RISC-V Server Platform Test 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 Test specification defines a set of tests to verify if the requirements specified in RISC-V Server Platform specification are implemented. The tests specified in this specification are not intended to exhaustively verify the implementation. In most cases the tests only check for existence of the feature. Future versions of this specification may include more exhaustive tests.

The Server Platform specification builds on top of the Server SoC, Boot and Runtime Services and Platform Security specifications, which in turn have their own test specifications and/or compliance requirements. This test specification does not duplicate requirements in these dependent test specification, but provides additional tests on top the the tests already defined in these other documents.

The tests in this specification are documented use the following format:

TEST_ID#	Test algorithm
AB_CAT_NNN_MMM	The `CAT_NNN` identifies a requirement in the RISC-V Server SoC specification. Each requirement is associated with one or more tests identified by `MMM`. The test IDs are prefixed with two character prefix - `AB`. If character in position `A` is `M` then the test is for a requirement that MUST be supported and this test MUST pass. If character in position `A` is `O` then the test is for a requirement that SHOULD or MAY be supported; such tests may be skipped if the requirement is not implemented. The tests record if optional features were present in the test output log. The character in position `B` indicates the nature of the test. If this character is `F` then the test exercises some or all of the functionality associated with the feature. If the character is `E` then the test determines for evidence that the feature is implemented (e.g., check ACPI tables) but does not functionally exercise the feature.

This specification groups the tests 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.

Table 1. Terms and definitions

Term	Definition

Chapter 2. Server Platform Test Specification

2.1. Server Platform Hardware Requirements

2.1.1. RISC-V Harts

ID#	Algorithm
ME_RVA_010_010	For each application processor hart: 1. Determine the ISA node in ACPI RHCT table for that hart. 2. Parse the ISA string in the ISA node and verify that all mandatory extensions are supported. 3. Verify that the ISA string matches that of hart 0. 4. Report the ISA string of hart 0 into the test output log.
ME_RVA_020_010	See T_RVA_010_010.
ME_RVA_030_010	1. The T_RVA_010_010 verifies that all ISA strings are identical. 2. For each ISA extension reported in the ISA string, if there are CSRs associated with that extension, then probe the CSR to determine the width of the CSR fields and the legal encodings on each application processor hart. The CSR field widths and legal encodings supported by each hart must match that of hart 0.
ME_RVA_040_010	See ME_RVA_030_010.
ME_RVA_050_010	No test.
MF_RVA_060_010	Install 4 instruction address match triggers using the debug triggers SBI and verify that each trigger fires.
MF_RVA_060_020	Install 4 load address match triggers using the debug triggers SBI and verify that each trigger fires.
MF_RVA_060_030	Install 4 store address match triggers using the debug triggers SBI and verify that each trigger fires.
MF_RVA_060_040	Install an icount trigger using the debug triggers SBI and verify single-step.
MF_RVA_060_050	1. Install an interrupt trigger to match supervisor timer interrupt using the debug triggers SBI. 2. Program a timer deadline in stimecmp 3. Verify that the trigger fires on reaching the programmed deadline.

ID#	Algorithm
MF_RVA_060_060	1. Install an exception trigger to match ECALL to S-mode exception using the debug triggers SBI. 2. Transition to U-mode and invoke an ECALL. 3. Verify that the trigger fires.
MF_RVA_060_070	1. Verify hcontext exists. 2. Repeat MF_RVA_060_010 and MF_RVA_060_050 with a matching and non-matching hcontext value.
ME_RVA_060_080	1. Install and read-back triggers with VMID values between 0 and VMIDLEN.
MF_RVA_060_090	1. Verify scontext exists. 2. Repeat MF_RVA_060_010 and MF_RVA_060_050 with a matching and non-matching scontext value.
ME_RVA_060_100	1. Install and read-back triggers with ASID values between 0 and ASIDLEN.
ME_RVA_070_010	1. Request delegation of all HPM counters using the SBI. 2. Verify at least 6 programmable HPM counter are implemented. 3. Verify that the scountovf CSR is implemented 4. Verify cycles and instret are writeable. 5. Verify ability to toggle counter enable for each implemented HPM, cycles, and instret counters.

2.1.2. RISC-V SoC

ID#	Algorithm
ME_HSOC_010_010	The Server SoC tests must pass [1].
ME_HSOC_020_010	*FIXME*.

2.1.3. Peripherals

ID#	Algorithm
ME_HPER_010_010	*FIXME.*
MF_HPER_020_010	*FIXME.*
MF_HPER_030_010	*FIXME XHCI test validating register values.*
MF_HPER_040_010	*FIXME XHCI test validating register values.*
MF_HPER_050_010	*FIXME AHCI test validating register values.*
MF_HPER_060_010	*FIXME AHCI test validating register values.*
MF_HPER_070_010	*FIXME UEFI RT based test.*
MF_HPER_080_010	*FIXME.*

2.2. Server Platform Firmware Requirements

ID#	Algorithm
ME_FIRM_010_010	The BRS-I tests must pass [2].
ME_FIRM_020_010	*FIXME presence tests for block / FS protocols*
ME_FIRM_020_020	*FIXME presence tests for network protocols*

2.3. Server Platform Security Requirements

ID#	Algorithm
ME_SEC_010_010	*FIXME*
ME_SEC_020_010	*FIXME*
ME_SEC_030_010	*FIXME*
ME_SEC_040_010	*FIXME*
ME_SEC_050_010	*FIXME*
ME_SEC_060_010	*FIXME*

Bibliography

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

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