Qifa(Richard) Wang

✉ qifaw2000@gmail.com | ☎ 617-816-5834 | in qifa-wang-28a180170 | rthelionheart24

Objective: Motivated engineering graduate passionate about computer architecture, SW/HW co-design, accelerated computing, HPC, quantum computing, and software engineering. Quick to adapt to new technologies and concepts.

Education

Computer Science & Engineering — *Master of Science in Engineering* Aug 2023 - May 2025
Rackham Graduate School, University of Michigan GPA: 4.00

Computer Science, minors in Math and Physics — *Bachelor of Science in Engineering* Aug 2020 - May 2023
College of Engineering, University of Michigan *Summa Cum Laude, Dean's List, University Honors*, GPA: 3.77

Skills

- **Coursework:** Computer Architecture and Microarchitecture, Parallel Computing and Architecture, GPU Programming, Machine Learning, Quantum Computing and Architecture, Compiler Design, Data Structure and Algorithms, Operating System, Web Systems
- **Languages:** C/C++, Python, Go, Rust, Java, CUDA, OpenMP, MPI, Verilog/SystemVerilog, Chisel, Tcl, Javascript, TypeScript
- **Frameworks and tools:** PyTorch, CUDA-Q, Qiskit, AWS, Docker, Chipyard, GPGPU-Sim, Synopsys (DC, VCS, Verdi), GNU, LLVM, Valgrind
- **Multilingual:** English, Mandarin, Cantonese, French

Work Experience

Apple Inc. — *Hardware Technology Intern* May 2024 - Aug 2024

- Implemented **PPROC method** to evaluate hardware coverage on pixel processing module down to bit-field-level. Coverage result provides feedback and guidance for arch test improvement. Constructed feedback loop using PPROC to optimize test coverage by adjusting parameters. Internalize hardware coverage PPROC method into **C/C++** testing infrastructure to run and provide feedback on-the-fly.

University of Michigan — *Graduate Student Instrctor* May 2023 - Present

- EECS 498: Quantum Computing - Discussions and labs on quantum theories, algorithms and circuits using **Qiskit**.
- CSE 587: Parallel Computing - Discussions and labs on **OpenMPI, OpenMP, and CUDA**.

Werfen — *Software & Algorithm Development Intern* May 2021 - Aug 2021

- Drafted roadmaps and formalized criteria and limitations for Mercury Algorithm Prototype (MAP). Engineered **embedded** modules, **front-end web tool**, and GUI for the prototype using **C/C++ and Qt framework**.

Technical Projects

Architecture — *R10K Out-of-Order Processor based on RISC-V ISA* Jan 2024 - May 2024

- Designed and implemented out-of-order superscalar processor with N-way execution at **RTL level** using **SystemVerilog**, featuring **Tomasulo's algorithm**, instruction and data caches (prefetching, associative, and non-blocking with victim cache), G-share best branch predictor, return address stack, and reservation stations. Optimized performance with early tag broadcast and a robust memory hierarchy including a load-store queue with data forwarding. Achieved a 30% improvement in CPI and a clock period of 15.5ns through various architectural enhancements and pipeline optimizations. Verified and tested the design using **Synopsys DC** and exhaustive benchmarks.

GPU — *Batched Quantum Circuit Simulation on CUDA-ready GPU* Sep 2023 - May 2024

- Developed a **CUDA-based** quantum simulation framework for performing batched quantum experiments on GPUs, enabling parallel execution of quantum circuits with varying gate counts and types. Implemented a **synchronization strategy** to handle non-deterministic quantum operators and efficient **shot-branching**. Achieved **super-linear speedup** in runtime performance and optimized memory usage through careful task batching and state management.

Compiler — *Compiler Optimization for CUDA Memory Coalescing (COALDA)* Sep 2023 - Dec 2023

- Developed an **IR-level** CUDA compiler optimization tool using **AST and canonical forms** to transform uncoalesced memory accesses into coalesced patterns, achieving 9x L2 cache writeback reduction and 6.4x read bandwidth improvement, validated through **Clang-NVCC** integration and **NVIDIA Nsight Compute**.

Operating System — *Linux-Based Operating System* Jan 2023 - May 2023

- Developed a **custom thread library** to simulate multi-cpu, multi-threaded execution using **C/C++**. Implemented a sophisticated **kernel pager system** for efficient management of applications' virtual memory, encompassing the creation, copying, destruction, and allocation of address spaces. Engineered robust, **multi-threaded network file server** for reliable data exchange. Designed a hierarchical file system with comprehensive access control and fine-grained locking mechanisms to secure file ownership and permissions.