Caroline Trippel

Assistant Professor of Computer Science and Electrical Engineering, Stanford University

Stanford University
Computer Science Department
353 Serra Mall, Stanford, CA 94305

Phone: (574) 276-6171
Email: trippel@stanford.edu
Home: https://cs.stanford.edu/~trippel

Education

2013–2019 **Princeton University**, PhD, Computer Science / Computer Architecture
Thesis: Concurrency and Security Verification in Heterogeneous Parallel Systems
Advisor: Prof. Margaret Martonosi

2013–2015 **Princeton University**, MA, Computer Science / Computer Architecture

2009–2013 **Purdue University**, BS, Computer Engineering

PhD Dissertation Research

Despite parallelism and heterogeneity being around for a long time, the degree to which both are being simultaneously deployed poses grand challenge problems in computer architecture regarding ensuring the accuracy of event orderings and interleavings in system-wide executions. As it turns out, event orderings form the cornerstone of correctness (e.g., memory consistency models) and security (e.g., speculation-based hardware exploits) in modern processors. Thus, my dissertation work creates formal, automated techniques for specifying and verifying the accuracy of event orderings for programs running on heterogeneous, parallel systems to improve their correctness and security.

Awards and Honors

[1] Recipient of the 2020 ACM SIGARCH/IEEE CS TCCA Outstanding Dissertation Award
[2] CheckMate chosen as an IEEE MICRO Top Pick of 2018 (top 12 computer architecture papers of 2018)
[3] Selected for 2018 MIT Rising Stars in EECS Workshop
[4] Selected for 2018 ACM Heidelberg Laureate Forum
[5] TriCheck chosen as an IEEE MICRO Top Pick of 2017 (top 12 computer architecture papers of 2017)
[6] NVIDIA Graduate Fellowship Recipient, Fall 2017–Spring 2018
[7] NVIDIA Graduate Fellowship Finalist, Fall 2016–Spring 2017
[8] Richard E. Houser Trustee Scholarship in ECE (Purdue University, Full Tuition), Fall 2009–Spring 2013
[9] Michelle L. Ruth Memorial Scholarship (Purdue University), Fall 2012
[10] Thomas A. Prewitt Memorial Scholarship (Purdue University), Fall 2012
[11] Samuel David Williamson Scholarship (Purdue University), Fall 2012
[12] Donald E. Knebel Electrical and Computer Engineering Scholarship (Purdue University), Fall 2012
[13] Purdue University Dean's List, Fall 2012–Spring 2013
[14] Purdue University Semester Honors, Spring 2011–Spring 2013
[15] Intel International Science and Engineering Fair Second Place, Minor Planet named after me by MIT Lincoln Laboratory LINEAR (URL: https://ssd.jpl.nasa.gov/sbdb.cgi#top — search "Carolinejune")

Publications

Refereed*

[1] Naorin Hossain, **Caroline Trippel**, and Margaret Martonosi. "TransForm: Formally Specifying Transistency Models and Synthesizing Enhanced Litmus Tests". In Proceedings of the 47th ACM/IEEE International Symposium on Computer Architecture (ISCA), Valencia, Spain. May-June 2020. Acceptance rate: 77/421=18.3%.

Presents an axiomatic vocabulary for formally specifying transistency models and a tool for synthesizing litmus tests enhanced with transistency features, when supplied with a formal transistency model specification.

*Acceptance rates noted where available

[2] **Caroline Trippel**, Daniel Lustig, and Margaret Martonosi. "Security Verification through Automatic Hardware-Aware Exploit Synthesis: The CheckMate Approach". IEEE Micro, 39 (3), May-June 2018. Issue: Top Picks from the Computer Architecture Conferences of 2018.

Makes the important observation that MCM analysis is in many ways similar to hardware security analysis, paving the way for fresh approachs to security specification and verification in the hardware-software stack

[3] Hongce Zhang, **Caroline Trippel**, Yatin A. Manerkar, Aarti Gupta, Margaret Martonosi, and Sharad Malik. "Integrating Memory Consistency Models with Instruction-Level Abstractions for Heterogeneous System-on-Chip Verification". In Proceedings of the 2018 Conference on Formal Methods in Computer Aided Design (FMCAD), Austin, Texas. October-November 2018.

New approach for integrating axiomatic memory consistency model specifications with operational specifications of heterogeneous SoC components to reason about overall system behavior

[4] **Caroline Trippel**, Daniel Lustig, and Margaret Martonosi. "CheckMate: Automated Synthesis of Hardware Exploits and Security Litmus Tests". In Proceedings of the 51st IEEE/ACM International Symposium on Microarchitecture (MICRO), Fukuoka, Japan. October 2018. Acceptance rate: 74/351=21.0%.

Approach and automated tool for evaluating microarchitectural susceptibility to exploit classes and synthesizing proof-of-concept exploit code; out-of-the-box CheckMate synthesized Meltdown, Spectre and new attacks

[5] **Caroline Trippel**, Yatin A. Manerkar, Daniel Lustig, Michael Pellauer, and Margaret Martonosi. "Full-Stack Memory Consistency Model Verification with TriCheck". IEEE Micro, 38 (3), May-June 2018. Issue: Top Picks from the Computer Architecture Conferences of 2017.

First full-stack language/ISA/microarchitecture memory consistency model verification tool, which uncovered concrete and fundamental problems in the C11/C++11 and RISC-V memory models

[6] **Caroline Trippel**, Yatin A. Manerkar, Daniel Lustig, Michael Pellauer, and Margaret Martonosi. "TriCheck: Memory Model Verification at the Trisection of Software, Hardware, and ISA". In Proceedings of the 22nd ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Xi'an, China. April 2017. Acceptance rate: 56/321=17.4%.

Established the value of full-stack memory consistency model verification and highlighted issues with the RISC-V memory model that were subsequently fixed through the RISC-V Memory Model Working Group

[7] Daniel Lustig, **Caroline Trippel**, Michael Pellauer, and Margaret Martonosi. "ArMOR: Defending Against Memory Consistency Model Mismatches in Heterogeneous Architectures". In Proceedings of the 42nd ACM/IEEE International Symposium on Computer Architecture (ISCA), Portland, Oregon. June 2015. Acceptance rate: 58/305=19.0%.

Framework that supports specifying, algorithmically reasoning about, and automatically translating between heterogenous MCMs

Non-refereed

[1] **Caroline Trippel**, Daniel Lustig, and Margaret Martonosi. "MeltdownPrime and SpectrePrime: Automatically-Synthesized Attacks Exploiting Invalidation-Based Coherence Protocols". ARXIV, abs/1802.03802, 2018.

Overview of MeltdownPrime and SpectrePrime exploits that were automatically synthesized by CheckMate and which were demonstrated to leak private data on Intel hardware

[2] Yatin A. Manerkar, **Caroline Trippel**, Daniel Lustig, Michael Pellauer, and Margaret Martonosi. "Counterexamples and Proof Loophole for the C/C++ to POWER and ARMv7 Trailing-Sync Compiler Mappings". ARXIV, abs/1611.01507, 2016.

Identification of an error in a compiler mapping proof from C11 to the IBM Power and ARMv7 ISAs; the error is what enabled TriCheck to identify concrete compiler mapping counterexamples

Dissertations

[1] **Caroline Trippel**. "Concurrency and Security Verification in Heterogeneous Parallel Systems." PhD Dissertation. Princeton University. November, 2019.

Tutorials

[1] Yatin A. Manerkar, **Caroline Trippel**, and Margaret Martonosi. "Demystifying Memory Models Across The Computing Stack." Tutorial. 46th ACM/IEEE International Symposium on Computer Architecture (ISCA), Phoenix, Arizona. June 2019.

[2] **Caroline Trippel** and Yatin A. Manerkar. "Why Memory Consistency Models Matter... And tools for analyzing and verifying them." Tutorial. 2018 Uppsala Programming for Multicore Architectures Research Center (UPMARC) Multicore Computing Summer School, Uppsala, Sweden. June 2018.

[3] Yatin A. Manerkar, **Caroline Trippel**, and Margaret Martonosi. "Why Memory Consistency Models Matter... And tools for analyzing and verifying them." Tutorial. 44th ACM/IEEE International Symposium on Computer Architecture (ISCA), Toronto, Ontario, Canada. June 2017.

Grants

[1] **Caroline Trippel**. FMitF: Track II: Scaling Formal Hardware Security Verification with CheckMate from Research to Practice, 2020. Funded by NSF for $100,000.00/year for 1 year.

[2] **Caroline Trippel** and Margaret Martonosi. Security Exploit Analysis and Synthesis: Formal Specifications and Model Finding Techniques, 2018. Funded by Intel for $100,000.00/year for 3 years. Intel SCAP Program.

Press

[1]	Communications of the ACM	December 2018. How to Live in a Post-Meltdown and -Spectre World. Retrieved from https://cacm.acm.org/magazines/2018/12/232898-how-to-live-in-a-post-meltdown-and-spectre-world
[2]	TechSpot	February 2018. Researchers discover two new Spectre and Meltdown variants. Retrieved from https://www.techspot.com/news/73305-researchers-discover-two-new-spectre-meltdown-variants.html
[3]	Digital Trends	February 2018. New 'Prime' Meltdown, Spectre exploits outlined by Nvidia, Princeton University. Retrieved from https://www.digitaltrends.com/computing/princeton-nvidia-prime-meltdown-spectre/.
[4]	Gizmodo	February 2018. Researchers Find New Ways to Exploit Meltdown and Spectre Vulnerabilities in Modern CPUs. Retrieved from https://gizmodo.com/researchers-find-new-ways-to-exploit-meltdown-and-spect-1823020029.
[5]	Hacker News	February 2018. MeltdownPrime, SpectrePrime: Exploiting Invalidation-Based Coherence Protocol. Retrieved from https://news.ycombinator.com/item?id=16430215.
[6]	Tech Xplore	February 2018. MeltdownPrime and SpectrePrime: Researchers nail exploits. Retrieved from https://techxplore.com/news/2018-02-meltdownprime-spectreprime-exploits.html.
[7]	Engadget	February 2018. Researchers discover new ways to abuse Meltdown and Spectre flaws. Retrieved from https://www.engadget.com/2018/02/15/meltdownprime-spectreprime-research/.
[8]	Princeton University	April 2017. Tool for checking complex computer architectures reveals flaws in emerging design. Retrieved from http://www.princeton.edu/main/news/archive/S49/20/10O81/index.xml?section=topstories
[9]	RISC-V Organization	April 2017. The RISC-V Memory Consistency Model. Retrieved from https://riscv.org/2017/04/risc-v-memory-consistency-model/.
[10]	Phys.org	News and Articles on Science and Technology. (2017). Tool checks computer architectures, reveals flaws in emerging design. Retrieved from https://phys.org/news/2017-04-tool-architectures-reveals-flaws-emerging.html.
[11]	Semiconductor Engineering	April 2017. System Bits: April 18 – RISC-V errors; spin-wave logic gates; deep learning is old. Retrieved from http://semiengineering.com/system-bits-april-18/.
[12]	Electronics Weekly	April 2017. RISC-V bugs found by Princeton. Retrieved from https://www.electronicsweekly.com/open-source-engineering/risc-v-bugs-found-princeton-2017-04/.

[13] Electronic Design April 2017. Memory Ordering Flaw Found in Rare Version of RISC-V Hardware. Retrieved from http://www.electronicdesign.com/embedded/memory-ordering-flaw-found-rare-version-risc-v-hardware.

[14] Design and Reuse April 2017. RISC-V: When a bug is really a feature. Retrieved from https://www.design-reuse.com/industryexpertblogs/41877/risc-v-when-a-bug-really-is-a-feature.html.

Professional Experience

2020–	**Assistant Professor**, Department of Computer Science and Department of Electical Engineering	**Stanford University**, Stanford, CA
2019–2020	**Research Scientist**, FAIR SysML Manager: Hsien-Hsin Sean Lee	**Facebook**, Boston, MA
2013–2019	**PhD Candidate**, Computer Science Department Advisor: Margaret Martonosi	**Princeton University**, Princeton, NJ
2016	**Graduate Research Intern**, Architecture Research Group Supervisor: Steve Keckler Mentor: Michael Pellauer	**NVIDIA**, Westford, MA
2015	**Graduate Research Intern**, Architecture Research Group Supervisor: Steve Keckler Mentor: Michael Pellauer	**NVIDIA**, Westford, MA
2013	**Verification Graduate Intern**, P-Series Verification Group Supervisor: David A. King	**IBM**, Austin, TX
2012	**Undergraduate Technical Intern**, FEV/DFT Group Supervisor: Sulakshana Nath	**Intel**, Hillsboro, OR
2011	**Continuous Improvement Intern** Supervisor: Chris Young	**BlueCross BlueShield**, Providence, RI

Teaching and Advising

2020	**Research Advising** Yao Hsiao, Master's Thesis	Stanford University
2020	**Research Advising** Saranyu Chattopadhyay, Master's Thesis	Stanford University
2018	**Research Advising** Naorin Hossain, PhD Thesis	Princeton University
2018	**Research Advising** Vivian Mo, Bachelor's Thesis	Princeton University
2016	**Teaching Assistant** Computer Organization and Design (ELE/COS 375)	Princeton University
2014	**Teaching Assistant** Computer Organization and Design (ELE/COS 375)	Princeton University
2012	**Teaching Assistant** Introduction to Digital Systems Design (ECE 270)	Purdue University
2011	**Teaching Assistant** Introduction to Digital Systems Design (ECE 270)	Purdue University

Invited Seminar Talks

Spring 2019: "Concurrency and Security Verification in Heterogeneous Parallel Systems"

[1] University of California, Berkeley
[2] Carnegie Mellon University
[3] Cornell University
[4] Cornell Tech
[5] Duke University
[6] Northeastern University
[7] Tufts University
[8] University of California, Santa Barbara
[9] University of Illinois Urbana-Champaign
[9] Columbia University
[10] Stanford University
[11] University of Washington
[12] Yale University
[13] University of Pennsylvania
[14] Brown University
[16] University of Wisconsin-Madison
[17] University of Texas at Austin
[18] Microsoft

Other Invited Seminar Talks

[1] "Synthesizing Security Exploits". 2019 State-of-the-Art in Program Synthesis Workshop, San Francisco, CA. September 2019. **Host: Saurabh Srvastava and Rastislav Bodik**

[2] "Concurrency and Security Verification in Heterogeneous Parallel Systems". 2018 IBM Workshop on Architectures for Secure, Cognitive, and Datacenter Computing, Yorktown, NY. November 2018. **Host: Michael Healy and Arvind Kumar**

[3] "Concurrency and Security Verification in Heterogeneous Parallel Systems". MIT CSAIL, Cambridge, MA. November, 2018. **Host: Saman Amarasinghe**

[4] "Automated Exploit Program Generation for Hardware Security Verification". Apple Security Engineering and Architecture Group, Cupertino, CA. March 2018. **Host: Ivan Krstić**

[5] "Relational Model Finding for Hardware-Aware Program Synthesis and Security Verification". Brown University Computer Science Department, Providence, RI. March 2018. **Host: Iris Bahar and Maurice Herlihy**

[6] "Why Memory Consistency Models Matter in a World of Parallelism and Shared Memory". Rutgers University DIMACS Research Experiences for Undergraduates, New Brunswick, NJ. July 2017. **Host: Lazaros Gallos**

[7] "TriCheck: Memory Model Verification at the Trisection of Software, Hardware, and ISA", SRI International, Menlo Park, CA. November 2016. **Host: Prashanth Mundkur**

[8] "Exploring the Interplay of Software, Hardware, and ISA in Memory Model Design", Duke University Computer Science/Electrical Computer Engineering, Durham, NC. September 2016. **Host: Benjamin Lee**

Other Selected Talks and Presentations

[1] Poster and Talk "CheckMate: Automated Synthesis of Hardware Exploits and Security Litmus Tests". 51st International Symposium on Microarchitecture (MICRO), Fukuoka, Japan. October 2018.

[2] Poster "Concurrency and Security Verification in Heterogeneous Parallel Systems". Heidelberg Laureate Forum, Heidelberg, Germany. September 2018.

[3] Talk "Concurrency and Security Verification in Heterogeneous Parallel Systems". Princeton University Preliminary Final Public Oral, Princeton, NJ. September 2018.

[4] Poster and Talk "Relational Model Finding for Hardware-Aware Program Synthesis and Security Verification". NVIDIA GPU Technology Conference (GTC), San Jose, CA. March 2018.

[5] Poster and Talk "Relational Model Finding for Microarchitectural Security Exploits and More". SRC TECHCON 2017, Austin, Texas. September 2017.

[6] Poster and Talk "TriCheck: Memory Model Verification at the Trisection of Software, Hardware, and ISA". 22nd International Conference on Architectural Support for Programming languages and Operating Systems (ASPLOS), Xian, China. April 2017.

[7] Poster and Talk "Memory Consistency Model Aware ISA Design and Specification". SRC TECHCON 2016, Austin, Texas. September 2016.

[8] Poster "Design Space Exploration Techniques for Memory Consistency Model Aware ISA Specification". CRA-W Grad Cohort, San Diego, California. April 2016.

[9] Poster "ArMOR: Defending Against Memory Consistency Model Mismatches in Heterogeneous Architectures". 2nd Career Workshop for Women and Minorities in Computer Architecture, Honolulu, Hawaii. December 2015.

[10] Talk "Accurate and Precise Translation and Compilation in the Face of Varying Memory Consistency Models". Princeton University General Examination, Princeton, NJ. May 2015.

Selected Technical Reviewing

[1] 2021 International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '21) PC Member
[2] 2020 International Symposium on Computer Architecture Industrial Track (ISCA '20) PC Member
[3] 2020 International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20) PC Member
[4] 2019 International Symposium on Microarchitecture (MICRO '19) PC Member
[5] External expert reviewer for the 2018 International Symposium on Memory Management
[6] Reviewer for a Morgan & Claypool Synthesis Lecture on Computer Architecture, titled *Principles of Secure Processor Architecture Design*, authored by Jakub Szefer, published October 2018.

Professional Memberships

[1] RISC-V Memory Model Working Group
[2] ACM Member, ACM SIGARCH
[3] IEEE Member
[4] SWE Student Member, Society of Women Engineers (*past*)
[5] PWICS Member, Princeton Women in Computer Science (*past*)

Relevant Technical Coursework

Computer Architecture, Parallel Computation, Programming Languages, Advanced Computer Networks, Computer Design and Prototyping, Digital Systems Senior Design, Introduction to Compilers and Translation Engineering, Microprocessor Systems and Interfacing, ASIC Design

Environments, Languages, and Design Tools Experience

UNIX, LINUX, CUDA, MPLAB, MATLAB, Alloy, Perl, Python, Korn Shell, C, Assembly, Abel, JavaScript, Visual Basic, HTML, C#, SQL, VHDL, Verilog, C++, Java, Alloy, PSPICE, ModelSim, CodeWarrior, Pin, DynamoRIO, gem5