BREKER™ The Leader in Portable Stimulus

RISC-V®

Breker TrekSoC
RISC-V TrekApp Test Generation

John Sotiropoulos
BrekerSystems.com

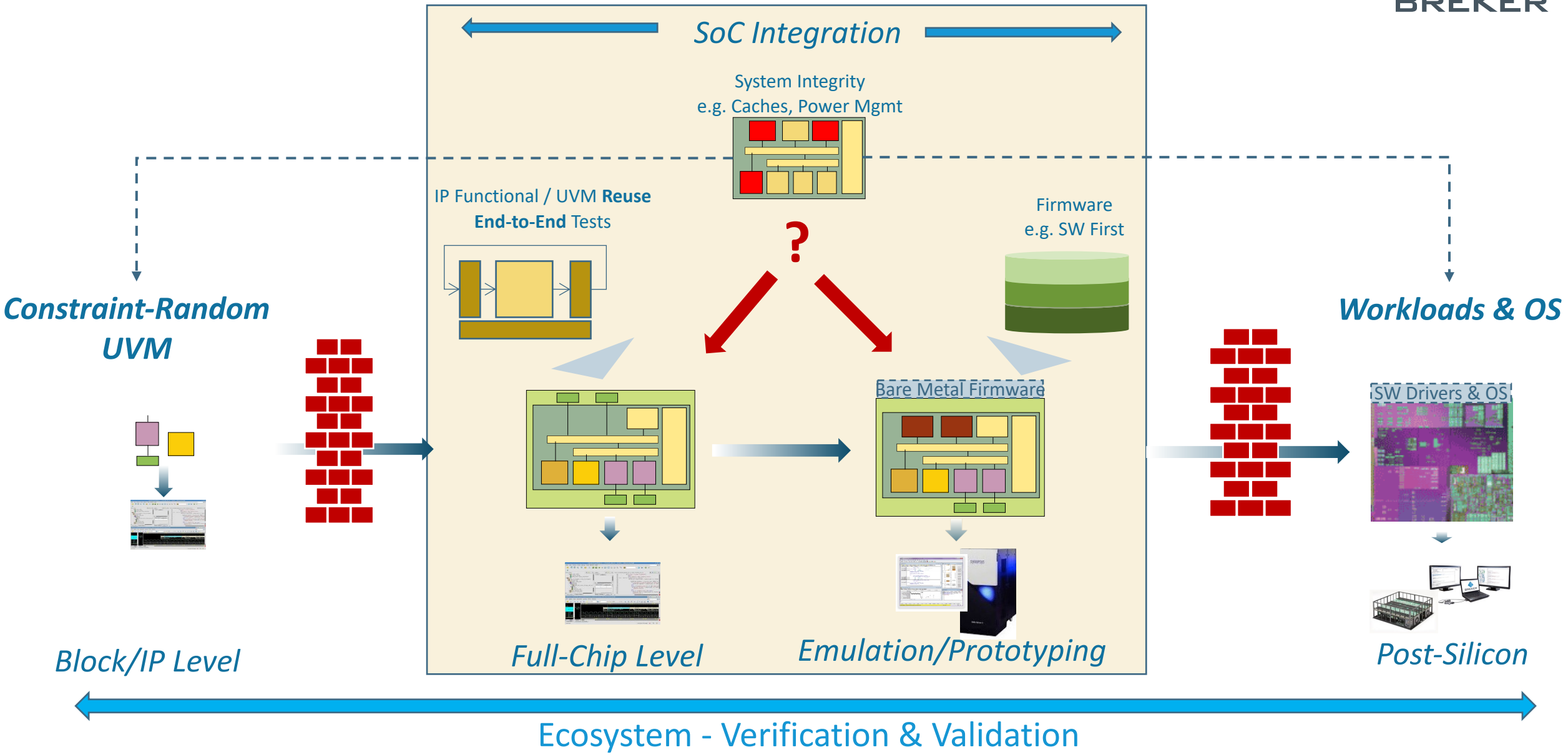
AGENDA

- Introduction to Breker Technology
- Introduction to Breker's "Core-Integrity" Tier TrekApp
- Review of Breker's "System-Integrity" Tier TrekApp
- Summary

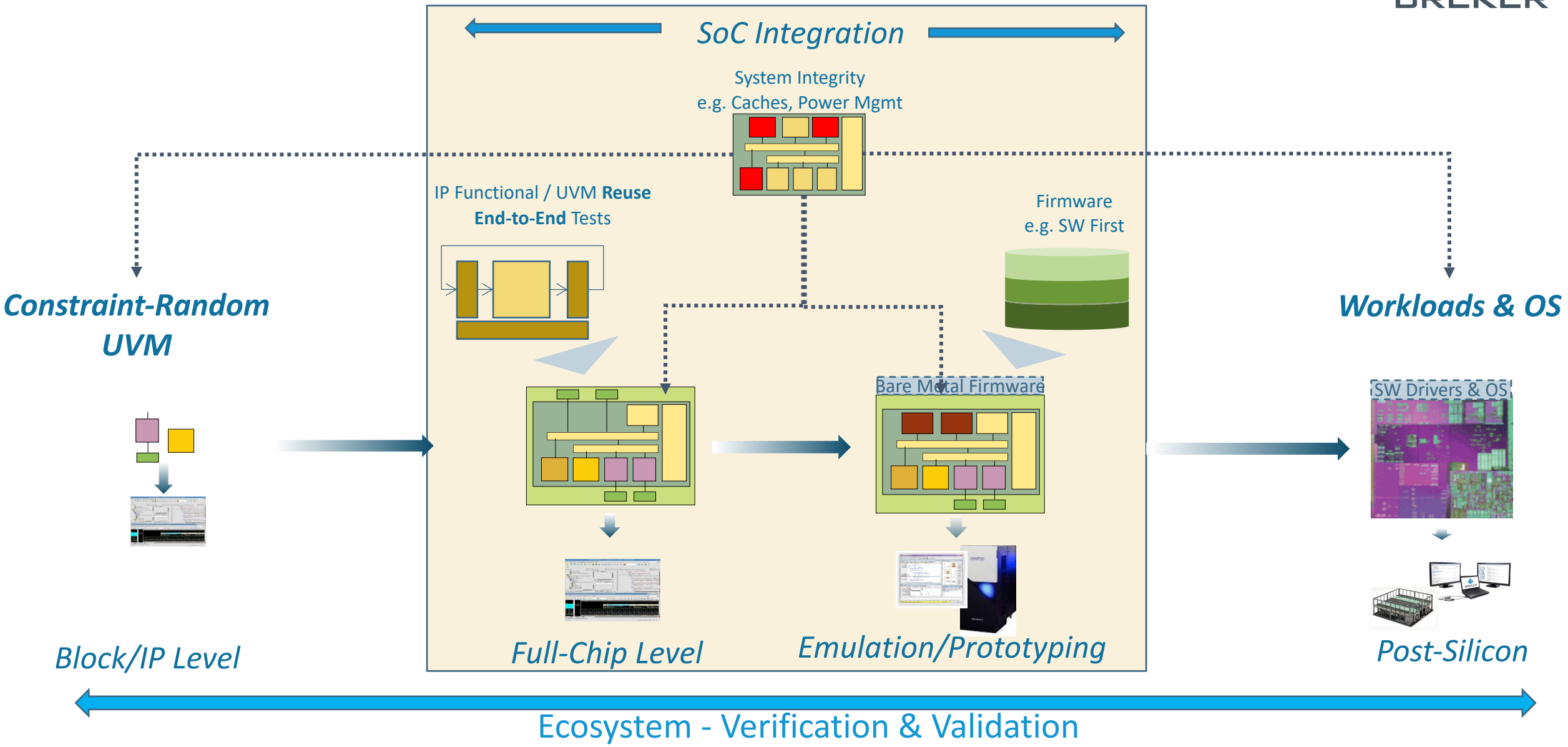
RISC-V Verification

- Processors are hard to verify
  - Consider the verification investment at Arm and Intel, for example
- Automation is the answer
  - Number of diversified test generators, etc.
- RISC-V special requirements
  - Custom instruction verification
  - Compliance assurance
  - Broad range of architectures
- Different processor levels have different needs
  - Embedded cores
  - Processor clusters
  - Application processors
- Processor integrity testing yields results for processor developers & integrators
  - Load-store, interrupt processing, performance/power profiling, etc.

The SoC Verification Gap



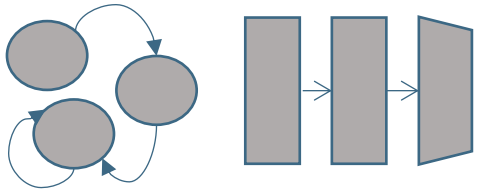
Portable Stimulus Vision



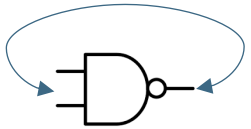
Test Suite Synthesis… Analogous to Logic Synthesis

BREKER™

Design Synthesis

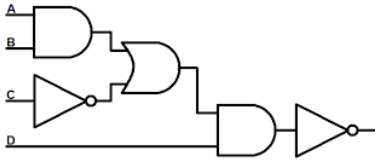


Constrain

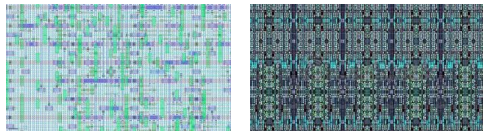


Timing/Area Constraints

Synthesize

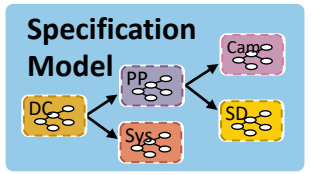


Optimize



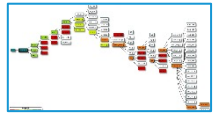
Test Suite Synthesis

Describe intent



Constrain

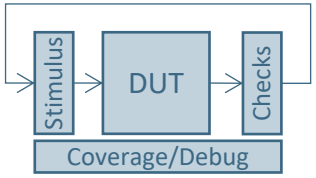
Specify goals



Coverage Constraints

Synthesize

Generate implementation



Optimize

Map to platform

Breker Core Technology

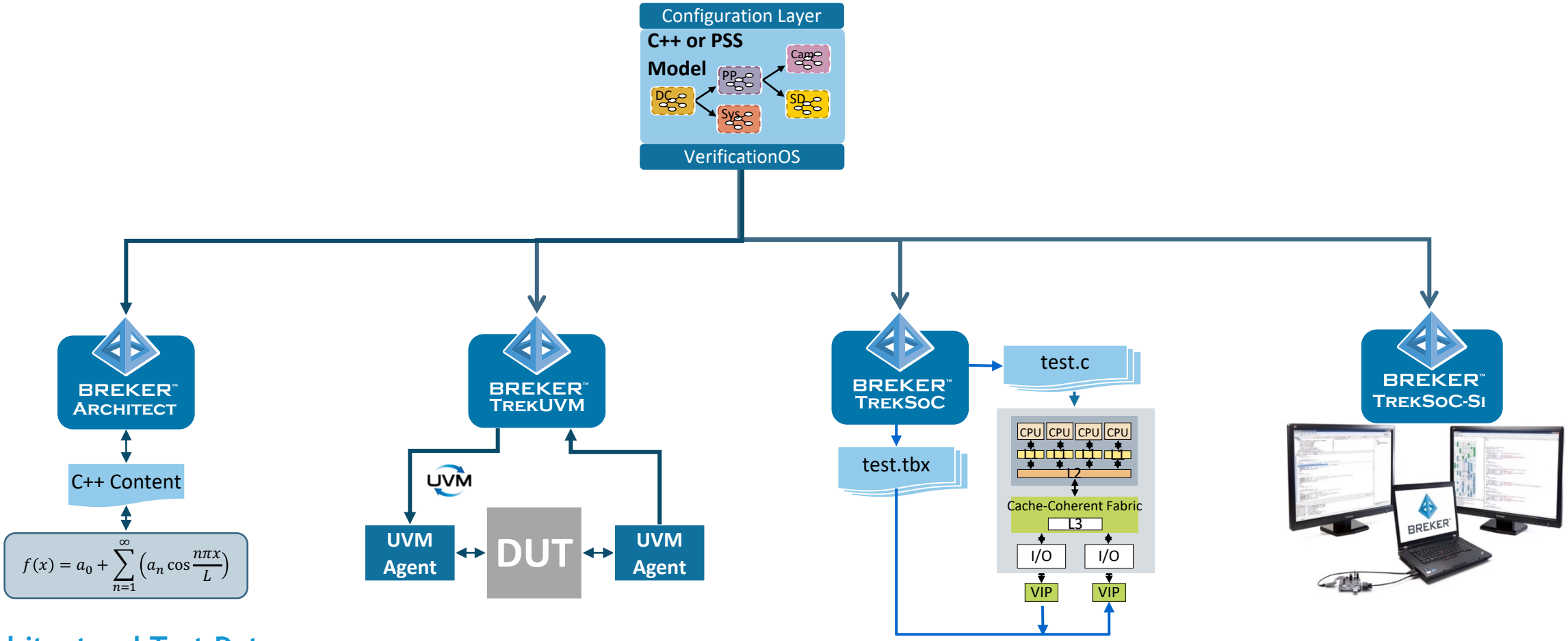
AI Planning Algorithms

3D Coverage Closure

Synthesizable VerificationOS

AI Planning Algorithms

Synthesizing Test Suites Across the Verification Flow



Architectural Test Data
C++ data and control for system exploration

UVM test content
Complex sequence, coverage scoreboard synthesis

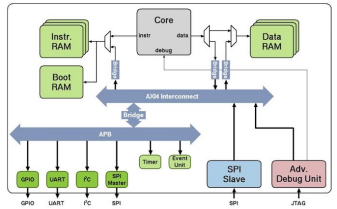
Software-driven SoC
Rigorous, high-coverage tests from a single, simple spec

Prototyping & Post-silicon
Verification test reuse with observability/controllability

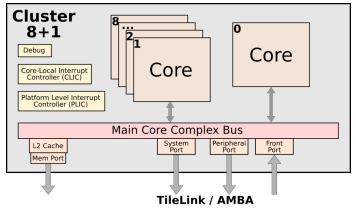
Breker Tiered Test Generation for Developers & Integrators

Processor Integrator

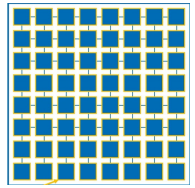
Single core, multi-thread



Multicore SoC

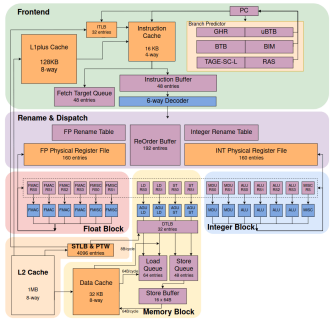


Processor Cluster

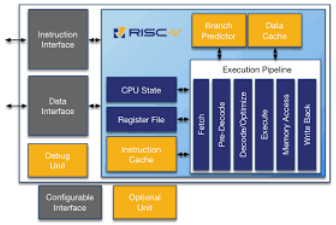


Processor Developer

Application Processor



Embedded Processor



System-Integrity

Cache-Coherency
Early Firmware
Power Domain

Core-Integrity

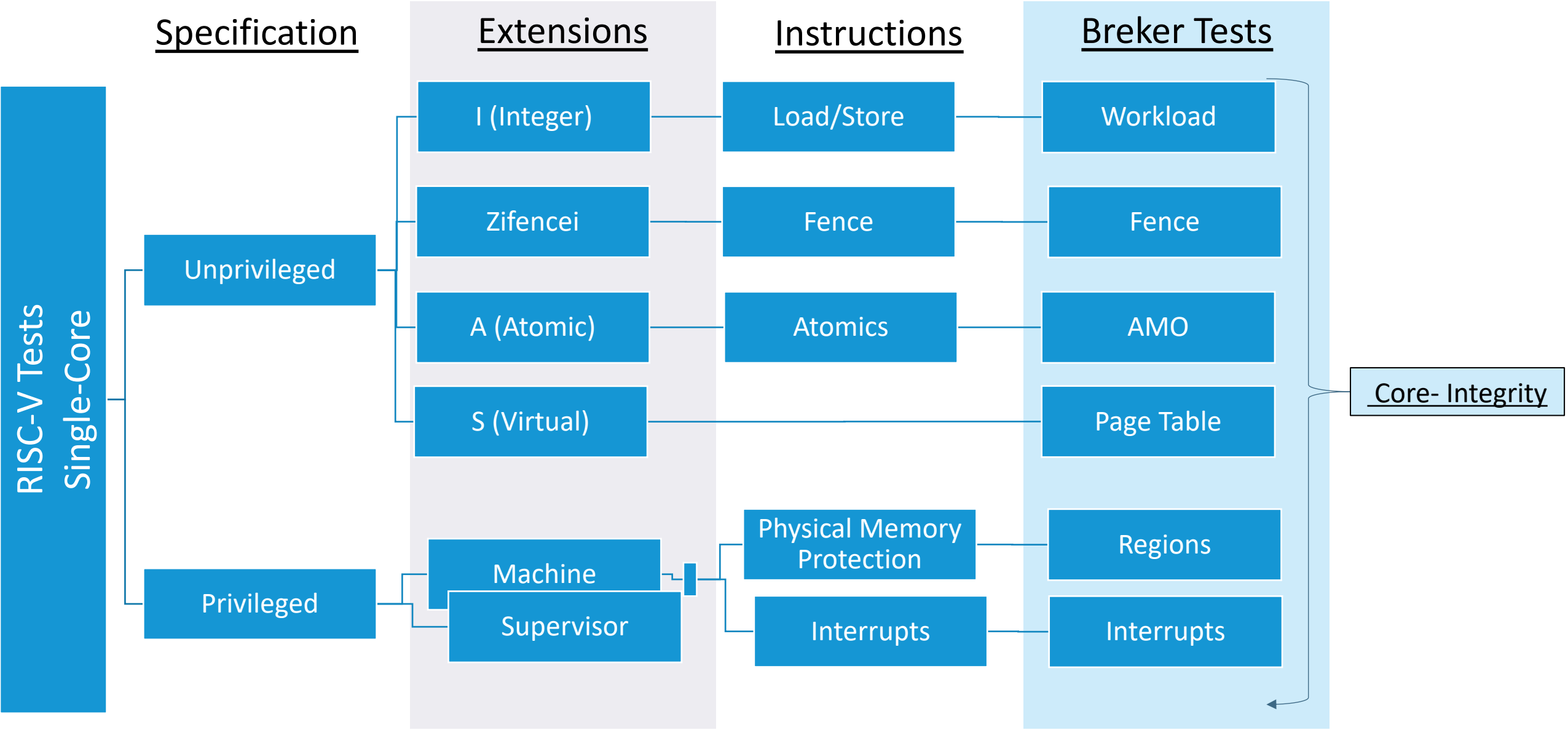
Memory load-store
Interrupt Management
Security PMP

Tiered test generation scenarios for developers and integrators

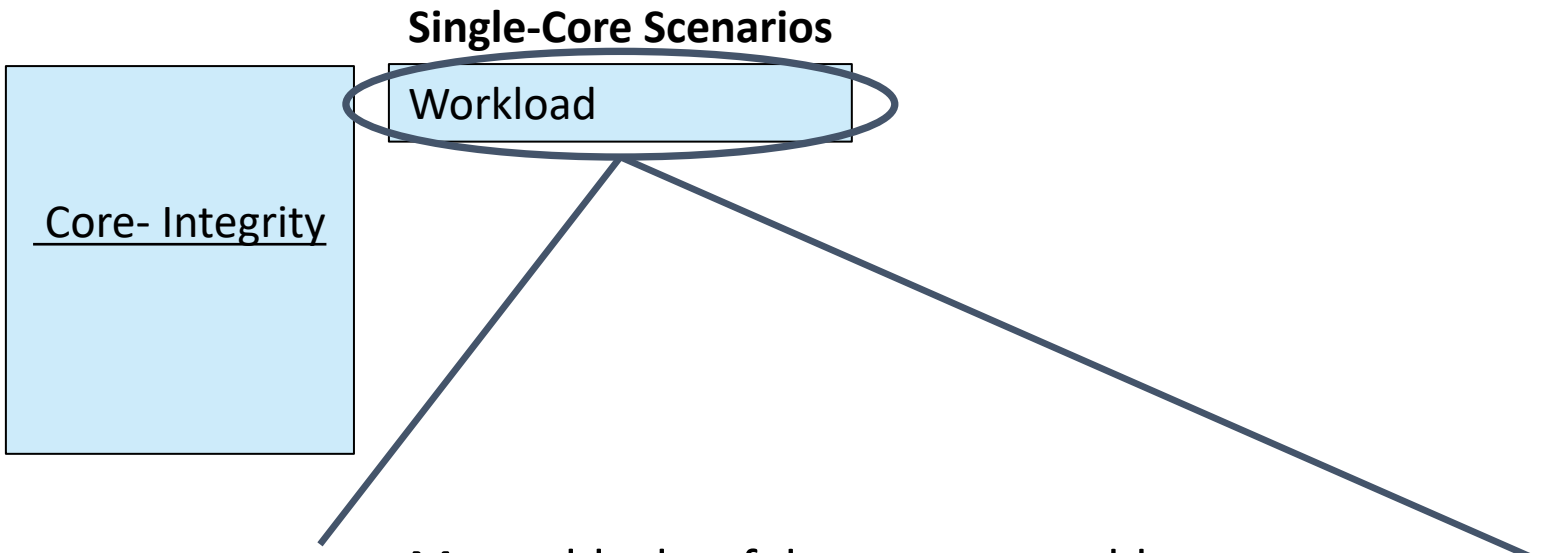
AGENDA

- Introduction to Breker Technology
- Introduction to Breker's "Core-Integrity" Tier TrekApp
- Review of Breker's "System-Integrity" Tier TrekApp
- Summary

Core-Integrity Test Generation Against the RISC-V ISA



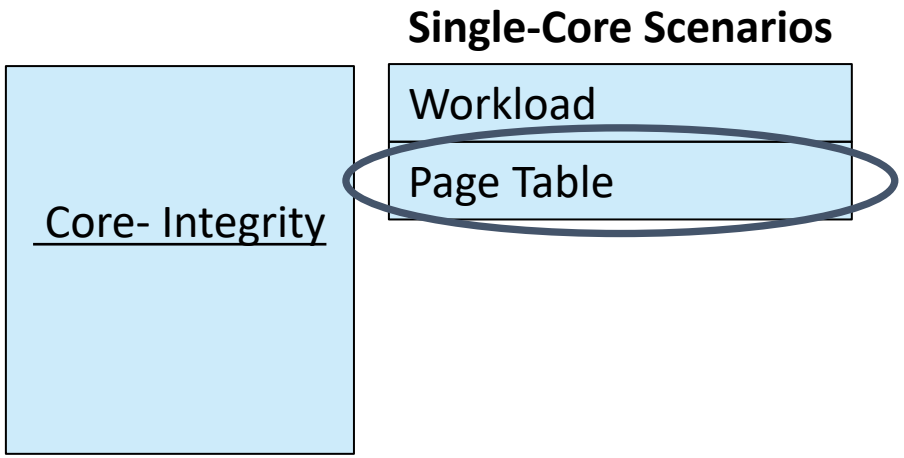
Breker App Tiers : Core-Integrity



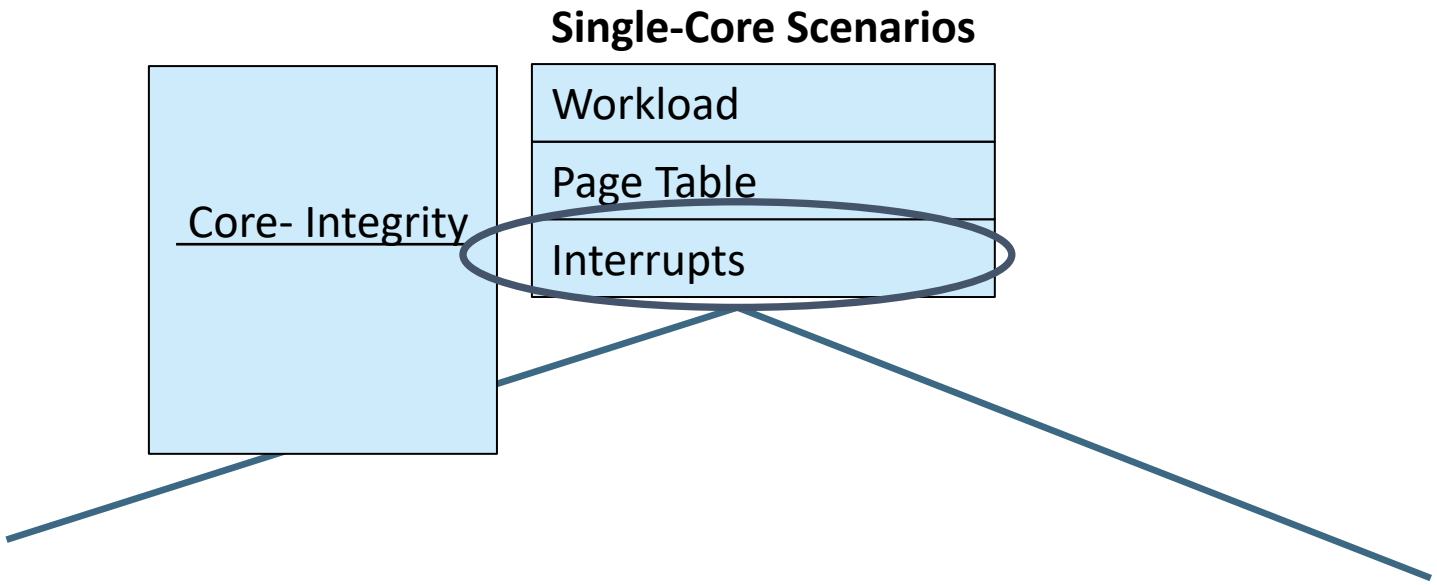
Moves blocks of data to many addresses

- Utilizes Processor's available resources/software Threads

Breker App Tiers : Core Integrity

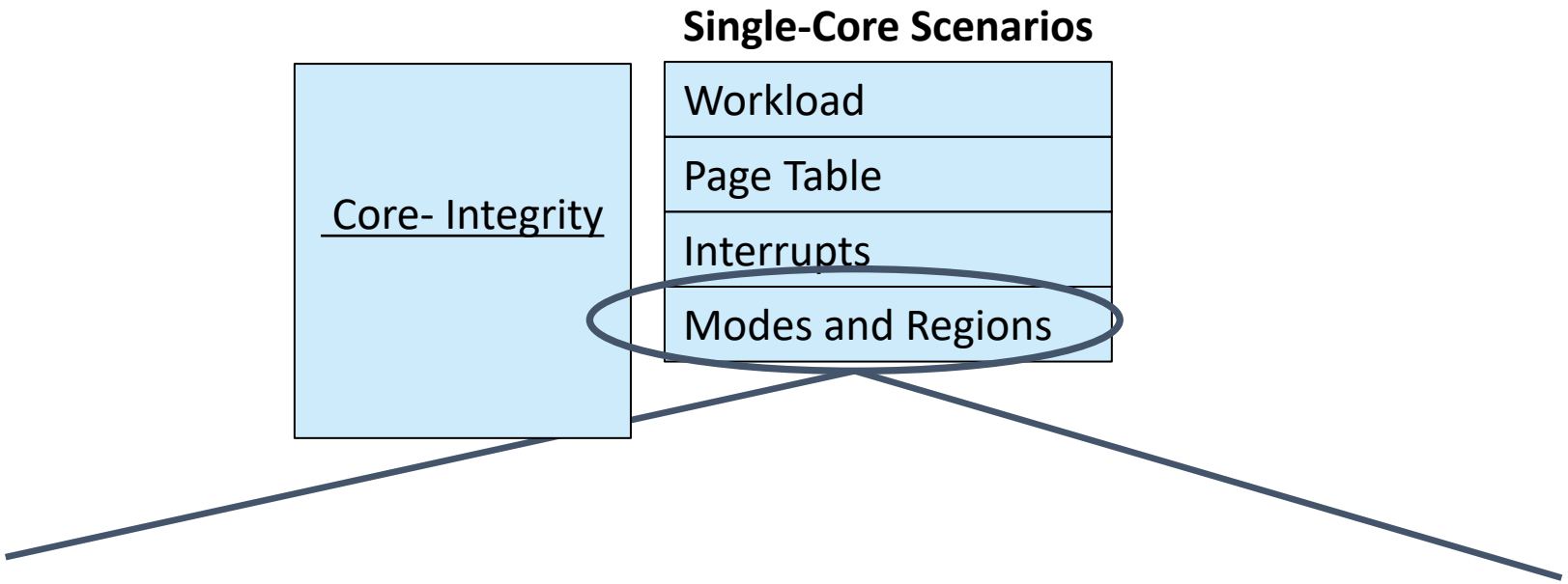


Breker App Tiers : Core Integrity



Generate random interrupts, while doing Workload Tests
(Assumes available software for GIC, registering interrupt handlers etc)

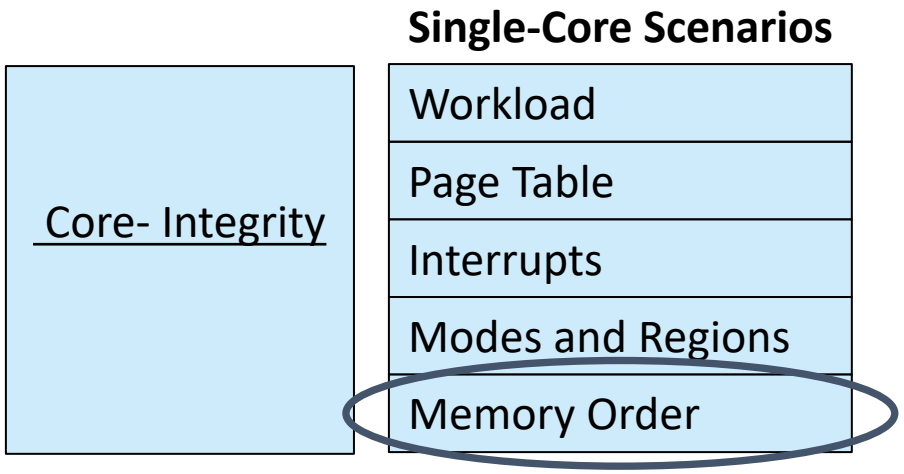
Breker App Tiers : Core Integrity



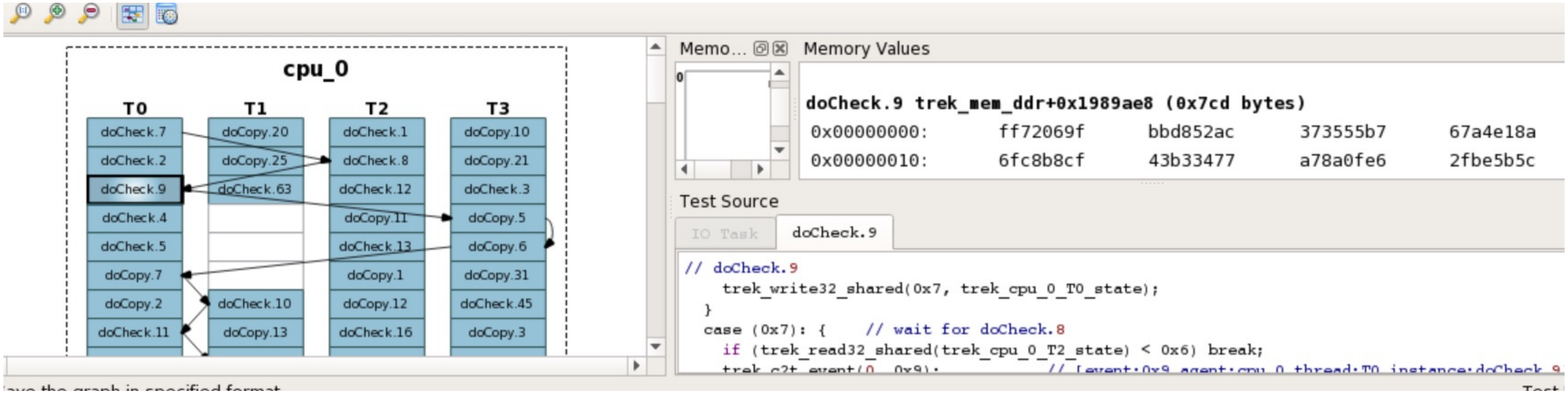
Randomly switch modes* while doing workload tests

*modes – Machine, Supervisor, User

Breker App Tiers : Core Integrity

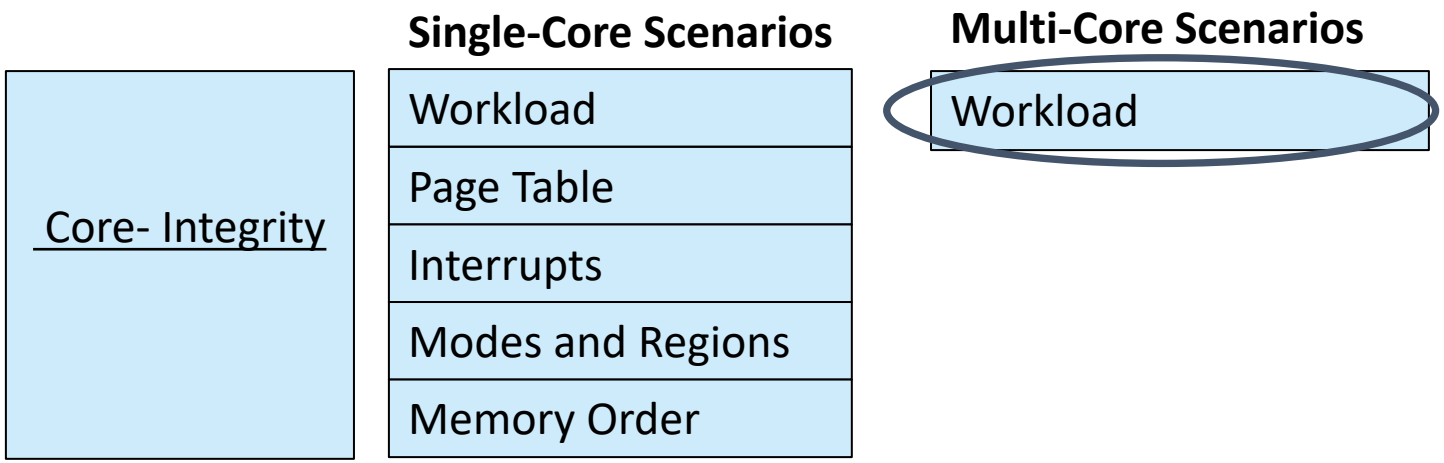


Core-Integrity: Single Core, 4 Threads

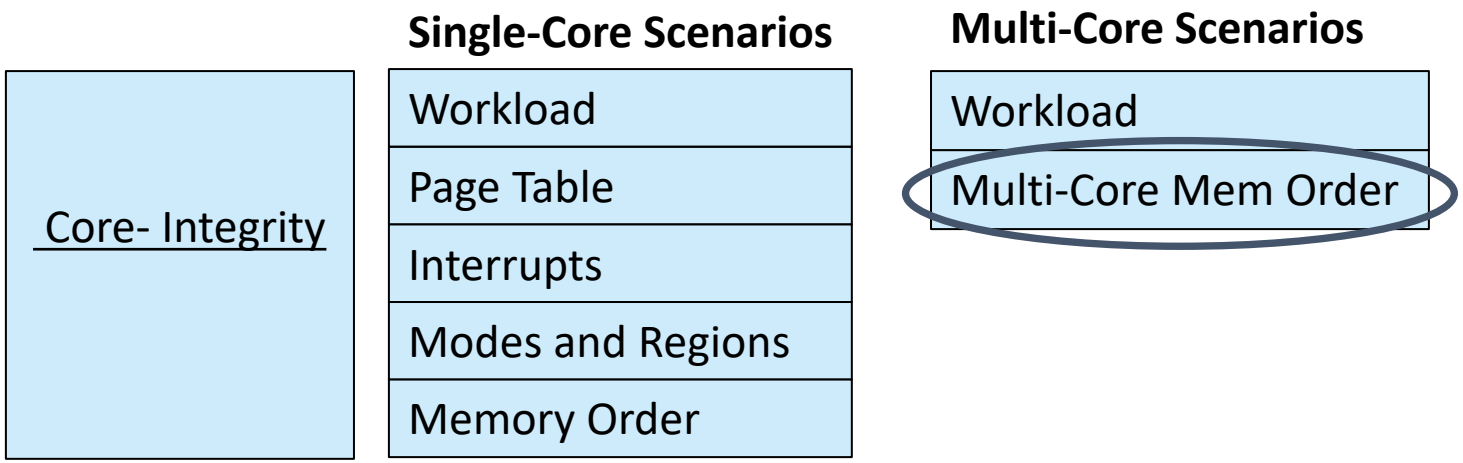


- Tests utilizes processor's available resources/software threads

Breker App Tiers : Core Integrity



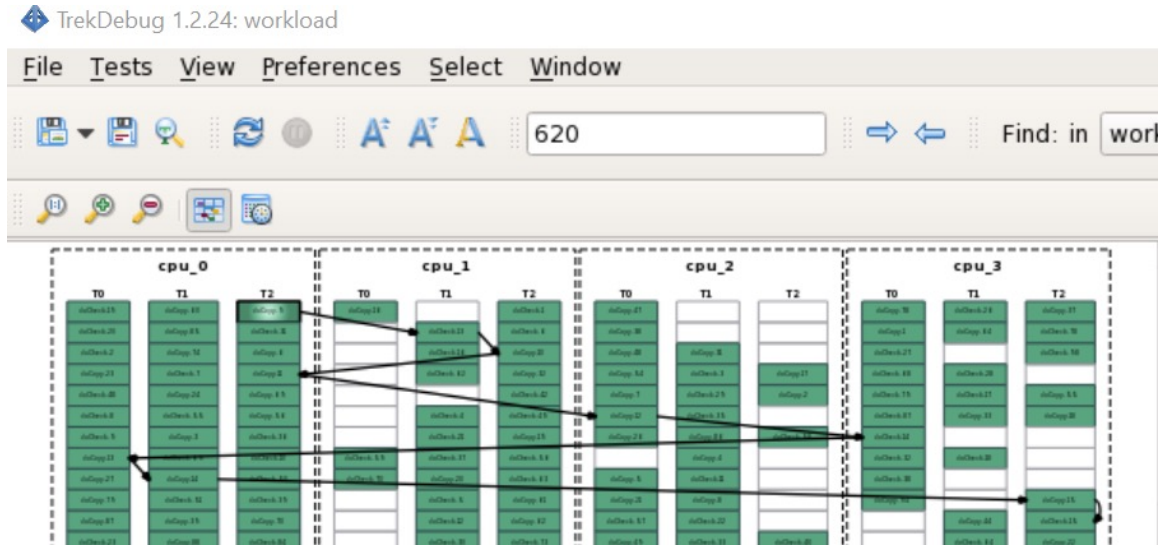
Breker App Tiers : Core Integrity



Core-Integrity: Multi-hart(x4), 3 Threads Each

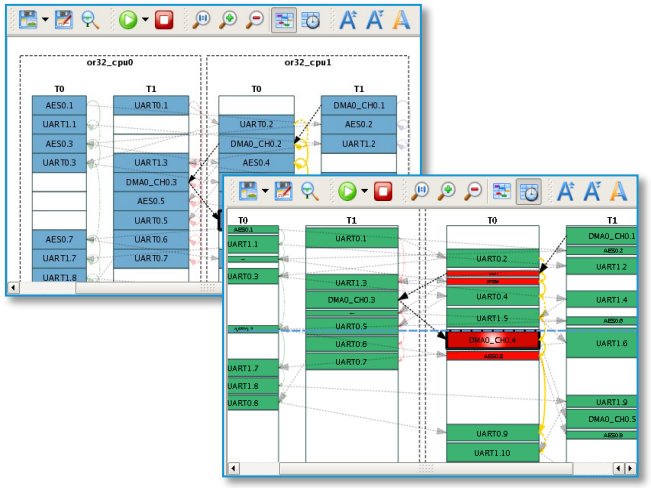
Breker Concurrent Test Scheduling Stress Tests the Processor

Advanced, Abstract Debug



Quickly observe test progress and DUT reaction

Execution Profiling



Post –run analysis of design performance/power bottlenecks

RISC-V TrekApp - Parameters

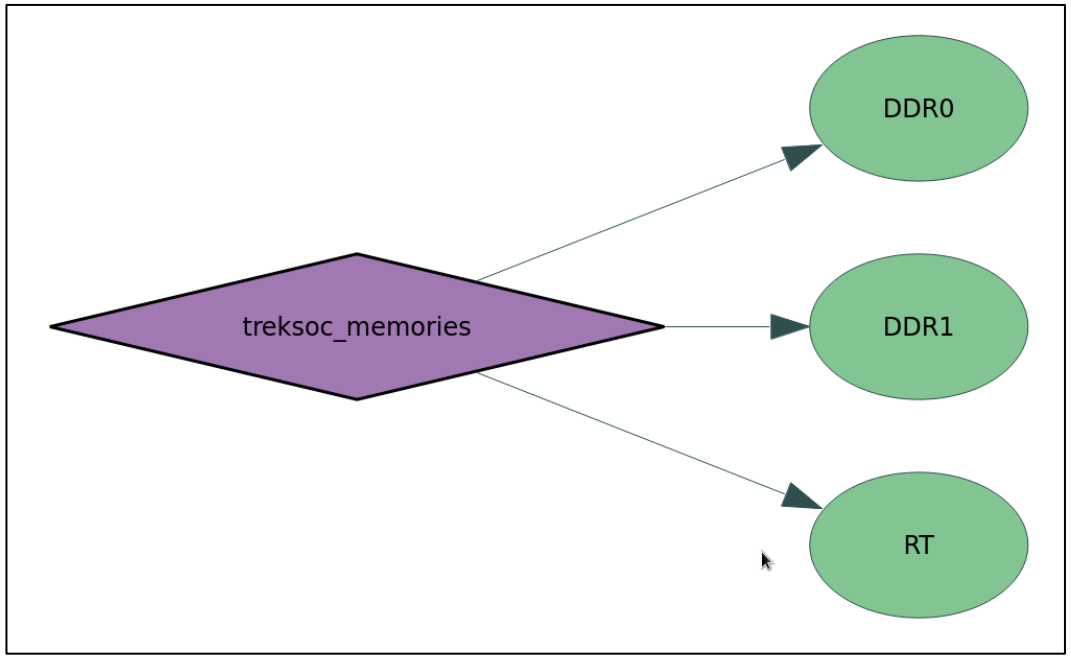
Randomization

- **Multiple Memory Regions**

	Single-Core Scenarios	Multi-Core Scenarios
Core- Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

Multiple Memory Regions

- Configure memory regions to cover
  - Different cacheability properties
  - Different memory controllers
  - Different physical memory types



RISC-V TrekApp Parameters

Randomization

- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**

	Single-Core Scenarios	Multi-Core Scenarios
Core- Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

- Must consider differing Byte operations for Loads and Stores
- All variants of Load/Store operations including:

  Acquire/Release, Exclusive, Pair Operations (16 byte), Atomics

RISC-V TrekApp Parameters

Randomization

- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**
- **Mem Allocation Strategy**

	Single-Core Scenarios	Multi-Core Scenarios
Core- Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

RISC-V TrekApp Parameters

Randomization

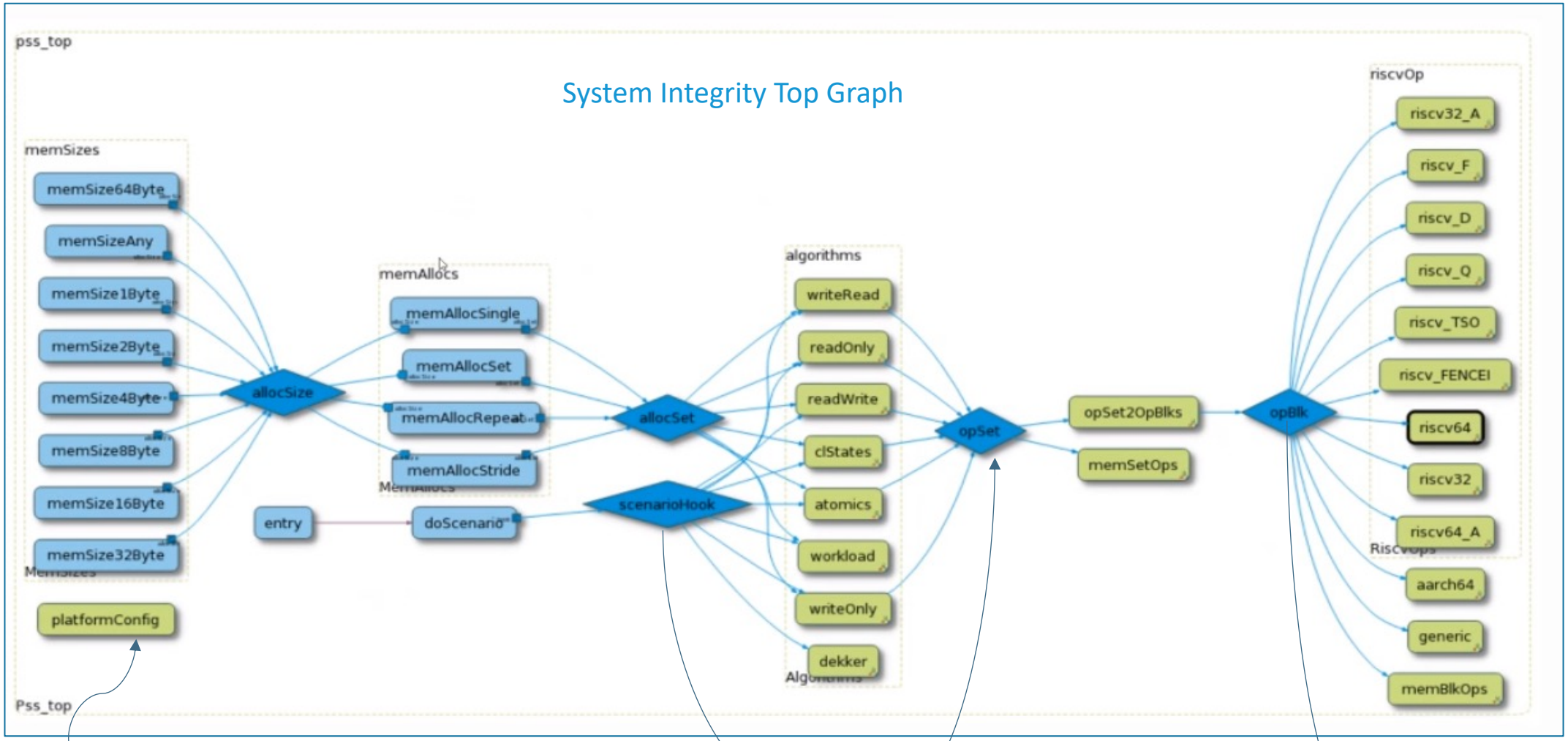
- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**
- **Mem Allocation Strategy**
- **Number of:**
  - **Harts, Threads per hart**
  - **Load/Store sources**

	Single-Core Scenarios	Multi-Core Scenarios
Core- Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

AGENDA

- Introduction to Breker Technology
- Introduction to Breker's "Core-Integrity" Tier TrekApp
- Review of Breker's "System-Integrity" Tier TrekApp
- Summary

Modular, Configurable and Extendable TrekApps



Example Customizations

Max Memsize

Specific Component Characteristic

Specialized Algorithm

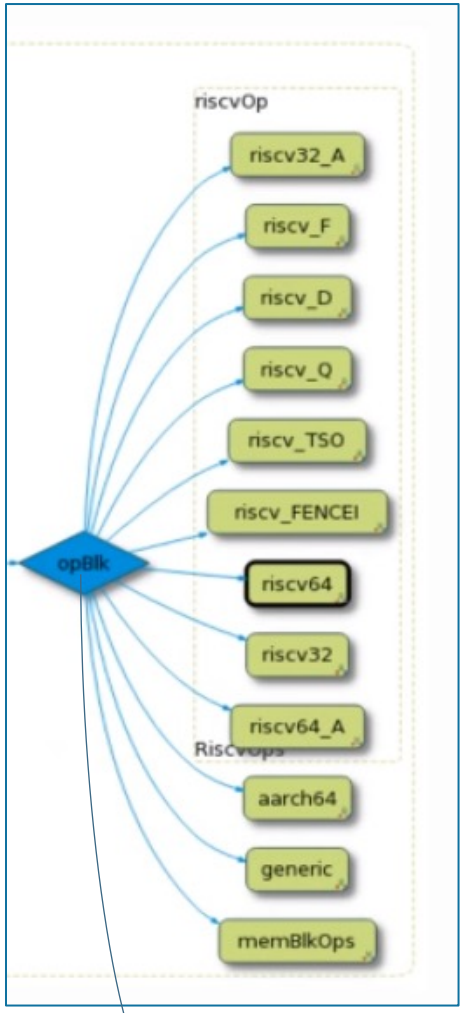
Special Coherency Test Algorithm

Add Instructions

Extra Processor Instruction

Testing a Custom Instruction

- RISC-V ISA custom instructions pose a particularly difficult verification challenge
- Custom instructions need to be tested with the processor tests, not as an after thought
- Breker solution allows custom instruction tests to be easily added into test graph
- Breker synthesis combines these tests with the app to ensure full custom processor testing



Add Instructions

Extra Processor Instruction

RISC-V TrekApp System-Integrity Tier

BREKER™

Randomization

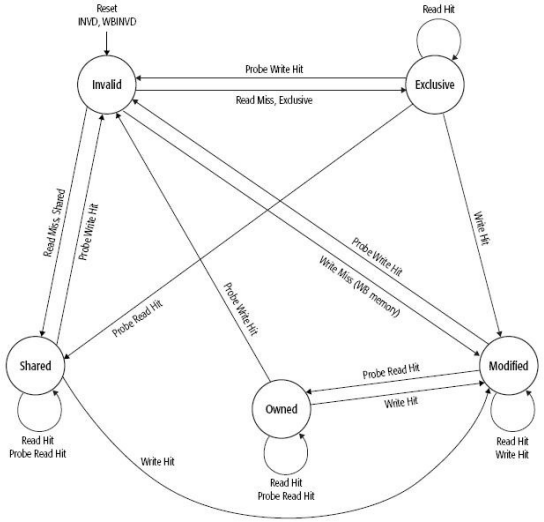
- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**
- **Mem Allocation Strategy**
- **Number of:**
  - **Harts, Threads per hart**
  - **Load/Store sources**

	Single-Core Scenarios	Multi-Core Scenarios
Core-Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

	Single-Core Scenarios
System-Integrity	Cache States/Transitions

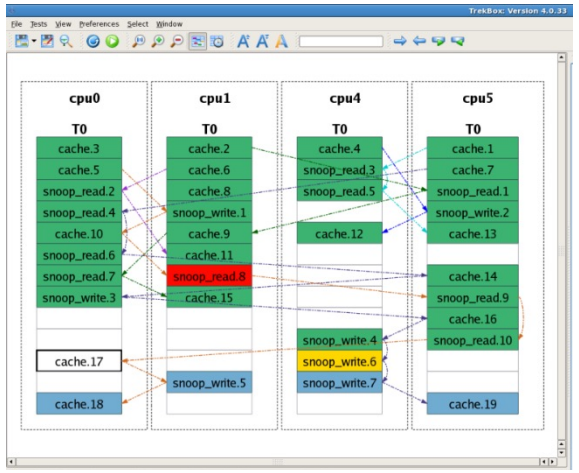
Coherency Oriented Test Generation

- "One Address, Many Data"
- Start with end state, work backwards to find transition scenario

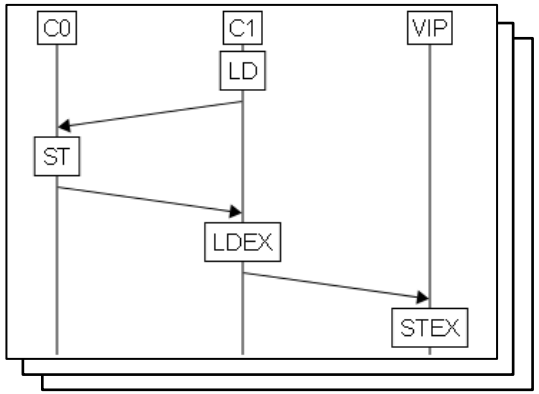


N Transition Sequences

Concurrent Scenario Test Case



N Transition Scenarios



Schedule Memory
Interleave & Pack
Resolve Dependencies

RISC-V TrekApp System-Integrity Tier

BREKER™

Randomization

- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**
- **Mem Allocation Strategy**
- **Number of:**
  - **Harts, Threads per hart**
  - **Load/Store sources**

	Single-Core Scenarios	Multi-Core Scenarios
Core-Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

	Single-Core Scenarios
System-Integrity	Cache States/Transitions
	Cross Cache line Boundary

RISC-V TrekApp Coherency Tier

BREKER™

Randomization

- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**
- **Mem Allocation Strategy**
- **Number of:**
  - **Harts, Threads per hart**
  - **Load/Store sources**

	Single-Core Scenarios	Multi-Core Scenarios
Core-Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

	Single-Core Scenarios
System-Integrity	Cache States/Transitions
	Cross Cache line Boundary
	Evictions

RISC-V TrekApp Coherency Tier

BREKER™

Randomization

- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**
- **Mem Allocation Strategy**
- **Number of:**
  - **Harts, Threads per hart**
  - **Load/Store sources**

	Single-Core Scenarios	Multi-Core Scenarios
Core-Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

	Single-Core Scenarios	Multi-Core Scenarios
System-Integrity	Cache States/Transitions	Cache Line Sharing
	Cross Cache line Boundary	
	Evictions	

RISC-V TrekApp Verification Metrics

- **Cache Line Sharing Cases**

  - Need to consider all possible cache line sharing cases across caches
    - How many caches are sharing the cache line
    - Which caches are involved
    - Is the shared line clean or modified

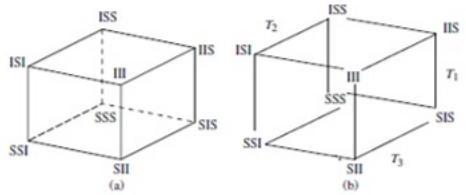


Fig. 2. (a) State space of SI protocol with 3 cores. Each global state is presented with 3 letters, e.g., IIS means core 2, core 1, and core 0 are in states I, I, and S, respectively. (b) Viewed as a composition of 3 isomorphic trees.

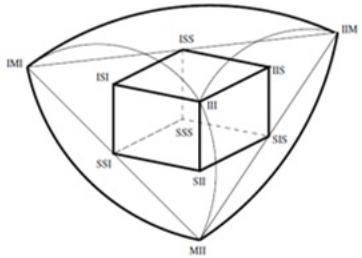


Fig. 3. State space of MSI protocol with 3 cores. For the clarity of presentation, the transitions to global modified states (IIM, IMI, MII) are omitted, if the transition in the opposite direction does not exist.

source: Qin et al., http://www.cise.ufl.edu/tr/DOC/REP-2012-537.pdf

RISC-V TrekApp Coherency Tier

BREKER™

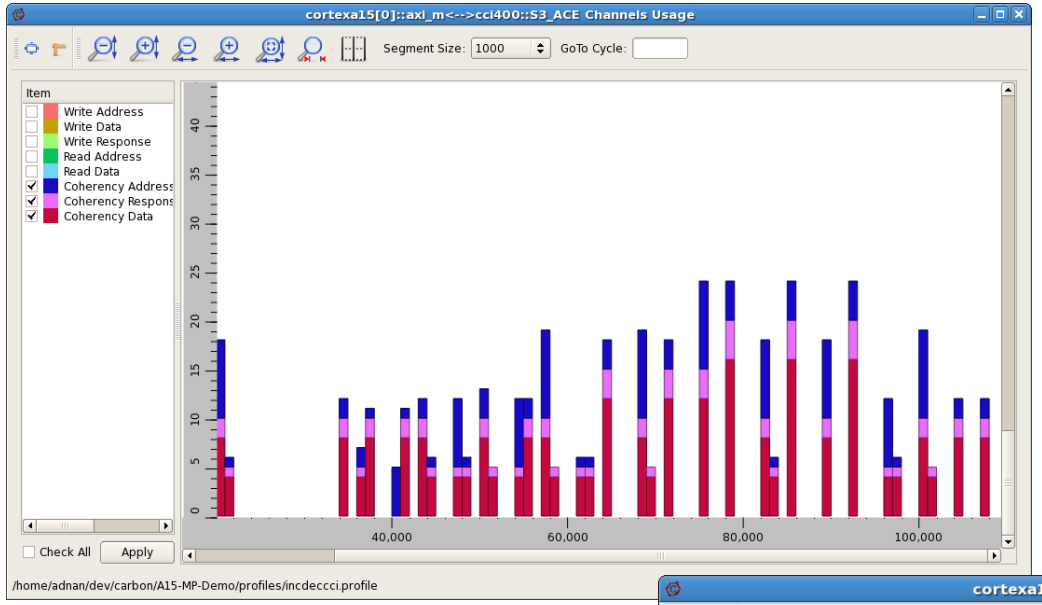
Randomization

- **Multiple Memory Regions**
- **Load/Store Sizes**
  - **(including Atomics)**
- **Mem Allocation Strategy**
- **Number of:**
  - **Harts, Threads per hart**
  - **Load/Store sources**

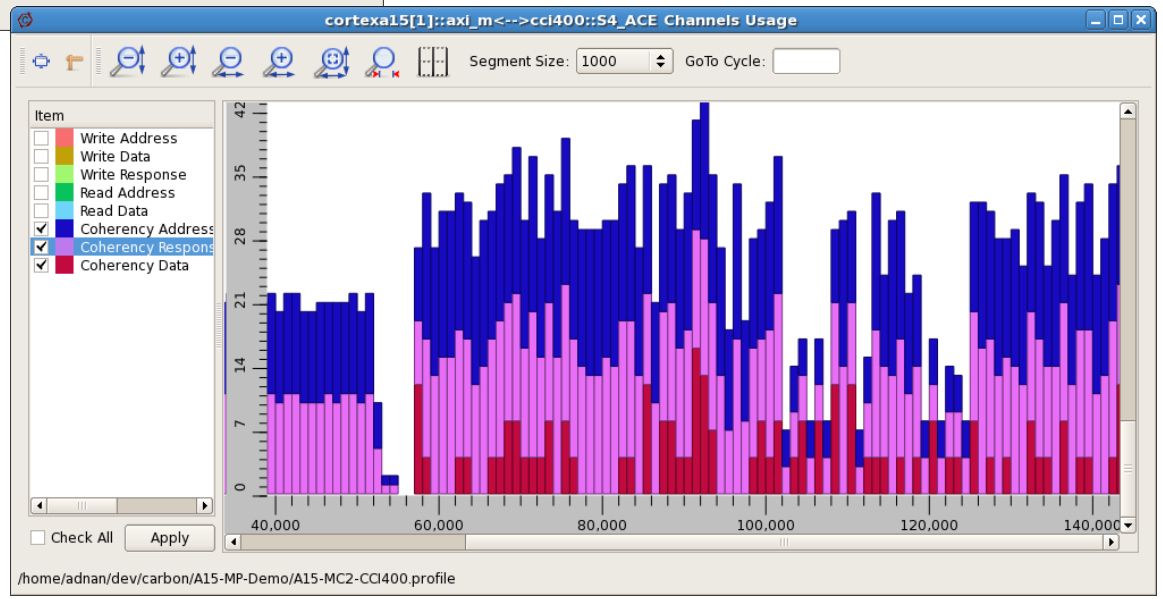
	Single-Core Scenarios	Multi-Core Scenarios
Core-Integrity	Workload	Workload
	Page Table	Multi-Core Mem Order
	Interrupts	
	Modes and Regions	
	Memory Order	

	Single-Core Scenarios	Multi-Core Scenarios
System-Integrity	Cache States/Transitions	Cache Line Sharing
	Cross Cache line Boundary	False Sharing
	Evictions	

Efficacy of System-Integrity Testing using the RISC-V TrekApp



Typical directed coherency test …



… vs. RISC-V TrekApp automated Sys-Integrity tests

AGENDA

- Introduction to Breker Technology
- Introduction to Breker's "Core-Integrity" Tier TrekApp
- Review of Breker's "System-Integrity" Tier TrekApp
- Summary

Summary - Next Steps

- Breker has been part of the verification ecosystem of processors and SoCs based on x86 and ARM architectures

- Breker is also becoming part of the verification ecosystem of processors and SoCs based on RISC-V architectures
  - We already working with multiple RISC-V developers and users/integrators

- We also believe that RISC-V has room to grow
  - We have been there throughout the evolution of x86 and ARM systems, so we know what challenges lay ahead as RISC-V grows and succeeds

Thank You For Listening

For more information:

- Go to **brekerSystems.com**
- Email: **info@brekersystems.com**
- My email **johns@brekersystems.com**