Ganesh Venkatachalam
https://www.linkedin.com/in/ganesh-vk
Email: gvenkata@andrew.cmu.edu
Mobile: +1-412-926-6193

EDUCATION

Carnegie Mellon University — Pittsburgh, Pennsylvania
Master of Science in Electrical and Computer Engineering — January 2022 – May 2023
Relevant Coursework: *Introduction to Computer Architecture, Modern Computer Architecture, Parallel Computer Architecture & Programming Hardware Arithmetic for Machine Learning, Reconfigurable Computing, Introduction to Computer Systems.*

PSG College of Technology — Coimbatore, TamilNadu
Bachelor of Engineering in Electronics and Communication Engineering — July 2013 – April 2017
Relevant Coursework*: FPGA Based System Design, VLSI Design, Data Structures and Algorithms*.

SKILLS

Languages: Verilog, System Verilog, C, C++. Python, Perl, Tcl, SKILL
Tools: Synopsys - Design Compiler, VCS, PrimeTime, **YosysHQ –** Yosys, SVA, **Cadence –** Virtuoso, Genus, ADE-L, ADE-XL, Maestro, **Xilinx -** Vitis HLS

ACADEMIC & RESEARCH PROJECTS

- **RISCV RV32I Architectural Simulator |** *C99*: Built an architectural-level simulator for the RV32I and RV32M subset. Modelled the effects of instructions on the architectural state, tested and verified the simulator with various RISCV C benchmarks compiled in riscv-gcc.
- **Designing a 5-stage Pipelined RISC-V Processor [with Branch Predictor and L1 cache] |** *System Verilog*: Developed an In-order RISCV processor with Pipeline forwarding & stalling. Implemented a Return Address Stack and direct mapped BTB to improve branch prediction accuracy along with a 2-way set associative D-Cache to improve Load/Store latency. Implemented Next-line Prefetcher and Stride prefetcher for the D-cache in order to improve latency for Loads as part of the Memory-Wall project in 18-447. Optimised the branch predictor and Cache area to reduce Power (Perf/Watt) and improve IPC, Performance for various RISC-V C benchmarks.
- **Superscalar Structures for Out of order Processors |** *System Verilog*: Experimented and analysed co-dependencies among different configurations of Out-of-order GEM5 CPU model and arrived at an optimal ROB and IQ size of 128 entries for a target frequency. Designed and synthesized the structures in System Verilog/Genus for scheduling a 4-way superscalar, out-of-order processor.
- **Integration and verification of an efficient MAC with NVDLA (In-Progress) |** *System Verilog, C++*: Integrating an efficient MAC unit with NVDLA (NVIDIA's Deep learning accelerator), and currently developing a suite for system-level verification and metric extraction (area, timing, power) for comparative analysis against NVDLA's mac design.
- **Verification and Tapeout of an 8-bit Microprogrammed CPU (In-Progress) |** *System Verilog, Python, C++*: Designing and formally verifying the datapath and control unit of a Micro-coded CPU as part of the TinyTapeout University Programme. The project involves using Open source Suite (Yosys, Verilator, Cocotb) which has support for most of the high-level convenience features of the SVA language.
- **Neural Network Accelerator |** *System Verilog*: Designed an IEEE FP-32 adder/ multiplier and modelled a two-layer fully connected Neural network for Iris dataset inference. Optimised the NN's Area and timing by using Google's BF-16 FP format while maintaining inference accuracy to be within 10% of IEEE FP32.
- **True Random Number Generator** | *System Verilog*: Developed a Ring oscillator based TRNG on the Cyclone-V FPGA to generate a random bitstream, using the on-board PLL's Jitter as a source of randomness. Used the NIST Test suite to evaluate the randomness of the resulting bitstream.
- **k-NN Classifier |** *System Verilog*: Developed a pipelined k-NN classifier and optimised the pipeline depth to explore the design space on Area and timing.
- **Hybrid Perceptron Branch Predictor |** *C++13*: Implemented various branch predictors and measured mis-prediction accuracy in GEM5 (MinorCPU). Developed a custom Hybrid branch predictor with a meta predictor that chooses between a Local PHT predictor and a path-history Perceptron predictor. Achieved a 10% speedup in program IPC and improvement in Misprediction accuracy compared to PHT/BHT Branch predictors for various GEM5 benchmarks.
- **Asymmetric Multiprocessing for CMP |** *C++13*: Developed a multithreaded C benchmark to demonstrate the effects of EPI throttling in Snapdragon 888 SOC in symmetric and asymmetric CPU core configurations. Analysed power consumption & EDP for different Linux benchmarks & different CPU governors.
- **Microbenchmarking single & multi-threaded cache performance for Snapdragon 888 SOC |** *C99*: Developed a single-threaded and multithreaded matrix multiplication benchmark to study the effects of matrix size on L1 cache misses and L2 cache misses by inducing false-sharing.
- **Cache Way Partitioning:** Modelled and studied the effects of way partitioning on LLC cache on a quad-core CPU. Analysed and measured the speedup of different multicore benchmarks with respect to a baseline 8-way set associative cache.
- **Parallel programming on the OpenGL framework** | *Java/OpenGL*: Implemented and optimised Parallel sum reduction kernel (Adreno 660 GPU) using interleaved addressing to improve DRAM bandwidth utilisation and reducing idle threads. Achieved an execution time speedup of 45% over the baseline kernel.
- **Parallel Prefix Sum Computation on GPU |** *CUDA/C++:* Implemented Parallel Prefix Sum on the RTX 2080 GPU and optimised the scan algorithm using the Blelloch Scan. Improved the memory access bandwidth by using rewriting the Kernel to perform scan on SM Shared memory.

WORK EXPERIENCE

Synopsys — Mountain View, California
R&D Intern, Silicon Realization Group — *May 2022 – August 2022*
- Interned with the PrimeTime Delay calculation Group with a prime focus on developing an algorithm for improving the accuracy of Timing constraint measurements for Flip-flops in the presence of waveform distortions. Formulated a new Primetime flow to calculate a more accurate Setup/Hold constraint and measured its accuracy against reference values obtained from HSPICE.
- Developed Automation Scripts in Tcl and Python to automate the above algorithm and constraint calculation & measurement across different cells and libraries.

Texas Instruments — Bangalore, India
Electronic Design Automation (EDA) Engineer (Software Engineer – II) — *July 2017 - July 2021*
- **Reliability and Aging Analysis:** Conceived and worked on developing a methodology and an Integrated workflow for Reliability Asserts Analysis and Disposition. Designed the Cadence SKILL GUI interface and the Python back-end from ground-up. Achieved a Cycle time reduction for design review from 2 weeks to 2 days.
- Prototyped and implemented a declarative rule-based filtering feature enabling reviewers to automate the review process. Reviewers can use simple human-readable, declarative rules capable of carrying out complex filtering actions on Review data.
- **MOSFET Characterisation GUI Tool:** Integrated a MOSFET visual analysis tool within the Native Cadence Environment. Revamped the GUI, added new features requested by customers across various TI design sites and extended support for then recent Analog PDK nodes.
- **Process Corner Filtering:** Formulated a framework for decreasing the Monte-Carlo regression simulation time. Corners with parameters (like threshold voltage V_t, saturation current $I_{d\text{-}sat}$) lying outside the PCD limit are discarded.

Graduate Student Intern — *January 2017 - June 2017*
- Experimented on a Piece-wise Linear (PWL) compression algorithm for testbench trace currents generated by Cadence Spectre for Electromigration (EM) analysis.

AWARDS AND ACADEMIC ACHIEVEMENTS

- **Texas Instruments Global Gold Recognition Award:** Received recognition for critical contributions to the Reliability Asserts Analysis tool and the Cycle time reduction methodology.

PUBLICATIONS

- Sri Harsha Reddy Kaliki, Chanakya K.V, Ganesh V, & Senthil Kumar Sundaramoorthy (2021*) Raaga - Sign-off your Circuit Design Reliability*, Texas Instruments India Technical Conference, Bangalore.