RISC-V Server Platform Specification

Server Platform Task Group

Version v0.0.1, 2024-07-08: This document is in development. Assume everything can change. See http://riscv.org/spec-state for details.

Table of Contents

Preamble

This document is in the Development state

Assume everything can change. This draft specification will change before being accepted as standard, so implementations made to this draft specification will likely not conform to the future standard.

Copyright and license information

Contributors

This RISC-V specification has been contributed to directly or indirectly by (in alphabetical order):

Andrea Bolognani, Andrei Warkentin, Greg Favor, Ved Shanbhogue

Chapter 1. Introduction

The RISC-V Server Platform specification defines a standardized set of hardware and sofware capabilities, that portable system software, such as operating systems and hypervisors, can rely on being present in a RISC-V server platform.

A server is a computing system designed to manage and distribute resources, services, and data to other computers or devices on a network. It is often referred to as a 'server' because it serves or provides information and resources upon request. Such computing systems are designed to operate continually and have higher requirements for capabilities such as RAS, security, performance, and quality of service. Examples of servers include web servers, file servers, database servers, mail servers, game servers, and more. This specification focuses on defining requirements for general-purpose server computing systems that may be used for one or more of these purposes.

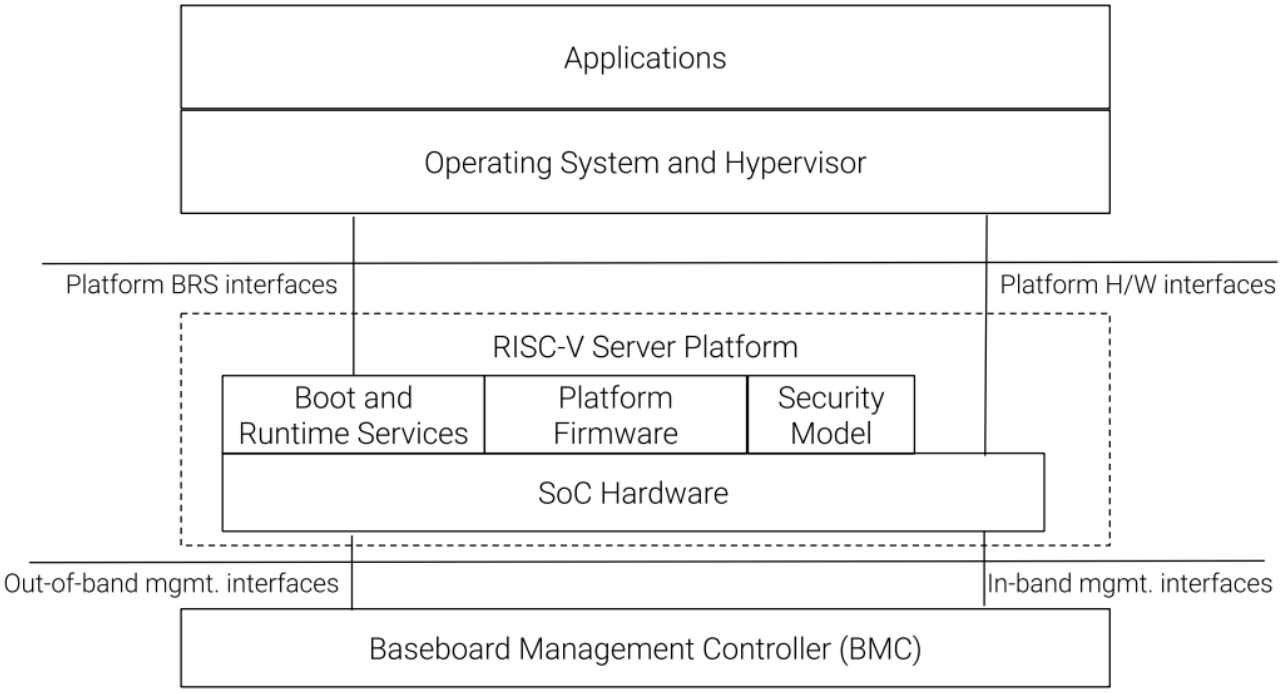

Figure 1. Components of a RISC-V Server Platform

The RISC-V server platform is defined as the collection of RVA profile-compliant application processor harts, SoC hardware, peripherals, platform firmware, boot/runtime services, and platform security services. The platform provides hardware interfaces (e.g., harts, timers, interrupt controllers, PCIe root ports, etc.) to portable system software. It also offers a set of standardized boot and runtime services based on the UEFI and ACPI standards. To support provisioning and platform management, it interfaces with a baseboard management controller (BMC) through both in-band and out-of-band (OOB) management interfaces. The in-band management interfaces support the use of standard manageability specifications like MCTP, PLDM, IPMI, and Redfish for provisioning and management of the operating system executing on the platform. The OOB interface supports the use of standard manageability specifications like MCTP, PLDM, Redfish, and IPMI for functions such as power management, telemetry, debug, and provisioning. The platform security model includes guidelines and requirements for aspects such as debug authorization, secure/measured boot, firmware updates, firmware resilience, and confidential computing, among others.

The platform firmware, typically operating at privilege level M, is considered part of the platform and is usually expected to be customized and tailored to meet the requirements of the SoC hardware (e.g., initialization of address decoders, memory controllers, RAS, etc.), boot/runtime services and platform security.

This specification standardizes the requirements for hardware and software interfaces and capabilities by building on top of relevant RISC-V standards, such as the RISC-V Architecture Profiles, Server SoC, Boot and Runtime Services and Platform Security specifications for server software executing on the application processor harts at privilege levels below M. It enables OS and hypervisor vendors to support such platforms with a single binary OS image distribution model. The requirements posed by this specification represent a standard set of infrastructural capabilities, encompassing areas where divergence is typically unnecessary and where novelty is absent across implementations.

To be compliant with this specification, the server platform MUST support all mandatory requirements and MUST support the listed versions of the specifications. This standard set of capabilities MAY be extended by a specific implementation with additional standard or custom capabilities, including compatible later versions of listed standard specifications. Portable system software MUST support the specified mandatory capabilities to be compliant with this specification.

The requirements in this specification use the following format:

ID#	**Requirement**
CAT_NNN	The `CAT` is a category prefix that logically groups the requirements and is followed by 3 digits - `NNN` - assigning a numeric ID to the requirement. The requirements use the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" that are to be interpreted as described in RFC 2119 [1] when, and only when, they appear in all capitals, as shown here. When these words are not capitalized, they have their normal English meanings.
A requirement or a group of requirements may be followed by non-normative text providing context or justification for the requirement. The non-normative text may also be used to reference sources that are the origin of the requirement.	

This specification groups the requirements in the following broad categories:

- Hardware
- Firmware
- Security

1.1. Glossary

Most terminology has the standard RISC-V meaning. This table captures other terms used in the document. Terms in the document prefixed by 'PCIe' have the meaning defined in the PCI Express (PCIe) Base Specification [2] (even if they are not in this table).

Table 1. Terms and definitions

Term	Definition
ACPI	Advanced Configuration and Power Interface [3].
BMC	Baseboard Management Controller. A motherboard resident management controller that provides functions for platform management.
Guest	Software in a virtual machine.
Hypervisor	Software entity that controls virtualization.
ID	Identifier.
OS	Operating System.
SoC	System on a chip, also referred as system-on-a-chip and system-on-chip.
UEFI	Unified Extensible Firmware Interface. [4]
VM	Virtual Machine.

Chapter 2. Server Platform Hardware Requirements

2.1. RISC-V Harts

A RISC-V server platform includes a RISC-V application processor and may include one or more service processors. These service processors may provide services such as security and power management to software executing on the application processors, and they may themselves implement the RISC-V ISA. The requirements in this section apply solely to harts in the application processors of the SoC.

ID#	Requirement
RVA_010	The RISC-V application processor harts in the SoC MUST support the RVA23 ISA profile [5].
RVA_020	The RISC-V application processor harts in the SoC MUST support the following extensions: • Sv48 • Svadu • Sdtrig • Sdext • Zkr • Ssccfg • Sscsrind • Ssstrict • Smcntrpmf • Ssaia
Many of these mandated extensions are optional in the RVA23 ISA profile. This requirement is placed here as a placeholder. These mandates may be moved into a new ISA profile specification.	
RVA_021	The RISC-V application processor harts in the SoC MUST support the Ssctr extension with a CTR depth value of 32. Additional CTR depth values MAY be supported.
Mandating implementation of CTR depth of 32 provides a common CTR depth across implementations for purposes of VM migration. *Ssctr is under construction.*	
RVA_022	The RISC-V application processor harts MUST raise an illegal-instruction exception when attempting to execute unimplemented opcodes or access unimplemented CSRs.

ID#	Requirement
RVA_030	The ISA extensions and associated CSR field widths implemented by any of the RISC-V application processor harts in the SoC MUST be identical.
The RVA23 profile supports a set of optional extensions. The set of optional extensions implemented by the harts must be identical. Where the extension supports optionality in the form of field widths (e.g., ASIDLEN, VLEN, allowed vstart values, physical address width, debug triggers, cache-block size, etc.), the implementation of these must also be identical. Having an identical ISA on all harts allows system software to migrate tasks among the harts without constraints.	
RVA_040	The RISC-V application processor harts in the SoC MAY support different power and performance characteristics but MUST be otherwise indistinguishable from each other from a software execution viewpoint.
All harts in the SoC being indistinguishable from a software execution viewpoint allows system software to migrate tasks among the harts without constraints.	
RVA_050	The RISC-V application processor hart MUST support: • Single stepping using the step bit in `dcsr` • Debug scratch register 0 (`dscratch0`)
RVA_060	The RISC-V application processor hart MUST support: • At least 4 instruction address match triggers. • At least 4 load/store address match triggers. • At least one icount trigger to support single stepping. • At least one interrupt trigger. • At least one exception trigger. • Trigger filtering using `hcontext`. • Trigger filtering using all VMID encodings supported by the hart. • Trigger filtering using `scontext`. • Trigger filtering using all ASID encodings supported by the hart.
RVA_070	The RISC-V application processor MUST support at least 6 hardware performance counters defined by the Zihpm extension in addition to the three counters defined by Zicntr extension.

2.2. RISC-V SoC

ID#	Requirement
HSOC_010	The RISC-V SoC MUST comply with the Server SoC specification [6].
The Server SoC specification is still under construction. This specification should be updated once the specification versioning info is finalized.	

ID#	Requirement
HSOC_020	All peripherals that are intended for assignment to a VM or a user space device driver MUST be PCIe devices or be compliant to rules for SoC-integrated PCIe devices ([6], Section 2.4).

2.3. Peripherals

ID#	Requirement
HPER_010	For remote-access and system engineering purposes, a fully 16550-compatible [7] UART MUST be implemented.
This is a stronger requirement than the Server SoC MNG_030 requirement [6]. This specification does not provide guidance around how the UART is physically exposed, i.e. via RS232 signalling, USB, a BMC or other mechanism.	
HPER_020	The implemented UART MUST support: • Interrupt-driven operation using a wired interrupt. • Flow control. • 115200 baud operation.
HPER_030	If a USB controller is implemented, it MUST comply with XHCI 1.2 or later [8].
HPER_040	Implemented XHCI controllers MUST support: • 64-bit addressing (AC64 = '1'). • A 4K PAGESIZE.
HPER_050	If a SATA controller is implemented, it MUST comply with AHCI 1.3.1 or later [9].
HPER_060	Implemented AHCI controllers MUST support: • 64-bit addressing (S64A = '1').
HPER_070	A battery-backed RTC or analogous timekeeping mechanism MUST be implemented.
HPER_080	A Trusted Platform Module (TPM) MUST be implemented and adhere to the TPM 2.0 Library specification [10].

Chapter 3. Server Platform Firmware Requirements

ID#	Requirement
FIRM_010	The RISC-V SoC MUST comply with the BRS-I recipe described in the Boot and Runtime Service specification [11].
The Boot and Runtime Services specification is still under construction. This specification should be updated once the specification versioning info is finalized.	
FIRM_020	MUST include the ability to boot from disk (block) and network (PXE, HTTP) devices.

Chapter 4. Server Platform Security Requirements

Security requirements straddle hardware and firmware.

TBD: it is expected the high-level RoT / boot flow requirements will come from the platform security spec.

ID#	Requirement
SEC_010	MUST implement UEFI Secure Boot and Driver Signing ([4] Section 32)
SEC_020	MUST back the UEFI Authenticated Variables implementation with a mechanism that cannot be accessed or tampered by an unauthorized software or hardware agent.
SEC_030	MUST implement in-band firmare updates as per [11].
SEC_040	Firmware update payloads MUST be digitally signed.
SEC_050	Firmware update signatures MUST be validated before being applied.
SEC_060	It MUST not be possible to bypass secure boot, authentication or digital signature failures.

Bibliography

[1] “Key words for use in RFCs to Indicate Requirement Levels.” [Online]. Available: datatracker.ietf.org/doc/html/rfc2119.

[2] “PCI Express® Base Specification Revision 6.0.” [Online]. Available: pcisig.com/pci-express-6.0-specification.

[3] “Advanced Configuration and Power Interface (ACPI) Specification.” [Online]. Available: uefi.org/specifications.

[4] “Unified Extensible Firmware Interface.” [Online]. Available: uefi.org/specifications.

[5] “RVA23 Profiles.” [Online]. Available: github.com/riscv/riscv-profiles/blob/main/rva23-profile.adoc.

[6] “RISC-V Server SoC Specification.” [Online]. Available: github.com/riscv-non-isa/server-soc.

[7] “National Semiconductor PC16550D UART Datasheet.” [Online]. Available: www.scs.stanford.edu/10wi-cs140/pintos/specs/pc16550d.pdf.

[8] “eXtensible Host Controller Interface for Universal Serial Bus 1.2.” [Online]. Available: www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/extensible-host-controler-interface-usb-xhci.pdf.

[9] “Advanced Host Controller Interface (AHCI).” [Online]. Available: www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/serial-ata-ahci-spec-rev1-3-1.pdf.

[10] “TPM 2.0 Library.” [Online]. Available: trustedcomputinggroup.org/resource/tpm-library-specification/.

[11] “RISC-V Boot and Runtime Services Specification.” [Online]. Available: github.com/riscv-non-isa/riscv-brs.