Tao Wang

ttwag@berkeley.edu | linkedin.com/in/ttwag | github.com/ttwag

Education

University of California, Berkeley — Berkeley, CA
Master of Engineering, Electrical Engineering and Computer Science — *2026 – 2027*

University of California, Davis — Davis, CA
Bachelor of Science, Electrical Engineering (ENG. GPA: 3.930) — *2021 – 2025*

Experience

Software Development Engineer Intern — Sep. 2025 – Dec. 2025
Amazon.com, Inc. — *Seattle, WA*

- Authored a design doc for the CAPTCHA monitoring workflow, incorporating feedback across multiple review rounds with a team of 7; completed 30+ code reviews with a mentor
- Designed and implemented a **Step Function** workflow with a **Distributed Map** state to fan out parallel CAPTCHA endpoint checks and aggregate results, using AWS CDK for infrastructure as code
- Built a **Lambda** function using Selenium to capture UI screenshots at each endpoint, then invoked **AWS Bedrock** to classify screenshots as malformed or regular UI
- Achieved **80% accuracy** across 860 test runs on synthetic test cases spanning both regular and malformed UI elements

Advanced Design Verification Engineer Intern — Jun. 2025 – Sep. 2025
Astera Labs — *San Jose, CA*

- Partnered with a formal verification engineer to scope dashboard requirements, defining which metrics (e.g., assert/cover assertion density) belonged in the tool vs. his verification scripts
- Built a scalable full-stack **formal verification dashboard** using Python, SQLAlchemy ORM, and Dash with a layered parsing, data, and presentation architecture to visualize VC Formal and Cadence Jasper logs
- Designed a table-driven system letting engineers define custom schemas and field-to-log mappings for new EDA tools, replacing hardcoded logic with a registry pattern
- Refactored **4,000 lines** of monolithic scripts into a **1,600-line** modular system

Software Engineer Intern — Jun. 2024 – Sep. 2024
Cadence Design Systems — *Pittsburgh, PA*

- Investigated an O(nˆ2) performance bottleneck in PCB routing software using Oracle Developer Studio's Performance Analyzer
- Achieved an **1800x runtime speedup** by replacing quadratic loops with hash sets to reduce time complexity to O(n)
- Converted over **50%** of the Cadence PCB routing software codebase from C to C++ across 3,300 C files

Projects

Compiler and RISC-V CPU, Built from Scratch | *Rust, SystemVerilog*

Writeup: ttwag.github.io/posts/compiler-cpu-writeup.html

- Designed a toy programming language and built its compiler in Rust — implemented the scanner, recursive-descent parser, and code generator targeting RISC-V (RV32I) assembly under Linux ABI conventions
- Derived a register allocation scheme limiting expression evaluation to 3 temporary registers, spilling to the stack only for group expressions and function calls, avoiding naive spill-everything overhead
- Validated the compiler end-to-end with **46 unit tests and 63 integration tests** run against QEMU and iverilog simulation, and then simulated the generated program on a self-implemented RISCV32I CPU in SystemVerilog
- Debugged a cross-system failure (compiler, CPU, testbench) by examining disassembly with simulation waveforms, tracing the root cause to stack pointer initialization and undersized data memory

Technical Skills

Languages & Tools: Rust, C/C++, Python, Git
Relevant Coursework: Machine Learning (CS 289A, in progress), Probability and Stochastic Processes, Computer Architecture, Digital System FPGA Lab, Computer Networks, Embedded Systems