Conclusions and Bigger Picture

Caroline

Stanford | ENGINEERING

Summary: Addressing the Verification Challenge

Verification Challenge – a "no manual property writing" approach:

For general-purpose contracts: Template-based property generation and decomposition

- **RTL2MμPATH:** Uncovering bugs in hardware implementations of **functional contracts**
- **SynthLC:** Verifying **SystemVerilog RTL** against **side-channel leakage contracts**
- **RTL2μSPEC + Blend:** Verifying **core SystemVerilog RTL** against **memory consistency models**
- **qcmbr:** Verifying **coherence protocol Murphi models** against **memory consistency models**

For bespoke (functional) contracts: AI-assisted property generation and decomposition

Critical bugs escape to first-silicon in >85% of HW projects (20-yr first-silicon success low!) w/ >50% effort on verification [Siemens, 2024].

GETTING WORSE →

Due to more contracts, driven by design diversity and solving The Contracts Challenge; and more bugs + shorter design times, driven by AI design.

Ongoing Work: Addressing the Verification Challenge

Outer-loop optimizations: Decomposing end-to-end verification tasks into many simple ones

- More contracts with synthesis-based approach (e.g., execution contracts [Zhu+, YARCH'26])
- Aggressive AI- and formal methods-driven decomposition for functional verification

Inner-loop optimizations: Scaling checks for hard-to-prove decomposed properties

- Inductive strengthenings leveraging semantic design information (e.g., [Qian+, YARCH'26])
- Integration of automated and interactive theorem provers

Design for verification to reduce the fraction of hard-to-prove decomposed properties

Early-stage verification of high-assurance requirements targeting simulators (e.g., [Eaton+, YARCH'26])

Takeaways & Conclusions

Rethinking the industrial approach to hardware verification – critical for enabling AI-driven design

- Automated property generation, given architect-friendly design metadata
- Automated decomposition of end-to-end properties for scalable checking with automated formal tools

Distilling the hardest parts of hardware verification tasks into a minority of properties to be solved with inner loop optimizations

Broader impact:

- SynthLC, RTL2μSPEC, and ARTEMIS evaluated in commercial settings, keynote at Jasper User Group conference
- ***All work is open source and designed to be usable and extensible.***

ACKNOWLEDGMENTS • https://trippel-lab.stanford.edu/

High Assurance Computer Architectures Lab

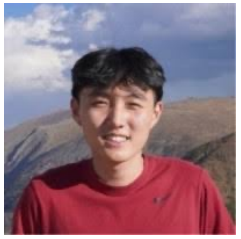
Caroline Trippel
Asst. Prof. of CS & EE

Yao Hsiao
PhD Student

Nick Mosier
PhD Student

Saranyu Chattopadhyay*
PhD Student

Daniel Mendoza
PhD Student



William Zhou
Coterm Student

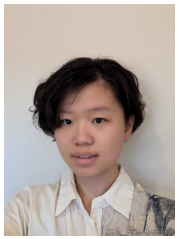
Ioanna Vavelidou
PhD Student

Rachel Cleaveland*
PhD Student*

Samantha Archer
PhD Student

Yicheng Qian*
PhD Student

Anna Eaton
PhD Student



Xiaofu (Kaia) Li
Coterm Student

*co-advised