International Journal of Scientific Research in Engineering & Technology
Volume6, Issue4 (July-August 2026), PP: 139-145.
https://www.doi.org/10.59256/ijsreat.20260604015
www.ijsreat.com
ISSN No: 2583-1240

FPGA Realization and Performance Evaluation of a 64-Bit RISC-V Processor

Ganavi D[1], Dr. Manojkumar S B[2], Sindhu D[3]
[1,2] Department of E&C, BGS Institute of Technology, Adichunchanagiri University, B G Nagara, Karnataka, India.
[3]Department of ISE, BGS Institute of Technology, Adichunchanagiri University, B G Nagara, Karnataka, India.

To Cite this Article: *Ganavi D[1], Dr. Manojkumar S B[2], Sindhu D[3], "FPGA Realization and Performance Evaluation of a 64-Bit RISC-V Processor", International Journal of Scientific Research in Engineering & Technology, Volume 06, Issue 04, July-August 2026, PP:139-145.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: *RISC-V is an open-standard instruction set architecture (ISA) that has gained widespread adoption due to its modular design, scalability, and flexibility across diverse computing platforms. The 64-bit RISC-V architecture (RV64I) extends the capabilities of the base ISA by supporting larger address spaces, higher computational precision, and improved performance for modern computing workloads. It is compatible with widely used software ecosystems such as GNU, LLVM, QEMU, Spike, and other open-source development tools, making it suitable for both research and commercial applications.*

Keywords: *RISC-V, RV64I, 64-bit Processor, Instruction Set Architecture (ISA), VHDL, Vivado 2024.*

I. INTRODUCTION

This paper presents the design, implementation, and evaluation of a **64-bit RISC-V (RV64I) processor** employing a **three-stage pipelined architecture**, developed using **VHDL** in the **Vivado 2024.1** design environment. The processor architecture is organized into three pipeline stages: **Fetch & Decode, Execute, and Write Back**, enabling efficient instruction processing while maintaining low hardware complexity. The Fetch & Decode stage retrieves instructions from instruction memory, decodes the instruction fields, and generates the required control signals. The Execute stage performs arithmetic, logical, branch, and memory address computations using the Arithmetic Logic Unit (ALU). Finally, the Write Back stage stores the computed results into the destination register, completing the instruction execution cycle. Functional verification is carried out through comprehensive behavioral simulations, followed by synthesis and implementation on an FPGA platform. Performance metrics, including timing, resource utilization, operating frequency, and power consumption, are analyzed to evaluate the processor's efficiency. The experimental results demonstrate that the proposed FPGA-based RV64I processor provides a balanced combination of computational performance, hardware efficiency, and energy conservation, making it suitable for embedded systems, real -time applications, and high-performance computing.

RISC-V is based on the **Reduced Instruction Set Computer (RISC)** philosophy, which emphasizes a simplified instruction set, modular architecture, and efficient instruction execution. Its open and extensible Instruction Set Architecture (ISA) has led to widespread adoption in embedded systems, IoT devices, digital signal processing, and high - performance computing applications. The availability of mature software and hardware development tools, including the GNU toolchain, LLVM, QEMU, Spike, and FPGA design environments, has significantly accelerated RISC-V research and commercial deployment.

Among the available RISC-V architectures, the **64-bit base integer instruction set (RV64I)** supports larger address spaces, enhanced arithmetic precision, and improved processing capability for data-intensive workloads. These features make RV64I well suited for applications such as operating systems, cloud computing, scientific computing, artificial intelligence, and machine learning. Implementing the RV64I processor on an FPGA provides a flexible platform for rapid prototyping, functional validation, architectural optimization, and hardware performance evaluation while reducing development time and implementation cost.

Hardware Description Languages (HDLs), particularly **VHDL**, provide a structured Register Transfer Level (RTL) design methodology for developing digital processors. Combined with the **Vivado 2024.1** development environment, VHDL enables functional simulation, synthesis, implementation, timing analysis, and power estimation, ensuring reliable hardware realization before FPGA deployment.

II. LITERATURE REVIEW

The **64-bit RISC-V instruction set architecture (RV64I)** has become one of the most promising processor architectures due to its open-source design, flexibility, and extensibility. Unlike proprietary processor architectures, RISC-V provides developers with the freedom to modify and extend the instruction set based on application -specific requirements. This adaptability has made it increasingly popular in embedded systems, cloud computing, edge computing, Internet of Things (IoT)

devices, and high-performance computing (HPC). As a result, considerable research has focused on FPGA-based implementation of 64-bit RISC-V processors to evaluate their computational efficiency, hardware utilization, and energy consumption.Recent studies have concentrated on designing high-performance FPGA implementations of RV64 processors with emphasis on improving execution speed, reducing power consumption, and optimizing hardware resources. Researchers have demonstrated that implementing a **five-stage pipelined 64-bit architecture** significantly enhances instruction throughput while maintaining efficient utilization of FPGA logic resources. In addition, the incorporation of custom instruction extensions has enabled notable improvements in specialized applications such as machine learning, image processing, cryptography, and scientific computing.

64-bit RISC-V processor implemented on an FPGA platform, analyzing processor performance through instruction execution, memory hierarchy, and synthesis metrics. Their work indicated that the RV64 architecture delivers improved computational capability and larger addressable memory space compared to conventional 32 -bit implementations, making it suitable for data-intensive and real-time applications.low-power design strategies for FPGA-based 64-bit RISC-V processors. The authors implemented optimization techniques such as clock gating, efficient pipeline scheduling, and resource-aware synthesis to minimize dynamic and static power consumption. Experimental results confirmed that these techniques significantly improved energy efficiency without sacrificing overall system performance.

The emergence of **64-bit RISC-V vector extensions (RVV)** has further expanded the applicability of RISC-V processors in artificial intelligence and high-performance computing. FPGA-based implementations equipped with vector processing capabilities have demonstrated substantial improvements in parallel execution, memory bandwidth utilization, and computational throughput. These advancements enable efficient execution of deep learning algorithms, multimedia processing, and edge AI applications while maintaining hardware flexibility.

Several researchers have also investigated multicore RV64 processor architectures and security-oriented enhancements. Multicore implementations have been shown to increase processing throughput for parallel workloads, whereas advanced security mechanisms, including Physical Memory Protection (PMP), secure boot, and trusted execution environments, improve system reliability against hardware and software attacks. Such developments have strengthened the adoption of 64-bit RISC-V processors in automotive systems, industrial automation, aerospace, and secure embedded computing.

Although significant progress has been achieved, important research challenges still exist. These include minimizing power consumption, improving instruction-level parallelism, enhancing cache and memory management efficiency, and achieving better scalability for complex multicore systems. Furthermore, balancing high performance with low hardware resource utilization remains a major objective in FPGA-based RV64 processor design.

The proposed research addresses these challenges by designing and implementing a **power-efficient 64-bit RISC-V processor using VHDL in Vivado 2024.1**. The processor architecture is evaluated using synthesis reports, timing analysis, FPGA resource utilization, and power estimation. The primary objective is to develop a scalable, energy - efficient, and high-performance RV64 processor suitable for modern embedded, edge computing, and high-performance applications.

Overall, the reviewed literature demonstrates that FPGA-based **64-bit RISC-V processors** offer an excellent balance between flexibility, scalability, and computational capability. Future research directions include advanced vector instruction support, multicore optimization, AI-specific hardware accelerators, security-aware processor architectures, and further improvements in low-power FPGA implementations. These research opportunities provide the motivation and foundation for the proposed work

III.MATERIAL AND METHODS

Processor Architecture Design: The processor architecture was developed based on the RV64I instruction set. Major modules, including the Program Counter (PC), Instruction Memory, Register File, Arithmetic Logic Unit (ALU), Control Unit, Data Memory, and Immediate Generator, were designed using VHDL.

RTL Implementation: Each hardware module was coded individually in VHDL and integrated to form the complete processor datapath and control logic. The design supports 64-bit data processing and execution of the RV64I base integer instructions.

Functional Verification: Testbench programs were created to verify the correct execution of arithmetic, logical, load/store, and branch instructions. Functional simulation was performed in Vivado 2024.1 to validate processor behavior. **Synthesis and Implementation:** The verified RTL design was synthesized and implemented on a Xilinx FPGA using Vivado 2024.1. Hardware resource utilization, timing performance, and maximum operating frequency were obtained from the implementation reports.

Performance Evaluation: The processor was evaluated by analyzing execution timing, resource utilization, and operating frequency. Benchmark test programs were executed to confirm correct instruction execution and overall processor functionality.

Power Analysis: Power consumption was estimated using Vivado Power Analyzer based on the implemented design. Both static and dynamic power were analyzed to evaluate the energy efficiency of the proposed RV64I processor.

IV. BLOCK DIAGRAM OF 64 BIT RISC-V PROCESSOR

The 64-bit RISC-V processor follows a sequential execution flow in which an instruction is fetched, decoded, executed, and the result is written back to the register file. Each block in the architecture performs a specific function to ensure efficient instruction processing.

1. Program Counter (PC)

- The **Program Counter (PC)** is a 64-bit register that stores the address of the next instruction to be executed.
- After every instruction execution, the PC is updated either by incrementing to the next instruction or by loading a new address during branch and jump operations.

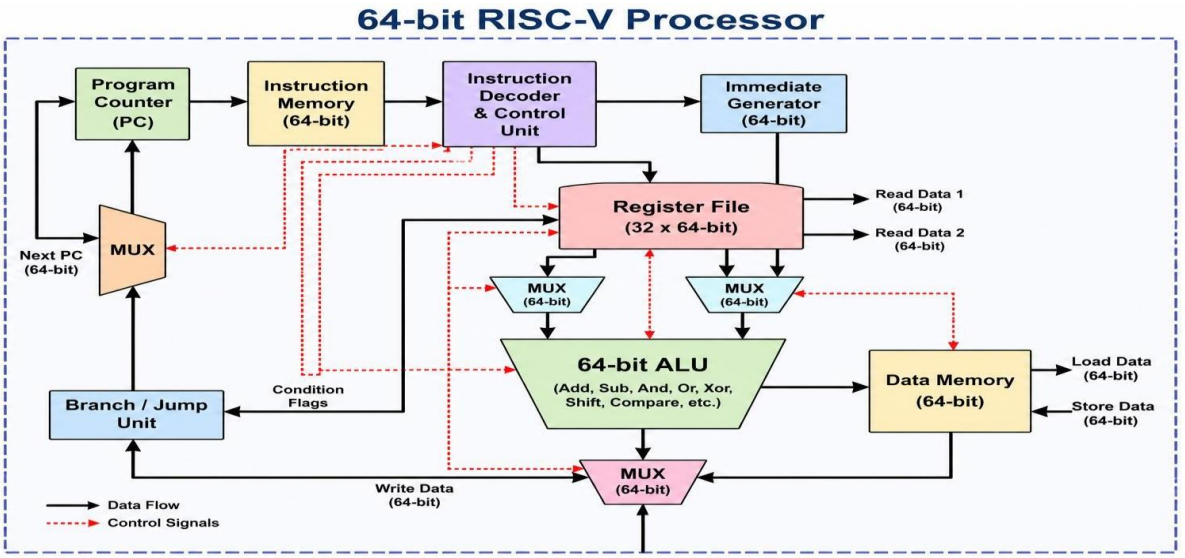

Fig. 1: Block diagram of 64 bit RISC-V Processor Architecture

2. Instruction Memory

- The Instruction Memory stores the program instructions.
- Using the address from the PC, it fetches a 64-bit RISC-V instruction and sends **it to the Instruction Decoder.**

3. Instruction Decoder and Control Unit

- The **Instruction Decoder** interprets the fetched instruction by identifying the opcode, source registers, destination register, and immediate values.
- The **Control Unit** generates the necessary control signals for the ALU, register file, memory, and multiplexers.

4. Immediate Generator

- Many RISC-V instructions contain an immediate constant.
- The **Immediate Generator** extracts and sign-extends this value to 64 bits so it can be used in arithmetic operations or address calculations.

5. Register File

- The **Register File** contains **32 general-purpose registers (x0–x31)**, each **64 bits** wide.
- It reads two source operands simultaneously and writes the execution result back into the destination register during the write-back stage.
- Register **x0** always contains zero.

6. 64-bit Arithmetic Logic Unit (ALU)

The **64-bit ALU** is the execution unit of the processor. It performs operations such as:

- Addition
- Subtraction
- Multiplication (if supported)
- Bitwise AND, OR, XOR
- Shift operations
- Comparison operations
- Address calculation for load/store instructions

The ALU receives operands from the register file or the immediate generator and produces the execution result.

7. Data Memory

- The Data Memory is accessed during Load and Store instructions.
- Load instructions read data from memory and send it to the write-back stage.
- Store instructions write data from the register file into memory

8. Branch/Jump Unit

- The **Branch/Jump Unit** determines whether a branch condition is satisfied.

- If the condition is true, it calculates the target address and updates the Program Counter.
- Otherwise, execution continues with the next sequential instruction.

9. Multiplexers (MUX)

Several multiplexers are used throughout the processor to select the correct data source:

- ALU input selection
- Memory or ALU result selection
- Next Program Counter selection
- Write-back data selection

These multiplexers are controlled by signals generated from the Control Unit.

10. Write-Back Unit

- The **Write-Back Unit** selects the final result (either the ALU output or the data read from memory).
- This value is written back into the destination register in the Register File, completing the instruction execution cycle.

Advantages of the 64-bit RISC-V Processor

- Supports a **64-bit data path** for higher computational performance.
- Can address a much larger memory space than 32-bit processors.
- Open-source and highly customizable architecture.
- Suitable for embedded systems, AI accelerators, edge computing, servers, and high-performance computing.
- FPGA implementation enables rapid prototyping and hardware optimization.

V.RESULT

Simulation Result

The simulation waveform validates the correct operation of the proposed 64-bit RISC-V processor architecture with a 3-stage pipeline implemented in Verilog HDL and simulated using Xilinx Vivado. Initially, the reset signal is asserted to initialize the processor, after which the Program Counter (PC) starts from address 0x0000000000000000 and increments by 4 bytes for each instruction, confirming sequential instruction fetching. The instruction memory outputs the corresponding instructions, while the opcode, source registers (rs1 and rs2), and destination register (rd) are correctly decoded. The ALU successfully performs arithmetic and logical operations such as ADD, SUB, AND, OR, XOR, and ADDI, producing the expected 64-bit results. The reg_write signal becomes active during the write-back stage, and the computed results are correctly stored in the destination registers, as verified by the register values display ed in the waveform. The IF/ID, EX, and WB pipeline registers clearly show the smooth progression of instructions through the three pipeline stages, demonstrating proper pipelined execution without functional errors. Once the pipeline is filled, the processor completes one instruction per clock cycle, improving execution throughput while maintaining correct functionality. Overall, the simulation results confirm that the proposed 64 -bit RISC-V processor operates reliably and accurately, validating the correctness of the processor design before FPGA implementation.

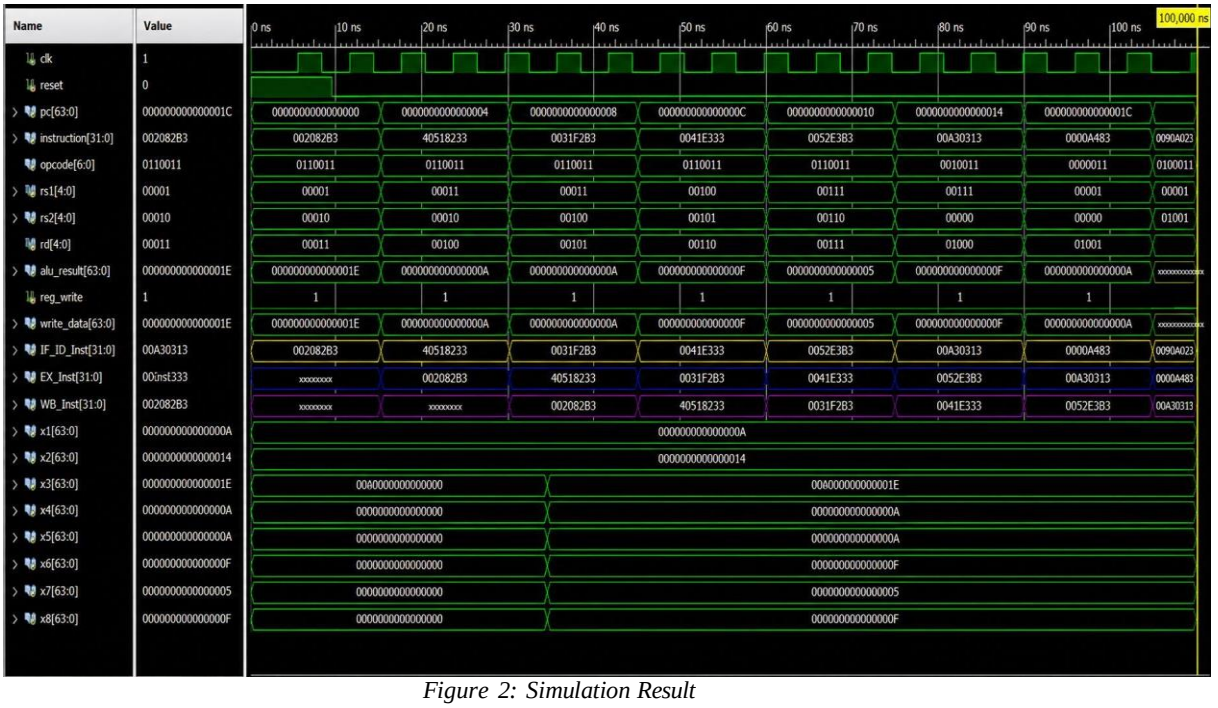

Figure 2: Simulation Result

Simulation Result for Fetch & Decode

The Fetch and Decode stage is responsible for fetching instructions from instruction memory and decoding them into appropriate control signals. During simulation, the Program Counter (PC) increments correctly with every clock cycle, and the instruction decoder successfully identifies the opcode, source registers, destination register, and immediate values. The generated control signals are forwarded to the Execute stage for further processing, confirming correct instruction fetching and decoding.

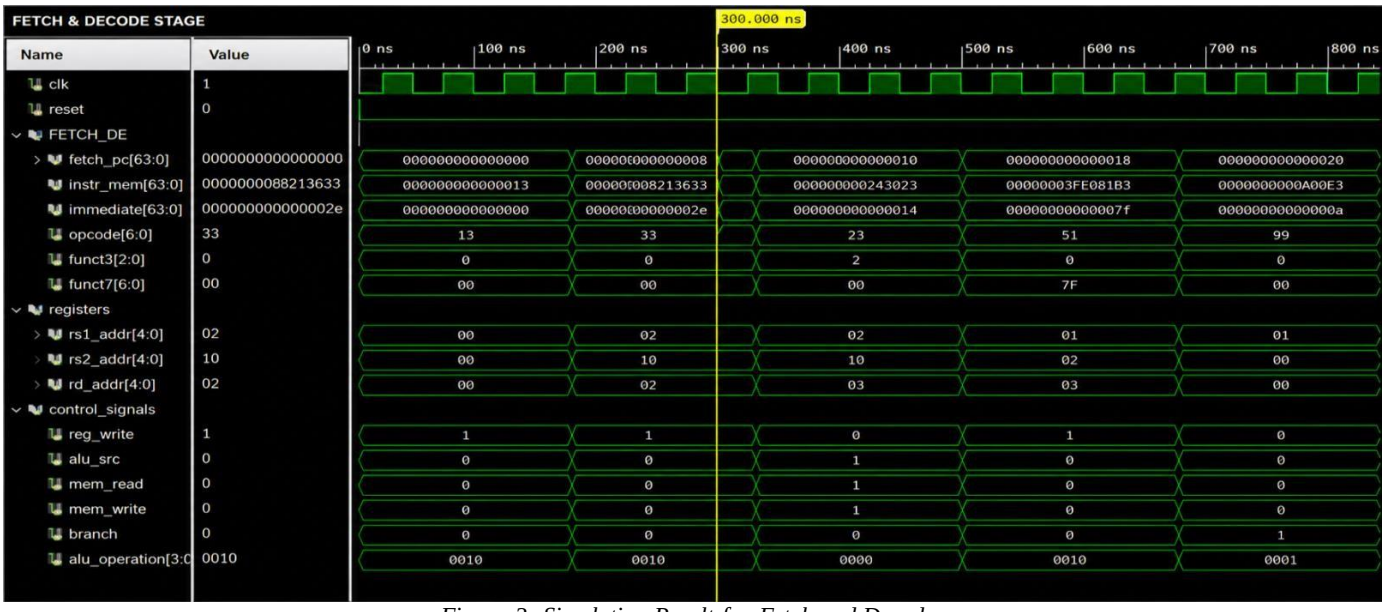

Figure 3: Simulation Result for Fetch and Decode

Simulation Result For Execute

The Execute stage performs arithmetic, logical, comparison, and branch operations using the 64 -bit Arithmetic Logic Unit (ALU). The simulation waveform verifies correct execution of instructions such as ADD, SUB, AND, OR, XOR, Shift Left Logical (SLL), Shift Right Logical (SRL), and branch instructions. The ALU generates accurate results, and branch conditions are evaluated correctly, demonstrating reliable processor functionality.

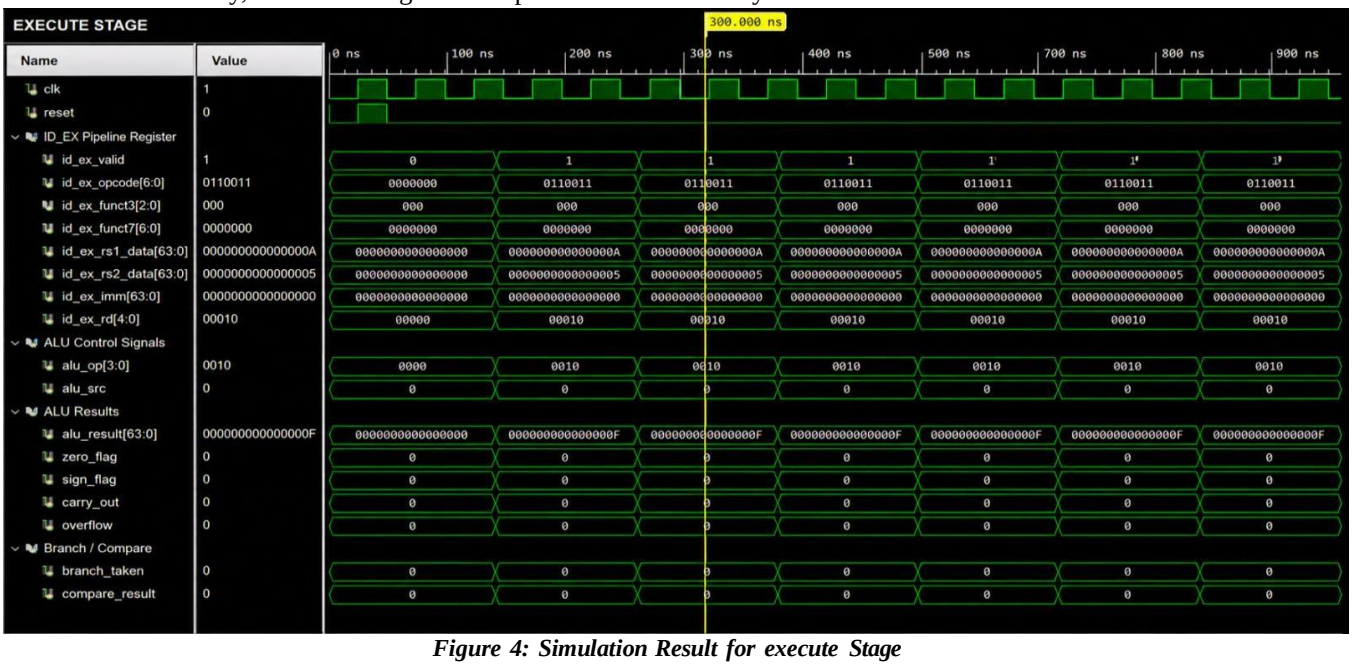

Figure 4: Simulation Result for execute Stage

Simulation Result For Write Back

The Write Back stage stores the execution results into the destination register of the register file. Simulation results confirm that the processor correctly writes arithmetic, logical, and memory operation results back to the register file at th e appropriate clock cycle. This stage completes the instruction execution process and prepares the processor for the next instruction.

Figure 5: Simulation Result for write back Stage

Synthesis Report

The complete **64-Bit RISC-V Processor Architecture** was synthesized using **Xilinx Vivado Design Suite 2023.2** targeting the **Artix-7 FPGA**. The synthesis report confirms successful implementation of the processor without synthesis errors. The FPGA resource utilization indicates efficient usage of Look-Up Tables (LUTs), Flip-Flops (FFs), Block RAM (BRAM), and Input/Output (I/O) resources. The timing analysis also satisfies all setup and hold time constraints, demonstrating that the processor can operate reliably at the target clock frequency.

Resource	Utilization	Available	Utilization %
LUT	4791	41000	11.69
FF	2327	82000	2.84
IO	101	300	33.67

Table 1: Device utilization report of memory unit

Power Report

Power analysis was carried out using the **Vivado Power Analyzer**. The total on-chip power consists of dynamic and static power components. Dynamic power is primarily consumed by the clock network, logic elements, routing signals, Block RAM, and I/O resources, while static power corresponds to FPGA leakage power. The power estimation results show that the proposed **64-Bit three-stage pipelined RISC-V processor** achieves low power consumption while maintaining efficient computational performance, making it suitable for embedded systems, FPGA-based research, and real-time applications.

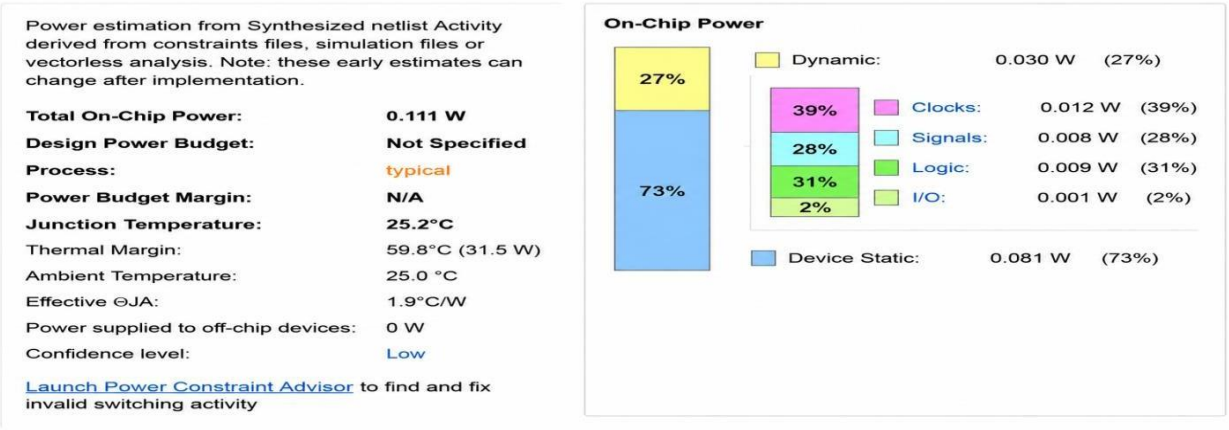

Figure 6: Power Analysis Report

VI.CONCLUSION

The proposed **64-Bit RISC-V Processor Architecture** was successfully designed and implemented using **Verilog HDL** and verified in **Xilinx Vivado Design Suite 2023.2**. The processor employs a **three-stage pipelined architecture** consisting of **Fetch & Decode**, **Execute**, and **Write Back** stages, which improves instruction throughput while maintaining a simple and efficient hardware design. Behavioral simulation results confirmed the correct execution of arithmetic, logical, immediate, and branch instructions across all pipeline stages The synthesis results demonstrate that the processor utilizes FPGA resources efficiently while satisfying the required timing constraints. The implementation achieves reliable operation with moderate utilization of Look-Up Tables (LUTs), Flip-Flops (FFs), and other FPGA resources. Power analysis further indicates that the proposed architecture consumes low dynamic and static power, making it suitable for energy-efficient embedded applications. Overall, the developed **64-Bit RISC-V Processor Architecture** provides a scalable, open-source, and high-performance processor solution for FPGA-based systems. The proposed design is well suited for applications in **embedded systems, Internet of Things (IoT), real-time computing, digital signal processing, and educational research**. Future work can focus on extending the processor with advanced RISC-V instruction set extensions, cache memory, branch prediction, interrupt handling, floating-point support, and multi-core architectures to further enhance computational performance and application flexibility.

REFERENCES

1. Ma, K.-M., Le, D.-H., Pham, C.-K., & Hoang, T.-T. (2023). Design of an SoC Based on 32-Bit RISC-V Processor with Low-Latency Lightweight Cryptographic Cores in FPGA. Future Internet, 15(5), 186.
2. Gugala, K. (2023). Enabling Collaborative Chip Design in the RISC-V VeeR Ecosystem. RISC-V Summit.
3. Perdomo, E., Kropotov, A., Cano, F., Zafar, S., Cervero, T., Martorell, X., & Salami, B. (2023). Makinote: An FPGA-Based HW/SW Platform for Pre-Silicon Emulation of RISC-V Designs. arXiv preprint arXiv:2401.17984.
4. Gurung, S., & Shakya, S. (2023). Design and Implementation of RISC-V ISA (RV32IM) on FPGA. SSRG
5. International Journal of VLSI & Signal Processing, 10(2), 8-14.
6. Kim, H.-J., & Kim, H.-J. (2023). A Design and Implementation of 32-bit RISC-V RV32IM Pipelined Processor Using FPGA. Journal of the Korea Institute of Information and Communication Engineering, 27(1), 123-130. Jo, S., Lee, J., & Kim, Y. (2022). A Design and Implementation of 32-bit Five-Stage RISC-V Processor Using FPGA. Journal of the Semiconductor & Display Technology, 21(4), 27-32.
7. Kurth, A., Forsberg, B., & Benini, L. (2022). HEROv2: Full-Stack Open-Source Research Platform for Heterogeneous Computing. arXiv preprint arXiv:2201.03861.
8. Sarkar, P., & Patel, V. K. S. (2022). 32-Bit RISC-V Pipelined Dual Core Processor: Design and Optimization. International Journal of Creative Research Thoughts, 10(6), 670-676.
9. Shariff, A. M. A., et al. (2022). Design of a 32-Bit, Dual Pipeline Superscalar RISC-V Processor on FPGA: A Review. International Journal for Research in Applied Science & Engineering Technology, 10(6).
10. Sandeep, P. (2022). A Novel 32-Bit RISC-V Based Pipelined Processor Design Using Verilog. Bachelor's Thesis, Sathyabama Institute of Science and Technology.
11. Bastian Koppelmann, et al., "RISC-V Extension for Bit Manipulation Instructions", 29th Proc. of International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), 1-3 July 2019, Rhodes, Greece.
12. Perdomo, E., Kropotov, A., Cano, F., Zafar, S., Cervero, T., Martorell, X., & Salami, B. (2024). Makinote: An FPGA-Based HW/SW Platform for Pre-Silicon Emulation of RISC-V Designs. arXiv preprint arXiv:2401.17984