Grenze International Journal of Engineering and Technology, June Issue

Design and Optimization of a RISC-V Processor for Modern Applications

Annesha Dey[1], Manprita Sarma[2], Rohit Rajput[3] and Brinda Bhowmick[4]
[1-4]Department of Electronics and Communication Engineering, NIT Silchar, Assam, India
Email: anneshad_ug_22@ece.nits.ac.in, manpritas_ug_22@ece.nits.ac.in, rohitr_ug_22@ece.nits.ac.in, bbhowmick@ece.nits.ac.in

***Abstract*— The design and optimization of a 32-bit in-order RISC-V processor (RV32I ISA) has been presented in this work, targeting improved performance, reduced power consumption, and efficient area and resource utilization for advanced modern applications. RISC-V development in India has progressed at a very fast pace, but most of the implementations have been restricted to single-cycle designs with limited performance optimization. To overcome these limitations, this work includes the progressive design and implementation of RISC-V processors starting from a single-stage configuration and extending through the 2-stage, 3-stage, 4-stage, and finally a complete 5-stage pipelined architecture. At each stage of development, clock constraints were applied to enable systematic power optimization while ensuring maximum utilization of the available hardware resources. The design and functional verification are carried out using Verilog HDL and verified using custom testbenches. Synthesis, timing, and power analyses are done using the Xilinx Vivado Design Suite. The five-stage pipelined processor under consideration combines data forwarding, hazard detection with stall logic, and a dynamic branch prediction mechanism to reduce pipeline hazards and minimize hazards. So far, power reduction has been achieved at every pipeline depth, with significant savings observed in the single-cycle, second, third, fourth, and fifth stages of the RISC-V processor. In addition to these power optimizations, key performance metrics such as CPI (Cycles Per Instruction), maximum clock frequency, dynamic power efficiency (enhanced through clock-gating techniques), and overall area utilization have been systematically evaluated to quantify the effectiveness of the proposed architecture. Mathematical validation through dynamic power modelling, non-ideal CPI formulation, EPI analysis, and an Optimization Efficiency Metric confirmed that deeper pipelining delivers superior energy efficiency despite modest increases in area and control complexity, establishing the proposed architecture as well-suited for next-generation low-power processing platforms. The combined results confirm that the designed 32-bit in-order RV32I processor is well suited for modern high-performance, low-power processing applications across a range of embedded and edge-computing scenarios.**

***Index Terms*— RISC-V, RV32I, Optimization Efficiency Metric (OEM), Power-Performance-Area (PPA), Verilog and Simulation.**

I. INTRODUCTION

Achieving an optimal balance between performance, power efficiency, and area remains a fundamental challenge in modern processor design.

Grenze ID: 01.GIJET.12.2.125_24

A. Background and Motivation

Modern computing demands high-performance and energy-efficient processors, particularly for IoT and embedded systems where power consumption, cost, and customization are critical constraints. Proprietary architectures such as ARM and Intel x86 dominate the market but impose licensing fees and restrict customization, thereby limiting accessibility for academic research and innovation. Reduced Instruction Set Computing – Fifth Generation (RISC-V), an open-source Instruction Set Architecture (ISA) built on the principles of RISC, eliminates these barriers through a free, modular, and extensible design. It allows researchers and designers to build customized, application-specific processors without licensing restrictions. Successful implementations such as IIT Madras's SHAKTI and CDAC's VEGA processors have validated the practicality and efficiency of RISC-V in both academic and industrial domains. This project is motivated by the potential of RISC-V as a platform for developing efficient, scalable, and customizable processors tailored for embedded and IoT workloads.

B. Significance

While RISC-V provides an excellent foundation for open hardware development, many existing academic and open-source 32-bit implementations face notable challenges. Pipeline hazards arising from data and control dependencies often introduce stalls and bubbles, degrading overall performance. Additionally, conventional pipelined designs tend to keep all stages active irrespective of instruction requirements, resulting in unnecessary dynamic power dissipation. Furthermore, many academic cores lack fine-grained control mechanisms for effective hazard resolution and comprehensive power-performance-area (PPA) optimization, limiting their adaptability to modern workloads. As a result, existing 32-bit RISC-V implementations frequently exhibit inefficiencies in power consumption, throughput, and resource utilization, making them less suitable for next-generation embedded systems.

This project addresses these issues through the design and implementation of an optimized five-stage in-order pipelined RISC-V processor (RV32I ISA). The processor integrates advanced hazard detection and resolution mechanisms (forwarding, stalling, and branch prediction) along with power-aware design features such as clock gating and dynamic stage control to achieve measurable improvements in power efficiency, performance, and area utilization.

II. Literature Review

Manjusha Rao P et al. detail the design of a 32-bit single-cycle RISC-V processor in Verilog, synthesizing components like the ALU and register file to execute basic instructions. The design prioritizes simplicity to execute each instruction in one clock cycle. The authors conclude that this straightforward approach provides a clear educational foundation for understanding fundamental processor architecture before moving to more complex designs. Shahriar Ahmed et al. present the design of a five-stage pipelined 32-bit RISC-V core intended to be lightweight, scalable, and easy for beginners to understand. The core was verified on a small FPGA and an online EDA tool, demonstrating highly efficient and low-power operation. The work concludes that this "barebone" architecture is an effective and accessible design for small-scale FPGA applications. Yuqing Li et al. present a five-stage pipelined RISC-V processor design focused on performance optimization. To mitigate common pipeline issues, the architecture incorporates data forwarding to resolve data hazards and a static branch prediction technique to handle control hazards. They conclude that the design achieves correct functionality with low hardware overhead, making it well-suited for high-performance, cost-constrained embedded systems. Anushka Ganguly et al. optimize a RISC-V core for Power, Performance, and Area (PPA) at the physical design level. Using strategies like Low Voltage Threshold (LVT) cells and refined clock constraints, they increased the clock frequency by 50% while reducing die area by 3%. They conclude that such physical optimizations are highly effective for boosting performance but highlight the critical trade-off that often results in higher power consumption.

III. Methodology

A. Tools

Verilog HDL was employed to model and implement all processor modules. Functional verification and debugging were carried out using the Vivado Simulator with custom-designed testbenches. For synthesis and implementation, the Xilinx Vivado Design Suite was used to perform logic synthesis, placement, and routing targeting an FPGA platform. Power and performance evaluation were conducted using the Vivado Power

Analyzer for estimating dynamic and static power based on switching activity, while the Vivado Timing Analyzer was used for critical path identification and slack analysis. Additionally, resource utilization reports were analyzed to monitor the usage of Look-Up Tables (LUTs), Flip-Flops (FFs), and memory resources.

B. Approach

The flowchart in Fig. 1 depicts the entire process followed to implement the processor. The overall methodology was divided into five distinct phases, as described in Fig. 2.

In Phase 1, a single-cycle baseline processor was implemented where all instructions—arithmetic, logical, load/store, and branch—executed in one clock cycle using core modules such as the Program Counter, Instruction Memory, Control Unit, Register File, ALU, and Data Memory, establishing functional correctness and initial power benchmarks. Phase 2 extended this architecture into a five-stage pipelined design (IF, ID, EX, MEM, WB) using pipeline registers (IF/ID, ID/EX, EX/MEM, MEM/WB) to enable parallel instruction execution and improve throughput, albeit with a four-cycle latency due to pipeline filling and draining. Phase 3 addressed pipeline hazards by implementing data forwarding from EX/MEM and MEM/WB stages, inserting stalls for load-use hazards, and mitigating control hazards using a 2-bit dynamic branch predictor (64 entries) with pipeline flushing on misprediction. Phase 4 focused on PPA optimization, incorporating clock gating for dynamic power reduction, analysing switching activity via Vivado Power Analyzer, improving timing through critical path optimization using the Vivado Timing Analyzer, integrating performance counters for CPI and IPC evaluation, and reducing area through efficient LUT and FF utilization. Finally, Phase 5 involved comprehensive hierarchical verification, including unit testing of individual modules, integration testing of data path and control interactions, performance benchmarking for CPI, stall cycles, pipeline utilization, and branch prediction accuracy, along with post-synthesis validation for timing and power, ensuring a high-performance, power-efficient, and FPGA-optimized processor design.

IV. Results and Discussions

Fig. 3 shows the single Cycle 32-bit RISC processor. In this paper, the fifth stage implementation is done. Fig. 4 shows the Five stage pipelined structure.

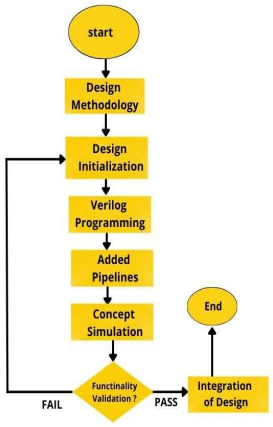

Figure 1. Flowchart of the entire methodology

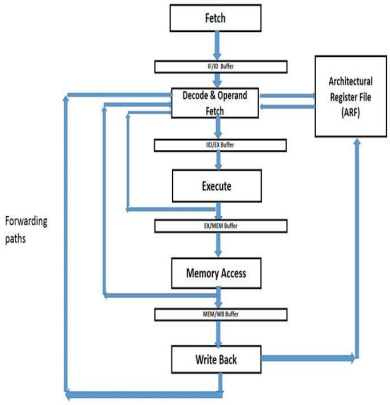

Figure 2. Pipeline architecture flowchart

A. Description

The proposed 32-bit 5-stage RISC-V processor implements a pipelined architecture compliant with the RV32I instruction set, comprising Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory Access (MEM), and Write Back (WB) stages to enable overlapping execution and improved throughput. In the IF stage, the Program Counter generates instruction addresses and retrieves instructions from memory, forwarding them via the IF/ID register. The ID stage decodes instruction fields (opcode, rs1, rs2, rd, funct3, funct7), generates immediates, reads operands from the register file, and produces control signals. The EX-stage performs ALU operations, evaluates branch conditions, and computes target addresses using forwarded or immediate operands. The MEM stage handles load/store operations via data memory, while the WB stage writes results back to the register file, excluding x0. Pipeline registers (IF/ID, ID/EX, EX/MEM, MEM/WB) ensure stage synchronization.

Data hazards are mitigated using forwarding from EX/MEM and MEM/WB stages, while load-use hazards are resolved through stall insertion by a hazard detection unit. Control hazards are addressed using a 2-bit saturating counter-based branch predictor with global history. Supporting multiple instruction formats (R, I, S, B, U, J), the design achieves an efficient balance between performance, power, and area through a modular Verilog implementation suitable for FPGA synthesis and analysis.

B. Simulation Results of 5th Stage Implementation

The simulation results validate the functional correctness of the processor across all pipeline stages and modules. The control unit waveform (Fig. 6) confirms accurate control signal generation for store, load, and R-type instructions based on opcode and funct3 fields, with correct assertion of RegWrite, MemWrite, ALUSrc, MemtoReg, and Branch signals. Store operations enable only memory writes, load instructions ensure memory-to-register transfer, and R-type instructions correctly route ALU results to the register file, while Branch remains inactive in the observed interval. The ALU waveform further verifies correct execution of arithmetic and logical operations, with ALUCtrl selecting functions and the Zero flag asserted appropriately. Pipeline register waveforms (Figs. 7) demonstrate proper latching and propagation of data and control signals across IF/ID, ID/EX, and EX/MEM stages, ensuring synchronization and stable inter-stage communication without unintended stalls or flushes, thereby maintaining one-instruction-per-cycle throughput under normal conditions. The EX/MEM and ID/EX registers correctly buffer execution results, operands, and control signals, validating reliable pipeline operation. For the fifth-stage RISC-V processor, the on-chip power analysis is obtained from the 12 post-implementation reports generated by the Xilinx Vivado Design Suite and shown in Table I. The device on chip static power decreases from 0.236 W before optimization to 0.056 W after optimization, corresponding to a reduction of approximately 76%. Furthermore, the device on-chip dynamic power is reduced dramatically from 31.654 W to 0.003 W, achieving a 99.999% decrease in switching power for the final five-stage pipelined design. Different parameters are compared in Fig. 12 and Fig. 13. Fig. 14 depicts the number of LUTs and number of flip flops utilized in different stages of implementation. As expected, it increases with the number of stages.

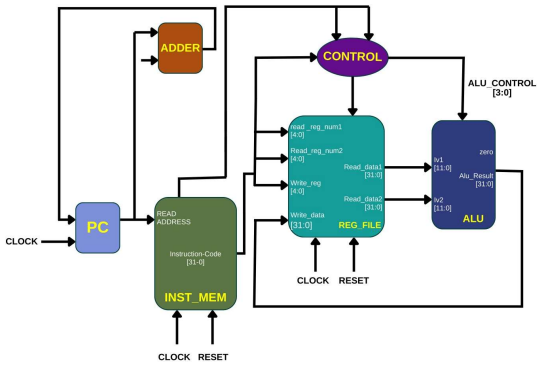

Figure 3. Block Diagram of Single-Cycle 32-bit RISC-V Processor

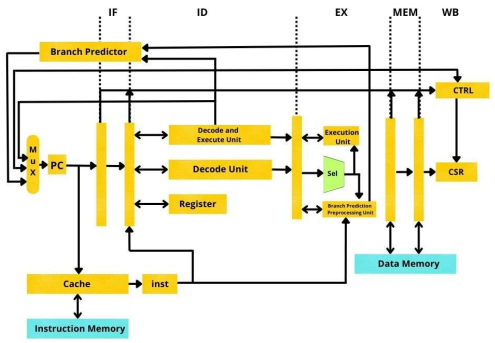

Figure 4. Block Diagram of Five stage 32-bit RISC-V Processor

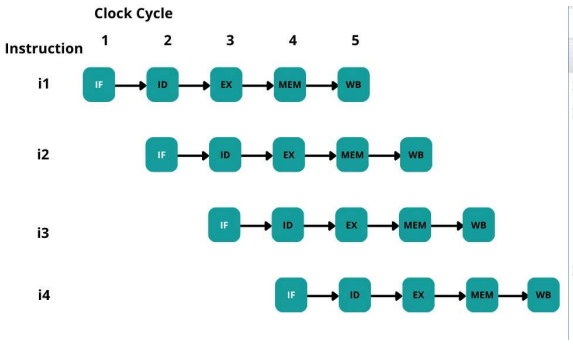

Figure 5. Data Forwarding

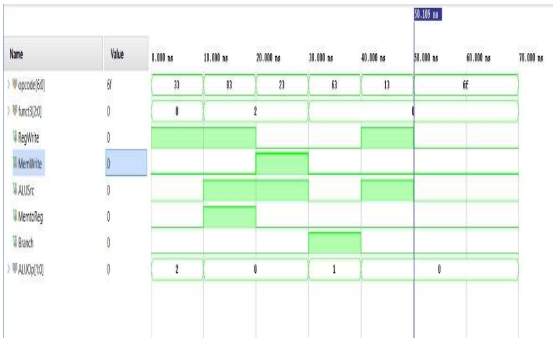

Figure 6. Control Unit Simulation Waveform

Static power, dynamic power, Maximum frequency, number of flip flops, LUTs are plotted in Fig. 15 in star plot. Static power and dynamic power are reducing with the number of stages increased. However, the other parameters are increasing.

C. Mathematical Analysis of 5th Stage Implementation

The general non-ideal CPI formula is:

$$CPI = 1 + CPI_{datastall} + CPI_{controlstall}. \quad (1)$$

$$CPI = 1 + (f_{load} \cdot p_{loaduse} \cdot S_{load}) + (f_{bran} \cdot p_{mispredict} \cdot B_{penalty}). \quad (2)$$

Parameter Definitions:
f_{load} = fraction of instructions that are loads (LW) ≈ 0.20
$p_{loaduse}$ = probability of a load followed immediately by a dependent instruction ≈ 0.40
S_{load} = stall cycles per load-use hazard = 1 cycle
f_{branch} = fraction of instructions that are branches ≈ 0.15
$p_{mispredict}$ = branch misprediction rate of 2-bit predictor ≈ 0.10
$B_{penalty}$ = pipeline flush penalty for misprediction =2 cycles (for 5-stage pipeline)

TABLE I: ON-CHIP POWER ANALYSIS

Power Type	Before Optimization W	After Optimization W	Change %
Device On-Chip Static Power	0.236	0.056	76
Device On-Chip Dynamic Power	31.654	0.003	99.99

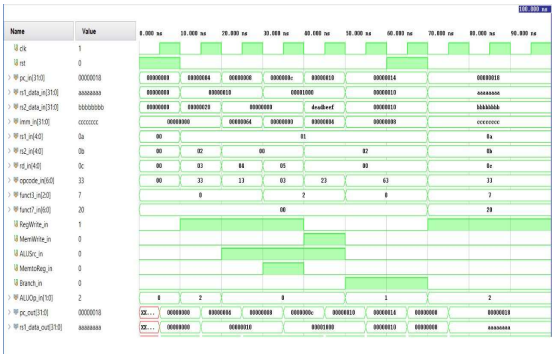

Figure 7. Output Simulation Waveform (Part 1)

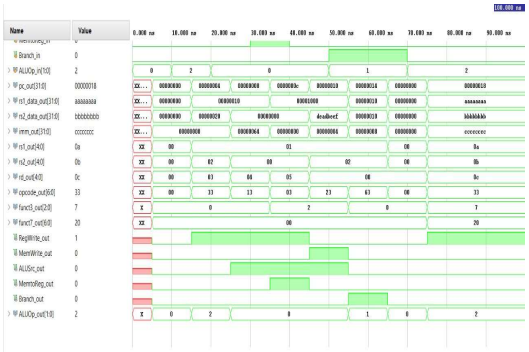

Figure 8. Output Simulation Waveform (Part 2)

$$CPI_{data} = f_{load} \cdot p_{loaduse} \cdot S_{load}. \quad (3)$$

$$= 0.20 \cdot 0.40 \cdot 1 . \quad (4)$$

$$= 0.080 . \quad (5)$$

$$CPI_{ctrl} = f_{branch} \cdot p_{mispredict} \cdot B_{penalty}. \quad (6)$$

$$= 0.15 \cdot 0.10 \cdot 2 . \quad (7)$$

$$= 0.030 . \quad (8)$$

$$CPI_{total} = 1 + 0.080 + 0.030. \quad (9)$$

$$= 1.110 . \quad (10)$$

This represents a 10% degradation from the ideal CPI of 1.0. The 2-bit dynamic branch predictor (64-entry BHT) reduces $p_{mispredict}$ to approximately 10%, which is why the branch penalty contributes only 0.030 to the total CPI despite $B_{penalty}$ = 2 flush cycles.

A single scalar metric is required to compare the overall effectiveness of the optimization across all five pipeline designs, accounting for power savings, area overhead, and frequency of performance simultaneously. We define the Optimization Efficiency Metric (OEM) as:

$$OEM = \left(R_{dyn} R_{static} f_{norm}\right) / \left(A_{norm} CPI_{norm}\right). \quad (11)$$

Where:
R_{dyn} = Dynamic power reduction ratio (0–1)
R_{static} = Static power reduction ratio (0–1)

Figure 11. Power variation in each stage without constraints

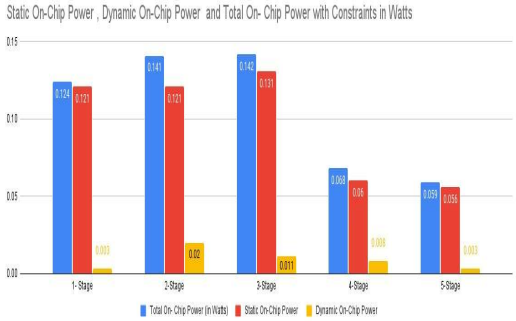

Figure 12. Power variation in each stage with constraints

$$f_{norm} = \frac{f_{stage}}{f_{max,\,all\,stages}}. \quad (12)$$

$$A_{norm} = \frac{LUT_{stage}}{LU_{\;max,\,all\,stages}}. \quad (13)$$

$$CPI_{norm} = \frac{CPI_{stage}}{CPI_{min,\,all\,stages}}. \quad (14)$$

A higher OEM indicates a design that achieves greater power savings at a given operating frequency, with lower area overhead and minimal CPI penalty—representing a genuinely better-optimized design.
The OEM ranking highlights the trade-off between performance, power, and area across pipeline configurations as shown in Table II. The single-cycle design (Stage 1) achieves a high OEM due to minimal area (13 LUTs) and ideal CPI = 1 but suffers from low operating frequency and limited practicality. Stage 2 provides a balanced design with significant dynamic power reduction (99.9%) and moderate area overhead (133 LUTs), resulting in a strong OEM score. The five-stage pipeline (Stage 5), although penalized in OEM due to higher area (806 LUTs) and CPI = 1.11, delivers the highest dynamic power savings (99.999%) and remains the most suitable architecture for real-world, high-performance applications.

V. CONCLUSIONS

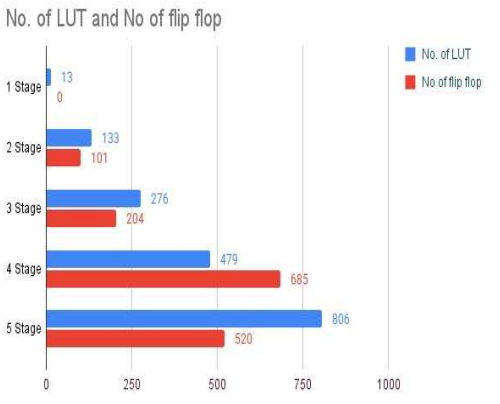

Figure 13. Variation in number of LUTs and FFs used in each stage

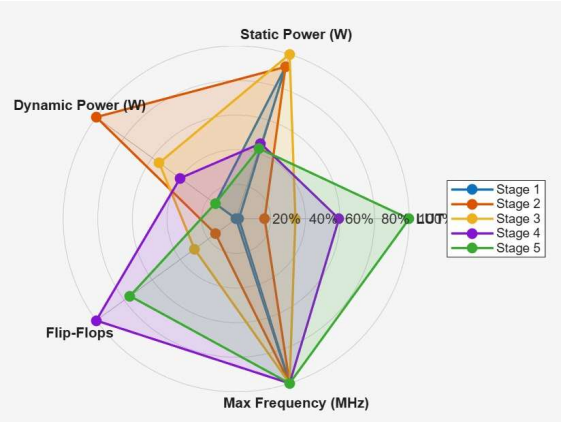

Figure 14. Radar-or-Star plot showing variation in 5 parameters

The work demonstrates the progressive design and optimization of a 32-bit in-order RISC-V (RV32I) processor, evolving from a single-cycle architecture to a fully pipelined five-stage design. This transition significantly improves performance metrics such as CPI and operating frequency while reducing power consumption through clock-gating and constraint-driven synthesis. The final architecture integrates data forwarding, hazard detection, and dynamic branch prediction to efficiently mitigate pipeline hazards. Although deeper pipelining introduces additional control complexity and area overhead, the overall power–performance–area (PPA) trade-off remains favorable. Analytical validation using CPI modeling, and optimization metrics confirms that the proposed design achieves superior energy efficiency, making it well-suited for modern embedded and edge-computing applications.

TABLE II: OEM CALCULATION

Stage	R_{dyn}	R_{static}	f_{norm}	A_{norm}	CPI_{norm}	OEM	Rank
1st	0.999	0.129	0.833	0.016	1.000	6.681	#1
2nd	0.999	0.410	1.000	0.165	1.095	2.265	#2
3rd	1.000	0.763	0.714	1.000	1.110	0.491	#4
4th	0.999	0.400	0.714	0.594	1.110	0.433	#5
5th	0.999	0.030	1.000	0.342	1.110	0.078	#3

REFERENCES

[1] Y. Li, Z. Yu, and W. Ji, “Design and Performance Optimization of a RISC-V Processor with Five-Stage Pipeline,” 2025 IEEE 6th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), pp. 1475–1481, 2025.

[2] S. Ahmed and A. B. M. Harun-Ur-Rashid, “Design, Implementation and Verification of Five Stage Pipeline RISC-V Core (RV32I ISA),” 2025 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–7, 2025.

[3] M. R. P., P. Niranjan, and D. K. M. J, “Design and Implementation of 32-bit RISC-V Processor Using Verilog,” 2024 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 181–188, 2024.

[4] A. Ganguly, A. Chakraborty, A. A. Ambekar, and H. K. Mondal, “Power, Performance, and Area Optimization of the RISC-V Processor,” 2024. IEEE International Symposium on Smart Electronic Systems (iSES), pp. 271–278, 2024.

[5] Ricardo Núñez-Prieto, David Castells-Rufas and LluísTerés-Terés, ”RisCO2: Implementation and Performance Evaluation of RISC-V Processors for Low-Power CO2 Concentration Sensing,” Microcontrollers and Microprocessors: The Advanced System on the Chip, Micromachines 2023, 14(7), 1371.

[6] M. Själander "Optimizing Energy Efficiency in Subthreshold RISC-V Cores" arXiv preprint arXiv:2502.06588, Feb. 2025. doi: 10.48550/arXiv.2502.06588.