2023
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Security Verification using Perspec/Portable Stimulus

Junxia Wang, Leven.Li, Siyan.Li - Mediatek

A T S Prasad, Kiran Kumar Palla, Yung Cheng Chen - Cadence

MEDIATEK

cadence

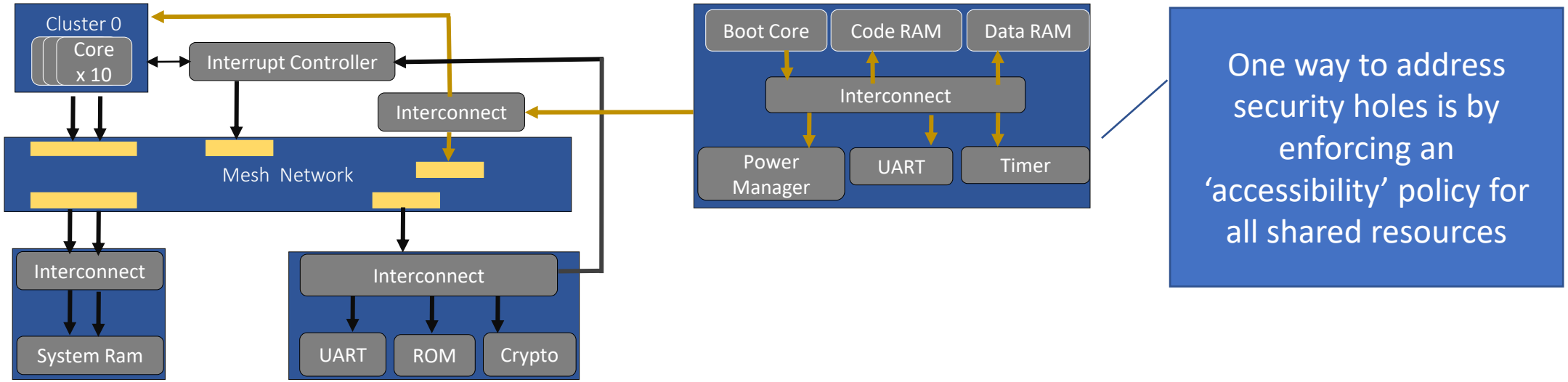
accellera SYSTEMS INITIATIVE

Agenda

- Why security verification?
- Requirements & Challenges
- Using Perspec and PSS to create reusable security verification scenarios

Trends & Risks

- Open access of automotive, mobile or data center to shared memory and resources by all agents in the system leaves security holes in the SoC design



Agenda

- Why security verification?
- Challenges
- Using Perspec and PSS to create reusable security scenarios

Security Hardware Block - PMP

- RISC-V architecture uses Physical Memory Protection Unit (PMP) [3] to enforce the ‘accessibility’ aspect of security :
  - PMP hardware unit limits the physical addresses accessible by software programming
  - PMP unit tackles the security aspect related to physical memory access privileges – read, write, execute permissions – in different execution modes

Overview of RISC-V PMP Specification

- There are three kinds of privilege modes in RISC-V environment
- PMP policies are checked for all accesses whose privilege mode is either S or U.
- S or U mode accesses without proper permissions will trigger access fault exceptions

Level (Mode)	Name	Abbreviation
0	User	U
1	Supervisor	S
3	Machine	M

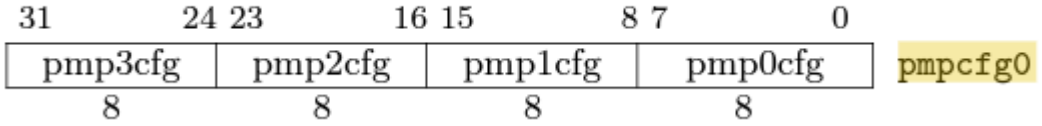
RISCV-64 Privilege Level (Mode)

Overview of RISC-V PMP Specification

- A PMP unit supports dividing the physical address space into different regions with varying access permissions.
- Number of regions could be 0 (no PMP), 16 or 64 entries
- Achieved by configuring *pmpcfg* and *pmpaddr* registers for each region

Overview of RISC-V PMP Specification

- For RV32, 16 CSRs, pmpcfg0-pmpcfg15, hold the PMP configurations pmp0cfg-pmp63cfg
  - Example: PMP region-01 configuration is programmed in pmpcfg0.pmp1cfg field
- *pmpaddr* registers hold the address of the corresponding PMP region
- Each pmpNcfg field has permission bits
  - Read, Write, Execute
  - Address matching
  - Lock bit



A	Name	Description
0	OFF	Null region (disabled)
1	TOR	Top of range
2	NA4	Naturally aligned four-byte region
3	NAPOT	Naturally aligned power-of-two region, ≥8 bytes

Overview of RISC-V PMP Specification

- With a combination of PMP configurations and address matching modes, lock bit, the PMP unit enhances security by supporting granular control of permissions across multiple physical memory regions
- The permissions can be dynamically re-programmed by each CPU core to enforce its own security policy on the shared system memory and resources

Challenges

- Security verification is hard:
  - PMP unit - a programmable hardware security block - allows multiple memory regions to be specified, each with its own privilege access policy per CPU core
  - Complex due to large space of concerning crosses of PMP regions, multi-cores, access policies
  - The complexity is amplified when Physical Memory Attributes (PMA)[3] - like Shareability, Cacheability, Exclusiveness - are thrown into the verification mix
  - Not easy to create tests and test infrastructure to verify negative security scenarios

Agenda

- Why security verification?
- Requirement & Challenge
- Using Perspec and PSS to create reusable security scenarios

Security Verification using Portable Stimulus

- Perspec and PSS simplified and shortened our verification tasks
  - Basic memory operation actions are provided by Perspec libraries
  - Building block Security actions are also provided
- Overall verification process
  - Model compute subsystem
  - Model PMP features
  - Develop PMP security test scenarios
  - Create test variations to cover concerning cross of PMP features

Model Compute Subsystem

- Processor Info Table

@table: processor_info													
@package: sml_pkg	@size_const: NUM_OF_CORES	@struct: sml_processor_info_s											
#tag	#kind	#cluster	#cluster_id	#core_id	#enabled	#scheme	#hdl_path	#coherency_level	#exclusive_able	#power_d...	#powers_...	#barrier_...	#clock_p...
hart0	RISCV	R0	0	0	TRUE	RISCV_C	NA	FULL	TRUE	FALSE	TRUE	TRUE	80ns
hart1	RISCV	R0	0	1	TRUE	RISCV_C	NA	FULL	TRUE	FALSE	FALSE	TRUE	80ns
hart2	RISCV	R0	0	2	TRUE	RISCV_C	NA	FULL	TRUE	FALSE	FALSE	TRUE	80ns
hart3	RISCV	R0	0	3	TRUE	RISCV_C	NA	FULL	TRUE	FALSE	FALSE	TRUE	80ns

- Memory Info Table

@table: memory_info								
@package: sml_pkg	@size_const: NUM_OF_MEM_B...	@struct: sml_memory_info_s						
#mem_block	#base_addr	#end_addr	#alignment	#enabled	#backdoor_enabled	#exclusive_able	#atomics_supported	#is_l2_lim
mem0	0x90000000	0x97FFFFFF	1	TRUE	FALSE	TRUE	TRUE	TRUE
mem1	0x98000000	0x99FFFFFF	1	TRUE	FALSE	TRUE	TRUE	TRUE
mem2	0xA0000000	0xA1FFFFFF	1	TRUE	FALSE	TRUE	TRUE	TRUE

Model PMP Features

- Physical Memory Protection (PMP) table- PMP entries in CSV tables

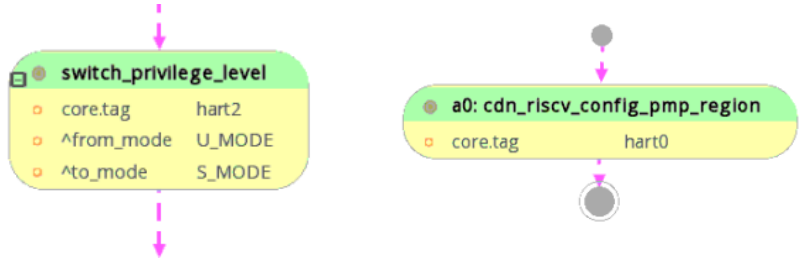
#region	#start_pa	#size	#region_type	#region_permissions
0	0x0000_0000	2G	TOR	L,R,W,X
1	0x8000_0000	2M	TOR	L,R,W,X
2	0x9000_0000	2M	TOR, NA4, NAPOT	W,X
3	0x9000_0000	16M	TOR, NA4, NAPOT	W,X
4	0x9200_0000	2M	NAPOT	R
5	0x9200_0000	8M	TOR, NA4, NAPOT	R,W
6	0x9300_0000	2M	TOR, NA4, NAPOT	R,W,X
7	0x9400_0000	16M	TOR, NA4, NAPOT	L,R,W
8	0x9200_0000	16M	TOR, NA4, NAPOT	L,R

Model PMP Features

- Physical Memory Attributes (PMA) table

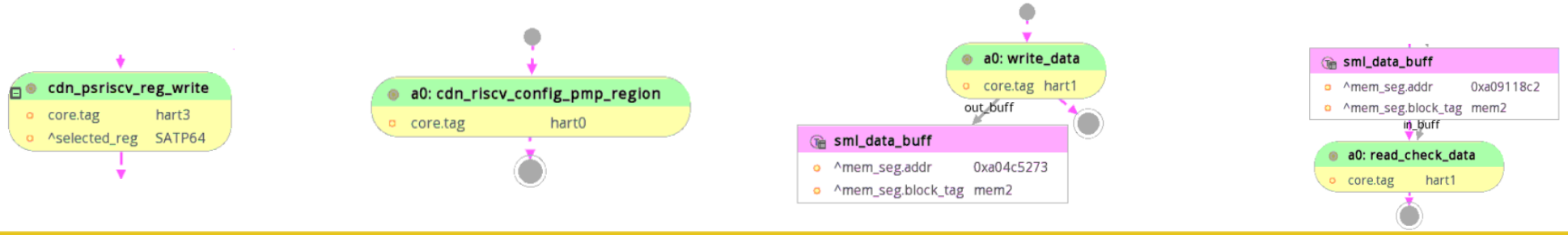
#va	#pa	#mem_block	#size	#shareability
0x9000_0000	0x9000_0000	mem0	128M	shareable
0x9800_0000	0x9800_0000	mem1	32M	shareable
0xA000_0000	0xA000_0000	mem2	32M	shareable

- Atomic PMP actions
  - Switch execution mode between M, S, and U modes
  - Configure (program) PMP registers based on the configuration specified in PMP table



Model PMP Features

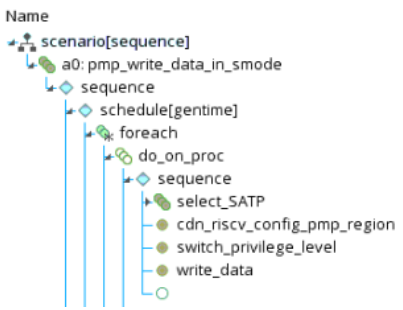
- Atomic PMP actions
  - Program RISC-V system register with specific value
  - Configure (program) PMP registers based on the configuration specified in PMP table
  - Write random data of specified size to a selected memory block
  - Read and check (previously written) data from a selected memory block



Security Test - Scenarios/ Solution

- PMP Write Data in S(supervisor)-Mode
  - Enable address translations in S-mode by Programming SATP (Supervisor Address Translation and Protection) register
  - Configure PMP entries as defined in PMP table
  - Switch privilege level from M to S mode
  - Perform write access in S-mode

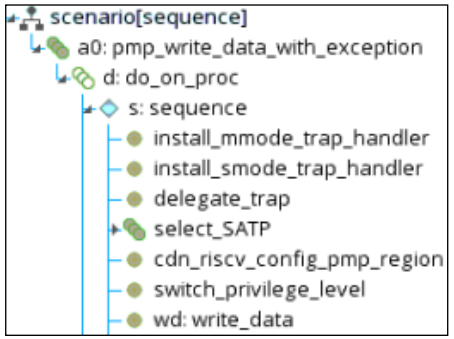
Solution



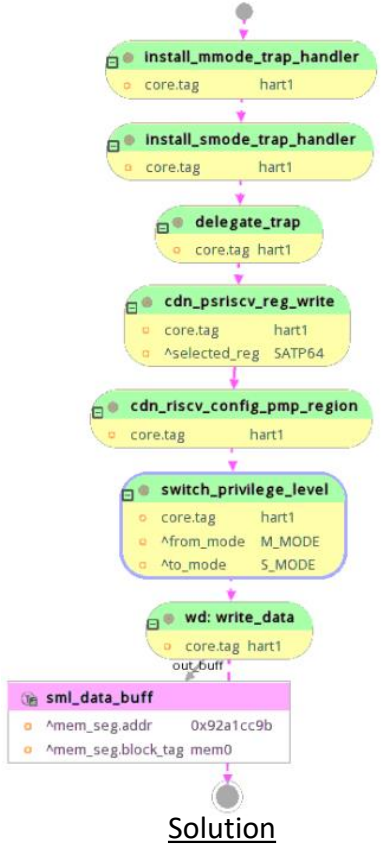
PSS Scenarios

Security Test – Scenarios/ Solution

- PMP Write Data With Exception
  - Install trap handlers to handle exception due to PMP violations
  - Enable address translations in S-mode
  - Configure all PMP region based on the PMP table
  - Switch privilege level from M to S mode
  - Do a write access to a PMP region that has no write permission.



PSS Scenarios

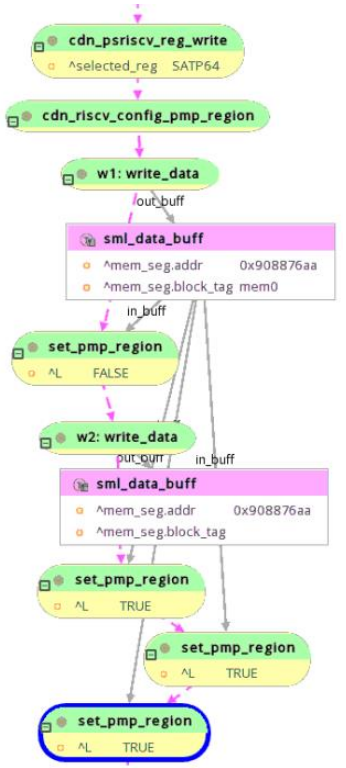


Solution

Security Test – Scenarios/ Solution

- PMP Region Overlap
  - Validate PMP privilege accesses with overlapping memory address regions (solution 1)
- PMP Change Lock
  - Enable PMP lock bit to force privilege access checks even in machine mode (solution 2)

Solution 1



Solution 2

Conclusions

- Writing security scenarios using Perspec enables efficient verification of complex SoC level security scenarios
  - Out-of-box atomic actions and scenarios are ready to be used flexibly
  - Create large number of tests, covering all crosses, in a relatively short amount of time.
  - Model both positive and negative security test scenarios with PSS
  - RISC-V security verification approach focusing on PMP features