Jamshed Ashurov

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

Education

UC San Diego — La Jolla, California, USA
BS/MS in Computer Engineering; ***GPA: 3.6/4.00*** — *Sep 2018 – June 2023 (Expected)*

Sandy Spring Friends School — Sandy Spring, Maryland, USA
High School Diploma; ***GPA: 3.6/4.00*** — *July 2016 – Jun 2018*

Industry Experience

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

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

- Developed from scratch an AR application on IOS devices that allows scientists, researchers, and pharmaceutical companies to analyze 3D proteins without the need for a VR experience.
- Developed a local database to store the elements fetched from an online database.
- Wrote initial scripts to enable cross-platform compatibility.

Research Experience

Characterizing WebAssembly Performance in the Era of Serverless Computing — UC San Diego
Graduate Research Assistant — *Jan 2022 – Present*

- Developed microbenchmarks to analyze the performance of WebAssembly System Interface(WASI) at a fine granularity.
- Wrote scripts to analyze the performance of WASI on various architectures using Linux profiling tools.
- Instrumenting the WebAssembly runtime to collect performance metrics.

Teaching Experience

Computer Networks — UC San Diego
Graduate Teaching Assistant — *Jan 2023 – June 2023*

- Designed and developed the source code and the testing framework for students to implement Sliding Window Protocol and TCP Congestion Control.
- Assisted students with understanding/debugging in building a simple router.

Projects

Snek Compiler

- Developed a compiler for the dynamically typed language Snek in Rust which incorporated essential language features such as arithmetic and binary operators, functions, dynamic data allocation, and other fundamental elements.
- Integrated the Mark and Compact garbage collection algorithm to efficiently manage memory and remove garbage data.
- Introduced optimization techniques, including constant folding and variable propagation, to enhance the binary file size and improve program execution efficiency.
- Orchestrated the transition to Administrative Normal Form and subsequently to Intermediate Representation for effective code transformation and optimization.

Dual Core Machine Learning Hardware Accelerator for Attention Mechanism

- Integrated an attention mechanism, enabling the computation of a weighted sum of values based on query-value similarity for efficient performance.
- Utilized RTL design, verification, logic synthesis, and place-and-route (PnR) methodologies to generate gate-level netlist and layout via Synopsis Design Compiler (DC) and Cadence Innovus.

- Implemented advanced low-power techniques, including pipelining, dynamic voltage and frequency modulation (DVFM), loop unrolling, memory double buffering, and clock gating, to optimize the power consumption of the design.

Intellectual Robotic Chess Player Arm

- Implemented a three-way network communication system involving a laptop, robotic arm, and camera.
- Utilized computer vision techniques to capture and process real-time images of the chessboard.
- Utilized a range of convolutional neural network (CNN) models, including Xception, MobileNet, SqueezeNet, and NASNetMobile, to perform chessboard detection.
- Employed different runtimes, such as TensorFlow, Keras, and ONNX, to optimize the detection process.
- Integrated an AI chess engine, Stockfish, to generate the next move and coordinated the Raspberry Pi to control the robotic arm and execute the determined move.

Fault Tolerant and Scalable Surfstore

- Developed a robust file-sharing application in Go and gRPC, closely resembling the functionalities of Dropbox.
- Implemented consistent hashing for even data distribution, minimal movement during node scaling, and fault tolerance for seamless operations across numerous nodes.
- Employed the RAFT consensus algorithm to establish leader election, log replication, and fault recovery mechanisms, ensuring high availability and data integrity within the distributed system.

Advanced Microprocessor Design

- Improved 5-stage pipelined MIPS 32 processor with advanced optimizations, achieving significant performance gains across all benchmarks. Utilized bash and Python scripts to analyze and collect performance data.
- Implemented 8KB 8-component TAGE branch predictor, reducing mispredictions by 94.3%(99.1%) on average across all benchmarks compared to the baseline predictor.
- Developed a victim cache using NRU replacement policy, resulting in an average CPI reduction of 10.2%(19.9%) compared to the baseline across all benchmarks.
- Implemented DRRIP cache replacement policy, reducing dcache misses by 31.4%(80.3%) compared to the baseline direct mapped cache across all benchmarks.
- Modified the baseline in-order pipeline to scalar OOO processor with Partial Ordering for memory disambiguation policy. Achieved an average CPI reduction of 8.9%(13.15%) across all benchmarks compared to the baseline.

Skills

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

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

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