DEVESH HITESHBHAI JANI

+1 (919)-520-5927 ⋄ Email: dhjani2@ncsu.edu ⋄ www.linkedin.com/in/deveshjani ⋄ Raleigh, NC

EDUCATION

North Carolina State University Raleigh, NC
M.S. in Computer Engineering May 2026
Coursework: Advance ASIC Verification, Advance Computer Architecture: Data Parallel Processors (GPGPU), Microprocessor Architecture, ASIC and FPGA Design using Verilog, Embedded Systems Architecture.

Charotar University of Science and Technology Anand, India
B.Tech in Electronics & Communication Engineering - **CGPA:** 9.05/10.00 May 2024
Coursework: VLSI Technology and Design, Design, Testing and Verification, Verilog and FPGA Programming.

TECHNICAL SKILLS

Programming	System Verilog, Verilog, C++, C, Python, TCL.
Tools and Software	CUDA, OpenCL, Vivado, Libero SoC/IDE, LabView, MATLAB, Synopsys VCS, Synopsys Verdi, LTSpice.

PROFESSIONAL EXPERIENCE

Indian Space Research Organization Ahmedabad, India
Research Trainee, FPGA and RTL Design Dec 2023 - May 2024

- Designed Moving average filter (Boxcar) utilizing SRAM interfaced with FPGA using Libero IDE 9.1.
- Optimized data synchronization and validation in optical receiver systems by leveraging Finite State Machine (FSM) coding techniques in Verilog.
- Integrated SystemVerilog Assertions (SVA) and Functional Coverage analysis to enhance transmission reliability.
- Worked on Asynchronous FIFO Clock Domain Crossing using Verilog and performed synchronization of clock signals (FSYNC and LSYNC) on PROASIC 3 FPGA.
- Developed a Dual Port FIFO interface and explored BRAM functionality on the ProAsic3E FPGA board, improving performance by 70%.

Tangible Synthesis Pvt Ltd Ahmedabad, India
Project and Test Intern, Embedded Systems/ Robotics May 2023 - June 2023

- Created 50+ test plans to verify subsystem interactions in Warehouse Robot and optimization of controlling the battery outrage and power dissipation with physical parameters.
- Performed ASTM E3426/E3426M-24 standard test method, evaluating robot performance over 1000 cycles of a predefined path.

ACADEMIC PROJECTS

Functional Verification of I2C Multiple Bus Controller (System Verilog, QuestaSim)

- Working on designing a Bus functional model for I2C Slave to verify Wishbone to I2C bridge for I2C Multi bus controller.
- Coding a layered testbench with environments, agents, predictors and scoreboards. Working on Directed and Randomized tests, assertions with cover-groups to collect functional coverage.

Scaled Dot-Product Self-Attention in Transformer Models (Verilog, ModelSim)

- Developed and implemented the self-attention mechanism within a Transformer model by computing scaled dot-product attention, using matrix multiplications for query, key, and value matrices using ModelSim.
- Utilized Verilog to design, verify, and synthesize modules for the Transformer's self-attention operations and optimized processing in memory-mapped SRAM structures and achieved the area of 9k using 2752 cycles with a clock period of 11.2ns.

8-bit Synchronous FIFO (Verilog, ModelSim)

- Developed an 8-bit FIFO design with Read Pointer (RP) and Write Pointer (WP) functionality. Added Almost Full (AF) & Almost Empty (AF) Flag register to infer previous FIFO run state at zero depth detection.

32-bit Arithmetic & Logic Unit (Verilog, ModelSim)

- Designed a 32-bit arithmetic & logic unit (ALU) for signed and unsigned numbers, using Ripple Carry Adder (RCA) slices with unsigned & signed overflow detection and 5 MIPS ISA functions: ADD, SUB, AND, OR & SLT.

RISC-V Superscalar Pipeline Simulator (C, C++)

- Design of a simulator for an out-of-order Super-scalar processor to model dynamic instruction scheduling and implemented a pipelined processor using Tomasulo algorithm that fetches and issues N instructions per cycle, with analyzing it's IPC for various super-scalar widths, scheduler and re-order buffer sizes.

Cache and Memory Hierarchy Simulator (C, C++)

- Developed an L1, and L2 cache simulator in C++ to gain in-depth knowledge of cache architectures and memory hierarchies and utilizing write-back (WB) and write-allocate (WA) techniques, along with LRU (Least Recently Used) replacement policies, for effective cache management.