ASHISH AJAYAKUMAR

+1 9197915842 | ashishajay.mec@gmail.com | Raleigh, NC, USA

EDUCATION

Master of Science | Computer Engineering *(08/2022 - 05/2024)*
North Carolina State University, Raleigh, USA GPA: 3.62/4.0
Coursework: Microprocessor Architecture, Advanced Microarchitecture, Architecture of Parallel Computers, ASIC Verification with System Verilog, Advanced Functional Verification with UVM, ASIC & FPGA Design with Verilog, IoT Architecture, Neural Network.

Bachelor of Technology | Electronics and Communication Engineering *(07/2014 - 05/2018)*
Model Engineering College, India GPA: 7.89/10.0
Coursework: VLSI Design, Microprocessors, Digital Signal Processing, Digital Electronics & Logic Design, Embedded Systems.

SKILLS

Languages: System Verilog, Verilog, Python, Tcl, C, C++, Perl.
Tools/Frameworks: Verdi, Modelsim, QuestaSim, Synopsys VCS, Xilinx Vivado, Linux/Unix tools, Git.

WORK EXPERIENCE

Research Assistant | NC State University | NC, USA *(08/2024 - 02/2025)*

- Designed and developed AXI, SPI, I2C and UART protocol modules using Verilog, referring to their architecture and specifications.
- Developed testplans and UVM-based test bench generating constraint-random and directed test sequences to verify their functionality.
- The package includes agents, drivers, monitors, environment, configuration, predictors, coverage monitors, and scoreboard classes.
- Implemented coverage models containing coverpoints and cross-coverage ensuring a 100% bin hit rate across all test stimuli.
- **ASIC Verification Engineer | Maven Silicon | Remote, NC, USA** *(02/2024 - 07/2024)*
- Verified Router 1x3 developing UVM-based test bench environment writing constrained random test sequences & analyzing coverage.
- Enhanced testbench with virtual sequences, pipelined drivers, out-of-order comparators, scoreboards, and cover groups.
- Developed an AHB to APB Bridge architecture design using Verilog and conducted functional testing through simulation on ModelSim.
- Synthesized the design to ensure it meets the required timing and area constraints using Synopsys Design Compiler.

Server Validation Engineer | Internship | Advanced Micro Devices (AMD) | Austin, Texas, USA *(08/2023 - 05/2023)*

- Developed a scripting tool using Python to automate and execute regressive reset tests on server platforms for multiple reset scenarios.
- Integrated error handling and logging mechanisms, improving the tool's robustness and simplifying the debugging process.
- Spearheaded unit testing initiatives on AMD Server products, meticulously debugging failures, managing bug tracking & close coverage.
- Captured and analyzed debug logs, collaborated with the firmware team to resolve critical device bugs, and managed issue tickets.
- Verified and maintained systems in the lab, executing and supporting post-silicon validation activities.

Senior Engineer | System Validation Team | Infineon Technologies | Bangalore, India *(09/2018 – 08/2022)*

- Developed automation test scripts using Tcl to validate the device drivers and achieving a 15% improvement in time efficiency.
- Spearheaded development of comprehensive test plans for testing firmware of PCIe and SDIO Wi-Fi cards used in automotive devices.
- Designed and developed python scripts to automate project workflows enhancing efficiency in data processing and reporting.
- Led creation of Android test plan over multiple versions and modified the automation test suite to ensure cross-platform compatibility.
- Mentored 4 engineers by delivering training on Wi-Fi concepts and features, enhancing their understanding of Wi-Fi features in project.

PROJECTS

Functional Verification of LC3 Microcontroller with UVM | SystemVerilog, OOP

- Designed and implemented hierarchical re-usable UVM testbench for chip-level verification of LC3 Microcontroller Architecture.
- The package included environment, sub-environments, agents, drivers, monitors, BFMs, sequences, predictors, and scoreboards.
- Developed verification test plan for block/chip verification, constrained random tests, and direct tests for full coverage.

Functional Verification of I2C Multi-Bus Controller in System Verilog | SystemVerilog, OOP

- Built a layered reusable testbench to verify the ability of the DUT to control multiple I2C buses through a Wishbone master.
- Test bench environment comprised generator, driver, monitor, predictor, scoreboard, and agent classes to emulate and analyze the DUT.
- Achieved full functional and code coverage developing test plan consisting of cover groups, directed & random tests, and assertions.

Nine Stage Out of Order (OOO) Superscalar Pipeline Simulator | C++, Performance Modeling

- Developed a cycle-accurate simulation model of a dynamically scheduled processor that issues multiple instructions per cycle.
- Managed data hazard and dependencies via Reorder Buffer(ROB), Rename Map Table(RMT), and multilevel data bypassing.
- Analyzed clock cycle timing and the impact of varying the window size, ROB size, and Issue Queue size on the machine's IPC.

Deep Neural Network Hardware using RTL | Verilog

- Designed and implemented RTL multistage neural network hardware to perform convolution, max pooling & matrix multiplication.
- Developed an FSM to fetch the dimension of input, weight, and kernel matrix from the interfaced SRAM and perform computations.
- Implemented pipelining in the NN architecture to allow concurrent execution of convolution, pooling, and activation operations.

Multi-level Cache Simulator | C++, Performance Modeling

- Simulated an L1-L2 n-way set associative cache & memory hierarchy system with an N-Next-Line Prefetcher (Stream buffer).
- Performed architectural trade-off analysis, simulating by using various cache configurations - cache size, block size, and associativity.
- Evaluated the impact on Average Access Time (AAT), hit time, miss rate, and miss penalty, with varying cache configurations

Muti-Core Cache Coherency Protocol Simulator | C++

- Simulated the cache coherence protocols MSI, and MESI on a shared multiprocessor with each processor having a private L1 cache.
- Implemented a snoop filter to track and differentiate between useful and wasted cache lookups in the MESI protocol.
- Conducted performance analysis of protocols under different cache sizes and associativity, ensuring accurate data consistency.