



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Design and Verification of Custom RISC Processor Architecture Using Verilog and Testing -Validation with System Verilog System

Karuna Gone

Assistant Professor, Department of ECE, Sri Vasavi Institute of Engineering & Technology (Autonomous),
Nandamuru, Pedana, A.P, India

S.Laharisa, A.Yasaswini, P.Nagendram, V.S.S.Prasad

Department of ECE, Sri Vasavi Institute of Engineering & Technology (Autonomous), Nandamuru, Pedana, A.P, India

ABSTRACT: This project presents the comprehensive design, implementation, and verification of a custom Reduced Instruction Set Computing (RISC) processor architecture using Verilog HDL, with an emphasis on advanced verification techniques using System Verilog. The processor features a modular architecture optimized for performance and scalability, incorporating key components such as the ALU, register file, instruction decoder, and control logic. The design process follows industry-standard digital design methodologies, from RTL modeling to synthesis and timing analysis. A robust System Verilog-based verification environment was developed to validate the processor's functionality, employing constrained-random test benches, coverage-driven verification, assertions, and functional coverage metrics. The test environment also includes scoreboard and monitor modules to support self-checking capabilities. Simulation results confirm the correct operation of all instruction types, including arithmetic, logic, branching, and memory operations, under a variety of edge cases and input scenarios. This work demonstrates the effectiveness of combining Verilog for hardware description with System Verilog for advanced verification, showcasing a methodology suitable for academic research, prototyping, or industrial application in embedded systems and digital processor design.

KEYWORDS: RISC Processor, Verilog, System Verilog, Pipeline Architecture, Hazard Detection, Functional Coverage.

I. INTRODUCTION

In the ever-evolving landscape of digital system design, the need for efficient, reliable, and high performance processors remains a cornerstone of modern computing systems. Among various processor design paradigms, the Reduced Instruction Set Computing (RISC) architecture has emerged as a dominant approach due to its simplicity, scalability, and performance efficiency. RISC architectures focus on a limited set of instructions, enabling faster execution, easier pipelining, and simplified control logic. This project delves into the advanced design and rigorous verification of a custom RISC processor architecture, leveraging Verilog for hardware description and System Verilog for comprehensive testing and validation. The goal is to build a processor that is not only functionally correct but also robust under varying conditions, with verification coverage that ensures both expected and edge-case behaviors are thoroughly examined. The Custom RISC Processor is designed from the ground up, starting with an architectural specification that defines a clean instruction set and data path. The processor includes essential components such as a 32-bit Arithmetic Logic Unit (ALU), general-purpose register file, control unit, instruction decoder, program counter, and memory interface. Special care is taken in designing the instruction set to support essential operations—arithmetic, logical, branching, and memory access—while maintaining architectural simplicity. To facilitate instruction execution, a multistage datapath is constructed, with considerations for future enhancements such as pipelining, forwarding, and hazard detection. The architectural model emphasizes modularity and reusability, making the processor suitable for educational, research, and embedded system development purposes.

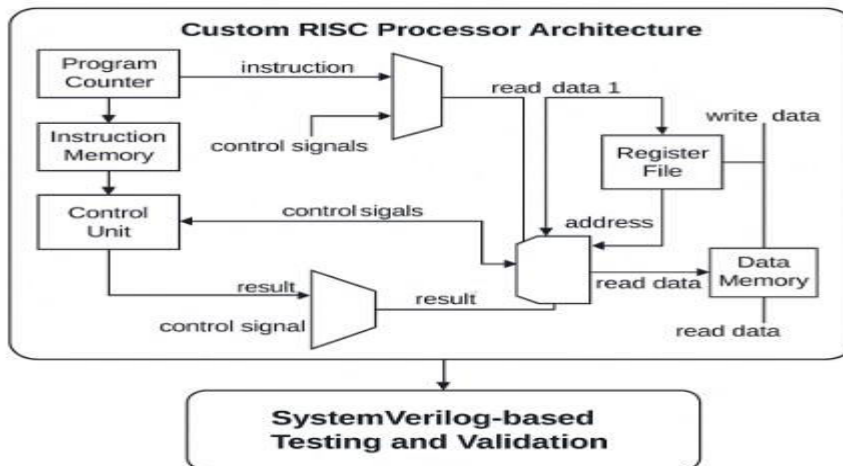
II. PROPOSED SYSTEM

The Custom RISC Processor is a simplified, yet fully functional processor core designed to execute a predefined instruction set with high efficiency and accuracy. It is implemented in Verilog HDL, adhering to a modular design approach that separates the processor's core functionalities into manageable and reusable blocks. The primary components of the processor include the Program Counter (PC), Instruction Memory, Control Unit, Register File, Arithmetic Logic Unit (ALU), Data Memory, and various multiplexers and adders to facilitate instruction flow. The Program Counter holds the address of the next instruction to execute, which is fetched from the Instruction Memory. The instruction is then decoded by the Control Unit to generate control signals that guide data movement and operation execution. The Register File supplies the source operands to the ALU, which performs arithmetic and logical operations based on the instruction opcode. The results can be stored back into the Register File or written to Data Memory, depending on whether it is a computational or memory access instruction. Each module is carefully interconnected to allow a seamless flow of instructions through a single cycle datapath, though the architecture can be scaled into a pipelined version in future enhancements.

The Control Unit plays a vital role in decoding instructions and generating precise control signals for multiplexers, register write enables, memory read/write, and ALU operation codes. ALU control logic is derived from instruction function fields and opcode combinations, enabling support for a variety of operations such as addition, subtraction, bitwise AND/OR/XOR, shift operations, and comparisons. The processor also supports branch instructions, where control logic evaluates branch conditions and appropriately updates the Program Counter to enable conditional or unconditional jumps. To manage hazards and ensure correct data access, the design ensures that no two stages interfere destructively in a single cycle. Although the current implementation does not feature forwarding or stalling mechanisms, it is built with architectural foresight that allows such mechanisms to be integrated later. Verification of the processor is conducted using System Verilog, where a comprehensive testbench simulates real-world execution and thoroughly checks the processor's behavior. The verification environment includes test generators, drivers, monitors, a scoreboard, and reference models. Constrained random testing is used to simulate thousands of instruction sequences, each varying in input values and execution paths. Assertions are embedded within the testbench to check for protocol adherence and to immediately catch illegal states or invalid outputs. A scoreboard mechanism compares the actual output from the DUT (Design Under Test) against expected results generated from a golden reference model, ensuring accuracy. Additionally, functional coverage points are defined to track which instruction types, registers, branches, and edge cases are exercised during simulation. This helps identify gaps in test coverage and refine test cases accordingly.

The processor is validated through a mix of directed and random test cases that confirm correct instruction execution, proper memory access, accurate data routing, and error detection. For instance, load and store instructions are verified by writing specific data to memory and reloading it into registers to check consistency. Branching logic is tested under various conditional inputs to verify the correctness of PC updates. Arithmetic operations are validated against expected outcomes across positive, negative, and overflow scenarios. The verification environment is modular, allowing unit testing of each core component (like the ALU or Register File) as well as system-level integration tests. Simulation results are analyzed using waveform viewers and log files to confirm the internal signal transitions, data path integrity, and timing behavior.

Finally, the verified RTL design is synthesized using FPGA tools to ensure that the design is not only functionally correct but also synthesizable and resource-efficient. Synthesis reports provide information on area utilization, maximum clock frequency, and power estimates, all of which guide future optimization. The entire processor design, including the testbench, is developed with scalability and clarity in mind, enabling further enhancements such as pipelining, exception handling, or interrupt support. The project demonstrates the complete lifecycle of processor development—from architecture definition and Verilog implementation to SystemVerilog-based functional verification—making it a strong example of practical digital design using modern hardware description and verification languages.



A custom RISC processor typically follows a streamlined architecture designed for simplicity, efficiency, and speed. Its core components include the Instruction Memory, where program instructions are stored, and the Program Counter (PC), which keeps track of the current instruction address. The processor operates in several stages: Instruction Fetch (IF) retrieves the instruction from memory; Instruction Decode (ID) interprets the instruction and accesses the required data from the Register File. The Execute (EX) stage performs operations using the ALU (Arithmetic Logic Unit), such as arithmetic or logic functions. If needed, the Memory (MEM) stage accesses Data Memory for load or store instructions. Finally, the Write Back (WB) stage stores the result back into the register file. A Control Unit coordinates all these operations by generating appropriate control signals. This modular and pipelined design allows RISC processors to execute instructions efficiently and is widely used in embedded systems and custom hardware solutions.

III. EXPERIMENTAL RESULTS

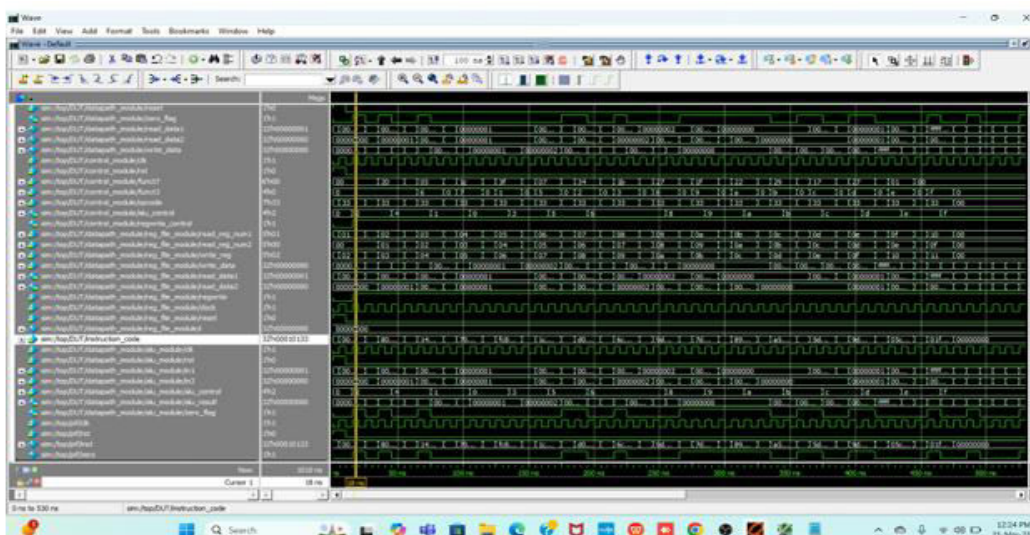


Fig.1.DUT Results

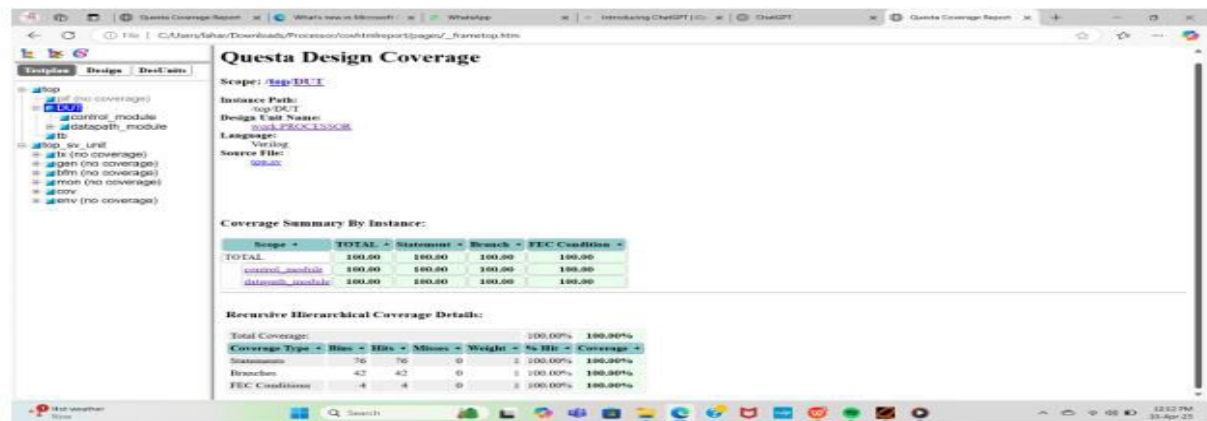


Fig.2.Functional Result

IV. CONCLUSION

The successful design and verification of a custom RISC processor architecture demonstrate the powerful synergy between efficient hardware description using Verilog and robust functional verification through SystemVerilog. This project highlights the complete lifecycle of processor development—from instruction set design, datapath creation, and control unit integration, to comprehensive testing and validation of the processor’s functionality. The RISC architecture implemented provides a simplified yet powerful core capable of executing arithmetic, logical, branching, and memory operations, making it ideal for embedded and educational applications. The use of Verilog allowed for a clean and modular RTL design, ensuring the processor is easily extendable for future enhancements such as pipelining, hazard detection, and interrupt handling. On the verification front, SystemVerilog enabled the creation of an intelligent testbench architecture with self-checking capabilities, constrained-random testing, and functional coverage analysis. These verification strategies significantly increased confidence in the correctness and robustness of the processor design, minimizing the risk of undetected logic errors. Furthermore, this project reinforces the importance of verification as a critical phase in hardware development, showcasing how simulation-based methods and assertions can help catch subtle bugs and edge cases early in the design process.

REFERENCES

- [1] Andrew Waterman, Krste Asanovi, and Five Inc. “The RISC-V Instruction Set Manual Volume I: User Level ISA Document Version 2.” CS Division, EECS Department, University of California, Berkeley 2017.
- [2] David A. Patterson and John L. Hennessy “Computer Organization and Design RISC-V Edition: The Hardware Software Interface.” Morgan Kaufmann Publishers Inc., 2017, San Francisco, CA, USA.
- [3] Ludovico Poli, Sangeet Saha, Xiaojun Zhai and Klaus D. McDonald- Maier, "Design and Implementation of a RISC V Processor on FPGA" 17th International Conference on Mobility, Sensing and Networking (MSN)
- [4] Yassin M. Y. Hasan and Lina J. Karam, “Morphological Text Extraction from Images”,IEEE Transactions On Image Processing, vol. 9, No. 11, 2000
- [5] Don Kurian Dennis, Ayushi Priyam, Sukhpreet Singh Virk, Sajal Agrawal, Tanuj Sharma, Arijit Mondal and Kailash Chandra Ray, "Single Cycle RISC-V Micro Architecture Processor and its FPGA Prototype", Seventh International Symposium on Embedded Computing and System Design (ISED), Durgapur, India, 2017, pp. 1- 5, doi: 10.1109/ISED.2017.8303926.
- [6] Wendi Zhang, Yonghui Zhang and Kun Zhao, “Design and Verification of Three-stage Pipeline CPU Based on RISC-V Architecture” 5th Asian Conference on Artificial Intelligence Technology (ACAIT) 2021.
- [7]] P. Singh, A. Rai, A. Rajput, P. C. Joshi and A. Prakash, "Design and Analysis of High Speed RISC Processor Using Pipelining Technique," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 1642-1645, doi: 10.1109/ICAC3N56670.2022.10074423.

- [8] J. Jeemon, "Pipelined 8-bit RISC processor design using Verilog HDL on FPGA," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology(RTEICT),Bangalore,India,2016,pp.2023-2027,doi: 10.1109/RTEICT.2016.7808194
- [9] S. Prabhakaran, M. N and V. Vedanarayanan, "Design and Analysis of a Multi Clocked Pipelined Processor Based on RISC-V," 2022 International Conference on Communication, ComputingandInternetofThings(IC3IoT),Chennai,India,2022,pp.1-5,doi: 10.1109/IC3IOT53935.2022.9767960
- [10] A. Tiwari, P. Guha, G. Trivedi, N. Gupta, N. Jayaraj and J. Pidanic, "IndiRA: Design and Implementation of a Pipelined RISC-V Processor," 2023 33rd International Conference Radioelektronika (RADIOELEKTRONIKA), Pardubice, Czech Republic, 2023, pp. 1-6, doi: 10.1109/RADIOELEKTRONIKA57919.2023.10109058
- [11] I. Thanga Dharsni, K. S. Pande and M. K. Panda, "Optimized Hazard Free Pipelined Architecture Block for RV32I RISC-V Processor," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 739-746, doi: 10.1109/ICOSEC54921.2022.9952122.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details