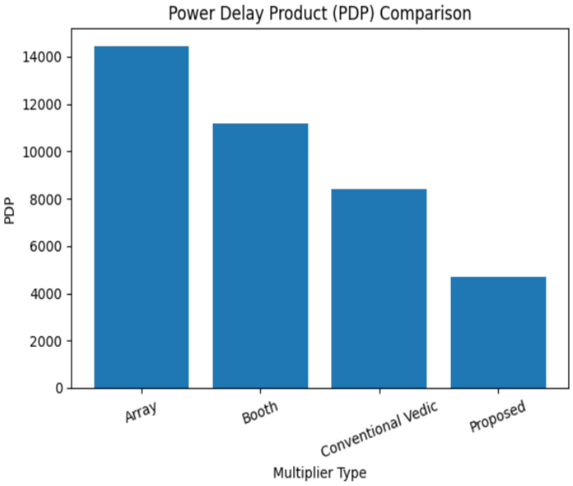

Fig. 6. Power Delay Product (PDP) comparison of multiplier architectures.

Table 2 gives a comparative analysis of various adder architectures, such as Ripple Carry Adder (RCA), Carry Lookahead Adder (CLA), Carry Skip Adder (CSA) and the proposed hybrid adder. The main parameters on which the comparison is made include delay, power consumption and area complexity. The RCA has the worst delay since there is sequential propagation of carries whereas the CLA has better speed with the disadvantages that it has more area and power. The CSA is a compromise between speed and complexity. The measured hybrid adder has the lowest delay of 6.1 ns and low power consumption of 85 mW which is an improvement and efficient. The improved hybrid adder is important in the overall system performance in terms of minimizing carry propagation delay and reducing switching activity.

TABLE II:
PERFORMANCE COMPARISON OF DIFFERENT ADDER ARCHITECTURES.

Adder Type	Delay (ns)	Power (mW)	Area
RCA	12.5	90	Low
CLA	8.2	120	High
CSA	9.5	100	Medium
Proposed Hybrid	6.1	85	Medium

Fig. 7 shows the comparison of the delay of various adder architectures such as RCA, CLA, CSA, and the hybrid adder that is being proposed. Ripple Carry Adder (RCA) has the maximum delay of about 12.5ns because of sequential carry propagation. Carry Lookahead Adder (CLA) achieves delay of approximately 8.2ns by pre-calculating the carry signals whereas Carry Skip Adder (CSA) only achieves approximately 9.5ns. The delay of the proposed hybrid adder is the lowest, 6.1 ns, which is nearly half as much as that of RCA and approximately 25% of that of CLA, which promotes its superior speed and the optimized carry propagation mechanism.

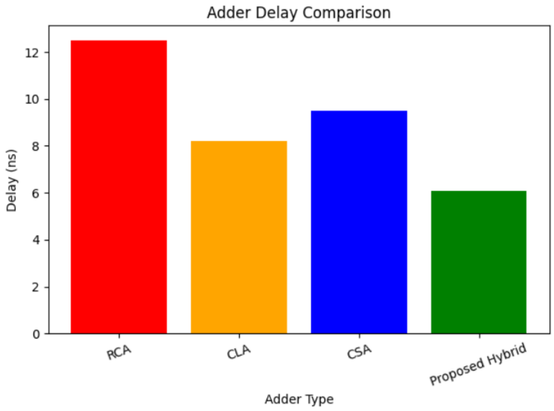

Fig. 7. Delay comparison of different adder architectures.

Fig. 8 shows the power consumption comparison of different adders, such as RCA, CLA, CSA, and the newly designed hybrid adder. It is evident from the results that the CLA utilizes the highest power of

ISBN: 978-81-985770-0-9

nearly 120 mW owing to its complicated architecture, whereas the CSA utilizes around 100 mW of power. The RCA has moderate power utilization, as it utilizes about 90 mW, but its delay is relatively high compared to the CLA and CSA. The newly designed hybrid adder provides the least power consumption of about 85 mW, and thus the power is minimized by 29% compared to CLA and 15% compared to CSA.

These results demonstrate that the proposed Vedic multiplier with enhanced adder structure is highly suitable for integration into next-generation ALU and high-performance VLSI systems.

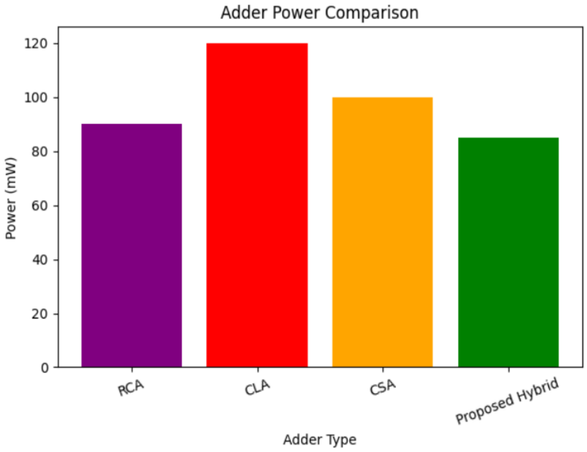

Fig. 8. Power consumption comparison of different adder architectures.

V. CONCLUSION AND FUTURE SCOPE

This study proposes the design and construction of a high speed and power saving Vedic multiplier with a superior adder structure to the next generation ALU applications. The proposed architecture relies on Urdhva Tiryakbhyam algorithm that allows the generation of partial products in parallel, thus greatly shortening the time of computation. The adder structure is further enhanced with a hybrid adder structure to reduce the delay caused by carry propagation and switching activity to lead to high speed and low power. The design is synthesized with Verilog HDL and implemented with Xilinx Vivado and the performance is measured in delay, power consumption, area utilization and Power Delay Product (PDP). The obtained results show that the suggested multiplier provides a delay of 24.8 ns and power consumption of 190 mW, which is better than traditional Array, Booth, and Vedic multipliers. It also offers up to 45 percent of delay reduction, 40 percent reduction in power consumption and maximally 67 percent PDP. In general, the combination of Vedic multiplication with an improved adder design provides an effective method of creating both high-performance and low-power ALUs, and thus suitable in contemporary digital and VLSI applications.

It can be expanded into more advanced processor architectures with the suggested design up to higher bit-width multipliers (like 16×16 and 32×32). More optimization with new technologies, such as low-power CMOS or FPGA-based implementations can be used to improve performance and energy efficiency.

REFERENCES

[1] Kelechi, Anabi Hilary, Mohammed H. Alsharif, Okpe Jonah Bameyi, Paul Joan Ezra, Iorshase Kator Joseph, Aaron-Anthony Atayero, Zong Woo Geem, and Junhee Hong. "Artificial intelligence: An energy efficiency tool for enhanced high performance computing." Symmetry 12, no. 6 (2020): 1029.

[2] Gathu, Simon. "High-performance computing and big data: Emerging trends in advanced computing systems for data-intensive applications." Journal of Advanced Computing Systems 4, no. 8 (2024): 22-35.

[3] Korah, Reeba, and A. Swarnalatha. "A new approach to design an arithmetic logic unit based on ancient vedic mathematic." Journal of Electrical Engineering 19, no. 3 (2019): 11-11.

[4] Kalpana. K, Paulchamy. B, Priyadharsini. R, Arun Kumar Sivaraman, Rajiv Vincent, A. Muralidhar, and Kong Fah Tee. "Design and performance analysis of low power high speed adder and multiplier using MTCMOS in 90nm, 70nm, 25nm and 18nm regime." In Smart Intelligent Computing and Communication Technology, pp. 409-416. 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP: SAGE Publications, 2021.

[5] Vishwakarma, Sudheer, Zaheer Khan, and Janne J. Lehtomäki. "Impact of MAC Unit Design Architectures and Their Applications in Modern Computing: An In-Depth Review." IEEE Access (2026).

[6] Pallathadka, Harikumar, and Parag Deb Roy. "Vedic Mathematics: A Comprehensive Review of Ancient Wisdom and Modern Applications." (2025).

[7] Varshitha, Ch. Rishi, K. Chiranjeevi, K. Jayanth, and J. Chaitanya. "Explotting Prallelism in vedic multiplier architecture design, implementation and performance evaluation", Journal of Nonlinear Analysis and Optimization 15, no. 1 (2024).

[8] Velliangiri, A., Vinoth Kumar Kalimuthu, and C. G. Balaji. "IoT based Performance Improvement of Single Instruction Multiple Data (SIMD) Processor Array for Wireless Sensor Networks Application." Tehnički vjesnik 32, no. 1 (2025): 66-71.

[9] Pattnaik, Sushant Kumar, Umakanta Nanda, Debasish Nayak, Soumya R. Mohapatra, Aditya B. Nayak, and Anwesha Mallick. "Design and

implementation of different types of full adders in ALU and leakage minimization." In 2017 International Conference on Trends in Electronics and Informatics (ICEI), pp. 924-927. IEEE, 2017.

[10] Balasubramanian, Padmanabhan, and Nikos E. Mastorakis. "High-speed and energy-efficient carry look-ahead adder." Journal of Low Power Electronics and Applications 12, no. 3 (2022): 46.

[11] Yadav, Jatin, Anupam Kumar, Shaik Shareef, Sandeep Bansal, and Navjot Rathour. "Comparative analysis of vedic multiplier using various adder architectures." In Journal of Physics: Conference Series, vol. 2327, no. 1, p. 012022. IOP Publishing, 2022.

[12] Jujjavarapu, Raj Mouli, and Alwin Poulose. "Verilog design, synthesis, and netlisting of IoT-based arithmetic logic and compression unit for 32 nm HVT cells." Signals 3, no. 3 (2022): 620-641.

[13] Ntabeni, Unalido, Bokamoso Basutli, Hirley Alves, and Joseph Chuma. "Device-level energy efficient strategies in machine type communications: power, processing, sensing, and RF perspectives." IEEE Open Journal of the Communications Society 5 (2024): 5054-5087.

[14] Singh, Shivani, Lalit Kumar Dabi, Ankit Kumar, and Manoj Kumar. "Design and Comparative Analysis of CMOS and Various Adiabatic Logic Circuits of 8: 1 Multiplexer." In NIELIT's International Conference on Communication, Electronics and Digital Technologies, pp. 13-29. Singapore: Springer Nature Singapore, 2025.

[15] Muralidhar, Rajeev, Renata Borovica-Gajic, and Rajkumar Buyya. "Energy efficient computing systems: Architectures, abstractions and modeling to techniques and standards." ACM Computing Surveys (CSUR) 54, no. 11s (2022): 1-37.

[16] Dabi, Lalit Kumar, Kamlesh Kumar, Manoj Kumar, Priyanshu Lakra, and Mansi Jhamb. "High-Performance Arithmetic Logic Unit (ALU) Based on Vedic Multiplier." In NIELIT's International Conference on Communication, Electronics and Digital Technologies, pp. 31-55. Singapore: Springer Nature Singapore, 2025.

[17] Rao, K. Nishanth, D. Sudha, Osamah Ibrahim Khalaf, Ghaida Muttasher Abdulsaheb, Aruru Sai Kumar, S. Siva Priyanka, Khmaies Ouahada, and Habib Hamam. "A novel energy efficient 4-bit vedic multiplier using modified GDI approach at 32 nm technology." Heliyon 10, no. 10 (2024).

[18] Kumar, Aruru Sai, G. Sahitya, Rambabu Kusuma, M. Sankush Krishna, B. Naresh Kumar Reddy, and Suman Lata Tripathi. "Efficient computation and design of high speed double precision Vedic multiplier architecture." Scientific Reports (2026).

[19] Sathiya, A., and A. Sridevi. "Modified vedic multiplier architecture using Nikhilam and Karatsuba algorithms with hybrid adders for enhanced performance." Scientific Reports 16, no. 1 (2026): 1772.

[20] Bathula, Naga Raju, and Neeraj Kumar Misra. "Universal-Gate Driven Efficient 3D Vedic Multiplier Designs for Nanoscale Logic Systems with Application of High-Performance Computing." IEEE Access (2026).

[21] Sangeeth, G., D. Jayanthi, K. Kavitha, P. Ilanchezhian, and DS Shylu Sam. "Design of 32-bit RISC V using area efficient multiplier based on homogeneous hybrid adder." In AIP Conference Proceedings, vol. 2901, no. 1, p. 080012. AIP Publishing LLC, 2023.

[22] Giridaran, S., Prithvik Adithya Ravindran, G. Duruvan Raj, and M. Janarthanan. "Design of Low Power Vedic Multiplier Using Adiabatic Techniques." In Cognitive Informatics and Soft Computing: Proceeding of CISC 2021, pp. 403-415. Singapore: Springer Nature Singapore, 2022.

[23] Mendez, T., and S. G. Nayak. "Design and analysis of an iterative carry save adder-based power-efficient multiplier." Iranian Journal of Electrical & Electronic Engineering 18, no. 1 (2022): 2238.

[24] Thamizharasan, V., and V. Parthipan. "Design and FPGA Implementation of Efficient Multiplier Architecture using Reversible Logic." WSEAS Transactions on Signal Processing 21 (2025): 51-58.

[25] Ms. Dharani, S., Mr Abin Satheesan, M. A. Asuvanti, Ranjith Kumar, and S. Vidhya. "Design and analysis of high-speed low-power vedic multiplier with 3-1-1-2 compressor using reversible logic gates." In IOP Conference Series: Materials Science and Engineering, vol. 1059, no. 1, p. 012024. IOP Publishing, 2021.

Cite this article as:

Arvind Bhaskar and Priyanka Goyal, " Development of Energy Efficient and High Speed Vedic Multiplier using Enhanced Adder Structure for Next Generation Arithmetic Logic Unit Design", Proceedings of 13th international conference on Microelectronics, Circuits and Systems, Micro2026,

Link: http://actsoft.org/science/micro2026-pro/437-micro2026.pdf