in 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.

Skills

Hardware	Zynq US+ RFSoC, Zynq US+ MPSoC, Zynq7000, Artix US+
Software	LaTeX, Git, Perforce, Confluence, MS Visio, Linux, Windows
Languages	VHDL, TCL, Python (VUnit/CocoTB simulation, and hardware integration scripting)
Standards	AXI4-Stream, AXI4-Lite, Ethernet, SPI
Functional	Mentorship, Tasking, JIRA Sprints, Daily Standups

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 in C++ to unwrap the 3D renders into nail-to-nail fingerprint rolls. Presented at the IARPA N2N Challenge 2017 in Laurel, Maryland.

Work Experience

SRC Inc *Syracuse, NY* June 2019 – Present

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

Protean Program *RF Front End Lead* October 2023 – Present

- Design/Code of RF Channelizer FPGA boot sequence. Artix is not MPSoC, so no PS. Therefore, supports booting from base image used to initialize Ethernet to an MPSoC that writes down tactical image to AXI4L addressable SPI Flash. Interface to Flash via STARTUPE3 primitive to access FPGA configuration pins. Also uses ICAPE3 to remotely schedule a restart 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.
- clock management. matching hardware requirements for io planning. user guide for xilinx clock constraints. mapping hardware requirements to fpga board io. matching specifications for lvds (vswr). wrote fw to queue up data (rf queue). wrote interfaces to hardware, matched device specifications (rf interface, spi, filter shift registers, rf amps, attenuators). performed hardware integration using python scripts.

Protean *RFSoC Lead* May 2023 – September 2023

- FPGA Design for AMS WB3XR2 Dual RFSoC. Also did high level AXI4L memorymap from Versal to RFPE0 and RFPE1. Versal uses AXI over LAN IP to talk to RFPE1 over 10G, which then talks to RFPE0 over HSS. Each RFPE was allocated 4.25GB of user space addressable by PL (derived from UG1085: system addresses).
- Integration/Build with third party IP and Zynq block design. Initial bringup of card and build script design.
- Design/Code of RF Channelizer FPGA boot sequence. Artix is not MPSoC, so no PS. Therefore, supports booting from base image used to initialize Ethernet to an MPSoC that writes down tactical image to AXI4L addressable SPI Flash. Interface to Flash via STARTUPE3 primitive to access FPGA configuration pins. Also uses ICAPE3 to remotely schedule a restart 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.

Oberon *RFSoC Individual Contributor* June 2022 – May 2023

- Various C drivers. 1-Wire driver for DS25EC20P-T EEPROM. 1-Wire driver for MAX31826MUA+ temperature

andrewdavis
andrewkdavis.com
andrewkdavis97

ANDREW K. DAVIS

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

sensor. GPIO driver for MAX9208EAI+ LVDS Deserializer.

- Code/Sim custom packet parser within a Transmit/Receiver front-end scheduler on RFSoC. Transmit packet headers with channel ID's were extracted and used to mux remainder of packet to scheduler on respective DAC channel. Likewise incoming receive data was packetized with incoming scheduled ID and headers. TX/RX data was FIFOd and compared against externallylocked timestamp for scheduling. Hardware propagation delay was adjustable by AXI4L registers.
- Design/Code/Sim for a pipelined Multi-Channel scaler. Multiple incoming transmit data streams flowed into the same amplifier hardware downstream, and needed to be scaled by up to 1/3 to prevent saturation on DAC, depending on the number of valid streams during a transmit. Scales were based off additions of 2^n shifts of the incoming data.
- Optimization of front end Transmit Receive resources. Reduced the buffered descriptor records to only the necessary fields required for that front-end application (BSP level). Transmit and Receive logic were split out to optimize Transmit (receive required all fields to be sent back to off-board Scheduler). Buffers were changed from 1K to 512 depth to infer half BRAMs. Various buffers were optimized to infer LUTRAM instead of BRAM. Ultimately reduced utilization from 192 BRAM downto 24.

NGEW *Individual Contributor* June 2019 – November 2019

Xylem on

- Created SVD to Latex Python command line tool. Based off the CMSIS SVD specification. Used to generate memory-mapped documentation for AXI4-Lite register, FIFO, and RAM memory style regions.

RECU *MPSoC Individual Contributor* December 2019 – November 2020

Motor Control on

- Design/Code/Sim DAC8771 (single channel, 16bit) AXI4-Lite to SPI interface. Also used Python script to configure and set output voltages via SPI, through AXI4-Lite commands issues through Zynq US+ PS. Verification performed with Oscilloscope. (what FPGA?)
- Design/Code/Sim for an electrical Resolver (azimuth/elevation/speed) feedback loop. (what FPGA?)
- Design/Code/Sim for an I/O Recoder IP. Could detect edge detections on GPIO and would record and store time deltas between transitions. Stored in BRAM and could be later played back for debugging efforts. (what FPGA?)

RFSTIM *RFSoC Individual Contributor* December 2020 – May 2022

Multi-array RF Stimulator on ZCU111.

- Implemented Xilinx RFDC to use 7 DACs across 2 Tiles. Integrated with IFGEN (custom transmit scheduler. took in frequency,pulse width, starttime, end time, and I/Q adjustments to generate static PDWs)
  – Implemented I/Q scaling AXI4-Lite controlled. Scale factors were a multiply of I/Q from full scale (1) downto 0. Usedto adjustboth phase and amplitude values. Inputs were polar and were converted using Xilinx CORDIC. One CORDIC was usedto serially take in 6 channels of input. The new complex channel data fedinto a complex multipliers with single I/Q data stream from IFGEN.
- Implemented 10GbE UDP for ZCU111. Used Xilinx 10/25G Ethernet Subsystem (PG210) and shared MAC/IPV4/UDP stack IP.
  – Used 10GbE for one ofthe SFP PHY on the ZCU111. External dynamic PDW generator transmitted ethernet packets containing transmit scheduled PDWs. Created message parsing logic on top of the extracted UDP data that contained the fields for the IFGEN scheduler on the ZCU111.
  – Used seventh DAC to feed back into an ADC usedto statically adjust ADC NCO tuning. Acted as a feedback loop to enable DAC calibration at power-on and any time during run-time. A constant waveform was transmitted and received from DAC to ADC and would be used to compare phase offsets against the other DACS. Manually had to captureoffsets by wiring each DAC into the ADC, but afterinitial capture, these phase adjustments created a calibration table that could be used to adjust output phase of each DAC to reduce potentially perceived dopper offset in downstream multi attenna receiver hardware.