Vidya Bhanu
Austin, Texas | (602) 702-8370 | vidya.bhanu96@gmail.com | linkedin.com/in/vidyabhanu

KEY QUALIFICATIONS

- Proficient in the complete chip design lifecycle: IP architecture, specification creation, RTL design, static front-end checks, debugging with verification engineers, and managing ECO processes to tapeout
- Skilled in operating systems including Windows, Linux
- Extensive experience with front-end and back-end digital design workflows

EDUCATION

Master of Science, Computer Engineering (Electrical Engineering) January 2021 – December 2022
Arizona State University
Courses: Digital Systems and Circuits, VLSI Design, Computer Architecture, Neuromorphic Computing Hardware Design

PROFESSIONAL EXPERIENCE

NXP Semiconductors Inc., USA, *Digital Design Engineer* February 2023 – Present

Working in mixed signal IP team to create digital wrappers for ADCs and System Clock Generator IP.

- RTL design of soft IP blocks. Used XML-based generation tools to create IP specification and register RTL
- Involved in debug, working closely with verification team
- RTL2GDSII flow through signoff of the digital firm blocks using legacy Cadence Genus and Innovus flow and Cadence Stylus in 40nm and 16nm FinFET technology nodes. Analyzed timing reports and made necessary updates to fix any violations
- Performed linting, DFT analysis, CDC/RDC analysis and synthesis

Tools and flows used: Synopsys DC, Spyglass Lint, Spyglass DFT; Cadence Stylus, Genus, Innovus; Questa CDC/RDC

Intel Corporation, USA, *Front-end Design Automation Intern* May 2022 – December 2022

- Created Synopsys Spyglass Lint flow using Python for standalone memory wrappers to ensure they are lint clean
- Enabled Spyglass lint for a sub-system and helped setup black-boxes and waivers in the design
- Created tool flow in a template repository to leverage these flows in different project repositories

PROJECT EXPERIENCE

RTL to GDSII design of energy-efficient custom hardware for MNIST hand-written digit recognition (7nm FinFET technology)

- Developed BNN using ‘Larq’ library to achieve low energy and computationally less expensive model with 95.56% accuracy
- Implemented the model in hardware using RTL in System Verilog. Verified functionality against testbench using ModelSim
- Synthesized the behavioral netlist using Synopsys DC with no design violations
- Used Cadence Innovus to place and route the design achieving a frequency of 1.1 GHz, area of 7730.41 μm^2 and power of 0.84 mW for reduced convolution

RTL to GDSII design of a distance and sort engine for K-Nearest Neighbor Algorithm (7nm FinFET technology)

- Developed RTL System Verilog code to implement distance and sort engine for K-Nearest Neighbor Algorithm and verified functionality by simulations in ModelSim
- Synthesized the behavioral code to generate Verilog structural netlist using Synopsys Design Compiler
- Performed Automatic place and route on Cadence Innovus by Tcl scripting and generated reports for area and timing
- Performed DRC and LVS checks on the resulting layout following APR using Cadence Virtuoso and Calibre
- Obtained frequency of 0.5 GHz, power of 94.94 mW, area of 1334.59 μm^2

Implementation of an Inverter Chain to optimize delay between initial inverter and capacitive load

- Developed a Python script using shutil packages to read from and write into HSPICE testbench file
- Determined optimal fan and number of inverters to obtain optimal delay