Aashir Farooqi

(949)-226-9612 | aashir24@gmail.com | https://github.com/AashPointO

Education

University of California, Davis **Davis, CA**

College of Engineering: B.S. Computer Engineering *GPA:* 3.32

Expected graduation: June 2020

Technical Skills

Programming/Markup Languages:

Fluent: C/C++.
Advanced: Verilog, HTML/CSS, Bash, MATLAB.
Beginner: Rust, Java, JavaScript LaTeX.

Technological softwares/libraries:

SPICE, UNIX-Based OS's, ModelSim, Vim, Android Studio, Quartus, GitHub.

Experience

Research Assistant **April 2018 - Present**
Miller Lab (*millerlab.faculty.ucdavis.edu*) **Auditory Neuroscience and Speech Recognition Lab**

- Independently brought up, prototyped, and implemented a hybrid hardware/software solution to cross-reference external audio and serial data inputs, with our EEG acquisition system in real time. Brought latency down from the previous iteration by a factor of 10.
- Wrote a MATLAB wrapper which grabs the gaze angle from our eye tracker through the Lab Streaming Layer API. Designed as a proof of concept to be incorporated into future studies which will require eye tracking data.
- Wrote embedded firmware code, created hardware schematics, and designed/assembled multiple PCBs.

Software Engineering Intern **June 2018 - August 2018**
General Atomics **EMS - Software and Controls**

- Converted thousands of lines of code from the mathematical intensive algorithms of an aircraft landing simulation from MATLAB to C, bringing the runtime of the simulation down by over a factor of 2. My conversion is now used in research and development of the actual aircraft landing system contracted for the world's most expensive aircraft carriers.
- Only intern in department of over 20 to earn "Most Valuable Player" award for saving "hundreds of hours in simulation time and greatly reducing control system tuning efforts".

Independent Projects (*source code available on GitHub*)

IOS Apps: *Round 'a Bound*, *Tic-Tac Emoji* *Swift* **Winter 2017 & Spring 2018**

- Mobile games utilizing the Spritekit API to detect physics collisions between nodes, and to exhibit independently made animations and sounds.
- Online leaderboard via a realtime database through Google's Firebase API, which parses through JSON data.
- Both originally published and reviewed on the App Store, culminating in over 250 downloads.

Websites: *aashpointo.github.io/KmapWebsite* *HTML/CSS & JavaScript* **Winter 2018**

- Given a set of truth table inputs, website outputs the *Sum of Products* and *Product of Sums* equations via the Quine-McCluskey method.
- Unlike other K-Map Generating websites, mine allows for multiple outputs, an algorithm which is scalable up to an arbitrary number of bits, and a dynamically sizing table through incorporation of JavaScript.