andrewdavis
andrewkdavis.com
andrewkdavis97

ANDREW K. DAVIS

andrewkdavis@protonmail.com
115 Solar Street, Apartment 407
Syracuse, NY 13204
(315) 630-8565

US Citizen with Security Clearance

Professional Summary

I am an FPGA Engineer with 5 years of VHDL industry experience. I am looking for opportunities to accelerate my technical proficiency and growth. I have been both an individual contributor since June 2019, and I have been a lead since October 2023. My work has been primarily with Zynq Ultrascale+ family of devices.

Education

Clarkson University *Potsdam, NY* August 2015 – May 2019

GPA 3.520, Bachelor of Science in Computer Engineering

Undergraduate Research *3D Fingerprint Scanning and Modeling*
Created a prototype to photograph fingers and output 3D renders. Helped develop an algorithm to unwrap the 3D renders into nail-to-nail fingerprint rolls. Presented at the 2017 IARPA N2N Challenge in Laurel, Maryland.

Work Experience

SRC Inc *Syracuse, NY* June 2019 – Present

- As team lead: oversee a team of 2-3 engineers; lead daily standups; assign bi-weekly tasking.
- As individual contributor: architecture and documentation; coding and simulation; hardware integration.
- As interview team member: performed over 20 interviews, evaluate technical skill; evaluate culture fit.

NGEW *Python, Individual Contributor* June 2019 – November 2019
Created SVD to LaTeX tool in Python, based off of CMSIS SVD specification. This tool generates memory-mapped documentation for AXI4-Lite regions: register, FIFO, and RAM memory styles.

RECU *MPSoC Motor Control, ZCU106, Individual Contributor* December 2019 – November 2020
Motor control with DAC8771 (single channel, 16bit DAC). AXI4-Lite to SPI interface. Wrote to DAC registers via SPI from Zynq US+ PS interfaced through Python. Verification with Oscilloscope.

RFSTIM *RFSoC Multi-array Stimulator, ZCU111, Individual Contributor* December 2020 – May 2022

- Implemented 10GbE with SFP PHY/MAC. An external dynamic PDW generator sent ethernet packets containing transmit-scheduled PDWs. Designed message parsing logic on top of the extracted UDP data to extract the fields needed for transmit on the ZCU111 DACS.
- Designed AXI4-Lite controlled I/Q scaling. Scale factors were multiplied onto I/Q from full scale (1) downto 0. Adjusted both phase and amplitude values. Inputs were polar and were converted to complex I/Q using Xilinx CORDIC. Data was then fed into complex multipliers against a single I/Q data stream from IFGEN, to create I/Q manipulated streams before outputting to DACS.

Protean *RFSoC, XCZU43DR, Lead* May 2023 – September 2023

- FPGA design for Zynq Ultrascale+ AMS WB3XR2 Dual RFSoC. Match hardware requirements to FPGA IO planning. Documented high level AXI4L memory map from betweeb RFPE0 and RFPE1. Communication over 10G talks to RFPE0, which talks to RFPE1 over HSS LVDS. Each RFPE was allocated 4.25GB of user space addressable by PL.
- Performed initial bringup of card using TCL build scripts and Petalinux image using third party IP and Zynq US+ block design.

Protean *RF Front End, XCAU15P, Lead* October 2023 – Present

- FPGA design for Artix Ultrascale+ RF Front End card. Matched hardware requirements to FPGA IO planning and clock management.
- Designed front end to queue RF Control MORA messaging and write to interface hardware (filter shift registers, amplifiers, attentuators). Performed hardware integration using python scripts.
- Designed FPGA boot sequence. Artix is not MPSoC, so there is PS image or power management. Initialized ethernet to receive down tactical image for AXI4L addressable SPI Flash. Interface to Flash via STARTUPE3 primitive to access FPGA configuration pins. Configured ICAPE3 primitive to schedule a reboot to the FPGA.
- Test and Development of FPGA. Non-tactical image meant to control GPIO and various peripherals attached to FPGA. All of the GPIO is AXI4L controlled, from Xilinx JTAG to AXI4L Crossbar. Documented an I/O test plan and wrote top level FW to implement each test case. Did manual clock routing for external clock source. Routedthrough an HDIO not on a CMT, which can cause Vivado to fail on implementation. Manually constrainted to go through IBUFDS to BUFG to MMCM on the same CMT.