Nathan Teshome

Cambridge, Massachusetts | nnt@mit.edu | +1 617 719 0373 | linkedin.com/in/nathan-teshome-84b675325 | github.com/nnt-git13

Education

Massachusetts Institute of Technology (MIT) Cambridge, MA
B.S. Electrical Engineering; B.S. Mathematics; GPA: 4.60/5.00 *Anticipated 2028*

- Pursuing NEET Autonomous Machines Certification.
- **Currently Enrolled:** Software Construction; Computation Structures; Semiconductor Electronic Circuits; Fundamentals of Statistics; C & Assembly.
- **Completed:** Algorithms I; Probability & Random Variables; Machine Learning; Circuits; Differential Equations; Linear Algebra; Electricity & Magnetism; Fundamentals of Programming (Python).

Academic & Personal Projects

FlashFlow: GPU-Optimized Transformer Inference Runtime *GitHub: github.com/nnt-git13/flashflow*

- Trained a small GPT-style Transformer achieving <0.5 perplexity gap vs a strong PyTorch baseline on held-out data.
- Implemented fused CUDA/Triton kernels for attention and MLP using tiling and shared-memory blocking, yielding 1.4–1.9× kernel-level speedups in internal benchmarks.
- Built a C++ scheduling runtime lowering FX graphs into fused GPU operators with auto-tuned tile sizes; improved tokens/sec by 1.3–1.6× vs PyTorch eager.
- Integrated FP16/BF16 quantization and an early INT8 path; observed 2–3× higher arithmetic throughput with minimal accuracy degradation.

RISC-V Vector CPU + Quant Finance Accelerator + Custom PCB *GitHub: github.com/nnt-git13/riscv_quant.accel*

- Built a functional RV32I soft core with preliminary custom vector instructions and a memory-mapped accelerator interface on an Artix-7 FPGA.
- Implemented a Monte-Carlo pricing accelerator with SIMD-style parallelism, fixed-point numerics, and reduction trees; achieved order-of-magnitude speedups vs Python/C++ baselines.
- Designed a 4-layer PCB including regulators, level-shifting, and PMOD expansion; validated core functionality through SystemVerilog benches and golden-model testing.

Mini-TPU: MLIR Compiler + Programmable Tensor-Core Accelerator *GitHub: github.com/nnt-git13/TODO*

- Developing a programmable tensor-core accelerator (systolic array + DMA engine) with a custom MLIR dialect supporting tiling, fusion, and quantization-aware lowering.
- Early FPGA prototypes validate correctness with small FP16/BF16 workloads; initial tests show 2–3× throughput gains on matrix-multiply kernels.

Work Experience

MIT Brain-Score AI Neuroscience Benchmarking Platform Cambridge, MA
Undergraduate Software Engineer — Leaderboard Infrastructure *Jun. 2025 – Ongoing*

- Redesigned the Brain-Score model leaderboard (Python, Django, AG Grid) to support timestamp-based filtering, multi-index score aggregation, and cross-model comparison; improved load latency by ∼35%.
- Implemented a dynamic Wayback Slider for historical model evaluation, wiring dual-date sliders to Score-level metadata and adding backend API routes for time-bounded queries.
- Built new aggregation pipelines for recursive benchmark hierarchies (vision & language), adding BFS tree construction, bottom-up score fusion, and robust handling of missing leaf scores.

Satellite Swarm Reinforcement Learning Control — MIT Cambridge, MA
Undergraduate Researcher *Nov. 2026 – Ongoing*

- Ported EMFF swarm dynamics + RL controller into a GPU-parallel pipeline using vectorized integrators; increased simulation throughput by 8–12× on mid-range GPUs.
- Redesigned PPO/SAC components with custom rollout buffers and batched normalization, reducing training wall-time from multiple days to under 10 hours in internal tests.
- Prototyped attention-based multi-agent policies (transformer + GNN hybrids) achieving more stable convergence and reduced ΔV usage across held-out orbital scenarios.

Technical Skills

Programming: Python, C, C++, SystemVerilog/Verilog, Bash; PyTorch, NumPy/Pandas, LLVM/MLIR; ROS 2, OpenCV; Django/REST; JavaScript; CI/CD (pytest, CMake, GitHub Actions).
Hardware: FPGA (Zynq-7000/Artix-7/ZCU104), Vivado/Vitis, PYNQ; AXI4/AXI-Lite/AXI-Stream, BRAM/DSP48; cocotb, riscv-dv, SymbiYosys; oscilloscopes, logic analyzers, JTAG; KiCad PCB design, soldering.
Tools: Linux, Docker, Git, SLURM, RViz2, Onshape, 3D printing/rapid prototyping.