

# RAGHUL SRINIVASAN

📞 860-962-1742   ✉ [rsriniv7@ncsu.edu](mailto:rsriniv7@ncsu.edu)   🔗 [linkedin.com/in/raghulsrini](https://www.linkedin.com/in/raghulsrini)   🐙 [github.com/sraghul97](https://github.com/sraghul97)   📍 Wilmington, MA

## EDUCATION

### North Carolina State University

May 2024

*Master of Science in Computer Engineering (GPA: 3.96 / 4.00)*

*Raleigh, NC*

- **Relevant Coursework:** CPU & GPU Architecture, ASIC & FPGA Design using Verilog, Design Verification using System Verilog & UVM

### SASTRA Deemed University

June 2018

*Bachelor of Technology in Electronics and Communication Engineering (GPA: 8.28 / 10.00)*

*Thanjavur, India*

## TECHNICAL SKILLS

**Languages** : C/C++, Python, Verilog HDL, System Verilog, x86/RISC Assembly, VisualBasic

**Frameworks** : Unified Verification Methodology (UVM), Nvidia CUDA, POSIX Threads

**Tools** : GPGPU-Sim, Synopsys Design Compiler, Synopsys Verdi Analyzer, Vivado, Mentor Graphics Questa, Git

## EXPERIENCE

### Analog Devices Inc.

June 2024 – March 2025

*Post-silicon Verification Engineer*

*Wilmington, MA*

- Worked closely with the digital design and verification teams to develop test plans for validating IP Cores.
- Collaborated with verification engineers to re-create post-silicon bugs in the pre-silicon test bench.
- Developed tests to identify power consumption discrepancies across blocks & collaborated with RTL engineers to debug clock gating issues.
- Conducted silicon characterization to assess process variations, measuring timing, and performance metrics to validate design robustness.

### Analog Devices Inc.

May 2023 – August 2023

*Test Engineering Intern*

*Wilmington, MA*

- Conducted level and timing characterization on GPIO & SPI interfaces in a multi-band RF receiver at the post-silicon level.
- Verified setup and hold time violations of the silicon with an accuracy of 0.01%.

### Syrma SGS Technology Ltd.

July 2018 – July 2022

*Sr. Test Development Engineer*

*Chennai, India*

- Developed automated test software for a high-volume 4-up End-of-Line tester, improving throughput.
- Employed communication protocols like RS485, I2C, SPI, UART, Zigbee, USB to interface with modules and sensors.

## PROJECTS

### ASIC Multi-Stage Neural Network Accelerator | Verilog

- Developed a hardware module for deep learning operations, enabling multi-layer feature extraction on input images stored in DRAM.
- Achieved a synthesizable design with optimal performance per area using Finite State Machine (FSM).

### Functional Verification of I2C Multiple Bus Controller | System Verilog

- Developed a layered testbench in System Verilog to test I2CMB functionalities with components including environment, generator, agent, driver, monitor, predictor, and scoreboard.
- Enabled test plan links & coverage collection through random/direct tests, SV assertions & cover groups.

### Verification of LC3 Processor | UVM

- Developed interface packages, and implemented Bus Functional Model logics for driver, monitor and scoreboard components for IP level verification of Decode stage in a five-stage pipeline in a RISC machine.
- Implemented system-level environment comprising of sub-environments of all 5 stages for SoC verification.

### Hybrid Branch Predictor | C++

- Implemented a hybrid branch predictor that selects between gshare and bimodal configurations, using a chooser table of  $2^k$  2-bit counters.
- Compared design trade-offs in terms of misprediction rates using SPEC's Benchmark trace files with 2 million traces.

### Functional modeling of Cache and Memory Hierarchy with Prefetch Stream Buffers | C++

- Designed a user-configurable WBWA multi-level cache and memory hierarchy simulator with prefetch having LRU replacement policy.
- Measured memory performance using parameters such as miss rate, average access time, and memory traffic by varying cache parameters.

### Bus Based Cache Coherence on Shared Multiprocessor System | C++

- Implemented MSI and MESI coherence protocols in a multi-core environment with distributed L1 cache.
- Designed a snoop filter, reducing 32% of cache tag lookups by tracking invalidated blocks and failed tag accesses.

### Super-scalar 9 Stage Out Of Order Processor Simulator | C++

- Developed a functional model of a pipelined processor implementing the Tomasulo algorithm, fetching and issuing N instructions per cycle.
- Analyzed the IPCs for various combinations of super-scalar widths, Issue Queue size and re-order buffer sizes.

### Check Point Processing Recovery (CPR) Implementation | C++

- Implemented coarse-grain retirement and aggressive register reclamation to establish an expansive instruction window size with a smaller PRF.
- The CPR approach demonstrates higher IPC efficiency with a reduced PRF size, with negligible impact observed with further PRF size increases.

### Acceleration of quantum gate simulation using GPU | CUDA

- Developed simulation of multi-qubit gates using CUDA within the GPGPUsim framework to achieve data parallelism.
- Used shared memory and thread coarsening optimization techniques to enhance performance by 85%.