Ameya Gurjar

(+1) 480-809-0492 | agurjar2@asu.edu | www.linkedin.com/in/ameyagurjar | github.com/Ameya674

Education

Arizona State University — Tempe, Arizona
Master of Science in Computer Science — *Aug. 2024 – Present*

College of Engineering Pune — Pune, India
Bachelor of Technology in Electrical Engineering — *Jun. 2020 – Jun 2024*

Skills

Languages: Python, C, C++, Javascript, SQL, HTML, CSS, R
Developer Tools: Docker, Git, VSCode, gem5, Jupyter Notebook, QEMU, GNU Debugger, Power BI, Tableau
Frameworks: Cuda, MLIR, Node.js, React.js, Streamlit, Django, Flask,
Libraries: Pandas, Numpy, Matplotlib, Seaborn, Scikit-Learn, TensorFlow, Keras, PyTorch, NLTK

Publications

Ameya Gurjar[5] et al., **"DSP-MLIR: A MLIR Dialect for Digital Signal Processing."** Accepted at the Languages, Compilers, Tools and Theory of Embedded Systems conference (LCTES) 2025.

Experience

Compiler Researcher | *MPS Lab at ASU* — Aug. 2024 – Present

- Developed a DSP compiler with a DSP DSL and a DSP dialect with the **Multi-Level Intermediate Representation (MLIR) framework.**
- Performed **domain-specific** optimizations and lowered to LLVM IR to achieve greater compilation speed than the Matlab compiler and GCC.

Projects

DNN Workload benchmarking | *Python*

- Benchmarked simple MLPs and Convolutional neural networks for different no. of layers, different batch and kernel sizes, and no. of neurons in each layer on a CPU and a GPU.
- Benchmarked ResNet and BERT for different batch sizes.
- Benchmarks were, training time, single batch inference time, accuracy, no. of parameters, no. of MACs, etc.

Forget gate based RNN | *Python - Pandas, Numpy, Matplotlib, Seaborn, Sci-kit Learn, Pytorch*

- Developed a simplified **LSTM** variant for short-term solar energy forecasting using only the forget gate, with **less computational complexity** than the traditional LSTM model, to achieve nearly the same accuracy.

IPL Win Probability Predictor | *Python - Pandas, Numpy, Matplotlib, Seaborn, Sci-kit Learn, Streamlit*

- Designed an end-to-end application to forecast the outcome of an IPL cricket match based on real-time match statistics using logistic regression.

Gshare Branch Predictor | *Python, C++, gem5*

- Designed and implemented a **Gshare branch predictor** using a **2-bit counter** and a **64 predictor size**, utilizing a NAND operation between the lower 6 bits of the global history and the branch address to index the Pattern History Table (PHT).
- Benchmarked the predictor on the **Dijkstra** and **Quicksort** algorithms, achieving prediction accuracies of **84%** and **93%**, respectively.

LRUIPV Cache Replacement Policy | *Python, C++, gem5*

- Designed and implemented a custom **Least Recently Used** cache replacement policy using an **Insertion Promotion Vector** on gem5. The IPV decides the insertion and replacement of the new and victim cache blocks on the recency stack.
- Benchmarked the predictor on the **Dijkstra** algorithm, achieving a prediction accuracy of **84%**.

xv6 Bootloader | *C, xv6 OS, RISC-V Assembly, QEMU*

- Developed a bootloader linker script to boot the xv6 kernel on a RISC-V system using the QEMU emulator.
- Implemented a secure boot protocol to verify kernel integrity and enabled **Physical Memory Protection (PMP)** using TOR and NAPOT configurations.

Trap-and-Emulate Virtualization System | *C, xv6 OS, RISC-V Assembly, QEMU*

- Developed a **virtual machine monitor (VMM)** enabling secure execution of privileged instructions by user-mode processes.
- Managed in-memory virtual machine state and designed mechanisms to **redirect and emulate traps**.
- Built an instruction decoder to emulate trapped instructions via opcode and register mapping.
- Implemented Physical Memory Protection (PMP) with custom page tables and enforced U/S-mode memory access restrictions.