Reduced Instruction Set Computer (RISC-V)

Hannah Lee, Master of Science in Cybersecurity

Sumeet Singh, Master of Science in Computer Science

Christian Morris, Master of Science in Computer Science  
CS533 Computer Architecture, City University of Seattle

[leehannah@cityuniversity.edu](mailto:leehannah@cityuniversity.edu), [singhsumeet1@cityuniversity.edu](mailto:singhsumeet1@cityuniversity.edu), [morrischristian@cityuniversity.edu](mailto:morrischristian@cityuniversity.edu)

**Abstract**

This paper focuses on Reduced Instruction Set Computer (RISC-V). Initially Designed at the University of California, Berkeley, RISC architecture evolved into RISC-V. While several RISC implementations were available in the market, they were often limited by restricted access. In 2011, the first version of RISC-V was released, and after several iterations, it has been adopted by numerous organizations. RISC-V offers simplicity, modularity, and extensibility, driving its growing adoption across various industries. As an open-source instruction set architecture (ISA), RISC-V provides extensive customization options, enabling developers to tailor it to specific use cases. This flexibility has led to its adoption across industries ranging from embedded systems and IoT devices to high-performance computing and artificial intelligence. This report will provide examples of the RISC-V instruction set, details on memory and I/O handling, and highlight specialized processing units, with a focus on parallel processing and pipelining for enhanced performance. It will discuss the performance characteristics of RISC-V, examining factors such as instruction throughput and efficiency. Furthermore, it will explore RISC-V's application and suitability across various industries, from embedded systems like IoT to high-performance sectors such as AI.

**Keywords:** RISC-V, Reduced Instruction Set Computing, modularity, microprocessors, low-power computing, computer architecture, hardware development, System-on-Chip, edge computing

**1. INTRODUCTION**

RISC-V was conceived at the University of California, Berkeley, as an alternative to proprietary instruction set architecture. Since its first public release, it has gained considerable attention for its open-source nature, which reduces cost barriers and fosters innovation. Commercial and academic adoption has accelerated, with numerous vendors offering RISC-V-based microcontrollers, development boards, and design tools (What is RISC-V?, 2024). This openness encourages collaboration across diverse sectors—ranging from low-power Internet of Things (IoT) devices to high-performance data center accelerators—making RISC-V a versatile option for modern computing demands.

**2. RISC-V ARCHITECTURE AND INSTRUCTION SET**

A defining characteristic of RISC-V is its modular instruction set design. The base instruction set encompasses essential integer arithmetic and logic operations, while a series of optional extensions add capabilities like floating-point arithmetic, vector operations, and atomic instructions. These extensions can be selectively included or omitted, depending on the needs of the application (Sharma, 2024). For example, an embedded sensor node might exclude floating-point instructions to minimize power consumption, whereas a data center processor might incorporate specialized vector instructions to accelerate machine learning or cryptographic workloads.

**2.1 Modularity of the RISC-V Instruction Set**

With RISC-V's modular designs, hardware designers are able to select only the extensions and features they need, avoiding the constraints of a one-size-fits-all approach. This flexibility is essential for customizing processors to meet the specific demands of various applications.

**2.2 Base Instruction Set**

The foundational RISC-V instruction set includes basic operations like arithmetic, logic, and control flow, which are sufficient for many embedded and low-power applications. These instructions form the foundation of the architecture, and additional features can be added via extensions.

**2.1 Sample Instructions**

A screenshot of a computer

AI-generated content may be incorrect.**ADD (Addition):** Adds two integer registers and stores the result. In **Figure 1**, Rd is the destination register, rs1 and rs2 are source registers.

**Figure 1**

A screen shot of a computer

AI-generated content may be incorrect.**LW/SW (Load Word / Store Word):** Transfers data between registers and memory. As shown in **Figure 2**, rd is the destination register and rs1 is the source register holding the memory address. t0 is loaded with the word at address 0x200.

**Figure 2**

A computer screen with green text

AI-generated content may be incorrect.In **Figure 3**, rs2 is the source register and rs1 holds the memory address. Memory location 0x200 now contains a word with the value 40.

**Figure 3**

A black screen with green text

AI-generated content may be incorrect.**BEQ/BNE (Branch if Equal / Branch if Not Equal):** Implements conditional branching to manage control flow. **Figure 4** illustrates the BEQ, where if x5 and x6 are equal, the next instruction, the if\_loop, is executed.

**Figure 4**

**A screenshot of a computer

AI-generated content may be incorrect.**In Figure 5, the next instruction, exit, is executed, if x5 and x6 are not equal.

**Figure 5**

**2.2 Memory Model**

RISC-V adopts a load-store architecture, which stipulates that all operations (beyond simple load and store) must be performed in registers. This design choice simplifies pipelining and reduces instruction complexity.

**3. MEMORY AND I/O HANDLING**

The load-store paradigm extends to how RISC-V interfaces with memory and I/O devices. Data movement relies on dedicated instructions that interact either with a unified memory address space or with memory-mapped I/O registers (Sa et al., 2022). System implementers have the freedom to choose or invent alternative I/O models—such as port-mapped or hybrid schemes—without breaking compatibility with the base instruction set. This flexibility is particularly beneficial in embedded systems, which often have unique memory layouts and stringent real-time requirements.

**4. SPECIALIZED PROCESSING UNITS, PARALLEL PROCESSING, AND PIPELINING**

**4.1 Specialized Processing Units**

RISC-V permits designers to integrate co-processors or custom accelerators without altering the fundamental base architecture. Such specialized units may include hardware encryption blocks for security-sensitive applications or vector engines for handling large-scale data parallelism. This modular approach streamlines hardware-software co-design, allowing the ecosystem to incorporate rapid innovations without waiting for proprietary ISA updates.

**4.2 Parallel Processing and Pipelining**

Many RISC-V implementations incorporate pipeline stages (such as fetch, decode, execute, memory access, and write-back) to maximize throughput. The architecture also supports multicore and multithreading approaches for further parallelism (Shukla & Ray, 2022). Effective pipelining and parallel execution can significantly boost performance in domains like data analytics, cryptography, and scientific computing.

**5. PERFORMANCE AND USE CASES**

RISC-V’s focus on simplicity, tailored approach and extensibility makes it well-suited for a variety of applications:

* **Embedded Systems and IoT:** Minimal power consumption and a small silicon footprint.
* **Consumer Electronics:** Flexible designs for smartphones, smart TVs, and other devices.
* **Automotive:** Open-source verifiability and long-term reliability.
* **Data Centers:** Configurable extensions for high-throughput tasks such as AI or large-scale data processing.

Key performance factors include pipeline depth, cache design, branch prediction mechanisms, and specialized hardware extensions. Designers optimize these components based on the target application’s performance per watt requirements, latency constraints, and cost considerations.

In October 2024, NVIDIA announced plans to ship approximately one billion RISC-V cores across its GPUs, CPUs, SoCs, and other products, marking a significant milestone for the architecture (Shilov, 2024). This transition from proprietary microcontrollers to RISC-V cores underscores the architecture's scalability and flexibility. The embedded RISC-V cores play vital roles in tasks such as power management, security, and compute resource allocation, enhancing both performance and efficiency. Moreover, RISC-V’s increasing prominence in AI chip development highlights its adoption by leading tech companies across multiple industries (Eadline, 2024).

**6. STRATEGIC PLAN**

In addition to evaluating RISC-V at the technical level, organizations adopting this architecture should articulate a strategic plan integrating security, innovation, and cost management goals.

**6.1 Overall Smart and Secure Program Mission**

A “Smart and Secure” mission often guides the adoption of cutting-edge technologies that balance innovation with robust safeguards. RISC-V’s open-source model supports transparent hardware design, enabling deeper scrutiny and more flexible security extensions.

**6.2 External and Internal Influencers**

* **External Influencers:** Customer demands, regulatory compliance for safety/security, and the increasing complexity of software-defined products.
* **Internal Influencers:** Corporate culture favoring innovation, budget constraints, and the existing engineering talent pool’s experience.

**6.3 Strategic Goals and Objectives**

* **Innovation Acceleration:** Reduce time-to-market by leveraging open-source tools and customizable extensions.
* **Security and Trust:** Implement verified and auditable RISC-V cores that enhance cybersecurity.
* **Cost Optimization:** Avoid licensing fees typical of proprietary ISAs.
* **Scalability:** Adapt the same ISA for both low-power and high-performance products.

**6.4 Key Business Benefits**

**Lower Development Costs:** Open ISA eliminates expensive royalty agreements.

**Supply Chain Transparency:** Entirely traceable hardware design.

**Strong Community Support:** An active ecosystem offering extensive design resources.

**6.5 Metrics for Measuring Outcomes**

**Performance per Watt:** Critical for both battery-powered devices and large-scale data centers.

**Time to Prototype:** Evaluates how quickly engineers can iterate on hardware designs.

**Total Cost of Ownership:** Captures both upfront development and long-term maintenance costs.

**Security Posture:** Tracks vulnerabilities found and resolved over product lifecycles.

**6.6 Specific Initiatives and Timelines**

**6-Month Pilot:** Validate RISC-V cores on FPGAs for selected product features.

**1-Year Integration:** Transition pilot successes into full-scale system-on-chip (SoC) designs.

**Ongoing Upgrades:** Evaluate new RISC-V extensions annually for evolving AI, IoT, and security needs.

**6.7 Cloud Framework and AWS Services**

A robust cloud framework can complement RISC-V-based devices by providing scalable data handling and analytics:

**AWS IoT Core:** Securely connect RISC-V embedded endpoints.

**Amazon EC2:** Host RISC-V-compatible virtualization layers or cross-compilation servers.

**AWS Lambda:** Manage edge-to-cloud data processing for real-time analytics.

**6.8 AWS Well-Architected Tool and AWS Trusted Advisor**

To ensure best practices in cloud deployment, organizations can leverage:

**AWS Well-Architected Tool:** Assesses key pillars—security, reliability, performance efficiency, cost, and operational excellence—for workloads that integrate RISC-V endpoints.

**AWS Trusted Advisor:** Provides actionable recommendations to optimize resources, lower costs, and enhance security, ensuring a sustainable and secure pipeline between RISC-V devices and the cloud.

## 7. CONCLUSION

## RISC-V stands as a powerful alternative to proprietary ISAs, offering customizable instruction sets, efficient pipelining, and a growing ecosystem of development tools. Its open-source nature removes barriers to innovation and allows for a transparent, auditable hardware foundation. At the same time, a well-defined strategic plan—encompassing security, cost management, cloud integration, and performance metrics—can position organizations to harness RISC-V’s benefits effectively. As markets evolve and demand increasingly specialized computing solutions, the versatility of RISC-V is poised to play a key role in the next generation of embedded devices, consumer electronics, and high-performance data centers.

**9.** **REFERENCES**

Eadline, D. (2024, October 28). *Nvidia, Google to speak about RISC-V use at annual summit*. HPCwire. https://www.hpcwire.com/2024/10/19/nvidia-google-to-speak-about-risc-v-use-at-annual-summit/

Sa, B., Martins, J., & Pinto, S. E. (2022). A first look at RISC-V virtualization from an embedded systems perspective. *IEEE Transactions on Computers*, *71*, 2177–2190. https://doi.org/10.1109/tc.2021.3124320

Sharma, S. (2024, March 6). *RISC-V architecture: A comprehensive guide to the open-source isa*. Wevolver. https://www.wevolver.com/article/risc-v-architecture?utm\_source#introduction

Shilov, A. (2024, October 24). *Nvidia to ship a billion of RISC-V cores in 2024*. Tom’s Hardware. https://www.tomshardware.com/pc-components/gpus/nvidia-to-ship-a-billion-of-risc-v-cores-in-2024

Shukla, S., & Ray, K. C. (2022). A low-overhead reconfigurable RISC-V quad-core processor architecture for fault-tolerant applications. *IEEE Access*, *10*, 44136–44146. https://doi.org/10.1109/access.2022.3169495

*What is RISC-V?*. Wind River. (2024). https://www.windriver.com/solutions/learning/risc-v