Medhaav Chandra Mahesh

(510) 705-3647 medhaav.mahesh@gmail.com www.linkedin.com/in/medhaav

Education

University of California Berkeley **Aug. 2020 – May. 2024**

Bachelor of Arts in Computer Science *Berkeley, California*

GPA: 3.7

Relevant Coursework

- EE 194 - IoT 16nm SoC Tapeout
- EECS 151 - Intro to Digital Design & Integrated Circuits
- CS 170 - Efficient Algorithms & Intractable Problems
- CS 152 - Computer Architecture and Engineering
- CS 61B - Data Structures & Algorithms
- CS 162 - Operating Systems

Technical Skills

Languages: Verilog, SystemVerilog, C, C++, RISC-V, Python, Java, Javascript, x86
Developer Technologies: Git, UVM, GDB, VCS, JIRA, Visual Studio, Questa, Vivado, VCS, Linux Command Line
Other: Computer Architecture, FPGA, Digital Design, Caches, Operating Systems, TCP

Experience

Aril Inc **June 2024 – Present**

Digital Verification Engineer | Python, C++, SystemVerilog, RISC-V *Los Gatos, California*

- Work in a 20-person RISC-V startup to develop Python scripts that generate constraint random verification RISC-V tests for units of a RISC-V processor - CSRs, interrupt handlers, L2 cache - which helped me find 40+ bugs in RTL
- Used SystemVerilog to modify RTL signals for test benches and UVM to write a test bench that checks correct functionality and coverage of 100 performance counters
- Implemented components of a RISC-V software model using C++ - indirect CSR access, breakpoints, vector FP exception flags - which is used in thousands of regressions tests to check the functionality of the processor

University of California Berkeley EECS **January 2022 – September 2023**

Tutor & Reader | C, RISC-V, x86 *Berkeley, California*

- Grade exams and homework about security concepts, debug programs written in low-level programming languages like C and RISC-V, and answer questions about security and cryptography concepts during office hours
- Led tutoring sessions of 5 students on Computer Architecture concepts and assisted in teaching students to understand these concepts during class discussions and labs of 10-20 students

Authentic8 **June 2022 – August 2022**

A8 Labs (R&D Engineer) Intern | Python, JSON, Javascript *Redwood City, California*

- Design and integrate APIs using Python and JSON for Authentic8's products, such as the secure Silo browser and data-mining Harvester tool, with 3rd party apps
- Research and develop new technologies, extensions, and programs for integration with Silo, which will be used by the company's 100+ employees and 500+ clients
- Experience in Software development processes such as daily stand-ups, creating proof of concepts, use of Jira software, and Agile development using the Scrum project management framework

Projects

BearlyML | *Verilog, Chisel/Scala*

- Developed a 16nm SoC built for machine learning applications with 4 RISC-V cores, a prefetcher, and multiple ML accelerators in a team of 20+. Complete every design stage within one semester and send it for fabrication in May
- Designed RTL for a RELU unit for quantized vision transformer accelerator and integrate it with other components of the accelerator such as SoftMax and near-memory MAC units
- Worked with industry tools such as Cadence Innovus and Virtuoso for physical design tasks such as Place & Route and fixing DRC and LVS violations

3-stage RISC-V Pipelined CPU | *Verilog, SystemVerilog*

- Developed a 3-stage RISC-V CPU on a Xilinx FPGA board with synchronous memory modules. Pipelined to improve cycle time with optimizations like forwarding to improve CPI while preventing hazards
- Implemented UART serial protocol for user input through buttons and switches on the FPGA to be stored in the CPU
- Implemented components such as the DAC and NCO to play audio of a certain wavelength that allows the FPGA board to output sounds of user-specified wavelengths
- Wrote comprehensive test benches that I simulated and tested using waveform debugging tools like Synopsys VCS

Pintos Operating System | *C, x86*

- Led a team of 4 to create a working multi-threaded operating system
- Implemented basic OS services based on Linux kernel such as system calls for process control and file access
- Develop management of multiple threads and processes through the use of scheduling and synchronization, and allow for user programs to create user threads