Phani Teja Vempati

California | +1 (213) 301-7078 | pvempati@usc.edu | https://www.linkedin.com/in/tejavempati/

Education

University of Southern California - Viterbi School of Engineering **Jan 2024 - May 2025**

Masters of Science in Computer Engineering | **GPA : 4/4**

Course work -Computer System Organization (EE 457), MOS VLSI Design (EE 477), Computer Systems Architecture (EE557), Digital System Design (EE 560), VLSI System Design (EE577A and 577B), Design and Diagnosis of Reliable Systems (DFT) (EE658)

VIT University **August 2014 - June 2018**

Bachelor of Technology - Electronics and Communication Engineering | **GPA : 9.13/10**

Work Experience

DFX RTL Intern - Nvidia **Jan 2025 - Present**

- Working on scan chain insertions using Tessent SSN tools and generating EDTs based on flip-flop count and compression ratios to enhance test coverage and reduce test time
- Collaborating with Physical Design (PD), Placement and Routing (PnR), and timing engineers to ensure scan chain integration was aligned with design constraints and timing closure goals

Course Producer under Prof Gandhi Puvvada(EE457) - University of Southern California **May 2024 - Present**

- Assist students during office hours by offering support with their assignments and helping them prepare for exams.

Silicon Firmware Engineer - Intel **July 2018 - May 2023**

- Developed firmware microarchitecture, design, and validation for power management in SoC products
- Developed mechanisms to adjust voltage and frequency (DVFS) and implemented low-level features to power down system components (power collapsing) in response to the current workload.
- Worked on several hardware bring ups, boot sequences including low level debug of board and SoC's
- Translated controls algorithms into software implementation and integrate them into firmware and validation frameworks
- Expert in creating calibration plans which were deployed on production lines to guarantee quality and performance
- Exploring subsystem architecture performance trade-off for FW and HW optimization

Academic Projects

RTL for Out-of-Order execution 32-bit Tomasulo CPU with Speculative Execution | VHDL, Verilog, Vivado

- Built an Out-of-Order Machine with in-order commitment and speculative branch execution using Speculative RAT and RRAT
- Developed Copy-Free Checkpointing (CFC) using BRAMs with a Dirty Flag Array, tag, and valid array for branch misprediction
- Engineered Free Register List (FRL), Dispatch Unit, Issue Unit, and Common Data Bus (CDB) for streamlined instruction flow
- Implemented Reorder Buffer (ROB), Return Address Stack (RAS), Branch Prediction Buffer (BPB), and Arithmetic Logic Unit (ALU)

GPU Performance Analysis | C++, CUTLASS

- Utilized DWMMA, SGEMM, and DGEMM testbenches to benchmark GPU performance, experimenting with different tile sizes, matrix data layouts (row-major vs. column-major), and dimensions
- Focused on maximizing data locality and minimizing memory access latency by efficiently reusing cached data
- Captured KPI's such as simulation cycles, IPC, warp occupancy, memory stalls, shared memory bank conflicts, and L1/L2 cache miss rates
- Compared the efficiency of GPUs across different GEMM kernels, assessing the impact of varying configurations on overall performance

PCIE – PHY layer RTL implementation | Verilog, FPGA

- Designed elastic buffer, lane deskew buffer, 8b/10b encoding and decoding, Byte striping and unstriping and SOS addition and removal on a 2 lane link for the Physical Layer
- Implemented Serial to Parallel conversion, 2-clock FIFO, Symbol lock detection and primed method for elastic buffer

Implementation of Branch Predictors | Gem5, C++

- Analyzed performance of Always Taken, Global 2-bit, Bimodal, Correlated, TAGE, and Perceptron predictors using C++
- Achieved prediction rates of 0.45, 0.47, 0.79, and 0.83 for respective models

MIPS 5-Stage Pipeline CPU Design with BRAM | ModelSim,Xilinx ISE, VHDL

- Implemented Pipeline Design with flow through and pipelined BRAM (SSRAM), with Instruction Fetch, Decode, Execution, Memory and Writeback stage which supports MIPS ISA using RTL coding in Verilog
- Optimized design for removal of redundant stage registers using shadow and keep register logic

RISC-V Out-of-Order Performance Analysis | Gem5

- Utilized Gem5's RISC-V architecture within a Docker image to analyze performance, focusing on four benchmarks: integer/floating point operations, recursion, branch prediction, and memory operations
- Cached division results minimized recomputation, reducing the number of division operations for repeated inputs
- Incorporated a Victim Block Cache Replacement strategy using a Least Recently Used (LRU) stack to improve cache efficiency

Divider with Fully Associative Cache - CAM Design | ModelSim, VHDL

- Designed a cache along with LRU (least recently used) unit to help replace victim block
- Developed a fully associative cache using Content Addressable Memory (CAM) to store and retrieve results of previous division operations.

Key Skills

- **Knowledge:** CDC, Low Power Design, STA, DFT, DFX RTL, GPGPU, Cache Coherency, UART, AXI, PCIe, I2C, Virtual Memory, SRAM, Logic Design, CAD, Physical Design, PnR, RF Conformance testing (3gpp 36.101, 36.521)
- **Programming and Platforms:** Verilog, VHDL, Python, C, Matlab, Git, Linux, Unix, Perforce
- **Instruments:** Logic Analysers, Spectrum Analysers, Call boxes (R&S, Anritsu, Keysight), Direct far-field and Indirect far-field test chambers