SooHyuk Cho

soohyuk-cho | soohyuk-cho | soohyuk.cho@berkeley.edu | +1 (484) 761-4166

Education

Princeton University *Ph.D. in Electrical and Computer Engineering* Aug 2025 - In-Progress
Research Interests: Formal Verification Methodology, Architectural Simulation Framework, Design Automation
Relevant Coursework: (ECE 516) Automated Reasoning about Software, (ECE 574) Security and Privacy in Computing and Communications, (COS 518) Advanced Computer Systems, (ECE 580) Domain-Specific Computer Systems Architecture (all in-progress)

UC Berkeley *B.A. in Computer Science & Data Science, EECS Honors Program* (GPA: 3.989/4.000) Aug 2020 - May 2025
Relevant Coursework (CS): (CS 152) Computer Architecture and Engineering, (CS 162) Operating System and System Programming, (CS 164) Programming Languages and Compilers, (CS 168) Internet:
Architecture and Protocols, (CS 186) Database Systems, (CS 189) Machine Learning, (CS 170) Intro to CS Theory, (CS 285, Grad) Deep RL, (CS 61B) Data Structures, (CS 61C) Machine Structures
Relevant Coursework (EE): (EE 194) 16nm SoC for IoT Tapeout, (EECS 151) Digital Design and Integrated Circuits, (EECS 151LB) FPGA Lab, (EE 105) Microelectronic Devices and Circuits, (EECS 126) Probability and Random Process, (EECS 127) Optimization Models, (EE 120) Signals and Systems, (EE 221A, Grad) Linear System Theory, (EECS C106A) Intro to Robotics, (EECS C106B) Robotic Manipulation
Involvement: CS Scholars, IEEE Student Branch, Computer Science Mentors, Upsilon Pi Epsilon (UPE) Nu Chapter

Skills

Languages C, C++, Verilog, ARM ISA, Bash, Assembly (RISC-V, x86), Java, Python, SQL, Lisp
Miscellaneous Git, Vim, Unix, NumPy, Matplob, Pygame, Scipy, Pytorch, LaTeX

Research Experience

UC Berkeley SLICE Lab *Architectural Checkpoint on FireSim* Jan 2024 - May 2025
- Developed and integrated a sparse memory checkpointing mechanism in FireSim, enabling the reloading of only non-zero memory regions, which reduced memory dump sizes by up to **48%** and decreased simulation loading times by up to **40%**.
- Created a C++ tool to analyze ELF executable files, accurately identifying non-zero memory regions to support efficient checkpointing.
- Designed an automated script to extract active data sections from ELF files and generate custom linker scripts, ensuring correct memory section placement during simulation and optimizing simulation performance.
- Improved initial program loading times by implementing parallel processing techniques and optimizing memory transfer processes between host and FPGA's DRAM.
- Performed rigorous testing on bare-metal and Linux workload binaries using FireSim, demonstrating significant improvements in simulation efficiency and highlighting the importance of scalable, accurate simulation tools for complex computing systems.
- Automated thread-level switching process between the Debugger Module Interface (DMI) and Tethered Serial Interface (TSI) to streamline memory initialization sequence bypass and core module switching, enhancing the efficiency of checkpoint loading and reliability of simulation debugging.

Ping Pong Robotics Group *Ping Pong Robotics (PPR) Project* Aug 2023 - Mar 2024
- Developed a custom control system for a KUKA manipulator to intercept high-speed ping pong balls using real-time computer vision and predictive modeling.
- Designed and integrated a four-camera vision system with OpenCV to detect and triangulate the ball's position, applying curve fitting and filtering to enhance trajectory prediction accuracy.
- Implemented custom Jacobian and Inverse Kinematics controllers to improve robot responsiveness and resolve self-collision issues, enabling precise visual servoing based on dynamic inputs.
- Successfully demonstrated the robot's ability to track and follow high-speed objects in real-time, overcoming challenges in hardware-software integration and system latency.

Berkeley Artificial Intelligence Research Lab *Prompt Inject Project & AdvPrompt Project* May 2023 - Dec 2023
- Designed and executed adversarial attack strategies to test LLM robustness across classification, exact match, and perplexity tasks, achieving high success and diversity rates.
- Developed optimization techniques like sparsemax projection, proxy model tuning, and batch-based querying, reducing query time and enhancing attack efficiency.
- Collaborated on refining the attack pipeline, implementing high-dimensional vocabulary constraints, and integrating gradient-based and exhaustive search methods to improve LLM safety testing accuracy and reliability.

UC Berkeley SLICE Lab *Gemmini Project* Jan 2023 - Dec 2023
- Developed and integrated multithreading capabilities into the Gemmini systolic array generator, enabling parallel execution across multiple threads for convolutional operations.
- Increased compute throughput by partitioning matrix computations over rows and columns, optimizing resource allocation for efficient processing.
- Designed and implemented thread management and synchronization mechanisms to ensure correct and efficient multithreaded operations, eliminating race conditions.
- Investigated and applied data and pipeline parallelism strategies to further optimize the performance of deep learning accelerators.

- Leveraged ONNXRuntime integration with Gemmini to execute BERT models end-to-end, demonstrating the effectiveness of multithreading improvements.
- Performed rigorous testing in multithreaded environments using functional simulators like QEMU and Spike.

Prof. Edward Lee's Group *Lingua Franca* May 2021 - Sep 2021

Converted and Implemented each top ROS node of autoware's April 2021 Autonomous Valet Parking (AVP) release into reactor components of Lingua Franca (LF), a language intended to make it easy to write highly performant concurrent and potentially distributed programs that render deterministic behaviors under well-defined assumptions by exploring the usage of reactive components.

Industry Experience

Apple **CPU Design Intern** *ARM Architectural Specification Language Compiler Development* May 2025 - Aug 2025

- Implemented comprehensive Scalable Vector Extension (SVE) support in ASL compiler framework, enabling C++ code generation for formal verification of 760+ SVE instructions across Apple CPU architectures.
- Designed and implemented register modeling system with full support for SVE vector and predicate registers, improving architectural accuracy of formal models.
- Re-architected memory subsystem to enable physical address-based data flow verification for load/store operations, enhancing memory consistency checking capabilities.
- Improved compiler performance through optimized case splitting strategies and extended loop range analysis for generic iteration patterns.
- Validated compiler correctness against ARM ISA reference simulator, achieving SVE 100% instruction coverage.

Synopsys **ML for Hardware Internship** *μArchitectural Block Classifications* Jul 2024 - Dec 2024

- Created a comprehensive taxonomy for μarhchitectural components (e.g., FIFOs, arbiters, counters) to enhance hardware design automation by enabling better classification and reuse.
- Leveraged LLMs with specialized prompts to classify RTL code into specific microarchitectural components, achieving a 78% classification accuracy on a test set of $\sim$ 1,000 RTL code samples.
- Implemented a framework to assess model precision and recall for μArch type and parameter identification, optimizing classification accuracy through active learning techniques.
- Collaborated on enhancing the dataset by generating advanced metadata, performing error-checking, and conducting semantic validation, thereby supporting accurate component classification and ongoing taxonomy evolution.

Synopsys **ML for Hardware Internship** *System Verilog Code Generation* Feb 2024 - Jul 2024

- Developed pipeline using LLMs to generate and fine-tune SystemVerilog (SV) code, streamlining the synthesis of hardware description languages.
- Processed over 7,000 code chunks and curated 9,000+ high-quality SV samples with a 21.4% pass rate, implementing filtering mechanisms to eliminate low-quality code and enhance dataset reliability.
- Incorporated PySlang for syntax validation and utilized LLM scoring to analyze code complexity, ensuring the generation of syntactically correct and efficiently complex SV code.
- Adapted pre-trained models like SantaCoder and StarCoder on SV instruction-code pairs, improving code correctness and compilation success rates through targeted fine-tuning.
- Established an evaluation framework to measure performance metrics such as pass@K, facilitating the assessment and optimization of code generation models.

Projects

3-stage Pipelined RISC-V CPU Design on FPGA *Verilog* Report Paper Feb 2024 - June 2024

- Developed and optimized a 3-stage pipelined RISC-V CPU, achieving a maximum clock frequency of 110MHz.
- Implemented a comprehensive memory architecture with separate instruction and data memory, BIOS, and MMIO.
- Enhanced performance through critical path analysis and optimizations in forwarding, pc_sel, and ALU bypassing.
- Demonstrated proficiency in Verilog HDL, FPGA design, and synthesis tool optimization.

ChocoPy Compiler *Java, Python, RISC-V* Jan 2024 - May 2024

- Developed a statically typed subset of Python compiler with JFlex and CUP's parser generator combined with semantic analysis and code generation via RISC-V architecture.
- Implemented a multi-core test harness that handles file input and hanging tests for a robust development cycle.
- Optimized compiler performance by implementing common subexpression elimination, efficient temporary register allocations, and unboxing primitive type objects.

State-Diffuser *PyTorch/Python* Report Paper Oct 2023 - Dec 2023

- Analyzed the Synthetic Experience Replay (SynthER) framework and proposed a variant called State-Diffuser that combines both SynthER and Model-based Policy Optimization (MBPO).
- Evaluated State-Diffuser framework on three Mujoco environments and showed that the simplified framework can achieve similar performance as SynthER while reducing computational complexity.

Pintos Operating System *C, x86* Sep 2023 - Dec 2023

- Enhanced functionality to the PintOS codebase, a compact x86-based educational OS.
- Accelerated reads and writes to/from disk and added functionality of extensible files by implementing the Berkeley Fast File System and in-memory caching.

- Improved efficiency of concurrent multithreaded operations by implementing a strict priority scheduler and relevant pthread syscalls.

Adversarial Machine Learning *Pytorch, Scipy, Numpy* May 2023

- Implemented and analyzed the algorithmic characteristics of two main adversarial attack models: L-BFGS and Fast Gradient Sign Method.
- Compared the effectiveness of each algorithm in terms of the tradeoff between the amplitude of the perturbation and adversarial effects to cause misclassification.
- Explored different initialization schemes for the L-BFGS attack and compared adversarial examples generated by each in terms of how similar the perturbed image is to the original image.

Custom Branch Predictor *C++* Feb 2023 - March 2023

- Implemented four different types of branch predictors: tournament, share, gshare & bimodal combined, and perceptron. Improved the CPI of the various benchmarks (dhrystone, qsort, vvadd, etc) by 20∼50%.
- Analyzed the difference in performances among four branch predictors in terms of resource competition, table size, and parallelism efficiency.

RISC-V NN *RISC-V* Sep 2022 - Oct 2022

Built a RISC-V artificial neural network that classifies handwritten digits based on pre-trained data by implementing linear algebra operations in RISC-V. Programmed file operations that read and write matrices from and to files.

Teaching Experience

CS 189 (Intro to Machine Learning) TA/UCS2 *SP25* Jan 2025 - May 2025

Worked 20 hours per week as a TA for CS 189. Will be teaching discussion sections, holding office hours and review sessions, creating and grading quizzes and exams, managing course logistics including but not limited to student-facing communication (EdForum), DSP (Disabled Student Program) extensions, exam logistics, etc. for 650+ students/semester, providing pedagogical support for junior staff members.

EE 120 (Signals and Systems) Admin TA/UCS2 *FA24, SP24* Jan 2024 - Dec 2024

Worked 8/10 hours per week as a TA for EE 120. Holding office hours, creating problems for homework, discussion, and exams, teaching discussion sections, and managing the course website for 120+ students/semester. Teaching properties of LTI system, Fourier transform, z-transform, Laplace Transform, Sampling, etc.

EECS 16A (Designing Information Devices and Systems I) Head TA/UCS2 *SU24* Jun 2024 - Aug 2024

Worked 20 hours per week as a Head/Admin Teaching Assistant for EECS 16A. Taught 1-hour discussions 4 times per week, wrote and debugged exam problems, answered conceptual questions on the online Ed Forum, and managed course and exam logistics.

EECS 16A (Designing Information Devices and Systems I) Tutor/UCS 1 *SU23, FA23* Jun 2023 - Dec 2023
EECS 16A (Designing Information Devices and Systems I) Reader *FA22, SP23* Aug 2022 - May 2023

Worked 6 hours per week for Fall 2022/Spring 2023, 12 hours per week for Summer 2023, and 8 hours per week for Fall 2023 as a tutor for EECS 16A. Grading weekly homework, giving feedback to students, debugging exams, and answering conceptual questions on the online Ed Forum. Taught about vector spaces, eigenvalues, basic circuit elements, charge sharing algorithm, least-squares, trilateration, etc.

Academic Intern @ UC Berkeley EECS Jan 2021 - Aug 2021

Summer 2021 CS 70 (Discrete Mathematics and Probability Theory) Academic Intern @ UC Berkeley
Spring 2021 CS 61A (The Structure and Interpretation of Computer Programs) Academic Intern @ UC Berkeley

Awards & Achievements

Korean Presidential Science Scholarship July 2020

Selected as one of 20 scholarship recipients for outstanding academic excellence and potential to contribute to the areas of science, engineering, and technology. $50,000/yr scholarships awarded by the Government of South Korea for 4 years under the name of the President of the Republic of Korea (Total $200,000)

Honors to Date Fall 2023, Spring 2023, Fall 2022, Spring 2021, Fall 2020
Dean's List - College of Letters and Science Fall 2023, Spring 2023, Spring 2021

Activities & Involvements

EECS 16A Course Coordinator @ Compute Science Mentors Dec 2023 - Dec 2024

Leading a team of 40+ Senior and Junior Mentors in providing about 20+ small group tutoring sections to students as a supplement to EECS 16A: Designing Information Devices and Systems I course. Communicating with the executive team about initiatives and incorporating feedback. Manage logistical responsibilities such as recruiting, event planning, team meetings, and student support with the co-coordinator.

Junior & Senior Mentor (CS 70 & EECS 16A) @ Compute Science Mentors Aug 2022 - May 2025

Teaching a 4-5 person section for CS 70 (Discrete Mathematics & Probability Theory) and EECS 16A (Designing Information Devices and Systems I) every week to accelerate student learning. Explaining the key concepts of the course through mini-lectures and promote group discussion. Collaborating with other mentors to generate and lead review sessions for students before exams.

Socials Chair & Genral Member @ Upsilon Pi Epsilon (UPE), CS Honor Society Jan 2023 - Jun 2023

CS Scholar @ CS Scholars Program Aug 2020 - Present

The CS Scholars Program at UC Berkeley is a community in which students can learn and grow together. The program is intended to serve those from under-represented communities who have had little or no exposure to Computer Science.

Industrial Relations Officer @ UC Berkeley IEEE Student Branch Aug 2020 - Aug 2021

Assisted with the development of UC Berkeley's annual global startup fair. Sourced 25+ startups for invitation to our career fair while managing communications between these startups and IEEE executives.