Comprehensive Verification of the RISC-V Memory Management Unit: Challenges and Solutions

Huda Sajjad, Hammad Bashir, Yazan Hussnain, Fatima Saleem

10xEngineers

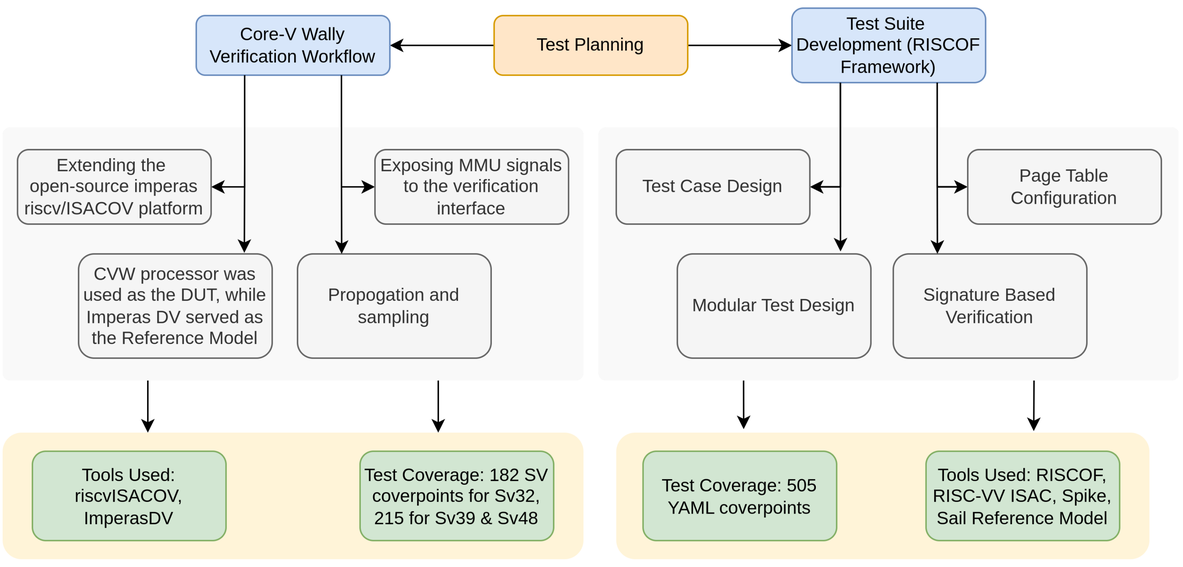
Introduction

The Memory Management Unit (**MMU**) enables virtual address translation, memory protection, and multitasking. Ensuring compliance with the RISC-V Privileged ISA is crucial for interoperability. However, its configurability — supporting multiple paging schemes and superpage translations — poses significant verification challenges, especially in open-source cores where edge cases and ambiguities can cause critical flaws.

Test Planning

A robust and unified design verification (DV) plan forms the foundation for validating all critical aspects of the MMU.

- **Transaction Access Validation:**
  Verified read, write, and execute transactions by evaluating PTE permission bits (**R**/**W**/**X**) in leaf entries across all page table levels in both supervisor and user modes.
- **PTE Global Mapping:**
  Verified the functionality of the Global bit (pte.**G**) in PTEs, across all levels, for different address spaces identified by satp.**ASID**.
- **Address Translation in M-Mode:**
  Verified the functionality of mstatus.**MPRV** allowing address translation in M-Mode for data accesses while bypassing translation for eXecute accesses.
- **Supervisor-User Memory Access:**
  Verified the functionality of mstatus.**SUM**, enabling S-Mode to access U-Mode pages mapped with the pte.**U** bit set.
- **Read Access on eXecutable Pages:**
  Verified the functionality of mstatus.**MXR**, enabling read access on execute-only pages (pte.**R**=0, pte.**X**=1).

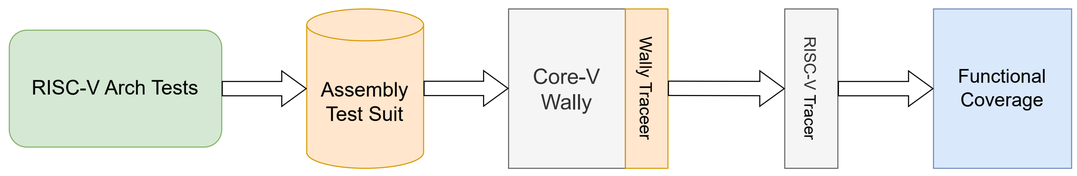


Checkout Core-V Wally Github Repository

For more information, visits 10xEngineers.ai

Results/Findings

This work was implemented and validated on Core-V Wally, a 5-stage pipelined processor supporting configurations from RV32E to a full RV64GC application processor. The proposed test suite successfully uncovered a critical bug in the MMU through the reserved_pte_s_mode test. The bug caused Core-V Wally to fail in triggering a page fault when accessing memory regions mapped by Page Table Entries (PTEs) with reserved RWX encoding (pte.**W**=1 and pte.**R**=0), violating the RISC-V Privileged ISA specification.



Conclusion

- Enhanced verification framework for RISC-V MMUs
- Discovered a major flaw in Core-V Wally's MMU implementation
- Improved compliance testing for open-source processor designs

Our methodology strengthens MMU validation for open-source RISC-V cores, ensuring better reliability and compliance.

References

1. **RISC-V Foundation**. RISC-V Privileged Architectures Manual, Version 1.12. Accessed: 2025-01-27. 2021. url: https://github.com/riscv/riscv-isa-manual.
2. **RISC-V Software Source**. RISCOF: RISC-V Architectural Compliance Framework. Accessed: 2025-01-27. 2025. url: https://github.com/riscv-software-src/riscof.
3. **OpenHW Group**. CVW: Core Verification Workflows. Accessed: 2025-01-27. 2025. url: https : / / github . com /openhwgroup/cvw.
4. **RISC-V Verification Group**. RISC-V ISA Coverage Analysis Tool (riscvISACOV). Accessed: 2025-01-27. 2025. url: https://github.com/riscv-verification/riscvISACOV.
5. **RISC-V Foundation**. RISC-V Architecture Test Framework. Accessed: 2025-01-27. 2025. url: https://github.com/riscv-non-isa/riscv-arch-test.
6. **OpenHW Group**. CVW Architectural Verification. Ac-cessed: 2025-01-27. 2025. url: https : / / github . com /openhwgroup/cvw-arch-verif.
7. **Synopsys**. ImperasDV – RISC-V Processor Design Verification. Accessed: 29-Jan-2025. 2025. url: https://www.synopsys.com/verification/imperasdv.html.
8. **OpenHW Group**. GitHub Issue #1198: CVW Repository. Accessed: 2025-01-27. 2025. url: https://github.com/openhwgroup/cvw/issues/1198.

FOUNDATION OPENHW PROVEN PROCESSOR IP

10x ENGINEERS

Have Question? Chat now!