Leiqi Ye

website | LinkedIn | GitHub |

Location: Ann Arbor, Michigan
Email: yeleiqi@umich.edu | Mobile: 7348827730

Qualifications Summary

Highly motivated and eager to learn new technologies and develop better solutions with comprehensive knowledge of compiler and Architecture. Committed to staying current with the latest technologies and trends to provide insight to projects.

Education

University of Michigan Ann Arbor, Michigan
Bachelor of Science in Computer Science | 3.9/4.0 *Sep 2020 – Apr 2023*

Professional & Research Experience

Undergraduate Research Assistant Apr 2022 – Present
University of California, Berkeley *Remote, in Uclid5 group*

- Improved the precision of syntax error reporting in the Uclid-5 parser by upgrading from the original Scala parser library, "PackratParsers," with a custom-built grammar for error syntax
- Constructed a semantic-level rewrite grammar to allow for the use of identical identifier names in different scopes, providing greater flexibility for code writing
- Assisting in refining the tutorial of Uclid5's floating-point support to ensure clarity and ease of understanding
- Engaging in Formal Verification of the DAWG Cache Architecture, Involving Division Policy of cache

Additional Projects

Code Size reduction Compiler Source Code

- Developed an Improved Implementation of FMSA (Function Merging by Sequence Alignment) to Decrease Code Size in LLVM, Achieved by Merging Similar Functions and Optimizing Memory Utilization on Resource-Constrained Platforms
- Achieved a 21% increase in compilation speed compared to the best configuration of the original FMSA, with only a minimal 0.5% decrease in code size reduction

Decaf Compiler Source Code

- Implemented Lexical Analysis with Flex and Regular Expressions for Effective Text Processing
- Conducted Syntax Analysis on Generated Tokens and Constructed Corresponding Abstract Syntax Tree (AST) using Yacc and C++
- Achieved a 1.8x Time Speed-Up in Code Execution by Generating MIPS Code in the Backend, Implementing Chaitin's Register Allocation Algorithm, and Utilizing Control Flow Profile Information for Instruction Hoisting

A 2-way Superscalar R10K Out-of-Order Processor Source Code

- Collaborated with teammates to synthesize and devise a 2-way R10K out-of-order execution processor with System Verilog (VCS). Managed executing stage, reservation station, system integration, and LSQ
- Advanced features: 2 Way Superscalar, LSQ with speculating on load dependencies and forwarding, Hardware Prefetch,4-Way Associative L1 Cache, Branch predictor, Visual Debugger, Automated regression testing

Accelerating Ultrasound Beamforming on the Xeon Phi Source Code

- Achieved a 100x speedup over the baseline single-threaded performance on the Xeon Phi
- Implemented Project with AVX-512 Instructions and Pthreads (POSIX Threads) for Enhanced Performance and Scalability

Technical Skills

Skills	:	LLVM, SIMD, Bison, Flex, Intel Intrinsics
Dev Tools	:	Visual Studio Code, Git, Gitlab
Programming Languages	:	C++, Java, Scala, SystemVerilog , Ocaml, Python