Ansh Shah

(408) 207-8992 | ansh_shah@berkeley.edu | linkedin.com/in/ansh-shah-eecs/ | github.com/The-Ansh-Shah | Berkeley, CA

Education

University of California Berkeley May 2027 (Expected)
Bachelor of Science in Electrical Engineering and Computer Science ***GPA: 3.953/4.0***

- **Relevant Coursework**: Digital Integrated Circuits (ASIC), Computer Architecture, Microelectronic Devices and Circuits, Embedded and Cyber-Physical Systems, Signals and Systems, Operating Systems, Formal Methods
- **Organizations:** HKN (Top 25% of EECS Undergraduate Students at Berkeley)
  - Organizing Info Sessions/Networking events, POC for EECS Career/Internship Fairs, Industrial Relations Head, Weekly Tutoring and Test Prep, Updating course guides/materials for lower division EECS courses
  - Hosting review of concepts in Circuits, Devices, and Computer Architecture for courses with 400+ students

Technical Skills

HDLs and Hardware: Verilog, RISC–V, Chisel, CPU microarchitecture (pipelining, microcoding), GPU architecture, CMOS Design, Testbenches; **Synopsys VCS** (simulation/waveforms), **Cadence Innovus** (PnR), **LTSpice**
Programming and Scripting: C, Python (Matplotlib, Numpy, Jupyter), Java, C++, Rust
Lab Tools: Oscilloscopes, Parameter Analyzers, Waveform Generators, DMM; MCU platforms: RP2040, STM32, ESP32
Developer Tools: Linux/UNIX Terminal, Git, VS Code, Vim, Visual Studio, Valgrind, GDB, PyCharm, IntelliJ

Experience

Undergraduate Researcher Apr. 2025 – Present
SpeciaLIzed Computing Ecosystems (SLICE) Lab *UC Berkeley EECS*

- Helping create a processor tracing framework for minimal-overhead performance profiling; supports delta encoding
- Software simulation of lossless compression techniques (Huffman, Delta-of-Delta, RLE) to further compress trace
- Programmed decode and compression simulations in Rust; Baremetal IDE used to analyze compression results

Undergraduate Course Staff Aug. 2025 – Dec. 2025
CS 61C: Computer Architecture and Machine Structures *EECS/DS ASE*

- Supported **600+ students** in the largest computer architecture and digital logic course at Berkeley
- Held Office Hours, Discussion Mini-Lectures, C/RISC-V/Digital Logic Debugging (CGDB, Valgrind)
- Created exam walk-throughs to explain challenging concepts; supported Academic Interns in video creation

Projects

RV32I CPU and Direct-Mapped L1 Cache | *Verilog, Sky130 PDK, Cadence Innovus, Synopsys VCS* Fall 2025

- Designed a **3-stage pipelined RV32I datapath** in Verilog with Cached I/DMEM, **achieving 62.4 MHz**
- Reduced MEM-access latency by **integrating a 4KiB L1-Cache** using smaller 1KiB SRAMs, reducing area
- Developed a Cache FSM to handle 512b over 4 cycles, integrating a **Dirty Flag to reduce cycle usage by 36%**
- Leveraged **SystemVerilog Assertions** to verify reset behavior, memory address masking, and x0 invariants

RP2040 Bootloader Rootkit Analysis | *C, ARM Thumb, Embedded Security, Formal Verification* Fall 2025

- Studied RP2040 boot sequence, contributing to **PoC early-boot execution redirection** via watchdog registers
- Achieved code persistance by redirecting **ROM-to-flash handoff** and modifying second-stage bootloader
- Modeled boot-flow state machine and **verified persistence properties using CBMC formal verification**
- Designed mitigation strategy to neutralize watchdog-based persistence and improve secure-boot robustness

RISC-V Neural Network Classifier | *RISC-V, Venus Simulator (VDB), Linear Algebra* Spring 2025

- Implemented low-level **math kernels in RISC-V assembly** (ReLU, argmax, stride-aware dot product, matmul)
- **Integrated kernels into digit-classification pipeline**, with matrix operations and in-place activation
- Developed file I/O routines (read/write matrices using headers, heap allocation, error handling)

Treble Booster | *LTSpice, Oscilloscope, Network Analyzer, DMM, Circuit Analysis* Spring 2025

- Designed a **band-pass filter** to operate as a treble-frequency booster stage for a guitar and amplifier
- Analytically determined capacitance and resistance specifications to achieve desired corner frequencies
- Used LTSpice to design and simulate the circuit, and used Scope and Network Analyzer to quantify amplification