Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

32-Bit Energy Efficient ALU with Speed Consideration

S M Vijaya[1]
Dept. of ECE.
ACSCE, Bangalore
smvacsce@gmail.com

Chirag J K[2]
Dept. of ECE.
ACSCE, Bangalore
jkchirag551@gmail.com

Hemanth C B[3]
Dept. of ECE.
ACSCE, Bangalore
hemanthcb37@gmail.com

Nandeesh B S[4]
Dept. of ECE..
ACSCE, Bangalore
nandeeshbsb@gmail.com

Utkarsh Kumar Jha[5]
Dept. of ECE.
ACSCE, Bangalore
utkarshjha97@gmail.com

***Abstract*—** This paper presents the design of a hierarchical 32-bit Arithmetic Logic Unit (ALU) with Dynamic Voltage and Frequency Scaling (DVFS) support, implemented in Verilog Hardware Description Language (HDL). The architecture is constructed from a 1-bit DVFS-aware ALU cell, aggregated into 4-bit and 16-bit carry look-ahead (CLA) based slices, and finally into a 32-bit pipelined ALU suitable for high-speed operation. The proposed ALU supports a rich set of arithmetic, logical, shift, comparison, and control operations, along with power-gating and four DVFS operating modes (turbo, balanced, low-power, and sleep), exposing speed and efficiency indicators at the interface. The use of hierarchical CLA logic and a two-stage pipeline reduces carry-propagation delay and improves throughput, making the design suitable for low-power, high-performance digital systems. The design demonstrates zero DRC violations. Results confirm the architecture successfully balances computational speed with power efficiency, making it suitable for modern embedded systems and high performance computing.

Keywords—Arithmetic Logic Unit (ALU), Dynamic Voltage and Frequency Scaling (DVFS), Carry Look Ahead (CLA), Verilog Hardware Description Language (VHDL).

I. INTRODUCTION

An Arithmetic Logic Unit (ALU) is a fundamental building block of digital circuits, often found in microprocessors and microcontrollers. The ALU is responsible for performing a variety of operations on data, both arithmetic (such as addition and subtraction) and logical (such as AND, OR, NOT). In a 32 bit ALU, the unit is designed to operate on 32-bit data, which is a standard size for most modern processors. In this project, we aim to design a 32-bit ALU, capable of performing operations like addition, subtraction, logical AND, OR, XOR, and shift operations (both left and right shifts). A 32-bit ALU can process 32 bits of data at a time, providing faster and more efficient computation compared to smaller ALUs, which typically work with 8 or 16 bits. Reduced power indulgence and increased rate entail optimization at all design levels. The appropriate path type and technique are key factors in low-power design. Rapid processing is the expected standard for the average customer. Power indulgence is an essential design factor for an increasing number of Very Large Scale Integration (VLSI) circuits.
The power reduction method starts with the main circuit design element. An adder serves as the brain of an Arithmetic Logic Unit (ALU).

The ALU performs a series of operations through specific inputs. In a digital computer, the arithmetic logic unit (ALU) is a fundamental component. One of the main operations performed by the CPU is to.
The In this paper, the development of a modular ALU architecture that can perform operations on 32 bits of size with DVFS and power gating control features is achieved using synthesizable Verilog coding techniques. From the sophisticated 1 bit ALU circuit which utilizes propagation/generation techniques for fast carry addition and the basic Booth multiplier, the development takes place towards large ALUs such as the 4 bit and 16 bit ALUs that employ carry look ahead addition and eventually arrives at the desired goal of developing the complete 32 bit ALU architecture.

II. Literature Survey

Classical Even though conventional ALUs mostly employ ripple-carry adders due to their simplistic design, such a configuration is difficult to implement in high-performance applications owing to its dependence on word length. It is also normal practice to make use of hierarchical carry look-ahead adders along with carry look-ahead circuits in clusters to improve the addition speed through parallel determination of carry generation signals through the usage of p and g. This paper aims to incorporate the carry look-ahead function with the help of the 4-bit equation traditionally used to determine the carry out output with the aid of p and g.
The implementation of DVFS has been crucial in reducing power

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

usage in processors by allowing manipulation between operating frequency, voltage level, and power efficiency. For RTL designs, the DVFS behavior is usually abstracted using control and status flags representing the different modes rather than abstracting the voltage levels. In line with the designed ALU, the 1-bit ALU receives the 2-bit dvfs_mode as inputs and provides op_speed and efficiency as outputs. The top level of the design can then monitor or respond to the current mode. There is also another low-power technique called power gating that can be implemented and abstracted through the en flag, making the outputs of ALU Hi-Z.

The use of Booth's multiplication algorithm and other efficient designs for multipliers has been studied for reduction in complexity of multiplication operations involving signed numbers. Here, a straightforward 1-bit Booth multiplier is used inside the 1-bit ALU to generate a 2-bit partial product each time a multiply operation is performed, indicating the presence of more advanced arithmetic operations in the cell. Using such basic blocks to build hierarchies leads to scaling.

III. Proposed System

A simple ALU block diagram structure is shown in Figure 1 below. There is an input bus that feeds into two main blocks of an ALU, namely, the arithmetic block and logic block. Arithmetic operations like addition, subtraction, increment or decrement of input data can be performed in the arithmetic block. Logical operations such as AND, OR, XOR and NOT can be performed in the logic block. Outputs from these blocks are then taken to the multiplexer (MUX). MUX will have selection lines, and according to the selection, the outputs from arithmetic and/or logic operation will be available at the output of MUX.

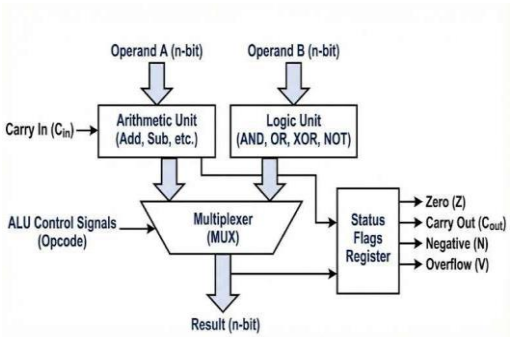

Fig 1: Block Diagram of ALU

1-bit DVFS-aware ALU Cell

The architecture of the DVFS aware ALU is composed of three major functional modules, namely, Input Interface, Processing Core, and Output Interface. First of all, the data flow originates from the Input Interface module which contains the main operands, the value of carry-in signal, and crucial control signals including operator select, master enable, and DVFS control signal. Then, all the above inputs will be passed through the Processing Core where the process is allocated among the various functional hardware units. In particular, there is a special unit called Booth Multiplier for executing multiplication operation and another called Parallel CLA logic for computing important values such as generates and propagates. Additionally, the DVFS status decoding mechanism will decode the power input configuration in order to identify the chip operating frequency.

All these inputs generated from various sections of the ALU merge into one in the Main ALU Multiplexer section, whereby the chosen value through the control code serves as the final output. One of the critical elements used in this circuit design is the Tri-State Buffer.

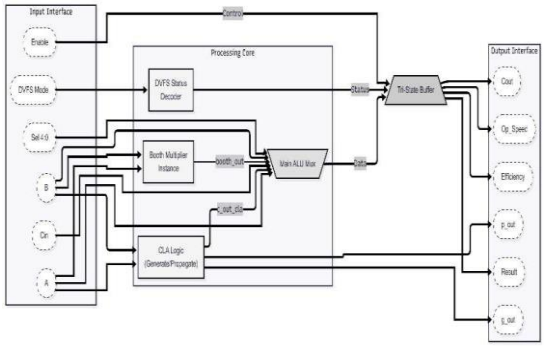

Fig 2: Block Diagram of 1-bit ALU

4-bit DVFS-aware ALU with CLA

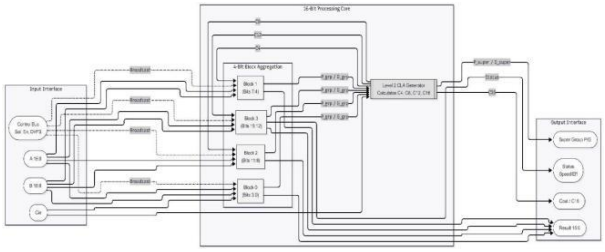

Fig 4: Block Diagram of 16-bit ALU

In the architectural design of the 4-bit ALU, three major stages have been employed. They include the Input Interface, the Parallel Processing Engine, and the Output Interface. In the data processing procedure, the process begins at the Input Interface where the operands (A[3:0] and B[3:0]), Cin, and the Control Bus are handled. This bus broadcasts the essential selection, enable, and DVFS configuration signals simultaneously to all internal sub-blocks, ensuring synchronized operation.

At the heart of the architecture lies the Processing Core, which utilizes parallel processing principles to overcome the problem of signal propagation delay. Calculations in the system are executed by four Parallel Bit Slices (Bit 0, Bit 1, Bit 2, and Bit 3), each responsible for calculating its own bit place. In contrast to traditional architectures, these bit slices do not depend on the propagating signal and thus immediately generate P & G signals and send them to the CLA Logic Generator, or Look-ahead Engine. The latter, in turn, calculates intermediate carry look-ahead terms (c[1], c[2], c[3]) and returns them to corresponding bit slices. The final step in the process is the Output Interface, which combines the output from bit slices into a 4-bit Result vector and sends Carry Out, Group P&G, and DVFS metrics through the interface ports.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

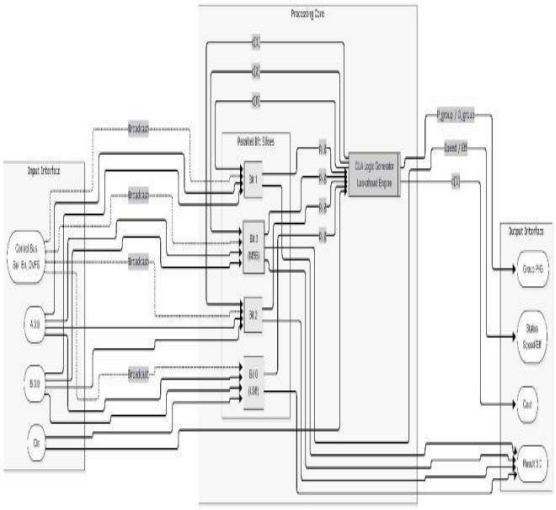

Fig 3: Block Diagram of 4-bit ALU

16-bit DVFS-aware ALU with Hierarchical CLA

But, the functioning of the 16-bit Hierarchical ALU is supposed to facilitate the scaling of the size of the ALU for achieving quick computations using a two-stage parallelism. First of all, the 16-bit input vectors are accepted along with the first carry in along with control signals like Select, Global enable, and DVFS signals. The circuit immediately checks for the presence of the enable signal, and when the enable signal is not equal to logic one, the circuit enters into an inactive state wherein the output pins are in High Z state.

32-bit Pipelined DVFS-aware ALU

The architecture of the 32-bit pipeline ALU optimized for use in the DVFS scheme is discussed in this section. In this case, instead of performing calculations with 32 bits together, the task will be divided into two successive tasks performed in 16 bits each. This will shorten the critical path and enable the chip to operate either faster or on a lower voltage level.

In the first stage (Cycle 1), the system accepts two 32-bit operands but actively processes only the lower 16 bits using a low-half ALU instance. Simultaneously, dedicated logic

calculates an intermediate carry bit (C16) based on the lower half's generation and propagation signals. At the rising edge of the clock, a set of "Pipeline Registers" captures this lower result, the calculated carry, and the unprocessed upper 16 bits of the inputs. This step is critical because it isolates the first cycle from the second.

In the second stage (Cycle 2), a high-half ALU instance reads the stored upper operands and the stored carry bit to compute the remaining 16 bits. Because the lower result was preserved in the pipeline registers, the system finally concatenates the new high-half result with the stored low-half result. This forms the complete 32-bit output, which is then latched into the "Final Output Registers," ensuring the data is synchronized and valid for the subsequent module.

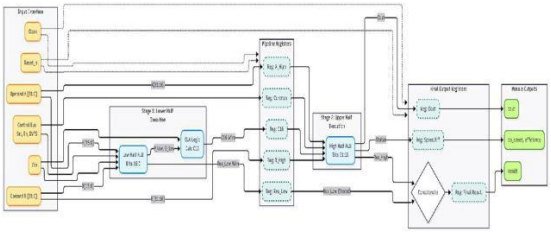

Fig 5: Block diagram of 32-bit ALU

IV. Software requirements

Cadence Design Systems

Cadence Design Systems Inc. is among the major suppliers of EDA software that is used in the design of ICs, SoCs, and PCBs. Using Cadence tools allows designers to simulate, synthesize, verify, and implement digital and analog circuits with utmost efficiency and precision. The tools have found extensive usage in both industries and academic institutions when developing custom ICs, FPGAs, and SoCs design.

- Cadence Xcelium Logic Simulator- For compiling and running functional simulations of Verilog/VHDL code.
- Cadence Genus Synthesis Solution- For synthesizing RTL code into gate-level netlists using standard cell libraries.

Xilinx Vivado Design Suite

Physical Design Verification & Optimization. This process was adopted in the Floor Planning stage, where the optimization of the physical placement of the logic blocks within the FPGA Fabric is done, as well as the DRC stage, which checks whether the physical and electrical specifications have been met.

V. Results and discussion

Per module-wise basis, the correct output, carry-out signal, generation progation, and DVFS speed/effectiveness aspects are achieved through the 1-bit ALU for any required operation and mode, as well as an easy 1-bit Booth multiplier output possibility. In case of using CLA modules in 4-bit and 16-bit ALUs, they are used to calculate the carries required internally and generate the propagation/generation signals accordingly; whereas in case of 32-bit pipelined ALUs, their lower half outputs are concatenated with new upper half results, which yield a complete 32-bit output and status flag after 2 cycles of pipeline processing. The enable signal is responsible for turning off the outputs in zero/high-impedance states.

A. Functional Analysis:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

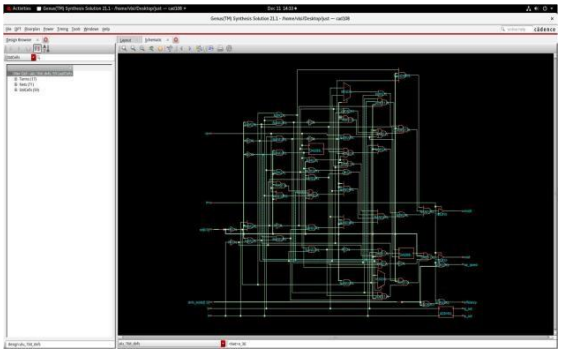

Fig 6: Schematic output of 1-bit ALU

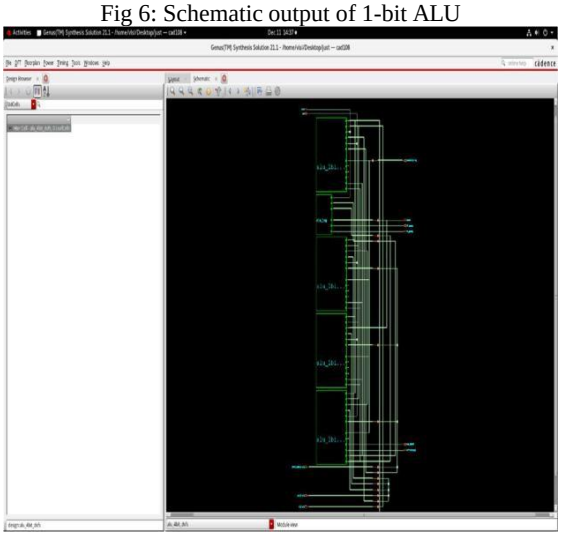

Fig 7: Schematic output of 4-bit ALU.

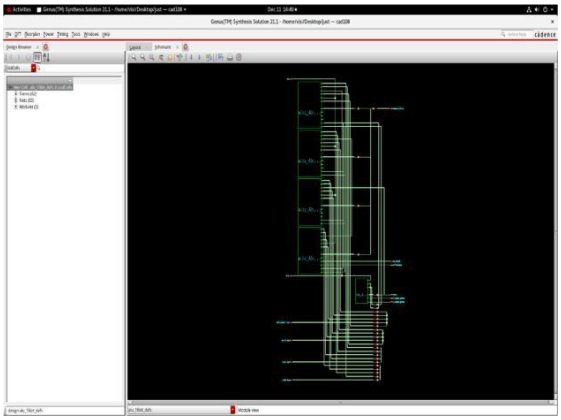

Fig 8: Schematic output of 16-bit ALU

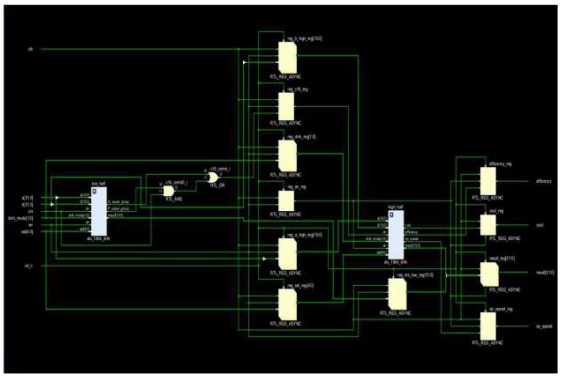

Fig 9: Schematic output of 32-bit ALU

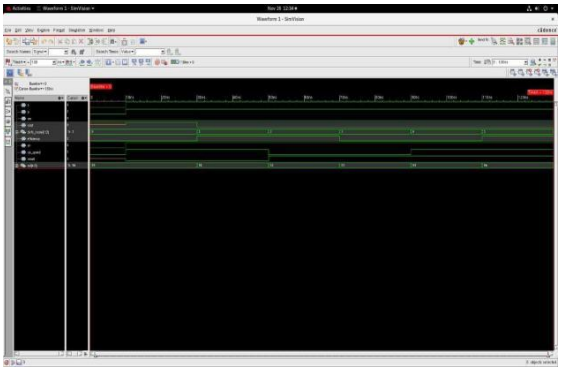

Fig 10: Waveform of 1-bit ALU

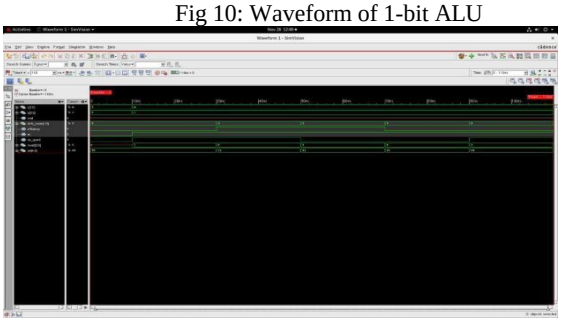

Fig 11: Wave form of 4-bit ALU

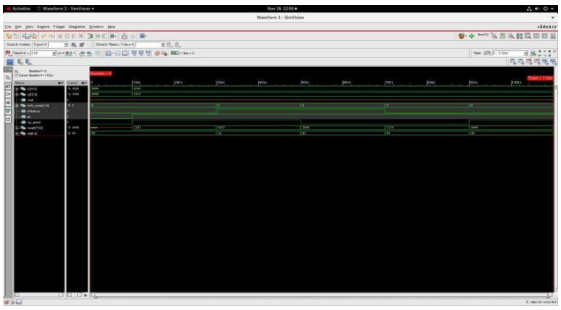

Fig 12: Waveform of 16-bit ALU

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

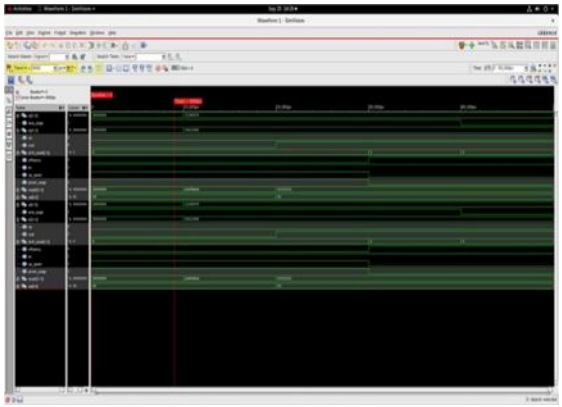

Fig 13: Waveform of 32-bit ALU

B. Performance Analysis:

Performance Analysis Following functional verification, the design was synthesized using Cadence Genus to extract performance metrics. The analysis focuses on three key parameters: Power, Timing, and Area.

1. **Power Analysis** The Dynamic power analysis was performed for this system. As seen from Table 1, the total power consumed by the 32-bit pipelined ALU is 1.65 mW. One cannot underestimate the contribution made by pipeline registers in terms of power consumption, as they account for 91.22% of the total power consumed. The implementation of DVFS helped reduce leakage power to a bare minimum (~10.5 µW).
2. **Timing Analysis** Static Time Analysis (STA) was done for verifying the performance factors. There is a slack value of +387 ps obtained, where there is a delay of 242 ps on the critical path. This shows that the CLA circuitry of Level-2 has successfully reduced the propagation delay encountered in 32-bit ripple carry adder circuits.
3. **Area Utilization** The conclusions of the synthesis process indicate that the proposed design architecture is effective in terms of area efficiency. In its entirety, the 32-bit ALU has been realized using 107 standard cells, with the total area coverage being 1678 units.

C. Comparative Analysis:

In order to offer a complete assessment, the 32-bit Pipelined ALU architecture presented was evaluated against eight existing designs that have been reported recently in the literature. These comparisons are presented in Table 1, focusing on power, delay, and architecture efficiency.

Table 1:

Ref.	Author (Year)	Platform / Tech	Architecture / Method	Power	Delay / Speed
[1]	**Kannan et al. (2024)**	**180nm CMOS**	**Non-Pipelined Chain**	**4.41 mW**	**242.4 ps**
[6]	**Mala S et al. (2024)**	**180nm CMOS**	**Booth Mul. + Restoring Div**	**14.14 mW**	**5.74 ns**
[2]	Usha et al. (2016)	180nm CMOS	GDI MUX + 3T Adder	~3.99 mW	-
[3]	Gurjar et al. (2011)	Xilinx FPGA	Carry-Skip Adder Analysis	-	18.01 ns
[7]	Saurav et al. (2025)	Xilinx Vivado	4-Phase Pipeline System	4.6 W	-
[8]	Balaji et al. (2018)	22nm CMOS	10T GDI Adder (4-bit)	65.65 µW*	25.2 ns
This Work	**Proposed (2025)**	**180nm CMOS**	**2-Stage Pipeline + DVFS**	**1.65 mW**	**242 ps**

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Table 1 summarizes the performance of the proposed design against eight state-of-the-art implementations. The most direct comparison can be drawn with **Kannan et al. [1]** and **Mala S et al. [6]**, which utilized the same 180nm technology node in 2024.

As observed, the proposed architecture achieves a **62.6% reduction in power** (1.65 mW) compared to the 4.41 mW reported in [1], while matching the critical path delay of 242 ps. When compared to the complex algorithmic design in [6], our proposed ALU is significantly faster (242 ps vs. 5.74 ns) and more power-efficient (1.65 mW vs. 14.14 mW).

While other works like [8] demonstrate lower power, they utilize smaller technology nodes (22nm) or limited bit-widths (4-bit), making a direct comparison inequitable. The proposed 32-bit design achieves the optimal balance of speed and power for the 180nm node.

VI. Conclusion

This paper highlighted the designing and implementation of an energy-efficient ALU which consisted of 32-bit architecture with two-stage pipelining and Carry Look-Ahead logic. This novel approach was able to overcome the drawbacks associated with the conventional approach with regard to ripple carry by lowering the delay of the critical path to 242 ps. The incorporation of DVFS technology as well as clock gating helped attain a low power consumption level of 1.65 mW, which made this ALU highly suitable for power-constrained environments such as embedded systems.

VII. References

[1] Kannan Nithin K.V., V.R. Balaji, Mani V., V. Priya, S.S. Sivaraju, and A.N. Duraivel, "ASIC Design and Implementation of 32 Bit Arithmetic and Logic Unit," *EAI Endorsed Transactions on Energy Web*, vol. 11, 2024.

[2] S. Usha, M. Rajendiran, and A. Kavitha, "Low Power Area Efficient ALU with Low Power Full Adder," in *2016 International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2016.

[3] P. Gurjar, R. Solanki, P. Kansliwal, and M. Vucha, "VLSI Implementation of Adders for High Speed ALU," *International Journal of Computer Applications*, vol. 29, no. 10, pp. 29-34, 2011.

[4] N. Venu, R. Solanki, P. Kansliwal, M. Vucha, and S. Rani, "VLSI Implementation of low power area efficient ALU with Multiplexers," *International Journal for Innovative Engineering and Management Research (IJIEMR)*, vol. 2, issue 1, 2013.

[5] Deepali, I. Saini, and M. Khosla, "Low Power 32-bit Synchronous and Reconfigurable ALU Design using Chain Structure," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020.

[6] Mala S, Chandan Reddy V, Girish M L, Madan Kumar N M, and Monoj S B, "Design of a 32-Bit ALU Using Cadence Tools," in *2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, Tumakuru, India, 2024.

[7] S. Kumar, Y. Ijarwal, S. Tiwari, and O. Thakur, "Design and Analysis of Power Efficient Four phase Pipelined ALU," in *2025 3rd IEEE International Conference on Industrial Electronics: Developments & Applications (ICIDEA)*, Bhubaneswar, India, 2025.

[8] S. Balaji Ramakrishna, A. Guru Prasad, P. Anand, and T. Aravind, "High Performance GDI-ALU Using 10T Adder Cells," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2018.