Mahdi Hassen

github.com/MahdiHassen linkedin.com/in/mahdi-hassen mhassen@uwaterloo.ca 647-546-5883

Summary of Qualifications

2nd year MASc student researching HW/SW co-design for memory Quality of Service in heterogeneous Systems-on-Chip. Strong background in C, C++, Python and SystemVerilog, as well as CPU & accelerator architectures, heterogeneous systems, x86, ARM, RISC-V ISAs and AI/ML workloads through research experience, coursework and projects.

Education

Master of Applied Science, Electrical & Computer Engineering Exp. Apr 2027
University of Waterloo *GPA: 4.0/4.0*
Key Courses: On-Chip Interconnect, Quantum ML, Computer Architecture, RTL Systems, AI/ML Hardware

Bachelor of Engineering, Computer Engineering Apr 2025
Toronto Metropolitan University *GPA: 3.6/4.0*

Skills

Languages: **C**, **C++**, **Python**, Shell scripting, SystemVerilog, RISC-V / ARM Assembly
Performance Engineering: Linux `perf` / `perf_event`, performance counters, cache/DRAM microbenchmarking, GDB
Systems Software: Linux kernel drivers, pthreads, bare-metal C, cross-compilation, Git
Data & ML: compression algorithms, pandas, TensorFlow/Keras, PyTorch, MySQL
Hardware Acceleration: HW/SW co-design, AXI4, FPGA (Vivado, Vitis HLS), memory QoS (MPAM, RDT, CBQRI)

Experience

Graduate Researcher | *University of Waterloo* Apr 2025 – Present

- Developed a **token-bucket memory bandwidth regulation** mechanism in C on a dedicated monitoring core, correlating LLC and main-memory events across 16 hardware counters to enforce per-core budgets with a **3.02** μs control loop, cutting execution-time variation under memory contention from **138.9% to 20.7%**
- Built a **function-level performance attribution** system with zero-overhead profiling, isolating **2.7–3.8×** **slowdowns** in memory-bound vision benchmarks (SD-VBS) to LLC latency contention
- Booted **Linux** on a custom quad-core RISC-V SoC and integrated core-level hardware counters into the `perf_event` subsystem for standard `perf` tooling
- Characterized cache and DRAM latency with **pointer-chasing and stride-sweep microbenchmarks** in C
- Researching **Arm MPAM**-style memory tagging QoS architecture for **virtualized environments**, **I/O** and **accelerator-side QoS**, co-designing hardware partitioning mechanisms with software-level policy
- Wrote **bare-metal C firmware** and synthetic memory stressors; built **Python** pipelines decoding JTAG memory dumps into pandas datasets for analysis

Projects

GoSPA Sparse CNN Accelerator & Compiler | *SystemVerilog, Python, PyTorch*

- Designed and implemented the **full RTL** for a **sparse CNN inference accelerator** (GoSPA, *Deng et al.*) in SystemVerilog, specifying the dataflow, Processing Element (PE) array sizing and on-chip routing
- Built a **compiler** in Python that lowers CNN layers onto the accelerator (**operation-to-PE mapping**, scheduling, configuration generation); now **optimizing** compiled mappings for throughput and PE utilization
- Built **performance and golden-reference models** in Python to validate RTL and project throughput/energy

SZ Lossy Compression Hardware Accelerator | *SystemVerilog, Python, cocotb, Vivado*

- Designed a hardware accelerator for **SZ lossy compression** of scientific datasets, sustaining **1.6 GB/s** at **6.5:1–12.8:1 compression ratios** with bounded error, verified byte-exact against a software reference model
- Built **Python** tooling for Huffman table generation, improving data compression by up to **54%**

Real-Time MJPEG Video Decoder | *C, pthreads, Linux kernel, Vitis HLS, KV260*

- Accelerated an MJPEG decode pipeline from **7 FPS to 1080p60** on a quad-core ARM + FPGA SoC by profiling the baseline, offloading IDCT and color conversion to hardware and pipelining decode across cores with **pthreads**
- Wrote a **Linux kernel driver** (CMA allocation, `mmap`, cache-coherency `ioctl`) for direct C control of DMA engines and accelerator registers

Publications

A Centralized Performance Monitoring Architecture for Heterogeneous Multicore SoCs | **EMSOFT 2026**, Accepted