# DEVESH HITESHBHAI JANI

+1 (919)-520-5927 ◇ Email: dhjani2@ncsu.edu ◇ www.linkedin.com/in/deveshjani ◇ Raleigh, NC

## EDUCATION

**North Carolina State University** — Raleigh, NC
M.S. in Computer Engineering **GPA:** 3.3/4 — May 2026
**Relevent Coursework**: Advance Computer Architecture: Data Parallel Processors, Operating Systems Principles

**Charotar University of Science and Technology** — Anand, India
B.Tech in Electronics & Communication Engineering - **CGPA:** 9.05/10.00 — May 2024

## TECHNICAL SKILLS

| | |
|---|---|
| **Programming** | C++, C, Python, Rust, Perl, TCL, System Verilog, Verilog. |
| **Tools and Software** | CUDA, OpenCL, GPGPU-sim, Vivado, LabView, Docker, MPI, OpenMP, OpenGL, Vulkan. |
| **Communication Protocols** | SPI, I2C, UART, CAN, APB-AHB, Ethernet. |

## PROFESSIONAL EXPERIENCE

**Western Semiconductor** — Tempe, AZ
**SoC Design Intern** — May 2025 - Aug 2025

- Worked on the design and integration of a RISC-V compliant Vector Processing Unit (VPU) with a multi-lane vector register file, based on RVV 1.0 specifications.
- Led the intern team on the design of a custom SIMD-based GPGPU architecture with warp scheduling and hierarchical memory, integrated as a coprocessor within a RISC-V SoC via TLC interconnect.
- Developed a comprehensive C++ simulation testbench for a custom vector register file (VRF) module with dual-read and conflict detection logic, generating VCD waveforms for RTL verification.

**Indian Space Research Organization** — Ahmedabad, India
**Research Intern** — Dec 2023 - May 2024

- Developed and tested embedded firmware logic for a Moving Average (Boxcar) filter on an FPGA platform, utilizing Libero IDE Integrated external SRAM for data buffering and handled timing synchronization critical for real-time signal processing.
- Contributed to system bring-up and bench-level hardware validation, iteratively debugging embedded firmware and resolving system-level integration challenges across microcontroller peripherals like ADC, GPIO, SPI, and PWM.
- Executed functional validation of FIR filter on ProASIC3 FPGA by applying randomized and corner-case test vectors, and correlating results against MATLAB golden model within $\pm 1$ LSB error margin.

## ACADEMIC PROJECTS

**Microarchitecture Reverse Engineering and Performance Analysis (C)**

- Reverse engineered CPU components (cache hierarchies, branch predictors, TLB systems) using timing-based side-channel attacks across Intel server platforms, identifying 2-5x performance differences between 4KB vs 2MB pages and analyzing CPU vs GPU architectural tradeoffs through AMX accelerator performance characterization.

**Process Scheduling in Xinu OS (C)**

- Implemented two starvation-avoiding CPU schedulers (exponential distribution and Linux 2.2–style epoch/quantum goodness) in the Xinu OS, extending kernel data structures and rescheduling logic.
- Modified kernel data structures: proc table (added quantum, goodness, epoch tracking fields), ready queue handling, scheduler class state, and supporting math/util routines for randomness.

**Demand-Paged Virtual Memory Subsystem (C)**

- Implemented demand paging in Xinu OS by extending system calls (xmmap, xmunmap, vgetmem, vfreemem) and managing private virtual heaps, enabling processes to allocate and map memory beyond physical limits.
- Integrated page replacement algorithms (Second-Chance, Aging) through page fault handlers, frame table management, and backing store support, reducing TLB misses and handling page faults efficiently.

**RISC-V Superscalar Pipeline Simulator (C++)**

- Developed a simulator for an out-of-order Super-scalar processor to model dynamic instruction scheduling, focusing on pipeline stages and handling structural hazards to optimize instruction throughput.
- Implemented a pipelined processor using Tomasulo algorithm that fetches and issues N instructions per cycle, with analyzing it's IPC for various super-scalar widths, scheduler and re-order buffer sizes.

**GPGPU-SIM Performance and Functional Analysis (CUDA, C++)**

- Modified the functional simulator to alter instruction semantics and analyzed benchmark performance on different GPU architectures.
- Implemented counters in the performance simulator to track branch divergence and memory access patterns for GPU execution analysis.