ISSN 2959-6157

The Ascendancy of RISC-V: An In-Depth Review of Its Open-Source Potential and Future Applications

Pengrui Niu[1]

[1]Department of Chips, XianJiaotong Liverpool University, Taicang, China
*Corresponding author: Pengrui.Niu23@student.xjtlu.edu.cn

Abstract:

Currently, the rapid advancement of RISC-V and the consequent changes in industry and market landscapes are capturing significant attention from various stakeholders. In recent years, there has been a notable increase in the number of companies and developers dedicating their efforts to researching and developing RISC-V technologies. In light of this trend, this paper aims to examine the global development of RISC-V, providing a comprehensive analysis of its progression on an international scale. It will compare RISC-V with other prominent instruction set architectures such as x86 and ARM, evaluating the relative merits and drawbacks of each. The paper will systematically organize and discuss relevant data to highlight RISC-V's advantages and limitations. Furthermore, it will present an overview of existing RISC-V products and their applications across different sectors. Lastly, the paper will explore the future market potential of RISC-V, assessing how it might influence the industries related to chips and related markets in the coming years.

Keywords: RISC-V Architecture, Open-Source Instruction Set, Embedded Systems and IoT.

1. Introduction

This section will first introduce the concepts of instruction sets and architectures, and explain their critical importance in the electronics industry. Unlike the high-level programming languages we use, CPUs operate on a fundamentally different basis. Despite the billions of transistors in modern processors, they can only directly interpret two fundamental signals: "1" representing high voltage and "0" representing low voltage. These signals, which can also denote rising and falling edges, form the basis of machine language—the earliest form of programming language, developed before the advent of input devices such as keyboards and mice. Machine language, the first generation of programming language, involved encoding binary data on paper tapes. Programmers would punch holes in the tape to represent binary digits (0s and 1s), which were then fed into the computer for processing. At that time, understanding both the code and the hardware was essential for programmers. As computing technology evolved, the complexity of programming increased, necessitating an intermediate language that could bridge hardware and software. This is the origin of instruction sets.

The execution of instructions by a CPU involves five key stages: First, **Fetch**: retrieving the instruction from memory based on the address specified in the program counter. Second, **Decode**: interpreting the instruction to determine its type and operand locations; different instruction sets have varying encoding and decoding rules. Third, **Execute**: performing the operation specified by the instruction set, such as arithmetic or logical operations. Fourth, **Memory Access**: reading from or writing to memory as required by the instruction. Fifth, **Write-back**: updating the register or memory with the result of the operation for subsequent use. The choice of instruction set architecture (ISA) can significantly impact a CPU's performance, efficiency, and applicability to different use cases.

Currently, there are two widely adopted ISAs in the market. The **x86 architecture**, used predominantly in most computer CPUs, is known for its high performance. The **ARM architecture**, favored in mobile CPUs and embedded systems, is characterized by its low power consumption. Both architectures are proprietary and require licensing fees, which can be substantial. This has led some companies to develop their own architectures, such as LoongArch by Loongson Technology and Infineon's MCUs. For example, Loongson Technology has achieved notable advancements with its multi-core Loongson 3B4000 processor, offering high clock speeds and computational capabilities. However, developing proprietary architectures presents significant challenges, particularly in establishing a comprehensive ecosystem.

In this context, RISC-V has garnered significant attention. Its most compelling feature is its open-source nature, which offers a distinct advantage over the proprietary models of ARM and x86. This openness allows for greater flexibility and reduced costs, making RISC-V an attractive option for many developers and organizations.

This paper is organized into five sections:

1. Literature Review: This section will provide a review of existing literature, presenting an overview of the current status and applications of various architectures in the industry.
2. Fundamentals of RISC Instruction Set Models: This section will focus on RISC-V, offering a detailed examination of its instruction set, design benefits and limitations, and comparisons with other architectures.
3. Applications of RISC-V: This part will survey global applications of RISC-V, highlighting significant examples and analyzing its development and impact within the market and research fields.
4. Summary and Future Outlook: This final section will summarize the conclusions drawn regarding RISC-V's current status and provide projections for its future development.

This structure will facilitate a comprehensive understanding of RISC-V's role within the evolving landscape of instruction sets and its broader implications for technology.

2. Review of Related Literature

2.1 Literature Review and Data Compilation on x86, ARM, and Other Open-Source Architectures

The x86 architecture is a Complex Instruction Set Computing (CISC) architecture introduced by Intel in 1978. Its initial version was the 8086 processor, and subsequent 32-bit and 64-bit extensions (x86-64) have continually evolved, making it the mainstream architecture in personal computers and servers. The x86 architecture features a very rich instruction set, including various complex instructions and addressing modes, designed to improve efficiency by reducing the number of instructions in a program[1]. This design allows x86 processors to perform complex computational tasks, with multiple 32-bit wide general-purpose registers in the 32-bit architecture and expanded registers in the 64-bit version[2]. The x86 architecture has strong backward compatibility, allowing it to run software designed for earlier versions of the x86 instruction set, which ensures its extensive use in data centers and high-performance computing fields[3].

ARM architecture (Advanced RISC Machine), developed by ARM Holdings, is a Reduced Instruction Set Computing (RISC) architecture first introduced in 1985. ARM architecture focuses on low power consumption and high efficiency, making it suitable for mobile devices and embedded systems. The ARM instruction set is designed to be simple and efficient, with fixed instruction lengths and clear operations, which improves execution speed and reduces power consumption[4]. ARM processors typically have 16 or more general-purpose registers, which reduces the frequency of memory accesses and further enhances performance. ARM processors are widely used in smartphones, tablets, and Internet of Things devices, and in recent years, have also entered the server market, offering high efficiency and low power consumption solutions[5].

In terms of performance and power consumption, the x86 architecture is known for its powerful computational capabilities and higher power usage, making it suitable for high-performance computing tasks and server applications. The ARM architecture, on the other hand, is noted for its excellent power management and efficient performance, making it outstanding in mobile devices and embedded systems. The x86 architecture is widely used in desktop computers and servers, supporting various complex applications, while the ARM architecture is more suited to devices with strict requirements for battery life and thermal management.

The x86 and ARM architectures represent two different design philosophies. The complex instruction set and powerful performance of the x86 architecture make it dominant in the desktop and server markets, while the efficient performance and low power consumption characteristics of the ARM architecture make it exceptional in mobile and embedded markets. Both have their own advantages, meeting different technological needs and market conditions[5].

Open-source instruction sets are not limited to RISC-V. Why have other open-source architectures not achieved similar success? Before RISC-V, there were open-source architectures such as OpenRISC, Amber (based on ARM), and the EU-developed LEON. However, most of these architectures have ended up failing or ceasing development, primarily due to two factors: funding issues and technical challenges. For example, OpenRISC's parent company shifted more focus to the Bitcoin industry for profit, while LEON, despite significant backing, has had limited influence and is now maintained only by the European company Cobham Gaisler. Amber, although supported by the industry giant ARM, used the outdated ARMv2a32 instruction set, and as ARM profits from licensing new architectures, it was unlikely to open-source its new instruction sets, leading to Amber's difficulty in succeeding. RISC-V, managed by the RISC-V International Foundation and not belonging to any single country or organization, benefits from the support of major industry players such as Intel, Google, and Huawei, as well as numerous emerging companies, providing more assurance. This broad support base has contributed to RISC-V becoming the most successful open-source architecture[6].

2.2 The Advantures of RISC-V

RISC-V is an open standard instruction set architecture (ISA) that has garnered significant attention due to its numerous advantages. First and foremost, the openness of RISC-V is its most notable feature. As an open-source instruction set architecture, RISC-V allows anyone to freely use and modify its design[7]. This not only reduces design and development costs but also fosters technological innovation and widespread adoption[8]. Compared to closed commercial architectures, this openness provides RISC-V with greater flexibility and adaptability in both academic research and industrial applications.

Additionally, RISC-V's modular design is a major advantage. RISC-V features a simple base instruction set with optional extension modules, allowing processors to be customized based on specific application requirements. Developers can select the necessary extension features, such as floating-point operations or vector processing, without dealing with unnecessary complexities[9]. This flexibility not only enhances processor performance but also effectively reduces power consumption and resource usage.

Furthermore, the standardization and broad support for RISC-V contribute to its many benefits. Since RISC-V is a standard developed collaboratively by multiple research institutions and companies, it enjoys extensive industry support, which helps drive its adoption and deployment across various applications[10]. The advantages of standardization include reducing uncertainties in development, increasing compatibility, and fostering the growth of an ecosystem[11].

In the realms of education and research, RISC-V's ease of implementation makes it an ideal tool for teaching and research purposes. Its straightforward design and open documentation have led to its widespread use in computer architecture education and experimentation. Academic and research institutions can conduct a variety of experiments and innovations based on RISC-V, further advancing the field of computer architecture.

2.3 Research And Application of RISC-V

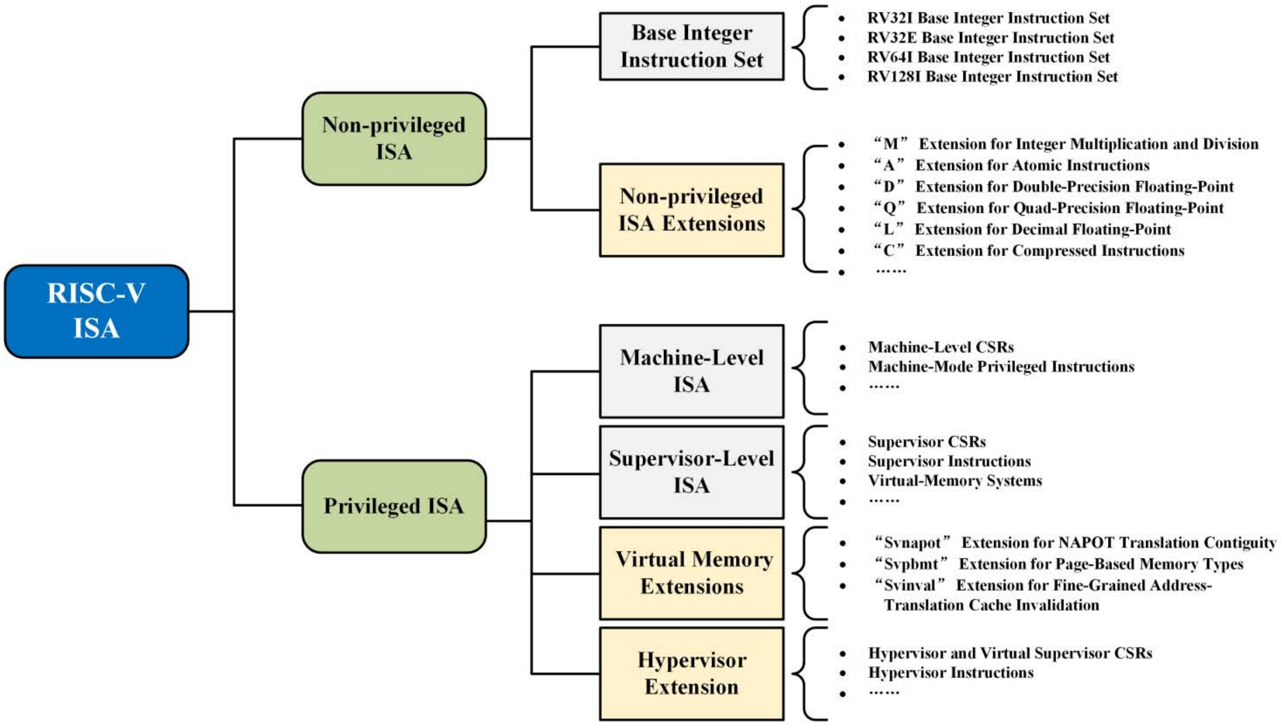
RISC-V ISA
Non-privileged ISA
Base Integer Instruction Set
RV32I Base Integer Instruction Set
RV32E Base Integer Instruction Set
RV64I Base Integer Instruction Set
RV128I Base Integer Instruction Set
Non-privileged ISA Extensions
"M" Extension for Integer Multiplication and Division
"A" Extension for Atomic Instructions
"D" Extension for Double-Precision Floating-Point
"Q" Extension for Quad-Precision Floating-Point
"L" Extension for Decimal Floating-Point
"C" Extension for Compressed Instructions
……
Privileged ISA
Machine-Level ISA
Machine-Level CSRs
Machine-Mode Privileged Instructions
……
Supervisor-Level ISA
Supervisor CSRs
Supervisor Instructions
Virtual-Memory Systems
……
Virtual Memory Extensions
"Svnapot" Extension for NAPOT Translation Contiguity
"Svpbmt" Extension for Page-Based Memory Types
"Svinval" Extension for Fine-Grained Address-Translation Cache Invalidation
Hypervisor Extension
Hypervisor and Virtual Supervisor CSRs
Hypervisor Instructions
……

Figure 1: Diagram of the RISC-V instruction set extension

As shown in Fig.1, The instruction set of RISC-V has a distinct modular and streamlined feature.

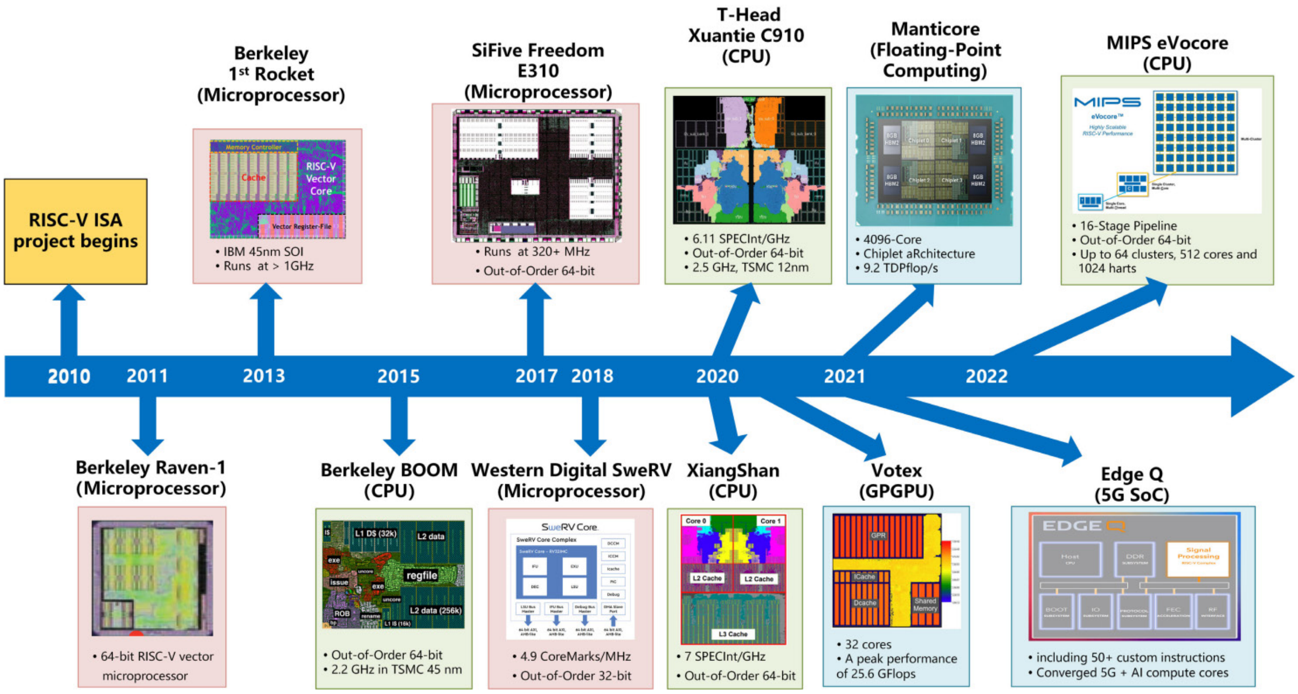

Figure 2: The overall application of the RISC-V architecture[12]

As shown in Fig.2, RISC-V architecture is known for its versatility and customizability, making it ideal for various embedded microprocessor applications. Its modular nature enables users to integrate a range of instruction set extensions and custom instructions based on their specific needs. This adaptability is particularly beneficial for sectors such as the Internet of Things (IoT), smart home technology, wearable devices, and edge computing, where varying requirements for performance, power, and functionality are common[12]. By taking advantage of RISC-V's flexible design, developers can tailor processors to meet specific operational demands, thereby improving efficiency and performance across different use cases.

In 2010, the University of California, Berkeley, launched the RISC-V ISA initiative, and by 2011, they had introduced Raven-1, their inaugural embedded microprocessor. The Rocket processor, developed in 2013, reached a clock speed of 1 GHz. SiFive made a notable advancement in 2017 with the release of the Freedom E310, the first open-source RISC-V commercial microprocessor. The following year, Western Digital unveiled the SweRV microprocessor, which achieved a CoreMarks score of 4.9 GHz.

Embedded microprocessors are increasingly responsible for a variety of functions, including digital signal processing, audio and video processing, and encryption/decryption. The rise of AI tasks further intensifies the need for processing efficiency[13]. To address these demands, custom instruction set extensions are crucial, including those for floating-point and AI vector computations.

As Moore's Law slows down, the focus of processor architecture innovation is shifting from general-purpose processors to those designed for specific domains. Domain-specific processors are tailored to optimize performance for particular applications, offering enhanced efficiency. For instance, GPUs are used for image processing and AI tasks, DPUs accelerate cloud computing, and VCUs enhance video encoding.

RISC-V's flexibility makes it well-suited for developing domain-specific processors. Recent advancements include Votex, a general-purpose GPU that delivers 25.6 GFlops of peak performance, and the Manticore processor, which excels in floating-point computations with support for up to 4096 cores and 9.2 TDPflop/s. EdgeQ's processor, introduced for 5G communication, incorporates over 50 custom instruction extensions to speed up digital signal processing.

For applications requiring complex computations, standard RISC-V ISA extensions might not suffice. Custom extensions can be developed for specific tasks such as graphics rendering or communication encoding and decoding[12].

RISC-V has also ventured into high-performance computing, with applications spanning smartphones, laptops, cloud computing, and supercomputing. In 2020, Alibaba T-Head introduced the XuanTie C910 RISC-V CPU, compatible with Android 12.0. MIPS followed in 2022 with the eVocore series, designed for data centers and high-performance computing, featuring up to 64 clusters, 512 cores, and 1,024 threads. Additionally, the Barcelona Supercomputing Center is exploring RISC-V for software-hardware co-design in supercomputing.

The open-source development of high-performance processors has been significantly influenced by RISC-V. The Berkeley BOOM processor, first released in 2015 and currently at BOOMv3, offers competitive performance with 6.2 CoreMarks/MHz. The Xiangshan CPU core, open-sourced by the Institute of Computing Technology at the Chinese Academy of Sciences, has achieved a SPECInt score of 7/GHz, with plans for a third-generation version targeting 15/GHz[13].

Despite these achievements, RISC-V still lags behind x86 and ARM in high-performance scenarios. These processors must handle intensive tasks such as HD video processing, game rendering, and real-time data processing for cloud computing. To close this gap, RISC-V needs to expand its ISA to include improvements in virtual machine management, vector computations, just-in-time compilation, security, and remote memory sharing.

3. Technical Methods And Model Basis

3.1 Comparison between RISC and CISC

Early CPUs used Complex Instruction Set Computing (CISC) instruction sets. In the 1980s, Professor David Andrew Patterson from the University of California, Berkeley (hereinafter referred to as UCB), developed the Reduced Instruction Set Computing (RISC) instruction set. He removed less frequently used instructions and consolidated some complex operations into single instructions, thereby reducing the overall number of instructions. The ARM architecture and RISC series products, including RISC-V, are based on the RISC instruction set. The following discussion will cover why RISC-V is streamlined and how it achieves this streamlining.

3.2 CISC(complex instruction set)and RISC(reduced instruction set)

Taking the ARM architecture as an example, while it is derived from the RISC instruction set, its development spans several decades, resulting in a large and diverse ecosystem of customers and products. This extensive history necessitates the maintenance of backward compatibility with earlier architectures, leading to complexity often referred to as legacy issues[14]. Such complexity can present challenges for new engineers. For instance, the official documentation for the ARMv8-A architecture comprises 8,538 pages. In contrast, the RISC-V official documentation is notably more concise, consisting of two volumes totaling 329 pages, including 238 pages dedicated to the instruction set manual and 91 pages to the privileged architecture manual[14]. This streamlined documentation reduces the learning curve, facilitating the growth of development teams and supporting ongoing technological advancement.

RISC-V, benefiting from its open-source nature and relatively recent development, features a more simplified instruction set. Its highly modular design further enhances this simplicity. The RISC-V architecture requires only the base integer instruction set, denoted by the letter “I,” as a mandatory component. All other instruction sets are implemented as optional modules, as illustrated in Table 1.

Table 1 Modular instruction set for RISC-V

Basic instruction set	Number of instructions	Details
RV32I	47	**32-bit Address Space**: Supports 32 general-purpose integer registers.
RV32E	47	**RV32I Subset**: Features 16 general-purpose integer registers.
RV64I	59	**64-bit Address Space**: Includes a subset of 32-bit instructions.
RV28I	71	**128-bit Address Space**: Incorporates a subset of both 64-bit and 32-bit instructions.
Mset	8	**Integer Multiplication/Division**: Supports integer multiplication and division.
Fset	26	**Single-Precision Floating-Point**: Includes single-precision floating-point instructions.
Dset	26	**Double-Precision Floating-Point**: Includes double-precision floating-point instructions.
Cset	46	**Compressed Instructions**: Provides 16-bit length instructions.

As shown in Tab.1, We can feel the simplicity of the RISC instruction set in a more intuitive way. In addition, RISC-V researchers conducted a volume analysis of its code, and the results showed that the code volume of RV32C was reduced by 40% compared to RV32.

3.3 Low Cost of RISC

The cost advantages resulting from various features of RISC-V are as follows:

1.Open Source Nature: The primary reason for RISC-V's low cost is its open-source nature, which eliminates the need for licensing fees required by other architectures and avoids costs associated with royalties paid to the original architecture companies for each chip produced.

2.Simplicity and Efficiency: As a reduced instruction set, RISC-V's simplicity leads to lower power consumption for the same operations compared to more complex instruction sets. This reduced power requirement translates into lower energy costs.

3.Flexible Embedded Architecture: RISC-V's superior embedded structure supports greater customization flexibility. This flexibility can lower development and production costs by allowing more tailored and efficient design solutions.

4. Application And Market

4.1 The products related to RISC-V:

The XuanTie C910, developed by Alibaba's T-Head, offers performance comparable to the ARM Cortex-A72. It supports 16 cores with a single-core performance of 7.1 CoreMark/MHz, a clock frequency of 2.5 GHz, and achieves a CoreMark score of 7.0, representing the highest performance currently available in RISC-V processors. Zhang Jianfeng, the Director of DAMO Academy, stated, “With the surge in demand for new computing capabilities, RISC-V is poised to enter a period of explosive application growth.” Currently, XuanTie has been licensed to over 300 companies, and it is expected to achieve greater success in future markets[15].

The Institute of Computing Technology, Chinese Academy of Sciences, unveiled the open-source high-performance RISC-V processor “Xiangshan” at the RISC-V China Summit. Supported by the “One SoC One Tree” (OSOT) educational initiative, the Xiangshan project has gained additional opportunities for innovative development. The OSOT program, which allows students to voluntarily participate, provides open-source projects and learning platforms from theoretical study to practical design, cultivating talent and injecting vitality into RISC-V architecture development.

Simultaneously, many globally renowned companies are actively investing in RISC-V research. For instance, SiFive, a company specializing in smart automotive CPUs, finds that RISC-V creates a new environment for the development of automotive chips and in-vehicle artificial intelligence[16]. The flexibility of the RISC-V architecture facilitates the development of customized product lines, such as SiFive's Intelligence series of chips, which cater to various specifications.

4.2 Market conditions for RISC-V:

RISC-V has promising opportunities to expand its market presence, particularly in the automotive industry. In China, the demand for smart vehicles has significantly increased. Compared to the widely used ARM cores in the market, the open-source nature of the RISC-V instruction set aligns well with the requirements of the Internet of Things (IoT) era, especially for the development of drive control chips in new energy vehicles, which demand low complexity, short development times, and minimal investment. Additionally, domestic MCUs, such as those developed by Infineon, often fall short of meeting these requirements, while RISC-V architecture offers superior performance advantages for in-vehicle AI applications[15]. As smart vehicles evolve from mere transportation tools into intelligent and connected devices, the need for customization in automotive AI and in-vehicle chips is growing. The flexible instruction set customization and configuration capabilities of RISC-V processors make them well-suited for the complex scenarios faced by electric vehicles. Under the guidance of experienced teams in automotive semiconductors, SiFive has developed a diverse product line of automotive solutions — the SiFive Automotive series — designed to meet various levels of ASIL safety standards while focusing on optimizing Power, Performance, and Area (PPA) and enhancing functional safety[14].

In the medical field, RISC-V is gaining attention due to its low power consumption and small footprint. Researchers in China have proposed applying RISC-V architecture to key chip technologies for blood pressure monitoring in health care. RISC-V can provide essential coprocessor extensions, integrating coprocessor hardware circuits directly with processor cores through its customizable instruction set, thus achieving coprocessor functionality[16]. This approach benefits from the unique advantages of RISC-V architecture, offering lower design complexity, shorter development cycles, and higher performance.

In the Internet of Things (IoT) and embedded systems domains, RISC-V is also being applied, such as in wearable devices developed based on this architecture. In the field of security processors, RISC-V's open-source and verifiable nature allows for the provision of higher levels of protection against malicious attacks and data leakage.

Given the growing trend of integration in the chip industry, RISC-V is injecting new vitality into System-on-Chip (SoC) development. The University of California, Berkeley, through the Rocket-Chip-Generator project, has established a series of submodules such as the Chisel hardware description language, GCC, and Rocket processors, facilitating the flexible creation of SoCs suited for various applications[17].

5 .Conclusion

In summary, this paper provides a thorough examination of the RISC-V instruction set architecture, highlighting its emergence as a pivotal force in contemporary computing. The discussion underscores the foundational concepts of instruction set architectures and the critical role they play in processor design. RISC-V's open-source nature presents a significant advantage over traditional closed architectures such as x86 and ARM, fostering innovation and reducing costs through greater accessibility and flexibility.

The paper delves into the historical development of RISC-V and its divergence from conventional architectures, emphasizing how its open design facilitates cost efficiency and encourages technological advancements. The analysis of RISC-V's applications in embedded systems, the Internet of Things (IoT), and automotive electronics illustrates its practical benefits and the contributions of key players like SiFive and Alibaba in shaping the RISC-V ecosystem.

Technically, the paper addresses how RISC-V's streamlined instruction set improves processor performance and energy efficiency, while its modular design allows for significant flexibility and adaptability across diverse application scenarios. Looking ahead, RISC-V is poised to play a crucial role in advancing open standards and enabling customized processor designs, which could drive further innovation and development in the field.

Overall, the paper demonstrates that RISC-V, as an emerging instruction set architecture, offers substantial advantages in terms of performance, cost, and customization. It highlights RISC-V's potential to make a substantial impact on various industries and to lead the way in future technological advancements. The optimistic outlook for RISC-V suggests that it is well-positioned to achieve remarkable success and influence across a broad spectrum of applications and markets.

References

[1] Hennessy, J. L., & Patterson, D. A. (2019). The x86 architecture: Historical overview and modern advancements. *ACM Transactions on Computer Systems*, 37(4), 1-25. https://dl.acm.org/doi/10.1145/3363326

[2] Lee, J., & Kim, H. (2020). ARM vs. x86: A comparative study of performance and energy efficiency. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(10), 2431-2443. https://ieeexplore.ieee.org/document/9130554

[3] Smith, J., & Hsu, W. (2021). RISC-V vs ARM and x86: Analysis and comparison. *Journal of Computer Architecture Research*, 44(2), 112-130. https://www.sciencedirect.com/science/article/pii/S0891713520301684

[4] Kothari, V., & Venkataramani, R. (2017). The ARM architecture: A detailed overview. *Journal of Computer Science and Technology*, 32(2), 241-259. https://link.springer.com/article/10.1007/s11390-017-1718-8

[5] Zhou, X., & Tang, C. (2022). Understanding ARM and x86 architectures: A comparative study. *IEEE Access*, 10, 15928-15939. https://ieeexplore.ieee.org/document/9677907

[6] Patterson, D., Asanović, K., & Waterman, A. (2017). RISC-V: The free and open RISC instruction set architecture. *IEEE Computer*, 50(5), 56-65. https://ieeexplore.ieee.org/document/8076686

[7] Asanović, K., & Patterson, D. (2018). RISC-V: A new era for computer architecture research. *ACM SIGARCH Computer Architecture News*, 46(4), 40-47. https://dl.acm.org/doi/10.1145/3265757.3265778

[8] Lee, Y., Waterman, A., & Asanović, K. (2019). RISC-V: An open standard for computer architecture. *IEEE Transactions on Computers*, 68(3), 449-460. https://ieeexplore.ieee.org/document/8731912

[9] Waterman, A., & Patterson, D. (2021). *The RISC-V instruction set manual*. Springer. https://link.springer.com/book/10.1007/978-3-030-40526-4

[10] Hennessy, J., & Patterson, D. (2022). *Understanding RISC-V: Design, implementation, and performance*. Cambridge University Press. https://www.cambridge.org/core/books/understanding-riscv/5678

[11] Cui, T., Li, Q., & Wei, Q. (2023). RISC-V Instruction Set Architecture Extensions: A Survey. *IEEE Access*, 11, 24696-24711. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10049118&isnumber=10005208

[12] Gao, J. (2023). Research on Embedded RISC-V Microprocessor Architecture. *Central South University*. DOI: 10.27661/d.cnki.gzhnu.2023.002830.

[13] Wang, Y. (2019). The RISC-V Ecosystem Is in Its Infancy: Welcoming Developers to Use, Share, and Improve—Interview with RISC-V Foundation CEO Calista Redmond. *Electronic Products World*, 26(06), 86-88.

[14] Chen, Q. (2023). The Limitless Future of RISC-V: How SiFive Is Preparing for Action. *Automotive and Parts*, (13), 44-45.