Formal Verification of Security-Properties on RISC-V Processors

Czea Sie Chuah, Christian Appold, and Tim Leinmüller

Motivation

- High security demands of upcoming applications, e.g. autonomous driving
- In past years several famous bugs and security-vulnerabilities in processors found
- Design flaws can be exploited by attackers

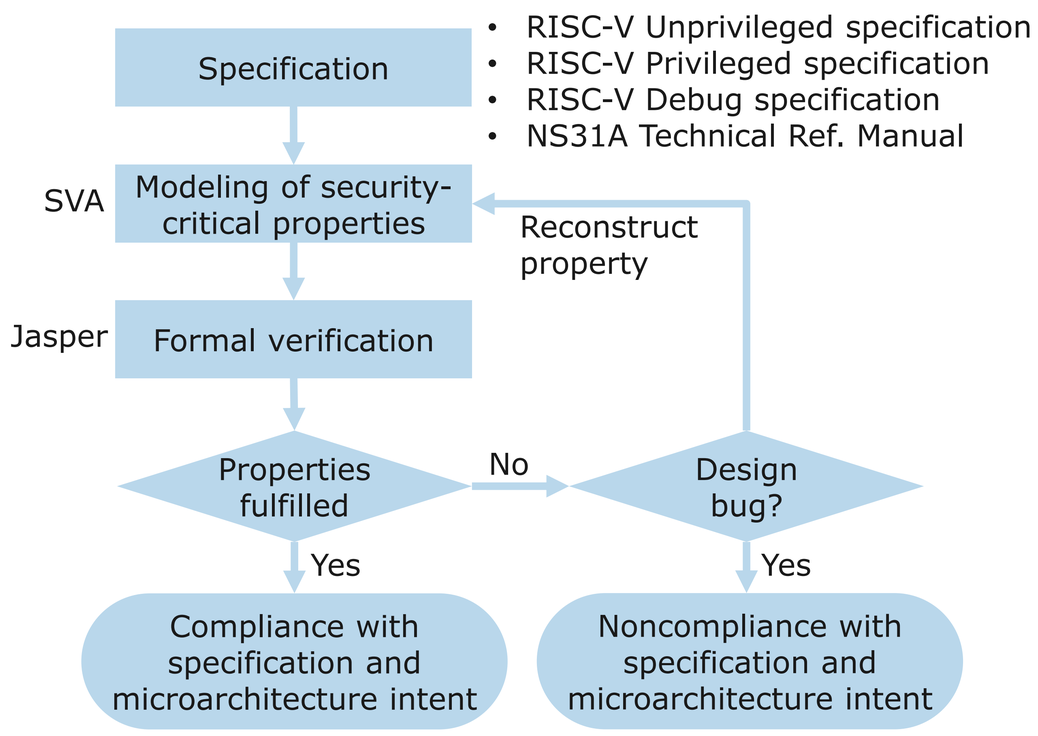
Formal verification of security-critical functionality required

- RISC-V security verification guidance needed
- **<u>Our work increases RISC-V security by:</u>**
  1. **Comprehensive identification of security-critical functionality**
  2. **Derivation of properties for security-critical functionality correctness**
  3. **Formal verification of the properties**

Collaboration with TUM Chair of Security in Information Technology

Methodology

- Akaria NS31A commercial RISC-V processor verified
  - 32-bit, 4 pipeline stages, 3 privilege modes
- Verification using Cadence Jasper Formal Verification Platform



Work presents comprehensive set of security-properties for RISC-V security hardening

Security-Properties

- <u>Identified security-critical functionality:</u>

Instruction Execution *Check instruction flow through pipeline*	**CSRs** *Comply with CSR access rules*	**Debug Operation** *Comply with Debug CSR access rules*
Exception and Interrupt *Proper handling required*	**Mode Transition** *Mode transition rules need to be met*	**PMP** *Access control rules for memory regions need to be met*
Control Flow *Correct setting of program counter*	**Register Update** *Correct target register is updated*	**Memory Access** *Value and address of memory transfers as intended*

- <u>Example mode transition SVA properties:</u>

When exception happens, privilege mode is set to machine mode:

```
assert property (@(posedge clk) disable iff (rst)
        exception_happen |-> ##1 PrivMode==MachineMode);
```

When return from debug/machine mode, MRET update privilege mode with the previous mode saved:

```
assert property (@(posedge clk) disable iff (rst)
        mret_occured |-> ##1 PrivMode==$past(MstatusMPP));
```

Results

- <u>Used Jasper settings:</u>
  - Jasper contains formal verification engines for full-proofs and bug-hunting
    - engines use different model checking techniques, e.g. BDD or SAT-based
  - verified all properties of a category in parallel
  - use of several Jasper engines in parallel
  - automatic choosing of most suitable engine
- We identified 1146 properties and grouped them under 9 categories
- Achieved full-proof for passing properties
- Runtime: Control Flow < 24h, others < 4000s

Categories	Properties		
	Assertion	***Pass***	***Fail***
Instruction Execution	10	10	0
CSR	394	280	114
Debug Operation	14	14	0
Exception and Interrupt	87	87	0
Mode Transition	13	13	0
PMP	574	574	0
Control Flow	9	8	1
Register Update	33	33	0
Memory Access	12	12	0
Total	**1146**	**1031**	**115**

3 bugs found

SVA : System Verilog Assertions, CSR : Control and Status Register, PMP: Physical Memory Protection

DENSO
Crafting the Core