

SATVIK GUPTA

(412) 760-5228 | satvikg@andrew.cmu.edu | linkedin.com/in/satvik-gupta-960879325 | github.com/satgupta23

EDUCATION

Carnegie Mellon University — Pittsburgh, PA

Bachelor of Science in Electrical and Computer Engineering
GPA: 4.0; Dean's List all Semesters

Expected Graduation Date: May 2028

TECHNICAL SKILLS

Hardware & Semiconductor Systems: SystemVerilog, RTL Design, RISC-V, FPGA Design, Computer Architecture, Embedded C, Zephyr RTOS, CNFET Characterization, Wafer-Level Testing, Semiconductor Parameter Analysis

Hardware & Verification: Flex-PCB Bring-Up, Sensor Integration, Register-Level Debugging, Testbenches, Waveform Debugging, AMD Vivado, Oscilloscope, Logic Analyzer, Multimeter, Soldering, 3D Printing, Regression Testing, CAD

Programming & Software: Python, C/C++, Java, JavaScript, SQL, Assembly, CUDA, MATLAB, Linux, Git/GitHub, PyTorch, FastAPI, scikit-learn, Pandas, NVIDIA cuBLAS, Tailwind CSS, HTML

EXPERIENCES

Undergraduate Researcher – Semiconductor Machine Learning Analysis

NEXUS Research Group, Carnegie Mellon – Advised by Prof. Tathagata Srimani

Aug 2026 – Present

- Developing a PyTorch pipeline to classify functional, open, shorted, and anomalous CNFETs by extracting subthreshold slope, saturation on-current, peak Ion/Ioff ratio, and device variability from electrical sweep data.
- Analyzing wafer-level voltage- and frequency-sweep measurements acquired with Keysight B1500 SMU/MFCMU, NI PXI, and Spectrum Vision eVue V systems to evaluate device performance, reliability, and fabrication consistency.

Robotics Engineer — Carnegie Mellon Robotics Club

Oct 2025 – Present

- Build combat and performance robots for NHRL competition and an autonomous robotic orchestra using microcontrollers, actuators, solenoids, and custom PCBs; contribute across soldering, 3D printing, and laser-cut fabrication.

Teaching Assistant for 18-100 (Introduction to ECE) — Carnegie Mellon ECE

Jan 2026 – Present

- Teach circuit analysis and hands-on laboratory debugging to 150 students; diagnose wiring, component, measurement, and reasoning errors while guiding students through systematic troubleshooting and clear technical communication.

QA Engineering Intern — CARPL (Medical Imaging AI)

Jul 2026 – Aug 2026

- Test radiology AI applications on de-identified clinical cases, reviewing model outputs and result-delivery workflows.
- Reproduce software defects, document expected-versus-actual behavior, write actionable bug reports, and regression-test fixes while working directly with the engineering team to verify resolution.

Undergraduate Researcher (Computational Biology) — Bochkis Lab, University of Pittsburgh

Feb 2026 – Aug 2026

- Execute ChIP-seq and RNA-seq workflows on a Linux HPC cluster using command-line tools for alignment, filtering, duplicate removal, sample merging, and PeakSeq-based peak calling.

PROJECTS

Bio-wearable Hot-Flash Detector (Embedded C, Zephyr RTOS, BLE, nRF54L15, I²C, Devicetree)

Carnegie Mellon University – Advised by Prof. Pulkit Grover and Dr. Hongting Zhao

Aug 2026 – Present

- Developing firmware and validation infrastructure for a multimodal flexible wearable that combines infrared temperature, 9-axis motion, and electrodermal sensing to support objective, ambulatory hot-flash detection research.
- Brought up BMI270, BMM350, and MLX90632 sensors on a shared I²C bus with an nRF54L15; configured addresses, sampling rates, ranges, oversampling, and Zephyr devicetree mappings for 25 Hz acquisition.
- Designing repeatable flex-PCB benchtop tests for sensor accuracy, I²C integrity, mechanical bending, BLE packet reliability, power draw, and radio interference in the EDA analog front end.

Real-Time Multiplayer Controller Platform (TypeScript, Node.js, Express, WebSockets, Playwright)

Sep 2026

- Architected an end-to-end real-time input system spanning smartphone clients, a stateful Node.js/WebSocket server, and a reusable TypeScript game SDK; ingested high-frequency device events at up to 50 Hz and published normalized controller state at 60 Hz to 6+ browser games.
- Built the server's connection and session-management layer for 4 concurrent controllers, implementing client-slot ownership/recovery, duplicate-client takeover, protocol validation, packet rate limiting, heartbeat/ping-pong health checks, RTT-adaptive stale/disconnect thresholds, and safe state cleanup after network failure.
- Engineered a fault-tolerant client abstraction with silent-connection detection, exponential-backoff reconnection, offline-state handling, and device-independent input processing, decoupling game logic from networking and hardware-specific behavior; validated the stack with 130+ unit, integration, and Playwright end-to-end tests.

CUDA SGEMM Performance Optimization (CUDA, NVIDIA cuBLAS, Nsight Compute)

Aug 2026

- Developed and optimized six CUDA SGEMM kernels, increasing 4096×4096 matrix-multiplication throughput 67.9× from 93.8 GFLOP/s to 6.37 TFLOP/s, reaching 90% of NVIDIA cuBLAS performance.
- Applied coalesced memory access, shared-memory tiling, 1D/2D register blocking, and vectorized float4 loads; used profiler- and roofline-driven analysis to diagnose memory bottlenecks and validate optimizations.

32-bit RISC-V Processor Design in Verilog (Verilog, AMD Vivado)*May 2026 – Aug 2026*

- Designed and integrated a modular single-cycle 32-bit RISC-V processor implementing core RV32I and the complete RV32M extension through a custom datapath, control unit, register file, and memory subsystem.
- Built an automated integration testbench and 92-checkpoint self-checking instruction suite covering ALU, multiply/divide, load/store, branch, jump, and writeback behavior; debugged datapath and control faults using Vivado waveforms.

Vivado Packager: Git Automation CLI for FPGA Projects (Python, Git, AMD Vivado)*Jul 2026*

- Built an open-source Python CLI that converts AMD Vivado projects into reproducible Git-ready repositories, preserving HDL, constraints, Tcl scripts, memory files, and project metadata while excluding generated artifacts.
- Implemented dry-run previews, automatic `.gitignore` and README generation, Git initialization, remote setup, optional GitHub push, and unit tests using Python's `unittest` framework.

AI Stress Prediction & Health Insights Platform (Python, FastAPI)*Aug 2025*

- Built a full-stack machine-learning application comparing 15 classification, regression, and clustering models, with evaluation dashboards and real-time stress predictions delivered through a FastAPI backend and Tailwind CSS interface.