IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier 10.1109/ACCESS.2022.DOI

RISC-V Instruction Set Architecture Extensions: A survey

ENFANG CUI, (Member, IEEE), TIANZHENG LI, QIAN WEI
China Telecom Research Institute, Beijing, 102209 China
Corresponding author: Enfang Cui (e-mail: cuief@chinatelecom.cn).

ABSTRACT RISC-V is an open-source and royalty-free instruction set architecture (ISA), which opens up a new era of processor innovation. RISC-V has the characteristics of modularization and extensibility, and explicitly supports domain-specific custom extensions. Nowadays, RISC-V is a popular ISA for embedded processors. However, there is still a gap between the capabilities of RISC-V and the requirements of various emerging computing scenarios (e.g., artificial intelligence, cloud computing). Recently, the RISC-V standards organization has continuously introduced new ISA extensions to meet the needs of advanced computing. There are also a variety of novel research proposed customized extensions of RISC-V for certain scenarios. As far as we know, there is a lack of a survey to systematically present the research progress of RISC-V ISA extensions. The goal of this paper is to provide a comprehensive survey on existing works about RISC-V ISA extensions. First, the application scenarios of RISC-V are introduced, and the requirements for ISA extensions are analyzed. Then, we survey the progress of official RISC-V ISA extension specification and recent research on RISC-V ISA extension. Finally, we highlight some possible future research opportunities on the RISC-V ISA extension.

INDEX TERMS RISC-V, instruction set architecture, extensions, survey.

I. INTRODUCTION

THE chip is the foundation of the information industry, and the instruction set architecture (ISA) is the basic technology of the chip. In the development history of the chip industry in the past decades, x86 and ARM ISAs have gradually come to the fore and occupied the mainstream market position. However, x86 and ARM are not open source and have high license fees. RISC-V [1] is an open source ISA which has grown rapidly since its inception, owing to its benefits of no licence fee, flexibility, and scalability, and has gradually emerged as a key driver of global chip innovation. For instance, Europe launched the European Processor Initiative [2], which plans to develop and produce high-performance chips based on RISC-V, and apply them to the field of supercomputing. China established the China RISC-V Alliance [3] to encourage the application and promotion of RISC-V related technologies and products. It is anticipated that the RISC-V processor architecture will play a significant role in both industry and research, and there will be more than 60 billion RISC-V cores in 2025 [4].

However, although RISC-V has achieved significant success in recent years, its ISA standard is still under development and needs to be further enhanced for various scenarios. RISC-V is reaching maturity in the field of embedded processors, but it is still in its infancy in other areas such as cloud computing, high-performance computing, etc. Additionally, the instruction set must also be further enriched to suit the requirements of domain-specific application scenarios such as artificial intelligence, graphic computing, etc.

RISC-V ISA consists of basic instruction sets and modular ISA extensions, and it also explicitly supports domain specific custom extensions. RISC-V basic instruction set only contains the basic integer instruction set for general integer operations (e.g., addition and subtraction). The RISC-V ISA extensions contain a variety of modular extensions to meet different computing needs. The RISC-V standard organization has constantly introduced new instruction set extensions to enhance the functionality of RISC-V, such as the V-extension for vector calculation and the H-extension for virtualization. In addition to the instruction set extension standards developed by the standard organization, academic researchers have also carried out a number of studies to customize instruction set extensions for a variety of scenarios to accelerate specific applications or computing tasks. For instance, work [5] proposed a DSP instruction set extension for Internet of Things applications to improve the energy efficiency of wireless signal processing. For artificial intelligence applications, works [6] and [7] proposed instruction

set extensions for acceleration of deep learning algorithms. A lot of research is also being done on customising RISC-V instruction set extensions for high performance computing [8], graphics computing [9], communication [10], [11], security [12], and so on.

Even though a tremendous amount of study has been done, the research of RSIC-V ISA extension is still in its early stages, and we believe that both academics and industry professionals would benefit from a comprehensive survey. We have noticed that there are some early surveys [12]–[14] focused on RISC-V. The work [13] reviews the software support for RISC-V. The work [12] presents an overview of RISC-V security technologies. The work [13] compares the different implementations of RISC-V processors. As far as we know, there is a lack of a survey to systematically present the research progress of RISC-V ISA extensions. In this paper, we systematically present recent advances in RISC-V ISA extensions. The main contributions of this paper are:

- We present the main application scenarios of RISC-V and analyze the requirements of different scenarios for ISA extension.
- We survey the progress of official RISC-V ISA extension specification and recent research on RISC-V ISA extension. We identify the main contributions of recent research.
- We highlight some possible future opportunities for RISC-V ISA extension.

The structure of this paper is shown in Figure 1. The rest of this paper is structured as follows. The applications of RISC-V are presented in Section II. Section III reviews the progress of the official RISC-V ISA extension specification. Section IV reviews the recent research on RISC-V ISA extension. Section V discusses possible opportunities. Finally, Section VI concludes this paper.

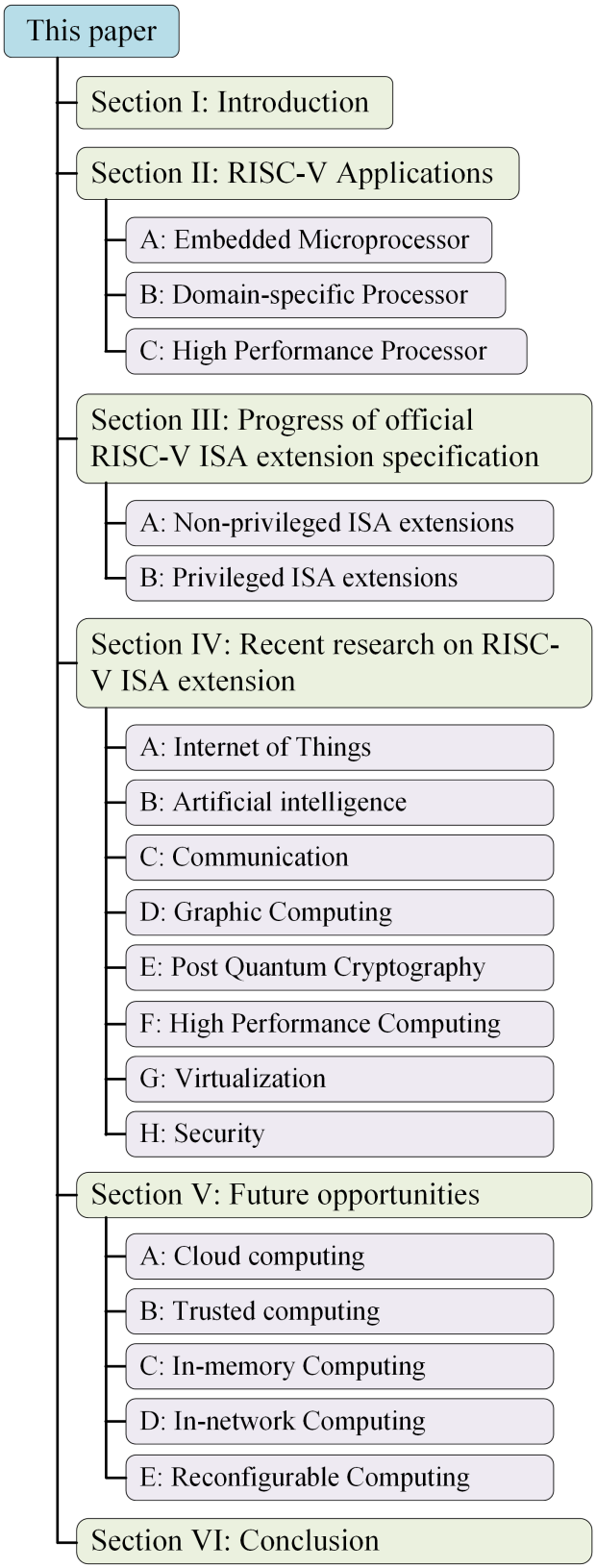


FIGURE 1. The structure of this paper.

II. RISC-V APPLICATIONS

In this section, we present the main application scenarios of RISC-V, including embedded microprocessor, domain-specific processor and high performance processor. We introduce some typical RISC-V processors as shown in Fig. 2. In addition, the requirements of different scenarios for instruction set extension are analyzed.

A. EMBEDDED MICROPROCESSOR

Embedded microprocessor application scenarios are diversified and different. RISC-V is widely used in the above scenarios because of its modularity and customization. Users can freely combine RISC-V instruction set extensions and customize private instruction sets according to their own scenario requirements. Typical application cases include the Internet of Things [15], [16], smart home [17], wearable devices [18], [19], and edge computing [20], [21], etc. The academia and industry has released several RISC-V embedded microprocessors with different characteristics for the above applications. As shown in Figure 2, Berkeley University launched the RISC-V ISA project in 2010 and taped out

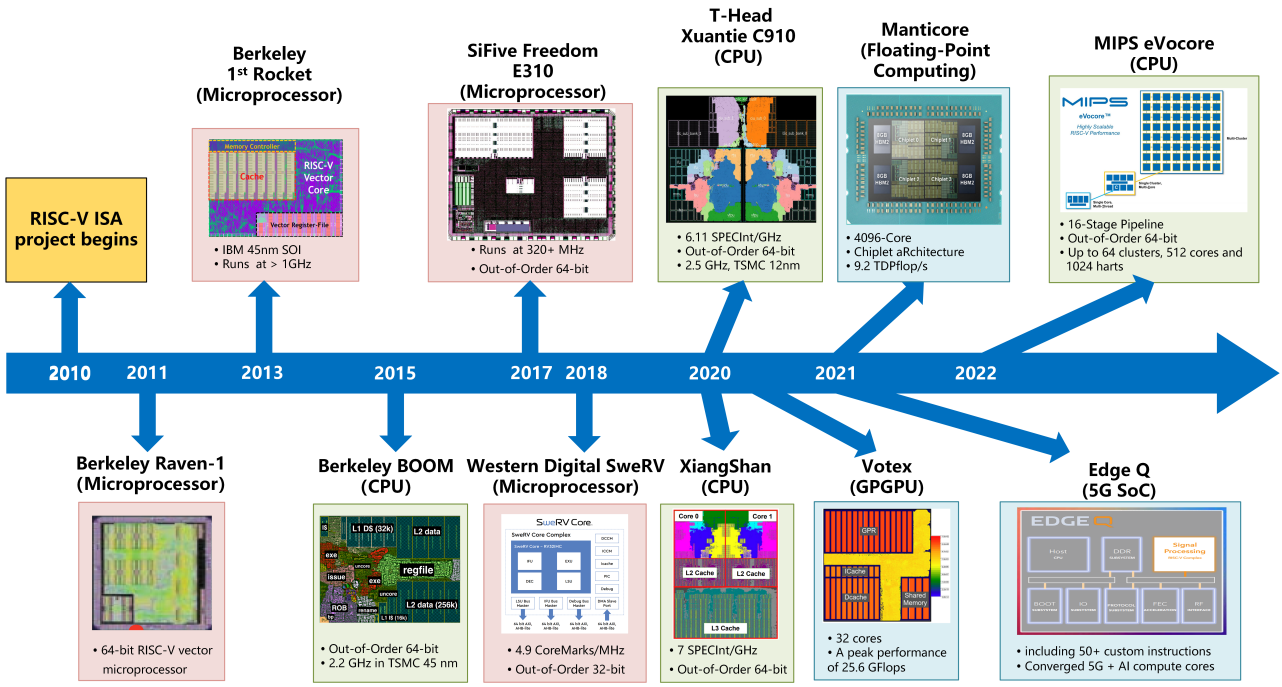


FIGURE 2. Typical RISC-V processors.

the first embedded microprocessor Raven-1 in 2011. In 2013, they further designed Rocket [22] microprocessor, which realized its operation at 1GHz frequency. The above microprocessors are mainly used for early verification of RISC-V architecture. SiFive company released the Freedom E310 [23] microprocessor in 2017, which is the first open-source commercial RISC-V microprocessor. Western Digital company also released the open-source microprocessor SweRV [24] in 2018, achieving a CoreMarks score of 4.9/GHz. Since then, RISC-V has gradually become a popular architecture for microprocessors.

Embedded microprocessors often need to perform tasks such as digital signal processing, audio and video processing, and encryption/decryption, etc. In addition, emerging AI computing tasks are also gradually deployed on embedded processors to improve real-time performance. The above tasks put forward higher requirements for the computing efficiency of the processor. Since embedded devices generally have energy consumption constraints, it is urgent to customize efficient and low-power instruction set extensions, such as efficient floating point number computing extensions, AI vector computing extensions, to meet the application requirements.

B. DOMAIN-SPECIFIC PROCESSOR

With the fading of Moore's law, there is a shift of processor architecture innovation from general-purpose to domain-specific processors [25]. The advantage of domain-specific processors is that it can optimize the processor architecture for a specific class of applications to achieve better efficiency. Typical domain-specific processors include graphic processing unit (GPU) [26] for image processing and artificial intelligence (AI), data processing unit (DPU) [27] for cloud computing infrastructure acceleration, video coding unit (VCU) [28] for video codec acceleration, etc. RISC-V is extensible and customizable, and naturally suitable for the construction of domain-specific processors. There are some works that propose domain-specific processor schemes based on RISC-V for different application scenarios. As shown in Figure 2, Elsabbagh *et al.* [9] proposed Votex, a general-purpose graphics processing unit (GPU) that can be used for graphics computing and rendering, in 2020. It has a peak performance of 25.6 GFlops. Zaruba *et al.* [29] proposed Manticore, a processor used for floating point computing, in 2021. It adopts the chiplet architecture, supports up to 4096 cores, and its peak performance reaches 9.2TDPflop/s. In the same year, EdgeQ company released the processor of the same name, EdgeQ [11], a SoC specially used for 5G communication, and customized more than 50 instruction extensions to accelerate 5G digital signal processing.

The domain-specific applications involve some specific complex computing tasks. In the case that the official ISA extensions provided by RISC-V cannot meet the acceleration requirements of domain specific tasks, it is necessary to customize RISC-V instruction set extensions to accelerate these tasks. For example, customize the graphic processing instruction set extensions for graphic rendering applications, and customize the communication encoding and decoding instruction set extensions for communication applications.

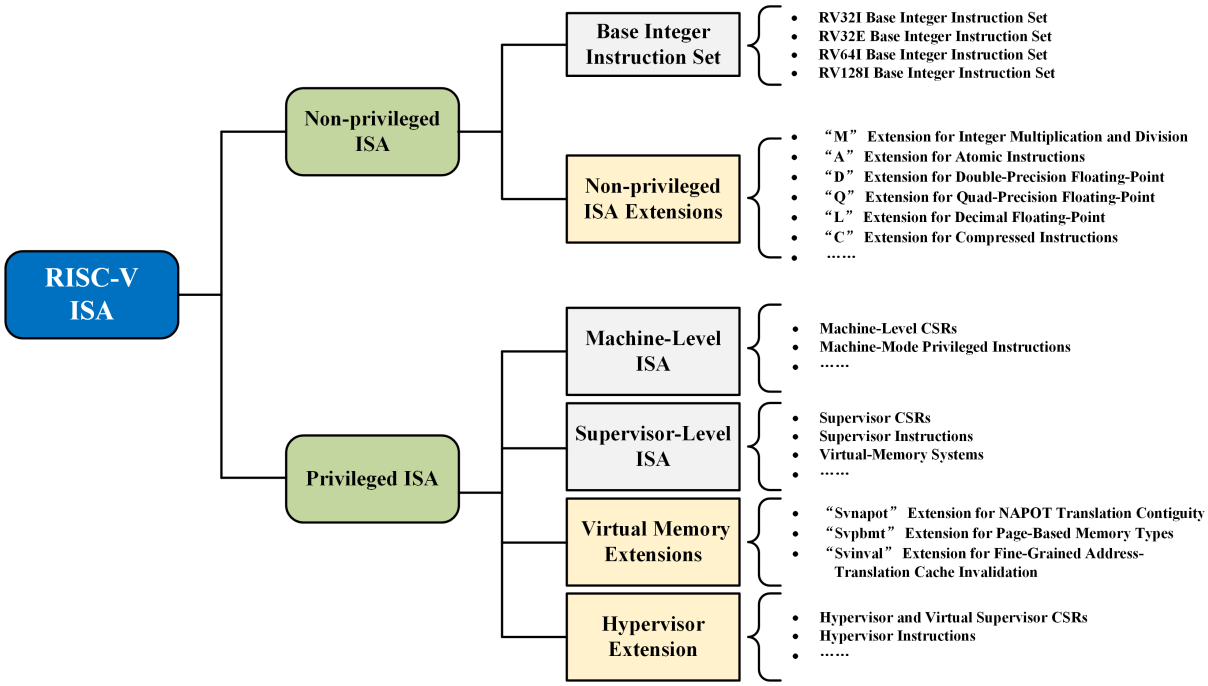


FIGURE 3. The RISC-V ISA.

C. HIGH PERFORMANCE PROCESSOR

In recent years, RISC-V has been used to develop high-performance computing processors, and potential application scenarios include smartphones, laptops, cloud computing, and supercomputing, etc. For instance, Alibaba T-Head released the XuanTie C910 [30] RISC-V CPU and realized the adaptation of Android 12.0 system in 2020. MIPS released eVocore, a series of multi-core-multi-cluster processors for datacenter and high-performance applications, in 2022. It combines a 16-stage deep pipeline with multi-issue Out-of-Order execution, and it can scale up to 64 clusters, 512 cores and 1,024 harts/threads. The Spain Barcelona Supercomputing Center explored software-hardware co-design of supercomputers based on RISC-V [31]. RISC-V also promotes the open source of high-performance processors (such as CPUs). The most representative open source RISC-V processor is Berkeley Out of Order Machine (BOOM) [32]. Berkeley University released the first generation BOOM in 2015, and the current version is BOOMv3. With a performance of 6.2 CoreMarks/MHz, BOOMv3 is competitive with commercial high-performance cores. Besides, the Chinese Academy of Sciences Institute of Computing has open sourced the Xiangshan [33] CPU core, which has a SPECInt score of 7/GHz. They are working towards the third generation version, which targets 15/GHz.

Although RISC-V has made significant progress in the field of high-performance processors, there is still a gap between the current RISC-V instruction set and the x86 and ARM instruction sets in this field. In the high-performance processor application scenarios, the processors usually need to perform a large number of compute-intensive tasks. For example, smart phones need to perform high-definition video processing, game rendering, and just-in-time compilation of advanced programming languages, etc. Cloud computing needs to perform large-scale virtualization, real-time big data processing, high-throughput communication protocol encryption and decryption processing, etc. High performance computing requires remote memory sharing. RISC-V needs to further extend the ISA in terms of hypervisor, vector computing, just-in-time compilation, security, and remote memory sharing, etc.

III. PROGRESS OF OFFICIAL RISC-V ISA EXTENSION SPECIFICATION

RISC-V standards organization, namely RISC-V International Foundation, has formulated the official RISC-V specification. This paper presents the progress of RISC-V ISA extensions based on the latest version of RISC-V specification released in 2019. As shown in Figure 3, RISC-V ISA consists of two parts: non-privileged ISA and privileged ISA. The non-privileged ISA includes the basic integer instruction set and several non-privileged ISA extensions. The privilege ISA contains the basic machine-level and supervisor-level ISAs, as well as virtual memory extensions and hypervisor extension. In this section, we introduce the non-privileged ISA extensions and privileged ISA extensions.

A. NON-PRIVILEGED ISA EXTENSIONS

1) M-extension

The M-extension is the integer multiplication and division instruction set extension, which is used to multiply or divide

the values held by two integer registers, including signed and unsigned integer multiplication and division instructions.

2) A-extension

The A-extension, also known as the atomic instruction set extension, consists of instructions that read, modify, and write memory atomically to provide synchronisation across several RISC-V harts in the same memory space. The A-extension provides two types of atomic instruction, which are atomic fetch-and-op memory instructions and load-reserved/store-conditional instructions.

3) F-extension

The F-extension offers instructions for single-precision floating-point operations that adhere to the IEEE 754-2008 arithmetic standard [34]. It also includes a floating-point control and status register and extra thirty-two floating-point registers that are each thirty-two bits wide.

4) D-extension

The D-extension is used to support double-precision floating-point operations and widen 32 floating-point registers to 64 bits. The D-extension depends on the F-extension, and the instruction type is similar to F-extension.

5) Q-extension

The Q-extension is the quad-precision binary floating-point ISA extension. In Q-extension, the floating-point register is extended to 128-bit. In particular, Q-extension adds new quad-precision load, store, convert, move instructions, and quad-precision floating-point classify, compare, computational instructions. The D-extension is a prerequisite for using Q-extension.

6) L-extension

The L-extension is used to support decimal floating point arithmetic. It is mainly utilized to save 64-bit and 128-bit decimal floating-point values in registers and move these values to and from memory with load and store instructions. L-extension is currently in the draft stage and has not been ratified by the RISC-V International Foundation.

7) C-extension

The C-extension is the compressed instruction-set extension. By replacing a few 32-bit common operations with short 16-bit instructions, it can significantly reduce the size of static and dynamic code. Additionally, any base ISA could be enhanced with C-extension.

8) Counters-extension

The Counters-extension provides up to 32 64-bit counters that can be accessed through specific read-only CSR registers. The first three counters (CYCLE, TIME, and INSTRET) are specifically used for cycle counting, real-time clock, and instruction retirement, whereas the remaining counters can be used for programmable event counting.

9) B-extension

The B-extension is the bit manipulation instruction-set extension, which consists of instructions such as zba, zbb, zbc, zbs, etc. These instructions can support bit manipulation including bit counts, shifts, rotations, insertions, extractions, and permutations, etc.

10) J-extension

The J-extension is designed to support dynamically translated languages, such as Java, JavaScript, etc. These languages are typically implemented via dynamic translation. The J-extension aims to provide additional instructions to support dynamic checks, garbage collection, and JIT acceleration, etc.

11) T-extension

The T-extension is the transactional memory extension. It will provide instructions to support transactional memory operations, and will add a small, limited-capacity transactional memory buffer to support atomic operations involving multiple addresses.

12) P-extension

The P-extension is the packed single instruction multiple data (Packed-SIMD) instruction set extension. It provides Integer/fixed-point SIMD/DSP instructions, and can support efficient data-parallel computing such as audio, voice and images processing, etc. P-extension is mainly designed for embedded or DSP devices.

13) V-extension

The V-extension is the vector ISA extension and adds a set of vector registers and vector operation instructions to the basic scalar RISC-V ISA. The V-extension is a kind of SIMD instruction extension that supports Vector Length Agnostic SIMD operations. Most conventional SIMD instructions (e.g., Intel AVX) focused on fixed vector length, any modification to the vector length usually makes previously compiled code obsolete. The Vector Length Agnostic SIMD operations decouple the vector length from the compiled code so that the same executable could run on processors with different implementations of vector length. The V-extension makes it easier to design more portable software.

Currently, V-extension is a research hotspot in both industry and academia. Chen *et al.* [35] of Alibaba T-Head designed a commercial processor Xuantie 910 that supports V-extension. Maisto *et al.* [36] extended the open source RISC-V processor CAV6 and designed a pluggable vector unit to support V-extension. Assir *et al.* proposed Arrow [37], an edge machine learning inference hardware accelerator based on RISC-V V-extension. Taking advantage of the flexibility of the vector architecture, we will see plenty of RISC-V cores supporting V-extension in the future, to accelerate data parallel processing, while reducing energy consumption.

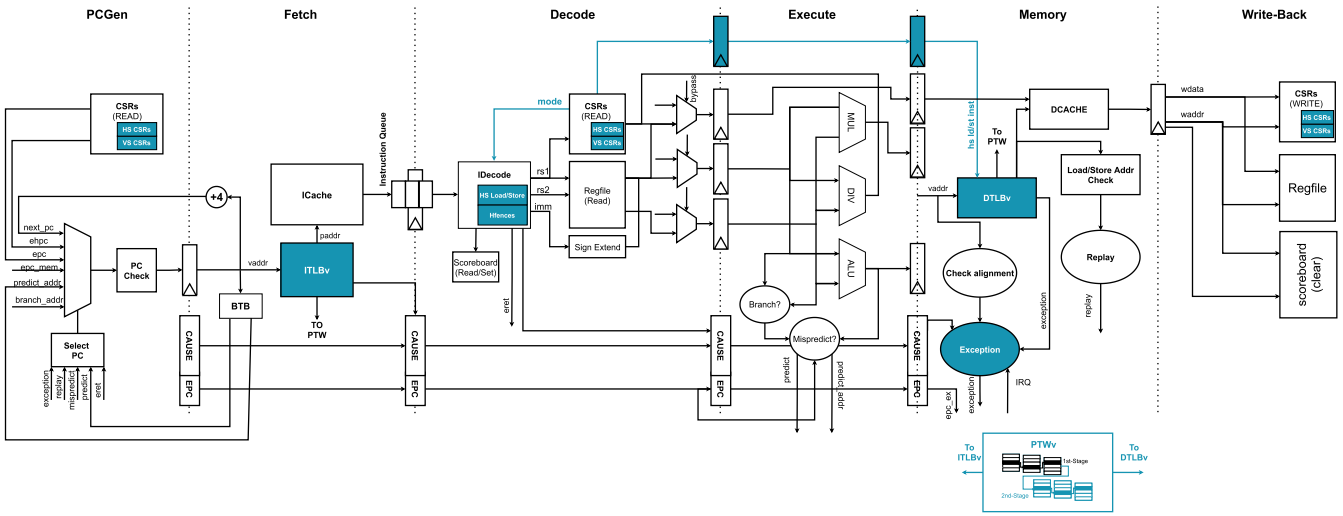


FIGURE 4. Rocket core microarchitecture with H-extension [38].

14) Zicsr-extension

The Zicsr-extension is the control and status register (CSR) instruction set extension, and provided instructions to read-modify-write CSRs atomically. RISC-V defines an independent address space of CSR for each hart. RISC-V has a CSR space of 12 bits, that is, 4096 CSRs at most. Zicsr extension provides 6 instructions to operate CSRs in this space.

15) Zifencei-extension

The Zifencei-extension is the instruction-fetch fence extension. It only specifies one instruction, FENCE.I, which offers explicit synchronisation between instruction memory writing and instruction fetching on the same hardware thread.

16) Zam-extension

The Zam-extension is the misaligned atomic instruction set extension. It adds standardised support for misaligned atomic memory operations (AMOs) to the A-extension. By utilizing Zam-extension, misaligned AMOs need only execute atomically with regard to other accesses to the same address and of the same size.

17) Ztso-extension

The Ztso-extension is the total store ordering instruction set extension. A basic memory consistency model is defined in the official RISC-V specification, which is called RISC-V weak memory ordering (RVWMO). Similar to the weak memory model used by ARMv8, RVWMO has more relaxed ordering rules, providing more freedom for hardware implementation and performance optimization. The Ztso-extension provides a RISC-V total store ordering (RVTSO) memory model, which is an extension of RVWMO. Similar to the strong memory models used by SPARC and x86, RVTSO has strict ordering rules.

B. PRIVILEGED ISA EXTENSIONS

The RISC-V provides privileged architecture for running operating systems and supporting hardware virtualization. The above privileged architecture includes the basic privileged ISA, namely machine-level ISA and supervisor-level ISA, as well as the hypervisor extension and virtual memory extensions. In this section, we will introduce hypervisor and virtual memory extensions.

1) Hypervisor Extension

The hypervisor extension (H-extension) offers hardware support for efficient virtualization. The goal of hypervisor extension is to allow guest operating systems to run under Type-1, Type-2, and hybrid hypervisors, as well as to provide recursive virtualization. The H-extension transforms the supervisor mode (S-mode) into a hypervisor-extended supervisor mode (HS-mode) and introduces two additional modes: virtual supervisor mode (VS-mode) and virtual user mode (VU-mode). The hypervisors run in HS-mode, the guest operating systems run in VS-mode, and user applications run in VU-mode. The H-extension also provides additional CSRs and hypervisor instructions, as well as extends machine CSRs for guest virtual memory control.

Currently, there are several research on RISC-V virtualization hypervisor and H-extension hardware implementation. For instance, work [39] proposes a lightweight, open source embedded hypervisor Bao. Bao is a hypervisor supported by hardware virtualization, has low virtualization overhead and supports RISC-V hypervisor extension (v0.5). Bruno *et al.* [38] provided a public hardware implementation based on the Rocket core for evaluating hypervisor extensions (v0.6.1). Figure 4 depicts an overview of the Rocket core microarchitecture with the H-extension. The blocks marked in blue are new functional blocks based on the Rocket core, including Decoder, PTW, TLBs, and CSRs. They also adapted the Bao open-source hypervisor to test and evaluate the hardware

implementation. Furthermore, this work reduced interference and interrupt latency, by enhancing the RISC-V platform level interrupt controller and timer infrastructure.

2) Virtual Memory Extensions

RISC-V privileged ISA provides a variety of virtual memory systems, including Sv32, Sv39, Sv48 and Sv57, which support 32-bit, 39-bit, 48-bit and 57-bit virtual address spaces respectively. These virtual memory systems can translate virtual addresses into physical addresses through multi-level page tables. Sv32 is used for RV32 architecture, and supports two levels page tables. Sv39, Sv48 and Sv57 are used for RV64 architecture, and support three, four and five levels page tables, respectively. On this basis, RISC-V provides several virtual memory extensions to assist in the implementation of memory virtualization, including Svnapot, Svpbmt and Svinval extensions. Svnapot extension is the naturally aligned power-of-two (NAPOT) translation contiguity extension of supervisor-level ISA. It can realized contiguous virtual-to-physical address translations, and the translation range must be of a NAPOT granularity larger than the base page size. Svpbmt extension is the page-based memory types extension of supervisor-level ISA, and it specifies several memory types with different attributes such as cacheability, indempotency and ordering. Svinval extension is the fine-grained address-translation cache invalidation extension. It splits some supervisor-mode and hypervisor-mode memory fence instructions into more precise invalidation and ordering operations that can be batched or pipelined more efficiently on certain classes of high-performance implementation.

IV. RECENT RESEARCH ON RISC-V ISA EXTENSIONS

In addition to the ISA specification developed by the RISC-V standard organization, there are also a variety of novel research proposed customized extensions of RISC-V for many scenarios, including the Internet of Things, artificial intelligence, communication, graphics computing, post quantum encryption, high-performance computing, virtualization and security. In this section, we will systematically summarize the research on RISC-V ISA extensions according to the above scenario classification, and discuss the contributions of related works. A categorization of RISC-V ISA extension works is shown in Table 1.

A. INTERNET OF THINGS

The IoT has gained significant attractions in recent years. IoT devices generally need to perform data processing tasks and communicate with each other through wireless radio. However, IoT devices face the challenges of the shortage of computing, memory and energy resources. In recent years, several custom RISC-V instruction set extensions have been proposed to reduce the energy consumption of floating-point computation [40] of IoT devices, and improve the efficiency of digital signal processing (DSP) [5], [15].

Most IoT applications involve computations of data with floating-point formats. Optional formats include quad precision (binary128), double precision (binary64), single precision (binary32), and half precision (binary16), etc. The energy consumption of different precision floating point operations is different. There is a trade-off between computation energy consumption and computation precision. Recently, there is an emerging trend that focuses on adapting the floating point format according to the IoT application's requirements for computation precision and energy consumption. However, the current RISC-V official ISA specification only provides binary32, binary64 and binary128 instruction set extensions, and lacks support for lower precision floating-point formats. Tagliavini *et al.* [40] proposed a set of smaller-than-32-bit floating-point (smallFloat) extensions for RISC-V. The above extensions provide support for smallFloat formats on embedded processors. The smallFloat formats include half-precision floating-point format, and two non-standard custom formats, namely alternative half-precision floating-point (binary16alt) and quarter-precision floating-point (binary8). The smallFloat extensions can effectively reduce IoT processor power consumption.

DSP is a key technology for IoT. IoT devices often use DSP technology for data processing (such as vibration data spectrum extraction) and wireless signal processing. DSP involves various complex algorithms that require a lot of computation, such as Fast Fourier Transformations (FFT) and Discrete Cosine Transform (DCT) algorithms for spectrum transformation; PID algorithms for control, etc. Gautschi *et al.* [15] proposed an embedded core with customized DSP extension based on RISC-V to increase the energy efficiency of IoT devices. The customized DSP extension provides dot-product, shuffle instructions to improve computational efficiency. The authors evaluated the performance of the RISC-V core in processing DSP algorithms such as FFT and FDCT, etc. This scheme is superior to the traditional scheme in terms of computing time and power consumption. Amor *et al.* [5] concentrate on the acceleration of complicated physical-layer wireless DSP algorithms (e.g., FFT computation, gain control) used in IoT. They proposed an extension including complex arithmetic instructions, reconfigurable multiplication instructions and automatic gain control instructions, to reduce the number of cycles required by DSP algorithms and reduce energy consumption.

B. ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is becoming an important engine to promote social progress and sustainable economic prosperity. In recent years, new AI methods such as Deep Neural Network (DNN) and Convolutional Neural Network (CNN) emerge endlessly. However, the demand for huge computing and storage resources of AI methods limits their performance improvement and wider application. Recently, several research have been done on customizing RISC-V instruction set extension to optimize performance of AI algorithms such as DNN [6], CNN [7], Graph Convolutional Network (GCN) [41], Transformer [42], etc.

Neural networks have a large number of weight param-

TABLE 1. Categorization of RISC-V ISA extension works.

Fields	Applications	Ref.	Year	Instruction Extensions
Internet of Things	Floating-point computation	[40]	2019	smaller-than-32-bit floating-point extensions
	Digital signal processing	[15]	2017	DSP extension including dot-product and shuffle instructions
		[5]	2021	FFT computation, gain control related extensions for IoT wireless signal processing
Artificial Intelligence	DNN accelaration	[6]	2021	DNN weights compression and decompression extensions
	CNN accelaration	[7]	2021	convolution operation instruction extension
	GCN accelaration	[41]	2022	vector aggregation and combination instruction extensions
	Transformer accelaration	[42]	2021	activation and softmax instruction extensions
Communication	Signal processing	[11]	2021	5G baseband signal processing instruction extensions
	Error correction coding	[43]	2021	error correction codes related instruction extensions
	Radio resource management	[44]	2020	tanh and sigmoid instruction extensions
Graphics Computing	SIMT execution	[9]	2020	SIMT instruction extensions
	Graphics Rendering	[45]	2020	graphics rendering instruction extension
Post Quantum Cryptography	Module-LWE cryptography	[46]	2021	CT-butterfly, GS, and barret/reduce32 reduction instruction extensions
	Ring-LWE cryptography	[47]	2020	SHA256, polynomial multiplication, chien search, and modulo reduction instruction extensions
	Code-based cryptography	[48]	2022	Galois field instruction extension
High Performance Computing	Global address space	[8]	2021	extended base global address space (xBGAS) extension
	Remote atomic	[49]	2020	remote atomic extension
Virtualization	Delegated virtualization	[50]	2022	delegated virtualization extension
Security	Memory attack protection	[51]	2017	a set of instructions which ensure correct operation of various hardware blocks
	Control-flow integrity	[52]	2022	forward control flow integrity instruction extension
	Control-flow attacks protection	[53]	2021	instruction extensions to support always-encrypted code and code pointers

eters, which consume a lot of computing and storage resources. Neural network weight compression can accelerate the computation speed of neural networks, and is a typical method to optimize the performance of neural network models. The work [6] proposed a posit processing unit (PPU) based on RISC-V and customized an ISA extension to perform DNN weights compression utilizing the posit format to speed up DNN inference. This extension enables an efficient conversion between 8 or 16-bit posits and 32-bit IEEE floats or fixed point formats with very little precision degradation. As shown in Figure 5, the PPU can be integrated into a RISC-V core alongside the FPU and the ALU in an non-disruptive approach and accelerate sample inference time by 10 times.

In addition to compressing the weights, accelerating the computing-intensive part of different neural networks based on their computational characteristics is also a current research hotspot. For instance, a time-consuming process in CNN is the convolution operation. However, the matrix convolution between the CNN kernel and the input data is not capable of being done quickly or efficiently using the conventional RISC-V ISA. To solve this problem, work [7] customized a RISC-V extension that contains a convolution instruction to perform a convolution operation. This extension realized convolution operation within one single instruction, which greatly reduced the computing time. In addition to CNN, RISC-V extensions for the emerging GCN and Transformer neural network models have also been studied. The work [41] designed a hardware accelerator based

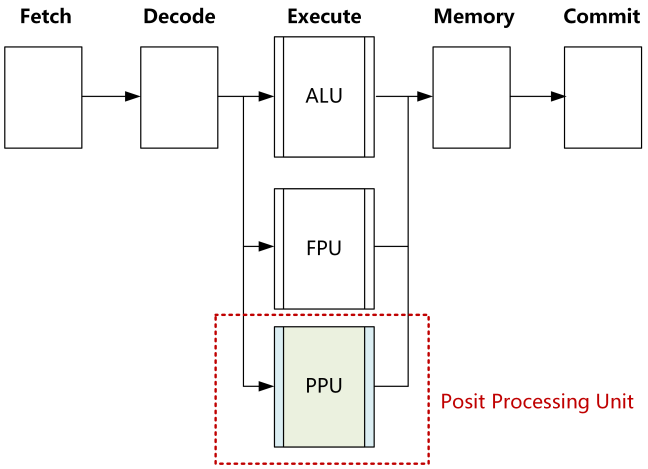


FIGURE 5. PPU can be integrated into a RISC-V core alongside the FPU and the ALU in an non-disruptive approach [6].

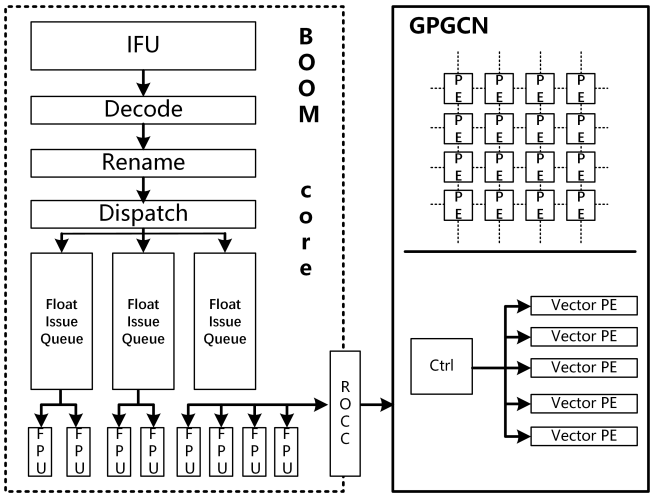


FIGURE 6. The architecture of GPGCN accelerator [41].

on custom instructions to accelerate general-purpose GCNs (GPGCNs). As shown in Figure 6, The GPGCN accelerator contains several process elements (PEs) and can be connected to the RISC-V core (such as BOOM) via the ROCC interface. The RISC-V core pipeline sends the GPGCN instructions to the GPGCN accelerator for processing. The authors analyzed the computational characteristics of GCN and points out that the main computational bottlenecks of GCNs consist of two aspects: feature vector aggregation and combination. In addition, GCN calculation also involves a large number of memory accesses and synchronization between GPGCN accelerator instruction stream and the CPU instruction stream. The authors customized several instruction extensions to speed up GCN calculations. They proposed vector and matrix operation instructions for feature aggregation and combination. Besides, they also designed memory-access instructions and some special instructions for GCN forward inference memory access, and synchronization. The above instruction extensions provide effective acceleration for various GCNs. The work [42] customized the activation instruction and softmax instruction extensions to improve performance efficiency for ReLU activation functions and Softmax functions commonly used in the Transformer.

C. COMMUNICATION

Wireless communication technologies are widely used in our lives. The increasing number of various wireless devices and people's demand for low latency and high bandwidth communication have led to the standardization of the fifth generation (5G) wireless communication technology. Sixth-generation (6G) wireless communication technology is expected to place greater emphasis on latency, throughput, and intelligence than its 5G predecessor. The adaptability and scalability of wireless communication infrastructure are under more scrutiny as a result of this trend. Application-specific instruction-set processors (ASIPs) offer a great balance of flexibility, affordability, and efficiency, and are expected to become one of the enabling technologies of communication processing hardware. The RISC-V offers designers an alternative that permits customization and innovation in communication ASIPs. Recent works have designed signal processing [10], [11], error correction coding [43], and radio resource management [44], [54] RISC-V ISA extensions to build flexible and efficient communication ASIPs.

Signal processing technologies (such as frequency division multiplexing, beamforming, etc.) are the basic technologies of communication systems, and signal processing is usually implemented using ASICs in current communication systems. However, recent radio access network virtualization technologies, such as Open RAN and vRAN, have encouraged telecom operators to implement signal processing using ASIPs with greater flexibility and programmability. The work [10] implements generalized frequency division multiplexing based on RISC-V vector extension, demonstrating the potential of RISC-V in signal processing. A typical signal processing ASIP based on RISC-V custom extensions is Edge Q [11]. The Edge Q designs more than 50 custom instructions and extensions for baseband signal processing, such as fast fourier transform, complex modulation, equalization and matrix decomposition, etc. The Edge Q makes it possible to dynamically programme a radio unit, distributed unit, or small cell.

Error correction code is a vital part of communication systems. Error correction code (such Turbo and LDPC codes) can improve the reliability of signal transmission and improve the transmission quality of communication system. However, error-correcting codes have high computational complexity and irregular memory accesses [55], [56]. In order to lower hardware costs and accommodate numerous error correction codes (Turbo, LDPC, Polar codes, etc.) that may be changed dynamically, the work [43] extends RISC-V and creates a dynamically re-configurable ASIP. The instruction extensions include absolute value instruction, addition instruction, subtraction instruction, multiplication

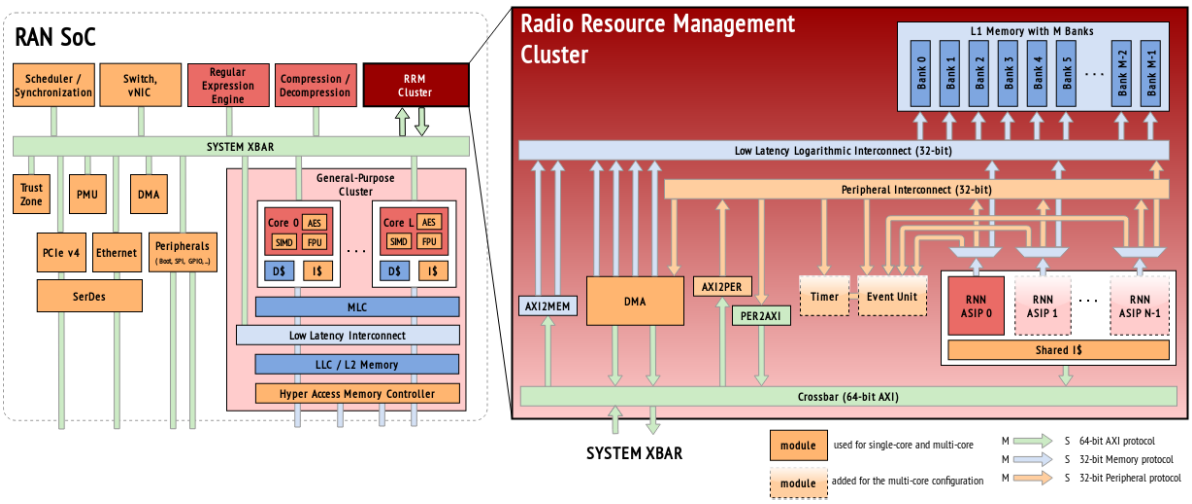


FIGURE 7. Communication ASIP RNN subsystem overview of including its integration in a typical RAN SoC [44], [54].

instruction and minimum instruction, etc., to accelerate the algorithms such as maxlog-MAP, Successive-Cancellation and Min-Sum used by error correction codes.

Radio resource management is challenging because it needs to make the best use of the limited available frequency band under the conditions of heterogeneous traffic and rapidly varying radio propagation conditions [58]. Recently, artificial intelligence has gained increasing attention for 5G radio resource management since it can realize intelligent prediction and dynamic decision-making [59]. There are some works extending the RISC-V ISA to accelerate AI-based radio resource management. As shown in Figure 7, the works [44] and [54] extend the RISC-V ISA for efficient RNN-based 5G radio resource management, and introduce a ASIP RNN subsystem, which is integrated into a typical radio access network (RAN) SoC. The extensions proposed in these works include tanh and sigmoid instructions to accelerate the nonlinear activation functions, and a load and computing instruction to load data and calculate the 16-bit packed SIMD sum-dot-product. The above works effectively speeds up RNN calculation in radio resource management.

D. GRAPHIC COMPUTING

Graphic computing is widely used in multimedia applications such as image and video processing, etc. In recent years, various GPUs have been proposed to speed up graphic computing. RISC-V has the advantages of extensibility, which provides a new choice for GPU implementation. However, RISC-V has no standard extensions for graphic computing. Recent works have designed Single Instruction Multiple-Thread (SIMT) execution [9], [57] and graphics rendering [45] instruction set extensions for efficient graphics computing.

To take full use of data-parallel multi-threading and boost throughput while keeping power consumption to a minimum, GPUs have widely supported SIMT execution model. Elsabbagh *et al.* [9] designed a general-purpose GPU Vortex using RISC-V. In order for Vortex to perform the SIMT execution, this work introduced five new instructions to RV32IM: warps spawn, threads activate, control-flow split and join, and barrier instructions. Furthermore, they also modified the software stack so that OpenCL programs can be executed in Vortex. After that, Tine *et al.* [57] continued this work and further enhanced Vortex. They realized the SIMT model and 3D graphics acceleration by using six new instructions. Then they explored the appropriate Vortex microarchitecture design on FPGA, and focused on elastic pipeline design and high bandwidth caches. Figure 8 shows the components of the enhanced Vortex microarchitecture which are augmented by the SIMT hardware components, including hardware wavefront scheduler, banked GPRs, high bandwidth caches, etc. In terms of software, they further modified the software stack to make Vortex support OpenCL and OpenGL. In addition to SIMT execution instruction set extension, Zhou *et al.* [45] proposed a RISC-V graphics rendering instruction set extension, and designed the corresponding SoC. This SoC can be used in AI chips for graphic rendering. Although the frament rendering process is designed as a fixed-pipeline, it can greatly boost performance while consuming less power compared with MCU chip with GPU core.

E. POST QUANTUM CRYPTOGRAPHY

Public-key cryptography (PKC) is the cornerstone of modern secure information and communication systems. PCK's security is dependent on complex mathematical problems which are difficult to solve with current electronic computers. However, with the development of powerful and reliable quantum computers, the majority of the current public key cryptography methods (such as RSA, elliptic curve) would be rendered ineffective. Only methods that can withstand the attack of quantum computers can survive after entering the era of quantum computing, which is called post quantum cryptography (PQC). Numerous PQC methods have been put out in recent years to resist quantum computer attacks,

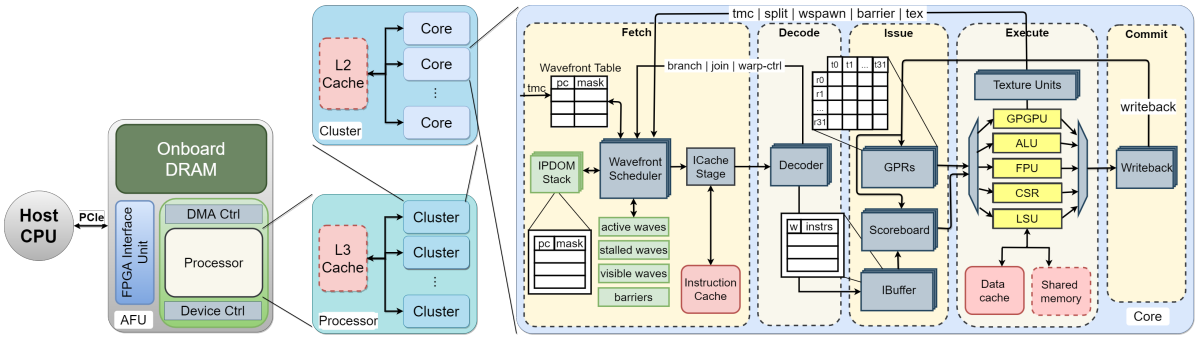


FIGURE 8. Vortex GPGPU microarchitecture [57].

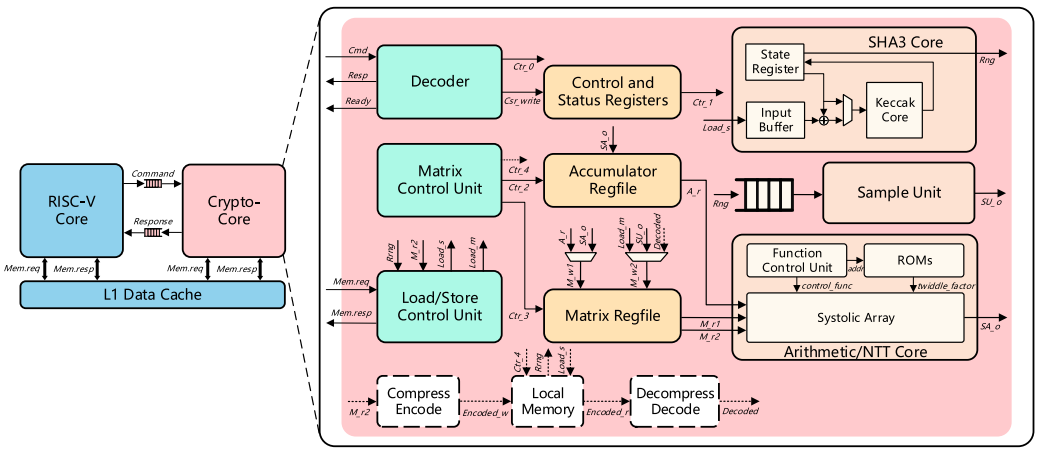


FIGURE 9. The domain-specific architecture for Module-LWE-based post quantum cryptography of [60].

including lattice-based PKC, code-based PKC, and isogeny-based PKC.

The lattice-based cryptography is regarded as one of the top contenders for PQC standards. However, the resources needed to run Lattice-based cryptography algorithms are not trivial. To fulfil the demands of high throughput and low latency in modern applications (such as 5G), several RISC-V instruction set extensions are proposed to accelerate these algorithms. Learning with error (LWE) problems are the foundation of lattice-based cryptography techniques. Typical LWE problems include Module-LWE and Ring-LWE.

Schemes based on module-LWE include Kyber and Dilithium. Nannipieri *et al.* [46] analyze Kyber and Dilithium algorithms and find that the Number Theoretic Transform (NTT) and modular reductions are the bottlenecks that need to be hardware accelerated. They proposed a Post-Quantum (PQ) RISC-V ISA extension includes CT-butterfly and GS instructions to accelerate NTT, and barret/reduce32 reduction instructions to accelerate modular reductions. Zhao *et al.* [60] proposed a domain-specific accelerator (DSA) with the goal of offering security services for 5G edge computing, as illustrated in Figure 9. In order to accelerate matrix-based NTT and arithmetic operations, the proposed DSA customized a RISC-V matrix extension. The DSA supports both Kyber and Dilithium algorithms.

Ring-LWE based schemes include LAC and NewHope, etc. Fritzmann *et al.* [47] extend the RISC-V ISA for acceleration of LAC. They focus on three performance bottlenecks of LAC: the polynomials generation, error correction, and polynomials multiplication. They proposed four RISC-V instructions (including SHA256, polynomial multiplication, chien search, and modulo reduction instructions) and corresponding hardware designs to accelerate these performance bottlenecks. Xin *et al.* [61] propose a vector processor based on RISC-V to speed up NTT of NewHope, LAC and Kyber. They designed a vector NTT unit and the corresponding customized RISC-V instruction extension to support the vectorization and parallel computation of NTT.

In addition to the research on lattice-based scheme, Kuo *et al.* [48] investigated RISC-V ISA extensions on a code-based scheme named Classic McEliece cryptography. They proposed a Galois field ISA extension to accelerate the Classic McEliece cryptography.

F. HIGH PERFORMANCE COMPUTING

Recently, RISC-V ISA has been expanding and penetrating into the high-performance computing (HPC) industry and is expected to be used in applications such as weather

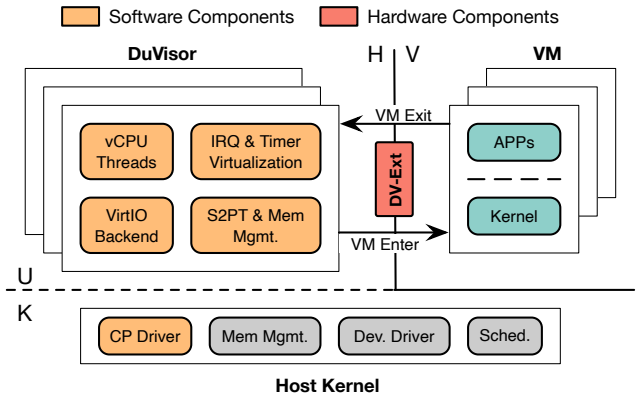


FIGURE 10. The architecture overview of DuVisor [50].

forecasting, car crash simulation, material simulation, etc. Current research on the extension of RISC-V in HPC scenarios include global address space extension [8] and remote atomic extension [49]. Wang *et al.* [8] proposed an Extended Base Global Address Space (xBGAS) extension. Through the addition of ISA support for remote direct shared memory accesses, the xBGAS extension helps to cut down on the unnecessary software overhead that is associated with inter-node connections. However, the xBGAS extension does not provide support for cross-node atomic operations, which is an essential feature that is being sought after by emerging HPC applications. In order to fulfil the needs for global atomicity, a remote atomic extension (RAE) that is based on the xBGAS architecture has been proposed by work [49]. This RAE makes use of high-performance remote atomic operations in order to access objects that are spread across many shared data stores.

G. VIRTUALIZATION

Although the RISC-V Foundation has formulated the H-extension specification to support virtualization, RISC-V virtualization technology is still in its infancy and needs further improvement. RISC-V provides user level interruption and Physical Memory Protection (PMP) mechanisms. These new hardware features promote the research of virtualization security. Chen *et al.* [50] proposed delegated virtualization and developed the user-level hypervisor DuVisor on RISC-V. Depicted in Figure 10, DV-Ext is a custom delegated virtualization extension that contains a set of additional registers and corresponding instructions. DuVisor can use the hardware features provided by DV-Ext to directly handle VM operations in user mode without trapping into the kernel, reducing virtualization overhead.

H. SECURITY

The number of worldwide networking devices is rising quickly along with the development of the 5G and Internet of Things, but there are also a lot of security concerns. Researchers have put a lot of effort into developing security solutions to address security risks of RISC-V.

The work [12] gave a thorough survey of RISC-V security methods and provided a summary of the key security mechanisms of RISC-V, including side-channel attack resistance, control flow integrity strengthening, system attack surface reduction, etc. In addition to the research mentioned in [12], there are several new research on security ISA extension. Menon *et al.* [51] developed a RISC-V 64 bit processor Shakti-T, which supports a customized ISA extension containing 8 new instructions for temporal and spatial memory attack protection. Das *et al.* [62] developed another RISC-V processor Shakti-MS. They proposed two new instructions and simply modified the compiler, which can automatically insert these new instructions to realize memory protection during hardware operation. Unlike [51], Shakti-MS does not need additional tables or tag bits to store pointer metadata, which can effectively reduce hardware complexity and storage overhead. Park *et al.* [52] proposed Bratter for forward control flow integrity solution. Batter contains a dedicated CSR Register Branch Tag register and two new instructions for this register. This work uses Bratter to implement branch regulation and function signature check to evaluate its performance. The experimental results show that even if the two solutions work together, the code size and execution time will only increase by 1.20% and 5.99%. Harris *et al.* [53] developed Morpheus II for control flow attack protection. Morpheus II resists attacks by implementing code pointers and code encryption. The processor architecture of Morpheus II is shown in Figure 11. The encrypted code from the I-Cache is decrypted during the fetch stage. The code pointers are decrypted in the execute stage, and encrypted in the writeback stage. It adds code pointers encryption related ISA extensions to RISC-V, including arithmetic instruction for encrypted pointer ALU operations, relational instruction for the relational tests of decrypted pointer, etc. Morpheus II uses the new RISC-V ISA extensions to implement the always-encrypted code pointer, which makes it difficult for attackers to attack by forging code pointers. In addition, Morpheus II also implements always-encrypted code. The encryption code pointer and code have a little impact on performance while effectively against attacks, and the area overhead is less than 2%.

V. FUTURE OPPORTUNITIES

Despite much effort on RISC-V ISA extensions, further research is needed to enhance the functions, improve the performance and enrich the application scenarios of RISC-V. In this section, we highlight some possible future research opportunities.

A. CLOUD COMPUTING

In recent years, RISC-V has achieved extraordinary success in the field of Internet of things, but it is still in its infancy in the field of cloud computing. At present, there have been some high-performance RISC-V CPU research, such as the Xiangshan processor of the Chinese Academy

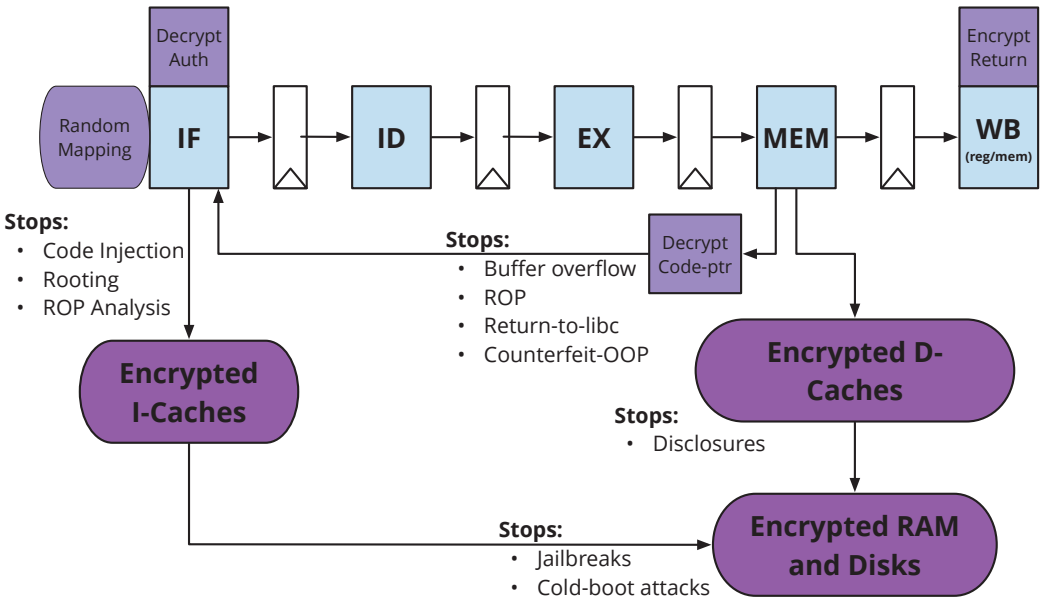


FIGURE 11. Processor Architecture of Morpheus II [53].

of Sciences, which shows the application potential of RISC-V in cloud computing. However, there is still a big gap for RISC-V to meet the needs of cloud computing-level CPUs. For example, hypervisor instruction extensions need to be further improved. Hypervisor technology is widely used in cloud computing, but according to the RISC-V Hypervisor SIG report, RISC-V hypervisor extension needs to be further improved in terms of I/O virtualization, nested virtualization, and virtual machine QoS assurance. In addition to hypervisor extension, it is also a feasible direction to research customized instruction sets for cloud computing applications to accelerate applications. In addition to hypervisor extension, it is also a feasible research direction to define ISA extensions to accelerate specific applications of cloud computing. For example, x86 provides vector AES, integer fused multiple add, Galois field instruction set extensions. Alibaba OpenAnolis operating system adapts these instructions to accelerate SSL/TLS of software (such as OpenSSL, Nginx, DPDK, Envoy) of cloud native gateway. RISC-V can target x86 and also provides corresponding instruction set extensions. RISC-V foundation has also set up Datacenter SIG to identify and address gaps between RISC-V and the needs of cloud computing.

B. TRUSTED COMPUTING

There is a growing need for trusted computing architecture in hardware, where security-critical applications can both examine the environment they are running in, and be isolated from other untrusted applications. The hardware giants have unveiled a series of trusted computing products like Intel SGX [63], AMD SEV [64] and ARM TrustZone [65] to host security-critical applications. In 2021, the RISC-V working group established the Trusted Computing SIG to investigate the typical hardware-assisted trusted computing technologies such as enclaves, confidential VM, hardware TEE, etc., and define the trusted computing extensions for RISC-V.

Secure hardware enclave is a popular technology widely used for protecting applications in the clouds. How to improve the scalability of enclave systems and meet the requirements of large scale microservice and serverless computing is an urgent issue. There are mainly three scalability limitations of the existing enclave systems, that is non-scalable memory partition/isolation, non-scalable memory integrity protection, and non-scalable secure memory initialization [66]. Feng *et al.* [66] proposed Penglai, which is a scalable TEE system based on RISC-V. Penglai system has implemented a new RISC-V ISA extension: s-mode physical memory protection (sPMP) extension, which allows scalable physical memory isolation in TEE OS or secure monitor. Generally speaking, research in this field is still limited, how to extend RISC-V ISA to build efficient enclave systems supporting esilient memory allocation, high resource utilization, auto-scaling of security-critical applications requires further efforts.

In addition to the enclave, trusted VM for RISC-V is also a potential research direction recently. Traditional trusted VMs have the problem of tight coupling between hardware virtualization and kernel mode, such as relying on the kernel to drive hardware extensions [50]. This problem creates an unnecessary performance-security tradeoff. RISC-V can expose physical resources to user mode that used to be managed only by the kernel. This feature is expected to improve the performance of trusted VMs.

C. IN-MEMORY COMPUTING

The von Neumann principle is the foundation of the majority of contemporary CPU designs. However, The frequent data fetching and movement between processing units and memory in the von Neumann architecture restrict processor performance. In-memory computing is one of the most promising alternatives to solve this problem. By integrating memory and processing units, in-memory computing makes it possible for data to be processed immediately inside the memory, hence minimising data traffic. Recently, there have been some attempts to apply RISC-V to the field of in-memory computing. For instance, Garofalo *et al.* [67] proposed a heterogeneous RISC-V processor with multiple cores and an in-memory computing accelerator for DNN inference. However, this work does not extend RISC-V to better support in-memory computing. Coluccio *et al.* proposed an in-memory computing framework based on a RISC-V processor that enables logic-in-memory (LiM) computing. They replaced data memory with a circuit that can both store data and execute calculations in-memory. In particular, they customized a LiM instruction set extension to support programming of the memory. However, the research on in-memory computing instruction set extension is still in its early stage. How to design in-memory computing processors based on RISC-V and customize more efficient instruction set extensions need further research.

D. IN-NETWORK COMPUTING

Over the past few years, a new research field called in-network computing has arisen. It shifts the burden of computation away from the host CPU and onto new programmable network devices (such as programmable switches, network processors, smart network cards) in order to increase network throughput, decrease processing delay, and cut energy consumption. Although in-network computing has significant advantages, how to design an efficient ISA that enables developers to write their applications on in-network computing devices is still a problem to be solved. Sankaran *et al.* [68] proposed a safe in-network computing framework for Industrial IoT. In order to enable the in-network computing of IoT devices, they designed an ISA based on RISC, including instructions for logical operations, elementary arithmetic, and regular expressions. This work provides valuable references for customizing ISA for in-network computing. RISC-V is also a competitive ISA for the implementation of in-network computing [69]. Additional research is still required to investigate how RISC-V instruction set extensions can be designed to facilitate in-network computing.

E. RECONFIGURABLE COMPUTING

The traditional CPU architecture is based on instruction-driven computing mode. A large proportion of the energy is consumed in the instruction overhead (such as instruction fetching, decoding and register renaming), and only a small part of the energy is actually used for computing. Reconfigurable computing processors such as dataflow-inspired processors [70] can realize instruction reuse, that is, repeatedly execute the same program instructions and bypass the costly instruction fetching and decoding processes, which can improve performance and lower energy consumption. How to extend RISC-V ISA to efficiently support reconfigurable computing architecture is also a problem that needs further research.

VI. CONCLUSIONS

In recent years, RISC-V has experienced tremendous success in the area of embedded processors such as IoT, and has gradually emerged in the field of domain specific processors and high-performance processors. Despite such great achievements of RISC-V, the RISC-V instruction set still needs to be extended and enhanced for diverse and complex computing scenarios.

A thorough analysis of the RISC-V ISA extensions' current status is presented in this paper. First, we described the main application scenarios of RISC-V (i.e., embedded processor, domain specific processor and high-performance processor), and discussed the requirements of each scenario for instruction set extension. Secondly, we investigated the status of current RISC-V extension standards and related research. In terms of RISC-V ISA extension standard, we introduced in detail which extensions are included in the current RISC-V ISA standard and the functions of each extension. In terms of RISC-V extension research, we reviewed the contents and contributions of RISC-V instruction set extension literatures about Internet of Things, artificial intelligence, graphic computing, high-performance computing, communications, etc. Finally, we anticipate RISC-V instruction set extension research opportunities in cloud computing, trusted computing, In-network computing and other fields.

REFERENCES

[1] K. Asanović and D. A. Patterson, "Instruction sets should be free: The case for RISC-V," Dept. EECS, Univ. California, Berkeley, Tech. Rep. UCB/EECS-2014-146, Aug. 2014. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-146.html

[2] The european processor initiative (EPI). accessed on Jan. 1, 2023. [Online]. Available: https://eurohpc-ju.europa.eu/participate/our-projects/european-processor-initiative-epi_en

[3] China RISC-V Alliance. accessed on Jan. 1, 2023. [Online]. Available: http://crva.ict.ac.cn/

[4] SEMICO Research Corporation, "RISC-V market analysis: The new kid on the block." Nov. 2019. [Online]. Available: https://semico.com/sites/default/files/TOC_CC315-19.pdf

[5] H. B. Amor, C. Bernier, and Z. Přikryl, "A RISC-V ISA extension for ultra-low power IoT wireless signal processing," *IEEE Trans. Comput.*, vol. 71, no. 4, pp. 766–778, Aug. 2021.

[6] M. Cococcioni, F. Rossi, E. Ruffaldi, and S. Saponara, "A lightweight posit processing unit for RISC-V processors in deep neural network applications," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 4, pp. 1898–1908, Oct. 2022.

[7] S. Wang, J. Zhu, Q. Wang, C. He, and T. T. Ye, "Customized instruction on RISC-V for winograd-based convolution acceleration," in *Proc. IEEE 32th Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, July 2021, pp. 65–68.

[8] X. Wang, J. D. Leidel, B. Williams, A. Ehret, M. Mark, M. A. Kinsy, and Y. Chen, "xBGAS: A global address space extension on RISC-V for high performance computing," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2021, pp. 454–463.

[9] F. Elsabbagh, B. Tine, P. Roshan, E. Lyons, E. Kim, D. E. Shim, L. Zhu, S. K. Lim, and H. Kim, "Vortex: OpenCL compatible RISC-V GPGPU," 2020, *arXiv:2002.12151*.

[10] V. Razilov, E. Matúš, and G. Fettweis, "Communications signal processing using RISC-V vector extension," in *Proc. Int. Wireless Commun. Mobile Comput (IWCMC)*, May 2022, pp. 690–695.

[11] S. Rajagopal, "EDGE Q 5G with an EDGE," in *Proc. IEEE Hot Chips 33 Symposium (HCS)*, Aug. 2021, pp. 1–13.

[12] T. Lu, "A survey on RISC-V security: Hardware and architecture," 2021, *arXiv:2107.04175*.

[13] 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 Access*, vol. 10, pp. 51 394–51 411, May 2022.

[14] A. Dörflinger, M. Albers, B. Kleinbeck, Y. Guan, H. Michalik, R. Klink, C. Blochwitz, A. Nechi, and M. Berekovic, "A comparative survey of open-source application-class RISC-V processor implementations," in *Proc. ACM International Conference on Computing Frontiers (CF)*, May 2021, pp. 12–20.

[15] M. Gautschi, P. D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gürkaynak, and L. Benini, "Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2700–2713, Feb. 2017.

[16] C. Palmiero, G. Di Guglielmo, L. Lavagno, and L. P. Carloni, "Design and implementation of a dynamic information flow tracking architecture to secure a RISC-V core for IoT applications," in *Proc. IEEE High Performance Extreme Computing Conference (HPEC)*, Sept. 2018, pp. 1–7.

[17] L. Lu, M. Zhang, and D. He, "Design and implementation of a smart home system based on the RISC-V processor," in *Proc. IEEE International Conference on Civil Aviation Safety and Information Technology (ICCASIT*, Oct. 2020, pp. 300–304.

[18] S.-Y. Lee, Y.-W. Hung, Y.-T. Chang, C.-C. Lin, and G.-S. Shieh, "RISC-V CNN coprocessor for real-time epilepsy detection in wearable application," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 4, pp. 679–691, June 2021.

[19] M. Eggimann, S. Mach, M. Magno, and L. Benini, "A RISC-V based open hardware platform for always-on wearable smart sensing," in *Proc. IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, June 2019, pp. 169–174.

[20] S. Sordillo, A. Cheikh, A. Mastrandrea, F. Menichelli, and M. Olivieri, "Customizable vector acceleration in extreme-edge computing: a RISC-V software/hardware architecture study on VGG-16 implementation," *Electronics*, vol. 10, no. 4, p. 518, Feb. 2021.

[21] A. Cheikh, S. Sordillo, A. Mastrandrea, F. Menichelli, G. Scotti, and M. Olivieri, "Klessydra-T: Designing vector coprocessors for multi-threaded edge-computing cores," *IEEE Micro*, vol. 41, no. 2, pp. 64–71, Jan. 2021.

[22] Rocket chip generator. accessed on Jan. 1, 2023. [Online]. Available: https://github.com/chipsalliance/rocket-chip

[23] Source files for SiFive's Freedom platforms. accessed on Jan. 1, 2023. [Online]. Available: https://github.com/sifive/freedom

[24] SweRV RISC-V core from Western Digital. accessed on Jan. 1, 2023. [Online]. Available: https://github.com/westerndigitalcorporation/swerv_eh1

[25] J. Hennessy and D. Patterson, "A new golden age for computer architecture: domain-specific hardware/software co-design, enhanced," in *Proc. ACM/IEEE International Symposium on Computer Architecture (ISCA)*, July 2018, pp. 27–29.

[26] J. Nickolls and W. J. Dally, "The GPU computing era," *IEEE micro*, vol. 30, no. 2, pp. 56–69, Apr. 2010.

[27] I. Burstein, "Nvidia data center processing unit (DPU) architecture," in *Proc. IEEE Hot Chips 33 Symposium (HCS)*, Aug. 2021, pp. 1–20.

[28] A. Kuusela and C. Smullen, "Video coding unit (VCU): Hot chips 2021," in *Proc. IEEE Hot Chips 33 Symposium (HCS)*, Aug. 2021, pp. 1–30.

[29] F. Zaruba, F. Schuiki, and L. Benini, "Manticore: A 4096-core RISC-V chiplet architecture for ultraefficient floating-point computing," *IEEE Micro*, vol. 41, no. 2, pp. 36–42, Dec. 2020.

[30] C. Chen, X. Xiang, C. Liu, Y. Shang, R. Guo, D. Liu, Y. Lu, Z. Hao, J. Luo, Z. Chen *et al.*, "Xuantie-910: Innovating cloud and edge computing by RISC-V," in *Proc. IEEE Hot Chips 32 Symposium (HCS)*, Aug. 2020, pp. 1–19.

[31] B. Perez, A. Fell, and J. D. Davis, "Coyote: An open source simulation tool to enable RISC-V in HPC," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Feb. 2021, pp. 130–135.

[32] BOOM: The Berkeley out-of-order RISC-V processor. accessed on Jan. 1, 2023. [Online]. Available: https://github.com/riscv-boom

[33] Open-source high-performance RISC-V processor. accessed on Jan. 1, 2023. [Online]. Available: https://github.com/OpenXiangShan/XiangShan

[34] "Ieee standard for floating-point arithmetic," *IEEE Std 754-2008*, pp. 1–70, 2008.

[35] C. Chen, X. Xiang, C. Liu, Y. Shang, R. Guo, D. Liu, Y. Lu, Z. Hao, J. Luo, Z. Chen, C. Li, Y. Pu, J. Meng, X. Yan, Y. Xie, and X. Qi, "Xuantie-910: A commercial multi-core 12-stage pipeline out-of-order 64-bit high performance RISC-V processor with vector extension : Industrial product," in *Proc. ACM/IEEE International Symposium on Computer Architecture (ISCA)*, June 2020, pp. 52–64.

[36] V. Maisto and A. Cilardo, "A pluggable vector unit for RISC-V vector extension," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar. 2022, pp. 1143–1148.

[37] I. A. Assir, M. E. Iskandarani, H. R. A. Sandid, and M. A. R. Saghir, "Arrow: A RISC-V vector accelerator for machine learning inference," 2021, *arXiv:2107.07169*.

[38] B. Sá, J. Martins, and S. Pinto, "A first look at RISC-V virtualization from an embedded systems perspective," *IEEE Trans. Comput.*, vol. 71, no. 9, pp. 2177–2190, Sep. 2021.

[39] J. Martins and S. Pinto, "Bao: a modern lightweight embedded hypervisor," in *Proc. Embedded World Conference*, Feb. 2020.

[40] G. Tagliavini, S. Mach, D. Rossi, A. Marongiu, and L. Benini, "Design and evaluation of SmallFloat SIMD extensions to the RISC-V ISA," in *Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Mar. 2019, pp. 654–657.

[41] W. Tang and P. Zhang, "GPGCN: A general-purpose graph convolution neural network accelerator based on RISC-V ISA extension," *Electronics*, vol. 11, no. 22, p. 3833, Nov. 2022.

[42] Q. Jiao, W. Hu, F. Liu, and Y. Dong, "RISC-VTF: RISC-V based extended instruction set for transformer," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2021, pp. 1565–1570.

[43] M. Tourres, C. Chavet, B. Le Gal, J. Crenne, and P. Coussy, "Extended RISC-V hardware architecture for future digital communication systems," in *Proc. IEEE 5G World Forum (5GWF)*, Oct. 2021, pp. 224–229.

[44] R. Andri, T. Henriksson, and L. Benini, "Extending the RISC-V ISA for efficient RNN-based 5G radio resource management," in *Proc. ACM/IEEE Design Automation Conference (DAC)*, Apr. 2020, pp. 1–6.

[45] Y. Zhou, X. Jin, and T. Xiang, "RISC-V graphics rendering instruction set extensions for embedded AI chips implementation," in *Proc. International Conference on Big Data Engineering and Technology (BDET)*, Jan. 2020, pp. 85–88.

[46] P. Nannipieri, S. Di Matteo, L. Zulberti, F. Albicocchi, S. Saponara, and L. Fanucci, "A RISC-V post quantum cryptography instruction set extension for number theoretic transform to speed-up CRYSTALS algorithms," *IEEE Access*, vol. 9, pp. 150 798–150 808, Nov. 2021.

[47] T. Fritzmann, G. Sigl, and J. Sepúlveda, "Extending the RISC-V instruction set for hardware acceleration of the post-quantum scheme LAC," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar. 2020, pp. 1420–1425.

[48] Y.-M. Kuo, F. Garcia-Herrero, O. Ruano, and J. A. Maestro, "RISC-V galois field ISA extension for non-binary error-correction codes and classical and post-quantum cryptography," *IEEE Trans. Comput.*, early access, May 2022, doi: 10.1109/TC.2022.3174587.

[49] X. Wang, B. Williams, J. D. Leidel, A. Ehret, M. Kinsy, and Y. Chen, "Remote atomic extension (RAE) for scalable high performance computing," in *Proc. ACM/IEEE Design Automation Conference (DAC)*, July 2020, pp. 1–6.

[50] J. Chen, D. Li, Z. Mi, Y. Liu, B. Zang, H. Guan, and H. Chen, "Duvisor: a user-level hypervisor through delegated virtualization," 2022, *arXiv:2201.09652*.

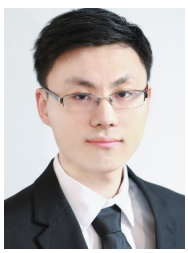
[51] A. Menon, S. Murugan, C. Rebeiro, N. Gala, and K. Veezhinathan, "Shakti-t: A RISC-V processor with light weight security extensions," in *Proc. Hardware and Architectural Support for Security and Privacy (HASP)*, June 2017, pp. 1–8.

[52] S. Park, D. Kang, J. Kang, and D. Kwon, "Bratter: An instruction set extension for forward control-flow integrity in RISC-V," *Sensors*, vol. 22, no. 4, p. 1392, Feb. 2022.

[53] T. Austin, A. Harris, T. Verma, S. Wei, A. Kisil, M. Aga, V. Bertacco, B. Kasikci, and M. Tiwari, "Morpheus II: A RISC-V security extension for protecting vulnerable software and hardware," in *Proc. IEEE Hot Chips 33 Symposium (HCS)*, Aug. 2021, pp. 1–18.

[54] G. Paulin, R. Andri, F. Conti, and L. Benini, "RNN-based radio resource management on multicore RISC-V accelerator architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 9, pp. 1624–1637, July 2021.
[55] H. Harb and C. Chavet, "Fully parallel circular-shift rotation network for communication standards," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 12, pp. 3412–3416, May 2020.
[56] C. Chavet, F. Lozachmeur, T. Barguil, A. Hussein, and P. Coussy, "Solving memory access conflicts in LTE-4G standard," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2019, pp. 1518–1522.
[57] B. Tine, K. P. Yalamarthy, F. Elsabbagh, and K. Hyesoon, "Vortex: Extending the RISC-V ISA for GPGPU and 3D-Graphics," in *Proc. IEEE/ACM International Symposium on Microarchitecture (Micro)*, Oct. 2021, pp. 754–766.
[58] N. D. Tripathi, J. H. Reed, and H. F. VanLandingham, *Radio resource management in cellular systems*. Springer Science & Business Media, 2006, vol. 618.
[59] M. Yao, M. Sohul, V. Marojevic, and J. H. Reed, "Artificial intelligence defined 5G radio access networks," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 14–20, Mar. 2019.
[60] Y. Zhao, R. Xie, G. Xin, and J. Han, "A high-performance domain-specific processor with matrix extension of RISC-V for module-LWE applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, pp. 2871 – 2884, Apr. 2022.
[61] G. Xin, J. Han, T. Yin, Y. Zhou, J. Yang, X. Cheng, and X. Zeng, "VPQC: A domain-specific vector processor for post-quantum cryptography based on RISC-V architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 8, pp. 2672–2684, Apr. 2020.
[62] S. Das, R. H. Unnithan, A. Menon, C. Rebeiro, and K. Veezhinathan, "SHAKTI-MS: a RISC-V processor for memory safety in C," in *Proc. ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, June 2019, pp. 19–32.
[63] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution." in *Proc. HASP ISCA*, vol. 10, no. 1, June 2013.
[64] AMD, "Amd secure encrypted virtualization (sev)." 2019. [Online]. Available: https://developer.amd.com/sev/
[65] Z. Hua, J. Gu, Y. Xia, H. Chen, B. Zang, and H. Guan, "vTZ: Virtualizing ARM TrustZone," in *Proc. USENIX Security Symposium (USENIX Security)*, Aug. 2017, pp. 541–556.
[66] E. Feng, X. Lu, D. Du, B. Yang, X. Jiang, Y. Xia, B. Zang, and H. Chen, "Scalable memory protection in the PENGLAI enclave," in *Proc. USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, July 2021, pp. 275–294.
[67] A. Garofalo, G. Ottavi, F. Conti, G. Karunaratne, I. Boybat, L. Benini, and D. Rossi, "A heterogeneous in-memory computing cluster for flexible end-to-end inference of real-world deep neural networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 12, no. 2, pp. 422–435, Apr. 2022.
[68] G. C. Sankaran, K. M. Sivalingam, and H. Gondaliya, "P4 and netfpga based secure in-network computing architecture for ai-enabled industrial internet of things," *IEEE Internet Things J.*, early access, Nov. 2021, doi: 10.1109/JIOT.2021.3125862.
[69] S. Di Girolamo, A. Kurth, A. Calotoiu, T. Benz, T. Schneider, J. Beránek, L. Benini, and T. Hoefler, "A RISC-V in-network accelerator for flexible high-performance low-power packet processing," in *Proc. ACM/IEEE International Symposium on Computer Architecture (ISCA)*, June 2021, pp. 958–971.
[70] D. K. Wang and N. S. Kim, "DiAG: A dataflow-inspired architecture for general-purpose processors," in *Proc. ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Apr. 2021, p. 93–106.

ENFANG CUI (M'22) received the Ph.D. degrees in Communication and Information Systems from Beijing Jiaotong University, Beijing, China, in 2022. He is currently working as a researcher at the China Telecom Research Institute. His research interests include cloud/edge computing, computer network and architecture. He has published more than 10 papers in prestigious journals and conferences, including the IEEE Transactions on Green Communications and Networking, IEEE Vehicular Technology Magazine, and IEEE HPCC. He serves/served as a workshop co-organizer of IEEE EUC 2022, and a reviewer for IEEE Globecom, Wireless Network, Transactions on Emerging Telecommunications Technologies.



TIANZHENG LI received the B.S. degree in electrical engineering and information technology from the University of Applied Sciences and Arts of Hannover, Germany, in 2018. and the M.S. degree in computer engineering from the Gottfried Wilhelm Leibniz University Hannover, Germany, in 2021. He is currently working as a researcher at the China Telecom Research Institute. His research interests include cloud computing and RISC-V.

QIAN WEI received the B.S. degree in the school of computer science and technology from Xidian University, Xian, China in 2019, and the M.S. degree in the school of computer science and engineering from Sun Yat-Sen University, Guangzhou, China in 2022. Her research interests include cloud/edge computing and computer architecture.