

**Languages:** Python (**PyTorch**, Numpy, Pandas, Flask), C++ (**CUDA**, Boost), MATLAB, Rust, Go, SQL, RISC-V, Verilog

**Tools:** Docker, Postman, REST, KiCAD, LTSpice, Cadence Virtuoso, LayoutEditor, Quartus Prime

## Experience

### POST-SILICON VALIDATION ENGINEER

NVIDIA - Contractor (6 months) 

March 2024 - Present | Santa Clara, CA, USA

- Debug **PCIe 5.0** failures on current and upcoming SoCs and GPUs
- Developed testing software in **Python** for validation during bringup of upcoming SoCs
- Identified high speed IO failure modes during bringup of upcoming and existing products

### SOFTWARE ENGINEER - HPC FIRMWARE

Groq Inc. 

Jan 2022 - Apr 2022 | Mountain View, CA, USA

- Designed algorithms for resource allocation involving SRAM and ALUs on Groq's Tensor Processing Unit (TPU)
- Developed **Python** and **C++** firmware APIs to improve throughput and minimize end to end latency of instructions and data
- Leveraged timing analysis to prevent data stream conflicts while maximizing ALU utilization
- Leveraged **PyBind11** for interoperability between C++ and Python firmware during codebase migration

### SOFTWARE ENGINEER - DIGITAL COMPRESSION

Huawei Technologies

May 2020 - Aug 2020 | Waterloo, ON, CA

- Designed non-cryptographic hash functions (NCHF) with **linear algebra**, **SAT** and self-designed  $GF(2)$  matrix solver to verify properties like collision resistance, distribution uniformity, and computational efficiency
- Benchmarked the optimized SIMD hashing function against existing NCHFs using **Rust** and **C++**
- Implemented novel border detection algorithms in **Go** using **probabilistic data structures** to maximize performance

### SOFTWARE ENGINEER - MACHINE LEARNING

MappedIn 

Sept 2019 - Dec 2019 | Waterloo, ON, CA

- Engineered **Clustering** + **SVM** + **Random Forest** ensemble models to replace existing LSTMs, reducing inference costs **2x** while maintaining prediction accuracy
- Increased prediction accuracy from **40%** to **80%** on existing **LSTM** models with feature engineering, hyperparameter optimization, and automated data cleaning (**Python**, **SQL**)
- Designed pipelines for data cleaning and analysis and integrated new **SQL** data warehouse

### ELECTRICAL ENGINEERING RESEARCH ASSISTANT - DISPLAY SEMICONDUCTORS

University of Waterloo 

Sept 2022 - Apr 2023 | Waterloo, ON, CA

- Designed custom PCBs in **KiCAD** for driving small  $\mu$ LED active/passive matrix displays using **STM32** MCU
- Developed research plan for packaging  $\mu$ LEDs onto TFT backplane using indium electroplating
- Designed characterization setups for  $\mu$ LEDs in **Fusion360** and **Arduino** interfaced with **Python**
- Validated flip-chip diebonding results with thermal and electrical simulations in **MATLAB**
- Designed and validated new  $\mu$ LED layouts to improve mechanical and electrical performance

### TEST AUTOMATION ENGINEER - ELECTRICAL COMPLIANCE

Enphase Energy 

Aug 2023 - Mar 2024 | Fremont, CA, USA

- Developed software systems in **Python** to test high and low voltage electrical systems under a variety of strenuous conditions
- Implemented **Jenkins** to enhance testing platform reliability by reducing software errors
- Engineered a logging and traceability system using **Go**, **Gin**, and **MySQL** to reduce errors during regulatory submission

## Education

### UNIVERSITY OF WATERLOO – B.A.SC ELECTRICAL ENGINEERING 2018 - 2023

- Key Courses: **Computer Architecture**, Multivariable Control Systems, Analog/Digital Integrated Circuits
- Select Awards and Certifications: Baylis Medical Capstone Design Award, QNFCF and G2N Cleanroom Certifications

## Projects

### C++ COMPILER FOR C++ LIKE LANGUAGE

- Designed a lexical tokenizer and compiler in **C++** to generate **RISC-V** assembly for custom programming language
- Used *Spike* RISC-V simulator to verify correctness of assembly
- Used **CMake** (build management tool), **Catch** (unit-testing framework), **Boost** (graph library/dotviz generator)

### 3D RAY TRACING ENGINE

- Implemented 3D recursive path-tracing for arbitrary materials on basic geometric shapes
- Used **nalgebra** for arbitrary rotations and positions of camera and objects
- Parallel processing of ray-tracing using **rayon** yielding **~10X** performance speed-up on CPU

### BEAMFORMING HEARING AID SYSTEM

- Designed 4-channel microphone array PCB with active analog bandpass filtering, diff. amp., and multichannel **ADC** over **SPI** to R-Pi (**KiCAD**)
- Created **Flask** server on R-Pi to compress and transfer audio data to **Pytorch** neural network for further digital filtering and beamforming
- Adapted and trained Pytorch quantized voice isolation model to minimize latency while maintaining desired audio quality
- Used **multiprocessing**, **asyncio**, and **websockets** to maximize system throughput, providing continuous audio output

### PIPELINED RISC-V CORE

- Designed 5-stage pipelined **RISC-V** 32-bit core in **Verilog** using only synthesizable constructs
- Added pipeline registers and feedback/data-forwarding paths to optimize performance for branching, memory and write-back operations
- Core synthesized on **FPGA** and successfully ran branching and recursive algorithms. Testbenches used to ensure cycle accuracy

### MULTIPLE SEQUENCE ALIGNER

- Wrote sequence aligner for novo assembly of short sequences using Progressive Alignment Construction using the Needleman-Wunsch algorithm
- Written in **Go** to take advantage of light weight green threads, used greedy heuristics to reduce  $O(n!)$  problem to  $O(n^2)$

### HYBRID MUSIC SYNTHESIZER

- Designed and fabricated a hybrid synthesizer for realtime music synthesis; digital synthesis on Teensy 4.1 Cortex-M7 MCU and analog synthesis using **ADC/DAC** and analog filters
- Designed PCB in **KiCAD**, wrote UI and audio processing code using **Rust**
- **RTIC** (**RTOS** library) used to manage concurrency on MCU, **Slint** library used to manage UI on display; priority based scheduling used to prevent timing conflicts between UI updates, audio synthesis, and input handling
- Developed custom digital signal processing (DSP) modules for oscillators, filters, and envelope generators in Rust, optimizing for **low latency** and **high throughput**
- Achieved sub-millisecond latency and low power consumption while maintaining high (24-bit) quality audio
- Applied real-time audio synthesis techniques, including wavetable synthesis, FM synthesis, and subtractive synthesis, to generate a wide range of sounds and effects