

JAKE EDMONSTONE

jbedmons@uwaterloo.ca | linkedin.com/in/jake-edmonstone | github.com/jake-edmonstone
jake-edmonstone.github.io

EDUCATION

University of Waterloo | GPA: 88.32

2022 – Expected 2027

Bachelor of Mathematics | Double Major in Computer Science and Pure Mathematics

Waterloo, ON

EXPERIENCE

Backend Compiler Engineer

January 2026 – April 2026 (Co-op)

Cerebras Systems Inc.

Toronto, ON (On-site)

- Designed and implemented a general instruction annotation mechanism in our custom assembler as a new MC-layer directive, originally to tag brk instructions with reasons such as assertion failures and debug breakpoints; also designed matching LLVM IR-level annotations.
- Updated LLVM backend lowering for a new pre-silicon target: wrote new TableGen instruction-selection patterns for immediates (handling an address-space change), adapted inline-asm lowering, and fixed compiler-rt functions.
- Contributed to a formal ISA description used to generate parts of the compiler, adding missing instructions and refactoring instruction formats and operand types.
- Landed 10+ bug fixes and diagnostic improvements in our custom assembler: decoder correctness, post-encoder validation for immediate conflicts, accurate error locations through macro/#define expansions, and empty task section validation.

Software Engineer

May 2025 – August 2025 (Co-op)

Constant Contact Inc.

Kitchener, ON (Hybrid)

- Shipped features on an email delivery product (Java, Next.js), including page persistence on a campaign picker page.
- Added automated tests and fixed bugs on the Java backend.

Software Developer

January 2024 – April 2024, September 2024 – December 2024 (Co-op)

BCS Automation Ltd.

Belleville, ON (On-site)

- Designed and trained LSTM autoencoders in PyTorch for unsupervised anomaly detection on time-series sensor data.
- Migrated the application from a relational store to CosmosDB to address query slowness, designing partition keys and indexes for sensor data (order of 10 billion rows) and improving throughput on the hot query path.
- Built out sensor-data ingestion into Azure Data Explorer and authored KQL queries against it, replacing an ad-hoc process with no persistent store.
- Refactored the back-end API to follow HTTP semantics (e.g. idempotent GET requests) and standard REST conventions.

PROJECTS

WLP4 Compiler [↗](#)

C++, Make

- Built an end-to-end compiler in C++ that lowers a subset of C-like source (WLP4) to MIPS assembly, structured as distinct scanner, parser, type-checker, and code-generation passes.
- Implemented a maximal-munch DFA scanner, an LR(1) parser driven by a precomputed SLR(1) DFA, a context-sensitive type checker, and a stack-based code generator emitting MIPS.

Sudoku-to-SAT Reduction [↗](#)

C++, Make

- Implemented a Karp reduction from 9×9 Sudoku to Boolean satisfiability (DIMACS CNF) and a decoder mapping SAT solver assignments back to grids.
- Compared minimal and extended CNF encodings, benchmarking solver performance on each.

EXTRA CURRICULARS

- Directed Reading Program (DRP) in analysis/differential equations, Spring 2026
- Algebra Tutor at the University of Waterloo Mathematics Tutoring Center
- Volunteer tour guide for the Mathematics Society

SKILLS

Languages: C++, Python, C#, SQL, JavaScript/TypeScript

Compilers & Toolchains: LLVM, TableGen

Developer Tools: Git, Vim/Neovim, tmux

Libraries: PyTorch, Pandas, NumPy