Nareg Megan

naregmegan@gmail.com | NaregAmirianMegan.github.io | GitHub | LinkedIn | (608) 843-7045

Experience

SOC (SYSTEM ON CHIP) SOFTWARE ENGINEER – APPLE **JUN 2022 – APR 2025**

- Contributed to the development and validation of Apple's custom silicon, ensuring new chips met cutting edge performance and reliability standards.
- Led engineering of highly optimized stress tests targeting CPU, cache, interconnect fabric, and ML (Machine Learning) accelerators for pre- and post-silicon validation across multiple generations of iPhone, iPad, Watch, and Mac SOCs.
- Designed and implemented low-level OS, kernel, and IP driver features in C, C++, and ARM/RISC-V to support evolving hardware, maximizing performance and ensuring compatibility across silicon revisions.
- Crafted experiments to quantify software impacts on hardware-level metrics such as voltage, current, IPC, thermal behavior, and memory latencies, influencing silicon and system-level decisions.
- Automated workflows and enhanced team infrastructure with Python and Perl, eliminating productivity bottlenecks and improving development efficiency.
- Resolved critical hardware/software integration issues using LLDB, JTAG, and custom debug tools, leveraging detailed knowledge of microarchitecture and SOC-level design to isolate bottlenecks and bugs.
- Communicated low-level technical insights to cross-functional teams of 40+ engineers, delivering concise updates on performance investigations, debug root cause, and design tradeoffs.
- Promoted in July of 2024

SOC SOFTWARE ENGINEERING INTERN – APPLE **JUN 2021 – AUG 2021**

- Collaborated on SOC design and fabrication validation by developing system-level CPU stress tests in ARM to optimize functional unit utilization and analyze power rail behavior under peak loads.
- Re-architected OS-level thermal throttling mechanisms to support a unified, real-time API for handling both interrupt-driven and polled thermal events from various SOC components.
- Created a low-level binary injection method using SPI-NOR memory markers, enabling firmware to dynamically detect and adapt to hardware configuration changes during form factor validation.

TA, COMPUTER ARCHITECTURE – UC BERKELEY **JUN 2020 – DEC 2020**

- Taught foundational microprocessor architecture, memory hierarchies, and parallel computing paradigms. Guided use of tools such as OpenMP and Intel Intrinsics in C and assembly (RISC-V, x86), emphasizing performance trade-offs and systems-level thinking.
- Advised students during labs and office hours on debugging performance-critical code, analyzing memory access patterns, and making design decisions at both hardware and software abstraction layers.

SOFTWARE ENGINEERING INTERN – PLATFORM9 **JUN 2020 – DEC 2020**

- Developed and maintained backend services in Go for Platform9's Kubernetes application catalog.
- Engineered and tested REST APIs to integrate Helm 3.0 into a managed Kubernetes environment.
- Collaborated with other engineering teams to ensure smooth integration of features into the product.

Education

UNIVERSITY OF CALIFORNIA, BERKELEY – B.A. COMPUTER SCIENCE **2018 – 2021**

Coursework: Basic/Advanced Computer Architecture, Operating Systems, Optimization and Information Systems, Analog and Digital Circuits, Control Systems, Intro to Computing, Data Structures, Advanced Algorithms, Machine Learning, Discrete Math, Advanced Probability, Multivariable Calc, Lin. Alg. and Diff Eq., Physics (E&M), Chemistry, Quantum Mechanics, Quantum Computing (GPA: 3.64, Grad. in 3 yrs.)

Activities: TA - Computer Architecture, Tutor - Intro to Computing, Cal Launchpad (Professional ML Contract Projects), Biomimetic Milli-Systems Lab

Skills

Tools/Software: C/C++/Java, Python, Assembly (ARM, RISC-V, x86), Assembly level debuggers, Cadence, RTL (Verilog), Versioning and CI/CD, ML Software Stack (TensorFlow, PyTorch, XLA)

Conceptual Expertise: Computer Architecture (CPU/Cache/ML Accelerators), Performant Low Level Software, Embedded/SOC Protocols, OS/System Design and Structures, Analog/Digital Circuits, Data Analysis, Optimization and Machine Learning