Saber Ganjisaffar

(408)594-5884 | sganj003@ucr.edu | archimantix.com/about | archimantix | saber-ganjisaffar

Education

Ph.D. Candidate in Computer Science - University of California, Riverside (UCR) *California, US*

GPA: 3.62/4 *09/2022 - 09/2027 (Expected)*

Thesis: Hardware-Software Co-Design for Performance and Security

M.S. in Computer Systems Architecture - Shahid Beheshti University *Tehran, Iran*

GPA: 3.77/4 *2018 - 2021*

Thesis: A Novel Hardware Cache Coherence Mechanism for GPU-Based Near Memory Architectures

B.S. in Computer Science - Semnan University *Semnan, Iran*

Last Two Years GPA: 3.42/4 *2012 - 2017*

Project: Congestion Management and Simulation for Network on Chip (NoC) Systems

Research Interests

Computer Architecture and Microarchitecture
Heterogeneous Systems Architecture, including GPUs and Specialized Accelerators
CPU Performance Optimization Mechanisms (Branch Prediction, Caching, Prefetching)
Cache Coherence and Memory Consistency
Microarchitectural Support for Security (Transient Execution Attacks, RowHammer)
Simulation Frameworks for Architecture/Microarchitecture Modeling

Publications

SpecASan: Mitigating Transient Execution Attacks Using Speculative Address Sanitization. *ISCA 2025*

***S. Ganjisaffar**, E. Mohammadian Koruyeh, J. Zellmer, H. Asghari-Esfeden, C. Song, N. Abu-Ghazaleh.*

(https://doi.org/10.1145/3695053.3731119) *20 June 2025*

PRACtical: Subarray-Level Counter Update and Bank-Level Recovery Isolation for Efficient PRAC Rowhammer Mitigation. *arXiv 2025*

*R. Nazaraliyev, **S. Ganjisaffar**, N. Nazaraliyev, N. Abu-Ghazaleh. (https://arxiv.org/pdf/2507.18581)* *24 July 2025*

Projects

CPU Performance Analysis: A Framework for Evaluating CPU Data Prefetchers via Memory Access Pattern Analysis *2025-Present*

DynamoRIO, LLVM, Scarab, Python

- I'm currently developing a comprehensive, automated framework in Python to analyze program memory behavior and evaluate the efficency of hardware prefetching strategies. The pipeline integrates industry-standard tools, including LLVM for static analysis and DynamoRIO for dynamic binary instrumentation to capture detailed memory access traces. These traces are then processed through a custom, trace-driven CPU architectural simulator (Scarab) to model a modern memory hierarchy and systematically identify the primary sources of memory latency. The framework concludes with a high-level analysis that correlates performance-limiting cache misses with recurring memory access patterns, producing structured data to guide the design of more efficient prefetching hardware.

GPU Architecture Modeling: Design of Per-SM Hardware Performance Monitoring Units (PMUs) and On-Chip Kernel Validation for GPUs *2025*

GPGPU-Sim, CACTI, Verilog

- This project introduces a novel hardware architecture for on-chip GPU kernel validation, providing a low-latency alternative to software-based methods that are orders of magnitude slower. The architectural extention was implemented and evaluated on GPGPU-Sim cycle-accurate simulator, while its hardware complexity and overhead were modeled using CACTI and a Verilog implementation of the Validator module. The design features two core components: a set of lightweight, per-SM Performance Monitoring Units (PMUs), inspired by the SMPC units in modern NVIDIA GPUs, and a dedicated on-chip Validator module. This Validator aggregates performance data and continuously compares it against a pre-loaded "golden model" of benign execution, providing a robust defense against anomalous behavior.

CPU Architectural Modeling: Extending Gem5 Simulator with ARM Memory Tagging Extension (MTE) Support *2023-Present*

Gem5, ARM ISA, LLVM

- I extended the gem5 simulator with support for the ARM Memory Tagging Extension (MTE), creating a robust platform for memory safety research. This required ARM ISA extension and modifying gem5's ARM O3 CPU model and memory subsystem to natively handle MTE instructions like STG/LDG for tag management. I implemented tag-checking logic within the caches and memory controller and extended the cache coherence protocol to ensure tag consistency, enabling the accurate simulation of MTE-protected multi-threaded workloads and compatibility with existing software toolchains for ARM MTE such as LLVM memTagSanitizer and Scudo hardened allocator.

GPU/NMP Architecture Modeling: A Simulator for Architectures Leveraging GPU Cores for Near-Memory Processing *2021-2022*

GPGPU-Sim, DRAMSim3

- During my research on Near-Memory Processing (NMP) architectures utilizing GPU cores as processing units, I identified a gap in available simulation tools for this type of architecture. My goal was to model a heterogeneous system, with a primary GPU on one side and smaller GPU cores integrated into the main memory, such as a 3D-stacked memory like HBM, on the other side. To achieve this, I integrated GPGPU-Sim to simulate the GPU cores and DRAMSim3 to model the 3D-stacked memory system.

GPU/NMP Architecture Modeling: MGPUSim Support for Near-Memory Processing *2020-2021*

MGPUSim

- As a central component of my Master's thesis research, I designed and implemented a novel cache coherence mechanism tailored for Near-Memory Processing (NMP) architectures that utilize GPU cores. I leveraged the high-performance MGPUSim simulator, modifying its modular architecture to integrate and rigorously test the custom protocol. This work involved detailed architectural modifications within the simulator to validate the mechanism's correctness and evaluate its performance under various workloads.

Hardware Design: Design of a Genetic Algorithm Processor *2019*

Verilog, Xilinx Vivado, Modelsim

- Due to the heavy computational load of genetic algorithms, they usually take a long time to find the optimum solution. I designed a fast, pipelined, and efficient, Genetic Algorithm Processor (GAP) that can find the optimum solution based on a fitness function defined. I adopted some of the best practices and research ideas at that time for designing this processor. As it is general-purpose and modular, one can easily use it for any fitness function.

Technical Skills

Languages and Frameworks: C/C++, Golang, Python, ARM and Intel X86 Assembly, Verilog, Chisel, SystemC/TLM

Multicore and Parallel Programming: CUDA, OpenCL, OpenMP

Tools and Technologies: Linux, Bash, Git/Github, Xilinx Vivado, Intel Quartus Prime, Intel HLS, ModelSim, LaTex

Architectural Simulators: GPGPU-Sim, MGPUSim, Vortex GPGPU, Gem5, Scarab, DRAMSim3, Ramulator, Verilator, CACTI

Experience

Teaching Assistant, CS 153 - Design of Operating Systems

University of California, Riverside *Fall 2024, Summer 2024*

Teaching Assistant, CS 010A - Introduction to Computer Science for Science, Mathematics, and Engineering 1 (Beginner C++ Programming)

University of California, Riverside *Spring 25, Winter 25, Spring 24*

Teaching Assistant, CS 010B - Introduction to Computer Science for Science, Mathematics, and Engineering 2 (Intermediate C++ Programming)

University of California, Riverside *Winter 24*

Research Assistant, Secure and Efficient Architecture Systems (SEAS) Lab

University of California, Riverside *July 2023 - January 2024*

Research Assistant, Computer Architecture Lab

Institute for Research in Fundamental Sciences (IPM) *July 2021 - January 2022*

Teaching Assistant, Design of Digital Circuits

Shahid Beheshti University *Fall 2020*

Honors and Awards

Dean's Distinguished Fellowship Award.

Awarded by University of California, Riverside (UCR) for Academic year 2022 *March 2022*

Services

Artifact Evaluation Committee Member

International Symposium on Microarchitecture (MICRO) 2025 — *August 2025*

Peer Reviewer

Integration, the VLSI Journal (Elsevier) — *July 2025 - Present*

Conference Organizing Team (Student Volunteer)

International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2024 — *April 2024*

Certificates

Deep Learning Specialization

Coursera

DeepLearning.AI — *December 2022*

https://www.credly.com/badges/544e29b5-28ac-4bc4-9218-9402f77510ee