Low Power Design of 5-Stage Pipelined RISC-V Processor

P. Chakradhar
Dept. of Electronics and Communication Engineering
Sri Venkateswara College of Engineering
Tirupati, India
chakradhar2205@gmail.com

C. Chandrasekhar
Dept. of Electronics and Communication Engineering
Sri Venkateswara College of Engineering
Tirupati, India
dr.chandrasekhar.c@svcolleges.edu.in

P. Sindhu Priya
Dept. of Electronics and Communication Engineering
Sri Venkateswara College of Engineering
Tirupati, India
sindhupriya.pamisetty@gmail.com

N. Sai Venkat
Dept. of Electronics and Communication Engineering
Sri Venkateswara College of Engineering
Tirupati, India
naraharisaivenkat@gmail.com

T. Sharon
Dept. of Electronics and Communication Engineering
Sri Venkateswara College of Engineering
Tirupati, India
angelinathiruttani27@gmail.com

S. Maheswari
Dept. of Electronics and Communication Engineering
Sri Venkateswara College of Engineering
Tirupati, India
maheswari4931@gmail.com

***Abstract*—This work presents the design and realization of a low-power, five-stage pipelined RISC-V processor described using Verilog hardware description language. The processor follows a classic pipeline organization consisting of instruction fetch, instruction decode, execution, memory access, and write-back stages. To reduce dynamic power consumption caused by unnecessary signal transitions, a clock-gating strategy is incorporated at the pipeline register level. The clock enable signals are controlled by hazard detection and pipeline control logic, allowing the clock to be disabled during stall conditions and inactive cycles. This approach ensures correct functional behavior while preventing redundant switching activity. The proposed architecture was synthesized and analyzed using industry-standard synthesis and power estimation tools. Experimental evaluation indicates a noticeable reduction in dynamic power consumption compared to an equivalent ungated pipelined design, without altering the processor's architectural features or performance. The proposed solution offers a scalable and energy-efficient approach suitable for RISC-V–based embedded processing applications.**

Keywords—RISC-V processor, pipelined architecture, low-power design, clock gating, dynamic power reduction, Verilog HDL.

I. INTRODUCTION

Reduced Instruction Set Computer (RISC) architectures have gained significant importance in modern processor design due to their simplicity, efficiency, and scalability. Among these architectures, RISC-V has emerged as a prominent open-source instruction set architecture that allows designers to customize processors according to application-specific requirements without proprietary limitations [1]. Its modular and extensible design enables efficient implementations, particularly in embedded and energy-constrained systems where performance-per-watt is a key design objective [2]. Instruction pipelining is a widely adopted technique used to improve processor throughput by overlapping the execution of multiple instructions. In a typical five-stage pipelined processor, instruction execution is divided into sequential stages: instruction fetch, decode and operand preparation, execution, memory access, and write-back. This staged organization provides an effective balance between performance improvement and implementation complexity [2], [12]. However, despite increasing instruction-level parallelism, pipelined architectures often experience higher dynamic power consumption due to continuous clock transitions and unnecessary switching activity within pipeline registers [4], [6]. Modern processors rely heavily on pipelining to enhance execution efficiency by processing multiple instructions simultaneously across different stages. Although this improves throughput, the resulting switching activity significantly contributes to dynamic power dissipation, which becomes a major concern in low-power and embedded systems [6], [7]. Reducing unnecessary clock activity is therefore essential for improving the energy efficiency of pipelined processor architectures. Clock gating has emerged as an effective low-power design technique that minimizes redundant switching by disabling the clock signal to inactive pipeline stages or registers without affecting functional correctness [4], [5]. The integration of clock gating with pipeline control and hazard detection mechanisms is critical to ensure proper timing and reliable processor operation while reducing power consumption [3], [5]. This project focuses on the design and implementation of a low-power five-stage pipelined RISC-V processor with pipeline-level clock gating. The proposed approach reduces dynamic power consumption by disabling clock transitions during stall and idle conditions while maintaining the architectural behavior of the processor. The design is implemented using Verilog HDL and evaluated through synthesis and power analysis tools [9], [14], [15]. The results demonstrate that pipeline-level clock gating provides a practical and scalable solution for enhancing energy efficiency in RISC-V based processors intended for embedded and FPGA-based applications [3], [10].

II. RELATED WORKS

Several researchers have contributed to the development of low-power processor architectures, RISC-V implementations and clock-gating techniques. Waterman et al. introduced the RISC-V instruction set architecture as an open and extensible platform that enables customized and efficient processor design for embedded applications [1]. Hennessy and Patterson highlighted the importance of pipelining in improving processor performance while emphasizing the associated power and design trade-offs [2]. Recent work by Kocaman and Kandemir demonstrated that fine-grain clock gating can significantly reduce switching activity and dynamic power consumption in RISC-V processors [3], while Zhang et al. provided a comprehensive review of clock-gating techniques for low-power VLSI design [4]. Jain and Gupta further showed that integrating clock gating with pipeline control logic improves energy efficiency without affecting processor functionality [5]. A survey by Mittal

emphasized the growing importance of energy-efficient architectures in embedded systems [6], and Alioto discussed the role of energy-efficient digital circuits in modern low-power processor design [7]. Earlier work by De and Borkar addressed the fundamental technology challenges of low-power and low-voltage integrated circuits [8]. Schaumont provided practical methodologies for FPGA-based system design and implementation [9]. Several studies, including works by Zhang et al., Thanga Dharsni et al., and Phangestu et al., demonstrated the implementation and verification of pipelined RISC-V processors with improved hazard handling and instruction throughput [10]–[12]. Foundational VLSI design principles by Weste and Harris continue to support power-aware processor development [13]. Finally, FPGA vendor documentation from Xilinx Inc. provides essential guidance for synthesis and power analysis used in evaluating processor implementations [14], [15]. Collectively, these studies emphasize the importance of combining pipelining with clock-gating techniques to achieve energy-efficient RISC-V processor architectures, motivating the proposed low-power pipelined RISC-V processor with pipeline-level clock gating.

III. Baseline 5-Stage RISC V Architecture

The processor architecture taken as a baseline in this research work is a traditional five-stage pipelined RISC-V processor that supports the RV32I instruction set. The pipeline stages are broken down into five stages, namely Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory Access (MEM), and Write Back (WB). Pipeline registers are inserted between the stages to support the concurrent execution of multiple instructions and to enhance the instruction throughput. Fig. 1 shows the block diagram of the five-stage RISC-V processor.

Fig. 1 Baseline 5-stage RISC - V Processor Architecture.

A) Instruction Fetch (IF) Stage

The Instruction Fetch (IF) stage is responsible for retrieving the next instruction from memory so it can be processed by the pipeline. During this stage, the Program Counter (PC) holds the memory address of the instruction to be fetched. The instruction memory is accessed using the PC value, and the corresponding instruction is read. At the same time, the PC is updated to point to the address of the next sequential instruction, typically by adding four, since most instructions are 32 bits (4 bytes) long. Both the fetched instruction and the updated PC value (often referred to as PC+4) are then stored in the IF/ID pipeline register.

B) Instruction Decode (ID) Stage

In the Instruction Decode stage, the fetched instruction is analyzed to identify its operation type and required operands. The register file is read to retrieve operand values specified by the instruction fields. Simultaneously, the control unit generates the necessary control signals for later pipeline stages. Immediate values are extracted and sign-extended according to the instruction format. All decoded data and control signals are stored in the ID/EX pipeline register.

C) Execute (EX) Stage

The Execute stage performs the core computations of the instruction. A selection mechanism determines whether the ALU receives input values from registers or from immediate operands. The ALU then executes arithmetic, logical, or address-calculation operations. For branch instructions, comparison logic evaluates branch conditions and determines whether a change in program flow is required. The results are forwarded to the EX/MEM pipeline register.

D) Memory Access (MEM) Stage

The Memory Access stage handles operations that interact with data memory. Load and store instructions access memory using the address computed in the Execute stage. Instructions that do not require memory access pass through this stage without modifying memory contents. Relevant outputs are saved in the MEM/WB pipeline register for final processing.

E) Write Back (WB) Stage

The Write Back stage completes the instruction's execution by updating the register file. Depending on the instruction type, either the result produced by the ALU or the data retrieved from memory is selected. This value is then written to the destination register, concluding the instruction's execution cycle. The baseline processor follows a classical five-stage pipelined RISC-V architecture designed to improve instruction throughput through parallel execution of instructions across different stages. Pipeline registers are inserted between stages to isolate combinational logic and enable concurrent operation. Functionally, the processor executes instructions in an overlapped manner, with each pipeline stage performing a specific task during every clock cycle. Instructions are fetched, decoded, executed, and completed in successive stages while new instructions enter the pipeline simultaneously. Control and data dependencies are managed through pipeline control mechanisms to maintain correctness. Upon completion, results are written back to the register file, ensuring architectural state consistency. This baseline functional operation serves as the foundation for incorporating low-power enhancements in the proposed design.

IV. Methodology

This section describes the design methodology and architectural organization of the proposed low-power RISC-V processor. The processor is implemented using a modular register-transfer-level (RTL) approach in Verilog HDL, where each pipeline function is realized as an independent functional unit. A five-stage pipelined microarchitecture is adopted, and low-power operation is achieved by applying clock-gating control at the top module level. The methodology focuses on reducing unnecessary switching activity in the program counter and pipeline registers while preserving functional correctness and pipeline performance. The overall organization of the proposed design is shown in Fig. 2.

A) Top-Level Low-Power Organization

At the top level, the processor integrates the Instruction Fetch Unit (IFU), Instruction Memory Unit (IMU), Register File Unit (RFU), Control Unit (CU), Arithmetic Logic Unit (ALU), and inter-stage pipeline registers. All clock-enable signals required for low-power operation are generated centrally within the top module. These enable signals control the updating of the program counter and the pipeline registers, ensuring that data is propagated only when valid instructions are present. This centralized clock-gating strategy avoids redundant transitions across the pipeline and forms the primary mechanism for dynamic power reduction.

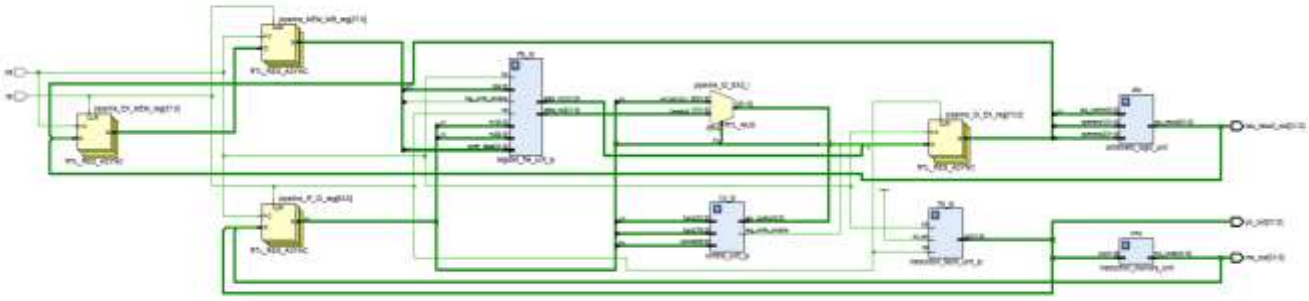

Fig. 2 Top RTL Schematic of the Processor

B) Instruction Fetch and Instruction Memory Units

The Instruction Fetch Unit (IFU) is responsible for maintaining and updating the Program Counter (PC). Unlike conventional designs where the PC updates on every clock edge, the IFU updates the PC only when a clock-enable signal is asserted. This prevents unnecessary toggling during idle or inactive instruction cycles. The Instruction Memory Unit (IMU) provides the instruction corresponding to the current PC value. The fetched instruction and PC value are captured in the IF/ID pipeline register only when instruction validity is asserted, thereby minimizing switching activity in the fetch stage.

C) Decode, Register File, and Control Units

Instruction decoding is performed by extracting the opcode, function fields, and register indices from the IF/ID pipeline register. The Register File Unit (RFU) supplies the source operands required for execution and supports synchronous register write-back from the final pipeline stage. The Control Unit generates the ALU control signals and register write-enable signal based on the decoded instruction fields. By separating control generation from datapath execution, the design maintains clarity and allows pipeline register updates to be selectively enabled under clock-gating control. Additionally, immediate values are sign-extended during this stage to ensure correct operand sizing for arithmetic and memory operations via the ID/EX register for further processing.

D) Execute Stage and Datapath Operation

The Execute Stage performs arithmetic and logical operations using a combinational Arithmetic Logic Unit (ALU). The ALU supports core RV32I operations such as addition, subtraction, logical operations, and set-less-than comparisons. Operand selection logic allows either register operands or immediate values to be forwarded to the ALU, depending on the instruction type. Since the ALU itself is purely combinational, power optimization is achieved by preventing modifying the ALU structure.

E) Pipeline Registers and Clock-Gating Strategy

Low-power operation is primarily realized through clock gating applied to all pipeline registers, including IF/ID, ID/EX, EX/MEM, and MEM/WB stages. An instruction-valid signal is derived based on the availability of meaningful instruction data from the instruction memory. When no valid instruction is detected, the clock-enable signals for the program counter and pipeline registers are deasserted, effectively freezing pipeline activity. This approach eliminates redundant switching transitions while maintaining correct pipeline sequencing and architectural behavior. The results produced in the execute stage are propagated through the pipeline registers under clock-enable control. Final results are committed to the register file directly from the MEM/WB stage when the register write-enable signal is asserted. By gating the MEM/WB pipeline register, unnecessary register file activity is avoided during idle instruction cycles, further contributing to dynamic power reduction.

V. Result Analysis

The experimental results presented in this section evaluate the functional correctness, power efficiency, resource utilization, and timing performance of the proposed low-power RISC-V processor. The design is analyzed under identical operating conditions and compared against a baseline implementation without clock gating. The results highlight the effectiveness of the proposed clock-gating strategy in reducing dynamic power consumption while preserving architectural behavior and performance.

A) Functional Verification

Functional verification of the proposed low-power RISC-V processor was performed through RTL simulation using Vivado 2018.3. The same testbench was used for both the baseline and clock-gated designs, with minor modifications limited to display statements for improved signal visibility. The simulation focused on validating correct instruction execution, program counter progression, and arithmetic logic unit operation under clock-gated conditions. The simulation waveform demonstrates correct functional behavior across multiple clock cycles. When valid instructions are present, the program counter increments as expected and instructions propagate correctly through the pipeline stages, producing correct ALU results. During inactive instruction cycles, the program counter and pipeline registers retain their previous values, indicating effective clock-enable control. This state-holding behavior confirms that clock gating suppresses unnecessary switching activity without affecting functional correctness or pipeline operation.

B) Power Consumption Analysis

Power consumption of the proposed clock-gated RISC-V processor was evaluated using post-implementation analysis in the Vivado Design Suite. The analysis focuses on understanding the power characteristics of the design when clock gating is applied to control activity in the program counter and pipeline registers. The summarized power metrics are presented in Table I.

TABLE I. POWER REPORT

Power Component	Power (Watts)
Total On-Chip Power	3.181
Dynamic Power	3.102
Logic Power	1.258
Signal Power	1.823
Static Power	0.079

The results indicate that dynamic power constitutes the major portion of the total on-chip power consumption, which is primarily influenced by switching activity in logic and signal interconnects. By selectively disabling clock transitions during inactive instruction cycles, the proposed clock-gating strategy effectively suppresses unnecessary switching within the pipeline. Static power remains relatively constant, as it is largely dependent on the underlying device characteristics rather than clock activity. These observations demonstrate that clock-aware pipeline control contributes to improved power efficiency while maintaining correct processor operation.

Parameter	Baseline Design	Clock-Gated Design
Total On-Chip Power	3.867	3.181

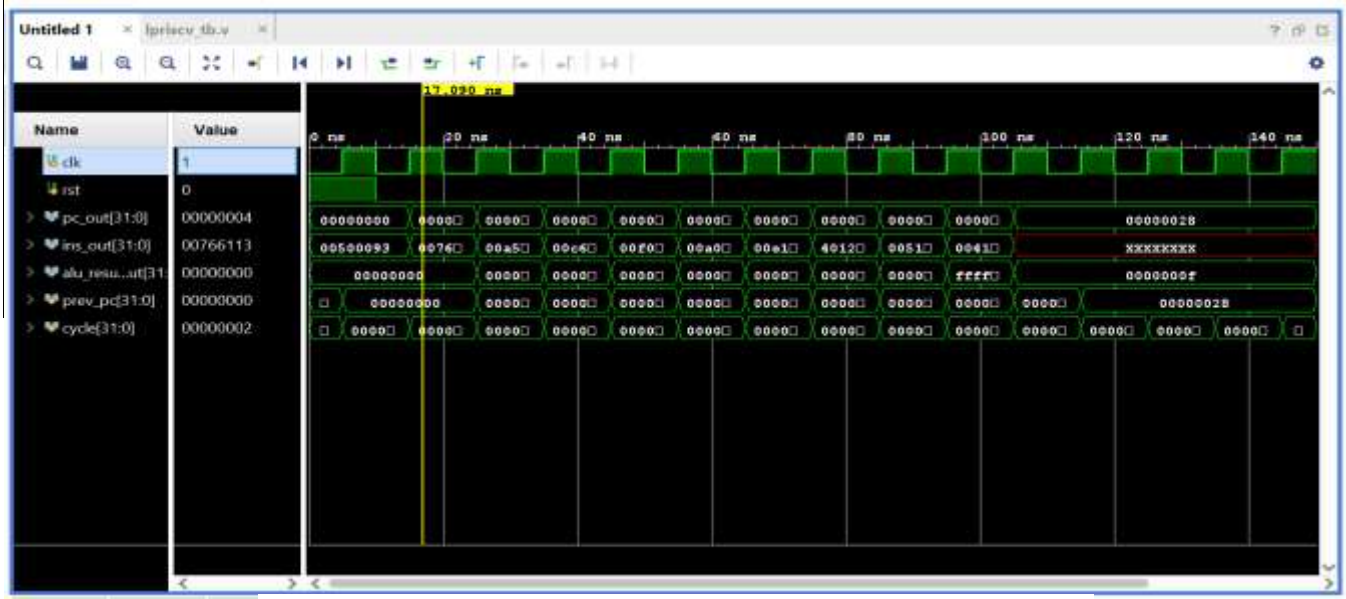

Fig. 3 Top RTL Schematic of the Processor

C) Resource Utilization Analysis

Resource utilization of the proposed design was analyzed to evaluate the area impact introduced by clock-gating logic. The utilization metrics, including slice lookup tables, slice registers, and other relevant resources, are summarized in Table II. These values provide insight into the hardware overhead associated with the low-power enhancements.

TABLE II. UTILIZATION REPORT

Resource	Utilization
Slice LUTs (out of 32,600)	449
Slice Registers (out of 65,200)	669
BUFGCTRL (out of 32)	1
Slices Used (out of 8,150)	376

The results show that the introduction of clock gating does not lead to a significant increase in resource usage. The number of slice LUTs and registers utilized remains minimal relative to the total available resources on the target device. This indicates that the additional control logic required for clock gating incurs negligible area overhead. Overall, the resource utilization analysis confirms that the proposed low-power technique achieves power savings without compromising area efficiency.

D) Results Comparison and Discussion

This subsection presents a comparative discussion of the baseline and clock-gated RISC-V processor implementations to highlight the overall impact of the proposed low-power technique. The comparison focuses on power consumption, resource utilization, and timing behavior under identical operating conditions. Quantitative results are analyzed to demonstrate the effectiveness of clock gating in reducing dynamic power without compromising functional correctness. Additionally, trade-offs between power savings and implementation complexity are discussed to provide a balanced evaluation.

TABLE III. COMPARISION TABLE

The results indicate that the introduction of clock gating leads to a significant reduction in power consumption. The clock-gated design achieves an overall reduction of approximately 17.7% in total on-chip power compared to the baseline implementation. This improvement is primarily attributed to a reduction of about 18.0% in dynamic power, resulting from suppressed switching activity in the program counter and pipeline registers during inactive instruction cycles. Importantly, this power reduction is achieved without any increase in resource utilization or degradation in timing performance. Overall, the comparison demonstrates that the proposed clock-gating strategy provides an effective trade-off between power efficiency and hardware cost. By reducing dynamic power while maintaining functional correctness and architectural behavior, the proposed design offers a practical solution for energy-efficient RISC-V processor implementations.

VI. Conclusion and Future Scope

This paper presented the design and evaluation of a low-power five-stage pipelined RISC-V processor employing clock gating at the top module level. The proposed approach focuses on reducing unnecessary switching activity in the program counter and pipeline registers without modifying the core data path or instruction execution behavior. Functional verification confirmed correct pipelined operation under clock-gated conditions, demonstrating that the introduced power optimization is transparent to architectural functionality. Experimental results show that the clock-gated design achieves a significant reduction in total and dynamic power consumption compared to the baseline implementation, while maintaining identical resource utilization and timing characteristics. The absence of area overhead highlights the efficiency of the proposed technique. Overall, the results demonstrate that pipeline-level clock gating is an effective and practical solution for improving power efficiency in RISC-V processor implementations. The proposed work can be extended to further improve power efficiency by incorporating fine-grain clock gating at sub-module levels and integrating additional dynamic power management techniques such as operand isolation or adaptive clock control. Future enhancements may include support for extended RISC-V instruction sets and evaluation under more complex workloads. In addition, analyzing power behavior across multiple operating conditions and adapting the design for ASIC implementation can provide deeper insight into scalability and practical deployment.

References

[1] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanović, *"The RISC-V Instruction Set Manual, Volume I: User-Level ISA,"* RISC-V Foundation, 2017.

[2] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann, 2019.

[3] Y. Kocaman and M. Kandemir, "Low-power RISC-V processor design using fine-grain clock gating," *IEEE Access*, vol. 10, pp. 30112–30124, 2022.

[4] Z. Zhang, W. Shi, and F. Lombardi, "Clock gating for low-power VLSI design: A comparative review," *IEEE Transactions on VLSI Systems*, vol. 29, no. 9, pp. 1613–1626, 2021.

[5] A. K. Jain and R. Gupta, "Clock-gated pipeline architectures for power-optimized processors," in *Proc. IEEE ISLPED*, 2020, pp. 1–6.

[6] S. Mittal, "A survey of techniques for improving energy efficiency in embedded processors," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–39, 2017.

[7] M. Alioto, "Energy-efficient digital circuits for embedded systems," *IEEE Transactions on Circuits and Systems I*, vol. 67, no. 5, pp. 1405–1418, 2020.

[8] V. De and S. Borkar, "Technology and design challenges for low-power and low-voltage ICs," *IEEE Transactions on VLSI Systems*, vol. 17, no. 6, pp. 697–710, 2009.

[9] P. D. Schaumont, *A Practical Guide to Designing FPGA-Based Systems*, Springer, 2021.

[10] W. Zhang, Y. Zhang, and K. Zhao, "Design and verification of a three-stage pipeline CPU based on RISC-V architecture," in *Proc. ACAIT*, 2021, pp. 697–703.

[11] I. Thanga Dharsni, K. S. Pande, and M. K. Panda, "Optimized hazard-free pipelined architecture for RV32I RISC-V processor," in *Proc. IEEE ICOSEC*, 2022, pp. 739–746.

[12] A. E. Phangestu, I. T. Mujiono, and S. A. Zaini, "Five-stage pipelined 32-bit RISC-V base integer instruction set architecture soft processor core," in *Proc. ISITIA*, 2022, pp. 304–309.

[13] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed., Pearson, 2011.

[14] Xilinx Inc., *Vivado Design Suite User Guide: Power Analysis (UG907)*, 2018.

[15] Xilinx Inc., *Vivado Design Suite User Guide: Synthesis (UG901)*, 2018.