Mathura Swaminathan

984-895-9340 | mathura.swaminathan@gmail.com | www.linkedin.com/in/mswamin2/

EDUCATION

North Carolina State University, Raleigh, NC, USA *Jan 2022 – Dec 2023*
Master of Science in Computer Engineering
Coursework: ASIC and FPGA Design with Verilog, Microprocessor Architecture, ASIC Verification, VLSI Systems Design, Architecture of Parallel Computers, Advanced Verification with UVM

Anna University, Chennai, India *Aug 2014–May 2018*
Bachelor of Engineering in Electronics and Communication

TECHNICAL SKILLS

Languages Hardware : Verilog, System Verilog, UVM, VHDL **Software:** C, C++, Python, Regex, Linux
Tools : Visual Studio Code, Simvision, SVN, QuestaSim, ModelSim, Design Vision, Xilinx ISE, Vivado, Git
Computer Architecture : Pipelining, ILP, Out-of-order Execution, Caches, Cache Coherence, AXI Protocol
Verification : OOP, SV Assertions, Constrained Random Testing, Reusable Layered Testbench, Functional Coverage
Design : Digital Design, Static Timing Analysis, Clock Domain Crossing, Power Optimization

PROFESSIONAL EXPERIENCE

Renesas Electronics America Inc. | Design Verification Engineer | Morrisville, NC *Sept 2024 - Mar 2025*

- Developed a good understanding of the SVI3 protocol, PWM phase assignment, power states, output voltage control, and assisted in defining test strategies for verification planning for digital multi-phase controllers.
- Improved functional coverage by implementing targeted SystemVerilog checkers and test cases, for robust verification of critical design features like register and CRC changes, NVM writes, power-on sequence, hardware bypass, and NACK errors.
- Restructured the testbench BFM to align with changes in register map, communication packetization and response mechanism and resolved test failures across multiple modules to attain a 100% regression pass rate and test stability.

DANFOSS | RTL Engineer Intern | Durham, NC *May 2023 – Aug 2023*

- Built Clock Domain Crossing Data capture and Data register, and successfully validated for diverse clock ratios.
- Enhanced the Power Bus Switch with a priority TDM using LUT, port prediction logic, verified the forwarded data, and achieved a 16x boost in transmission speed under worst-case scenarios.

DATA PATTERNS Ltd. | FPGA Design Engineer | Chennai, India *Jul 2018 – Nov 2021*

- Executed digital design and verification at block and system levels, with expertise in debugging and timing corrections.
- Designed intricate digital circuits including FSM, Linear Feedback Shift Register, CDC using FIFO, accumulators, high-speed DMA read-write on memory, interfaces such as PCI Express, Ethernet, UART, SPI, and I2C.

PROJECT EXPERIENCE

Digital Design/Verification

Functional verification of LC3 Processor (System Verilog, UVM)

- Created UVM interface and environment package, partly using UVMF code generator for block-level verification of decode stage and chip-level verification of all 5 pipeline stages in LC3.
- Generated testbench, conducted regression testing and achieved maximum functional coverage.

Functional verification of I2C Multiple Bus (System Verilog)

- Constructed a Reusable Layered Testbench to test I2CMB with test class, environment, generator, agent, driver, monitor, environment, predictor, scoreboard, and coverage in Questasim.
- Verified by Constrained Random Testing to obtain maximum code and functional coverage consisting of Cover groups, Cross coverage, and Assertions as per the test plan.

ASIC Design of Multi-stage Neural Network (Verilog)

- Input data from SRAM processed in CNN, pooling, and activation layers using ModelSim and synthesized with no errors in Design Vision to obtain a minimal Clock period of 6.7ns and Area of 9609um^2. Ranked among 10 best designs for performance.

Computer Architecture

5-Stage MIPS Instruction Pipeline Simulator (C++)

- Implemented a cycle-accurate 5-stage pipeline to execute different instructions for 32-bit integer and floating-point data. Handled structural, control, and data hazards for efficiency and appropriate CPI.

Dynamic Scheduled Processor Simulator (C++)

- Developed cycle-accurate dynamic scheduling using the Tomasulo algorithm. Handled out-of-order instruction execution with Register renaming, Reorder buffer, Load, and Store bypassing for an optimum CPI.

Cache Coherence Simulator (C++, Python)

- Devised bus-based cache coherence protocols MSI, MESI, MOSI in a multi-processor system to maintain coherence across private L1 caches using write-back, write-allocate and LRU replacement policy. Compared performance of the protocols.