Viranchee Lotia

vlotia@ncsu.edu | linkedin.com/in/viranchee | github.com/viranchee | 7438372837

EDUCATION

North Carolina State University - MS, Computer Engineering (CGPA: 3.6) Jan 2022 - Dec 2023

Courses: Compilers, OS, Architecture: CPU, GPU, Parallel, Cloud; Accelerating Deep Learning

SKILLS

Languages & Frameworks:	C, C++, Swift, Python; MLIR, LLVM, PyTorch, JAX; OpenMP, MPI, CUDA; pthreads
Debugging:	lldb, gdb, DTrace, dlopen, dlsym, ptrace, Trace32, JTAG

WORK EXPERIENCE

Passive Logic - Compiler Intern, Utah (Remote) Aug 2023 - Dec 2023

- Generating and improving code coverage for Differentiable Swift (S4TF) to ~89% line test coverage.
- Isolating differentiable Swift compiler bugs and generating lean reproducers.

Qualcomm - Software Engineer Intern, San Diego May 2023 - Aug 2023

- Built novel object and heap sanitizer algorithm over compiler, linker and runtime stage.
- Performed crash-dump and live JTAG debugging via Trace32, and gdb.

Multiple firms - iOS Software Engineer, Mumbai Nov 2018 - Oct 2021

- Shipped Popviewers, increased Zalora's user-engagement by 30% using A/B UI, UIKit, SwiftUI, CoreAnimation.
- Mentored 2 associate software developers for contributing to Zalora's platform

Earth Energy EV - Embedded Engineer, Mumbai Jun 2017 - Oct 2018

- Programmed electric bike ECU and BMS peripherals over I2C, SPI, UART, USB.

PROJECTS

Autodidax Understanding JAX core mechanisms: Jan 2024 - present

- (ongoing) Implementing simplified features of the DL framework JAX including JVP, VJP, Vmap, Jaxpr, JIT tracing to explain the core mechanisms of JAX.

Compiler Optimizations (**MLIR**, **LLVM** passes and analysis): Mar 2022 - present

- MLIR pass written for constant folding 2 shift lefts, for the arithmetic dialect using OpRewritePattern.
- LLVM Implemented Code optimization techniques: Common Subexpression Elimination, Dead Code Elimination, Instruction simplification, Load Store Elimination pass over LLVM SSA. Eliminated >20% instr. on avg. with M2R.
- Implemented a Loop Invariant Code Motion (LICM) pass. Moved 0.2 - 5 instr. on avg. outside the loop using LICM for LLVM benchmarks, with net reduction of 1238 instr.
- Built Availability analysis, Value numbering, Constant propagation passes for an academic compiler infrastructure.

Bitwise Processing Language (**LLVM Front end**, Docker): Feb 2022, Feb 2024

- Gained experience reading, interpreting a language specification and implementing a simple programming language to LLVM SSA conversion using parser generators Flex and Bison.
- Stood in the rankings of the class, for least amount of instructions generated.

GPGPU Simulator (C++): Jan 2023 - May 2023

- Implemented GPU kernel & host code performing quantum gate simulator. Optimized with CUDA shared memory.
- Enhanced GPGPU simulator by bypassing L1 cache for Cache Unfriendly benchmarks using Address frequency heuristics, achieving +12% on average improvement on IPC.

Dynamic Instruction Scheduling for out-of-order Superscalar processor (C++): Feb 2023

- Developed a simulator for 9 stage OOO superscalar pipeline processor that fetches and issues multiple instructions per cycle based on RISC-V ISA.
- Analyzed the effect on IPC by varying ROB size, Issue Queue size and width of the instructions to be fetched.

Xinu Operating System (C, Rust, QEMU, x86 Assembly) Aug 2022 - Dec 2022

- Implemented Exponential Distribution scheduler and Linux 2.2-like scheduler implemented with XinuOS
- Implemented concurrent reader, single writer locks for XinuOS
- OS in Rust + QEMU with Paging, UART, Heap allocator design, Exception handling.

MusiKid (Visual Studio Code extension, Javascript, Node) Aug 2023

- Wrote a VSCode extension to stream music while programming, without leaving the environment. (Github)