Arjun Bhamra

U.S. Citizen | abhamra@gatech.edu | linkedin.com/in/arjun-bhamra | github.com/abhamra | abhamra.com

Research Interests

Compilers, programming languages, and quantum computation

Education

Georgia Institute of Technology Atlanta, GA
Bachelor of Science in Computer Science, GPA: 3.87 *May 2026*

Threads: Theory, Systems & Architecture

Research Experience and Internships

UW-Madison Qool Quantum Qompilers Lab September 2025 – Present
Quantum Compilers Researcher with Prof. Aws Albarghouthi *Remote*

- Implemented key parallelization for QQQ's synthesis-based optimizing quantum compiler WISQ

Georgia Tech CRNCH Lab July 2023 – Present
Quantum Programming Languages Researcher with Prof. Tom Conte *Atlanta, GA*

- Aided in the design and implementation of Qwerty, a quantum programming language that moves away from gate-based quantum computing and towards higher abstractions
- Engineered custom ASTs and optimization passes for the Qwerty compiler using the Python AST, C++, and custom MLIR dialects and rewrite passes
- Placed 2nd at the CGO Student Research Competition for **"Type Inference for Qwerty"** by implementing Hindley Milner type inference via a Rust-based AST
- Designed and implemented efficient circuit synthesis for Qwerty's "basis generators", enabling cleaner writing of generic Quantum Fourier Transform (QFT) circuits with recursive structure

IBM Research May 2025 – August 2025
Research Intern *Yorktown Heights, NY*

- Designed a new qubit-efficient quantum algorithm for solving the Traveling Salesman Problem
- Leveraged permutation encoding and Pauli Correlation Encoding, along with a K-Means path slicing strategy to reduce qubit overhead by 10x, paving the way for solving utility scale problems on quantum hardware
- Paper in progress, in collaboration with IBM partners
- Operated on TSPLIB instances orders of magnitude larger than existing hybrid-quantum methods

IBM Research May 2024 – August 2024
Software Engineering Intern *Yorktown Heights, NY*

- Developed key components for IBM's Rust-based hardware-level production Quantum Assembly (QASM) quantum compiler that improved speed by 7x and memory usage by 3.5x
- Spearheaded the development of a 85x faster data parsing and caching API for quantum hardware configuration information and waveform data through binary (de)serialization
- Implemented the compiler's intermediate representations (IRs) at the pulse calibration level
- Added additional support for QASM parsing, enabling extensibility and more language coverage

Quantinuum May 2023 – August 2023
Software Engineering Intern *Broomfield, CO*

- Created a Pythonic Domain Specific Language (DSL) to aid internal theory team in parsing code to Quantum Intermediate Representation (QIR) with a 30% improvement in workflow efficiency
- Leveraged the Python Abstract Syntax Tree (AST) to build a high level compiler from custom language to LLVM IR that enables conditional branching on quantum measurements and dynamic circuits
- Added constant folding, constant propagation, method inlining and loop unrolling optimizations which reduced output QIR by 15% for faster compilation

Georgia Tech Computational Physics Lab Aug 2022 – July 2023
Quantum Algorithms Researcher with Prof. Spencer Bryngelson *Atlanta, GA*

- Created two variational quantum algorithms (VQAs) for solving PDEs with Qiskit, etc.
- Expanded upon Variational Quantum Linear Solver (VQLS) and VQE for fluid dynamics applications

Publications

Austin J. Adams, Sharjeel Khan, **Arjun S. Bhamra**, Ryan R. Abusaada, Jeffrey S. Young, and Thomas M. Conte. **"Qwerty: A Basis-Oriented Quantum Programming Language"** *IEEE International Conference on Quantum Computing and Engineering (QCE25).*

Austin J. Adams, Sharjeel Khan, **Arjun Bhamra**, Ryan Abusaada, Anthony M. Cabrera, Cameron Hoechst, Jeffrey S. Young, and Thomas M. Conte. **"ASDF: A Compiler for Qwerty, a Basis-Oriented Quantum Programming Language"** *2025 IEEE/ACM International Symposium on Code Generation and Optimization (CGO '25).*

Projects

Open Source Contributor to Quantum repositories | *Rust, Python, unittest, Git*

- Added necessary quality of life changes for the IBM Rustworkx repository, a Rust-based Python graph library
- Added major feature support for the IBM OpenQASM 3 Parser, a performant parser based on rust-analyzer
- Provided bug fixes and feature improvements for Xanadu's JIT quantum compiler, Catalyst

rs_micrograd: Mini Automatic Differentiation Engine | *Rust*

- Designed an auto-diff engine with Rust based on Andrej Karpathy's Python micrograd engine
- Implemented expression graph-based gradient calculations for efficient neural network backpropagation

xv6 Operating System | *x86 Assembly, C, GDB*

- Implemented Copy-on-Write and lazy zero page allocation to a virtual memory system for a **2x** speedup
- Added kernel threading support for processors and leveraged this to create a custom userspace threading library from scratch, with spinlocks, mutexes, and conditional variables

Out-of-order Processor Simulator | *C++*

- Designed a trace-based out-of-order processor simulator leveraging Tomasulo's algorithm, with high throughout and accurate simulation of hardware components
- Implemented Tomasulo's algorithm via hardware such as the Register Alias Table, reorder buffer, and scheduling/dispatch queues

Leadership

Dependently Typed: Programming Languages and Compilers Club | *Co-President* | 2024 - Present

- Organized/coordinated talks from industry experts, professors, and students
- Gave talks on MLIR, Quantum Programming Languages, and Program Synthesis
- Mentored students in programming languages and compilers as a research field, enabling future success and open communication

Georgia Tech Esports | *Co-President* | 2023 - 2024

- Managed casual and competitive scenes for 15+ games under the GT professional banner
- Aided in implementing a long lasting, effective organizational structure for future generations

Technical Skills

Languages: Rust, Python, Java, C, C++, LLVM, MLIR, JavaScript, LateX
Libraries: pandas, NumPy, Matplotlib, unittest, Qiskit, Pennylane, PyTorch, Scikit-learn, Python AST, functools