

Work Experience

- Jan 2022 - **Compiler Engineering Intern**, *Groq Inc.*, Mississauga ON (remote)
- Apr 2022
- Increased neural network inference throughput by up to 40% by designing algorithms in **C++** to efficiently utilize hardware resources for common tensor operations (e.g. convolutions)
 - Created optimization passes in **C++** using the **MLIR** compiler framework to manipulate neural networks described in **ONNX** format
 - Created machine learning models in **PyTorch** to run end-to-end compiler tests and measure cycle-accurate performance when run on custom neural network accelerator hardware
- Jan 2020 - **Software Engineering Co-Op**, *RadComm Systems*, Oakville ON
- Aug 2020
- Researched cutting-edge radiation detection and identification techniques using **GNU Octave** and **Python** for data visualization to assess development options
 - Implemented algorithms in **C#** to analyze radiation patterns using the **ReactiveX** library to handle real-time data emitted by an embedded device, processing energy histograms every 100ms
 - Automated device calibration process using **C#** to allow parallel setup of many devices
- Sep 2020 - **Undergraduate Research Assistant**, *University of Waterloo*, Waterloo ON
- Dec 2020
- Wrote **C** implementation of novel post-quantum cryptographic algorithms
 - Implemented cache-aware optimizations resulting in 60% speed improvement
 - Created custom boolean matrix library for use in cryptographic algorithms
- May 2019 - **Secure Software Developer**, *ESCRYPT*, Waterloo ON
- Aug 2019
- Implemented asynchronous process in **C++** for periodically provisioning digital certificates on-vehicle, improving anonymity in the system by enabling certificate swapping
 - Wrote ETSI-compliant tests using **GoogleTest** framework to prove functionality

Projects

- May 2022 - **Bayesian Network Accelerator**, Python — VHDL
- Aug 2022
- Created RTL design for inference over a Bayesian network leveraging parallelism, efficient discrete sampling algorithms, and Markov-Chain Monte-Carlo methods (e.g. likelihood weighting)
 - Created **protobuf**-based specifications for model description and elaboration
 - Implemented compiler in **Python** to analyze models and emit **VHDL** for accelerator
- Feb 2022 - **CHIP-8 Emulator**, C++ — SDL2 — ImGUI
- Mar 2022
- **C++** interpreter for CHIP-8 instruction set, runs publicly available ROMs
 - Includes graphical and audio interface using **SDL2**
 - Designed live debugger using **ImGUI** to inspect memory dumps and processor state
- Dec 2021 - **3D Rasterized Render System**, C++ — CMake — OpenGL
- Jan 2022
- 3D rasterized rendering system written with **OpenGL 3.3** in **C++17**
 - Implemented mesh generation, texture loading and phong lighting shaders
 - Enabled loading models from common file types based on the **Assimp** library
- Sep 2021 - **Pipelined 32-Bit RISC-V Core**, Verilog — Verilator
- Nov 2021
- Implemented RV32I spec in **Verilog** using a 5-stage pipeline design with register bypassing, simulated test programs (individual instructions and benchmark algorithms) using **Verilator** to verify design
 - Wrote **Python** script to run standardized RV32I instruction and benchmark tests

Education

- Sep 2018 - **University of Waterloo**, *Candidate for Computer Engineering B.A.Sc.*, Waterloo ON
- Apr 2023 (expected)
- Relevant coursework and projects in:
- | | | |
|-------------------------|--------------------------|-----------------------------|
| ○ Computer Architecture | ○ Operating Systems | ○ Digital VLSI |
| ○ FPGAs | ○ Compilers | ○ Computer Security |
| ○ ARM & RISC-V ISAs | ○ Reinforcement Learning | ○ Digital Signal Processing |