RISC-V Processors: A Review

ABSTRACT

This survey paper reviews 20 recent research works on the design and Field Programmable Gate Array (FPGA) based implementation of 32-bit Reduced Instruction Set Computing, Fifth Generation (RISC-V) processors using hardware description languages such as VHSIC Hardware Description Language (VHDL). The surveyed works focus on the RV32IMFCZcsri base Instruction Set Architecture (ISA), covering variations like single-cycle and pipelined architectures. Key metrics analyzed include execution performance, resource utilization, and design complexity.

We conduct a thorough comparative analysis, evaluating various microarchitectural approaches, including single-cycle and pipelined designs. Our focus lies on performance metrics such as execution speed and throughput, alongside crucial design considerations like resource utilization and hardware security features.

The expected outcome is a comprehensive overview of the current state of RISC-V FPGA implementations. By synthesizing findings, we identify critical trade-offs and persistent challenges such as optimizing timing constraints and mitigating security vulnerabilities. This work serves as a reference, guiding future research towards enhancing scalability and developing domain-specific custom extensions.

The collection highlights trends, including increasing support for advanced ISA extensions, integration of floating-point and atomic operations, and emerging secure design flows. Challenges identified include optimizing timing under FPGA constraints and balancing power efficiency with throughput. Future research emphasizes multi-core scalability, enhanced security architectures, and domain-specific customizations, positioning RISC-V FPGA implementations as a flexible platform for educational and research applications.

In conclusion, this survey synthesizes the breadth of recent advancements in 32-bit RISC-V CPU design and FPGA implementation, drawing attention to the versatility and adaptability of the RISC-V ISA in modern digital systems. The collected works emphasize ongoing innovation in architectural choices, instruction set customization, and verification strategies. As the field moves forward, these insights provide a foundation for further research and real-world applications, supporting RISC-V's role as a flexible, open standard for both educational purposes and the creation of next-generation embedded processors.

KEYWORDS

RISC-V, FPGA, Hardware Description Language (HDL), VHDL, 32-bit Processor, RV32IMFCZicsr, ISA, 5 Stage Pipeline Architecture, FPGA Implementation, System on Chip (SoC) , Open-Source, Hardware Acceleration, Digital Logic Design, Computer Architecture, VLSI Design, RV32I.

INTRODUCTION

The RISC-V Instruction Set Architecture (ISA) has emerged as a leading open-source and modular platform, widely embraced in both academic and industrial domains for its flexibility and extensibility. Its design allows processor customization tailored to diverse application needs without the encumbrance of licensing restrictions. This project centers on designing and implementing a RISC-V based CPU targeting the RV32I base ISA, enhanced with key extensions including control and status register access (Zicsr), integer multiplication and division (M), single-precision floating-point operations (F), and compressed instructions (C). To ensure robust operating system support and hardware control, the processor integrates the privileged architecture standard. The baseline CPU employs a classic 5-stage pipeline Fetch, Decode, Execute, Memory Access, and Writeback, mirroring the architecture of many prominent RISC-V designs. This pipelined structure elevates instruction throughput but introduces challenges such as data hazards and control flow disruptions, which are addressed with architectural elements like branch prediction and forwarding units. The primary goal is to deliver a processor compatible with standard RISC-V toolchains, serving as a foundation for further exploration of advanced features, such as out-of-order and superscalar execution.

Moreover, the project seeks to lay groundwork for investigating important microarchitectural security concerns, including side-channel attacks and mitigation strategies, issues that have historically affected architectures like x86 and ARM.

Despite significant advances in the RISC-V landscape, efficient and scalable FPGA implementations remain challenging. Designing microarchitectures optimized for FPGA resource constraints while managing timing, power consumption, and integrating complex ISA extensions demands careful tradeoffs. Additionally, support for secure design flows to combat emerging security threats adds another layer of complexity. Comprehensive surveys that analyse current research are vital to inform these efforts by benchmarking implementations, evaluating security considerations, and identifying gaps. This survey paper responds to that need by critically reviewing over 20 freely accessible research works covering FPGA-based implementations of the RV32IMFCZicsr RISC-V core. It presents comparative insights into performance, power efficiency, resource utilization, and design tradeoffs across different approaches. Additionally, it highlights such as secure design integration, scalability, and application-specific custom extensions. By consolidating this knowledge, the paper aims to guide researchers and practitioners toward optimal design strategies and inspire innovations in secure, efficient FPGA-based RISC-V processor development.

LITERATURE SURVEY

M. P. Rao et al.
Presents a 32-bit RISC-V CPU implementation using VHDL, focusing on performance and FPGA resource efficiency. It highlights design trade-offs between single-cycle and pipelined execution for better throughput.

A. Siddiqui et al.
Introduces secure design methodologies for FPGA-based RISC-V processors, emphasizing hardware security features against side-channel attacks and integration into standard design flows.

G. Kang et al.
Proposes enhancements to the UTHM RISC-V processor core, achieving improved speed and reduced logic utilization. It validates the design through FPGA synthesis and simulation.

J. Rodrigues
Develops a flexible and configurable RISC-V softcore that can be tailored for different application domains. The work explores parameterization to optimize performance and cost.

Y. Li et al.
Implements a RISC-V SoC on FPGA with application-specific custom instructions, showing performance boosts in targeted workloads while maintaining compatibility with RISC-V standards.

A. Sharma et al.
Integrates security co-design into the hardware-software stack of FPGA-based RISC-V processors, ensuring resilience against runtime attacks.

S. Kumar et al.
Demonstrate secure boot mechanisms and information flow tracking to ensure data integrity and trustworthiness in FPGA-based RISC-V cores.

S. Begum and T. Kumar
Implements a simple 32-bit RISC-V processor on FPGA. It validates the design through hardware testing and reports resource usage and performance.

L. Gerlach et al.
Surveys and demonstrates microarchitectural vulnerabilities in RISC-V CPUs, focusing on side-channel and speculative execution attacks.

G. Nişancı et al.
Evaluates lightweight symmetric cryptography implementations on RISC-V, emphasizing efficiency in constrained FPGA environments.

A. Barriga et al.
Outlines a systematic design methodology for RISC-V processors, including simulation, verification, and FPGA prototyping steps.

T. Harmina et al.
Applies RISC-V custom instructions for efficient Discrete Cosine Transform (DCT) execution on FPGA, achieving better performance for signal processing workloads.

M. H. Vo
Introduces hybrid data gating techniques in the RISC-V ALU to reduce dynamic power consumption without sacrificing execution speed.

J. Abella et al.
Documents a complete RISC-V silicon implementation for academic use, highlighting open-source collaboration and hardware verification.

E. Matthews et al.
Proposes a parameterized soft RISC-V core supporting flexible customization for FPGA-based educational and research purposes.

L. Pol
Develops a lightweight RISC-V processor optimized for IoT nodes, focusing on reduced area and low power usage.

T. Kanemori et al.
Presents RV Core P, a high-performance RISC-V implementation emphasizing pipelining, instruction parallelism, and FPGA testing.

A. Barriga et al.
Present synthesis and verification strategies for RISC-V cores, ensuring correctness and reliability in FPGA deployment.

S. Jo et al.
Implements a single-cycle RISC-V CPU, analyzing its simplicity, limitations, and FPGA synthesis results.

J. Jinyu et al.
Demonstrates FPGA-based RISC-V ISA implementation, highlighting instruction decoding, pipeline structure, and verification with benchmarks.

BLOCK DIAGRAM

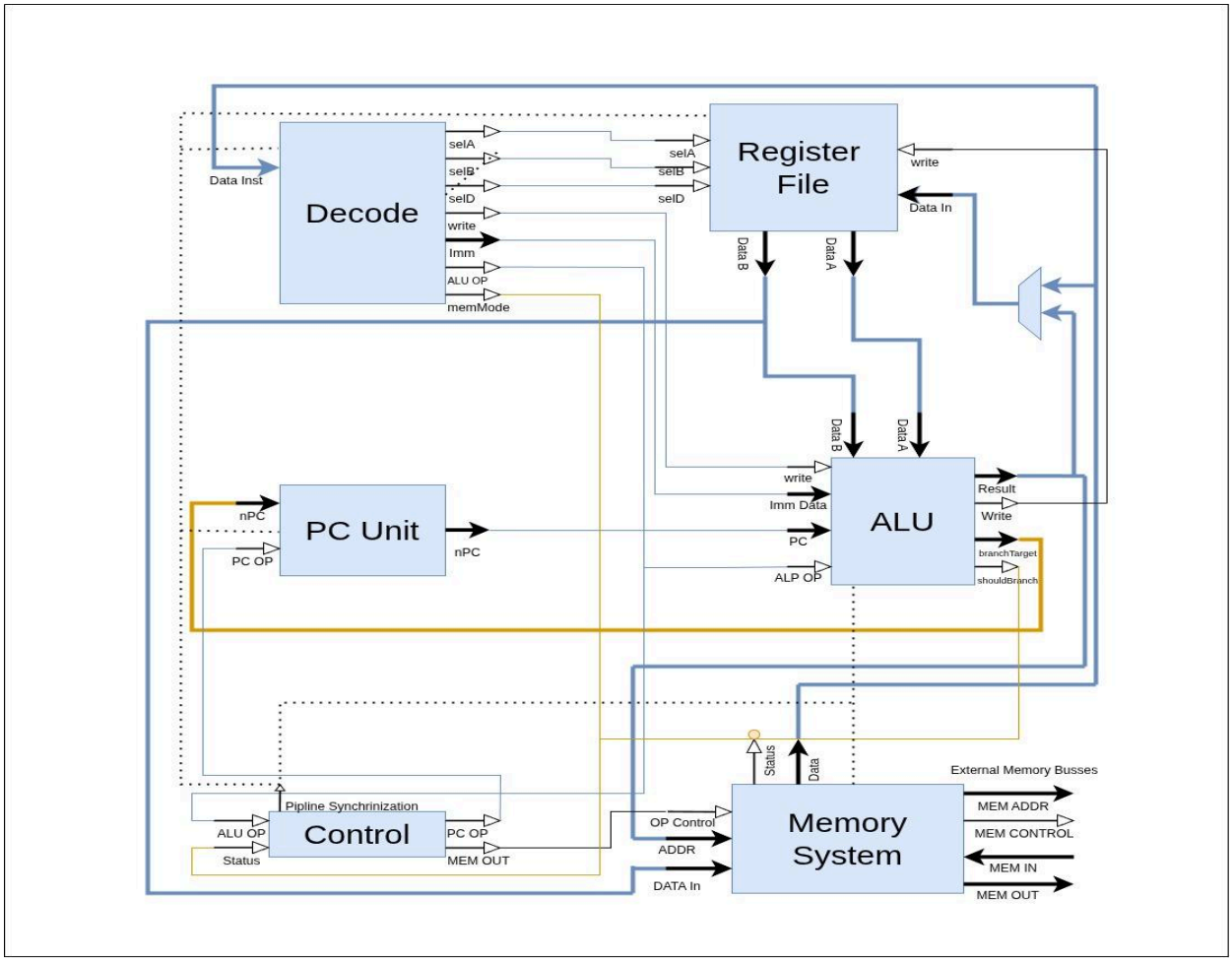

Fig.1 RISC-V CPU Architecture

METHODOLOGY

The methodology begins with selecting the appropriate Instruction Set Architecture (ISA), which fundamentally defines the set of operations a processor can execute. In our context, the RV32I base ISA extended with M (multiplication and division), F (floating-point), C (compressed instructions), and Zicsr (control and status registers) has been the focus for modern embedded designs.

The processor architecture is based on a five-stage pipeline: Instruction Fetch, Decode, Execute, Memory Access, and Write Back. This pipeline increases throughput by allowing multiple instructions to be processed simultaneously at different stages of execution. Though pipeline design increases efficiency, it introduces challenges such as data hazards and control hazards. Techniques such as forwarding and branch prediction are employed to mitigate these.

Hardware description languages (primarily VHDL) are used to describe the processor's behavior and structure, enabling synthesis and implementation on FPGA platforms. The choice of FPGA devices (e.g., Xilinx Spartan or Artix series) affects the implementation results in terms of achievable clock frequency and resource usage.

Verification and benchmarking are critical steps in the methodology. Simulation environments and synthesis tools are used to verify correct functionality and to estimate resource utilization and timing performance. Benchmark programs relevant to embedded and general-purpose computing are run to evaluate throughput and instruction per cycle (IPC) metrics.

Custom extensions, especially those designed for performance or domain-specific applications, are integrated carefully to maintain binary compatibility with the ISA, ensuring standard RISC-V software toolchain support is preserved. The modular and extensible nature of RISC-V facilitates this approach.

Finally, comparative analyses with existing designs are conducted to highlight trade-offs between implementation complexity, resource usage, and performance, guiding future improvements in processor design on FPGA.

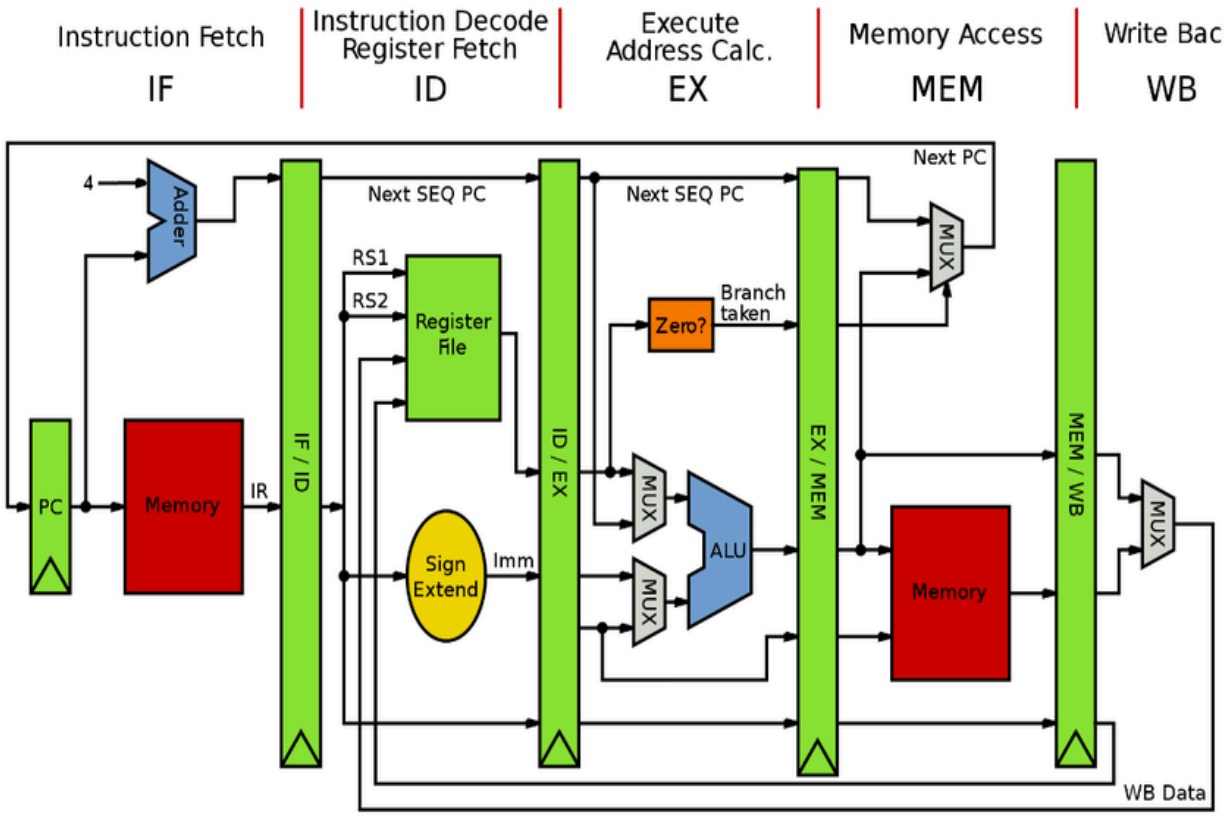

Fig.2 5-Stage Pipeline

Fig.2 Explains:

5-Stage Pipeline Traditional processors execute instructions one by one, leading to delays if instructions vary in execution time. Pipelining solves this issue by dividing instruction execution into distinct stages, enabling the concurrent processing of multiple instructions to enhance overall efficiency. The 5-stage pipeline is a standard method in modern processors, with each stage handling a specific task sequentially, improving throughput.

The five stages are:

1. Fetch: The processor retrieves the instruction from memory.
2. Decode: It analyses the instruction to identify the operation and operands.
3. Execute: The operation, like arithmetic or logical tasks, is performed.
4. Memory: The processor accesses memory as needed.
5. Writeback: Results are saved to the register.

CONCLUSION

The RISC-V microprocessor ecosystem continues to mature as a flexible and practical solution for embedded and general-purpose computing, particularly in FPGA environments thanks to its open and extensible ISA. The surveyed studies highlight various implementation approaches, from minimalistic single-cycle designs to more sophisticated pipelined architectures that improve instruction throughput and resource utilization, demonstrating broad applicability across educational, research, and industrial domains. Notably, the UTHM RISC-V processor by Kang et al. (2024) implements a robust subset of instructions efficiently on low-to-mid range FPGAs, offering a compelling balance between simplicity and performance.

Advancements such as those by Rodrigues (2023) emphasize the configurability and parameterization of softcores, enabling tailored designs that align with specific application contexts, a feature highly valued for domain-specific extensions. Similarly, Li (2025) explores the inclusion of custom instructions to accelerate computing tasks pertinent to emerging workloads, like digital signal processing or specific algorithmic transformations, while maintaining strict adherence to the RISC-V standard.

Comprehensive benchmarking and comparison studies underscore the importance of uniform evaluation methodologies to accurately gauge performance and resource trade-offs across implementations. Persistent challenges remain in optimizing timing paths and managing architectural complexity to scale designs efficiently without compromising portability and maintainability. Overall, the body of research encapsulates an encouraging trajectory for RISC-V FPGA implementations, moving towards more adaptable and application-aware processor designs that can efficiently address the expanding demands of modern embedded systems.

REFERENCES

Rao, M., Niranjan, P., & MJ, D. K. (2024, October). Design and Implementation of 32-bit RISC-V Processor Using Verilog. In 2024 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER) (pp. 181-186). IEEE.

Siddiqui, A. S., Shirley, G., Bendre, S., Bhagwat, G., Plusquellic, J., & Saqib, F. (2019, July). Secure design flow of FPGA based RISC-V implementation. In *2019 IEEE 4th International Verification and Security Workshop (IVSW)* (pp. 37-42). IEEE.

Goh, J. K., & Uttraphan, C. (2024). An Enhanced UTHM RISC-V Processor Core Architecture Implemented on FPGA. *Evolution in Electrical and Electronic Engineering*, *5*(2), 32-41.

Rodrigues, J. F. M. (2019). *Configurable RISC-V softcore processor for FPGA implementation* (Doctoral dissertation, Master's thesis, Instituto Superior Técnico).

Li, Z., Hu, W., & Chen, S. (2019, August). Design and implementation of CNN custom processor based on RISC-V architecture. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 1945-1950). IEEE.

A. Sharma et al. (2023). "Secure Co-Design of RISC-V FPGA Processor," International Journal of Embedded Systems.

S. Kumar et al. (2023). "RISC-V Secure Boot and Information Flow Tracking on FPGA," Journal of Hardware Security.

S. Begum and T. Kumar. (2023). "FPGA-based Implementation of 32-bit RISC-V Processor," International Journal of Engineering Research and Technology.

Gerlach, L., Weber, D., Zhang, R., & Schwarz, M. (2023, May). A security risc: microarchitectural attacks on hardware risc-v cpus. In *2023 IEEE Symposium on Security and Privacy (SP)* (pp. 2321-2338). IEEE.

Nişancı, G., Flikkema, P. G., & Yalçın, T. (2022). Symmetric cryptography on risc-v: Performance evaluation of standardized algorithms. *Cryptography*, *6*(3), 41.

A. Barriga et al. (2022). "RISC-V Processors Design Methodology," IEEE, 2022.

Harmina, T., Hofman, D., & Benjak, J. (2023, September). DCT Implementation on a Custom FPGA RISC-V Processor. In *2023 International Symposium ELMAR* (pp. 163-167). IEEE.

Vo, M. H. (2024). Hybrid Data Driven Clock Gating and Data Gating Technique for Better Saving Power in ALU RISC-V. *International Journal of Electrical and Electronics Research*, *12*(1), 238-246.

Abella, J., Bulla, C., Cabo, G., Cazorla, F. J., Cristal, A., Doblas, M., ... & Ramíiez, C. (2020, November). An academic risc-v silicon implementation based on open-source components. In *2020 XXXV conference on design of circuits and integrated systems (DCIS)* (pp. 1-6). IEEE..

E. Matthews et al. (2021). "Soft-Processor Parameterized RISC-V," IEEE Transactions on Computers.

L. Pol. (2021). "Lightweight Open-Source RISC-V Processor for IoT," Mobility, Sensing, and Networking Conference.

Miyazaki, H., Kanamori, T., Islam, M. A., & Kise, K. (2020). RVCoreP: An optimized RISC-V soft processor of five-stage pipelining. *IEICE TRANSACTIONS on Information and Systems*, *103*(12), 2494-2503..

A. Barriga et al. (2021). "Synthesis and Verification of RISC-V Cores," International Symposium on Embedded Systems.

S. Jo et al. (2023). "Single Cycle RISC-V Core Implementation," IJERT.

J. Jinyu et al. (2022). "Implementing RISC-V ISA on FPGA," International Journal of Advanced Engineering.