Daniel Christiansen

Recent graduate with a demonstrated history of tackling the core problem of a project.
I pride myself on having the ability to assess problems and break them down into workable parts.

Contact

Email christiansen.daniel@gmail.com
Github dmchristiansen
LinkedIn danielmatthewchristiansen
Ph# (503) 952-6047

Education

B.S. Computer Engineering
Minor in Computer Science
Portland State University
Graduated June, 2018

Skills & Abilities

Programming

C, C++
Verilog/SystemVerilog
Python, Tensorflow

Tools

Git, ModelSim
KiCad, LTSpice

General

Circuit Design, PCB Layout
Embedded Software Design
Solid Math Background

Employment Experience

Basilisk
(October 2016 - March 2018)
Line Cook

Smallwares
(Dec 2013 - July 2016)
Line Cook

Departure
(August 2012 - Nov 2013)
Line Cook

Relevant Projects

Senior Capstone - FPGA Quadcopter Flight Controller | Link

Used an FPGA development board to implement a flight controller module for a quadcopter in Verilog. Senior capstone project sponsored by Lattice Semiconductor.

- Uses I2C to poll external sensor module
- Implemented a PID feedback controller to use sensor data to successfully achieve stable flight
- Processes / generates control signals to communicate with receiver module and motor speed controllers

Industry Design Project - Mini Synth | Link

A one-term team project focusing on good design practice. Built a simple synthesizer from design documents to working prototype. The intent of this project was to teach necessary time and project management skills for working in industry.

- Implemented direct digital synthesis in C on an AVR microcontroller
- Designed PCB in CAD software
- Assembled and debugged prototype PCB

Cache simulator | Link

As a term project for a microprocessor system design course, I wrote a program in C to simulate a multi-level cache using the MESI protocol. The program takes in a memory trace file and produces statistics about the operation of the cache. Written to be easily re-configurable to simulate a wide variety of cache architectures.

Relevant Coursework

Embedded Operating Systems & Device Drivers
Covered basics of writing drivers for Linux

Microprocessor Interfacing
Taught basics of bare metal programming and ARM assembly

Microprocessor System Design
Covered memory systems, cache design, and bus operation

ASIC Modeling & Synthesis
Covered RTL design, timing analysis, and design for testing