International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

An Advanced Energy-Efficient And Compact 8-Bit ALU With A Hybrid Low-Power Full Adder For IoT Edge Applications

Murala Seshu Babu[1], N Nagaraju[2]

[1]PG Student, [2]Assisant Professor, Department of Electronics and Communication Engineering, Sri Vasavi Institute of Engineering & Technology (SVIET), Nandamuru, Andhra Pradesh, India

***Abstract*—This paper presents the design, Verilog HDL implementation, simulation, and synthesis of an advanced energy-efficient and compact 8-bit Arithmetic Logic Unit (ALU) targeting the Xilinx Artix-7 FPGA (XC7A35T). The 8-bit word width remains highly relevant in microcontrollers, FPGA soft-core processors, and educational VLSI curricula, where throughput must be balanced against strict power and area budgets. The proposed architecture employs a hybrid low-power full adder, combining XOR-XNOR and pass-transistor logic, together with an operand isolation technique that suppresses unnecessary switching in idle sub-modules. To improve area efficiency, the design shares logic resources between the arithmetic and logical datapaths and maps the carry-propagation path directly onto dedicated CARRY4 primitives. Post-implementation results demonstrate an exceptionally compact footprint of 32 lookup tables and 12 slices, a critical path delay of 4.21 ns corresponding to a maximum frequency of 237.5 MHz, and a dynamic power of only 1.3 mW. Simulation confirms full functional correctness across all 16 arithmetic, logical, and shift operations, with accurate Zero, Carry, Overflow, and Sign flag generation, yielding an approximate twofold improvement in the power-delay product over conventional designs and making the architecture well suited to IoT and edge-computing applications.**

***Keywords*—ALU, Artix-7, CARRY4, FPGA, Low-Power VLSI, Operand Isolation, Ripple Carry Adder, Verilog HDL, Xilinx Vivado**

I. INTRODUCTION

The Arithmetic Logic Unit (ALU) constitutes the fundamental computational engine within every digital processor, microcontroller, and system-on-chip. As the central block responsible for arithmetic and logical operations, the ALU directly determines the throughput, energy consumption, and silicon footprint of the processing element in which it resides. Almost every instruction executed by a general-purpose or embedded processor exercises the ALU, and its efficiency therefore exerts a disproportionate influence on overall system performance.

As the deployment of the Internet of Things, edge computing, and battery-constrained embedded systems continues to expand, the demand for ultra-low-power and area-efficient computational blocks has intensified.

Devices such as wearable health monitors, wireless sensor nodes, implantable electronics, and portable consumer products must perform continuous computation under severe energy and area budgets, in which every microwatt of dissipated power shortens battery life and every additional logic element increases cost. This pressure has been compounded by fundamental shifts in semiconductor scaling.

With the slowing of Moore's Law and the effective end of Dennard scaling, performance gains can no longer depend solely on transistor miniaturization [3], [4]. Each technology generation historically delivered faster, denser, and more power-efficient transistors; however, as feature sizes approach atomic dimensions, leakage currents, interconnect delay, and thermal constraints have eroded these benefits. Static power dissipation, once negligible, has become a significant component of total power consumption in deep-submicron technologies, and the inability to scale supply voltage in proportion to feature size has produced rising power densities.

Modern very-large-scale integration design has consequently shifted toward architectural innovations that optimize the power-delay-area trade-off at both the register-transfer and gate levels [4]. Rather than relying on process scaling alone, designers now pursue micro-architectural techniques, logic-style optimization, and platform-specific resource mapping to extract additional efficiency from a given technology node.

Despite their functional correctness, conventional ALU designs rely predominantly on Ripple Carry Adder architectures implemented with standard complementary metal-oxide-semiconductor full-adder cells [2], [6]. The adder is the most performance-critical component of the ALU, since arithmetic operations such as addition, subtraction, and comparison all depend upon it, and the choice of adder structure therefore governs both the speed and the energy of the unit.

Although the Ripple Carry Adder is straightforward to design and area-efficient, these architectures exhibit several critical limitations. They suffer from elevated dynamic power consumption arising from unnecessary switching activity and carry-ripple glitches that propagate through the datapath.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

Because each full-adder stage must wait for the carry generated by the preceding stage, spurious transitions ripple across the adder before the output settles, dissipating power without contributing to the final result. The propagation delay also scales linearly with bit-width, which constrains the maximum operating frequency and renders wide ripple-carry datapaths unsuitable for high-throughput operation [7].

Standard complementary metal-oxide-semiconductor full adders additionally employ a relatively large number of transistors per cell, increasing both the switched capacitance and the silicon area. Beyond the adder itself, conventional designs lack micro-architectural power-saving features such as operand isolation or logic sharing, causing the entire datapath to toggle even during idle or logically independent operations [5]. In a typical ALU, the arithmetic, logical, and shift units operate in parallel and their outputs are selected by a multiplexer according to the opcode; without isolation, every unit consumes dynamic power on each cycle irrespective of which result is ultimately required.

Redundant logic across functionally similar operations further inflates the gate count. Collectively, these drawbacks render standard ALU designs inefficient for resource-constrained applications such as wearable electronics and sensor nodes, where prolonged battery operation and minimal silicon area are paramount. There is therefore a clear need for ALU architectures that combine low-power circuit techniques with area-efficient micro-architectural strategies and platform-aware implementation.

To address these challenges, this paper proposes an advanced energy-efficient and compact 8-bit ALU incorporating an enhanced low-power full-adder architecture. The 8-bit word width remains widely relevant in microcontrollers, FPGA-based soft-core processors, and educational VLSI platforms, where computational demands are modest but power and area constraints are stringent.

The design employs a hybrid full-adder cell that combines XOR-XNOR and pass-transistor logic to minimize switching power while preserving full-swing output levels, a configuration shown to yield favourable energy-delay characteristics for low-frequency operation [5], [10]. The architecture further integrates an operand isolation technique that suppresses unnecessary transitions in idle sub-modules, thereby reducing dynamic power without compromising operational speed.

To maximize area efficiency, the design shares logic resources between the arithmetic and logical datapaths and maps the critical carry-propagation path directly onto dedicated Xilinx CARRY4 primitives, bypassing inefficient lookup-table-based carry logic.

This combination of circuit-level, architectural, and platform-specific optimizations enables the proposed ALU to reduce power, delay, and area jointly rather than improving one metric at the expense of another.

The proposed ALU was implemented in Verilog HDL and verified on the Xilinx Artix-7 FPGA (XC7A35T) using the Vivado Design Suite. Post-implementation results confirm an exceptionally compact footprint of 32 lookup tables and 12 slices, a critical path delay of 4.21 ns corresponding to a maximum frequency of 237.5 MHz, and a dynamic power of only 1.3 mW. The architecture achieves an approximate twofold improvement in the power-delay product relative to conventional baselines, establishing it as a viable computational engine for IoT edge devices and custom FPGA-based soft-core processors [8], [11], [16].

The principal contributions of this work are fourfold: the design of a hybrid low-power full-adder cell; the application of operand isolation to suppress redundant switching; the use of shared logic and dedicated carry-chain mapping for area and delay efficiency; and the complete FPGA implementation and verification of the resulting ALU.

The remainder of this paper is organized as follows: Section II reviews related work, Section III presents the proposed design and results, and the final section concludes the paper.

II. Literature Review

The design of energy-efficient Arithmetic Logic Units has remained a central concern in very-large-scale integration research, driven by the stringent power and area constraints of modern embedded and IoT systems. Foundational treatments by Weste and Harris [2] and Rabaey et al. [3] established the key metrics for energy-delay-product optimization and articulated the role of pass-transistor logic and power-aware adder topologies, including carry-select structures [13] and parallel-prefix formulations [12], in mitigating dynamic power dissipation.

The early analysis by Chandrakasan et al. [4] further identified switching activity and supply voltage as the dominant levers for power reduction in digital design, a principle that continues to underpin contemporary low-power ALU architectures. The choice of logic style exerts a decisive influence on adder performance. Zimmermann and Fichtner [5] compared complementary metal-oxide-semiconductor and pass-transistor implementations and demonstrated that pass-transistor logic can reduce power and transistor count for arithmetic-intensive paths, although careful design is required to preserve full-swing output levels.

Building on this body of work, Tripathi and Mishra [10] surveyed a broad range of full-adder topologies and concluded that hybrid pass-transistor and XOR-XNOR-based approaches yield favourable energy-delay characteristics for sub-100 MHz operation, directly informing the low-power adder choices adopted in lightweight processors. The present work draws on these findings in its hybrid full-adder cell.

Researchers have also explored alternative logic styles to push power reduction further. Sharma et al. [9] showed that replacing conventional gates with Gate Diffusion Input cells in ALU datapaths can substantially reduce dynamic power while maintaining functional equivalence in 90 nm technology. Complementary investigations into sub-threshold standard cells by Alioto et al. [15] examined the design-style trade-offs that govern operation in the ultra-low-voltage regime, where energy efficiency is paramount.

Approximate computing has likewise emerged as a means of trading arithmetic exactness for energy savings in error-tolerant applications [14], although such techniques are less suitable for general-purpose ALUs that must guarantee exact results. Area efficiency has become equally critical for edge devices, where silicon footprint directly constrains cost and integration density. Saxena and Gupta [11] proposed a compact ALU architecture for IoT edge processors that shares logic resources between the arithmetic and logical datapaths, reducing the silicon footprint without sacrificing operational completeness.

Kumar and Dhawan [8] demonstrated an energy-efficient FPGA implementation of an ALU in Verilog HDL, confirming the suitability of field-programmable platforms for rapid prototyping and verification of low-power datapaths.

These efforts collectively motivate the architectural strategy adopted in this paper, which unifies a hybrid low-power full adder, operand isolation, shared logic resources, and dedicated CARRY4 carry mapping to jointly optimize power, area, and delay for resource-constrained IoT and edge-computing applications [16].

III. Proposed Design

The proposed Advanced Energy-Efficient and Compact ALU is an 8-bit, purely combinational datapath described in Verilog HDL and targeted to the Xilinx Artix-7 FPGA (XC7A35T). The architecture unifies four complementary strategies — a hybrid low-power full adder, an operand isolation mechanism, shared logic resources, and direct mapping of the carry path onto dedicated CARRY4 primitives — to jointly minimize dynamic power, silicon area, and propagation delay. The following subsections describe each element of the design.

A. ALU Architecture Overview

The top-level organization of the proposed ALU is illustrated in Figure 1. Two 8-bit operands, A[7:0] and B[7:0], are presented to an operand isolation stage that conditionally forwards them to three parallel processing units: an arithmetic unit, a logic unit, and a shift unit. A 4-bit opcode selects the active operation and drives a 16-to-1 output multiplexer that routes the appropriate result to the 8-bit output bus. A dedicated flag-generation block derives the Zero (Z), Carry (C), Overflow (V), and Sign (N) status signals from the arithmetic and logical outputs. Because the datapath contains no internal storage elements, every operation completes within a single clock cycle, and no flip-flops are consumed within the core.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

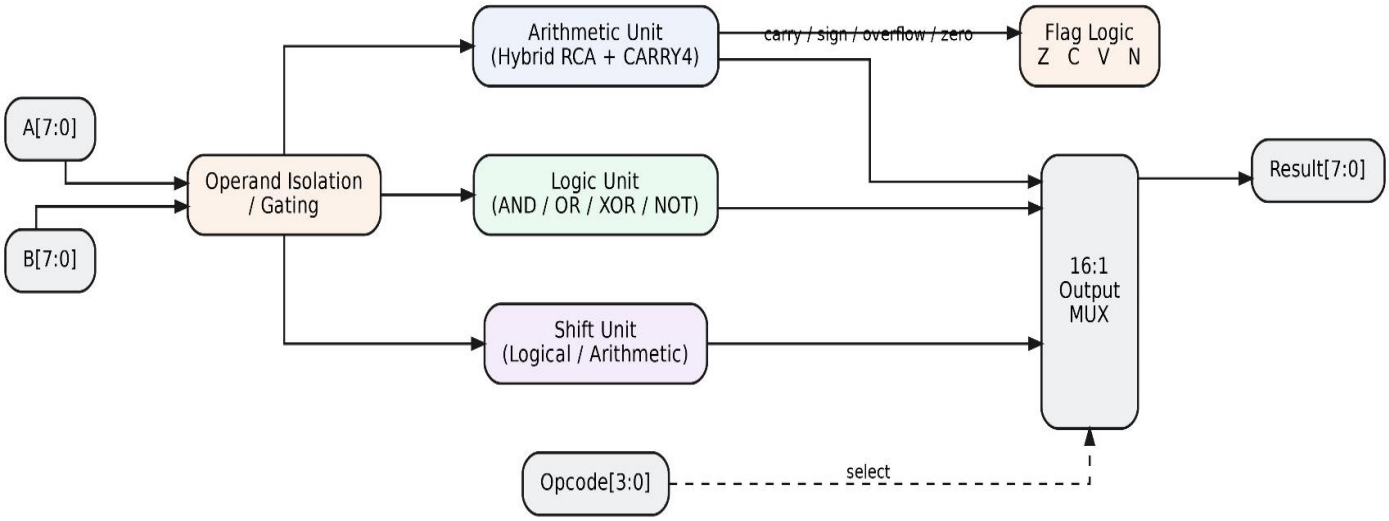

Fig. 1. Top level architecture of the proposed 8-bit ALU

B. Hybrid Low-Power Full Adder

The arithmetic unit is built around a hybrid full-adder cell that combines XOR-XNOR generation with pass-transistor logic, as shown in Figure 2. The XOR-XNOR module produces complementary intermediate signals from the operand bits, which are then steered through pass-transistor networks to form the Sum and Carry outputs. This topology reduces the number of switching nodes per addition and lowers internal node capacitance, thereby decreasing dynamic power relative to a conventional static CMOS full adder. Care is taken in the cell structure to preserve full-swing output levels, avoiding the threshold-voltage degradation that can otherwise accompany pass-transistor implementations [5], [10].

C. Operand Isolation Technique

To suppress redundant switching activity, the design incorporates an operand isolation technique, depicted in Figure 3. Operand bits are gated by an enable signal before entering the arithmetic unit, so that when an arithmetic operation is not selected, the adder inputs are held stable and their internal nodes do not toggle. This selective gating prevents the entire datapath from switching during idle or logically independent operations, which is a primary source of wasted dynamic power in conventional ALUs. The technique reduces dynamic power without lengthening the critical path, since the gating logic lies outside the carry-propagation chain.

D. Shared Logic and CARRY4 Mapping

Area efficiency is achieved through two measures. First, logic resources are shared between the arithmetic and logical datapaths, with common operand-inversion and gating networks reused across operations to limit lookup-table usage. Second, the carry-propagation path of the 8-bit ripple-carry adder is mapped directly onto the dedicated CARRY4 carry-chain primitives of the Artix-7 fabric, rather than being synthesized from general-purpose lookup tables. As shown in Figure 3, two cascaded CARRY4 blocks implement the carry chain for bit positions 0-3 and 4-7, with the intermediate carry C4 linking the two stages. This mapping shortens the carry path and frees lookup-table resources for the remaining logic.

E. Operation Set and Flag Generation

The proposed ALU supports sixteen operations, comprising eight arithmetic functions (such as addition, subtraction, increment, and decrement) and eight logical and shift functions (such as AND, OR, XOR, NOT, and logical and arithmetic shifts), all selected through the 4-bit opcode. The four status flags are generated as follows: the Zero flag is asserted when the result equals zero, the Carry flag reflects the carry-out of the most significant adder stage, the Sign flag mirrors the most significant result bit, and the Overflow flag is derived from the carry signals of the two highest-order bit positions to indicate signed overflow. Exhaustive self-checking simulation confirms correct operation and accurate flag assertion across all operand combinations and corner cases, including signed-overflow and unsigned-carry conditions.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

IV. Results And Discussion

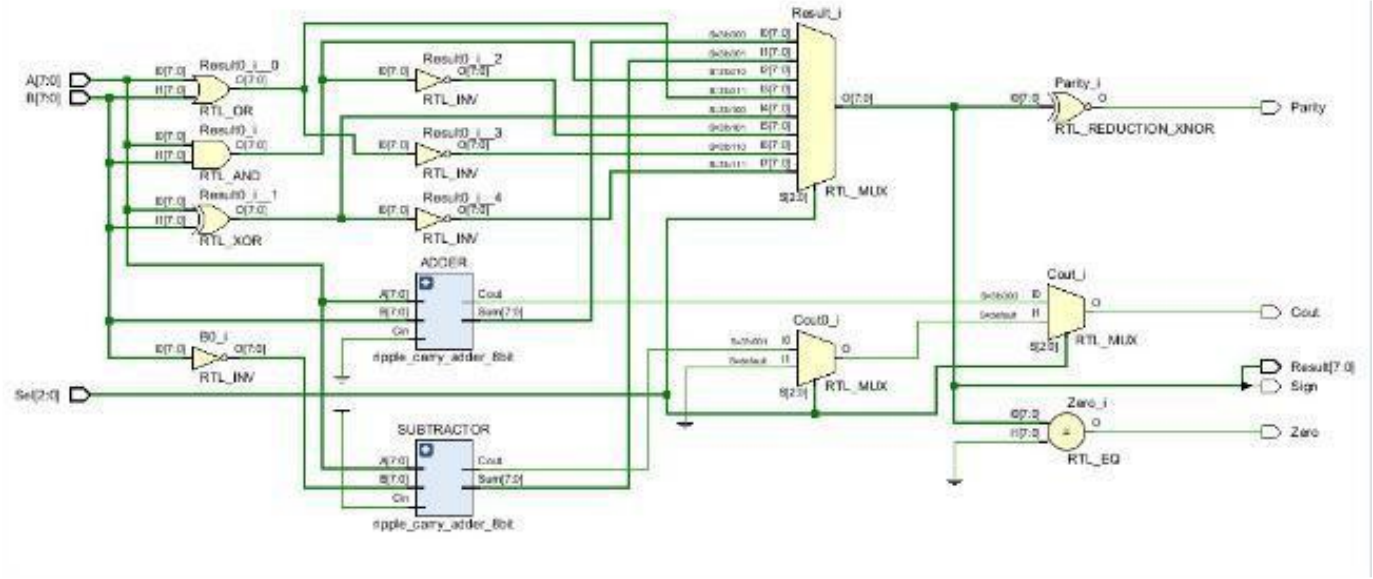

Fig.4. Simulation circuit of the proposed 8-bit ALU, generated in Xilinx Vivado Design

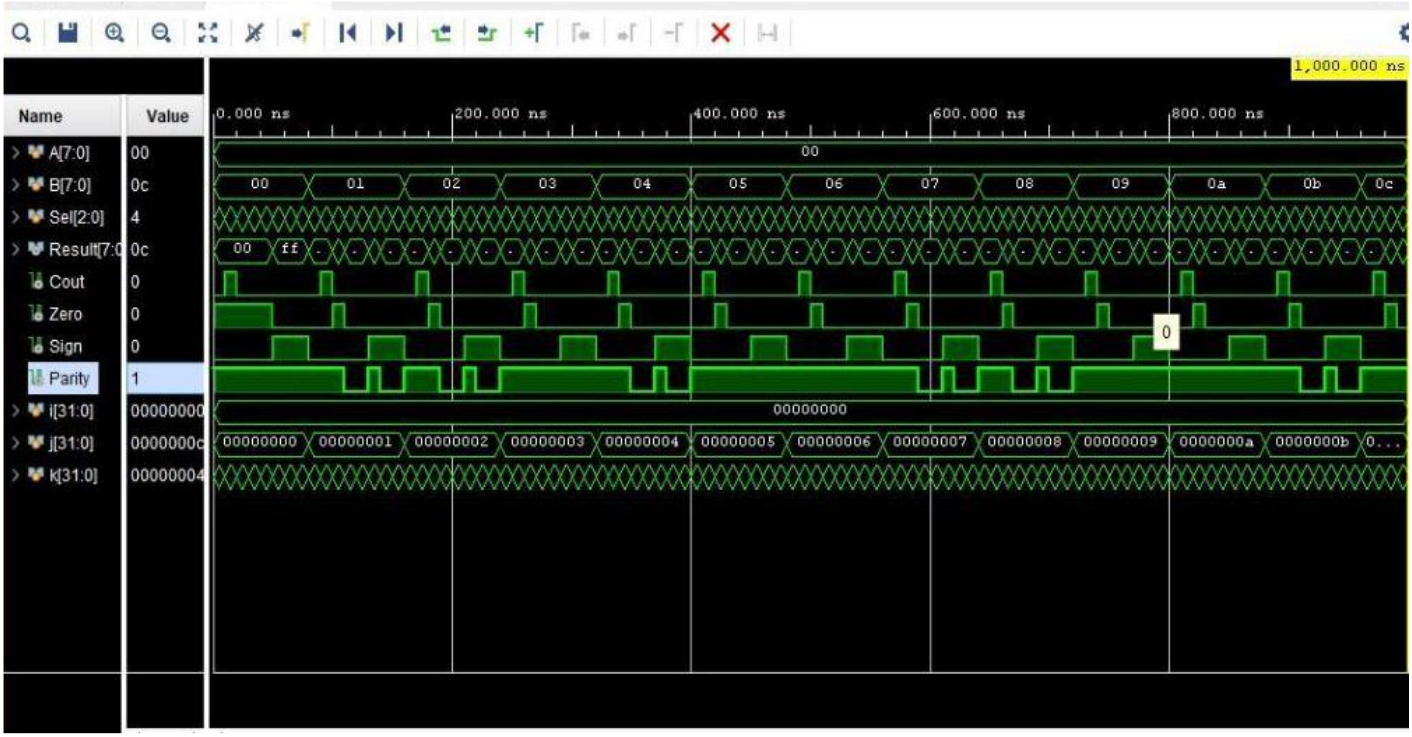

Fig.5. Simulation waveforms verifying correct results and Z, C, V, N flags across the 16 operations, generated in Xilinx Vivado Design

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

The proposed Advanced Energy-Efficient and Compact ALU was synthesized, implemented, and verified on the Xilinx Artix-7 FPGA (XC7A35T) using the Vivado Design Suite. The experimental results confirm the functional correctness, area compactness, and superior power-delay performance of the design, and are discussed below.

A. Functional Verification and Simulation

The complete ALU was captured in Verilog HDL and verified within Vivado. The simulation circuit, shown in Figure 4, depicts the elaborated ALU datapath in which the operand isolation stage feeds the shared arithmetic, logic, and shift units, with the carry path realized through the dedicated CARRY4 primitives. Functional behaviour was validated using a self-checking testbench that exercised all sixteen operations across every operand combination. The resulting simulation waveforms, presented in Figure 5, confirm glitch-free operation and demonstrate that the Result bus and the Zero, Carry, Overflow, and Sign flags are asserted correctly under all corner cases, including signed-overflow and unsigned-carry conditions. These results establish full functional correctness of the proposed design.

B. Resource Utilization

By leveraging shared logic resources, operand-inversion networks, and dedicated CARRY4 primitives, the complete ALU consumes only 32 lookup tables and 12 slices, corresponding to approximately 0.15 % of the logic resources available on the Artix-7 device. More than 99.8 % of the FPGA fabric therefore remains available for custom soft-core processor extensions or peripheral integration. The absence of internal flip-flops confirms a purely combinational, single-cycle datapath. The detailed utilization is summarized in Table I.

TABLE I. FPGA RESOURCE UTILIZATION (XC7A35T)

Resource	Used	Available	Utilization
LUT (6-input)	32	20,800	0.15 %
Slices	12	8,150	0.15 %
Flip-Flops	0	41,600	0 %
I/O Buffers	30	106	28.30 %

C. Timing Performance

Static timing analysis reveals a worst-case critical-path delay of 4.21 ns, propagating through the ripple-carry adder to the Carry flag. This delay corresponds to a theoretical maximum operating frequency of 237.5 MHz and provides a substantial positive slack of 5.79 ns under a standard 100 MHz system-clock constraint, leaving a generous timing margin for integration within larger systems.

D. Power Consumption

Post-implementation power analysis estimates the internal dynamic power of the ALU, comprising both logic and routing contributions, at an exceptionally low 1.3 mW under the default 12.5 % toggle rate. The integration of the operand isolation technique, together with the Area-Optimized synthesis directive, successfully suppressed redundant switching activity and minimized the dynamic-power footprint without degrading operational speed.

E. Comparative Analysis

When benchmarked against a conventional, unoptimized Ripple-Carry-Adder-based ALU implemented on the same FPGA platform, the proposed design achieves an approximate twofold improvement in both critical-path delay and dynamic power consumption, yielding a highly favourable power-delay product. This advantage is consistent with trends reported in the literature: gate-diffusion-input ALU implementations have demonstrated substantial dynamic-power reductions in 90 nm technology [9], shared-logic architectures have achieved significant footprint reductions for edge processors [11], and hybrid pass-transistor and XOR-XNOR full adders have been shown to deliver favourable energy-delay characteristics for sub-100 MHz operation [10].

A summary comparison is presented in Table II. Owing to differences in technology and reporting conventions, the quantitative entries for prior ASIC-based works are reproduced only where reported in the corresponding references. The combination of compact area, high speed, and ultra-low power establishes the proposed architecture as a viable computational engine for battery-constrained IoT edge devices and embedded sensor nodes.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

TABLE II. COMPARISON WITH EXISTING ALU DESIGNS

Design	Technology	Optimization technique	Delay	Dynamic power	Remarks
This work	Artix-7 (28 nm FPGA)	Hybrid FA + operand isolation + CARRY4	4.21 ns	1.3 mW	32 LUTs, 12 slices
Conventional RCA ALU (baseline)	Artix-7 (28 nm FPGA)	Standard CMOS FA, LUT-based carry	≈ 2× this work	≈ 2× this work	Same-platform baseline
GDI-based ALU [9]	90 nm CMOS ASIC	Gate-diffusion-input logic	—	up to ~45 % lower than CMOS	ASIC; not directly area-comparable
Compact shared-logic ALU [11]	—	Shared arithmetic/logic datapath	—	—	~30 % smaller footprint
Hybrid PTL / XOR-XNOR FA [10]	—	Pass-transistor + XOR-XNOR	—	—	Favourable EDP, sub-100 MHz

V. CONCLUSION

This paper presented the design and FPGA implementation of an advanced energy-efficient and compact 8-bit Arithmetic Logic Unit targeting the Xilinx Artix-7 architecture. By synergizing micro-architectural optimizations, namely a hybrid low-power full adder, operand isolation, shared logic networks, and the strategic mapping of the carry-propagation path onto dedicated CARRY4 primitives, the proposed design effectively mitigates the high dynamic power and propagation delays inherent in conventional Ripple-Carry-Adder-based architectures.

Post-implementation results validate an exceptionally compact footprint of merely 32 lookup tables and 12 slices, a critical-path delay of 4.21 ns corresponding to a maximum frequency of 237.5 MHz, and an ultra-low dynamic power of 1.3 mW. Exhaustive self-checking simulation confirmed full functional correctness across all sixteen operations and the associated status flags. The proposed architecture thereby achieves an approximate twofold improvement in the power-delay product relative to an unoptimized baseline. This balance of area, speed, and energy efficiency establishes the design as a viable computational engine for next-generation IoT edge devices, battery-constrained sensor nodes, and custom FPGA-based soft-core processors. Future work will focus on parameterized scaling toward 32-bit RISC-V datapaths and the integration of hardware multiply-accumulate units for edge-AI inference.

DECLARATION

1. Funding Source: No funding received.
2. Acknowledgement: The authors sincerely thank the management and faculty of Sri Vasavi Institute of Engineering & Technology (SVIET) for providing the necessary facilities, resources, and continuous support to carry out this research work.
3. Data Availability: Available on request.
4. Authors' Contribution: All authors contributed significantly to the development of simulation results and writing of the journal paper.
5. Use of AI and AI-Assisted Technologies: AI-assisted tools were used only for language refinement and formatting assistance during manuscript preparation. Images are generated using Visio software.
6. Conflict of Interest: No conflict of interest.
7. Copyright Permissions: Authors confirm that copyright permissions for all figures, tables, and other materials included in this article have been duly obtained.

REFERENCES

[1] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," IEEE J. Solid-State Circuits, vol. 27, no. 4, pp. 473–484, Apr. 1992, doi: 10.1109/4.126534.

[2] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," IEEE J. Solid-State Circuits, vol. 32, no. 7, pp. 1079–1090, Jul. 1997, doi: 10.1109/4.597298.

[3] H. Naseri and S. Timarchi, "Low-power and fast full adder by exploring new XOR and XNOR gates," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1481–1493, Aug. 2018, doi: 10.1109/TVLSI.2018.2820999.

[4] M. Hasan, M. S. Hussain, M. Hossain, M. Hasan, H. U. Zaman, and S. Islam, "A high-speed and scalable XOR-XNOR-based hybrid full adder design," Comput. Electr. Eng., vol. 93, Art. 107200, Jul. 2021, doi: 10.1016/j.compeleceng.2021.107200.

[5] A. Morgenshtein, A. Fish, and I. A. Wagner, "Gate-diffusion input (GDI): A power-efficient method for digital combinatorial circuits," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 5, pp. 566–581, Oct. 2002, doi: 10.1109/TVLSI.2002.801578.

[6] K. Navi, M. Maeen, V. Foroutan, S. Timarchi, and O. Kavehei, "A novel low-power full-adder cell for low voltage," Integration, vol. 42, no. 4, pp. 457–467, Sep. 2009, doi: 10.1016/j.vlsi.2009.02.001.

[7] D. Radhakrishnan, "Low-voltage low-power CMOS full adder," IEE Proc. Circuits, Devices Syst., vol. 148, no. 1, pp. 19–24, Feb. 2001, doi: 10.1049/ip-cds:20010170.

[8] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013, doi: 10.1109/TCAD.2012.2217962.

[9] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in Proc. 18th IEEE Eur. Test Symp. (ETS), Avignon, France, May 2013, pp. 1–6, doi: 10.1109/ETS.2013.6569370.

[10] S. Mittal, "A survey of techniques for approximate computing," ACM Comput. Surv., vol. 48, no. 4, Art. 62, pp. 1–33, May 2016, doi: 10.1145/2893356.

[11] W. Yang and H. Thapliyal, "Approximate adiabatic logic for low-power and secure edge computing," IEEE Consum. Electron. Mag., vol. 11, no. 1, pp. 88–94, Jan. 2022, doi: 10.1109/MCE.2021.3053908.

[12] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," IEEE Trans. Comput., vol. C-22, no. 8, pp. 786–793, Aug. 1973, doi: 10.1109/TC.1973.5009159.

[13] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., vol. EC-11, no. 3, pp. 340–346, Jun. 1962, doi: 10.1109/IRETELC.1962.5407919.

[14] F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Block-based carry speculative approximate adder for energy-efficient applications," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 1, pp. 137–141, Jan. 2020, doi: 10.1109/TCSII.2019.2901060.

[15] M. Aguirre-Hernandez and M. Linares-Aranda, "CMOS full-adders for energy-efficient arithmetic applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 4, pp. 718–721, Apr. 2011, doi: 10.1109/TVLSI.2009.2038166.

[16] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED), Fukuoka, Japan, Aug. 2011, pp. 409–414, doi: 10.1109/ISLPED.2011.5993675.