MUTHYA AMBATI

☎ +1(984)286-4903 ✉ mambati@ncsu.edu in linkedin.com/in/muthya-ambati

OBJECTIVE

Graduate student in Computer Engineering, specializing in Computer Architecture and Digital VLSI with extensive work experience in Verification and Validation, in the Embedded domain. Looking for a rewarding full-time role in SoC development, FPGA and ASIC Design/Verification to gain expertise and significantly contribute in this field.

EDUCATION

North Carolina State University **Raleigh, North Carolina**

Masters in Computer Engineering, GPA:4.0/4.0 August 2022 – May 2024

Coursework: ASIC Verification UVM, ASIC Verilog, Microprocessor Architecture, Embedded Systems, Compiler Optimizations.

Visvesvaraya National Institute of Technology **Nagpur, India**

Bachelors of Technology, Electronics and Communication Engineering, GPA:8.41/10 August 2016 – May 2020

TECHNICAL SKILLS

- **Programming Languages**: C++, System Verilog, Verilog, Python, C, Embedded C, Linux
- **EDA Tools**: Xilinx ISE, ModelSim, Synopsys Design vision, Cadence, Mutlisim, Keil
- **Architecture**: Cache Hierarchy, Multiprocessor Architecture, Static Timing Analysis, OOO Pipeline, RISC-V.

WORK EXPERIENCE

Advance Micro Devices (AMD-Xilinx) **San Jose, CA**

Product Development Intern *Sept 2023 – Dec 2023*

- Demonstrated an optimized Stamp and Repeat wrapper module to automate placement and route of kernels on the Versal FPGA AI Engine array for multi layered ML architectures, resulting in reduction of compilation time by 10%(*approx*).
- Developed software leveraging OOP concepts, to manage location constraints while routing tiles on a 50x8 AI Engine chip layout. This led to significant improvement in code organization and scalability for high computing applications.

Secure and Advanced Computer Architecture Lab **Raleigh, NC**

Graduate Research Assistant *Jan 2023 – Apr 2023*

- Implemented python scripts for including port specifications and statistics like IPC, number of memory transactions, to be recorded while simulating the target RTL model to generate its corresponding SST component for simulation runs.
- Contributed to the development of the Structural Simulation Toolkit (SST) simulator by actively working on the integration of multiple instances of RTL/C models to facilitate co-simulation of performance models and RTL.

Gilbarco Veeder Root **Mumbai, India**

Embedded Engineer *Aug 2020 – July 2022*

- Developed and debugged firmware for fuel dispensers on 32-bit STM32 Arm Cortex-M controllers, focusing on device drivers such as GPIO, RS485 and customized USART and I2C libraries to minimize EMI/EMC noise in the field.
- Designed an RS485 communication circuit with isolated ports using optocouplers for AdBlue fuel dispensers using Altium designer tool. Post fabrication, validated the circuit's performance and protocol compliance by transmitting data from the processor and observing reception on the PCB board via DSO oscilloscope.

ACADEMIC PROJECTS

Functional Verification of I2C Multiple Bus Controller | *SystemVerilog*

- Built a class-based layered test bench to verify the functionality of the I2C BFM using the object oriented approach with system verilog which includes components like monitor, scoreboard, agent, driver etc. Defined testplan, coverage groups, assertions, tests and obtained an overall coverage of 94% (including code coverage).

RTL Design for Deep Neural Network | *Verilog*

- Implemented a synthesizable multi-stage neural network hardware design, interfaced with SRAM for data transfer. Achieved an area-delay product of 270E6 in 1042 clock cycles at an 11ns clock period through detailed timing analysis.

Out-of-Order Superscalar Process Simulator | *C++*

- Devised a simulator for a 9-stage superscalar processor including register renaming, that fetches N-instructions per cycle, evaluated the instruction per cycle (IPC) trend by varying the issue queue, the re-order buffer size for different traces.

Coherence Protocols-based Cache Simulator | *C++*

- Designed a cache simulator for memory coherence bus-based protocols like MSI, MESI, MESI with Snoop filter and studied various metrics like the number of bus signals, and memory transactions.

Cache Hierarchy Simulator | *C++*

- Designed a cache hierarchy simulator based on WBWA write policy and LRU replacement policy, achieved the lowest average access time of 0.71ns leading to an increase in total area by 157% while implementing both L1, L2 cache levels.