CHANDRA KIRAN NARALA

LinkedIn chandrakirannarala@gmail.com github-chandrakirannarala +1(551)331-7492

OBJECTIVE

Detail-oriented engineering enthusiast with a strong foundation in logic design, Verilog, and a passion for creating efficient RTL modules. Seeking an entry-level Design Engineer position to apply my theoretical knowledge, dedication to quality, and eagerness to contribute to cutting-edge projects in a dynamic team environment.

EDUCATION

Master of Science in Electrical Engineering | NYU, USA | 3.405* GPA **December 2023**

- Relevant Courses: Advanced Very Large-Scale Integrated Circuits, Analog Systems and Circuits, Advanced Hardware Design, Computer System Architecture**,** Digital Signal Processing.

Bachelors in Electronics and Communications Engineering | GVPCOE, India | 3.453 GPA **September 2020**

- Relevant Courses: Switching Theory and Logic Design, Network Analysis, Microprocessors and Microcontrollers, Digital Signal Processing, Linear and Digital IC Applications, Data Structures.

SKILLS

Languages: C, Python, Verilog, VHDL, ARM Cortex Assembly Level Programming

Tools: Cadence Virtuoso, Cadence Innovus, Genus, Prime-Time, Matlab, Xilinx Vivado, NI Labview, Mutlisim, MS Office

Area of Expertise: Digital &

Analog Design, RTL Design and Coding, Debugging, Static Timing Analysis, Low Power Design Techniques, FPGA, STM32f429I discovery board computer architecture, network protocols

PROFESSIONALL EXPERIENCE

Programmer Analyst, Cognizant Technology Solutions, India **November2020 - December 2021**

- Executed UNIX queries, commands to trigger the jobs to start regression, monitored and analyzed the failed jobs.
- Restructured the selenium automation suites for a web application using data driven framework with maven dependencies and developed design test cases for testing deployed builds.

CASE STUDY/PROJECTS

MBIST engine with 256x4b SRAM Memory array using 7nm Technology

- Designed hardware to carry out BIST (Built-in self-test) algorithms on a 256x4 SRAM and performing STA.
- Developed a comprehensive behavioral Verilog description for each component of a 256 x 4b SRAM, incorporating a BIST engine at SRAM speed, enabling testing with 4 test patterns: Blanket 0 and 1, Checkerboard and reverse Checkerboard, March C, and March LR. Verified RTL through VIVADO simulations, Ensuring accurate functionality and performance.

8b ALU using 7nm Technology

- Fabricated hardware of 8b ALU to synthesize and build a physical layout using Genus and Cadence Innovus using ASAP 7nm library and programmed a behavioral Verilog description of 8b ALU and conducted synthesis to determine component dependencies on area power and cycle time and finalized the physical layout.

RISC-V RV32I 32-bit Processor

- Architected a 32-bit a processor which executes a subset of the open-source RISC-V RV32I instruction set.
- Interfaced this design on an FPGA basys3 board, conducted power, area, and timing analysis, with the design operating at 100MHz, reporting negative slack in timing constraints, allowing operation at even higher frequencies.

256x32 SRAM Memory Array using 7nm Technology

- Designed and simulated 8Kbits 6T SRAM memory cell, comprising essential peripherals such as row and column decoders, sense amplifiers, pre-charge circuits, and write drivers to support non-destructive read and reliable write operations.
- Used the Cadence simulator, HSPICE, to simulate the memory array circuit and performed simulations to evaluate functionality, timing, and performance for read and write operations.

Folded Cascode Operational Transconductance Amplifier using 45nm Technology

- Architected a folded cascode operational transconductance amplifier with a voltage biasing circuit, meeting design specifications with a 71dB gain, 180uW power consumption, and a 65^{o} phase margin using a 45nm CMOS design.
- Evaluated transistor sizing in OTA circuitry, leveraging overdrive voltage estimations to optimize voltage biasing and examined different transistor scales. Optimized phase margin and bandwidth performance through analysis and testing.

Fingerprint Matcher with GUI

- Developed a Python-based fingerprint matching application using OpenCV and tkinter, achieving a 95% accuracy rate through the implementation of the Scale-Invariant Feature Transform (SIFT) algorithm.
- Improved matching efficiency by 30% by integrating advanced preprocessing techniques, resulting in faster results for users, and reduced false positives by 20% through a refined matching process, ensuring more accurate fingerprint identification