# An Qi Zhang

*Mobile: +1 3853717095 | E-mail: anqizhang01@gmail.com | GitHub: https://github.com/userAZ*

## Technical Skills

| | |
|---|---|
| Languages | Lean4, C, C++, VHDL, Verilog, Bash, Python, LaTeX |
| Tools | Vim, Emacs, git, GDB, Make, Vivado, Quartus, Cadence, ADS |

## Work Experience

**NVIDIA - GPU Architecture Intern** *January-April, September-December 2018*

- Wrote scripts in Perl and Python to distribute the testing of instructions on an assembler
- Wrote a Python script to generate Intermediate Representation (IR) headers in C++ for an assembler
- Mapped some instructions between two ISAs for an assembler
- Wrote fullchip functional tests for functionality of instructions after reading their specifications

**IBM - C/C++ Compiler Front-End Developer** *May-August 2017*

- Worked with Basic Blocks and LLVM IR for an optimization pass to fix a Basic Block ordering issue
- Wrote tests to ensure test coverage after implementing features and fixing bugs
- Used GDB to debug Clang front-end

**CAE Inc. - Software Developer** *September-December 2016*

- Implemented changes and bugfixes to terrain loading server with C++ in Visual Studio
- Designed procedure to extract tiles across different zones in the CDB specification
- Worked with various graphics data structures while carrying out tasks and fixing bugs

**ISED Canada - Software Developer, Radio Hardware Acceleration** *January-April 2016*

- Synthesized hardware blocks from C code with Vivado HLS
- Integrated programmable logic and processing systems with Vivado IP Integrator
- Set up UHD and GNU Radio for testing RFNoC on a USRP E310

**University Health Network - Bioinformatics Research Assistant** *May-August 2015*

- Set up OTRS ticketing system and documentation for admins
- Used XAMPP and the Drupal CMS to develop websites front-end and back-end
- Tested genome sequencing tools and wrote documentation

## PhD Work

- Developed DSL (Domain Specific Language) for microarchitecture
- Developed a compiler tool to take microarchitectures specified and automatically add memory ordering mechanisms to the microarchitecture
- Wrote a translation from the DSL to the Murphi model checker code

## Master's Work

Thesis: Struck-at Fault Tolerance with Emerging Technology RAM in the NeuroSim MLP Neural Network System

- Searching and reading papers on CNFET to understand current research progress
- Researching techniques to counter the current process faults of an emerging process technology
- Modifying NeuroSim (Neural Network Simulator), measuring SRAM stuck at fault impact on accuracy
- Conducting a literature review on memory stuck at fault recovery and mitigation in order to develop memory stuck at fault recovery schemes

## Publications

- Zhang, An Qi, Amr MS Tosson, and Lan Wei. "Error Resilience and Recovery of Process Induced Stuck-at Faults in MLP Neural Networks using Emerging Technology." 2021 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH). IEEE, 2021.

## TA Work

University of Waterloo

ECE 351 Compiler Intro, OOP, RegEx & Automata, LL(1) Proofs, Design Patterns (ex. Visitor, Singleton), Optimization, Register Allocation, Object Allocation, Garbage Collection. *Jan-Apr, May-Aug 2020*

ECE 650 (University of Waterloo) Graduate Intro to Software Tools & Methods, systems programming, calls, libraries, propositional logic, data structures, scripting *Sept-Dec 2020*, *Jan-Apr 2021*

ECE 350 (University of Waterloo) Real Time Operating Systems, system calls, context switching, memory allocation, scheduling (ex. Min Max Fair Scheduling), synchronization, cache coherence, virtual memory *May-Aug 2021*

Work: set up student Git repos, answering student questions on a forum, making exam questions, gdb primer

University of Edinburgh

Inf2c-cs Introduction to computer systems. Intro to C, finite state machines, combination logic, a simple 5 stage processor, caches *Sept-Dec 2022*

University of Utah

CS6810 Computer Architecture. 5 stage pipeline, pipeline hazards, scoreboarding, tomasulo's, caches, cache coherence, memory consistency, synchronization *Aug-Dec 2023*

## Extracurricular Work

URA (Undergrad RAship) - Evaluated CNN acc. from faulty sigmoids in Caffe *Parallel to studies Jan-Apr 2019*

URA - Scripting, boolean equation reformatting, technology mapping *Parallel to studies May-Aug 2018*

## Education

University of Waterloo, Bachelors of Applied Science, Honours Computer Engineering (co-op) with Distinction

Final average of 80. Converts to a of GPA 3.7. *Convocation June 2019*

University of Waterloo, Master's of Applied Science in Electrical and Computer Engineering

Final average of 82. Converts to a GPA of 3.7 *Completed Summer 2021, Convocation Spring 2022*

University of Edinburgh, PhD Candidate in Informatics: ICSA: Comp. Architecture, Compilation & System Software, Networks & Communication *2021-2023*

University of Utah, PhD Student in Computer Science. *2023-Present*

## Taken Courses of Interest

University of Waterloo

- ECE 429 Computer Architecture: CPU Arch, Tomasulo, reorder buffer, register renaming, memory hierarchies, coherency, pipelining, checkpointing

  Project: Verilog basic 5 stage pipelined MIPS processor

- ECE 423 Embedded Computer Systems: HW/SW co-design, NoCs, formal methods LP (Linear Programs), ILP (Integer Linear Programs), MILP (Mixed Integer Linear Programs), SMT programs, iterative methods, Kahn Networks

  Project: Real time video decoding on FPGA, designing a schedule and synthesis of a DCT block

- ECE 413 Digital Signal Processing: Linear & Cyclic convolution, DFT, FFT, & Z transform, FIR and IIR filters, Butterworth, Chebychev, and Elliptical filters, Bilinear transformation

- ECE 327 Digital Hardware Systems: optimizing digital HW, pipelining circuits, dataflow diagrams, elmore delay, technology mapping to FPGA LUTs

  Project: VHDL Kirsch edge detector. Converted the kirsch edge detection algorithm into a dataflow diagram to determine how to optimally map the algorithm onto concurrent hardware, and wrote the VHDL for the hardware

- ECE 351 Compilers: OOP, RegEx & mapping to NFA (Non-deterministic Finite Automata) and DFA (Deterministic Finite Automata), LL(1) Proofs, Design Patterns (ex. Visitor, Singleton), Optimization, dataflow analysis, Register Allocation, Object Allocation, Garbage Collection.

  Project: A compiler for a subset of VHDL. Implemented parsing through recursive descent parsing, implemented basic optimizations in boolean expressions, and technology mapping to AND, OR, and NOT gates while also reusing common sub-expressions.

- ECE 637 Digital ICs: Logic families, registers, schmitt trigger, timing, wires, adders

  Project: schematic design and simulation of a 16 bit Han Carlson parallel prefix adder in the TSMC 65 nm process

- ECE 636 Adv. Analog ICs: Amplifiers, biasing, switch cap circuits, reference circuits, signal flowgraphs

  Project: schematic design and simulation of a 2x sample and hold switch-cap circuit in the TSMC 65 nm process

University of Utah

- CS 6520 Programming Languages: Functional Programming, Lambda Calculus, Lazy Evaluation, Continuations, Interpretation, Garbage Collection, Objects, Types, Type Checking, Typed Recursion, Typed Inference, Parametric Polymorphism