CHANDRA KIRAN NARALA

LinkedIn **naralachandrakiran76@gmail.com** **GitHub** **+1(551)331-7492** **Website**

OBJECTIVE

Actively seeking a full-time role as an FPGA Design Engineer to apply hands-on experience in developing and optimizing FPGA architectures. Eager to contribute to innovative projects and leverage expertise in hardware design to enhance the efficiency and performance of digital systems.

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, x86 Assembly Level Programming

Tools: Cadence Virtuoso, Cadence Innovus, Xilinx Vivado, Genus, HSPICE, Matlab, NI Labview, Mutlisim, MS Office, VS Code, OpenCV, Tkinter, Simulink

Area of Expertise: RTL Design, FPGA, ASIC Design, Digital & Analog Design, Debugging, Static Timing Analysis, Low Power Design Techniques, Network Protocols, STM32f429I discovery board, Memory Architecture, Signal Processing

PROFESSIONAL EXPERIENCE

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

- Enhanced and optimized Python modules, incorporating iterative processes to execute UNIX queries and commands. Triggered job start-ups for regression, and actively monitored and analyzed outcomes.
- Revitalized selenium automation suites for a web application, implementing a data-driven framework with Maven dependencies. Spearheaded the development of design test cases for rigorous testing of deployed builds.

CASE STUDY/PROJECTS

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

- Developed a comprehensive behavioral Verilog description for each component, incorporating a BIST engine at SRAM speed.
- Enabled testing with 4 test patterns: Blanket 0 and 1, Checkerboard and reverse Checkerboard, March C, and March LR.
- Conducted thorough verification using VIVADO simulations, ensuring accurate functionality and optimal 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 | GitHub

- 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.
- Utilized 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 | GitHub

- 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° 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 | GitHub

- 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 20% via advanced preprocessing techniques, resulting in faster results for users.