Donovan Sproule

d.sproule@columbia.edu | (562) 673-2654 | github.com/dsproule | linkedin.com/in/donovan-sproule

EDUCATION

Columbia University, The Fu Foundation School of Engineering and Applied Science

Masters of Science – Computer Engineering, (4+1 program), GPA: 3.9/4.0 Aug 2025 – Dec 2025

Bachelor of Science – Computer Engineering, GPA: 3.63/4.0 Aug 2023 – May 2025

Relevant Coursework: Operating Systems, Embedded Systems, Advanced Logic Design, Databases, System on Chip Platforms, Neural Networks & Deep Learning, GPU Heterogeneous Computing, Computer Architecture, Computer Communication Networks, Compilers, Formal Verification, Performance Evaluation & Modeling, Datacenter Processing

LANGUAGES AND TECHNICAL TOOLS

Languages: SystemVerilog, RISC-V, Mips, C, C++, C#, Python, SQL, CUDA, UNIX, Chisel

Tools: FPGA, Quartus, TCP/UDP, Jasper, PyTorch, TensorFlow, Linux, SoC, Bison/Flex, Git, Numpy

WORK EXPERIENCE

Apple, *Machine Learning Intern* May 2025 – Aug 2025

- Served as the sole engineer for developing a machine learning tool, presenting weekly on design decisions, implementation details, and results to cross-functional stakeholders
- Designed and implemented unsupervised neural network models to analyze high-dimensional ASIC validation data, automating anomaly detection and visualization workflows for production silicon
- Reduced manual investigation space from +14,000 samples to ~10 and accelerated failure analysis from +22 hours to about 30 minutes for pre-silicon validation

Robotics and Rehabilitation Lab, Columbia University, *Software Engineer and Researcher* May 2024 – May 2025

- Designed Augmented Reality (AR) experiments in C++ to gather postural data for ML models training robotic postural assistance to wheelchair-disabled patients
- Developed asynchronous, peer-to-peer TCP system to synchronize robotics frame using event-based RPC's

Systems Lab, Columbia University, *Kernel Developer and Research Assistant* May 2024 – Aug 2024

- Independently integrated changes from older 5.x kernel to newer 6.x to update custom hypervisor framework facilitating security of host and VM data from attacker access on an already compromised machine

Columbia University CS Department, *Embedded Systems Teaching Assistant* Jan 2025 – May 2025

- Held weekly lab sessions for 120+ students providing support for RTL design and low-level embedded software
- Mentored 5 projects with groups of up to 6 students, guiding system architecture, technical specs and IP integration

ENGINEERING PROJECTS

SystemVerilog RISC-V Out-of-Order P6-style Pipelined Processor

- Developed synthesizable, reduced-set RISCV-32I ISA processor verified via 50+ C, ASM and Verilog testbenches
- Featured 256B Dcache, non-blocking 256B prefetched Icache, split LSQ, two-bit branch predictor for 2x improvement on original design
- Performed formal verification using SVA and Jasper to prove liveliness and safety properties of system and IP

HW/SW Co-design of Memory Tracing Profiler for BOOM RISC-V Core

- Developed a firmware library in C and Assembly to provide a direct and configurable interface for software access
- Integrated a custom hardware tracer into the BOOM RISC-V Chisel core to enable precise, low-overhead memory access tracing for memory bound datacenter-scale workloads incurring only a 0.7% increase in power and area

hls4ml Surrogate Graph Model Randomized Quantizable Neural Network Dataset Generator (Open Source)

- Designed multithreaded Python framework to automate configurable, dataset generation for hls4ml graph-neural network generating an extensive dataset of over 100,000+ samples with a 99.95% model reliability rate
- https://arxiv.org/abs/2511.05615

Convolutional Neural Network Hardware Accelerator Design Space Exploration

- Constructed C++ SystemC convolutional neural network hardware accelerator for system-on-chip FPGA
- Performed design exploration to discover optimal designs, improving resource usage by 3x and latency by 187x
- Automated Verilator testbenches for incremental development before integration with USB controller and VGA

Advent of Code, Algorithmic Accelerators

- Re-designed algorithmic puzzle solutions as SystemVerilog accelerators, converting C++ logic into pipelined, parameterizable parallel RTL to improve throughput and asymptotic behavior.
- Verified designs with cycle-accurate testbenches and waveform-driven debugging.