STM JOURNALS

International Journal of
VLSI Circuit Design & Technology

ISSN: 3139-356X
Volume 4, Issue 1, 2026
January–June
DOI (Journal): 10.37591/IJVCDT

https://journals.stmjournals.com/ijvcdt

Review IJVCDT

64-Bit ALU Design Using Vedic Mathematics

Sirigineedi Mounisha[1], Uppalapati Joshna Venkata Srujana[1], Vemula Rajesh[1], Velugubanti Lakshman[1], Yepuganti Manikanta[1], I. Rama Satya. Nageswara Rao[2,*], T Saran Kumar[3]

Abstract

High-speed arithmetic operations are crucial for better performance in contemporary digital systems. Particularly for high bit-width operations, conventional arithmetic logic units (ALUs) frequently experience increased latency and complexity. This work presents the design and implementation of a 64-bit ALU using notions from Vedic mathematics. The suggested design makes use of a Kogge–Stone adder for effective addition and the Urdhva Tiryagbhyam Sutra for quick multiplication. Among other mathematical and logical operations, the ALU is capable of addition, subtraction, multiplication, AND, OR, XOR, and NOT. Xilinx Vivado is used for simulation, while Verilog hardware description language (HDL) is used for implementation. Time-sensitive applications can benefit from the suggested architecture's optimization to reduce critical path latency and increase processing performance. Additionally, the architecture exhibits enhanced scalability and hardware resource utilization, making it simple to expand to larger bit-width systems. Additionally, the ALU is appropriate for high-performance and low-power very large scale integration (VLSI) systems due to the incorporation of power efficiency concerns. Modern (central processing units) CPUs, digital signal processing units, and embedded systems may be easily integrated with the design because of its modular nature. It is appropriate for high-speed VLSI applications since the results demonstrate lower latency and better performance when compared to traditional ALU designs, guaranteeing dependable performance in next-generation technologies.

Keywords: ALU, Vedic mathematics, Urdhva Tiryagbhyam, Kogge–Stone adder, Verilog, VLSI

INTRODUCTION

Multiplying one number by another is an arithmetic operation. Currently, multiplication is a fundamental function required in a number of digital signal processing (DSP) applications, such as convolution, FFTs, filters, and the arithmetic logic unit (ALU) of microprocessors [1]. Multiplication is a frequently used process; hence, it is important to build multipliers with low latency and effective power use. Arithmetic computations are useful for a wide range of tasks, from basic daily tasks such as counting or multiplying to complex scientific and commercial computations.

Conventional multipliers and adders are used in traditional ALU designs, which increase the power consumption and propagation latency. The ancient Indian mathematical theory known as Vedic mathematics offers effective algorithms that can be utilized to increase processing speed [2].

Multiplying a quantity by another quantity is an arithmetic operation. Currently, some DSPs require operations such as multiplication for applications such as convolution, FFTs, filters, and the arithmetic logic unit (ALU) of microprocessors [3].

***Author for Correspondence**
I. Rama Satya. Nageswara Rao
E-mail: irsnrao435@gmail.com

[1]UG Student, Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, Andhra Pradesh, India
[2]Assistant Professor, Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, Andhra Pradesh, India
[3]Associate Professor, Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, Andhra Pradesh, India

Received Date: March 30, 2026
Accepted Date: April 03, 2026
Published Date: May 05, 2026

Citation: Sirigineedi Mounisha, Uppalapati Joshna Venkata Srujana, Vemula Rajesh, Velugubanti Lakshman, Yepuganti Manikanta, I. Rama Satya. Nageswara Rao, T Saran Kumar. 64-Bit ALU Design Using Vedic Mathematics. International Journal of VLSI Circuit Design & Technology. 2026; 4(1): 1–11p.

© STM Journals 2026. All Rights Reserved

Multiplication is a frequently used operation; hence, it is important to build multipliers with low latency and effective power consumption [4]. From basic everyday tasks such as counting and multiplying to complex scientific and corporate computations, arithmetic computations are useful for a wide range of tasks.

Battery-operated and power-sensitive applications have grown dramatically owing to the current phenomenon of the stunning fusion of chip size reduction and an increase in the number of circuits on chips, which has led to the expansion of the developing area of low-power electronics. We focus on static power reduction at the architectural level in our study because, in the near future, this region of power will control the entire amount of power wasted in system-on-chips (SOCs) System. To maximize the static power dissipation, we suggested synthesizing the POWER GATING TECHNIQUE, namely, the fine-grained approach [5]. This method uses N-type metal–oxide–semiconductor (NMOS) to block the inputs to the gates when they are not in use, which reduces the needless usage of input and significantly lowers power consumption. Therefore, the idea of reducing static power at the architectural level, beginning with 1 bit and expanding to 8 bits with a matching drop in power usage, is the central theme of this study. A processor is an essential component of every digital system. Additionally, one of the key components of a microprocessor is an ALU [6]. In simple terms, the CPU functions as the brain of any system, and the ALU functions as the brain of the CPU. Thus, it is a computer brain. They feature well-optimized architecture and are composed of quick dynamic logic circuits. The CPU is responsible for a substantial amount of the overall power consumption of any processor. Because the ALU is clocked at the fastest speed and is always active, it also contributes to one of the greatest power-density regions of the processor. This causes thermal hotspots and abrupt temperature gradients inside the execution cores [7].

To create digital multipliers, such as the array multiplier, the "Vertical Crosswise Algorithm" based on the Urdhva Tiryakbhyam Sutra is employed. By considering smaller portions of size (N/2 = n), the ancient formula (sutra) offers a way to determine the product of N × N, of N bits per multiplicand. These sections will be split into smaller numbers (n/2) until they reach a 2×2 size [8]. Thus, it streamlines the procedure into a tree-like structure.

This study focuses on the development of a 64-bit ALU that uses the Urdhva Tiryagbhyam Sutra for multiplication and the Kogge–Stone adder for addition. The suggested system aims for fast speed, low delay, and efficient hardware utilization.

VEDIC MATHEMATICS

Vedic mathematics is a gift to the world from ancient sages of India. A system that is far more enjoyable and straightforward than current mathematics. The term "Veda," which means "the storehouse of all knowledge," is where the name "Vedic" Between 1911 and 1918, Sri Bharati Krishna Tirthaji (1884–1960), a scholar of Sanskrit, mathematics, history, and philosophy, rediscovered Vedic arithmetic from ancient Indian writings [9].

Design of 64×64 Multiplier

To create a 64×64 multiplier, we need four 32×32 multipliers and two adders, similar to the previous design. We must first create 64-bit and 96-bit adders before creating a 64×64 multiplier by properly instantiating the module and making the connections shown in the picture [10]. At this point, it is vital to test the register transfer level (RTL) code and ensure that the hardware matches the design. Xilinx's PlanAhead tool provides a better view of the hardware design with complex options. Figure 1 shows the 64-Bit Vedic Multiplier.

RELATED WORK

Several scholars have proposed alternative ALU architectures to improve performance and efficiency. Traditional designs employ ripple-carry adders and array multipliers, which experience significant delays owing to sequential carry propagation.

© STM Journals 2026. All Rights Reserved

Parallel prefix adders, such as the Kogge–Stone adder, have been designed to reduce the delay by processing carry signals in parallel [11]. Similarly, Vedic multipliers based on Urdhva Tiryagbhyam perform faster multiplication by producing partial products simultaneously. However, many existing solutions focus solely on addition or multiplication. The proposed approach combines both strategies into a single 64-bit ALU to increase the overall performance [12].

Existing ALU Design

Conventional ALUs are commonly built with ripple-carry adders (RCA), as shown in Figure 2, and array multipliers, as shown in Figure 3. In these architectures, the carry created by one stage must propagate sequentially to the next stage, resulting in a significant propagation delay, particularly for large-bit-width operations, such as 64-bit processing [13]. Furthermore, conventional multipliers produce partial products sequentially, resulting in a longer calculation time and worse efficiency. These constraints render typical ALUs unsuitable for high-speed applications.

The suggested system architecture employs advanced techniques such as Vedic multiplication, Kogge–Stone adders, and fin field-effect transistor (FinFET) technology to create a high-performance 64-bit ALU. The architecture aims to achieve fast speed, low-power consumption, and effective area utilization.

Proposed ALU Design

The proposed 64-bit ALU consists of the following main components from Figure 4.

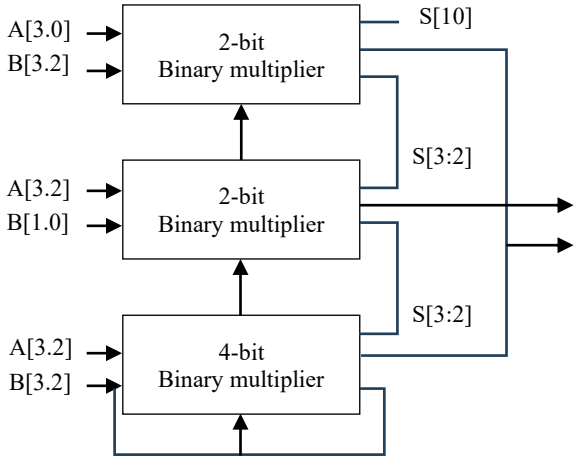


Figure 1. 64-bit Vedic multiplier.

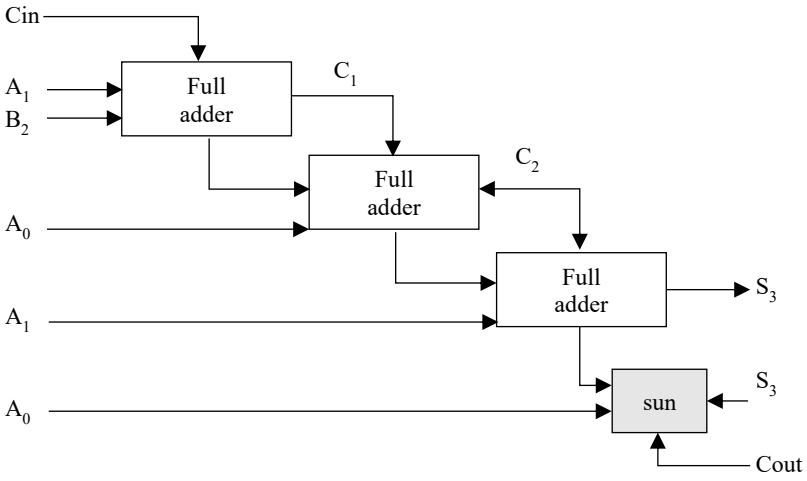


Figure 2. Ripple-carry adder.

© STM Journals 2026. All Rights Reserved

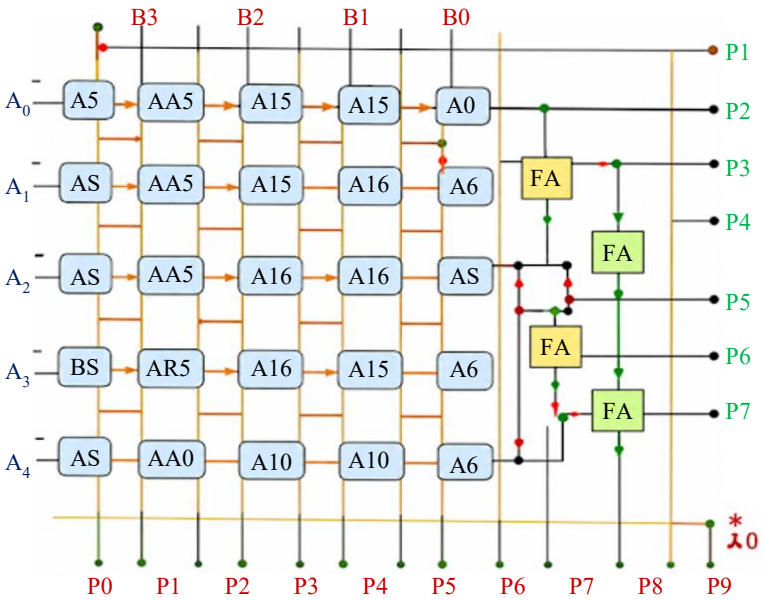


Figure 3. Array multiplier.

SYSTEM ARCHITECTURE

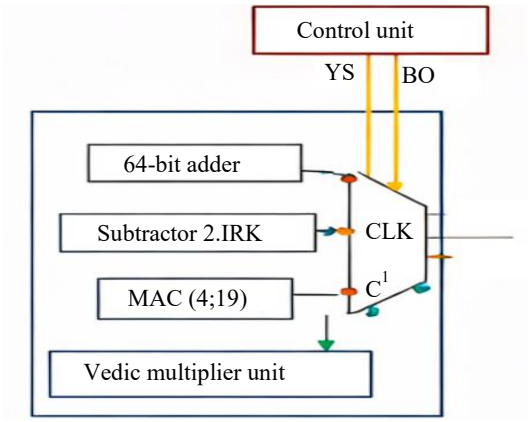


Figure 4. 64-bit ALU architecture.

Kogge–Stone Adder

The Kogge–Stone adder (KSA), as shown in Figure 5, is a high-speed parallel prefix adder that is frequently employed in high-performance digital systems. It lowers the carry propagation time by concurrently creating and propagating carry signals utilizing the generate (G) and propagate (P) signals. The adder uses a tree-structured prefix network to compute carry bits quickly, resulting in O (log n) time for n-bit additions. This makes the Kogge–Stone adder ideal for large word sizes, such as 32- and 64-bit ALUs [14, 15].

Subtractor

Subtraction using the Kogge–Stone subtractor method. The two's complement of the subtrahend is added to the minuend instead of performing a separate ripple-carry subtraction. The Kogge–Stone structure ensures that carry (borrow) signals are generated and propagated simultaneously, thereby reducing the time even for 64-bit processes. Subtraction is a basic arithmetic process on a digital computer. Partial Subtractor, is to subtract, dial a different number. The facts show this difference. Thus, the XOR gate also uses the AND gate to perform partial subtraction. One inverted input and one output were used in the operation. The logic circuit of the half adder is as follows:

© STM Journals 2026. All Rights Reserved

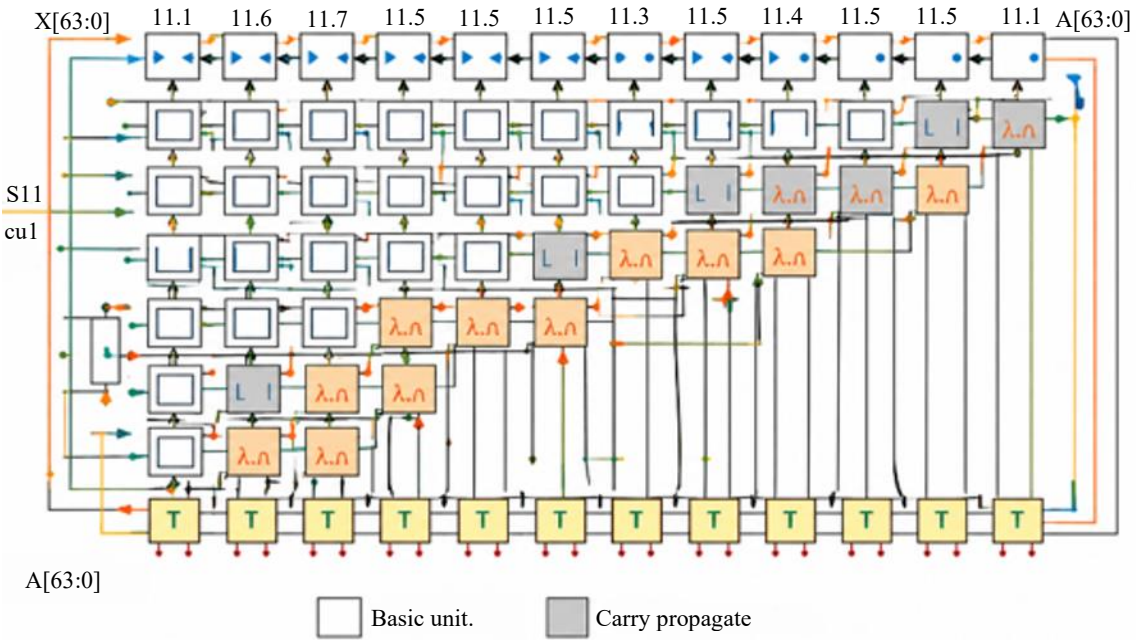


Figure 5. 64-bit Kogge–Stone adder.

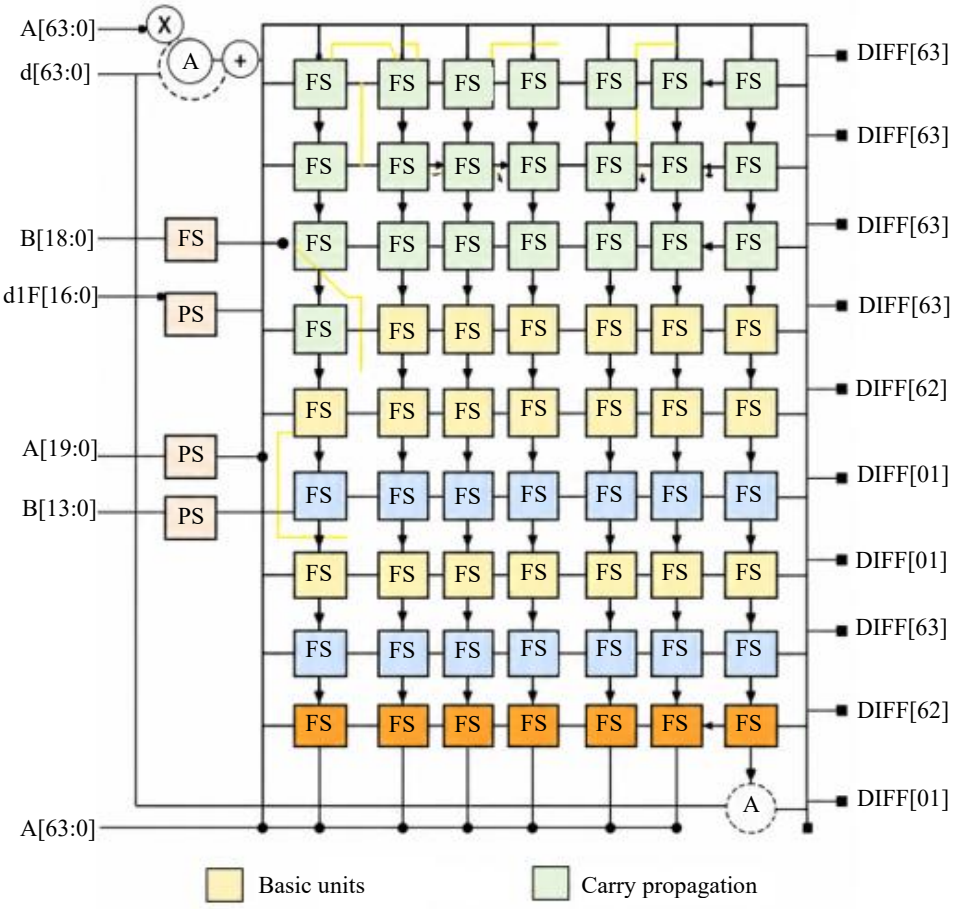


Figure 6. 64-bit subtractor.

The addition of the decremented input A before feeding the AND gate to finish the credit is the sole distinction. When subtracting two integers in multi-digit subtraction, the preceding number must be subtracted first. Consequently, three inputs are required for the subtractor, which is not feasible in practice, as a partial subtractor in the proposed method from Figure 6.

© STM Journals 2026. All Rights Reserved

Vedic Multiplier

The Vedic multiplier employs Vedic mathematics to accelerate multiplication and requires less technology. It uses the Urdhva Tiryakbhyam technique, which translates to "vertically and crosswise." The 32-bit multiplier divides inputs A [31:0] and B [31:0] into smaller parts and performs vertical and crosswise multiplication. Based on ideas from ancient Indian mathematics, namely the Urdhva Tiryagbhyam Sutra (vertical and crosswise technique), the Vedic multiplier is a fast arithmetic device. By producing partial products simultaneously, as opposed to sequentially, as in traditional techniques such as array or Booth multipliers, this method allows for quick multiplication. Vedic multipliers significantly reduce propagation latency and increase processing efficiency as a result of this parallelism from Figure 6.

Vedic multipliers are often implemented using complementary metal-oxide-semiconductor (CMOS) technology in digital design, and hardware languages such as Verilog or very high speed integrated circuit hardware description language (VHDL) are used to describe them. Large-bit multipliers (like 32-bit or 64-bit) are divided into smaller modules (such as 2×2 or 4×4 blocks) by the architecture, and these modules are then merged hierarchically. This modular approach reduces the design complexity and improves scalability. Vedic multipliers are suitable for high-performance applications such as DSP processors, ALUs, and embedded systems because they provide benefits over classical multipliers, including shorter latency, lower power consumption, and effective area usage. However, careful optimization of adders and connectors is required for very high bit-widths. Figure 7 shows the 64-bit Vedic multiplier.

Multiply-Accumulate (MAC) 64-bit Unit

The 64-bit arithmetic logic unit's (ALU) ncorporates a multiply-accumulate (MAC) unit, as shown in Figure 9, multiplies two values and adds the result to an accumulator in a single step. This speeds up calculations by combining both operations and saving time. It operates with ALU control signals, allowing it to work seamlessly with other ALU operations, increasing speed and efficiency while keeping the hardware simple.

One predictable component of many DSP applications that require multiplication and accumulation is the MAC unit, as shown in Figure 8. High-speed digital DSP systems are another application of this technology. DSP has several applications, such as inner products, filtering, and convolution. The nonlinear functions typically used in DSP techniques include discrete cosine transform (DCT) or discrete wavelet transform (DWT).

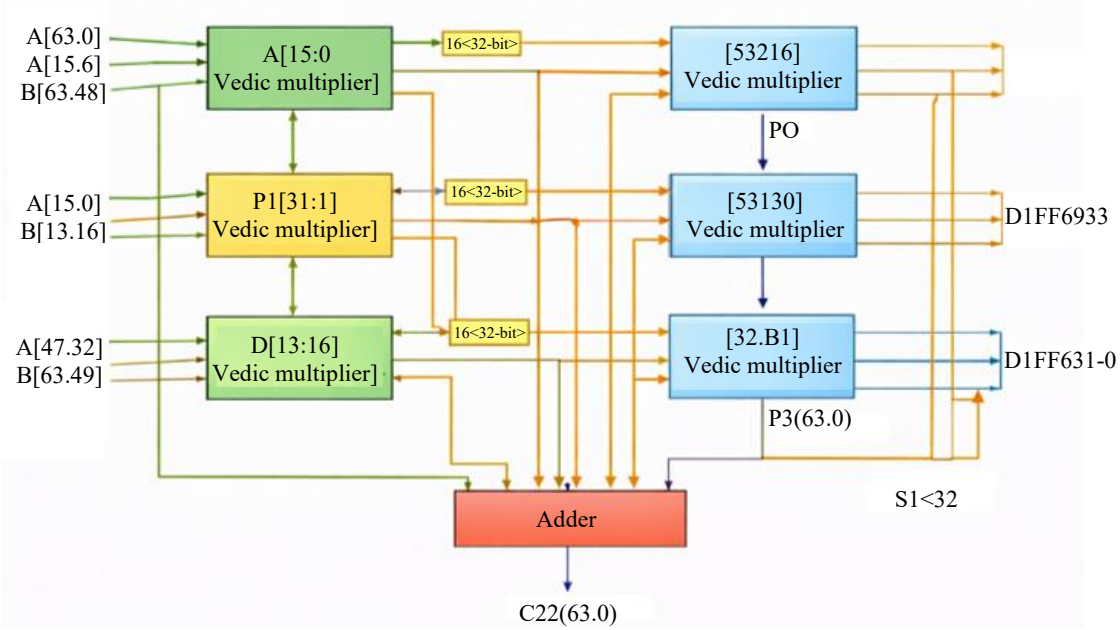


Figure 7. 64-bit Vedic multiplier.

© STM Journals 2026. All Rights Reserved

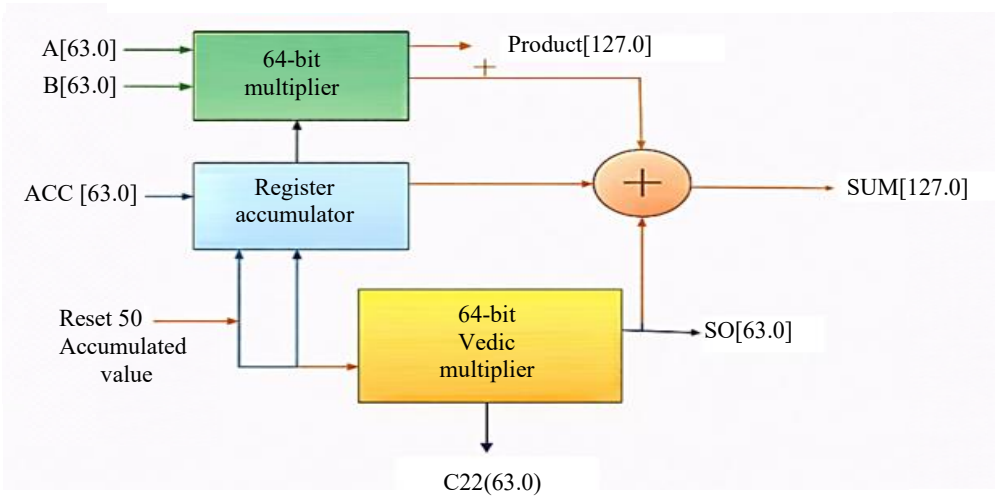


Figure 8. MAC 64-bit unit.

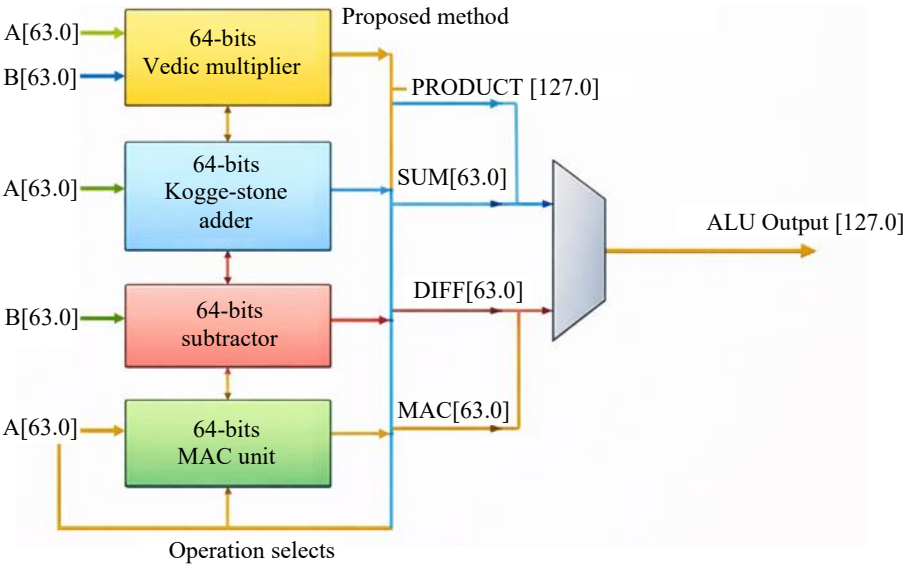


Figure 9. Proposed 64-bit ALU.

The speed of execution and overall calculation performance influence the overall speed of addition and multiplication arithmetic calculations, as they are primarily carried out by the cyclic application of addition and multiplication.

- Logic unit (AND, OR, XOR, NOT)
- Control unit for operation selection

The operation performed and the associated output produced are determined by the control signals.

WORKING PRINCIPLE

The ALU operates on the control signals provided as inputs. The module corresponding to the selected operation is activated, and the Kogge–Stone adder calculates the sums by generating and propagating signals. The second operand's 2's complement is added to the first operand to perform the subtraction. The Vedic multiplier performs both vertical and transverse multiplications.

Logical operations are performed bitwise on the input operands. The final output is selected using a multiplexer based on the control signals.

© STM Journals 2026. All Rights Reserved

To increase computing speed and efficiency, a 64-bit ALU utilizing Vedic mathematics integrates quick arithmetic methods from Vedic sutras, mainly the Urdhva Tiryagbhyam (Vertical and Crosswise) method. For 64-bit data inputs, the ALU executes logical operations such as AND, OR, XOR, and NOT, as well as arithmetic operations such as addition, subtraction, and multiplication. The fundamental idea is to use Vedic multipliers, which produce partial products simultaneously, instead of traditional multipliers. The Urdhva Tiryagbhyam method is implemented in each of the smaller modules (such as 8-bit or 16-bit blocks) that constitute the multiplication process in a 64-bit system.

High-speed adders, such as carry look-ahead adders (CLA) or carry-save adders (CSA), are used to aggregate the outputs of these smaller multipliers that calculate partial results concurrently. Compared with conventional multiplication methods, this parallelism dramatically lowers the propagation time. Optimized adder circuits are used for addition and subtraction, and these circuits are frequently combined with Vedic ideas to reduce the carry propagation time. Standard CMOS logic gates are used to perform logical operations, which are managed by a common selection mechanism. Based on the opcode inputs, a control unit (usually a multiplexer-based selector) decides what needs to be done. The ALU generates an output by routing the inputs via the appropriate arithmetic or logic block based on the control signals. Figure 10 shows the working principle of the 64-bit ALU.

IMPLEMENTATION AND RESULTS

Verilog HDL was used to implement the design, and Xilinx Vivado was used to simulate it. The performance analysis in Table 1 shows that the proposed ALU has a reduced delay compared with conventional designs.

The reduction in delay demonstrates the efficiency of the Vedic multiplier and Kogge–Stone adder. Figures 11–16 show the internal schematic of the Kogge–Stone adder 64-bit, subtractor 64-bit, Vedic multiplier 64-bit, MAC (64-bit), ALU 64-bit, and 64-bit ALU using Kogge–Stone adders (Figure 8).

Table 1. Delay comparison.

	Existing ALU	Proposed ALU
Delay (ns)	55.847	37.578

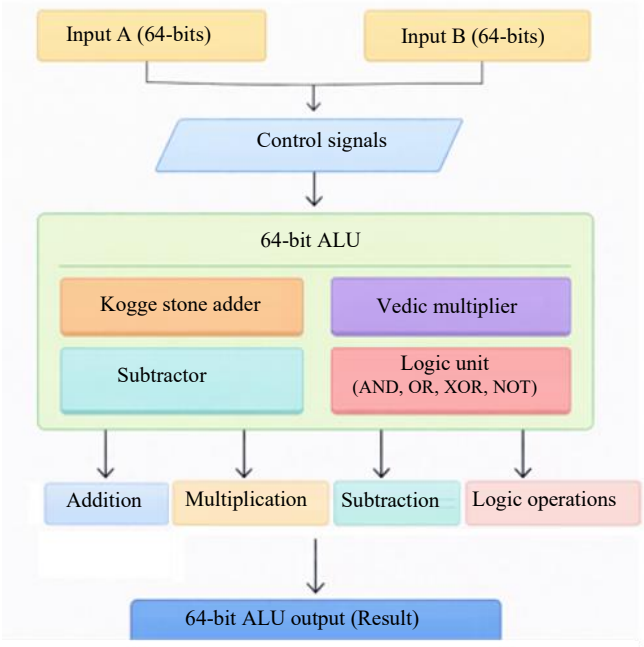


Figure 10. Working principle of 64-bit ALU.

© STM Journals 2026. All Rights Reserved

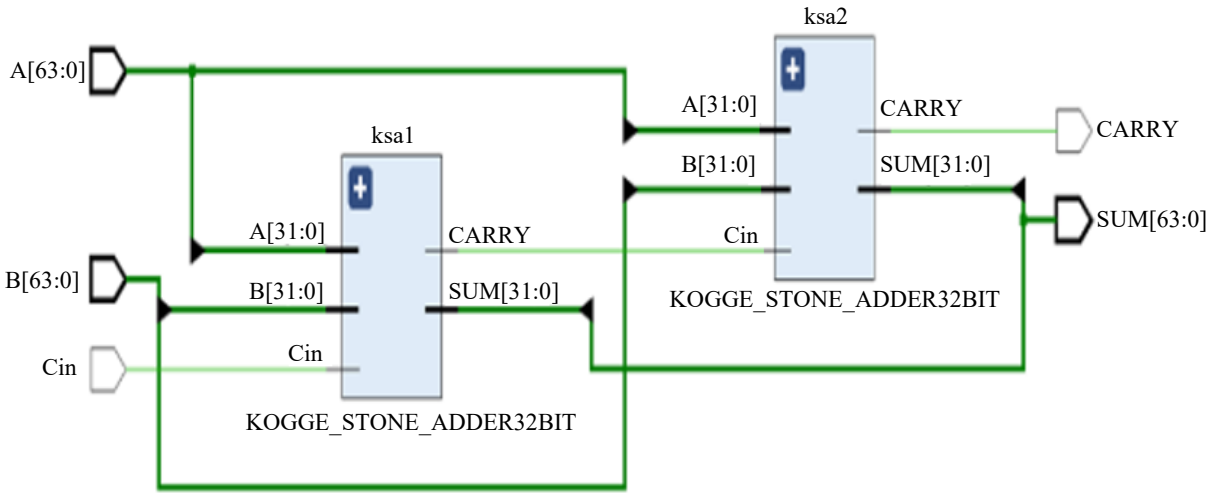


Figure 11. 64-bit ALU using Kogge–Stone adders.

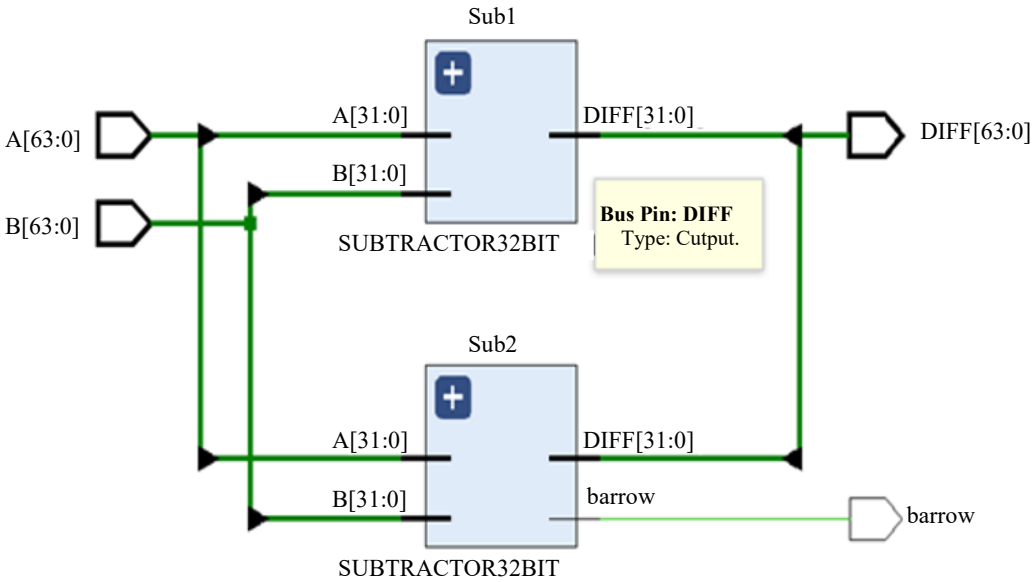


Figure 12. Internal schematic of 64-bit subtractor.

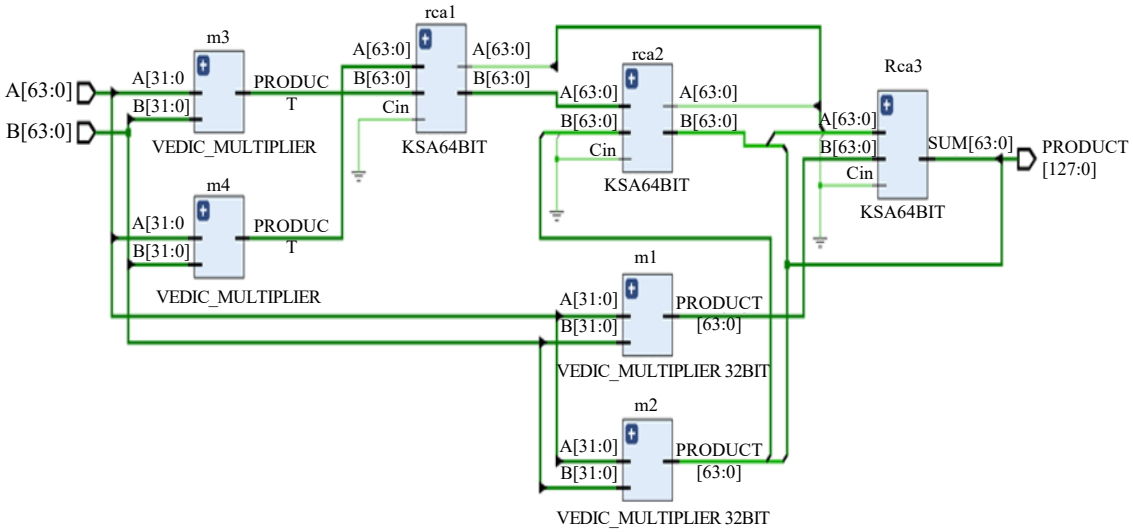


Figure 13. Internal schematic of a 64-bit Vedic multiplier.

© STM Journals 2026. All Rights Reserved

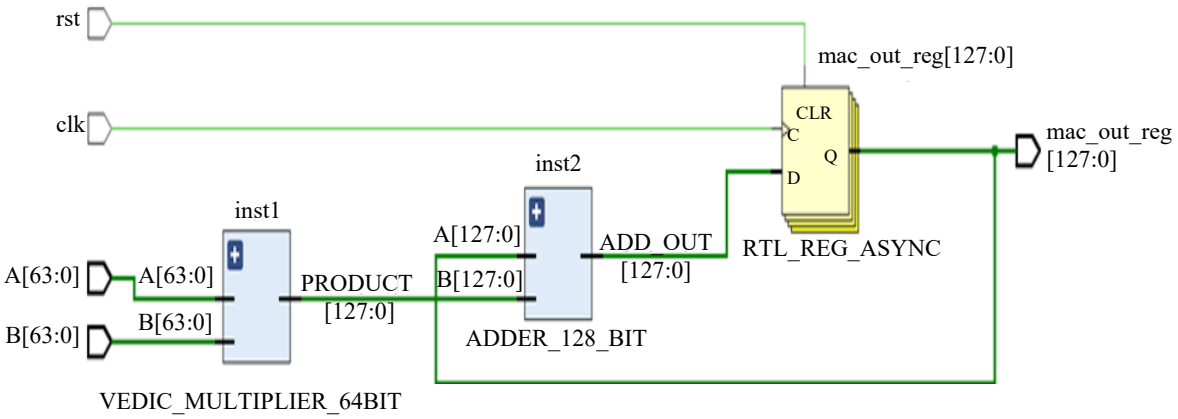


Figure 14. Internal schematic of MAC 64-bit.

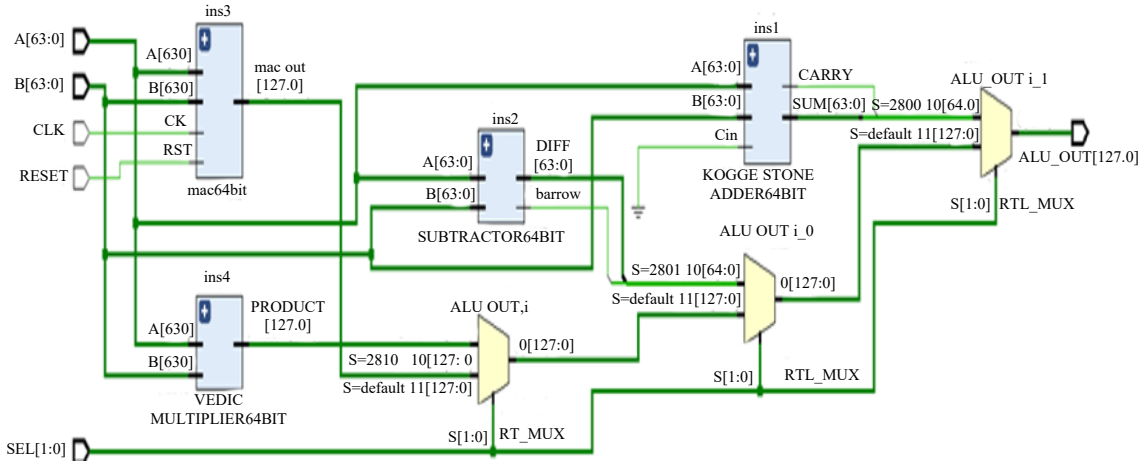


Figure 15. Internal schematic of a 64-bit ALU.

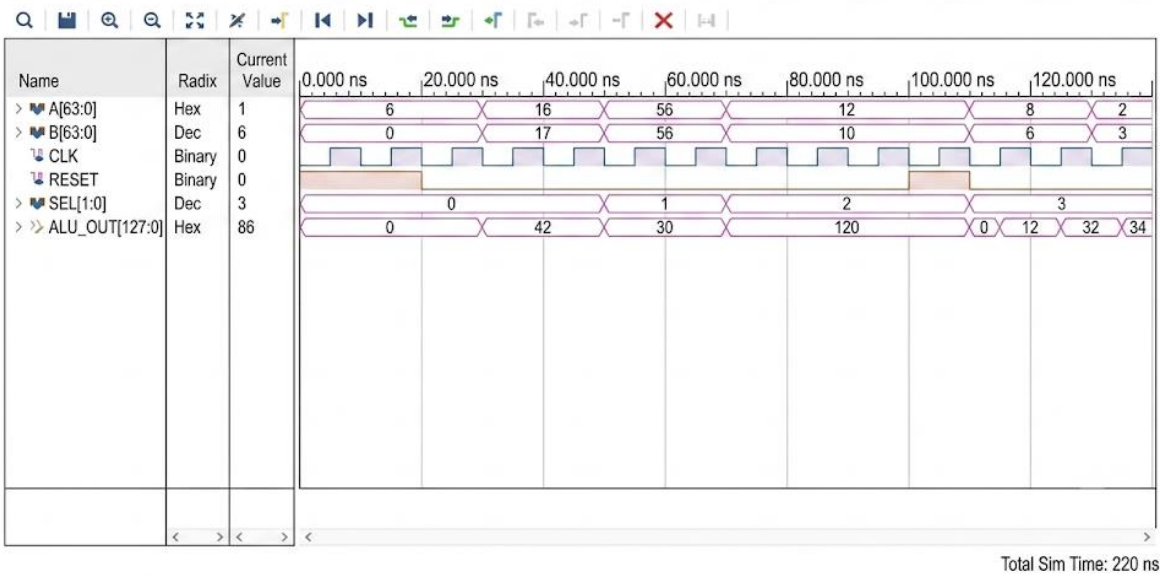


Figure 16. Output waveforms of a 64-bit ALU.

CONCLUSION

This study offers a high-speed 64-bit ALU based on Vedic mathematics. Compared to traditional approaches, the principles of Vedic mathematics are effective for a variety of mathematical operations.

© STM Journals 2026. All Rights Reserved

For various signal processing applications, several researchers have suggested computing units based on Vedic mathematics, and these designs have been shown to be effective. The proposed Vedic mathematics-based ALU implements addition, multiplication, and logical functions using the Urdhva Tiryakbhyam Sutra. Numerous signal processing domains can use the proposed ALU architecture. These structures can be utilized in the future to replace outdated integrated circuits, minimize changes to hardware circuit boards, boost productivity, and guarantee that operational limitations are satisfied.

REFERENCES

1. Kleidermacher D, Kleidermacher M. Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development. Waltham (MA): Elsevier; 2012.
2. Pavaloiu IB, Petrescu I, Raducanu M, Dragoi G. Software applications for teaching digital design. In: INTED2018 Proceedings: 12th International Technology, Education and Development Conference; 2018 Mar 5-7; Valencia, Spain. Valencia (Spain): IATED; 2018. p. 7457-7463. doi:10.21125/inted.2018.1755.
3. Ahmad M, Zhang L, Ng KTW, Chowdhury MEH. Complex-exponential-based bio-inspired neuron model implementation in FPGA using Xilinx System Generator and Vivado Design Suite. Biomimetics (Basel). 2023;8(8):621. doi:10.3390/biomimetics8080621. PMID: 38132560.
4. Gaikwad KM, Chavan MS. Vedic mathematics for digital signal processing operations: A review. Int J Comput Appl. 2015;113:10-14. doi:10.5120/19924-1503.
5. Kogge PM, Stone HS. A parallel algorithm for the efficient solution of a general class of recurrence equations. IEEE Trans Comput. 1973;C-22(8):786-793. doi:10.1109/TC.1973.5009159.
6. Parhami B. Parallelism in computer arithmetic: A historical perspective (invited paper). 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, ON, Canada. 2018. p. 165-169. doi:10.1109/MWSCAS.2018.8623956.
7. Satya R, Rao N, Rao KS. Comparison of different TFETs: An overview. In: Raj B, Tripathi SL, Chaudhary T, Srinivasa Rao K, Singh M, editors. Integrated Devices for Artificial Intelligence and VLSI. Hoboken (NJ): John Wiley & Sons; 2024. p. 49-73. doi:10.1002/9781394205158.ch3.
8. Isaac RD. The future of CMOS technology. IBM J Res Dev. 2000;44(3):369-378. doi:10.1147/rd.443.0369.
9. Rao IR, Krishna BM, Shameem S, Khan H, Madhumati GL. Wireless secured data transmission using cryptographic techniques through FPGA. Int J Eng Technol. 2016;e-ISSN:0975-4024.
10. Singh P, Dinendra C, Hemanth RJ, Vivek NM. Design of high-speed Vedic multiplier using Urdhva Tiryakbhyam sutra. 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichy, India. 2022. p. 1-5. doi:10.1109/ICEEICT53079.2022.9768542.
11. Kumar HS, Joshi PV. FPGA implementation of digital modulation schemes using Verilog HDL. Turk J Comput Math Educ. 2021;12:2442-2451.
12. Prasanna Kumar G, Budumuru PR, Krishna Chaitanya Varma A, Satyanarayana B, Saran Kumar T, Rao IRSN. Design analysis of memristor-based 7T SRAM using heterojunction tunneling transistors. J Circuits Syst Comput. 2025;34:2550280. doi:10.1142/S0218126625502809.
13. Kumar TS, Siddani DP, Rao IRSN, Harika P, Vijjapu A, Kumar GP, Satyanarayana BVV. Low power adder circuit design and implementation using lector technique. In: Pareek P, Mishra S, Reis MJCS, Gupta N, editors. Cognitive Computing and Cyber Physical Systems. IC4S 2024. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Vol. 598. Cham (Switzerland): Springer; 2025. p. 159-170. doi:10.1007/978-3-031-77078-4_15.
14. Ferdjallah M. Introduction to Digital Systems: Modeling, Synthesis, and Simulation Using VHDL. Hoboken (NJ): John Wiley & Sons; 2011. doi:10.1002/9781118007716.
15. Jais A, Palsodkar P. Design and implementation of 64 bit multiplier using Vedic algorithm. 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India. 2016. Piscataway (NJ): IEEE; 2016. p. 775-779. doi:10.1109/ICCSP.2016.7754250.

© STM Journals 2026. All Rights Reserved