Jamshed Ashurov

+1 571 315 8339 | jamshed00ashurov@gmail.com | Portfolio | LinkedIn | LeetCode

Education

UC San Diego — La Jolla, California, USA
MS in Computer Engineering (GPA:3.6) — *Sep 2022 – June 2024*

UC San Diego — La Jolla, California, USA
BS in Computer Engineering (GPA:3.9) — *Sep 2018 – June 2022*

Industry Experience

Ampere Computing — Portland, Oregon, USA
CPU Architect Intern — *June 2023 – Sep 2023*

- Analyzed CPU component vulnerability (L1, L2, D/I-TLB, and LSU) to cosmic-ray-induced soft errors (AVF) using Cadence Palladium emulation platform.
- Improved memory controller performance by implementing ML search algorithm model, reducing latency by 5% and increasing bandwidth by 3%.

Nanome Inc. — San Diego, California, USA
Software Engineering Intern — *Jun 2019 – Sep 2019*

- Developed an iOS AR application in C# for analyzing 3D protein structures as an alternative to VR systems.
- Designed and integrated a local database system to efficiently manage data from online sources.

Research & Teaching Experience

UC San Diego Graduate Teaching Assistant — Jan 2023 – Jun 2024
Undergraduate & Graduate Networked Systems — Aaron Schulman, Alex Snoeren

- Created source code and testing frameworks for students to implement Sliding Window Protocol and TCP congestion control.
- Assisted students in understanding and debugging the implementation of a simple router.
- Developed source code for a DropBox-like distributed file-sharing application using gRPC, consistent hashing, and the RAFT consensus protocol.

UC San Diego Research Assistant — Jan 2022 - Sep 2023
Characterizing WebAssembly Performance in the Era of Serverless Computing — *ISSTA 2023 SRC*
Jamshed Ashurov, Zixuan Wang, Jishen Zhao

- Developed CPU, memory, and file system micro-benchmarks to analyze WebAssembly System Interface (WASI).
- Wrote scripts to analyze WASI performance on x86 and ARM architectures using the Linux perf profiling tool.

Projects

Machine Learning for Robotics — *Python, ManiSkill, PyTorch, TensorFlow, Keras*

- Achieved 90% accuracy in a 3D segmentation task of classifying furniture components by using PointNet CNN.
- Orchestrated a simulation environment for a robotic arm to pick up a cube from a wall's top and accurately place it within a designated target region on the ground by developing observation and reward functions to successfully execute Proximal Policy Optimization.

Recommeder Systems — *Python, PyTorch, TensorFlow, Keras*

- Improved Amazon user-product rating accuracy from 78-90% using Neural Matrix Factorization.
- Improved Food.com user-recipe interaction accuracy from 51-72% using Bayesian Personalized Ranking model.

Intellectual Robotic Chess Player Arm — *Python, TensorFlow, Keras, ONNX, Raspberry Pi*

- Improved real-time chessboard image processing accuracy by 15% using the NASNetMobile CNN.
- Reduced chessboard detection time from 9 to 2 seconds by employing ONNX runtime.
- Integrated the Stockfish AI chess engine to generate subsequent moves.

Parallel Computing — *CUDA, C++*

- Quadrupled LeNet-5 convolutional layer inference speed while maintaining 90% accuracy by optimizing 2D convolutions through tiled matrix multiplication.

Advanced Microprocessor Design *System Verilog, Verilator, Python, Bash*

- Implemented 8KB 8-component TAGE branch predictor, reducing mispredictions by 94%.
- Developed a victim cache with an NRU replacement policy, reducing average CPI by 10%
- Implemented DRRIP cache replacement policy, reducing data cache misses by 31.4%.
- Transformed an in-order pipeline into a scalar out-of-order processor using Partial Ordering memory disambiguation policy, reducing average CPI by 11%.

Snek Compiler *Rust, x86*

- Built a compiler for a dynamically typed language Snek, implementing Mark and Compact garbage collection for 20% improvement in memory management efficiency.
- Optimized performance through IR transition, constant folding, and variable propagation, reducing binary size by 25% and improving program execution by 30%.

Dual Core ML Hardware Accelerator for Attention Mechanism *System Verilog, Python*

- Utilized RTL design, verification, synthesis, and PnR methods to generate gate-level netlist and layout using Synopsis DC and Cadence Innovus.
- Orchestrated data exchange between cores operatings on separate clock domains with asynchronous FIFO.
- Reduced WNS by 4x and power consumption by 2x via pipelining, loop unrolling, and clock gating.

Machine Learning Skills

Supervised ML: Multilayer Neural Networks(MNN), Boosting, MLE, Bayesian Parameter Estimation, EM.

Unsupervised ML: PCA, LDA, K-Center Clustering, K-NN, SVD.

Search & Optimization: Newton Descent, Simulated Annealing, Cross Entropy Minimization, Search Gradient, A*.

Reinforcement Learning: Deep Q-Learning, PPO, SAC, RAINBOW, Imitation Learning.

Collaborative Filtering: User-User CF, Item-Item CF, LFM, Bayesian Networks, Factorization machines.

Technical Skills

Programming: C, Python, System Verilog, C++, Rust, Golang, MATLAB, Java, C#, Shell, Tcl, CUDA, Haskell, x86/ARM/MIPS Assembly.

Tools: Cadence Palladium/Virtuoso/Innovus, Synopsis Verdi/Design Compiler, ModelSim, Quartus, Questa, Verilator, Unity3D, Fusion360.

Technical Skills: Microprocessor Design, Machine Learning, Compiler Design, Parallel Computing, Logic Design, Performance Profiling, Software Development, Reliable and Scalable System Development, Universal Verification Method(UVM), Analog Circuit Analysis.

Additional Skills

Soft Skills: Leadership, Teamwork, Public Speaking, Coaching, Teaching.

Art & Survival Skills: Ballroom and Latin Dancing, Martial Arts, Rock Climbing.

Languages: Tajik (Native), Russian (Native), English (Professional).