BREKER™

Ensuring Ultra-reliable RISC-V Designs for Space

RISC-V in Space Workshop

April 2025

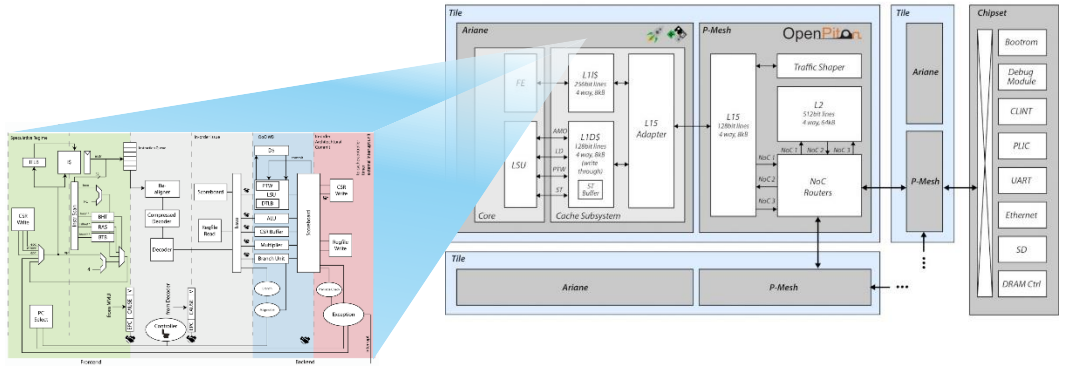
Adnan Hamid, CTO, Breker

RISC-V in Space Verification Requirements

- Verifying any RISC-V core or SoC is complex.
  Ensuring coverage for RISC-V space applications greatly compounds this issue
- DO-254 for space applications leverages the V model to drive reliability by linking requirements, test plans, and coverage
- Breker delivers advanced RISC-V test suites to many companies in this space.
  We automate V model execution to link requirements to coverage testing

Breker: RISC-V Verification Synthesized SystemVIPs

Suggested RISC-V verification "stack"



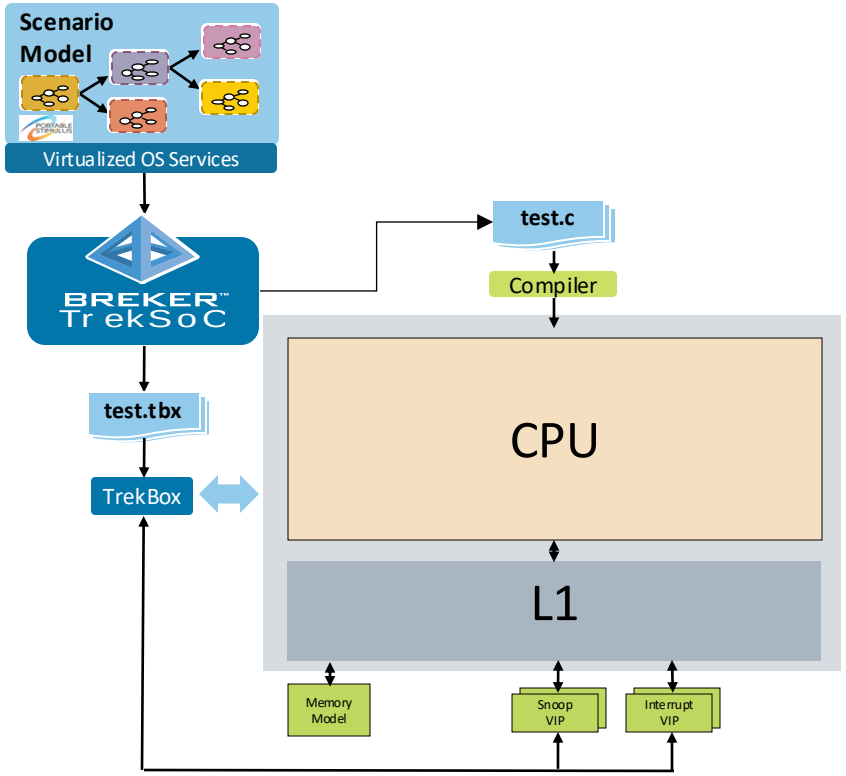
RISC-V Core Test Functionality

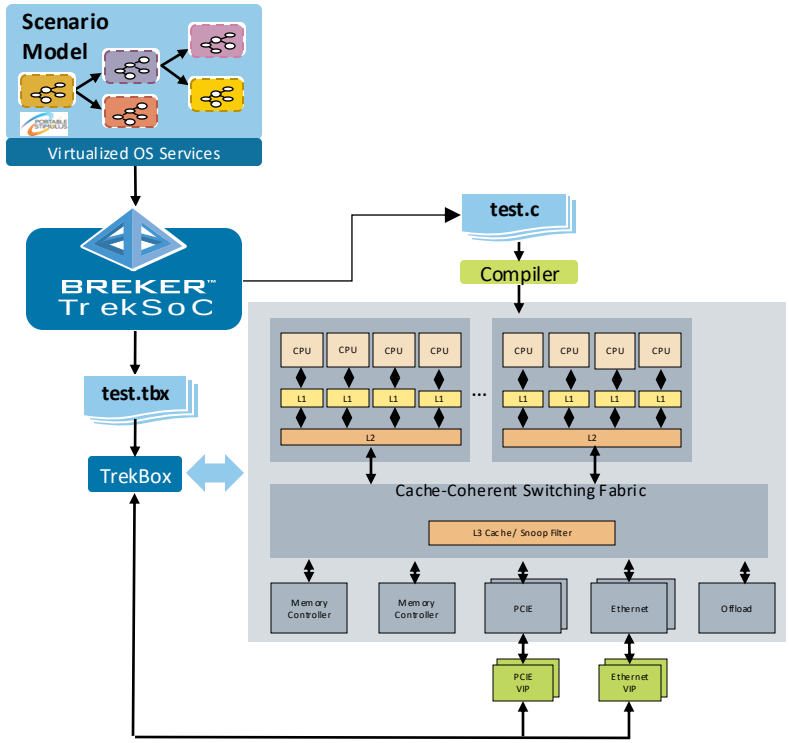
Random Instructions	Do instructions yield correct results
Register/Register Hazards	Pipeline perturbations dues to register conflicts
Load/Store Integrity	Memory conflict patterns
Conditionals and Branches	Pipeline perturbations from synchronous PC change
Exceptions	Jumping to and returning from ISR
Asynchronous Interrupts	Pipeline perturbations from asynchronous PC change
Privilege Level Switching	Context switching
Core Security	Register and Memory protection by privilege level
Core Paging/MMU	Memory virtualization and TLB operation
Sleep/Wakeup	State retention across WFI
Voltage/Freq Scaling	Operation at different clock ratios
Core Coherency	Caches, evictions and snoops

RISC-V SoC Test Functionality

Random Memory Tests	Test Cores/Fabrics/Memory controllers across DDR, OCRAM, FLASH, etc.
Random Register Tests	Read/write test to all uncore registers
System Interrupts	Randomized interrupts through CLINT
Multi-core Execution	Concurrent operations on fabric and memory
Memory Ordering	For weakly order memory protocols
Atomic Operation	Across all memory types
System Coherency	Cover all cache transitions, evictions, snoops
System Paging/IOMMU	System memory virtualization
System Security	Register and Memory protection across system
Power Management	System wide sleep/wakeup and voltage/freq scaling
Packet Generation	Generating networking packets for I/O testing
Interface Testing	Analyzing coherent interfaces including CXL & UCIe
SoC Profiling	Layering concurrent tests to check operation under stress
Firmware-First	Executing SW on block or sub-system without processor

Applying the tests to single or multi-HART SoCs





Testbench control needed for

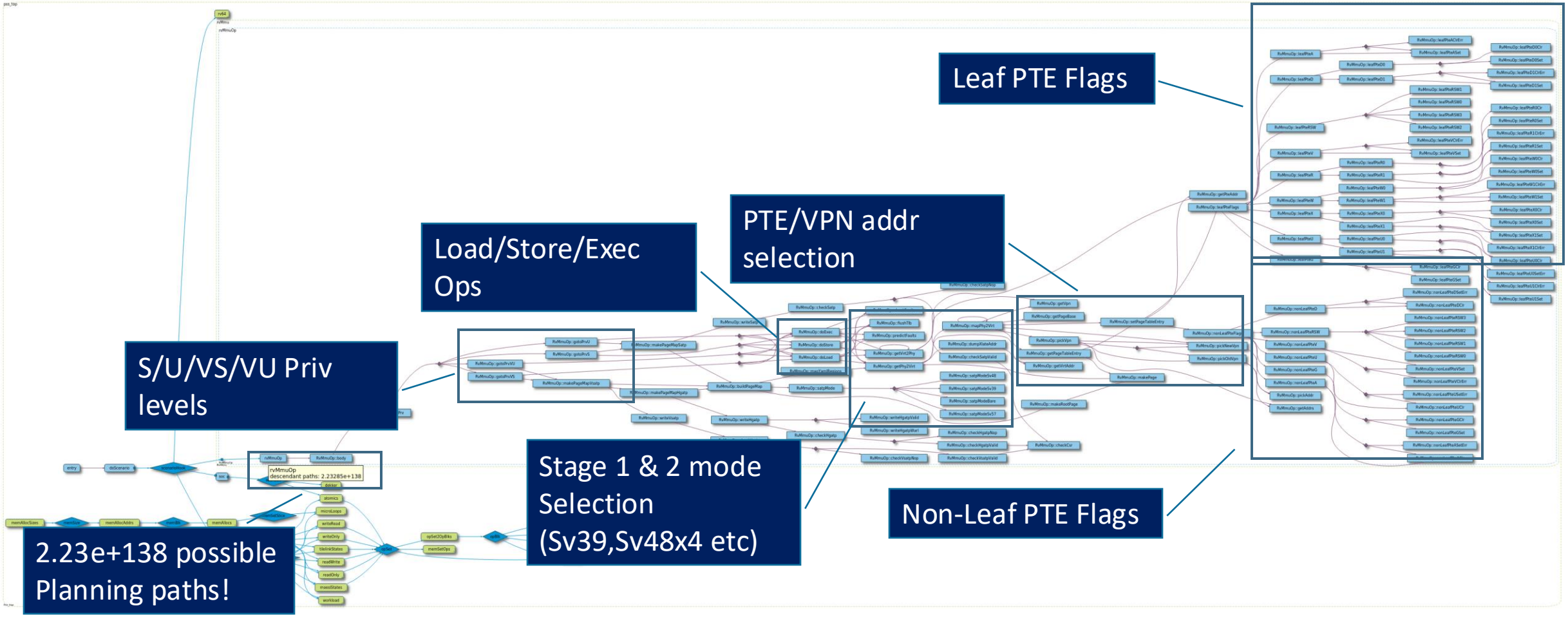
- External Interrupts,
- IOMMU,
- Debug Extension

Multi-core tests needed for

- AMO/atomics
- CMO/cache flush, invalidate
- RVWMO Memory Ordering

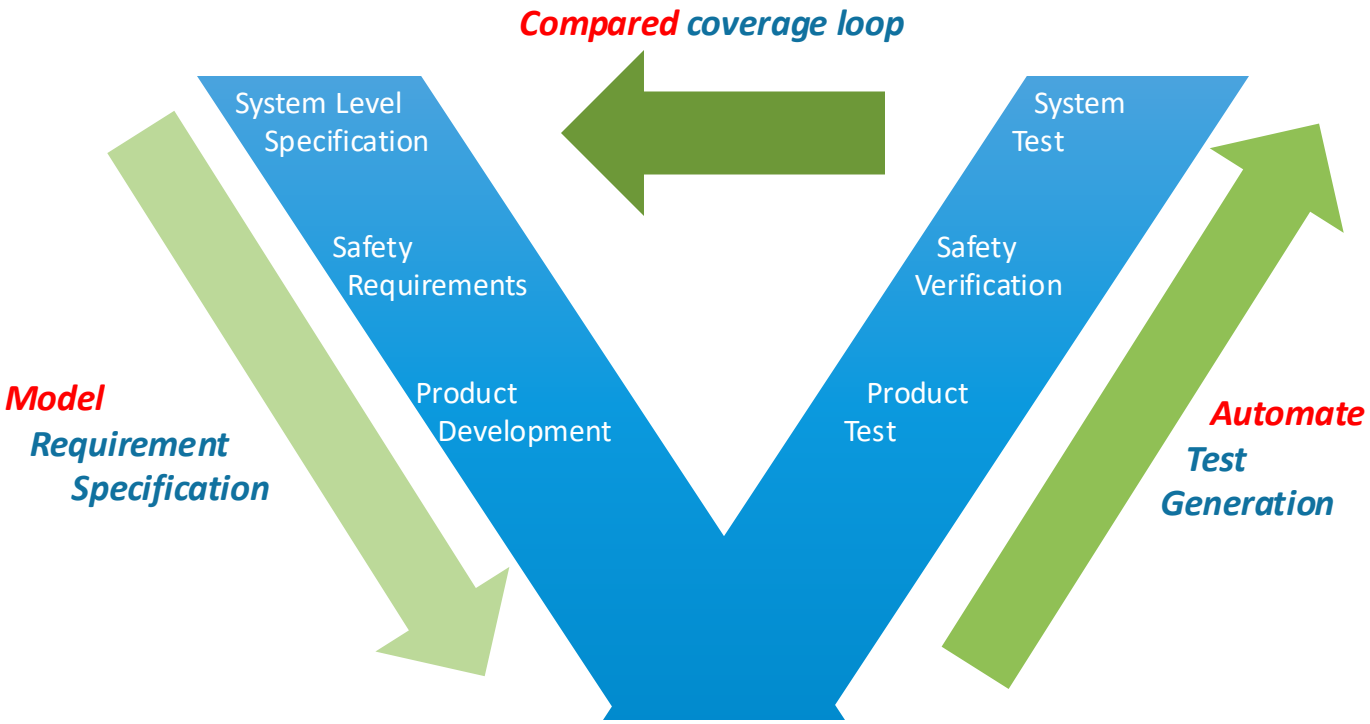
Example Scenario Graph: RISC-V MMU Hypervisor

Executable Test Plan

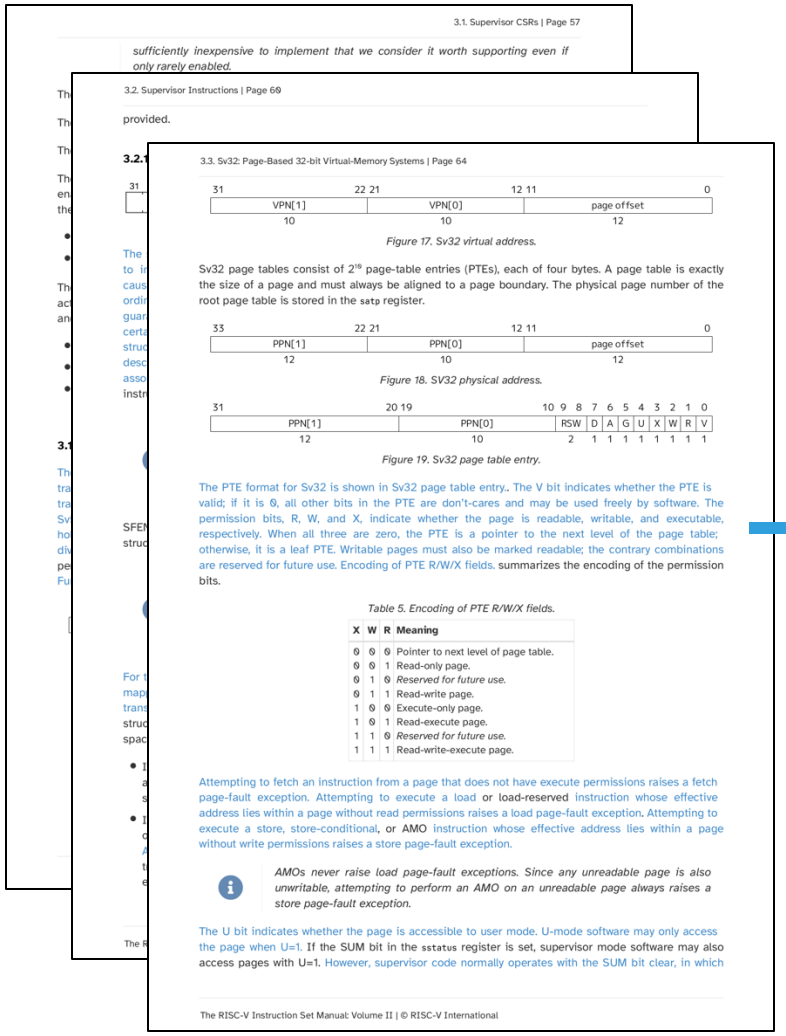


Automating the DO-254 V-Model

Ensuring the rigorous verification of device requirements



Use Appropriate ISA Specs, Derive Test Plan, and Drive Coverage



3.3. Sv32: Page-Based 32-bit Virtual-Memory Systems | Page 64

31	22 21	12 11	0
VPN[1]	VPN[0]	page offset	
10	10	12	

Figure 17. Sv32 virtual address.

Sv32 page tables consist of 2^{10} page-table entries (PTEs), each of four bytes. A page table is exactly the size of a page and must always be aligned to a page boundary. The physical page number of the root page table is stored in the `satp` register.

33	22 21	12 11	0
PPN[1]	PPN[0]	page offset	
12	10	12	

Figure 18. SV32 physical address.

31	20 19	10 9	8	7	6	5	4	3	2	1	0
PPN[1]	PPN[0]	RSW	D	A	G	U	X	W	R	V	
12	10	2	1	1	1	1	1	1	1	1	

Figure 19. Sv32 page table entry.

The PTE format for Sv32 is shown in Sv32 page table entry.. The V bit indicates whether the PTE is valid; if it is 0, all other bits in the PTE are don't-cares and may be used freely by software. The permission bits, R, W, and X, indicate whether the page is readable, writable, and executable, respectively. When all three are zero, the PTE is a pointer to the next level of the page table; otherwise, it is a leaf PTE. Writable pages must also be marked readable; the contrary combinations are reserved for future use. Encoding of PTE R/W/X fields. summarizes the encoding of the permission bits.

Table 5. Encoding of PTE R/W/X fields.

X	W	R	Meaning
0	0	0	Pointer to next level of page table.
0	0	1	Read-only page.
0	1	0	*Reserved for future use.*
0	1	1	Read-write page.
1	0	0	Execute-only page.
1	0	1	Read-execute page.
1	1	0	*Reserved for future use.*
1	1	1	Read-write-execute page.

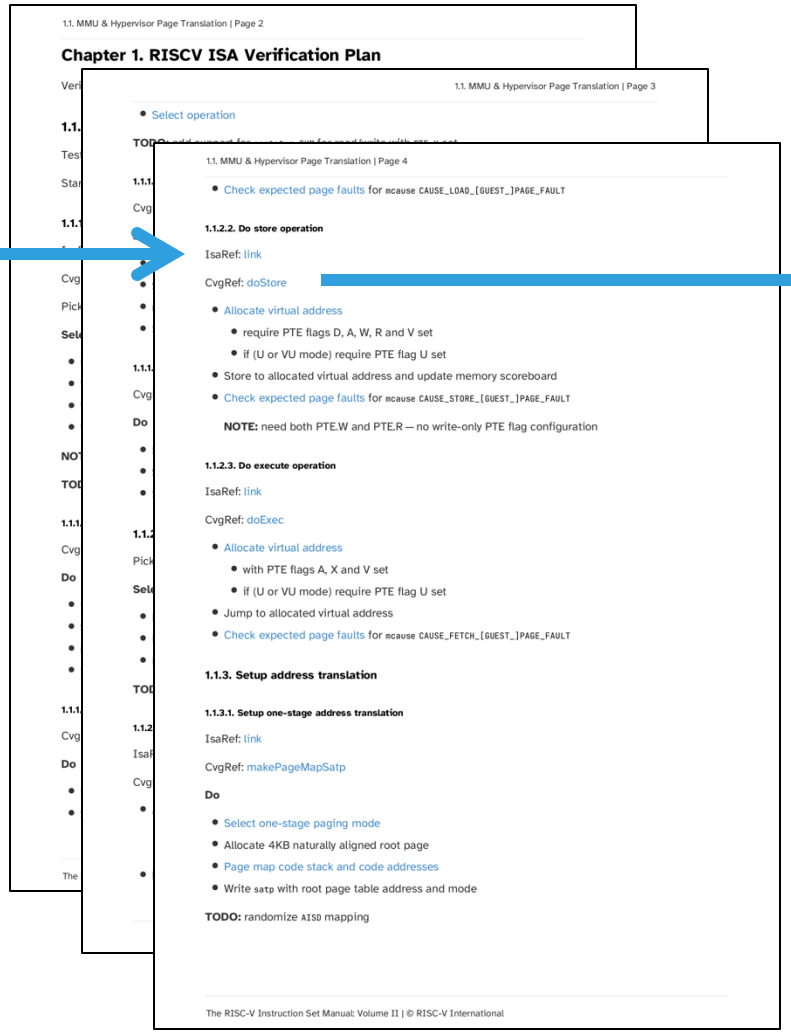
Attempting to fetch an instruction from a page that does not have execute permissions raises a fetch page-fault exception. Attempting to execute a load or load-reserved instruction whose effective address lies within a page without read permissions raises a load page-fault exception. Attempting to execute a store, store-conditional, or AMO instruction whose effective address lies within a page without write permissions raises a store page-fault exception.

AMOs never raise load page-fault exceptions. Since any unreadable page is also unwritable, attempting to perform an AMO on an unreadable page always raises a store page-fault exception.

The U bit indicates whether the page is accessible to user mode. U-mode software may only access the page when U=1. If the SUM bit in the `sstatus` register is set, supervisor mode software may also access pages with U=1. However, supervisor code normally operates with the SUM bit clear, in which

The RISC-V Instruction Set Manual: Volume II | © RISC-V International

ISA Specification



1.1. MMU & Hypervisor Page Translation | Page 2

Chapter 1. RISCV ISA Verification Plan

1.1. MMU & Hypervisor Page Translation | Page 3

- Select operation

1.1. MMU & Hypervisor Page Translation | Page 4

- Check expected page faults for `mcause CAUSE_LOAD_[GUEST_]PAGE_FAULT`

1.1.2.2. Do store operation

IsaRef: link

CvgRef: doStore

- Allocate virtual address
  - require PTE flags D, A, W, R and V set
  - if (U or VU mode) require PTE flag U set
- Store to allocated virtual address and update memory scoreboard
- Check expected page faults for `mcause CAUSE_STORE_[GUEST_]PAGE_FAULT`

NOTE: need both PTE.W and PTE.R — no write-only PTE flag configuration

1.1.2.3. Do execute operation

IsaRef: link

CvgRef: doExec

- Allocate virtual address
  - with PTE flags A, X and V set
  - if (U or VU mode) require PTE flag U set
- Jump to allocated virtual address
- Check expected page faults for `mcause CAUSE_FETCH_[GUEST_]PAGE_FAULT`

1.1.3. Setup address translation

1.1.3.1. Setup one-stage address translation

IsaRef: link

CvgRef: makePageMapSatp

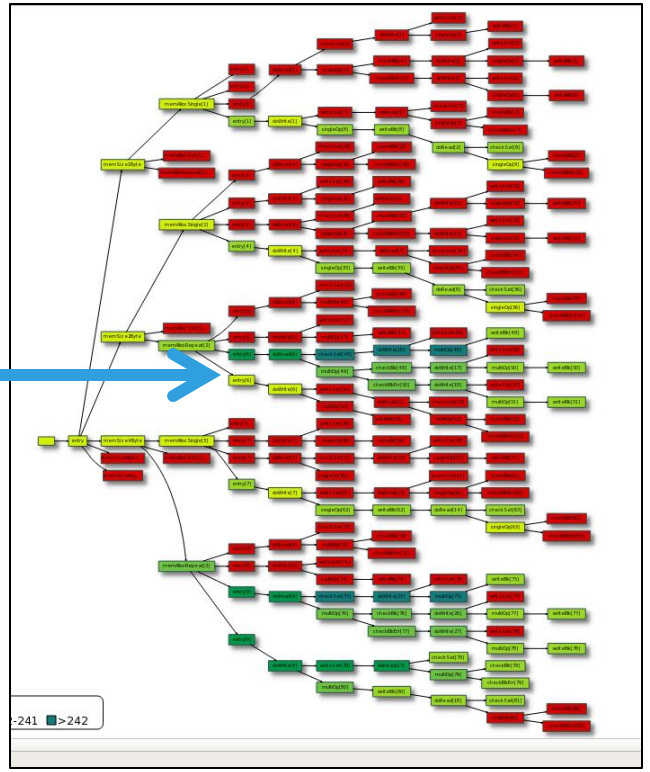
Do

- Select one-stage paging mode
- Allocate 4KB naturally aligned root page
- Page map code stack and code addresses
- Write `satp` with root page table address and mode

TODO: randomize `AISO` mapping

The RISC-V Instruction Set Manual: Volume II | © RISC-V International

RISCV Test Plan



Coverage Reports

Traceability for DO-254 Systematic Testing

Specification

3.1.15 Machine Cause Register (mcause)

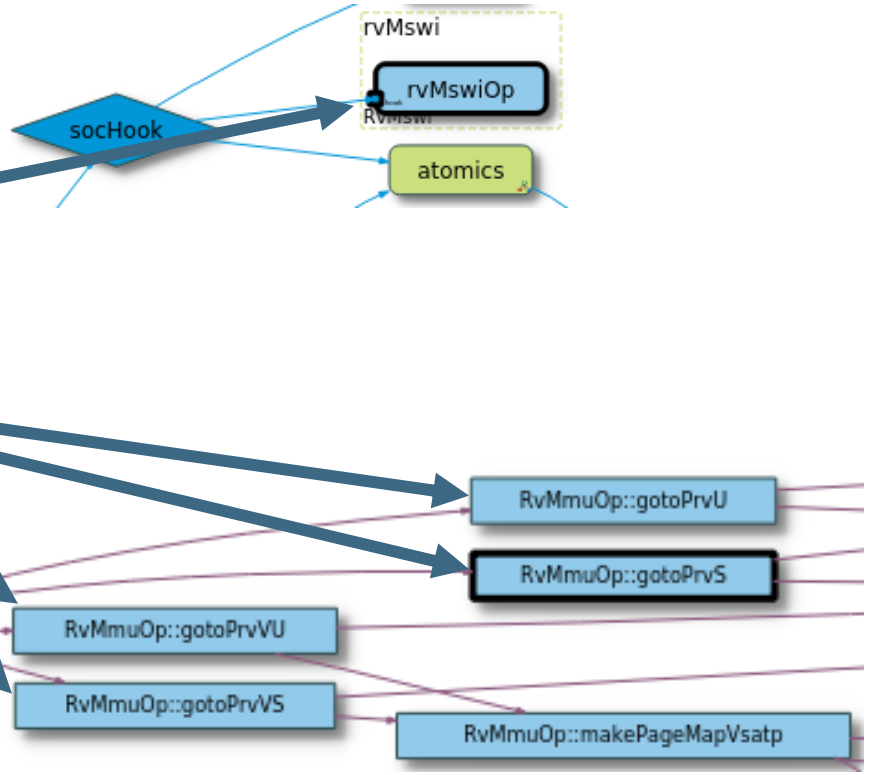
The mcause register is an MXLEN-bit read-write register formatted as shown in Figure 3.22. When a trap is taken into M-mode, mcause is written with a code indicating the event that caused the trap. Otherwise, mcause is never written by the implementation, though it may be explicitly written by software. [tp_excp]

Map Spec Features to Testplan

Testplan

7.2 [tp_excp] Exceptions
[…]
7.2.3 Machine software interrupt
[…]
7.2.8 Environment call from M-mode

Map Testplan to Graph

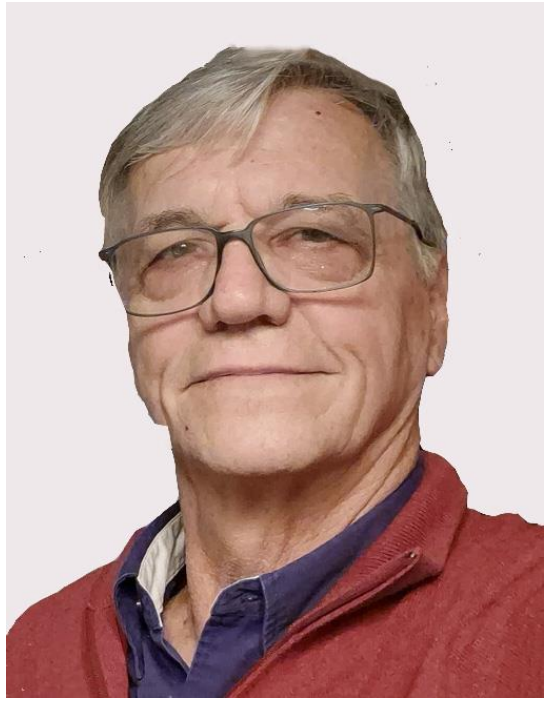


Breker RISC-V in Space Summary

- Meeting RISC-V space applications verification needs requires advanced techniques
- Automating the DO-254 V model allows the linkage of requirements with tests and coverage
- Breker delivers this level of automation using test suite synthesis and RISC-V SystemVIP

Miguel Koch
Breker European Director

He is at the conference
For more information, feel free to ask him



Thanks for Listening!
Any Questions?