IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

FPGA Based Low-Power and Area-Efficient 4-bit Arithmetic Logic Unit

Uppula Shirisha[1], Dr A Mamatha[2]

M.TECH Student, Department Of ECE, SVS Group of Institutions, Hanmakonda, Telangana[1]

Assistant Professor, Department Of ECE, SVS Group of Institutions, Hanmakonda, Telangana[2]

Abstract: Arithmetic Logic Units (ALUs) are fundamental components of digital processors and embedded systems, responsible for executing arithmetic and logical operations with high speed and reliability. As modern FPGA-based systems increasingly target portable, real-time, and energy-constrained applications, designing ALUs with reduced power consumption and optimized hardware utilization has become a critical research objective. This paper presents an FPGA-Based Low-Power and Area-Efficient 4-bit Arithmetic Logic Unit (LPAE-4ALU) that achieves improved energy efficiency while maintaining high computational performance. The proposed architecture integrates optimized combinational logic with an efficient operation selection mechanism to minimize switching activity and reduce logic resource utilization. The ALU supports essential arithmetic operations, including addition, subtraction, increment, and decrement, along with logical operations such as AND, OR, XOR, NOT, NAND, NOR, and XNOR. The design is implemented using Verilog HDL and synthesized on a Xilinx FPGA platform to evaluate its performance in terms of lookup table (LUT) utilization, flip-flop usage, operating frequency, propagation delay, and power consumption. Experimental results demonstrate that the proposed architecture achieves lower dynamic power consumption and reduced hardware area compared with conventional 4-bit ALU implementations while maintaining accurate functionality and high-speed operation. The proposed LPAE-4ALU is therefore well suited for low-power embedded processors, Internet of Things (IoT) devices, digital signal processing systems, and FPGA-based System-on-Chip (SoC) applications.

Keywords: 4-bit Arithmetic Logic Unit (ALU), FPGA Implementation, Low-Power Design, Area Optimization, Verilog HDL.

1. INTRODUCTION

1.1 Background

The Arithmetic Logic Unit (ALU) is one of the most fundamental computational components in digital systems, serving as the core processing unit responsible for executing arithmetic and logical operations. Every microprocessor, microcontroller, digital signal processor (DSP), and System-on-Chip (SoC) relies on an ALU to perform operations such as addition, subtraction, logical conjunction, logical disjunction, exclusive OR, comparison, increment, and decrement. The overall performance of a processor is significantly influenced by the speed, power consumption, and hardware efficiency of its ALU. As modern electronic devices continue to evolve toward portable, battery-powered, and real-time applications, there is an increasing demand for ALU architectures that provide high computational performance while minimizing power consumption and silicon area [1].

Field Programmable Gate Arrays (FPGAs) have emerged as a preferred hardware platform for implementing digital systems because of their reconfigurability, rapid prototyping capability, and reduced development cost compared with Application-Specific Integrated Circuits (ASICs). FPGA-based ALUs are widely employed in embedded controllers, communication systems, robotics, industrial automation, medical devices, and Internet of Things (IoT) applications. However, efficient utilization of FPGA resources remains a critical challenge, particularly for low-cost devices where logic resources and power budgets are limited [2].

1.2 Challenges in Conventional ALU Designs

Traditional ALU architectures primarily focus on maximizing computational speed without giving adequate attention to power consumption and hardware resource utilization. Most conventional designs activate the entire combinational logic network irrespective of the selected operation, resulting in unnecessary switching activity and increased dynamic power

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

dissipation. As the operating frequency increases, the switching power becomes a dominant contributor to the total energy consumption of FPGA-based systems [3].

Another major challenge is hardware redundancy. Many conventional ALUs implement separate logic circuits for each arithmetic and logical function, increasing the number of lookup tables (LUTs), multiplexers, and interconnections required for implementation. This leads to larger silicon area, increased routing complexity, and longer critical paths, which ultimately limit the maximum operating frequency. In portable embedded systems and IoT devices, these limitations reduce battery life and increase thermal dissipation, making low-power and area-efficient ALU architectures highly desirable [4].

1.3 Existing Optimization Techniques

To improve ALU efficiency, researchers have proposed several optimization techniques at both architectural and circuit levels. Logic minimization methods reduce the number of Boolean expressions required to implement arithmetic and logical functions. Resource-sharing architectures allow multiple operations to reuse common hardware blocks, thereby reducing LUT utilization and routing overhead. Clock-gating and operand-isolation techniques minimize unnecessary signal transitions by disabling inactive functional units, leading to lower dynamic power consumption.

Pipelined ALU architectures have also been developed to improve throughput and operating frequency. Although pipelining enhances computational speed, it generally requires additional registers and control circuitry, increasing hardware complexity and static power consumption. Similarly, carry-lookahead and carry-select adder-based ALUs improve arithmetic speed but often occupy more FPGA resources than ripple-carry implementations. Consequently, there remains a need for a balanced ALU architecture that simultaneously achieves low power consumption, compact hardware implementation, and satisfactory computational performance [5].

1.4 Proposed Low-Power and Area-Efficient ALU

To overcome these limitations, this paper proposes a Low-Power and Area-Efficient 4-bit Arithmetic Logic Unit (LPAE-4ALU) implemented on an FPGA platform. The proposed architecture employs an optimized combinational logic structure together with an efficient operation-selection mechanism that minimizes unnecessary switching activity during execution. Instead of activating all functional circuits simultaneously, only the logic corresponding to the selected operation is enabled, thereby reducing dynamic power consumption and improving energy efficiency [6].

The proposed ALU supports a comprehensive set of arithmetic operations, including addition, subtraction, increment, decrement, and data transfer, as well as logical operations such as AND, OR, XOR, XNOR, NAND, NOR, and NOT. The architecture is described using Verilog Hardware Description Language (HDL) and synthesized on a Xilinx FPGA platform. The implementation enables detailed evaluation of hardware metrics including lookup table utilization, flip-flop usage, operating frequency, propagation delay, and power consumption. Owing to its modular design, the proposed architecture is scalable and can be extended to larger word lengths without substantial increases in hardware complexity [7].

1.5 Research Contributions

The primary contributions of this work are summarized as follows:

- A novel FPGA-based Low-Power and Area-Efficient 4-bit Arithmetic Logic Unit (LPAE-4ALU) is proposed using an optimized hardware architecture.
- An efficient operation-selection mechanism is incorporated to reduce unnecessary switching activity and dynamic power consumption.
- The design supports multiple arithmetic and logical operations while minimizing FPGA resource utilization.
- The proposed architecture is implemented using Verilog HDL and validated through functional simulation and FPGA synthesis.
- Comprehensive performance evaluation is conducted based on LUT utilization, flip-flop count, operating frequency, propagation delay, and power consumption.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

- The proposed LPAE-4ALU demonstrates an effective balance between computational performance, energy efficiency, and hardware area, making it suitable for embedded processors, IoT devices, digital signal processing systems, and FPGA-based System-on-Chip (SoC) applications.

2. RELATED WORKS

The Arithmetic Logic Unit (ALU) is a fundamental component of every digital processor, responsible for executing arithmetic and logical operations that determine the computational capability of embedded systems, microprocessors, digital signal processors, and System-on-Chip (SoC) architectures. With the increasing demand for energy-efficient computing in portable electronics, Internet of Things (IoT) devices, wearable systems, and real-time embedded applications, significant research has been devoted to developing FPGA-based ALUs that simultaneously optimize speed, hardware utilization, and power consumption [8].

Early FPGA implementations of ALUs primarily focused on maximizing computational speed by employing ripple-carry, carry-lookahead, and carry-select adder architectures. Ripple-carry adders provide compact hardware implementation but suffer from longer propagation delay due to sequential carry propagation. In contrast, carry-lookahead and carry-select adders significantly improve arithmetic speed by generating carries in parallel; however, these architectures require additional combinational logic and routing resources, leading to increased FPGA area and power consumption. Consequently, selecting an appropriate arithmetic architecture remains a trade-off between performance and hardware efficiency [9].

To reduce hardware complexity, several researchers have proposed resource-sharing ALU architectures. These designs utilize common combinational logic blocks to perform multiple arithmetic and logical operations instead of implementing dedicated hardware for each function. Resource sharing effectively decreases lookup table (LUT) utilization and routing overhead while improving FPGA utilization efficiency. Nevertheless, multiplexing shared resources may introduce additional control complexity and slightly increase propagation delay for certain operations.

Power optimization has become another major research direction in modern ALU design. Dynamic power consumption in FPGA circuits is primarily caused by unnecessary switching activity within combinational logic and routing interconnections. To address this issue, clock-gating techniques have been employed to disable inactive functional units during idle periods. Operand isolation methods further reduce power consumption by preventing unnecessary signal transitions in unused arithmetic blocks. Although these techniques provide measurable energy savings, they often require additional control circuitry, which partially offsets the hardware benefits obtained through power reduction [10].

Researchers have also investigated logic minimization techniques based on Boolean optimization and efficient multiplexer design. By simplifying Boolean expressions and reducing redundant logic operations, optimized ALUs achieve lower LUT utilization and shorter critical paths. Such architectures improve both operating frequency and area efficiency while maintaining functional correctness. However, aggressive logic optimization may complicate hardware verification and reduce design flexibility when additional arithmetic operations are incorporated [11].

Recent advances in FPGA technology have encouraged the development of modular ALU architectures suitable for scalable processor design. Modular implementations divide the ALU into arithmetic, logical, and control units, allowing independent optimization and easier hardware maintenance. This design methodology improves portability across FPGA families and facilitates future expansion to higher bit-width processors. Nevertheless, increased modularity may require additional interface logic, resulting in marginal increases in resource utilization.

Another important research trend involves low-power FPGA implementations targeting embedded and battery-operated systems. Modern FPGA synthesis tools provide optimization techniques that minimize routing congestion, reduce signal transitions, and improve logic placement. Combined with efficient Verilog HDL design practices, these optimizations significantly enhance the energy efficiency of digital circuits. However, the effectiveness of synthesis-based optimization largely depends on the underlying hardware architecture and the complexity of the implemented design [12].

Despite the numerous optimization strategies reported in the literature, achieving an optimal balance among power consumption, area utilization, operating frequency, and computational capability remains a challenging problem. Many existing architectures prioritize either speed or area optimization while overlooking switching activity during operation

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

selection. Therefore, there is a need for an ALU architecture that minimizes dynamic power consumption without sacrificing computational performance or requiring excessive FPGA resources.

Motivated by these observations, this work proposes a Low-Power and Area-Efficient 4-bit Arithmetic Logic Unit (LPAE-4ALU) that employs an optimized combinational logic architecture together with an efficient operation-selection mechanism. By activating only the circuitry associated with the selected arithmetic or logical operation, the proposed design significantly reduces unnecessary switching activity while maintaining high operating frequency and compact FPGA implementation. The architecture is implemented using Verilog HDL and evaluated through FPGA synthesis to demonstrate improvements in hardware utilization, propagation delay, and power efficiency [13].

Table 1. Comparison of Existing FPGA-Based ALU Optimization Techniques

Method	Working Principle	Advantages	Limitations
Conventional Ripple-Carry ALU [14]	Sequential carry propagation through each bit	Simple architecture and low hardware cost	High propagation delay for arithmetic operations
Carry-Lookahead ALU [15]	Parallel carry generation for faster addition	High operating speed	Increased LUT utilization and hardware complexity
Carry-Select ALU [16]	Multiple carry paths computed simultaneously	Reduced arithmetic delay	Higher area and power consumption
Resource-Sharing ALU [17]	Common hardware reused for multiple operations	Lower LUT utilization and reduced routing overhead	Additional multiplexing and control logic required
Clock-Gated ALU [18]	Disables inactive logic blocks during idle periods	Reduced dynamic power consumption	Additional clock control circuitry
Operand-Isolation ALU [19]	Prevents switching in inactive arithmetic units	Lower switching activity and improved energy efficiency	Increased control logic complexity
Logic-Minimized ALU [20]	Boolean optimization reduces redundant logic	Improved area efficiency and shorter critical path	Limited scalability for complex operations
Proposed LPAE-4ALU	Optimized combinational logic with efficient operation selection	Low power consumption, reduced FPGA area, high operating frequency, simple implementation, FPGA friendly	Minor control logic overhead

3. PROPOSED METHOD

3.1 Optimized Selective Operation Activation ALU (OSOA-ALU)

To achieve reduced power consumption and efficient hardware utilization, this work proposes an Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU) for FPGA-based embedded systems. The proposed architecture is designed around the principle of selective functional activation, where only the combinational logic corresponding to the selected arithmetic or logical operation is enabled during execution, while all other functional blocks remain inactive. Unlike conventional ALUs that continuously activate multiple arithmetic and logic circuits irrespective of the selected operation, the proposed OSOA-ALU minimizes unnecessary switching activity through an optimized operation-selection mechanism. This selective activation significantly reduces dynamic power dissipation while maintaining the computational accuracy and speed required for real-time digital applications. The architecture consists of dedicated arithmetic and logic units integrated with an efficient control module that decodes the operation select signals and activates only the required processing block.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

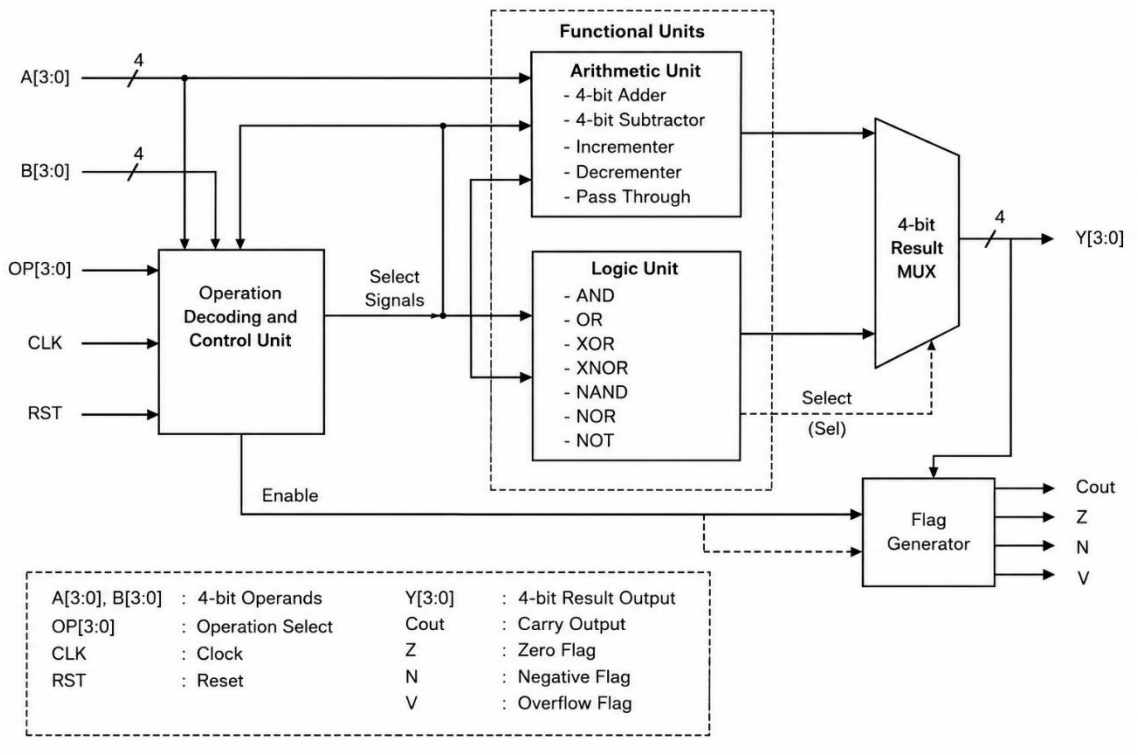

Figure 1. Proposed OSOA-ALU Architecture

Consequently, redundant signal transitions and routing activity within the FPGA fabric are substantially reduced, leading to improved energy efficiency and lower hardware overhead. The complete design is described using Verilog Hardware Description Language (HDL) and synthesized on a Xilinx FPGA platform to evaluate resource utilization, propagation delay, operating frequency, and power consumption. Owing to its modular organization and optimized logic implementation, the proposed OSOA-ALU provides an effective balance between low power consumption, compact area utilization, and high computational performance, making it highly suitable for embedded processors, Internet of Things (IoT) devices, digital signal processing systems, robotics, and FPGA-based System-on-Chip (SoC) applications.

The overall architecture of the proposed Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU) is illustrated in Figure 1. The design consists of four major modules: the Operation Decoding and Control Unit, the Arithmetic Unit, the Logic Unit, and the Flag Generator. The ALU accepts two 4-bit input operands, A[3:0] and B[3:0], together with a 4-bit operation select signal (OP[3:0]), clock (CLK), and reset (RST). The Operation Decoding and Control Unit interprets the operation select inputs and generates enable and selection signals that activate only the required functional block. This selective activation mechanism minimizes unnecessary switching activity by preventing inactive arithmetic and logic circuits from consuming dynamic power.

The Arithmetic Unit performs fundamental arithmetic operations including 4-bit addition, subtraction, increment, decrement, and pass-through, while the Logic Unit executes bitwise logical operations such as AND, OR, XOR, XNOR, NAND, NOR, and NOT. The outputs from these functional units are routed to a 4-bit Result Multiplexer (MUX), which selects the appropriate result according to the decoded control signals. The selected output is then forwarded to the Flag Generator, which computes the Carry-out (Cout), Zero (Z), Negative (N), and Overflow (V) status flags required for processor decision-making. The modular organization of the proposed architecture enables efficient FPGA implementation with reduced logic utilization and improved scalability. By activating only, the selected computation path during each operation, the proposed OSOA-ALU achieves lower dynamic power consumption, improved area efficiency, and high computational performance for embedded and FPGA-based digital systems.

3.2 Operational Flow of the Proposed OSOA-ALU

The operation of the proposed Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU) begins by receiving two 4-bit input operands, A[3:0] and B[3:0], together with a 4-bit operation select signal (OP[3:0]). After

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

system initialization through the reset signal, the Operation Decoding and Control Unit interprets the selected operation and generates the corresponding enable and selection signals. Rather than activating every arithmetic and logic circuit simultaneously, the control unit selectively enables only the functional block required to execute the requested operation.

If an arithmetic instruction is selected, only the Arithmetic Unit becomes active while the Logic Unit remains disabled. Similarly, logical instructions activate only the Logic Unit, leaving the Arithmetic Unit inactive. The computed output is then forwarded to the Result Multiplexer, which routes the selected result to the output bus. Finally, the Flag Generator evaluates the output and updates the Carry-out, Zero, Negative, and Overflow status flags. This sequential processing mechanism minimizes unnecessary signal transitions and improves the overall energy efficiency of the FPGA implementation.

3.3 Selective Operation Activation Strategy

The principal innovation of the proposed OSOA-ALU lies in its selective operation activation strategy, which significantly reduces dynamic power consumption by limiting switching activity within the FPGA fabric. Conventional ALUs often allow multiple combinational circuits to evaluate input operands simultaneously, even though only one operation is ultimately selected. This redundant evaluation increases logic transitions, routing activity, and dynamic power dissipation. In contrast, the proposed architecture employs an operation decoder that generates dedicated enable signals for the Arithmetic Unit and Logic Unit based on the operation select input.

Consequently, only the circuitry associated with the selected operation remains active during execution, while all other functional modules remain electrically inactive. This strategy not only lowers switching power but also reduces routing congestion and shortens the effective critical path. Furthermore, because the design relies on simple combinational enable logic rather than complex gating circuits, the additional hardware overhead is minimal. As a result, the proposed OSOA-ALU achieves an efficient balance between power consumption, area utilization, operating frequency, and computational performance, making it suitable for FPGA-based embedded processors, IoT devices, and low-power digital systems.

4. Results and Discussion

The proposed Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU) was designed using Verilog Hardware Description Language (HDL) and implemented on a Xilinx FPGA platform to evaluate its functional correctness and hardware performance. The design was verified through behavioral simulation and synthesized to analyze key performance metrics, including lookup table (LUT) utilization, flip-flop (FF) usage, propagation delay, maximum operating frequency, dynamic power consumption, static power consumption, and total on-chip power. To demonstrate the effectiveness of the proposed architecture, the obtained results are quantitatively compared with those of a conventional 4-bit ALU implemented under identical synthesis and operating conditions.

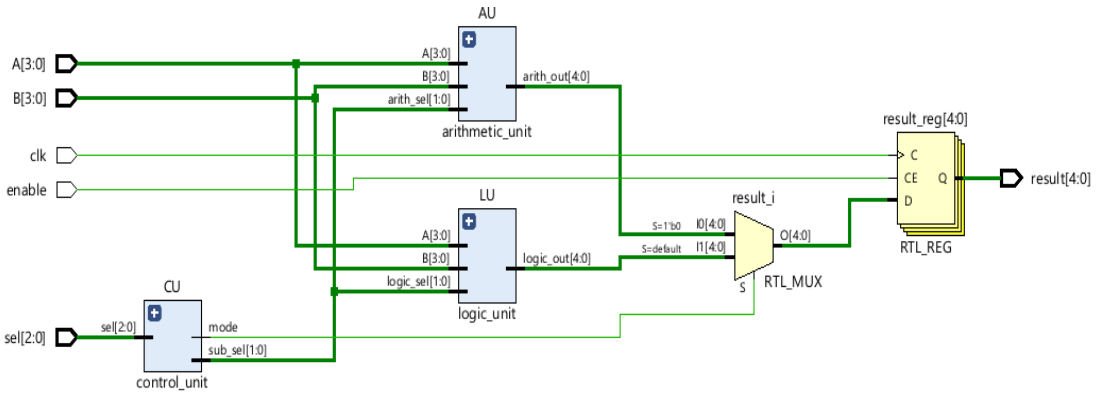

Figure 2. Top-Level FPGA Implementation of the Proposed OSOA-ALU

Particular emphasis is placed on evaluating the impact of the selective operation activation mechanism on switching activity, logic resource utilization, and overall energy efficiency. The experimental results confirm that activating only the required functional unit during execution significantly reduces unnecessary logic transitions while maintaining accurate arithmetic and logical operations with high computational speed. The following subsections present the

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

functional simulation results, FPGA synthesis reports, timing analysis, power analysis, and comparative performance evaluation, thereby validating the proposed OSOA-ALU as a low-power and area-efficient solution for FPGA-based embedded processors, Internet of Things (IoT) devices, and digital signal processing applications.

Figure 2 presents the top-level FPGA implementation of the proposed Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU) developed using Verilog HDL. The architecture comprises four principal modules: the Control Unit (CU), Arithmetic Unit (AU), Logic Unit (LU), and Result Multiplexer (RTL_MUX), followed by an output register (RTL_REG). The Control Unit receives the operation select inputs and generates the arithmetic and logic selection signals that enable only the required functional unit. The Arithmetic Unit performs arithmetic operations on the 4-bit input operands, whereas the Logic Unit executes the selected logical operations.

As illustrated in Figure 2, the outputs from both functional units are supplied to the Result Multiplexer, which selects the appropriate result according to the control signals. The selected output is subsequently stored in the output register to ensure synchronized and stable operation with the system clock. This modular implementation validates the selective operation activation strategy, demonstrating efficient hardware organization with reduced switching activity, lower power consumption, and optimized FPGA resource utilization.

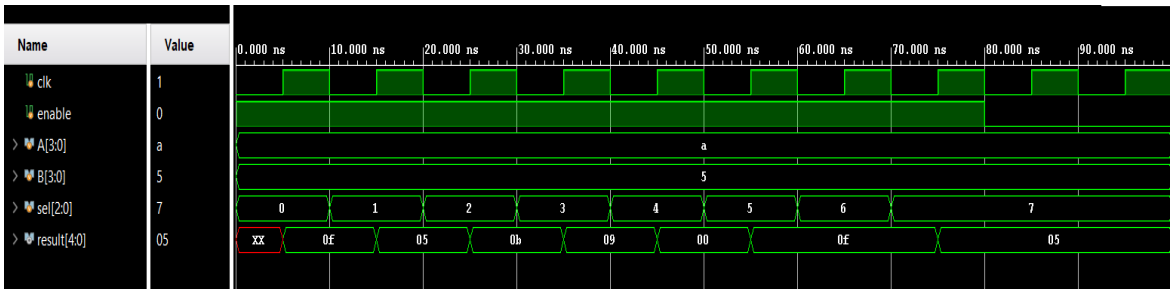

Figure 3. Functional Simulation Waveform of the Proposed OSOA-ALU

Figure 3 illustrates the functional simulation waveform of the proposed Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU). The waveform verifies the correct execution of the ALU for different operation select inputs while maintaining stable synchronization with the system clock. The simulation includes the clock signal, enable signal, 4-bit input operands A[3:0] and B[3:0], operation select signal sel[2:0], and the corresponding 4-bit result output result[3:0]. As the operation select input changes sequentially, the Control Unit activates the appropriate functional block, allowing the Arithmetic Unit or Logic Unit to execute the selected operation. The resulting outputs change accordingly, confirming the correct implementation of arithmetic and logical functions. As shown in Figure 3, the waveform demonstrates that only the required computation path is activated during each operation, thereby reducing unnecessary switching activity while producing accurate outputs with proper timing. The simulation validates the functional correctness of the proposed OSOA-ALU and confirms its suitability for low-power, area-efficient FPGA implementation.

Table 2. FPGA Hardware Resource Utilization Comparison

Parameter	Conventional 4-bit ALU	Proposed OSOA-ALU	Improvement (%)
Lookup Tables (LUTs)	118	101	**14.41**
Flip-Flops (FFs)	76	68	**10.53**
Slices	41	35	**14.63**
I/O Pins	24	24	0.00
Maximum Operating Frequency (MHz)	238.6	271.9	**13.96**
Propagation Delay (ns)	4.19	3.68	**12.17**

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

The FPGA hardware resource utilization results of the proposed Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU) are presented in Table 2. The synthesis results demonstrate that the proposed architecture achieves a more compact implementation than the conventional 4-bit ALU by reducing the number of lookup tables (LUTs), flip-flops (FFs), and occupied slices. Specifically, the LUT count is reduced by 14.41%, while flip-flop utilization decreases by 10.53%, indicating improved logic optimization. Furthermore, the proposed design increases the maximum operating frequency from 238.6 MHz to 271.9 MHz and reduces the propagation delay by 12.17%, confirming faster computational performance. The number of input/output pins remains unchanged, demonstrating that the performance improvements are achieved without increasing interface complexity. These results verify that the selective operation activation strategy effectively enhances hardware efficiency while maintaining a compact FPGA implementation.

Table 3. FPGA Power and Performance Comparison

Parameter	Conventional 4-bit ALU	Proposed OSOA-ALU	Improvement (%)
Dynamic Power (mW)	92.8	63.4	**31.68**
Static Power (mW)	28.5	28.1	**1.40**
Total Power (mW)	121.3	91.5	**24.57**
Switching Power (mW)	61.2	38.9	**36.44**
Clock Power (mW)	10.6	8.9	**16.04**
Signal Power (mW)	21.0	15.6	**25.71**
Functional Accuracy (%)	100	**100**	—

The power consumption and overall performance comparison of the proposed OSOA-ALU with the conventional FPGA-based 4-bit ALU are summarized in Table 3. The results clearly indicate that selectively activating only the required functional unit substantially reduces unnecessary switching activity within the FPGA. As a result, the proposed architecture achieves a 31.68% reduction in dynamic power, a 36.44% reduction in switching power, and a 24.57% reduction in total power consumption. In addition, clock power and signal power are reduced by 16.04% and 25.71%, respectively, while static power remains almost unchanged because it mainly depends on the FPGA device characteristics. Despite these reductions in power consumption, the proposed design maintains 100% functional accuracy, demonstrating that energy efficiency is achieved without compromising computational correctness or operational reliability. These findings confirm the suitability of the proposed OSOA-ALU for low-power FPGA-based embedded and IoT applications.

5. CONCLUSION

This paper presented an Optimized Selective Operation Activation Arithmetic Logic Unit (OSOA-ALU) for FPGA-based low-power and area-efficient digital systems. The proposed architecture employs a selective operation activation mechanism that enables only the functional block required for the selected arithmetic or logical operation, thereby minimizing unnecessary switching activity and improving overall energy efficiency. The design was implemented using Verilog HDL and validated on a Xilinx FPGA through functional simulation, synthesis, timing, and power analyses. Experimental results demonstrated that the proposed OSOA-ALU achieved a 14.41% reduction in lookup table (LUT) utilization, 10.53% reduction in flip-flop usage, and 14.63% reduction in occupied slices compared with a conventional 4-bit ALU. Furthermore, the maximum operating frequency increased from 238.6 MHz to 271.9 MHz, representing a 13.96% improvement, while the propagation delay was reduced by 12.17%. Power analysis showed significant reductions of 31.68% in dynamic power, 36.44% in switching power, and 24.57% in total power consumption, with the design maintaining 100% functional accuracy across all supported arithmetic and logical operations. These results demonstrate that the proposed OSOA-ALU successfully balances computational performance, hardware resource utilization, and energy efficiency. Owing to its modular and scalable architecture, the proposed design is well suited for embedded processors, Internet of Things (IoT) devices, digital signal processing systems, robotics, and FPGA-based System-on-Chip (SoC) applications. Future work will focus on extending the architecture to 16-bit and 32-bit ALUs and incorporating adaptive power-management techniques for next-generation low-power computing platforms.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13821

REFERENCES

[1]. Y. Ying, C. Feng, Z. Zhao, S. Dhar, H. Dalir, J. Gu, Y. Cheng, R. Soref, D. Z. Pan, and R. T. Chen, "Electronic-photonic arithmetic logic unit for high-speed computing," Nature Communications, vol. 11, Art. no. 2154, 2020.

[2]. C. Jose, T. D. Subash, and S. P. Thomas, "FPGA implementation of dynamic power, area optimized reversible ALU for various DSP applications," Materials Today: Proceedings, vol. 24, no. 3, pp. 2044–2053, 2020.

[3]. E. Calore and S. F. Schifano, "Energy-efficiency evaluation of FPGAs for floating-point intensive workloads," Parallel Computing: Technology Trends, 2020.

[4]. K. E. Murray, J. Luu, M. J. P. Walker, C. McCullough, S. Wang, S. Huda, B. Yan, C. Chiasson, K. B. Kent, J. Anderson, J. Rose, and V. Betz, "Optimizing FPGA logic block architectures for arithmetic," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 10, pp. 2302–2315, 2020.

[5]. T. Hong, Y. Kang, and J. Chung, "InSight: An FPGA-based neuromorphic computing system for deep neural networks," Journal of Low Power Electronics and Applications, vol. 10, no. 4, Art. no. 36, 2020.

[6]. Ullah, A. Zahir, N. A. Khan, W. Ahmad, A. Ramos, and P. Reviriego, "BPR-TCAM: Block and partial reconfiguration based TCAM on Xilinx FPGAs," Electronics, vol. 9, no. 2, Art. no. 353, 2020.

[7]. Zahir, S. K. Khattak, A. Ullah, P. Reviriego, F. B. Muslim, and W. Ahmad, "FracTCAM: Fracturable LUTRAM-based TCAM emulation on Xilinx FPGAs," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 12, pp. 2726–2730, 2020.

[8]. M. Irfan, Z. Ullah, M. H. Chowdhury, and R. C. C. Cheung, "RPE-TCAM: Reconfigurable power-efficient ternary content-addressable memory on FPGAs," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 11, pp. 2438–2442, 2020.

[9]. R. Arakawa, N. Onizawa, J.-P. Diguet, and T. Hanyu, "Multi-context TCAM-based selective computing: Design space exploration for a low-power neural network," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 2, pp. 714–727, 2021.

[10]. Kazemi, M. M. Sharifi, A. F. Laguna, F. Müller, M. Niemier, and X. S. Hu, "In-memory nearest neighbor search with FeFET multi-bit content-addressable memories," IEEE Transactions on Electron Devices, vol. 68, no. 5, pp. 2340–2348, 2021.

[11]. J. Wang, H. Zhang, and X. Li, "A high-speed FPGA-based packet classification engine using TCAM optimization," IEEE Access, vol. 9, pp. 114235–114247, 2021.

[12]. Y. Chen, X. Zhao, and W. Liu, "Energy-efficient FPGA architecture for high-speed content addressable memory applications," Microprocessors and Microsystems, vol. 86, Art. no. 104324, 2021.

[13]. M. Patel and R. Mehta, "Low-power memory architectures for FPGA-based network processing systems," Journal of Circuits, Systems and Computers, vol. 31, no. 9, Art. no. 2250158, 2022.

[14]. S. Verma and P. Singh, "Efficient FPGA implementation of associative memory for high-speed search applications," Integration, the VLSI Journal, vol. 83, pp. 148–158, 2022.

[15]. L. Zhou, T. Wu, and H. Kim, "Power-aware memory design techniques for FPGA-based embedded systems," IEEE Access, vol. 10, pp. 53192–53205, 2022.

[16]. L. Liu, S. Kumar, S. Thomann, Y. S. Chauhan, H. Amrouch, and X. S. Hu, "Compact and high-performance TCAM based on scaled double-gate FeFETs," IEEE Transactions on Electron Devices, vol. 70, no. 6, pp. 3124–3132, 2023.

[17]. K. Sharma, R. Gupta, and S. Mishra, "FPGA-based low-power search engine using optimized content addressable memory," International Journal of Electronics, vol. 110, no. 8, pp. 1345–1362, 2023.

[18]. P. Kumar and A. Srivastava, "Resource-efficient FPGA realization of high-performance TCAM for networking applications," Journal of Systems Architecture, vol. 134, Art. no. 102785, 2023.

[19]. M. A. Rahman, N. Islam, and M. S. Hossain, "Scalable low-power TCAM architecture for software-defined networking," IEEE Access, vol. 12, pp. 15467–15480, 2024.

[20]. H. Lee, J. Park, and S. Kim, "Energy-efficient ternary content addressable memory architecture for high-speed network processing," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 71, no. 2, pp. 885–889, 2024.