هيئة تنمية البحث
والتطوير والابتكار
Research Development
and Innovation Authority

التجمع الوطني لأشباه الموصلات
National Semiconductor Hub

RISC-V: The Open Revolution

RISC-V®

February 2025

RISC-V: The Open Revolution

February 2025

Developed by the Research Development & Innovation Authority (RDIA)

Executive Summary

The RISC-V instruction set architecture (ISA) represents a significant shift in processor design, offering an open-source alternative to proprietary architectures like x86 and ARM. Developed at UC Berkeley in 2010, RISC-V has gained substantial traction due to its flexibility, modularity, and extensibility. Unlike traditional ISAs that require licensing fees and impose restrictions, RISC-V's open nature enables developers to customize processors for specific applications while reducing development costs.

The architecture›s modular design includes a minimal base ISA that can be extended with optional features for different computing needs. This approach allows RISC-V to scale effectively from simple microcontrollers to high-performance computing systems. The base instruction set supports multiple word sizes (32-bit, 64-bit, and 128-bit), while standard extensions enable capabilities such as floating-point arithmetic, atomic operations, and vector processing. This flexibility makes RISC-V suitable for diverse applications, from embedded systems and IoT devices to artificial intelligence and national security infrastructure.

Current implementations of RISC-V demonstrate competitive performance with established architectures, particularly in power-efficient applications. While x86 processors typically outperform RISC-V in single-threaded tasks, RISC-V's simplified design offers advantages in power consumption and thermal management. The architecture has shown particular promise in embedded systems and specialized hardware, where custom extensions can optimize performance for specific workloads.

Despite its advantages, RISC-V faces challenges in market adoption. The relative immaturity of its software ecosystem, limited availability of commercial hardware options, and compatibility issues with legacy software present obstacles to widespread implementation. However, significant industry investment and growing community support are addressing these limitations. Major technology companies, including NVIDIA, Google, and Alibaba, are actively contributing to RISC-V development, while national initiatives in countries like China, India, and Japan are promoting its adoption for strategic independence in semiconductor technology.

Looking ahead, RISC-V is positioned to play an increasingly important role in the future of computing. The architecture›s open nature and customizability make it particularly well-suited for emerging technologies such as AI accelerators, quantum computing interfaces, and specialized hardware for autonomous systems. As the software ecosystem matures and hardware offerings expand, RISC-V is expected to provide stronger competition to established architectures across various computing sectors.

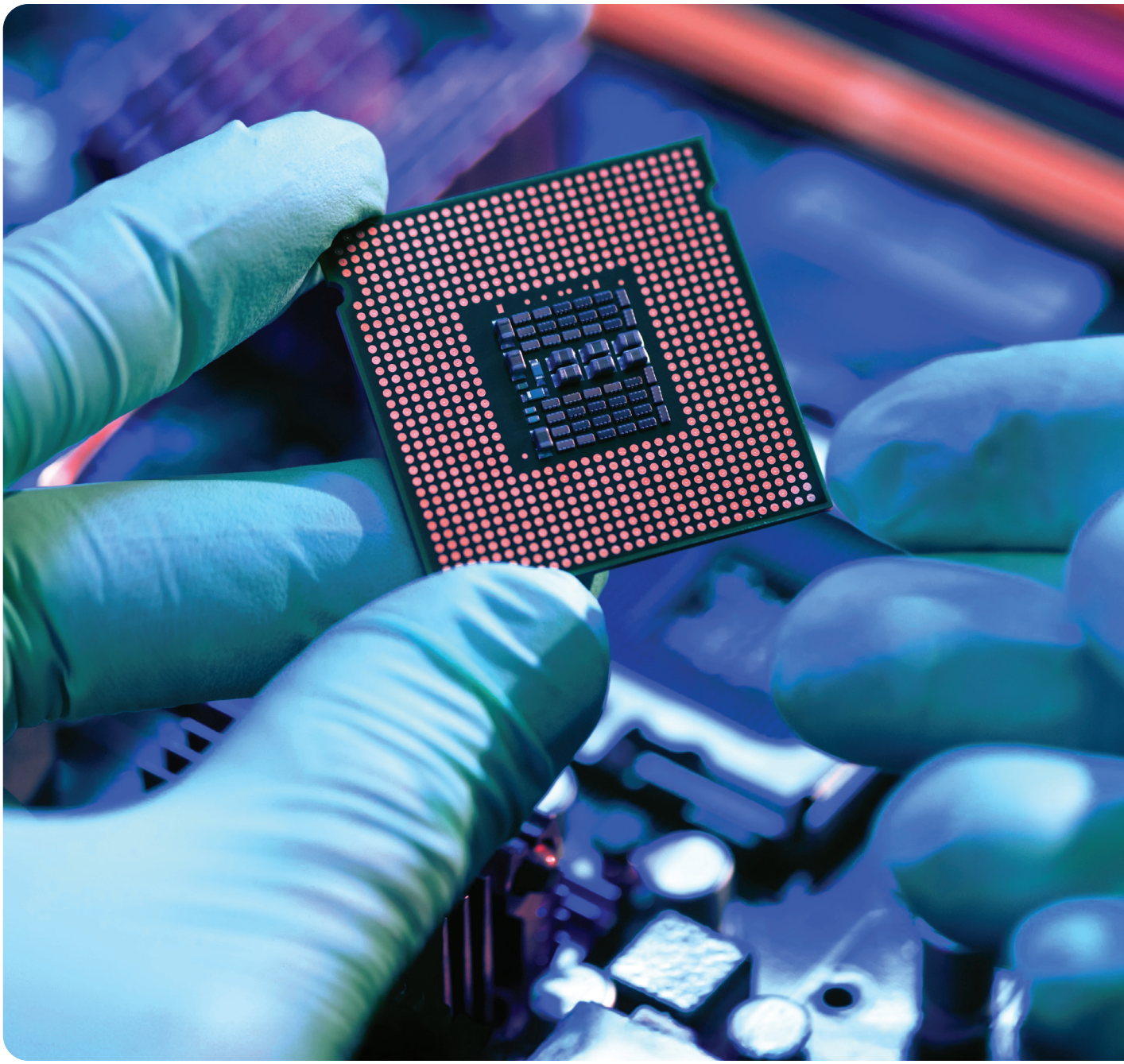


Table of Contents

01 **Introduction to RISC-V** 10
What is RISC-V? 10
Brief History of RISC-V 11

02 **Basics of RISC-V Architecture** 12
What is an Instruction Set Architecture (ISA)? 12
Primary Types of ISA 13

03 **RISC-V ISA and Its Extensions** 14
The RISC-V Base ISA 14
Standard Extensions and Their Applications 15
Specialized Extensions and Their Applications 16

04 **RISC-V Instruction Types and Format** 17

05 **Key Features and Performance Comparison** 19
Key Features of RISC-V 19
Performance Comparison 20

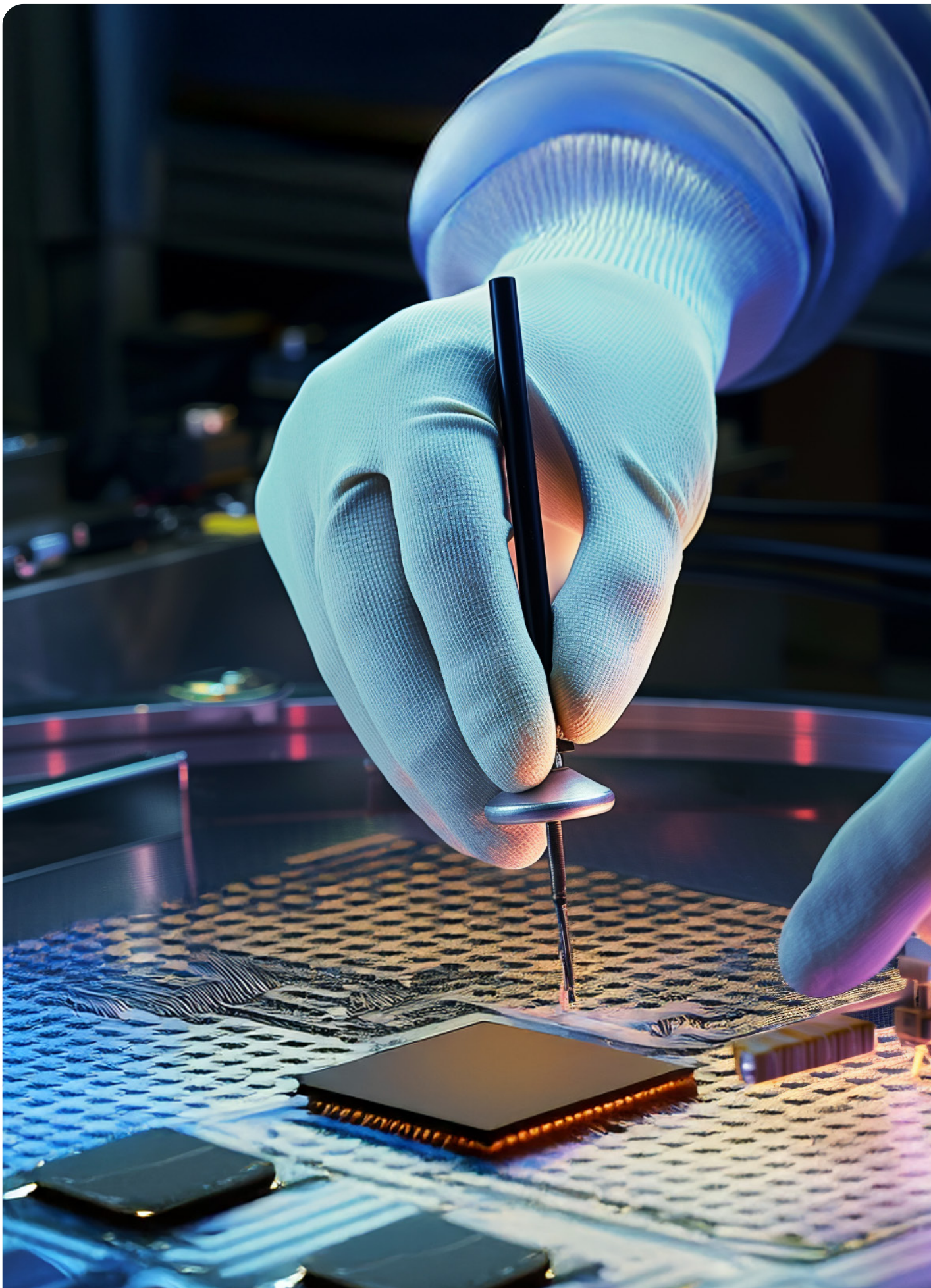
06 **Applications of RISC-V** 22

07 **Advantages of RISC-V** 25

08 **Challenges and Limitations** 28

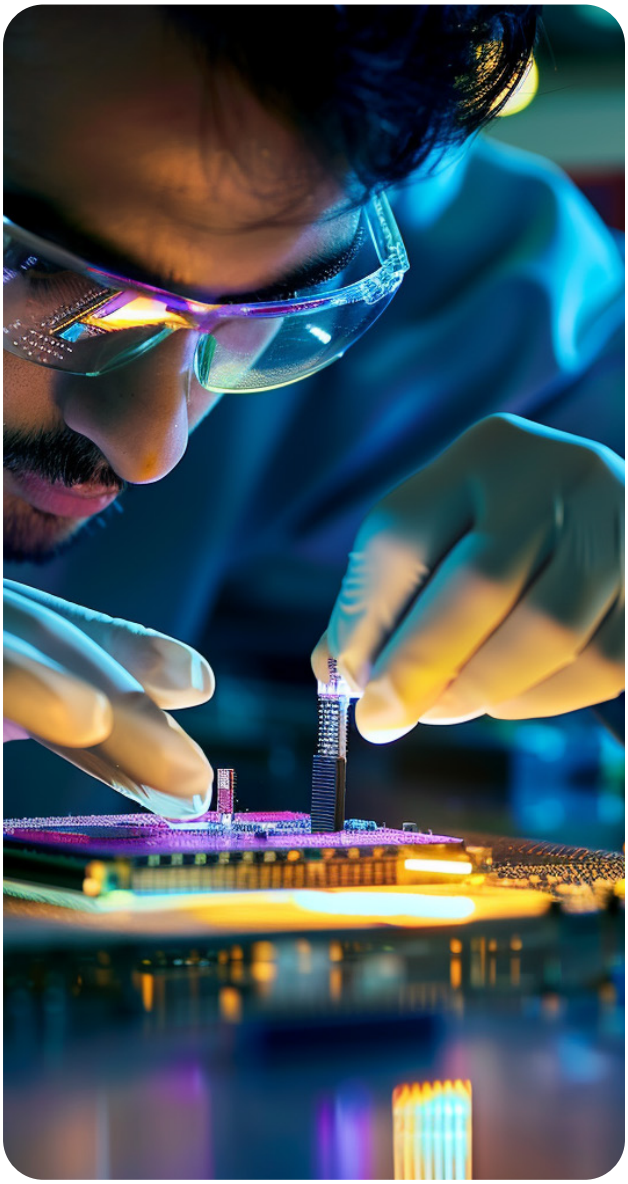
09 **Future Outlook of RISC-V** 30

10 **References** 32



01

Introduction to RISC-V



1.1 What is RISC-V?

RISC-V (pronounced "risk-five") is an open-source instruction set architecture (ISA) that has gained significant traction due to its flexibility, modularity, and extensibility. Unlike proprietary architectures, RISC-V provides open access to its specifications, allowing developers to customize it for specific applications. This approach enables the creation of processors tailored to various use cases, including embedded systems, IoT devices, high-performance computing, and artificial intelligence.

The key advantages of RISC-V include cost-effective processor design, enhanced customization, and robust security solutions. Proprietary ISAs, historically controlled by specific companies, imposed licensing fees and restricted access, limiting innovation and competition. The open nature of RISC-V has addressed these challenges, driving its adoption across industries.

RISC-V's growth is fueled by the demand for customizable and energy-efficient computing solutions while reducing reliance on proprietary ISAs. By offering an open, modular, and extensible ISA, RISC-V fosters innovation in processor design and has the potential to transform the semiconductor industry.

A major advantage of RISC-V is its freedom from licensing fees, which lowers costs for startups and encourages innovation by removing financial barriers. Its modularity allows developers to include only the necessary features, optimizing performance and energy efficiency. Furthermore, the open nature of RISC-V supports research and development, enabling the creation of custom extensions tailored to specific needs. The global adoption of RISC-V has led to a rapidly expanding ecosystem of tools, software, and community support, further accelerating innovation.

1.2 Brief History of RISC-V

RISC-V originated as a research project at the University of California, Berkeley, in 2010. It was developed by a team led by Professor Krste Asanović, along with graduate students Andrew Waterman, Yunsup Lee, and others. The goal was to create a modern ISA that addressed the limitations of existing architectures while remaining simple and extensible. Initially intended for academic research and teaching, RISC-V quickly gained traction for real-world applications.

The motivation behind RISC-V stemmed from several challenges. Existing ISAs, such as x86, were complex and difficult to implement, making them less suitable for education and experimentation. Meanwhile, proprietary ISAs like ARM imposed licensing fees and restrictions that hindered innovation and accessibility. The Berkeley team sought to overcome these issues by designing an ISA that was free, simple, and extensible, making it ideal for both academic and commercial use. Figure 1 shows the timeline of typical processors from academia to industrial applications.

Since its inception, RISC-V has achieved several milestones. In 2014, the first public release of the RISC-V ISA specification marked its transition from a research project to a widely accessible standard. In 2015, the RISC-V Foundation (now RISC-V International) was established to promote and govern the standard. The commercialization of RISC-V began in 2017 with the launch of the first RISC-V-based microcontroller by SiFive. By 2020, global adoption accelerated, with major companies such as Alibaba, NVIDIA, and Western Digital contributing to the ecosystem. Today, RISC-V is a significant force in the semiconductor industry, democratizing access to modern computing architectures and fostering innovation.

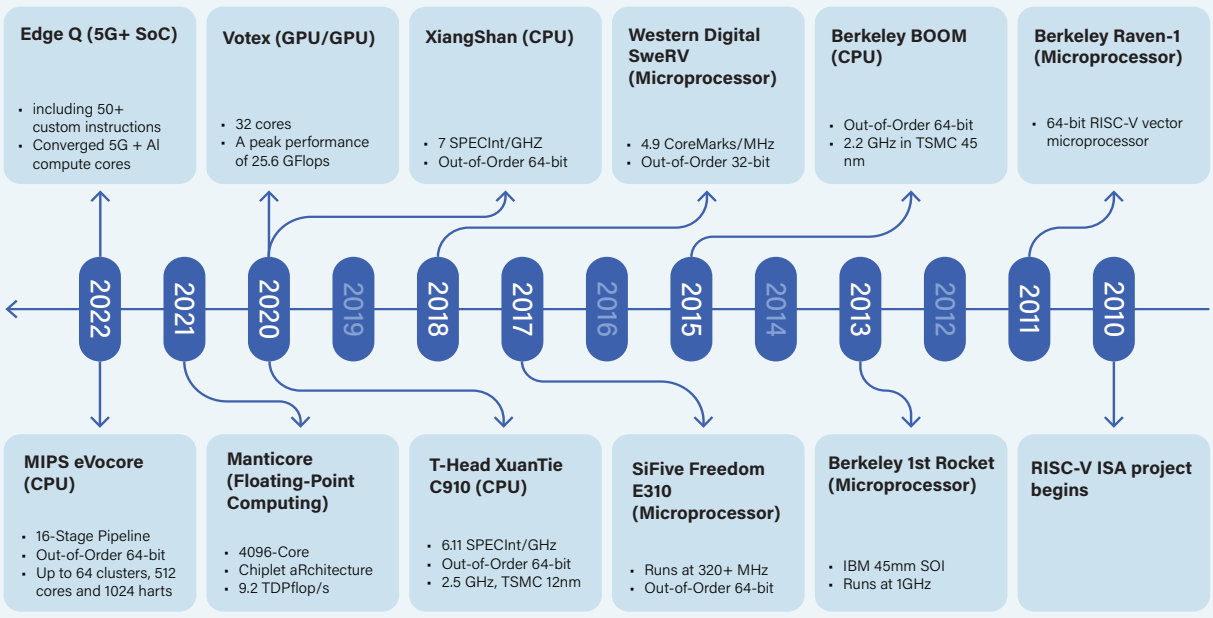


Figure 1: Timeline of typical RISC-V processors.
Credits: IEEE

02 Basics of RISC-V Architecture

The architecture of a processor determines its capabilities, efficiency, and flexibility. At the core of every processor lies its Instruction Set Architecture (ISA), which serves as a blueprint for hardware and software interactions. RISC-V, a modern ISA based on Reduced Instruction Set Computing (RISC) principles, stands out as an open-source design that has disrupted the traditional processor landscape. This chapter explores the fundamentals of RISC-V's architecture, highlighting its unique features, design principles, and comparisons with established ISAs like ARM and x86

2.1 What is an Instruction Set Architecture (ISA)?

An Instruction Set Architecture (ISA) serves as the interface between a computer›s hardware and software. It defines the set of instructions that software uses to command the hardware for tasks such as data processing, storage,

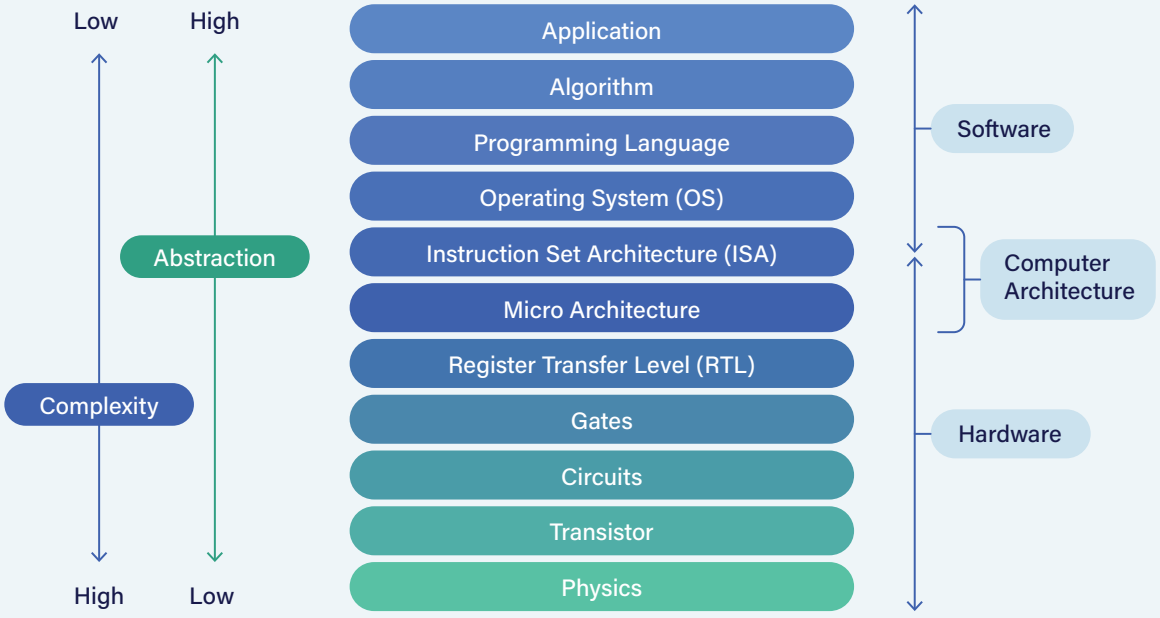


Figure 2: ISA as a Hardware-Software Interface

and transfer. Essentially, the ISA forms the foundation for interaction between a processor's design and the software that runs on it. Figure 2 illustrates how the ISA acts as the bridge between software applications, operating systems, and hardware components.

RISC-V's ISA is unique because it is open and free to use, allowing developers to implement and modify it without licensing restrictions. This freedom has made RISC-V an ideal platform for academic research and commercial product development. Unlike proprietary ISAs that retain legacy complexities, RISC-V provides a clean, modern, and extensible design.

ISA analogy with human language:

- **Instructions** are like words in a language, each with a specific meaning and purpose.
- **Instruction Set** is like vocabulary; a richer vocabulary allows for more complex expressions.
- **Syntax** is like grammar, defining how instructions (words) are structured into meaningful programs (sentences).
- **Semantics** define the meaning of instructions when executed, similar to word meanings in sentences.
- **Microarchitecture** is like handwriting or font style, determining how the same words (instructions) are

2.2 Primary Types of ISA

Instruction Set Architectures can be broadly categorized into several types, each with distinct design philosophies, advantages, and trade-offs.

A. Complex Instruction Set Computing (CISC)

CISC architectures aim to reduce the number of instructions a programmer writes by using complex, variable-length instructions capable of performing multiple operations in a single cycle. This design increases code density, reducing memory usage. However, the complexity of CISC instructions requires sophisticated decoders, increasing power consumption and reducing efficiency in parallel workloads. Despite these challenges, CISC architectures, such as x86 and x8664-, remain dominant in desktop and server computing due to their extensive software ecosystem.

B. Reduced Instruction Set Computing (RISC)

RISC architectures, developed as a response to CISC inefficiencies, focus on simplicity and efficiency. They use fixed-length instructions that perform only one operation at a time, relying on a load/store memory model. This approach enables faster execution and better pipelining, improving power efficiency and performance. Architectures such as ARM, RISC-V, PowerPC, and MIPS follow the RISC philosophy, making them dominant in mobile devices, embedded systems, and high-performance computing.

C. Very Long Instruction Word (VLIW)

VLIW shifts instruction scheduling to the compiler, allowing multiple operations to be executed in parallel within a single instruction word. This simplifies processor design but makes performance highly dependent on compiler efficiency. VLIW is commonly found in AI accelerators, multimedia processing, and GPUs, where workloads are predictable and well-optimized.

D. Explicitly Parallel Instruction Computing (EPIC)

EPIC builds on VLIW principles but introduces runtime optimizations such as predication (eliminating branch delays) and speculation (pre-executing instructions). Intel's Itanium processors were an example of EPIC but failed due to software incompatibility and complex compiler requirements. While EPIC did not gain widespread adoption, its principles continue to influence modern research in parallel execution.

Feature	CISC	RISC	VLIW	EPIC
Instructions Per Cycle (IPC)	Lower due to complex instructions	Higher due to simple instructions	High	Very high due to parallel executiona
Instruction Complexity	Complex	Simple	Complex	Complex
RAM Usage	Lower due to compact instructions	Higher due to more instructions	High due to long instructions	High due to long instructions
Memory	More efficient in terms of instruction storage	Requires more memory due to more instructions	Requires larger memory bandwidth due to long instruction words	Requires high memory bandwidth and registers
Scalability	Limited	High	Limited	High
Power Efficiency	Lower due to complex execution	High due to power optimization	Lower due to parallel execution	Moderate

Table 1: Comparison of the different ISA types

03 RISC-V ISA and Its Extensions

RISC-V is an open-source instruction set architecture (ISA) designed with simplicity, scalability, and flexibility in mind. Unlike traditional ISAs, RISC-V adopts a modular approach, allowing designers to implement only the necessary features for their specific applications. This chapter first explores the foundational aspects of the RISC-V base ISA before diving into its various extensions and their real-world applications.

3.1 The RISC-V Base ISA

The RISC-V base ISA is designed to support different word sizes, ensuring flexibility for various applications. The three primary versions are:

- **RV32 (32-bit)** - Suitable for embedded systems and low-power applications.
- **RV64 (64-bit)** - Optimized for general-purpose computing, servers, and high-performance applications.

- **RV128 (128-bit)** - A future-oriented design aimed at highly advanced computing tasks.

The word size determines the maximum addressable memory and register width, affecting overall processing power and efficiency. RV32 is commonly used in microcontrollers and IoT devices, while RV64 is widely adopted for mainstream computing. RV128, though not yet widely implemented, is intended for workloads requiring extremely large address spaces.

Base Integer Instruction Sets

The RISC-V base ISA is divided into two primary base instruction sets:

- **I (Integer) Base:** The standard base ISA that includes integer arithmetic, control flow, and memory access instructions. It provides a minimal yet complete instruction set for general-purpose computing.
- **E (Embedded) Base:** A reduced version of the I base, designed for resource-constrained embedded applications. The E base uses only 16 general-purpose registers (as opposed to 32 in the I base), reducing hardware complexity and power consumption.

The I base serves as the foundation for most RISC-V implementations, supporting applications ranging from microcontrollers to high-performance computing devices. The E base is particularly useful in environments where silicon area and power efficiency are primary concerns, such as IoT devices and low-power embedded systems.

Relationship Between the Base ISA and Extensions

While the base ISA provides essential functionality for a processor, it is intentionally minimal to ensure simplicity and portability. To accommodate diverse computing needs, RISC-V supports a modular extension system that enhances performance and enables specialized functionalities. These extensions build upon the base ISA without altering its core structure, ensuring backward compatibility while allowing developers to tailor processor capabilities to specific applications.

For example, embedded systems that prioritize energy efficiency may implement the E base ISA along with the C extension to reduce code size. High-performance computing applications, on the other hand, might combine the RV64 base ISA with the F and D extensions for efficient floating-point computations. This modular approach allows RISC-V to scale from low-power embedded processors to powerful data center CPUs.

The ability to selectively enable or disable extensions makes RISC-V adaptable to a wide range of industries, including artificial intelligence, cloud computing, networking, and cryptography. This flexibility is a key advantage over proprietary ISAs that enforce a fixed set of features.

3.2 Standard Extensions and Their Applications

Integer Multiplication and Division (M Extension)

The M extension introduces hardware-based multiplication and division instructions. Without this extension, these operations would need to be executed through software, which is much slower. The M extension is particularly useful in applications requiring frequent mathematical computations, such as digital signal processing, graphics rendering, and cryptographic algorithms.

Atomic Instructions (A Extension)

The A extension enables atomic memory operations, allowing multiple cores to manipulate shared data efficiently. This is crucial for multi-threaded applications, databases, and operating system kernels, where synchronization between different processes is required to maintain data consistency.

Floating-Point Arithmetic (F and D Extensions)

The F extension supports single-precision floating-point operations, while the D extension extends this capability to double-precision arithmetic. These extensions significantly enhance performance in scientific computing, artificial intelligence (AI), financial modeling, and engineering

simulations, where precise numerical computations are necessary.

Compressed Instructions (C Extension)

The C extension introduces 16-bit compressed versions of common 32-bit instructions, reducing code size and improving memory efficiency. This is highly beneficial in embedded systems and Internet of Things (IoT) applications, where memory and power constraints are critical.

3.3 Specialized Extensions and Their Applications

Vector Processing (V Extension)

The V extension adds support for vectorized computations, enabling the processing of multiple data elements simultaneously. This is essential for applications in AI, deep learning, image processing, and high-performance computing, where parallel execution can drastically improve efficiency and throughput.

Bit Manipulation (B Extension)

The B extension provides bit-level operations that are beneficial in cryptography, data compression, and network security. These instructions allow for more efficient handling of low-level data transformations, which are essential in encoding and encryption processes

.Hypervisor Support (H Extension)

The H extension introduces hardware-level virtualization capabilities, enabling multiple operating systems to run on the same hardware through hypervisors. This is critical for cloud computing, data centers, and enterprise servers, where resource sharing and secure execution environments are necessary.

Scalable Privileged Architecture (S and U Extensions)

The S (Supervisor Mode) and U (User Mode) extensions establish a hierarchical privilege model for system security. The user mode restricts access to critical resources, while the supervisor mode is used for system-level operations. These extensions are essential for secure operating systems, preventing unauthorized access and ensuring system stability.

Figure 3 illustrates the RV12, a highly adaptable single-issue, single-core RISC CPU that complies with the RV32I and RV64I architectures, designed specifically for the embedded market. As part of Roa Logic's 3264-/bit CPU family, the RV12 is built on the industry-standard RISC-V instruction set. It employs a Harvard architecture, enabling concurrent access to instruction and data memory. The processor incorporates an optimized folded 6-stage pipeline, which enhances execution efficiency by minimizing stalls and maximizing overlap between execution and memory access phases.

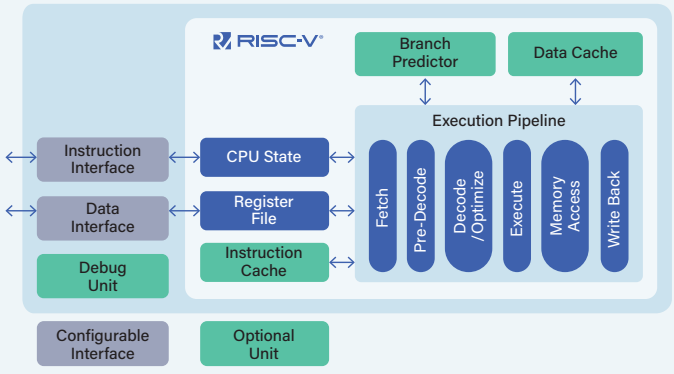


Figure 3: A single-core RV32I CPU based on RISC-V

04

RISC-V Instruction Types and Format

RISC-V follows a Reduced Instruction Set Computing (RISC) philosophy, meaning it has a small set of simple instructions. Instructions in RISC-V are generally 32 bits long (RV32), but there are also 16-bit compressed instructions (RVC) and 64-bit/128-bit versions for wider architectures (RV64, RV128).

Each instruction in RISC-V is classified into different types, with specific formats and fields that define the operation, Figure 4. These fields typically include:

- **opcode:** Determines the type of operation being performed.
- **rd (destination register):** Specifies where the result will be stored.
- **rs1 and rs2 (source registers):** Registers used as input operands.
- **funct3 and funct7:** Additional function codes that specify the operation.
- **immediate values:** Constant values encoded directly into the instruction.

RISC-V instructions are categorized into different types based on their format and functionality.

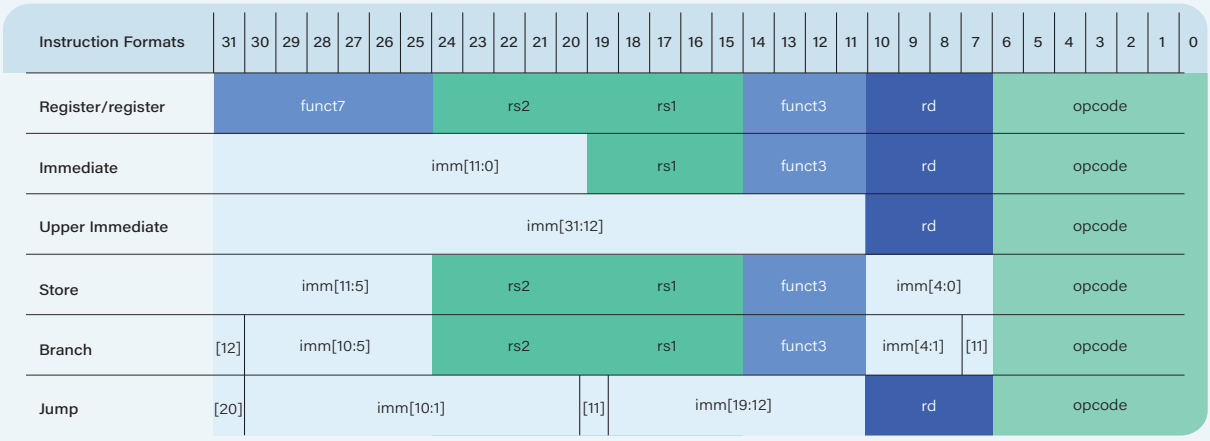


Figure 2: ISA as a Hardware-Software Interface

R-Type Instructions

R-type instructions involve register-to-register operations such as arithmetic and bitwise operations. These instructions do not use immediate values or memory access. The format follows:

```
opcode | rd | funct3 | rs1 | rs2 | funct7
```

Example: Addition (add x5, x6, x7)

```
add x5, x6, x7  # x5 = x6 + x7
```

I-Type Instructions

I-type instructions use immediate values and are often used for arithmetic, logic, and load operations. The format follows:

```
opcode | rd | funct3 | rs1 | imm[11:0]
```

Example: Addition with Immediate (addi x5, x6, 10)

```
addi x5, x6, 10  # x5 = x6 + 10
```

U-Type Instructions

U-type instructions load a 20-bit immediate into the upper bits of a register. These instructions are useful for large constants and address calculations.

```
opcode | rd | imm[31:12]
```

Example: Load Upper Immediate (lui x5, 0x12345)

```
lui x5, 0x12345  # Load upper immediate 0x12345000 into x5
```

S-Type Instructions

S-type instructions store data from registers into memory. These instructions are crucial for handling data storage in memory.

```
opcode | imm[11:5] | rs2 | rs1 | funct3 | imm[4:0]
```

Example: Store Word (sw x5, 0(x6))

```
sw x5, 0(x6)  # Store word from x5 to memory address in x6
```

B-Type Instructions

B-type instructions are used for conditional branching and allow for control flow in programs. These instructions compare register values and branch to a label if a condition is met.

```
opcode | imm[12] | imm[10:5] | rs2 | rs1 | funct3 | imm[4:1] | imm[11]
```

Example: Branch if Equal (beq x5, x6, label)

```
beq x5, x6, label  # Branch to label if x5 == x6
```

J-Type Instructions

J-type instructions are used for jump operations and allow for function calls and control flow redirection.

```
opcode | rd | imm[20] | imm[10:1] | imm[11] | imm[19:12]
```

Example: Jump and Link (jal x5, label)

```
jal x5, label  # Jump to label and store return address in x5
```

05

Key Features and Performance Comparison

5.1 Key Features of RISC-V

Simplicity

One of the defining characteristics of RISC-V is its simplicity. The base ISA includes only the essential instructions required for general-purpose computing, making it easier to implement and understand. This simplicity leads to reduced hardware complexity, enabling efficient designs that are both cost-effective and power-efficient. Compared to Complex Instruction Set Computing (CISC) architectures like x86, which include thousands of instructions, RISC-V follows a streamlined approach with a smaller number of well-optimized instructions.

Modularity

Another key feature of RISC-V is its modularity. The base ISA can be extended with optional standard extensions to meet the needs of specific applications. For example, extensions for floating-point arithmetic, vector processing, atomic operations, and compressed instructions are available. This modular design allows developers to tailor RISC-V processors to a wide range of applications, from low-power embedded devices to high-performance computing systems.

Scalability

RISC-V's scalability makes it suitable for a broad spectrum of devices, ranging from simple microcontrollers to advanced supercomputers. The architecture›s clean and extensible design allows it to adapt to different performance and power requirements. For instance, lightweight implementations can be used for energy-efficient IoT devices, while advanced implementations with multiple extensions can support high-performance workloads in data centers.

RISC-V, as an open-source instruction set architecture, presents a compelling alternative to established architectures such as x86 and ARM. Each of these architectures has distinct design philosophies, performance characteristics, and market applications. Understanding the comparative strengths and

weaknesses of RISC-V requires a detailed analysis of execution efficiency, power consumption, scalability, and software ecosystem maturity. This chapter examines how RISC-V stacks up against x86 and ARM in various computing environments, including embedded systems, desktop computing, and high-performance computing (HPC).

5.2 Performance Comparison

Computational Performance Comparison

The performance of a processor is often measured by its instruction throughput, branch prediction accuracy, and efficiency in handling memory operations. x86 processors, developed by Intel and AMD, utilize a complex instruction set computing (CISC) approach, allowing them to execute multiple operations within a single instruction. This leads to higher instruction density but also requires sophisticated out-of-order execution engines to maintain efficiency. ARM and RISC-V, on the other hand, follow a reduced instruction set computing (RISC) model, which simplifies the decoding and execution pipeline while relying on compiler optimizations to extract parallelism.

Benchmarks such as SPEC CPU and CoreMark indicate that high-end x86 processors outperform RISC-V and ARM counterparts in single-threaded performance due to aggressive speculative execution and deeper pipelines. However, ARM has made significant strides in efficiency with designs like Apple's M-series chips, which utilize high IPC (instructions per cycle) cores and tight software integration. RISC-V, being relatively new, is still maturing in high-end computing but has shown promise with out-of-order designs like BOOM, which is comparable to mid-tier ARM Cortex-A cores.

Power Efficiency and Thermal Performance

One of the key advantages of ARM and RISC-V architectures is their emphasis on power efficiency. ARM has long been dominant in mobile and embedded applications due to its ability to scale power consumption dynamically based on workload requirements. RISC-V follows a similar approach, but its open-source nature allows designers to further tailor implementations for power-sensitive applications. Implementations such as the SiFive U740 demonstrate competitive efficiency when compared to ARM Cortex-A55 cores, but higher-end power-efficient designs are still emerging.

x86 processors, despite optimizations in power management, consume significantly more power per instruction due to their complex microarchitectural features. This makes x86 less suitable for low-power applications, particularly in battery-powered devices. RISC-V, when optimized correctly, has the potential to match or exceed ARM's efficiency, particularly in custom implementations where unneeded features are stripped away.

Scalability and Use in High-Performance Computing

Scalability is a crucial factor in determining an architecture›s suitability for different computing environments. x86 has maintained dominance in high-performance computing due to extensive software ecosystem support, including mature compilers and parallel computing libraries. ARM, with its entry into the server market through designs like AWS Graviton and Fujitsu's A64FX, has demonstrated that RISC-V based architectures can successfully scale to enterprise-grade workloads.

RISC-V is still in the early stages of scaling to HPC workloads. While the architecture supports multi-core and vector processing extensions, industry adoption is limited by the need for extensive software optimization. The lack of a unified standard for high-performance RISC-V implementations has led to fragmentation, with different vendors developing their own custom cores. However, projects such as RISC-V's vector extensions and modular accelerators suggest that future designs could close the performance gap with x86 and ARM in HPC applications.

Software Ecosystem and Compatibility

The strength of an architecture is not only determined by hardware capabilities but also by the availability of a robust software ecosystem. x86 has a decades-long advantage with widespread software compatibility, making it the

default choice for many commercial applications. ARM, while historically focused on mobile and embedded devices, has gained increasing support in desktop and server markets through initiatives like Apple Silicon and Windows on ARM.

RISC-V, being newer, faces challenges in software maturity. While Linux has adopted RISC-V as a supported architecture, many applications still require significant porting efforts. Compiler support, particularly in LLVM and GCC, is improving, but optimizations remain behind those available for x86 and ARM. Additionally, the lack of a standardized firmware and boot process for RISC-V adds another hurdle for mainstream adoption.

The x86, ARM, and RISC-V architectures serve different purposes and market segments. The table below highlights key differences:

Feature	x86 (CISC)	ARM (RISC)	RISC-V (RISC)
Compatibility	Strong (legacy support)	Moderate (some old features removed)	Flexible (allows extensions)
Power Efficiency	Low (higher power use)	High (optimized for battery life)	High (simple and customizable)
Performance	High (advanced optimizations)	High (efficient and low power)	Varies (depends on customization)
Market Use	PCs, servers, supercomputers	Mobile devices, embedded systems, cloud servers	Emerging (AI, research, specialized hardware)
Customization	Proprietary (closed)	Licensed (some customization)	Open-source (highly customizable)

Table 2: Comparison of x86, ARM, and RISC-V

06

Applications of RISC-V

RISC-V has rapidly gained recognition as a versatile and transformative instruction set architecture (ISA) capable of addressing a wide range of computing needs. Its open-source nature, modularity, and energy efficiency have made it a preferred choice in various fields, from embedded systems and high-performance computing to education and custom hardware design. By enabling innovation and reducing barriers to entry, RISC-V has become a cornerstone for technological advancements across industries. This chapter explores the key applications of RISC-V, highlighting its adaptability and potential to drive innovation in diverse domains.

National Security and Strategic Applications

As nations prioritize technological sovereignty, RISC-V is emerging as a key enabler of secure and independent computing. Unlike proprietary ISAs such as x86 and ARM, RISC-V's open-source model allows governments and industries to develop customized processors free from external licensing restrictions, supply chain vulnerabilities, and geopolitical risks.

Countries like China, India, and Russia are actively investing in RISC-V to strengthen national semiconductor capabilities. China has integrated RISC-V into state-backed chip initiatives, while India's Shakti and VEGA projects focus on secure, domestically designed processors for defense and government infrastructure. These efforts aim to reduce reliance on foreign-controlled architectures, ensuring greater control over critical computing systems.

RISC-V's transparency and flexibility make it particularly attractive for national security applications. Military, aerospace, and defense sectors benefit from the ability to audit and customize hardware, mitigating risks of backdoors and unauthorized access. Custom implementations can integrate cryptographic extensions, secure boot mechanisms, and real-time processing capabilities tailored for mission-critical environments.

As global interest in secure and sovereign computing grows, RISC-V is poised to play a central role in national security strategies. Its adoption across defense, government, and critical infrastructure sectors reflects a broader shift toward open, customizable, and geopolitically resilient semiconductor solutions.

Embedded Systems

RISC-V has become a popular choice in the world of embedded systems due to its simplicity, flexibility, and energy efficiency. Microcontrollers and Internet of Things (IoT) devices are among the most prominent use cases. The lightweight and modular nature of the RISC-V architecture allows manufacturers to design processors optimized for low-power and resource-constrained environments, which are critical for embedded applications.

In microcontrollers, RISC-V enables the integration of the necessary instructions and extensions while avoiding unnecessary overhead, resulting in cost-effective and efficient designs. For IoT devices, which often operate on limited battery power, the ability to implement custom extensions allows developers to enhance energy efficiency and tailor processors for specific tasks such as sensor data processing, communication, and encryption. Companies like SiFive and Espressif have already introduced RISC-V-based microcontrollers, demonstrating the architecture's potential in these domains.

High-Performance Computing

RISC-V is increasingly being adopted in high-performance computing (HPC) environments, including GPUs, supercomputers, and accelerators. Its modularity and support for vector extensions make it an ideal choice for data-parallel workloads such as scientific simulations, artificial intelligence, and machine learning.

Vector processing, enabled by the RISC-V Vector (V) extension, allows processors to perform operations on large datasets efficiently, which is essential for modern HPC applications. RISC-V's open nature also encourages collaboration among research institutions and industry leaders to develop custom accelerators tailored to specific workloads. For instance, supercomputers leveraging RISC-V can be designed with domain-specific extensions for optimized performance in fields like climate modeling, genomics, and quantum simulations.

Additionally, RISC-V's adaptability makes it suitable for use in GPUs, where the architecture can be customized for high-throughput parallelism, rendering, and compute-intensive tasks. The ability to create bespoke hardware solutions for HPC applications has made RISC-V a strong contender in this competitive space.

Education

RISC-V's simplicity and accessibility have made it an invaluable resource in education, particularly in the fields of computer architecture, engineering, and programming. Unlike proprietary ISAs, RISC-V is freely available, allowing students and researchers to study, experiment with, and implement the architecture without legal or financial barriers.

Educational institutions around the world are adopting RISC-V as a teaching tool for courses in processor design, operating systems, and embedded systems. Its clean and minimalistic design makes it easier for students to grasp the fundamental concepts of instruction sets and hardware-software interaction. Additionally, RISC-V provides an excellent platform for academic research, enabling the development of novel extensions, experimental hardware, and innovative software solutions.

Open-source implementations and simulation tools, such as Spike and QEMU, further enhance its utility in education. These tools allow students to design and test RISC-V-based systems without the need for expensive hardware, making the learning process both accessible and practical.

Custom Hardware Solutions

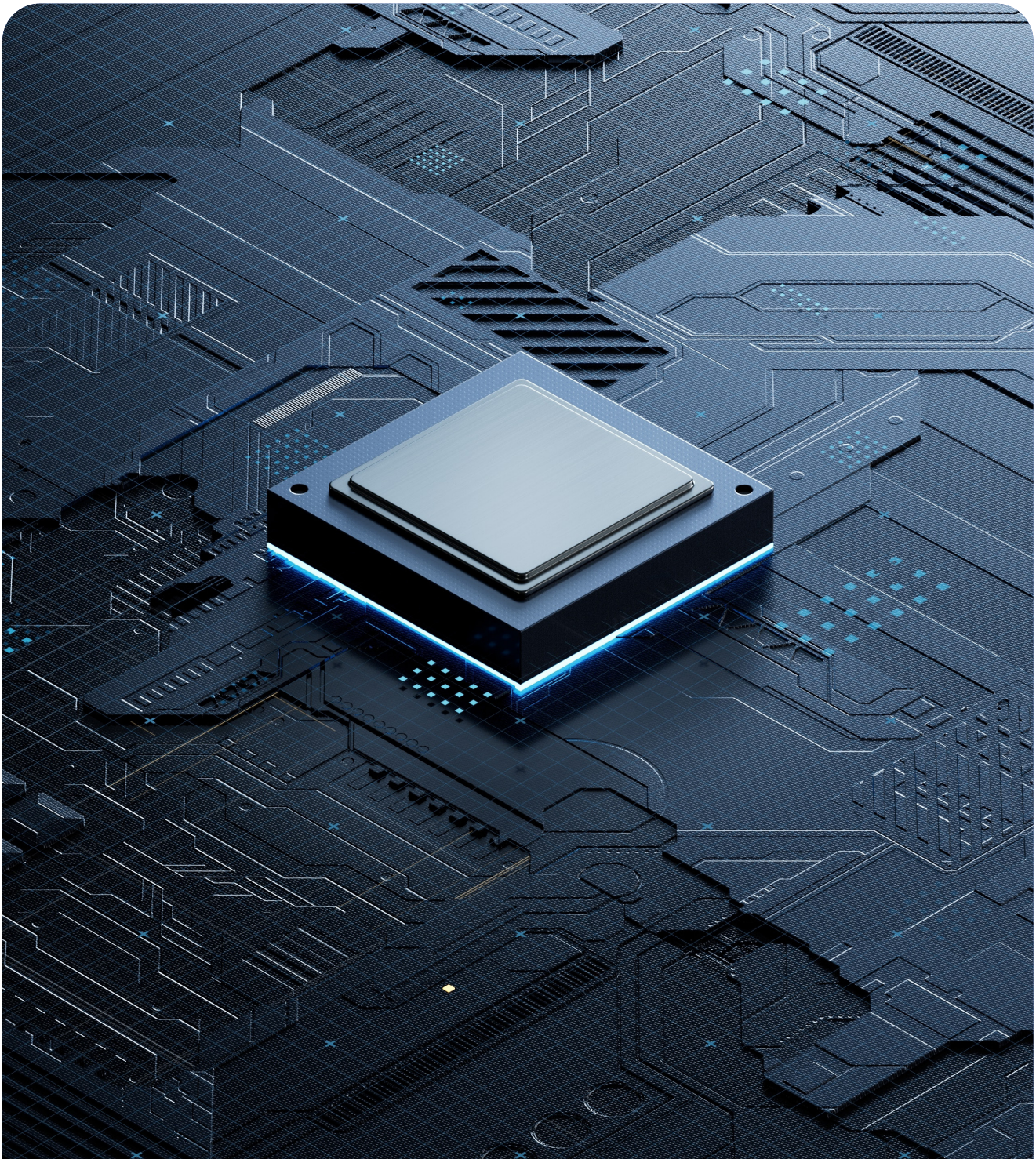
One of the most significant strengths of RISC-V is its adaptability for creating custom hardware solutions. The architecture's modular design and support for custom extensions empower developers to design processors tailored to specific applications and industries. This flexibility is particularly valuable in domains such as automotive, healthcare, telecommunications, and finance.

For example, in the automotive sector, RISC-V-based processors can be designed to handle real-time sensor data processing, autonomous driving algorithms, and advanced safety systems. In healthcare, custom RISC-V implementations can power medical devices such as

imaging equipment, portable diagnostic tools, and wearable health monitors. Similarly, telecommunications companies can use RISC-V to develop efficient baseband processors and network infrastructure optimized for 5G and beyond.

By enabling domain-specific architectures, RISC-V facilitates the creation of hardware that meets the unique demands of various industries, driving innovation and improving efficiency across sectors. This ability to customize hardware solutions has been a game-changer, positioning RISC-V as a versatile and future-proof technology for diverse applications.

RISC-V's wide range of applications, from embedded systems and high-performance computing to education and custom hardware solutions, highlights its adaptability and potential to revolutionize the semiconductor industry. Its open nature, simplicity, and flexibility make it a powerful tool for addressing the challenges of modern computing and fostering innovation across disciplines.



07

Advantages of RISC-V

RISC-V has emerged as a groundbreaking instruction set architecture (ISA) that redefines how processors are designed and deployed. Its open-source nature, coupled with its flexibility and scalability, has made it a preferred choice across industries ranging from embedded systems to high-performance computing. The architecture›s design philosophy emphasizes simplicity and adaptability, which not only reduces costs but also fosters innovation. This chapter explores the key advantages of RISC-V, including its open-source model, flexibility in tailoring to specific applications, robust ecosystem support, and inherent energy efficiency, all of which contribute to its growing adoption worldwide.

Open-Source Nature

One of the most significant advantages of RISC-V is its open-source nature. Unlike proprietary instruction set architectures (ISAs) such as ARM and x86, RISC-V is entirely free from licensing fees and restrictions. This openness lowers the barriers to entry for developers and organizations of all sizes, fostering innovation and collaboration across the global technology landscape. Without the need to pay royalties or comply with restrictive licensing agreements, smaller companies and startups can develop custom processors and systems tailored to their specific needs. Furthermore, the open nature of RISC-V enables researchers and academic institutions to experiment and advance the architecture without the constraints imposed by proprietary ISAs.

The open-source model also ensures transparency in the development process. Developers can review, modify, and extend the ISA to suit their requirements, leading to faster innovation cycles and broader adoption across industries. This openness has contributed to the rapid growth of RISC-V's global ecosystem, making it a disruptive force in the semiconductor industry.

Flexibility

RISC-V's modular and extensible design provides unparalleled flexibility compared to traditional ISAs. The base ISA is minimalistic, containing only the essential instructions needed for general-purpose computing. Additional functionality can be added through optional standard extensions or custom extensions tailored to

specific applications. This modularity allows developers to design processors optimized for various use cases, ranging from lightweight microcontrollers for embedded systems to high-performance computing platforms for artificial intelligence and scientific research.

The ability to create custom extensions is particularly beneficial for industries with unique requirements. For example, companies developing hardware for autonomous vehicles can implement custom instructions for real-time sensor data processing and machine learning inference. Similarly, IoT device manufacturers can optimize processors for ultra-low power consumption by excluding unnecessary features. This adaptability ensures that RISC-V can cater to a diverse range of applications, making it a versatile and future-proof architecture.

Ecosystem Support

The RISC-V ecosystem has grown rapidly, offering a wide range of toolchains, compilers, software libraries, and development platforms, Figure 5. Open-source tools such as GCC and LLVM provide robust support for RISC-V, enabling developers to compile and debug applications with ease. Additionally, a growing number of integrated development environments (IDEs) and hardware development kits (HDKs) make it easier than ever to design and test RISC-V-based systems.

Organizations such as RISC-V International and the global developer community have played a key role in expanding the ecosystem. Collaborative efforts have led to the development of software stacks, operating system support, and middleware tailored for RISC-V platforms. Major technology companies, including NVIDIA, SiFive, and Alibaba, have contributed to this ecosystem, further accelerating its adoption and maturity.

The availability of open-source implementations and development resources reduces the time and cost associated with creating RISC-V-based solutions. This robust ecosystem ensures that developers have the tools and support needed to bring their ideas to life, fostering innovation across industries.

Figure 2: ISA as a Hardware-Software Interface

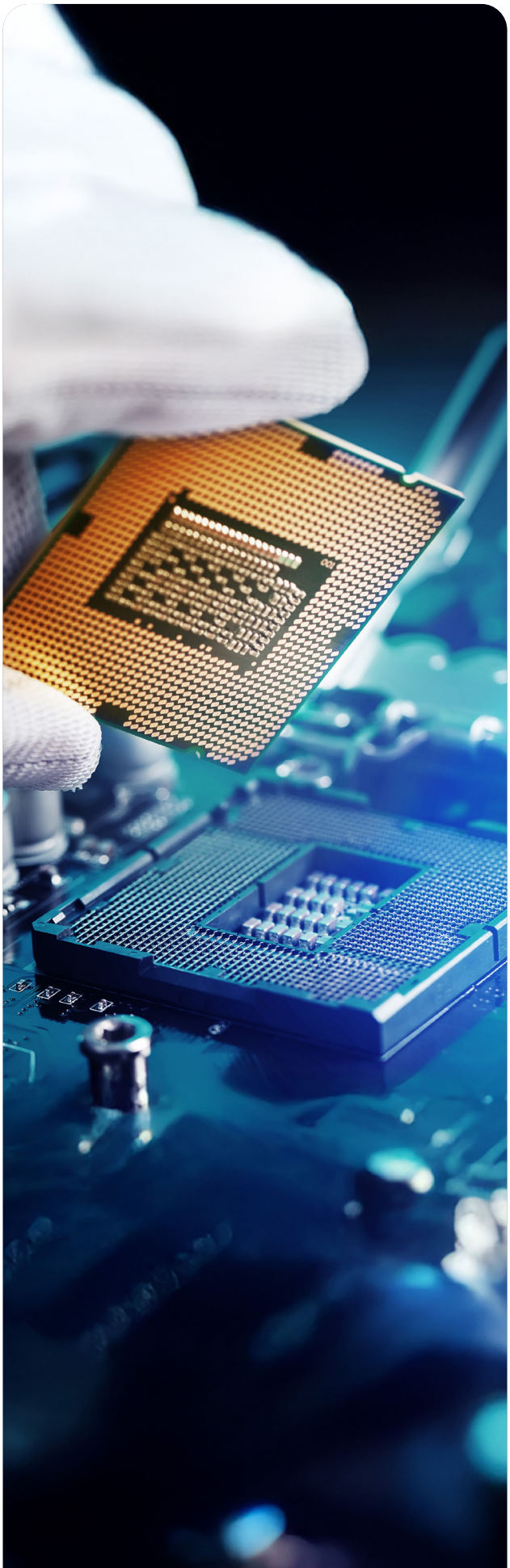
Energy Efficiency

RISC-V's simplified design makes it inherently energy-efficient, making it well-suited for low-power applications. The minimalistic base ISA reduces the complexity of hardware implementation, leading to lower power consumption compared to more complex architectures. Additionally, RISC-V's modular design allows developers to exclude unnecessary features, further optimizing energy efficiency for specific use cases.

This energy efficiency is particularly advantageous for battery-powered devices, such as IoT sensors, wearables, and mobile devices, where power consumption directly impacts the product's usability and lifespan. For example, RISC-V processors with the Compressed (C) extension can achieve significant code size reductions, minimizing memory access and reducing power usage. Moreover, custom extensions tailored for low-power modes and adaptive power management enable further optimizations for energy-constrained environments.

The combination of energy efficiency and scalability positions RISC-V as an ideal choice for a wide range of applications, from resource-limited embedded systems to large-scale data centers aiming to reduce operational costs and environmental impact.

RISC-V's open-source nature, flexibility, robust ecosystem, and energy efficiency collectively make it a transformative force in the computing world. These advantages empower developers and organizations to innovate freely, creating solutions that address the challenges of today and the opportunities of tomorrow.



08 Challenges and Limitations

While RISC-V has garnered significant attention for its open-source nature, flexibility, and scalability, it is not without its challenges and limitations. As a relatively young architecture, RISC-V faces hurdles in competing with established players like ARM and x86, which benefit from decades of development and ecosystem growth. This chapter explores the key challenges that RISC-V must address, including its relative maturity, the development of its ecosystem, and compatibility issues with legacy software.

Maturity

RISC-V is a relatively new instruction set architecture, having been introduced in 2010. In contrast, ARM and x86 have been evolving for decades, accumulating a wealth of optimizations, industry adoption, and software support. This difference in maturity places RISC-V at a disadvantage in markets where tried-and-tested solutions are preferred. For instance, enterprise-grade processors and servers often require proven reliability and long-term support, areas where RISC-V still has limited track records.

The lack of historical benchmarks and extensive production deployment also means that many developers and hardware manufacturers are cautious about adopting RISC-V for critical applications. While its open-source nature fosters innovation, it also requires significant investment in development and testing to reach the level of refinement seen in more established architectures.

Ecosystem

One of the key challenges for RISC-V is the ongoing development of its ecosystem. While the architecture has made remarkable progress in recent years, its ecosystem is still nascent compared to the well-established infrastructure surrounding ARM and x86. These established architectures benefit from extensive software libraries, toolchains, and developer communities, making them highly attractive for a wide range of applications.

RISC-V's ecosystem, although growing, currently lacks the breadth and depth of support found in its competitors. For instance, while tools like GCC and LLVM support

RISC-V, the availability of optimized compilers, debugging tools, and integrated development environments (IDEs) is still limited in comparison. Additionally, the software stack for certain specialized domains, such as gaming and high-end computing, remains underdeveloped.

Another aspect of ecosystem development is the availability of hardware. ARM and x86 dominate the market with a wide array of off-the-shelf processors and hardware platforms, making it easy for developers to prototype and deploy solutions. RISC-V, on the other hand, has fewer commercially available options, requiring developers to rely on niche providers or develop custom hardware solutions, which can be cost-prohibitive.

Compatibility

Compatibility with legacy software is another significant limitation of RISC-V. Existing ecosystems, particularly those of ARM and x86, have decades of legacy software that has been optimized for their respective architectures. Porting this software to RISC-V is a non-trivial task, often requiring significant time and resources to adapt and optimize the code for the new architecture.

This compatibility issue is particularly pronounced in enterprise environments, where legacy systems and applications are critical for day-to-day operations. Organizations may be hesitant to migrate to RISC-V if it means rewriting or re-optimizing large portions of their software stack. Additionally, proprietary software vendors may be slow to support RISC-V, further limiting its adoption in certain sectors.

To address these challenges, the RISC-V community is actively working on tools and frameworks to simplify software porting and compatibility. Emulators, cross-compilers, and middleware solutions are being developed to bridge the gap and facilitate smoother transitions for developers. However, achieving full compatibility with the extensive software base of established architectures will take time and sustained effort.

RISC-V's challenges, including its relative maturity, developing ecosystem, and compatibility issues, highlight the hurdles it must overcome to achieve widespread adoption. While these limitations are significant, they also present opportunities for growth and innovation. With continued investment and collaboration from the global community, RISC-V is well-positioned to address these challenges and solidify its place as a leading architecture in the future of computing.

09

Future Outlook of RISC-V

Despite its current limitations, RISC-V's modularity and open-source nature offer unique advantages over x86 and ARM. The ability to develop custom instructions and specialized accelerators allows companies to optimize processors for specific workloads without relying on proprietary licensing models. As software support continues to improve and industry adoption increases, RISC-V is expected to become a viable competitor in embedded systems, AI acceleration, and even general-purpose computing.

x86 will likely maintain its dominance in traditional computing markets, but its complexity and power consumption constraints make it less suited for emerging workloads that require energy efficiency. ARM, with its strong ecosystem and continued growth in server and high-performance computing markets, remains RISC-V's primary competitor in the RISC space. However, the rapid innovation within the RISC-V community suggests that it could disrupt both ARM and x86 markets in the coming decade.

RISC-V, x86, and ARM each have unique strengths and trade-offs that make them suitable for different computing environments. x86 excels in raw performance and software compatibility but suffers from power inefficiency. ARM balances efficiency with performance and has made inroads into both consumer and enterprise markets. RISC-V, while still developing, offers unparalleled flexibility and customization potential, making it an attractive alternative for specialized applications. As RISC-V continues to evolve, it is likely to become a more prominent force in the semiconductor industry, challenging both x86 and ARM in various sectors of computing.

As RISC-V continues to gain momentum across industries, its potential to reshape the semiconductor landscape grows stronger. The combination of its open-source nature, modular architecture, and cost efficiency positions RISC-V as a compelling alternative to proprietary instruction set architectures. The coming years are expected to bring significant advancements in adoption, ecosystem maturity, and technological evolution, paving the way for new innovations in computing.

Expanding Industry Adoption

The growing adoption of RISC-V is being driven by its ability to meet the demands of diverse industries. Companies in artificial intelligence (AI), high-performance computing (HPC), embedded systems, and consumer electronics are actively exploring RISC-V-based processors to reduce costs and enhance customization. Governments and research institutions are also investing in RISC-V as a way to develop independent, sovereign chip designs, reducing reliance on proprietary technologies.

Additionally, emerging markets and developing economies are increasingly looking toward RISC-V to build affordable and energy-efficient computing solutions. With major semiconductor firms such as SiFive, Alibaba, and NVIDIA actively investing in RISC-V research and development, its adoption across commercial applications is expected to accelerate. In January 2025, SpacemiT announced the VitalStone V100, a server processor featuring up to 64 RISC-V cores, aimed at high-performance computing and AI applications. Similarly, DeepComputing and Framework Computer introduced a RISC-V mainboard for the Framework Laptop 13, further demonstrating the adaptability of RISC-V in consumer devices.

In November 2024, Japan partnered with U.S. chip startup Tenstorrent to train up to 200 Japanese chip designers over five years. This $50 million contract is part of Japan›s strategy to revitalize its semiconductor industry. The program focuses on educating engineers in advanced AI chip design, enabling them to create their own RISC-V designs upon returning to Japan.

Strengthening the Software Ecosystem

One of the primary challenges for RISC-V has been the relative immaturity of its software ecosystem. However, significant progress is being made in compiler optimizations, operating system support, and developer tools. Efforts from organizations like RISC-V International, along with contributions from major technology firms, are helping expand the availability of software libraries and frameworks for RISC-V architectures.

Improvements in Linux distributions, real-time operating systems (RTOS), and cross-platform development tools will further lower the barriers to entry for developers. The launch of the RISC-V Software Ecosystem (RISE) project in May 2023, backed by Google and NVIDIA, has accelerated software development for RISC-V in consumer electronics, data centers, and automotive applications. Additionally, increasing support from cloud service providers and enterprise software vendors will make RISC-V a more viable option for large-scale deployment.

Performance and Innovation

RISC-V is continuously evolving to meet the growing demands of advanced computing workloads. The development of new extensions, including the Vector (V) extension for parallel processing, the Cryptographic (K) extension for secure transactions, and the Hypervisor (H) extension for virtualization, will enable RISC-V to compete with traditional architectures in high-performance computing, AI, and cloud infrastructure.

Future advancements in custom silicon design will further enhance RISC-V's performance efficiency. Companies are increasingly leveraging RISC-V for domain-specific architectures, allowing them to optimize processors for specialized applications such as machine learning accelerators, autonomous vehicles, and quantum computing research. The emergence of RISC-V-based AI accelerators and data center processors highlights the growing confidence in its ability to handle cutting-edge workloads.

In January 2025, SpacemiT announced its VitalStone V100 processor, which is designed for AI workloads, featuring a scalable architecture with up to 64 cores. This development showcases RISC-V's capability to enter the high-performance computing market, competing with proprietary alternatives. Additionally, RISC-V's presence in consumer electronics has expanded with the introduction of the Framework Laptop 13 mainboard, allowing users to experience an open-source processor ecosystem firsthand.

10 References

References

- RISC-V Foundation: RISC-V Tech Hub.
- Wevolver. " RISC-V Architecture: A Comprehensive Guide to the Open-Source ISA"
- Cui, Enfang, Tianzheng Li, and Qian Wei. "Risc-v instruction set architecture extensions: A survey." IEEE (2023)
- Google, NVIDIA, and RISC-V International Launch RISE Project. (2023).
- RISC-V's Hardware-Software Contract and Compatibility Issues. (2024).
- Waterman, Andrew, Yunsup Lee, Rimas Avizienis, David A. Patterson, and Krste Asanovic. "The RISC-V instruction set manual volume II: Privileged architecture version 1.7." EECS (2015).
- Sirin Software, "Inside RISC-V microarchitecture".
- A. Waterman, "Design of the RISC-V Instruction Set Architecture", University of California at Berkeley (2016).
- B. W. Mezger, D. A. Santos, L. Dilillo, C. A. Zeferino and D. R. Melo, "A Survey of the RISC-V Architecture Software Support," IEEE (2022)
- "RISC-V Market Trends And Predictions Till 2030", The SHD Group (2024)

@RDIA | 2025

هيـــأة تنـمـــيـة البـحـــث والتـطـويـر والابـتـــكار | www.rdia.gov.sa | The Research Development and Innovation Authority

4019 طريق الملك عبدالله - حي الرائد - الرياض - المملكة العربية السعودية 4019 King Abdullah Road - Al-Raed District - Riyadh - KSA