Kaushik Nikhil Shroff

kaushik.shroff2@gmail.com • +1(240) 598-9713 • linkedin.com/in/kaushik-shroff/ • Madison, WI

EDUCATION

University of Wisconsin-Madison, Madison, Wisconsin Sep 2024 – Dec 2025
Master of Science, Electrical and Computer Engineering GPA: 3.5/4.0
Coursework: Advanced Computer Architecture, VLSI Systems Design, Testing and Testable Design of Digital Systems

Savitribai Phule Pune University, Pune, India Jul 2020 – Jun 2024
Bachelor of Engineering, Electronics and Telecommunication GPA: 3.8/4.0
Coursework: Digital Circuits, Microcontrollers, VLSI

SKILLS

HDL: Verilog, System Verilog, VHDL
Languages: Python, C++, C, SLICC, Bash, Tcl
Tools: gem5, McPAT, CACTI, Design Compiler, ModelSim, Innovus, PrimeTime, Virtuoso, TetraMax, Git

ACADEMIC PROJECTS

Out Of Order CPU Power Modeling in gem5

Tools: gem5, McPAT, CACTI, Python, C++

- Built a **power model** for gem5's **Out Of Order CPU**, enabling power-performance analysis of out-of-order pipelines.
- Mapped **20+ micro-architectural components** (e.g., ROB, branch predictor, LSQ) to activation statistics in gem5.
- Extracted **per component energy** using McPAT, improving the modeling accuracy for ALU, caches, and branch units.
- **Automated energy computation** via Python scripts to parse statistics, producing component-wise power with **98% accuracy**.
- Validated model by **benchmarking 7 workloads**, observing 2× higher power for integer-intensive code.
- Extended gem5 by **adding 5+ custom stat counters** and integrated support for segment-based power analysis.

MUSI Cache Coherence Protocol Implementation in gem5

Tools: gem5, SLICC, C++, Python

- Implemented the COUP concept by extending gem5's MSI protocol into a **MUSI (Modified–Update–Shared–Invalid)** variant.
- Added **5+ new coherence messages and 10+ state transitions** across cache and directory controllers using SLICC.
- Designed and integrated a **new state of update (U)** to support commutative updates, enabling local aggregation.
- Built **custom C++ micro-benchmarks** to test COUP behavior under conflicting shared-memory access patterns.
- Identified and resolved protocol-level integration issues by instrumenting **fine-grained debug tracing** in cache hierarchy.
- Verified **syntactic and build correctness** of protocol extension.

Graph Neural Network (GNN) Accelerator Design

Tools: System Verilog, ModelSim, Design Compiler, Innovus, PrimeTime, Virtuoso, Tcl

- Designed a **GNN accelerator** with MAC, ReLU, and aggregation units for 4-task scheduling on heterogeneous Arm cores
- Achieved 869 MHz max frequency, 3.07 mW power, and **7.91 pJ·ns·mm² EDAP** through optimization techniques including **clock gating, bit-width pruning, and modular pipelining**.
- Synthesized, placed, and routed using 7nm standard-cell libraries, followed by post-layout **power and timing analysis**.
- **Reduced post-APR power by 5% and pre-APR by 38% using clock gating**, demonstrating hardware-aware optimization.
- Completed **full frontend-to-backend development**, from RTL design to physical layout and post-silicon estimation, including verification, synthesis, APR and power/timing closure

Scan-Based Test Generation for Digital Systems

Tools: Synopsys Design Vision, TetraMax, Tcl, VHDL

- Generated ATPG test sets for full and partial scan designs of a Viper processor (ITC'99 b14 benchmark), achieving **99.48% test coverage** with full scan.
- Designed and synthesized partial scan architectures using **SCOAP-based flip-flop ranking**, optimizing for test coverage, area, and test time.
- Conducted design space exploration across 17 configurations, achieving a **122% improvement** in test efficiency over full scan.
- **Automated** scan chain generation, test compression, and metric evaluation using custom **Tcl scripting** and analysis workflows.