SecureBOOM: Mitigating Spectre in an Out-of-Order RISC-V Core with a Formally Backed Design Flow

Tobias Jauch[1], Alex Wezel[1], Mohammad R. Fadiheh[2], Philipp Schmitz[1], Dominik Stoffel[1] and Wolfgang Kunz[1]

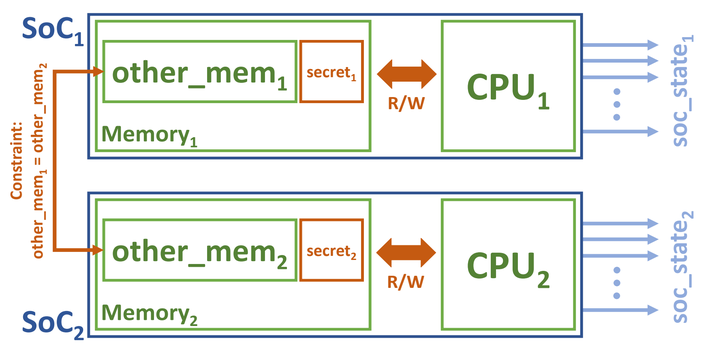
[1] RPTU Kaiserslautern-Landau, Germany [2] Stanford University, USA

Transient Execution Attacks

- Transient Execution Side Channel (TES) attacks exploit speculative and out-of-order execution in modern CPUs
- Attackers can trick the CPU into transiently accessing confidential data, thereby leaving a footprint in microarchitectural buffers, even though the architectural state is unaffected
- Examples are **Spectre**, **Meltdown** or the **MDS attacks**

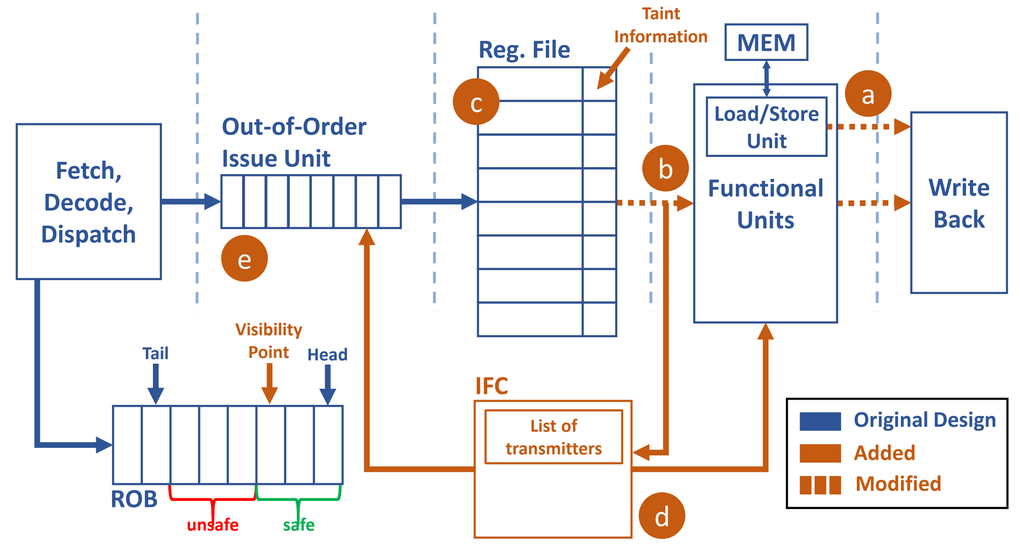
UPEC: Formal RTL Security Verification

- ✓ Exhaustively detect Transient Execution Side Channels in RTL implementations
- ✓ No need for *a priori* knowledge on attacks



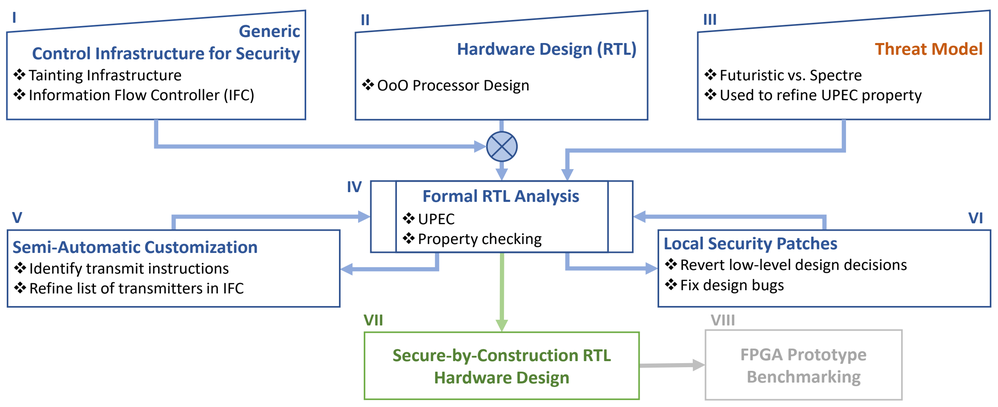
Generic Control Infrastructure for Security

- Taint result of unsafe loads (a) → propagate taint (b) → clear taint of safe instructions (c)
- IFC kills tainted transmit instructions (d) → issue queue re-issues them once they become safe (e)
- ✓ Allows to mitigate transmitters in a generic and centralized way



Secure-by-Construction Design Flow

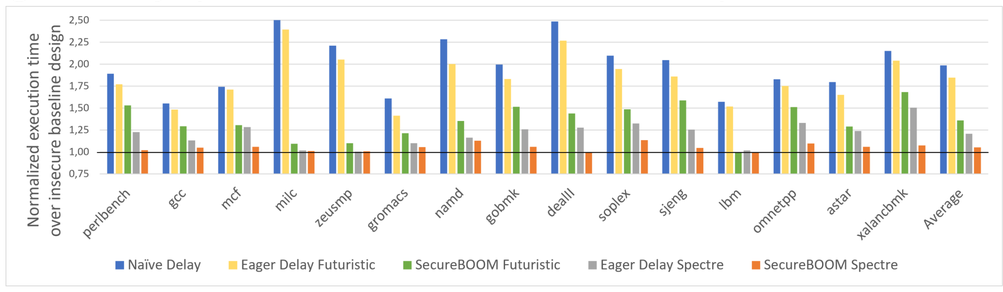
- Instrument initial design (**II**) with generic control infrastructure for security (**I**) and setup UPEC proof (**IV**) according to threat model (**III**)
- Counterexample points to a transmit instruction → add it to the list of transmitters in the IFC (**V**) or
- Implement local security patch if the vulnerability is introduced by a low-level design decision (**VI**)
- ✓ Interleave formal security verification by UPEC with design steps to iteratively verify and patch the design
- ✓ Allows for aggressive optimizations without risking the security of the design
- ✓ Separation of concerns by decoupling tool-based security analysis and manual design tasks



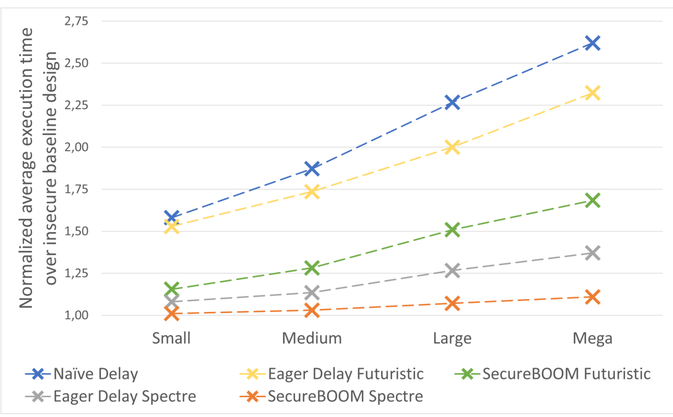
SecureBOOM - Performance

- Implementation of generic control infrastructure for security in BOOMv3
- ✓ Average performance overhead over unsafe baseline (medium configuration) for SPEC CPU 2006 benchmark suite:

Naïve Delay	98.5 %		
Eager Delay *futuristic*	84.6 %	SecureBoom *futuristic*	36.0 %
Eager Delay *spectre*	20.9 %	SecureBoom *spectre*	5.2 %



- Average overhead of SPEC CPU 2006 test workloads for different core sizes of BOOM
- ✓ Performance gain of SecureBOOM designs compared to conservative fixes increases with higher pipeline complexity (for each threat model)



SecureBOOM - Verification

- Set up the computational model and the UPEC property
- Used sound blackboxing to improve scalability of the proofs
- Cone-of-influence reduction enables parallelization of the proofs
- ✓ Found 29 transmit instructions, Meltdown vulnerability and two bugs in the taint propagation logic
- ✓ Secure design after 12 iterations
- ✓ The last (and longest) iteration finished after proving 329 properties (can run in parallel) and each property check took around 2 hours on average

Conclusion

- First formally verified RTL implementation of an out-of-order processor capable of running Linux OS featuring exhaustive TES mitigations with competitive performance
- ✓ High robustness against design mistakes thanks to the exhaustive nature of formal verification
- ✓ Design flow establishes a separation of concerns between implementation and security, relieving the designer from having security in mind with every design decision

Find our design on GitHub

ICCAD'23 Paper

RPTU Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

Stanford University

The reported research was partly supported by **DFG** SPP Nano Security, KU 1051/11-2 and by the **Intel** Corp. Scalable Assurance Program.

RISC-V Summit Europe, Munich
June 24th – June 28th 2024