International Journal of Innovative Trends in Engineering Science and Technology
www.ijitest.org | E-ISSN: 3139-6887

Design and Implementation of an Adaptive Execution Unit (AEU): A Low-Power Modular Architecture with Complete RTL-to-GDSII Implementation

[1]Kutikuppala Rajasekhar, [2]T Ravi Babu, [3]T Chandra Guptha
[1]*PG Scholar*, [2]Associate Professor, [3]Assistant Professor
[1,2,3] Department of Electronics and Communication Engineering
[1,2,3] Miracle Educational Society Group of Institutions, Bhogapuram-535216, AP, India

***Abstract* — The Arithmetic Logic Unit (ALU) is a fundamental component of modern processors, responsible for arithmetic, logical, comparison, and shift operations. Conventional ALU architectures face rising power consumption, limited scalability, and inefficient hardware utilization, and most existing designs stop at RTL functionality without addressing complete ASIC implementation. This work presents the design and implementation of an Adaptive Execution Unit (AEU), a modular execution architecture that integrates dedicated arithmetic, logical, multiplier, divider, comparator, and barrel-shifter modules under a centralized control unit. A dynamic precision controller activates only the datapath slices required by an operation, while integrated clock gating and operand isolation reduce unnecessary switching activity. The design is described in synthesizable Verilog HDL and implemented through an industrial RTL-to-GDSII flow using Cadence Genus for synthesis and Cadence Innovus for physical design, covering floorplanning, placement, clock-tree synthesis, routing, timing analysis, and physical verification. Results confirm positive timing slack, 97.5% clock-gated flip-flops, 0% routing congestion, and full antenna and DRC/LVS compliance, demonstrating an efficient, scalable, and implementation-aware execution architecture suitable for modern VLSI processors.**

***Index Terms* -** Adaptive Execution Unit, ALU, Clock Gating, Pipelining, RTL-to-GDSII, Cadence Genus, Cadence Innovus, Low-Power VLSI.

I. INTRODUCTION

Modern digital systems require execution units that simultaneously deliver high computational performance, low power consumption, and efficient hardware utilization. The ALU is one of the most critical components of any processor, and as technology has scaled into deep-submicron nodes, power dissipation, timing closure, routing congestion, and clock distribution have become increasingly significant design concerns. Conventional ALUs execute all supported operations using fixed hardware resources regardless of workload, resulting in unnecessary switching activity and inefficient resource use. In addition, most academic ALU designs conclude at RTL simulation or logic synthesis without investigating floor planning, placement, clock-tree synthesis, routing, congestion, antenna verification, or timing closure — the stages that determine real industrial feasibility.

This work proposes an Adaptive Execution Unit (AEU) that combines architectural innovation — modular execution units, pipelined datapaths, and a dynamic precision controller — with integrated clock gating for dynamic power reduction. Unlike prior work, the proposed architecture is validated through a complete industrial RTL-to-GDSII flow using Cadence Genus and Cadence Innovus, bridging the gap between academic RTL design and industrial ASIC implementation.

II. RELATED WORK

ALU architectures have been reported in the literature, targeting either transistor-level efficiency or architectural optimization. Representative approaches include:

- A 16-bit RISC-V ALU implemented on Cadence tools with a conventional single-stage datapath and no pipelining, clock gating, or physical-design analysis [1].
- A 32-bit reconfigurable ALU using a chain structure for hardware reuse, limited to RTL-level power/area analysis [2].
- An 8-bit ALU built with the m-GDI technique for transistor-count and delay reduction, without architectural-level optimization [3].
- An application-specific low-power ALU that trims redundant functional units but lacks adaptive execution or pipelining [4].

DOI: 10.5281/zenodo.22115427
IJITEST-2026-017| Volume 1, Issue 5, July 2026 | ISSN: 3139-6887

Article Received: July 3, 2026|Accepted: July 27, 2026

- A CPL-based dual-supply 32-bit ALU targeting sub-180 nm CMOS through transistor-level voltage scaling [5].
- A 4-bit ALU using Dual Mode Pass Transistor Logic for high-speed, low-power gate-level design [6].
- An IoT-centric low-power ALU emphasizing power-aware RTL methodology without multiplier/divider units [7].
- An ALU implemented through a full RTL-to-GDSII flow with FPGA validation, but without adaptive execution or clock gating [8].
- An area/power-optimized ALU using an optimized signed multiplier, with conventional (non-pipelined) remaining units [9].
- A ring-counter-controlled ALU that simplifies control logic but not physical implementation [10].
- A low-power RISC-V-compatible ALU limited to the base integer instruction set without pipelining or clock gating [11].

Across these works, common limitations recur: fixed (non-adaptive) execution hardware, absence of integrated clock gating and multi-stage pipelining, and evaluation restricted to RTL or logic synthesis without floorplanning, placement, CTS, routing, congestion analysis, antenna verification, or signoff. The proposed AEU addresses these gaps by combining a dynamic precision controller, integrated clock gating, and multi-stage pipelining with a complete RTL-to-GDSII implementation using Cadence Genus and Innovus.

III. Proposed Method

A. Design Philosophy

The AEU targets high performance reduced dynamic power, and ASIC implementation feasibility together. Rather than treating every instruction identically, the architecture analyzes operand-width requirements before execution and activates only the functional resources needed for a given instruction, using configurable execution slices — an 8-bit operation activates the lowest slice, 16-bit activates two slices, and 32-bit enables the full datapath, with unused slices isolated through operand masking and clock gating.

B. Architectural Organization

The AEU consists of an instruction decoder, control finite-state machine, dynamic precision controller, operand pre-processing block, arithmetic execution core, multiplier, divider, logic unit, barrel shifter, status register, pipeline registers, forwarding network, and write-back interface, as shown in Fig. 1. The modular hierarchy enables independent verification of each subsystem and allows synthesis tools to optimize hierarchy boundaries.

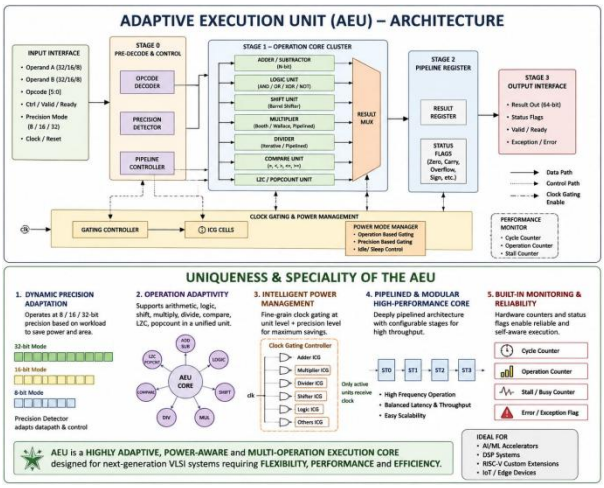

Figure 1: AEU architecture

C. Dynamic Precision Control and Low-Power Techniques

The precision controller monitors the instruction format and generates mutually exclusive 8-/16-/32-bit enable signals as combinational logic, avoiding added pipeline latency. These enables drive integrated clock-gating cells around inactive pipeline stages and operand-isolation multiplexers that hold inactive functional-unit inputs constant, which is particularly effective for the multiplier and divider given their large combinational depth.

D. Pipeline Architecture

The datapath is partitioned into fetch, decode, operand-preparation, execute, and write-back stages. Pipeline registers isolate critical combinational paths to improve maximum operating frequency, while forwarding logic minimizes stalls for dependent instructions and synchronized status flags preserve correctness.

E. RTL Implementation

The RTL is written in synthesizable Verilog with parameterized data path width, opcode size, and pipeline depth. Sequential logic uses non-blocking assignments, combinational logic uses complete sensitivity constructs, and reset behaviour is explicit to avoid latch inference. The top-level module aeu_top instantiates a hybrid adder, a Wallace-tree multiplier, an iterative FSM-controlled divider, a logic unit, a barrel shifter, a comparator, and auxiliary units (population count, leading-zero counter, priority encoder, CRC), coordinated by a pipeline controller and a performance counter. Key design relations used to evaluate the architecture are:

$$P_{dynamic} = \alpha \times C \times V^2 \times f$$
$$P_{leakage} = I_{leak} \times V \quad (1)$$
$$Clock_{gating\ efficiency} = \left(P_{without\ gating} - P_{with\ gating}\right) / P_{without\ gating} \times 100\%.$$

Table I summarizes the Cadence Innovus physical implementation results.

Metric	Result
Core utilization	~89%
Floorplan / power-planning DRC	0 errors / 0 violations
Placement congestion	0% overflow
Placement setup / hold	MET / MET
Clock-tree levels / sinks	5 / 240
Post-route setup / hold	MET / MET
Physical instances / area	4320 / 14063.72 units²
Total power / leakage share	2.182 units / 0.026%
Antenna verification	5000 nets, all compliant

F. Implementation Flow

The RTL is synthesized with Cadence Genus (RTL analysis and elaboration, SDC-based constraint application, generic optimization, technology mapping, and timing-driven optimization) and physically implemented with Cadence Innovus (floorplanning, power planning, standard-cell placement, clock-tree synthesis, global and detailed routing, and physical verification/signoff), targeting a 5 ns clock period (200 MHz) on the ASAP7 standard-cell placement, clock-tree synthesis, global and detailed routing, and physical verification/signoff), targeting a 5 ns clock period (200 MHz) on the ASAP7 standard-cell library.

Table II summarizes the Cadence Innovus physical implementation results.

Metric	Result
Clock period	5 ns (200 MHz)
Final standard cells	4219
Leaf instances	4227 (3979 comb. / 248 seq.)
Cell area / Total area	15032.27 / 20010.82 units²
WNS / TNS	+0.3 ps / 0 ps
Violating paths	0
Clock-gated flip-flops	234 / 240 (97.5%)
Switching / Internal / Leakage power	58.24% / 41.74% / 0.02%

IV. CONCLUSION

An Adaptive Execution Unit combining a dynamic precision controller, integrated clock gating, operand isolation, and a multi-stage pipeline has been designed in synthesizable Verilog and implemented through a complete industrial RTL-to-GDSII flow using Cadence Genus and Innovus. The design achieves positive timing slack, 97.5% clock-gated flip-flops, zero routing congestion, and full physical-verification compliance, confirming an efficient, scalable, and implementation-aware execution architecture suitable for modern low-power ASIC processors.

REFERENCES

[1] M. A. Raza, I. Shahzad, H. Anwar, M. A. Qureshi, F. H. Malik, and M. U. A. Khan, "An Optimum Design and Implementation of a 16-bit ALU on CADENCE Using RISC-V Architecture," IEEE Int. Conf. on Emerging Trends in Engineering, Sciences and Technology (ICES&T), 2023.

[2] Deepali, I. Saini, and M. Khosla, "Low Power 32-bit Synchronous and Reconfigurable ALU Design using Chain Structure," 11th IEEE Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7, 2020.

[3] S. Sarkar, H. Chatterjee, P. Saha, and M. Biswas, "8-Bit ALU Design using m-GDI Technique," 4th IEEE Int. Conf. on Trends in Electronics and Informatics (ICOEI), pp. 17–22, 2020.

[4] Y. Zhou and H. Guo, "Application Specific Low Power ALU Design," IEEE/IFIP Int. Conf. on Embedded and Ubiquitous Computing (EUC), Shanghai, China, pp. 214–220, 2018.

[5] B. Chatterjee, M. Sachdev, and R. Krishnamurthy, "A CPL-based Dual Supply 32-bit ALU for Sub-180 nm CMOS Technologies," Proc. Int. Symp. on Low Power Electronics and Design (ISLPED), IEEE, Newport Beach, CA, USA, pp. 248–251, 2004.

[6] A. Chauhan, K. K. Saini, N. Rajput, and R. Domah, "Implementation of High Performance 4-Bit ALU using Dual Mode Pass Transistor Logic," Int. Conf. on Intelligent Technologies (CONIT), IEEE, Hubli, India, 2021.

[7] G. Verma, "Design and Analysis of ALU for Low Power IoT Centric Processor Architectures," Global Conf. on Wireless and Optical Technologies (GCWOT), IEEE, Malaga, Spain, 2020.

[8] R. J. Kachhadiya, Y. Agrawal, and R. Parekh, "Implementation of ALU using RTL to GDSII Flow and on NEXYS-4 DDR FPGA Board," 2nd Int. Conf. on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, pp. 481–488, IEEE, 2021.

[9] V. Priyanka Brahmaiah, V. Kumar Gurrala, S. Teja Tuduru, and K. Hari Kishore, "Design of Area and Power-Optimized VLSI Architecture of ALU using Signed Multiplier," Int. Conf. on Recent Trends in Microelectronics, Automation, Computing and Communication Systems (ICMACC), Hyderabad, India, IEEE, 2022.

[10] J. L. V. Ramana Kumari, Y. Varshitha, C. Gopi, and M. Rakesh, "Design and Implementation of ALU Using Ring Counters," Int. Conf. on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, IEEE, 2023.

[11] K. Puli and V. Pudi, "Design of Low Power ALU for RISC-V ISA," IEEE Int. Symp. on Smart Electronic Systems (iSES), Ahmedabad, India, 2023.

Funding Declaration

The authors declare that no funds, grants, or other forms of financial support were received from any organization or institution for the conduct of this research.