David Y. Williams

1920 Barbara Drive, Palo Alto, CA 94303 • (650) 465-7765 • davidyw@umich.edu

EDUCATION

University of Michigan, Ann Arbor, MI — Expected Graduation: December 2016
M.S.E. Electrical Engineering — GPA: 4.00
Relevant Coursework: Digital System Testing (EECS 579), Microarchitecture (EECS 573), Parallel Architecture (EECS 570), Intro to Operating Systems (EECS 482)

University of Michigan, Ann Arbor, MI — May 2015
B.S.E. Computer Engineering — GPA: 3.79/4.00
Relevant Coursework: Logic Synthesis & Optimization (EECS 478), Computer Architecture (EECS 470), Embedded Control Systems (EECS 461), Programming & Data Structures (EECS 281)

WORK EXPERIENCE

Intro to Logic Design (EECS 270) Lab Instructor — January-April 2016

- Run weekly lab that teaches logic fundamentals in Verilog, and that has students test lab designs using Quartus to program an Altera FPGA. Answer student questions on lab designs and teach debugging skills. Grade weekly lab assignments.
- Assist lecture GSI with office hours, homework, answering piazza questions, exam question creation & exam grading

ARM System Validation Intern, Austin, TX — May-August 2015

- Developed workflow using Atlassian Bamboo & corresponding scripts to test the process of booting Linux & running stress tests (e.g. MMU stress or Cache Stress tests) for hardware under test. Workflow caught multiple errors. Determined source of errors and created corresponding patches in kernel code.
- Helped familiarize new full-time employee with code base, debugging strategies, documentation, etc.

Texas Instruments Applications Engineering Intern, Novi, MI — May-August 2014

- Developed Android application to monitor power consumption of automotive infotainment system. Modified Power-Management IC device tree and kernel driver to provide power rail measurements to android app. Ran tests to verify data and improve mapping of raw data to power consumption data.

Spinlectrix Summer Intern, Burlingame, CA — May-August 2013

- Developed a Python-based graphical user interface (GUI) to monitor and control the Spinlectrix flywheel-based energy storage system

COURSE PROJECTS

Cache Coherence Protocol Formal Verification (EECS 570) — March 2015

- Designed and formally verified a MESI cache coherence protocol using the Murphi verification language

Superscalar, Out-of-order Processor (EECS 470) — November 2013

- Designed, implemented, and tested a three-way superscalar out-of-order processor
- Primarily worked on load-store queue (LSQ), decoding logic, physical register file (PRF) & freelist, reservation station (RS), re-order buffer (ROB) and pre-fetching logic

SKILLS

- Experience coding in C++, System Verilog, Ruby, C, Perl, Python.
- OS, Embedded systems, and Computer Architecture fundamentals

* - Currently Enrolled