Zeren Li

li3103@purdue.edu|765-775-3389|🌐 lizerengeorge.com| in linkedin.com/in/zeren-li-george | github.com/lizeren

Education

Purdue University – West Lafayette, IN

Ph.D. in Electrical and Computer Engineering *(Advised by Santiago Torres-Arias)* Aug 2022 – May 2028

Master of Science in Electrical and Computer Engineering Aug 2022 – Dec 2024

Bachelor of Science in Computer Engineering Aug 2018 – May 2022

Experience

Microprocessor Systems and Interfacing Teaching Assistant, Purdue University Jan 2023 – Current

- Oversaw lab sessions for a Microprocessor Systems and Interfacing course, guiding 60+ students each semester.
- Guided students on **ARM/RISC-V** assembly programming, **STM32/Raspberry Pi Pico** microcontroller interfacing, and digital system design.

Publications

Can SOLARBURST go silicon? A systematic analysis on software supply chain attacks for Open Source VLSI design Unpublished Manuscript

Fabiha Hashmat, ***Zeren Li***, Santiago Torres-Arias

Projects

Multicore Processor Design and Prototyping Spring 2023

- Designed and implemented a **single-cycle** CPU that supports the MIPS instruction set and operates at **35 MHz** on an FPGA board.
- Designed and implemented a **60 MHz dual-core** MIPS CPU with a five-stage pipeline on an FPGA board. Features include hazard handling, L1 instruction and L1 data caches, **MSI-based cache coherence**, LL/SC instructions for synchronization, and **branch prediction** (2-bit predictor with branch target buffer), reducing control hazards by **15%**.

Source-to-Binary Comparative Analysis Tool Spring 2025

- Built a static analysis framework bridging C/C++ source code and decompiled binaries using **Clang libtooling AST** and **Ghidra**. Developed a Python/Tkinter GUI for function attributes comparative visualization.
- Engineered a parallel C++ symbol demangler, reducing processing time on a large VLSI open-source software from 20 mins to 1 min (**20× speedup**).

CUDA kernel Implementation & Optimization for Deep Learning Kernels Spring 2024

- Implemented and compared **CPU**, **baseline CUDA**, and progressively **optimized CUDA kernels** for convolution, MaxPooling, and GEMM layers, applying **shared memory tiling**, **bank-conflict avoidance**, and **memory coalescing** to improve performance.
- Built and optimized **AlexNet in CUDA** by integrating **shared memory** for filters/activations, restructuring data layouts to avoid **bank conflicts**, and applying **memory coalescing** to boost throughput.

CPU Cache Architecture with Gem5: Z-cache and Associativity Fall 2023

- Implemented and evaluated Z-cache in the Gem5 simulator, focusing on L2 cache misses across the SPEC2017 benchmark suite.
- Conducted comparative analysis of cache management techniques (4-way associative, skewed-associative, and Z-cache) on the TimingSimple CPU model to reduce architectural complexity.

USB Endpoint & AHB-Lite Module Fall 2022

- Designed RTL diagrams for UART Receiver, APB-Slave, USB receiver, and AHB-Lite modules. Implemented, integrated, and developed verification test benches for for APB-Slave, UART Receiver, and USB receiver with the AHB-Lite module using System Verilog.

Training Deep Learning Network on Edge Devices Spring 2025

- Implemented and evaluated **structured/unstructured pruning**, **quantization** using **PyTorch** to accelerate deep learning training on edge devices (Raspberry Pi 4).
- Achieved up to **29% faster training per epoch** with structured pruning + removal while maintaining competitive accuracy; analyzed trade-offs between efficiency and model expressiveness