

Michael Gionet

Michael.Gionet@protonmail.com
705-650-0187

Skills

Programming Languages:

C, C++, Python, Assembly Language (for x86 and RISC-V), MATLAB, Chisel, Verilog, C#, Scala, Java, Javascript.

Software and Hardware:

Linux, Bash, Git, Verilator, GTKWave, Arduino, Raspberry Pi, Valgrind, Soldering, Breadboard and Circuits.

Work

Research Intern – Hardware Implementation of Fully Homomorphic Encryption – University of Waterloo Sept to Dec 2023

- Used Chisel, Verilator, Python, and C++ to learn, prototype, design, develop, and debug a core in RTL that implements fully homomorphic encryption schemes, such as CKKS and BFV.
- Intend to do hardware explorations regarding optimization, as well as document the core, putting it on an FPGA, and open sourcing it during my Winter 2024 term.

Digital Signal Processing Software Developer – Coherent Logix Jan to Oct 2023

- Used C, MATLAB, and inline assembly to implement algorithms for a software defined radio.
- Implemented packet detection, fast walsh transform based demodulation, denoising filter, for 802.11b on the PHY layer.
- Heavily optimized with inline assembly for HyperX; a fabric based CPU architecture, to meet real time requirements.

Firmware Developer – Onsemi (Formerly ON Semiconductor) Jan to April 2022

- Used C, Git, Saleae logic analyzer, and J-Link Segger, to maintain USB-C power delivery MCU firmware, by fixing bugs such as adding a firmware fix for a faulty i2c to DMA hardware interface.
- Implemented new features, like auto-downgrading the PD standard for compatibility.

Projects

FHE Core – For research internship (to be open sourced)

- Learned about Fully Homomorphic Encryption by reading research papers, and by implementing a python prototype.
- Developed the core in Chisel. It features an ALU, register file, FFT, and NTT accelerators, and a custom instruction set.
- Debugging was done using verilator, with correctness verified against Microsoft SEAL.

RISC-V Virtual Machine – Personal ([github](#))

- Runs 64 bit RISC-V machine code, by generating control signals and using them to control the ALU, Register File, Constant Generator, Memory Unit, and Peripherals.
- Programmed it from scratch with C++, using a processor design from my textbook, Computer Organization and Design (Patterson, Hennessy), as a reference.

Flood Detector – Personal

- Result was my basement surviving the spring melt.
- Concocted with an Arduino, a water sensor, and a buzzer.
- Device polls periodically for sensor contact with water, and tells the buzzer to beep if so.

Education

Candidate for Bachelor of Applied Science; Computer Engineering – University of Waterloo (2019 – 2024)

Additional Experience

Waterloo Rocketry Club Jan to April 2022

- Got exposure to control systems in the context of guiding a payload using a rogallo wing.
- Experience soldering electronic components onto custom boards.

Independent study in Advanced Mathematics

- Self-directed exploration of advanced mathematics through intensive study of mathematics textbooks in areas such as Analysis and Number Theory, learning from the best authors such as Terence Tao, Elias M. Stein, and Tom Apostol.
- Independently discovered theorems, and documented them with corresponding proofs ([github](#)).