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 career growth. I have been both an individual contributor since 2019, and I have led a small team of firmware engineers since the beginning of 2024. My work has been primarily with Zynq Ultrascale+ family of devices, with a focus on DAC/ADC RF Data Converter and 10G Ethernet IP.

Skills

- Languages: VHDL, TCL, Python (for Vunit/CocoTB Simulation, and Hardware Integration)
- Standards: AXI4-Stream, AXI4-Lite, Ethernet, SPI
- Software: LaTeX, Confluence, Git, Perforce, MS Visio, Linux
- Hardware: Zynq US+ RFSoC (XCZU28DR:ZCU111, XCZU43DR, XCZU47DR), Zynq US+ MPSoC (XCZU9EG:ZCU101, XCZU7EV:ZCU106), Zynq7000 (), Artix US+ ()
- Functional: Mentorship, Overview Tasking, JIRA Sprints, Daily Standups

Education

Clarkson University *Potsdam, NY* May 2019
Bachelor of Science in Computer Engineering | 3.520 GPA

Undergraduate Research: *3D Fingerprint Scanning and Modeling*
Created a prototype to scan and produce 3D renders of users fingerprints. Helped to develop an algorithm in C++ to project and unwrap the 3D renders into fingerprint rolls. Presented the prototype and research at the IARPA N2N Challenge 2017 in Laurel, Maryland.

Work Experience

SRC Inc *Syracuse, NY* October 2023 – Present
Lead Firmware Engineer

- Interview team member for the Firmware Engineering department; performed over 20 interviews, evaluating technical skill and a candidate's overall fit for the company and department.
- Act in a lead role to oversee a team of 2-3 engineers; lead daily standups and assign bi-weekly Sprint tasking.
- Act as the responsible individual for Functional Overview architecture and documentation.
- Oversee Detailed Design architecture and documentation, coding and simulation, and hardware integration.
-

SRC Inc *Syracuse, NY* June 2019 – Present
Firmware Engineer

NGEW

- Created SVD to Latex Python command line tool to generate Latex. Based off the CMSIS SVD specification. Used to generate memory-mapped documentation for AXI4-Lite register, FIFO, and RAM memory style regions.
- Design/Code/Sim CORDIC Rotation that could accept pipelined requests. (what FPGA?)

RECU

- 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

- 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

andrewdavis
andrewkdavis.com
andrewkdavis97

ANDREW K. DAVIS

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

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.

Oberon

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

Protean

- 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).
- 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.