Leverage BRS standard to improve RISC-V SW compatibility

Haibo Xu <haibo1.xu@intel.com>
Andrei Warkentin <andrei.warkentin@intel.com>

intel

Legal Notices and Disclaimers

Statements in this document that refer to future plans or expectations are forward-looking statements. These statements are based on current expectations and involve many risks and uncertainties that could cause actual results to differ materially from those expressed or implied in such statements. For more information on the factors that could cause actual results to differ materially, see our most recent earnings release and SEC filings at www.intc.com.

All product plans and roadmaps are subject to change without notice. Any forecasts of goods and services needed for Intel's operations are provided for discussion purposes only. Intel will have no liability to make any purchase in connection with forecasts published in this document. Code names are often used by Intel to identify products, technologies, or services that are in development and usage may change over time. No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

 This document contains information on products and/or processes in development.

Background

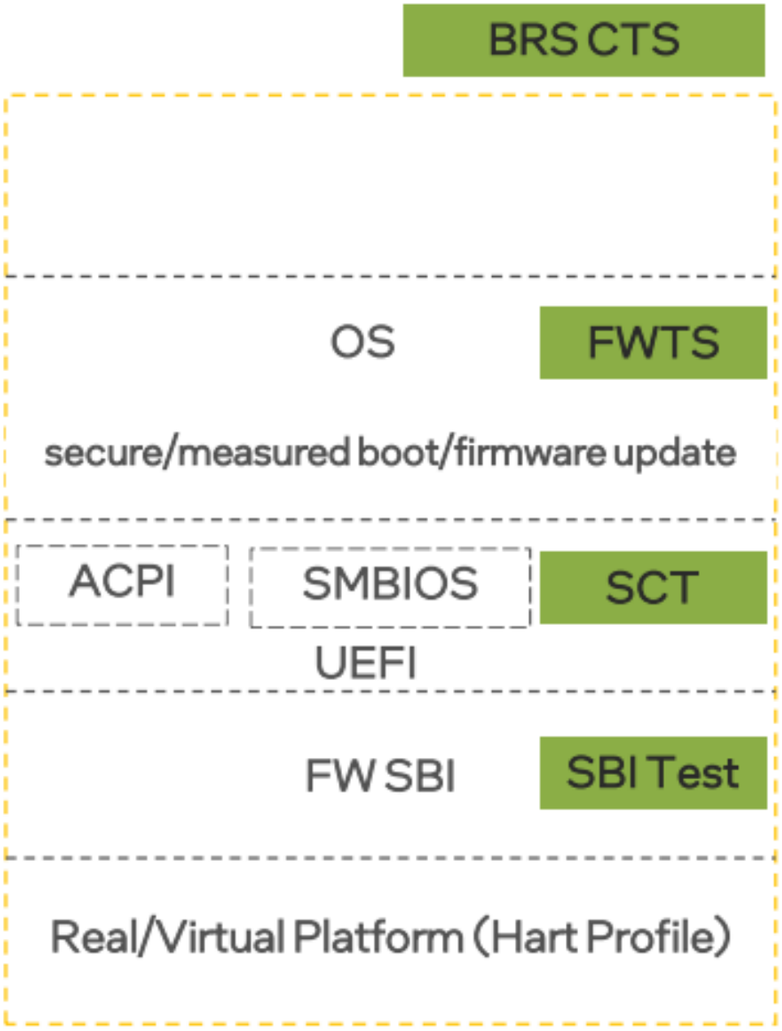
- With the growing of RISC-V based products, there is a need for a standardized way to ensure interoperability between different RISC-V platforms and OSs
- Rule of thumb is standardization and conformance test
- RISC-V BRS specification defined some requirements (based on SBI/UEFI/ACPI/SMBIOS/DT etc.) for Boot and Runtime services that system software can rely on

RISC-V BRS Spec

- Defined by RISC-V BRS TG and specify the Boot and Runtime Services requirements for firmware on a RISC-V platform.
- Include two recipes: BRS-I(Interoperable) and BRS-B(Bespoke)
- BRS-B is intended for software customization for specific devices
- BRS-I aims to ensure interoperability between different RISC-V platforms and OS/Hypervisor.
  - Hart requirements (>= RVA20S64)
  - SBI requirements (RISC-V Supervisor Binary Interface Spec v2.0 or later)
    - HSM is mandated, TIME/IPI/RFNC/PMU are conditionally required
  - UEFI requirements (>= UEFI 2.10/64 bits/Address Translation)
    - Security, I/O, Runtime Services, Firmware Update
  - ACPI requirements (>= ACPI 6.6/64 bits/hardware reduced mode)
    - ACPI Tables/Methods/Objects/IDs/DSD
  - SMBIOS requirements (>= SMBIOS 3.7.0, Structure Type 02/03 ...)
- https://github.com/riscv-non-isa/riscv-brs

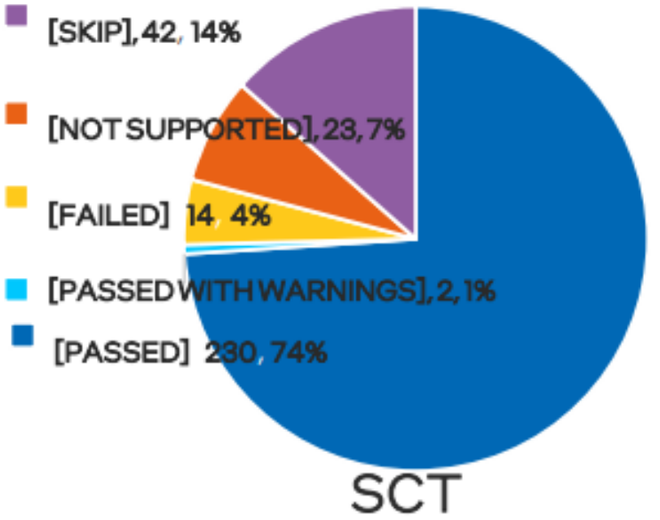
RISC-V BRS Test Suite

- RISC-V BRS TG defined the RISC-V BRS Test Specification to verify if the requirements specified in RISC-V BRS specification are implemented.
- RISC-V BRS Test Suite was supposed to be developed against BRS Test Specification and was mainly based on UEFI-SCT and FWTS projects.
- https://github.com/intel/rv-brs-test-suite

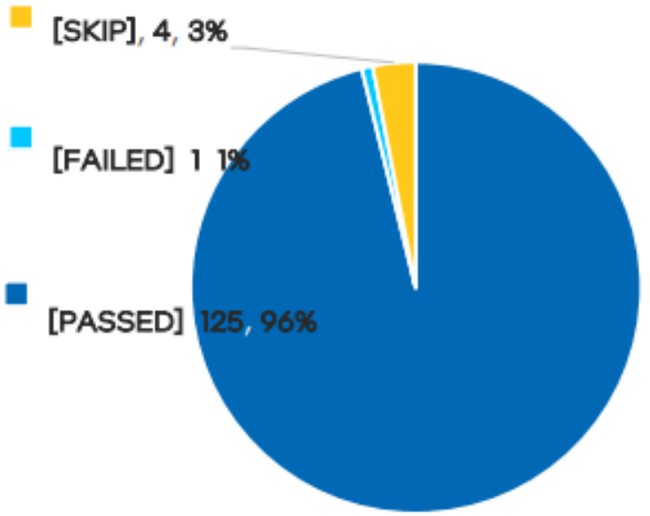


Current Status

- RISC-V BRS Specification
  - Freeze – under ARC review
  - Ratification Target Date – Q42024
  - BRS-I was required in RISC-V Server Platform Spec
  - https://jira.riscv.org/browse/RVS-1193
- RISC-V BRS Test Specification
  - Initial Spec was pushed to upstream
  - More test requirements will be added
  - https://github.com/riscv-non-isa/riscv-brs/pull/178
- The RISC-V BRS Test Suite
  - Results on a Qemu virt platform was ready
  - Gap analysis was done
  - More test cases are under development
  - https://github.com/intel/rv-brs-test-suite/issues



SCT



FWTS

Key Gaps & Call to Action

- Enhance the BRS Test Specification
- Join the discussion and review for BRS specification
  - https://github.com/riscv-non-isa/riscv-brs
  - https://lists.riscv.org/g/tech-brs
- Collaborate on the rv-brs-test-suite
  - https://github.com/intel/rv-brs-test-suite/issues
  - Summary of BRS Specification Test Coverage
  - Linux/Qemu
  - UEFI-SCT
  - FWTS
  - KVM-Unit-Tests for SBI testing


intel