Sanket Patil

Portland, OR   +1 971-427-8865   sanketpatilpdx@gmail.com   linkedin   Github

Education

Portland State University, Portland, OR, United States Sept 2022 - June 2024

Master of Science- Electrical and Computer Engineering ***GPA: 3.96 / 4.00***

- **Relevant Coursework:** Microprocessor System Design, Computer Architecture, System Verilog, Pre-Silicon Verification, ASIC Modeling and Synthesis, Formal Verification, Computational Tools

K J Somaiya College of Engineering, Mumbai, India Aug 2018 - May 2022

Bachelor of Technology- Electronics and Telecommunication Engineering ***GPA: 9.36 / 10.00***

Technical Skills

Programming and Scripting languages: Verilog HDL, System Verilog, C/C++, UVM, Python, Linux, Assembly Language
Concepts: RTL Design, Cache, Pipeline, Assertions, Constraints, Coverage, DFT, RISC-V, Formal Verification, Object Oriented Programming (OOPs), UVM, DDR, STA, Synthesis, ASIC, SOC, Microarchitecture, Debugging, Problem solving
Computer-based Tools: QuestaSim, ModelSIM, VCS, LtSpice, iVerilog, GTKWave, Quartus Altera, EDA, MATLAB
Operating Systems: Windows, Linux

Experience

AMD, Austin, TX, United States Aug 2023 – May 2024

CPU Post Silicon Validation Intern

- Developed Python scripts to validate Memory Interconnect of server socket, reducing debug time by around 50% and improving server CPU validation efficiency.
- Optimized test case re-execution strategy, improving root cause analysis speed, leading to quicker issue resolution.
- YML script generation for 4+ workloads automation, reducing manual workload by 50%.
- Analyzed and visualized test results using Power BI & MS Excel, presenting insights to the debug team, helping identify 5+ critical issues.

Projects

Design and Verification of synchronous FIFO as circular buffer *(System Verilog, OOPs, UVM)*

- Developed a synthesizable RTL design for a synchronous FIFO module and verified its correctness using class-based and UVM-based testbenches.
- Applied constrained random stimulus, functional coverage, and assertion-based verification to ensure design reliability with the coverage of 98% and scoreboard for functional checking.

Last Level Cache Simulation *(C++)*

- Designed and simulated a set-associative, write-back Last Level Cache (LLC) with MESI coherence protocol, pseudo-LRU replacement policy and cache coherence debugging.
- Optimized cache configurations dynamically through command-line parameters, achieving a 97% hit ratio in simulation tests.

Verification of sequence detector FSM using Formal Verification tools *(VC Formal, System Verilog)*

- Designed a sequence detector and created TCL scripts for comprehensive formal verification.
- Modified TCL scripts to explore different formal verification modes.
- Wrote assertions and utilized FPV mode to validate correctness in VC Formal with 100% assertion pass.

Dynamic Instruction Scheduling Simulator *(C++)*

- Built a simulator in C++ for an Out of Order superscalar processor that fetches instructions and passes it through a 9-stage pipeline, handling data hazards before storing it into Architectural Register File
- Determined minimum size of Reorder Buffer and Issue Queue for a given trace and given design of a superscalar processor

RISC-V ISA Simulator *(C++, Assembly Language)*

- Designed and implemented an RV32I Base Integer Instruction Set simulator, supporting 40+ instructions across R, I, S, B, U, and J formats.
- Verified correct instruction execution and register updates, ensuring ISA compliance.