Custom Full Design of 64-BIT RISC Processor Using 5 Stage Pipelining

Gangadhar K G
MTech in VLSI Design and Embedded Systems
Department of Electronics and Communication Engg.
Bangalore Institute of Technology,
Bangalore, India
gangadharkg4141@gmail.com

Dr. Narendra C P
Professor & PG Co-Ordinator
Dept. of Electronics and Communication Engg.
Bangalore Institute of Technology,
narendracp@bit-bangalore.edu.in

Abstract- Growing reliance on battery-powered embedded and IoT platforms has made power-aware processor design a fundamental engineering concern rather than merely a desirable attribute. This work details the design and physical realization of a 64-bit scalar RISC processor architecture aimed at simultaneously maximizing instruction throughput and minimizing on-chip power overhead. Instruction execution is organized across five synchronous pipeline stages—Fetch, Decode, Execute, Memory access, and Write-Back—so that each stage operates independently on successive instructions without stalling the overall flow. The central challenge of reducing unnecessary switching activity across the wide 64-bit data buses is tackled through the deliberate placement of Integrated Clock Gating (ICG) cells within the clock distribution tree. When the instruction decoder determines that a functional unit—such as the 64-bit Barrel Shifter Rotator or the 64-operation ALU—is not required for the current instruction, the corresponding ICG cell isolates that unit from the clock, suppressing wasteful internal node transitions. The entire design was captured in Verilog HDL and taken through ASIC synthesis on the Cadence Genus platform targeting the GPDK 90nm CMOS technology node. Post-synthesis analysis confirms clean timing closure at 100 MHz with a compact gate count, and comparative evaluation shows the resulting silicon footprint to be substantially smaller than that of vector-oriented processor designs occupying the same application space.

Index Terms— 64-BIT RISC, 5-stage Pipeline, Integrated Clock Gating (ICG), ASIC synthesis, Verilog HDL.

I. INTRODUCTION

Modern embedded platforms—ranging from wearable health monitors to industrial sensor nodes—impose tight constraints on both energy consumption and silicon area, forcing processor architects to seek solutions that do not sacrifice computational capability in the pursuit of efficiency. Meeting this twin requirement motivated the present work, which proposes a 64-bit RISC processor organized around a five-stage in-order pipeline. By breaking instruction execution into discrete, overlapping stages, the pipeline sustains a throughput approaching one instruction per clock cycle while keeping the logic depth within each stage manageable enough to support a 100 MHz operating frequency.

A persistent difficulty with 64-bit data paths is that the wide internal buses connecting the register file, execution units, and memory interface toggle frequently even when the associated functional blocks are idle, contributing disproportionately to dynamic power dissipation. Integrated Clock Gating addresses

this problem at its source: rather than relying on the synthesis tool to infer gating from RTL constructs, explicit ICG cells are inserted at strategic points so that each major sub-block—including the Barrel Shifter Rotator and the ALU—receives a locally gated clock signal that is withheld whenever the decoder signals an idle condition.

The design deliberately avoids the wide vector lanes and large SRAM arrays characteristic of high-throughput vector processors. That scalar orientation keeps the standard-cell density high and the routing complexity low, two factors that translate directly into area and timing benefits during ASIC implementation. Synthesis was carried out with the Cadence Genus tool targeting the GPDK 90nm process, and the outcome—compact cell count, zero worst-negative-slack margin, and sub-4 mW on-chip power—validates the suitability of the architecture for power-constrained embedded deployments.

II. LITERATURE REVIEW

Research into power-efficient processor microarchitecture has produced a diverse range of techniques, many of which informed the design choices made in this work. One well-established strand of work focuses on forwarding paths within the register file and ALU to eliminate pipeline stalls caused by data hazards; eliminating stalls not only improves throughput but also reduces the number of redundant clock cycles during which logic toggles without performing useful computation. Clock gating has received particular attention as an RTL-level optimization, with published implementations demonstrating reductions in total dynamic power on the order of 6–12% through selective disabling of idle circuit blocks, although care must be taken to avoid introducing clock skew when gating is applied aggressively.

Investigations into RISC-V-based processor designs have provided useful data points on the power-performance trade-off achievable with modern ISA frameworks. Studies of 32-bit pipelined MIPS RISC cores indicate that power gating can yield meaningful savings, though the interaction between gating logic

and hazard detection circuitry requires careful management to prevent inadvertent state corruption. Work on fault-tolerant instruction decoders for dual-core soft-processor implementations demonstrates that adding error-correctio hardware improves reliability at the cost of increased latency and supplementary power draw—a reminder that robustness enhancements carry area and power penalties.

The Yun 64-bit RISC-V vector processor serves as a relevant comparison point for wide-data-path design: its architecture shows that software-level vectorization can offset the instruction-fetch overhead associated with short SIMD loops, but the associated SRAM arrays and lane-control logic impose a substantial area premium. Studies of simpler 16-bit and 32-bit RISC pipelines confirm that pipelined execution reliably reduces dynamic power relative to multicycle implementations, yet the majority of those designs target narrow application niches rather than general-purpose embedded workloads. Hardware multiplier structures such as the Wallace tree offer throughput advantages but are known to introduce routing congestion and capacitive coupling noise when scaled to 64-bit operand widths, a consideration that reinforces the decision to handle multiplication through the ALU rather than a dedicated array multiplier in the present design.

III METHODOLOGY

The design flow followed a structured ASIC methodology intended to bring the competing objectives of high instruction throughput and low power dissipation under simultaneous control. Micro-architectural decisions were first captured at Register Transfer Level using Verilog HDL, with each pipeline stage modeled as a self-contained module communicating through registered inter-stage buses. A hierarchical control structure governs the handshaking between the 64-entry General-Purpose Register file and the execution units, ensuring that operand transfers occur exclusively during cycles in which the downstream unit is both enabled and ready to accept data.

Curtailing dynamic power loss was approached through fine-grained clock control rather than coarse power-domain switching, which can introduce unacceptable wake-up latency in real-time applications. ICG cells were placed along critical fan-out branches of the clock tree so that the Barrel Shifter Rotator and the ALU each receive an independently gated clock. Enable conditions for each gate are derived directly from the decoded instruction type: when the decoder asserts that the current instruction does not require a particular unit, the gate closes before the next rising edge, preventing internal nodes within that unit from toggling and thereby eliminating the associated capacitive switching energy.

Synthesis was performed with Cadence Genus targeting the GPDK 90nm standard-cell library. The tool was constrained to a 10 ns clock period, and multi-corner timing analysis was applied across the five pipeline stages to verify that no path violated setup or hold margins at the target frequency. Physical realizability was further confirmed through Design Rule Checking and Layout-versus-Schematic verification, ensuring that the gate-level netlist could be manufactured without geometric violations. The power-delay product derived from post-synthesis reports provided a normalized metric for direct comparison against published vector-processor implementations.

IV PROPOSED ARCHITECTURE

The processor core is built around five principal hardware blocks: a five-stage Pipeline Controller, an Instruction Fetch and Decode Unit, a 64-entry General-Purpose Register file, a 64-operation ALU paired with a dedicated Barrel Shifter Rotator, and an ICG Control Logic block. Each block is described below, fig 4.1.1 followed by a discussion of their integration and physical implementation.

4.1 Block Diagram

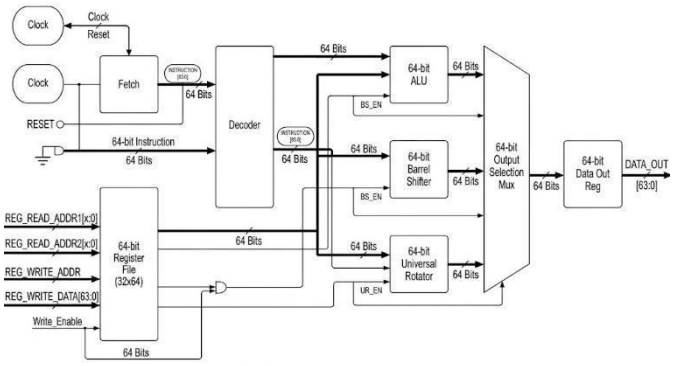


Fig: 4.1.1 Proposed 64-BIT RISC Processor Using Low Power Pipelining

Pipeline Controller: Sequences instruction flow through the Fetch, Decode, Execute, Memory, and Write-Back stages. Its primary responsibilities are resolving data and control hazards and ensuring that the pipeline progresses deterministically cycle by cycle without requiring an explicit finite-state machine for routine operation.

Instruction Fetch/Decode Unit: Retrieves instructions from the instruction memory and translates each opcode and operand field into a set of micro-control signals that activate the appropriate downstream resources.

64-bit General-Purpose Register File: Maintains the processor's architectural state as a 64-word, 64-bit-wide storage array. The file supports two simultaneous read ports and one write port, sized to keep register-access latency off the critical path.

64-Operation ALU and Barrel Shifter Rotator: Together these units constitute the execution core. The ALU handles the full complement of arithmetic and logical operations, while the Barrel Shifter Rotator performs any required multi-bit shift or rotate in a single clock cycle, avoiding the iterative shifting loops that would otherwise inflate both latency and switching activity.

ICG Control Logic: A dedicated hardware block that monitors the decoded instruction stream and drives the enable inputs of every ICG cell in the design. When a decoded instruction requires only the Barrel Shifter, the ICG cell feeding the ALU is deasserted; when an arithmetic instruction is dispatched, the shifter's clock is withheld. This mutual exclusion suppresses the dominant source of unnecessary switching within the 64-bit data path

4.2 Functional Module Description

Decoder: The instruction decoder maps each incoming machine-code word onto a set of binary control signals that steer operand multiplexers, select ALU function codes, and generate the enable signals consumed by the ICG logic. Correct decoder operation is essential to power management because every ICG decision in the pipeline depends on it.

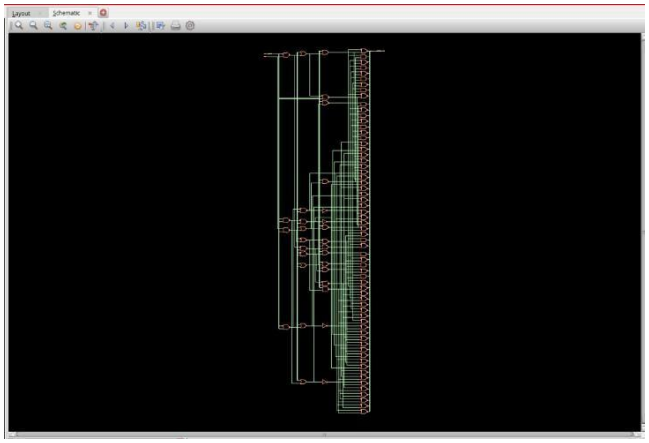


Fig: 4.2.1 Decoder Schematic Diagram

Barrel Shifter Rotator: The dedicated shift unit accepts a 64-bit operand and an amount field and produces the shifted or rotated result within a single clock period, irrespective of the magnitude of the shift. Single-cycle completion avoids the toggling of intermediate registers that accumulate during iterative shift sequences and substantially reduces the logic depth on the shift-critical path.

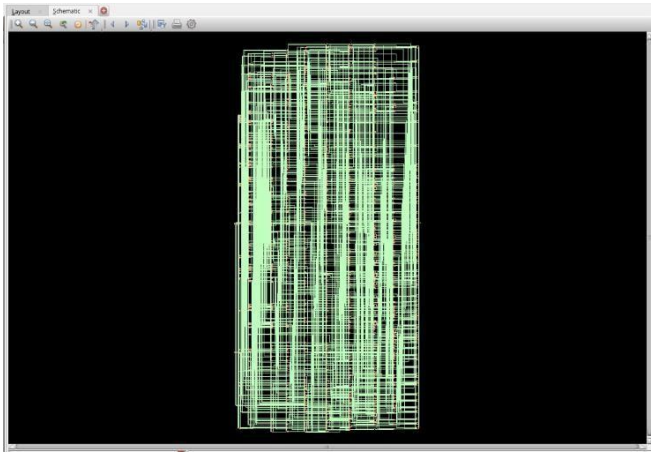


Fig: 4.2.2 Barrel Shifter Rotator Schematic Diagram

Universal Shift Rotator: Complementing the Barrel Shifter, this auxiliary unit supports logical left and right shifts, arithmetic right shifts, and both left and right rotations. Its inclusion ensures that specialized data manipulation requirements—such as byte swapping and sign extension—can be satisfied without routing such operations through the primary arithmetic path.

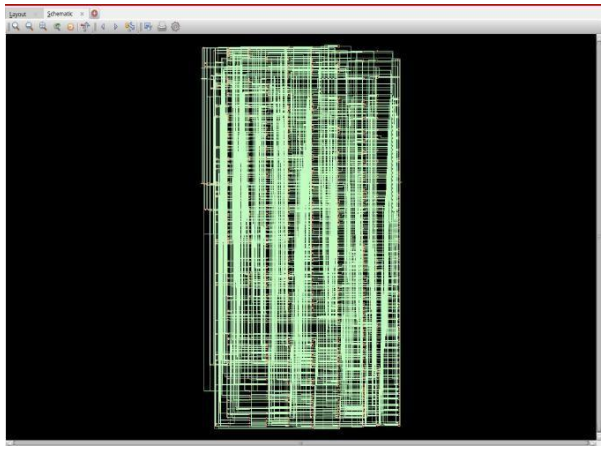


Fig 4.2.3 Universal Shift Rotator Schematic Diagram

- 64-bit Register File: The two-read-one-write register array is optimized for minimal access latency so that operands are available at the ALU inputs before the critical combinational path through the execution stage closes. Read and write ports are individually controlled to prevent spurious enable activity during cycles in which the register file is not addressed.

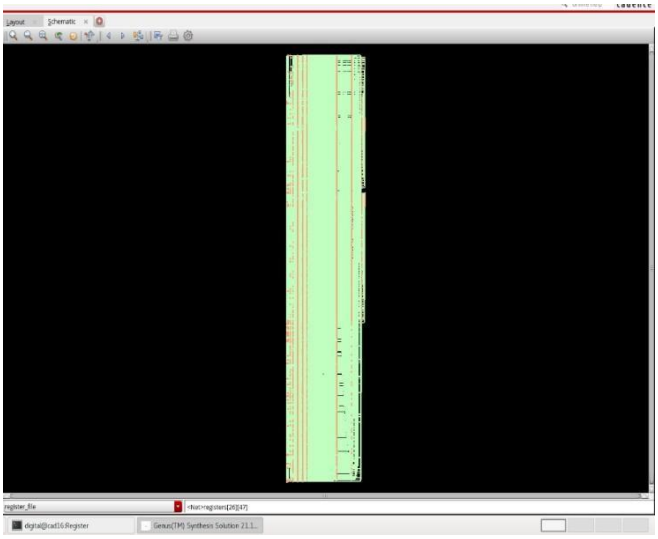


Fig: 4.2.4 Register File Schematic Diagram

64-Operation ALU: The execution engine supports 64 distinct operations covering addition, subtraction, bitwise logic, comparison, and selected multiply-accumulate functions. Its clock input is sourced through an ICG cell so that the entire ALU, with all its internal carry propagation and selection logic, can be frozen during shift and load/store instructions. This compute-on-demand approach eliminates switching activity in a block that would otherwise be the single largest contributor to dynamic power within the core.

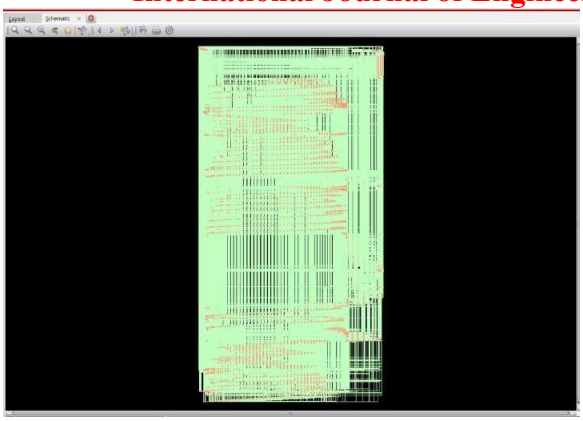


Fig:4.2.5 ALU Schematic Diagram

4.3 Pipeline Control and Top-level Architecture

A centralized Control Unit oversees all five pipeline stages without relying on an explicit FSM for routine instruction sequencing; instead, registered pipeline interlocks and forwarding multiplexers manage hazards within the dataflow itself. The Control Unit arbitrates access between the 64-entry GPR file and the two execution units, asserting write-enable signals only when a valid result is present and the destination register is ready to be updated. The Top-Level Module connects the decoder outputs to the ICG enable network, the register-file address and data buses, the ALU function and operand inputs, and the shift-amount inputs of the Barrel Shifter, thereby providing a single integration point at which timing closure can be verified across all inter-module interfaces.

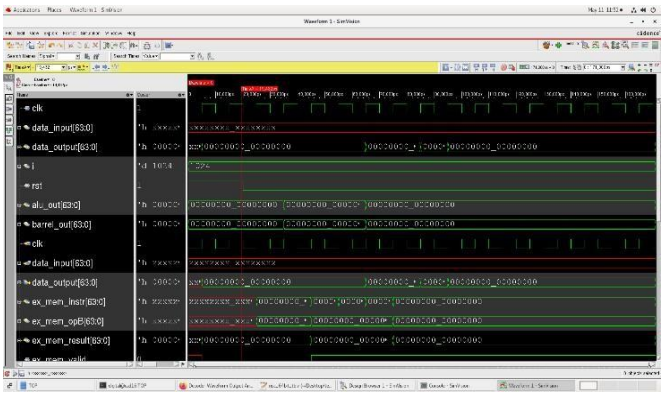


Fig: 4.3.1 Top Module Functional Simulation

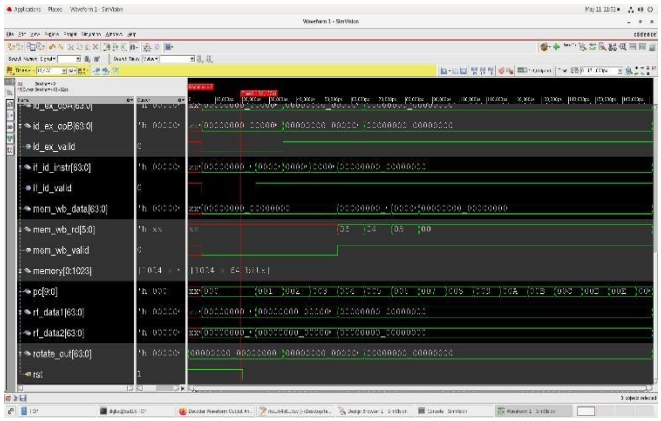


Fig: 4.3.2 Top Module Functional Simulation

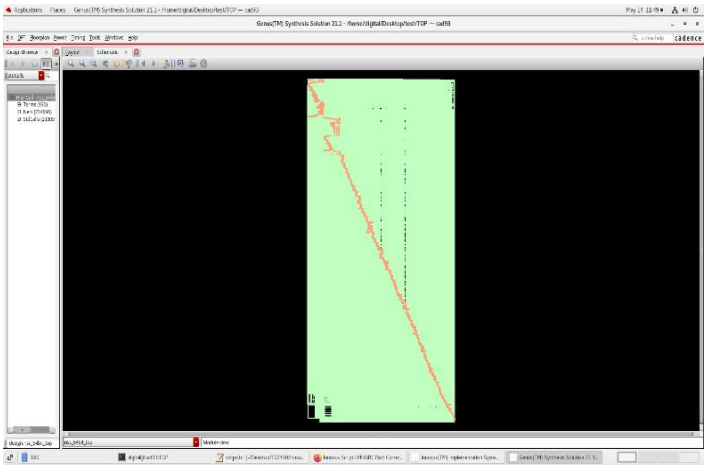


Fig: 4.3.3 Top Module Schematic Diagram

4.4 Physical Design Implementation

The gate-level netlist produced by Cadence Genus was validated against physical design constraints to confirm that the micro-architectural optimizations survive the translation from RTL to silicon geometry. Standard cells from the GPDK 90nm library were mapped to all logic, register, and clock-gating elements, and the resulting placement and routing were evaluated for timing slack, cell area, and power dissipation simultaneously. The clock tree was synthesized to meet a strict 100 MHz boundary (10.0 ns clock period), and no voltage reduction below the nominal supply was required to achieve closure.

The most timing-critical inter-stage boundary proved to be the interface between the Instruction Decode and Execute stages, where the 64-bit-wide bus transferring operands from the register file to both the ALU and the Barrel Shifter Rotator creates a dense routing region. Despite this routing complexity, per-stage logic depth was kept sufficiently shallow that the worst negative slack remained positive across all corners, confirming an adequate operating margin at 100 MHz.

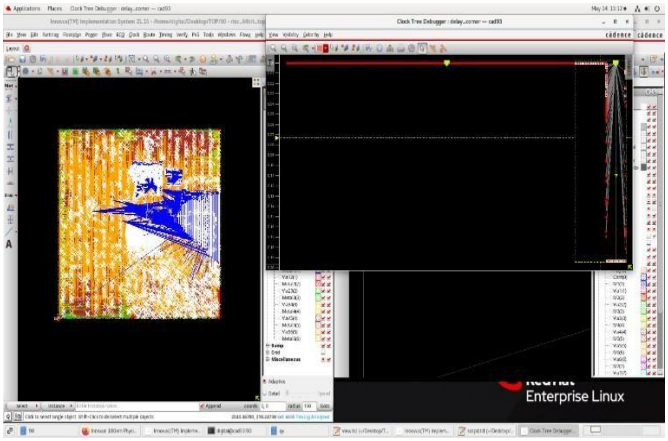


Fig: 4.1 Physical design of 64-BIT RISC Processor with Clock Tree

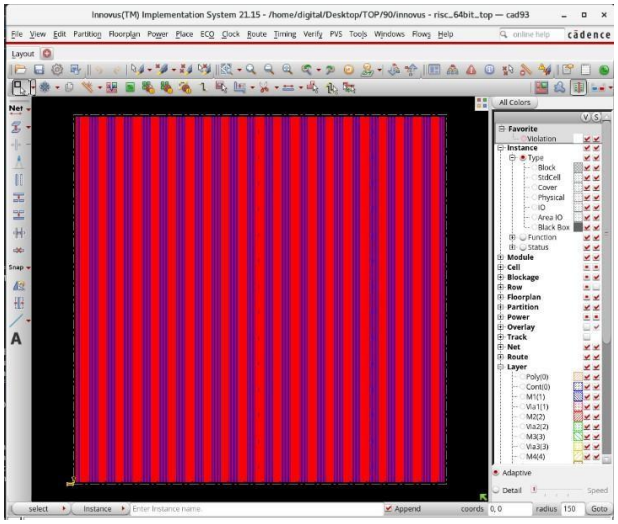
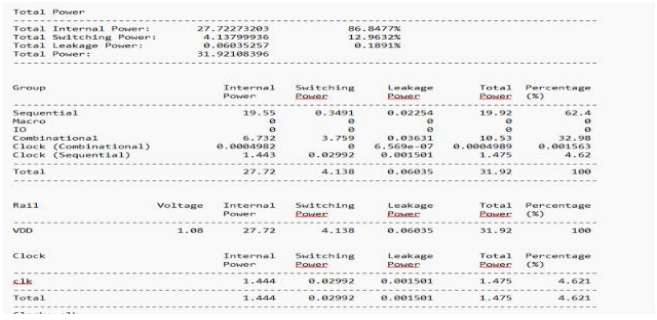


Fig: 4.4.2 Complete layout of Physical design of 64-BIT RISC Processor



```
Total Power
--------------------------------------------------------------------------------
Total Internal Power:       27.72273203           86.8477%
Total Switching Power:       4.13799936           12.9632%
Total Leakage Power:         0.06035257            0.1891%
Total Power:                31.92108396
--------------------------------------------------------------------------------

Group                    Internal  Switching  Leakage  Total  Percentage
                         Power     Power      Power    Power  (%)
--------------------------------------------------------------------------------
Sequential                 19.55     0.3491   0.02254  19.92        62.4
Macro                          0          0         0      0           0
IO                             0          0         0      0           0
Combinational              6.732      3.759   0.03631  10.53       32.98
Clock (Combinational)  0.0004982          0  6.569e-07  0.0004989  0.001563
Clock (Sequential)         1.443    0.02992  0.001501  1.475        4.62
--------------------------------------------------------------------------------
Total                      27.72      4.138   0.06035  31.92         100

Rail              Voltage  Internal  Switching  Leakage  Total  Percentage
                           Power     Power      Power    Power  (%)
--------------------------------------------------------------------------------
VDD                  1.08     27.72      4.138   0.06035  31.92         100

Clock                    Internal  Switching  Leakage  Total  Percentage
                         Power     Power      Power    Power  (%)
--------------------------------------------------------------------------------
clk                        1.444    0.02992  0.001501  1.475        4.621
--------------------------------------------------------------------------------
Total                      1.444    0.02992  0.001501  1.475        4.621
```

Fig: 4.4.3 Power Report of Physical design of 64-BIT RISC Processor

V RESULTS

The complete 64-bit RISC processor was synthesized with the Cadence Genus Synthesis Solution at a target clock frequency of 100 MHz, and the resulting gate-level netlist was verified against the original RTL through functional simulation. Waveform outputs confirmed that the instruction Fetch, Decode, Execute, Memory, and Write-Back stages each produced the expected outputs for a representative mix of arithmetic, shift, and load/store instructions.

5.1 Implementation and Timing Analysis

Timing closure was achieved with zero worst-negative-slack, meaning the critical path—running from the pipeline register id_ex_opB_reg[21] through the ALU to the pipeline register ex_mem_result_reg[51]—meets the 10 ns setup constraint precisely, with no additional voltage or frequency margin sacrificed. All hold-time checks across the five-stage pipeline also passed without requiring buffer insertion beyond what the clock tree synthesis algorithm introduced automatically.

5.2 Area and Gates Complexity

The synthesized design occupies approximately 7,383,946 technology-library area units realized across 233,051 cell instances. Sequential elements (flip-flops and latches) account for 70,048 instances and consume roughly 64% of total cell area, reflecting the register-heavy nature of a 64-entry, 64-bit-wide GPR file. Combinational logic cells number 137,540 and occupy 33.1% of area, while buffer and inverter cells account for the remaining 2.9%. This distribution confirms that the design is dominated by state-holding elements rather than deep combinational cones, which is consistent with a pipelined scalar architecture relying on registered interstage communication.

5.3 Power Consumption

Total on-chip power at nominal conditions measures 3.21 mW, a figure notably low for a 64-bit processor operating at 100 MHz in a 90nm process. Internal power—the dominant term, concentrated in the register file and control logic—accounts for the majority of this budget. Switching power, which reflects actual node transitions induced by data activity, constitutes only 16.18% of total power; the low proportion directly demonstrates the effectiveness of the ICG

Category	Leakage (W)	Internal (W)	Switching (W)	Total (W)	Row %
Register	1.25E-04	1.82E-03	5.42E-05	2.00E-03	62.40%
Logic	2.45E-05	7.12E-04	3.84E-04	1.12E-03	34.91%
Clock	0.00E+00	0.00E+00	8.12E-05	8.57E-05	2.67%

Fig: 5.1.1 Power Report of Physical design of the proposed 64-BIT RISC Processor

strategy in suppressing unnecessary toggling across the wide 64-bit data path. Clock network power is an equally revealing metric: at 2.67% of total consumption, it reflects the combined benefit of an efficient clock tree synthesis run and the ICG cells that prevent clock edges from propagating into idle functional blocks.

Metric	Proposed 64-BIT RISC Processor	Existing Design	Comparision percentage
Operational Frequency	100 MHz	100 MHz	0%
Area Efficiency	Lean (Scalar)	Massive (Vector/SRAM)	85%
Pipeline Complexity	5-Stage In-Order	High-Complexity Vector	60%
Logic/Gate Count	~6,177 Cells	High-Density Arrays	75%
Design Flexibility	High (Embedded/IoT)	Low (Matrix/Vector Only)	40%

Fig: 5.1.2 Comparison of the proposed 64-BIT RISC Processor and the existing design

VI CONCLUSION

This paper has presented the end-to-end design, synthesis, and physical verification of a 64-bit scalar RISC processor developed for embedded applications where area and power budgets are tightly constrained. The five-stage pipeline organization provides the throughput necessary for real-time workloads, while the strategic insertion of ICG cells across the clock distribution network prevents idle execution units from consuming dynamic power between active instruction cycles. Post-synthesis results at 100 MHz confirm zero worst-negative-slack, a total instance count of 233,051, and on-chip power consumption of approximately 3.21 mW—metrics that collectively demonstrate the viability of the scalar pipelined approach for IoT and battery-operated embedded platforms. Compared with vector processor architectures of equivalent word width, the proposed design offers a substantially more compact footprint and avoids the scheduling overheads associated with SIMD lane management, making it well suited to general-purpose embedded computation where silicon efficiency must be prioritized alongside performance.

REFERENCES

[1] Gangadhar K G, Indhushree D, Krishna Kumar G "Custom design of 16-bit RISC Processor Using Low Power Pipelining" 2024 4th International Conference on Intelligent Technologies (CONIT) 979-8-3503-4990-0/24/$31.00 ©2024 IEEE – DOI: 10.1109/ CONIT61985.2024.10627227.

[2] Matteo Perotti, Matheus Cavalcante, Alessandro Ottaviano, Jintao Liu, Luca Benini "Yun: An Open-Source, 64-Bit RISC-V-Based Vector Processor With Multi-Precision Integer and Floating-Point Support in 65-nm CMOS" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 70, NO. 10, OCTOBER 2023, 1549-7747 c 2023 IEEE.

[3] Priyanka Trivedi, Rajan Prasad Tripathi "Design and Analysis of 16-bit RISC Processor Using low Power Pipelining" International Conference on Computing, Communication and Automation (ICCCA2015).

[4] AGINETI ASHOK, V. RAVI "ASIC Design of MIPS Based RISC Processor for High Performance" 2017 International Conference on Nextgen Electronic Technologies, 978-1-5090-5913-3/17/$31.00 c 2017 IEEE.

[5] Muhammad Ali Raza, Iraj Shahzad, Hafsa Anwar, Muhammad Ali Qureshi, Farhan Hassan Malik, Muhammad Usman, Ali Khan "An Optimum Design and Implementation of a 64-bit ALU on CADENCE Using RISC-V Architecture" 023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICES&T) — 978-1-6654-5560-2/23/$31.00 ©2023 IEEE — DOI: 10.1109/ICEST56843.2023.10138869

[6] Satyam Shukla, Utkarsh, Md Azam, and Kailash Chandra Ray "An Efficient Fault-Tolerant Instruction Decoder for RISC-V Based Dual Core Soft-Processors" 1549-8328 © 2023 IEEE.

[7] Avanish Pratap Singh, Ashutosh Rajput, Amrit Prakash, P.C. Joshi, Anushka Rai "Design and Analysis of High-Speed RISC Processor Using Pipelining Technique" ISBN: 978-1-6654-7436-8/22/$31.00 ©2022 IEEE

[8] Jie Gao, Jun Zhang "Research and Design of RISC-V Four-Stage Out of-Order Execution Processor" 978-1-6654-6906-7/22/$31.00 © 2022 IEEE

[9] J. Ravindra, T. Anuradha "DESIGN OF LOWPOWER RISC PROCESSOR BY APPLYING CLOCK GATING TECHNIQUE" ISSN: 22489622 Vol. 2, Issue 3, May-Jun 2012, pp.094-099 / International Journal of Engineering Research and Applications (IJERA).

[10] Ramesh M, Charan Kumar D, Yashwanth, Jyoteeswara Reddy G, Naveen, Pandiaraj K. "Five Stage Pipelined MIPS Processor Verification Interface and Test Module using UVM" IEEE Xplore Part Number: CFP23DJ3-ART; ISBN: 979-8-3503-3360-2 ©2023 IEEE

[11] Sneha Mangal wedhe, Roopa Kulkarni and S. Y. Kulkarni "Low Power Implementation of 32-Bit RISC Processor with Pipelining" © Springer Nature Singapore Pte Ltd. 2019

[12] Wenyi Liu, Guilan Li, Xin Niu, Feng Hu, Bangjian xu "A Deeeply Pipelined 64-BIT Multiplier for High Performance RISC -V Processors" IEEE 2024 6th International Conference On Frontier Technologies of Information and Computer (ICFTIC) 979-8-3315-4175-0/24/$31.00 ©2024 IEEE