

# KARTHIKEYA SHARMA M

100, 10th ST NW, Atlanta, GA, 30309

☎ 404-203-4923

✉ [sharmakarthikeya6@gmail.com](mailto:sharmakarthikeya6@gmail.com)

🌐 [linkedin.com/in/karthikeyasharma16](https://www.linkedin.com/in/karthikeyasharma16)

🔗 [KarthikeyaSharma16](https://github.com/KarthikeyaSharma16)

## Education

### Georgia Institute of Technology

08/2023 – 05/2025 (Expected)

*Master of Science in Electrical and Computer Engineering*

GPA: 3.75/4

### SRM Institute of Science and Technology

07/2019 – 05/2023

*Bachelor of Technology in Electronics and Communication Engineering*

GPA: 9.83/10

## Technical Skills

**Coding:** C/C++, CUDA C, Scripting Language (Python, Bash, TCL), RTL (Verilog, SystemVerilog), High-Level Synthesis

**Libraries:** OpenGL, OpenMP, OpenMPI, SFML, PyTorch, Keras, CuPy

**Operating Systems:** Linux (Ubuntu, Redhat), Windows

**Open-source Simulators:** ScaleSim, TACOS, Chakra, ASTRA-Sim, ScaleHLS

**EDA Tools:** Xilinx Vivado, Vitis HLS, Modelsim, Quartus Prime, Icarus Verilog, Libero 12.5, TAPA, ALLO, Google XLS, Softconsole, Code Composer Studio

**Protocols:** I2C, SPI, CAN, UART, AXI3, APB, CHI (basics), Ethernet, PCIe (basics)

**Coursework:** Advanced Computer Architecture, Hardware-Software Co-Design for ML Systems, Parallel Programming for FPGA, Advanced Programming Techniques, Machine Learning, Digital Systems Test, ARM Based Embedded System Design, Introduction to Embedded Software Engineering, Digital Logic Design, Digital VLSI Design, Analog VLSI Design, CMOS IC Design, FPGA design for Embedded Systems, Digital Signal Processing, Deep Learning Applications for Computer Vision.

**Technical Concepts:** RTL design, Verification (w/ UVM basics), CPU microarchitecture, Interconnection networks (Flow control, Routing, Topology, Router Design), Cache Coherency, AI Accelerators, ASIC & FPGA design flow, OOP principles, System Optimizations for distributed ML, Benchmarking methodologies, DFT (Scan Chains, LFSRs, Fault Simulation & ATPG algorithms).

## Work Experience

### Research Assistant

04/2024 – Present

*Synergy Lab @Georgia Tech*

Atlanta, GA

- Prototyped **APEX**, an automated toolchain for analyzing network configurations and collective algorithms for ML algorithms via Design Space Exploration (DSE), revealing 64% of the workload bandwidth and topology dependent.
- Streamlined collective algorithm synthesis by translating TACOS outputs to MSCCLang IR and Chakra Execution Traces, bypassing ASTRA-Sim's system layer reducing engineering effort by 10×.
- Created a configuration file for heterogeneous topology translation and an interactive visualizer for analysis of GPU network collective flows and patterns for congestion and switch modeling.

### Graduate Research Assistant

01/2024 – Present

*SHARC Lab @Georgia Tech*

Atlanta, GA

- Developed Cryptonite, a toolchain for generating **hundreds of correct-by-design** HLS designs for FPGA-based accelerators from verified straight-line C implementations of cryptographic primitives.
- Built a DSE engine and analytical model for performance analysis, enabling scalable designs with up to 88.88% resource utilization reduction and 54.31% latency improvement.
- Used LLVM-Clang compiler libraries to generate flexible HLS designs through graph mining algorithms.

### Student Assistant

09/2023 – Present

*iSenSys Lab@Georgia Tech*

Atlanta, GA

- Developed a real-time frequency data acquisition system using SmartFusion2 SoC FPGA to control a closed-loop gas sensing system, reduced on-board power consumption by 42% by implementing custom power saving modes.
- Translated analog hammerhead resonator signals into frequency data by creating custom IP blocks in Verilog, interfaced capacitive sensors and implementation of ML models on FPGA using HLS.

## Projects

### Efficient Implementation of Attention Computation | *HLS, C++*

02/2025

- Optimized a scaled dot-product attention mechanism in HLS (110× speedup), used AXI burst mode and ping-pong buffering to overlap computation & data transfer, maximizing DRAM bandwidth of the FPGA SoC.

### GPU-Accelerated Framework for Hyperspectral Raman Imaging | *Python*

08/2024 - 12/2024

- Achieved 10× speedup for a pipelined architecture to stream multi-dimensional spectral data, enabling efficient vectorized signal processing transformations and automated metrics for SVD vector selection.

### Systolic Array Accelerator | *Verilog, Python*


06/2024 - 08/2024

- Built a 4x4 Systolic Array (@1 GHz), with MAC modules supporting pipelined FP32 addition and multiplication, achieving 2 operations per cycle per MAC.
- Supported sparse representations (CSR, CSC), weight stationary dataflow & access latency through AXI3 bursts.

### Federated training and profiling of large-scale ML model | *Python*

03/2024 - 04/2024

- Implemented single and distributed GPU training (2 GPUs) of a GPT-like model on an NVIDIA RTX 6000 cluster using Megatron-LM, utilizing tensor, data, and pipeline parallelism.
- Profiled training with TensorBoard showing tensor parallelism achieving 1.84× speedup over pipeline parallelism.

- Context-aware Deepfake Detection | *Python, PyTorch*** **02/2024 - 05/2024**
- Created custom dataset from YouTube & Reddit of 1298 videos to provide contextual information for HMCAN model.
  - Cleaned metadata for better tokenization and feature extraction achieving training accuracy of 91.7%.
- Performance-Aware Accelerator Design for ML workloads | *Python*** **02/2024 - 03/2024**
- Conducted DSE using Scale-Sim to optimize systolic array configurations for LeNet CNN, achieving 57% average mapping efficiency for operators while evaluating performance trade-offs across dataflow configurations.
  - Developed a mapping strategy (mapping efficiency 84.375%) to mitigate faulty processing elements in a weight-stationary 3x3 systolic array, optimizing GEMM computation latency (54 cycles).
- DUT Verification Framework | *SystemVerilog*** **12/2023 - 01/2024**
- Formulated verification plans for evaluating quality of handwritten RTL designs such as MAC, Circular FIFO, SPI, UART, I2C, and AXI3 memory through a hierarchical testbench architecture achieving 100% functional coverage.
- Multithreaded 3D Scene Animation | *C++*** **11/2023 - 12/2023**
- Created an interactive 3D Halloween themed scene using OpenGL, incorporating multiple objects to create scene depth.
  - Utilized OpenMP to enable independent movement, rotation, and realistic collisions of 5 animated objects.
- CUDA-accelerated Random Walk Simulation | *CUDA C*** **11/2023 - 12/2023**
- Implemented CUDA-accelerated random walk with 100,000 walkers and 1 million steps, observed  $420\times$  execution time increase for large workloads and 5-7% better performance with pinned/managed memory at scale.
- MIPS CPU and CMP memory simulator | *C++*** **09/2023 - 12/2023**
- Developed a 5-stage (8-pipe) MIPS CPU simulator with Out-of-Order execution, GShare Branch Prediction, and optimized cache hierarchy, reducing CPI by 50% and latency by 10%.
  - Engineered a dual-core simulator with L1, L2-shared caches (LRU & random eviction) and DRAM, optimized L2 utilization via SWP & DWP for a 0.85% performance gain.
- FaultForge-Sim for Post-Silicon Validation | *C++* ** **08/2023 - 12/2023**
- Built a logic simulator for parsing netlist enabling fault-free evaluation. Applied PODEM ATPG algorithm for test vector generation and achieved 99.71% fault coverage using deductive fault simulation.