Shinit Dinesh Shetty

San Jose, CA 95123 — blackhawk005 — shinit-shetty — shinitdshetty@gmail.com — +1 (919) 492-5639

Education

Master of Science in Computer Science, North Carolina State University *Aug 2023 – May 2025* GPA 3.60/4.00 *Relevant Courses:* Design and Analysis of Algorithms, Software Engineering, Neural Networks, Cloud Computing, Foundations of Data Science, Automated Learning and Data Analytics, DBMS, Artificial Intelligence.

Bachelor of Engineering in Computer Engineering, University of Mumbai *Aug 2019 – Jun 2023* CGPA 8.98/10 *Relevant Courses:* **System Programming and Compiler Construction, Digital Logic, Computer Organisation and Architecture, Operating Systems, Distributed Systems**, Data Mining, Big Data Analysis.

Technical Skills

Languages: C++ (C++11/14/17), C, SystemVerilog, Verilog, Python, Java, Swift, JavaScript
Systems & Performance: GDB, Valgrind, perf, gprof, AddressSanitizer, CMake, Make, Git, LLVM/Clang, multithreading, lock-free data structures, RAII, STL internals
Hardware / Simulation: Verilog/SystemVerilog LRM, event-driven simulation, RNM (Real Number Modeling), mixed-signal verification, RTL simulation kernels, CUDA / GPU programming model (familiarity)
AI-Powered Development: Cursor, GitHub Copilot, generative AI assistants; designed custom agentic workflows for code generation, debugging, and root-cause analysis
ML / Cloud (secondary): PyTorch, TensorFlow, Hugging Face, AWS, Docker, GCP

Work Experience

SDE II — Mixed-Signal Verification R&D *San Jose, CA*
Cadence Design Systems *Aug 2025 – Present*

- Architecting next-generation mixed-signal verification flows for **Xcelium** (Cadence's RTL simulator, direct counterpart to Synopsys VCS), focusing on event-scheduling efficiency and runtime/compile-time performance across analog–digital boundaries.
- Optimizing simulation kernel hot paths in modern C++ by profiling with perf, gprof, and Valgrind; refactoring active/inactive/NBA region handling and reducing cross-domain synchronization overhead in RNM workloads.
- **Built custom agentic AI workflows** that automate product-validation testing, operational issue triage, and R&D code debugging — including automated root-cause analysis of simulator crashes and errors — significantly reducing manual workload across QA and engineering teams.
- Designing intelligent debug utilities that detect LRM-conformance issues, waveform anomalies, and simulation drift using ML-based signal classification and pattern mining.
- Driving adoption of **Cursor and GitHub Copilot** within the team to accelerate C++ development of legacy simulation code paths; informally mentoring incoming interns on the Xcelium codebase and modern C++ best practices.
- Contributing to regression and CI infrastructure to surface performance regressions earlier in the development cycle, reducing iteration time for the broader R&D team.

C++ Software Developer Intern, R&D *San Jose, CA*
Cadence Design Systems *May 2024 – Aug 2025*

- Developed a C++ diagnostic tool for the **Xcelium compiler** that automates failure detection and memory-surge identification during RNM elaboration, cutting customer-facing triage time on mixed-signal regressions.
- Built reusable analytics utilities using modern C++ (smart pointers, RAII, move semantics, STL containers) and applied Factory / Strategy / Visitor design patterns to keep simulation components extensible.
- Enhanced Verilog optimization configuration layers in C++, achieving a **10% runtime speedup on customer RNM regression workloads** and introducing finer-grained tuning knobs for the performance team.
- Profiled RNM behavior across Verilog pipelines and re-architected configuration and analysis abstractions to reduce per-cycle overhead in the simulator's inner loop.

Academic and Professional Projects

RealNetOS — Modular Microkernel OS for Raspberry Pi 4 GitHub

- Designed a modular microkernel in **C and ARM Assembly** with process/thread management, syscall interface, signal handling, and IPC primitives — demonstrating systems-level fluency directly relevant to simulator-kernel work.
- Implemented a real-time preemptive scheduler with deadline-aware queueing and sleep/yield semantics for user-space threads.
- Built a multithreaded TCP/IP stack from scratch (Ethernet, IPv4, UDP, TCP) with concurrent RX/TX using lock-free ring buffers.
- Enabled zero-copy DMA networking via memory-mapped NIC registers and hardware descriptor rings, minimizing CPU overhead in packet transfer.
- Integrated a FAT32-based virtual file system (VFS) supporting mountable filesystems, path resolution, and vnode/file abstractions.

Verilog Discrete-Event Simulator — *Course Project, System Programming & Compiler Construction* GitHub

- Built a lightweight **Verilog parser and event-driven simulation engine in modern C++**, supporting combinational primitives, sequential blocks, and basic non-blocking assignments modeled on the SystemVerilog LRM scheduling semantics.
- Implemented an event wheel with active / inactive / NBA region handling — the same simulation kernel concepts that underpin production RTL simulators like VCS and Xcelium.
- Added a peephole-style event-coalescing optimizer to eliminate redundant signal-propagation events, achieving a measurable throughput improvement on benchmark testbenches.
- Used STL containers, smart pointers, and Google Test for unit coverage; profiled with `perf` to identify and fix hot-path inefficiencies in the scheduler.

Publications and Achievements

- IEEE International Conference on Power, Instrumentation, Control and Computing (PICC — 2023) — Link
- International Conference on Recent Trends in Multidisciplinary Research and Innovations (ICRMIR — 2023, Page 57) — Link