Digital LLRF Technology on a microTCA Platform

Eugene Wu
Northwestern University
eugenewu2017@u.northwestern.edu

Tim Berenc
Argonne National Laboratory
berenc@aps.anl.gov

August 19, 2016

Abstract

The low-level radiofrequency systems (LLRF) in the Advanced Photon Source (APS) storage ring at Argonne National Laboratory control the amplitude and phase of the accelerating voltage in the RF cavities. Currently these systems use analog hardware, but the long-term goal is to migrate to digital LLRF technology. This project explores the viability of migrating to a micro-Telecommunications Computing Architecture (μTCA) platform. Since the present systems use analog baseband processing, an intermediate step is to use baseband digitizers for various applications such as AM/PM noise suppression of power-line harmonics [1]. A μTCA based FMC carrier board fitted with baseband digitizers has been evaluated for such purposes. Initial results of this evaluation are presented here. The FMC carrier could later be fitted with high speed digitizers for migration to full digital LLRF.

1 Introduction

In the APS storage ring, the LLRF systems are responsible for regulating the amplitude and phase of the voltages in the RF cavities.

At present, this is done using analog hardware. In the long term, the APS would like to upgrade these systems to digital LLRF. The μTCA specification, a platform used in high energy physics, digital LLRF, military, and telecommunications applications, offers a large ecosystem of manufacturers and products. For these reasons, μTCA is being considered as a platform for future digital hardware systems.

This paper presents the results of a 2016 Lee Teng Undergraduate Internship project which evaluated μTCA based hardware for baseband signal processing. The primary goal of the project was to set up the hardware and program the FPGA to successfully run analog-to-digital (ADC) and digital-to-analog (DAC) converters. The secondary goal was to then use digital signal processing to demonstrate adaptive notch filtering [1] which can be used for RF amplitude and phase modulation (AM/PM) noise suppression.

2 Hardware Components

The main hardware component in this project was the VadaTech AMC502 Field-Programmable Gate Array (FPGA) Mezzanine Card (FMC) carrier board. It

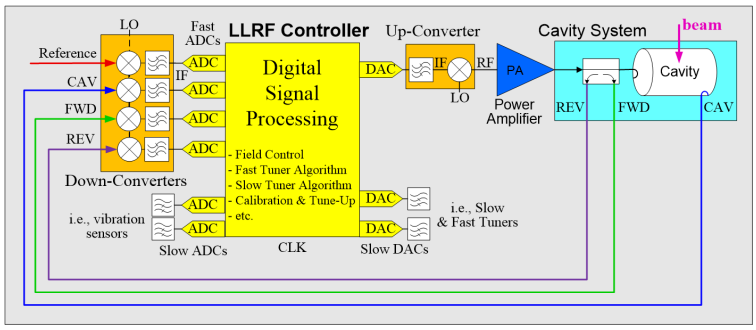

Figure 1: A digital LLRF system consists of a feedback loop that monitors signals from the cavity and adjusts the input voltage accordingly

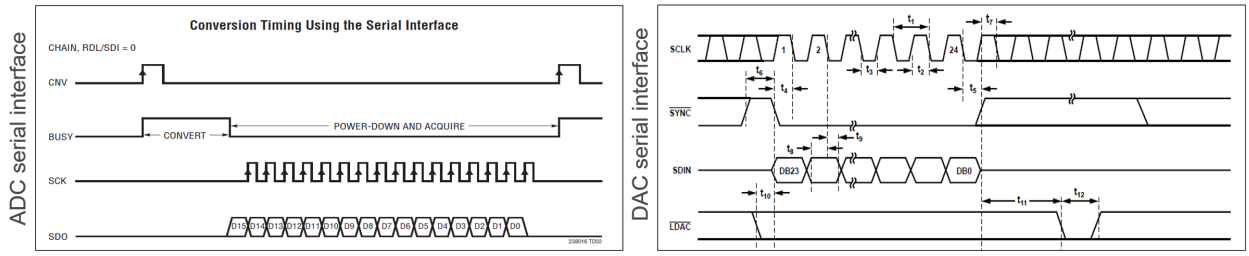

Figure 2: On the left, LTC2380-16 ADC Timing Diagram [2]. On the right, AD5780 DAC Timing Diagram [3].

has a Xilinx Kintex-7 FPGA along with two FMC slots.

The two FMC's mounted into these slots were the D-TACQ ACQ420 which has four 16-bit ADC's and the D-TACQ AO420 which has four 18-bit DAC's. The ACQ420 can run at 2 Mega-samples per second (MSPS), and the AO420 can run at 1 MSPS.

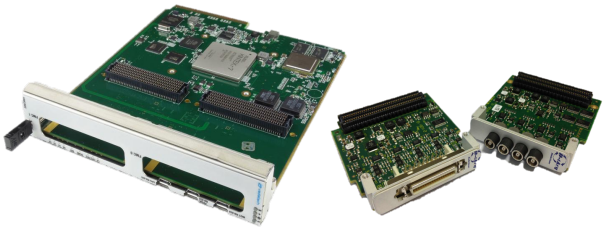

Figure 3: From left to right, FPGA Carrier board, ADC FMC, and DAC FMC

3 VHDL Design

The Virtex-7 FPGA was programmed using the Very High Speed Integrated Circuit Hardware Description Language (VHDL). Designs were created using Xilinx Vivado Design Suite, uploaded to the board via Gigabit Ethernet, and programmed onto the FPGA using the on-board CPU and flash memory.

3.1 Analog-to-Digital Interface

D-TACQ provided the VHDL code for the ADC interface which was then modified to fit our carrier board hardware and project needs.

The control logic module decodes the generics of the design and sets the gain bits, enable bit, clock divisor, and conversion time. It divides the 100 MHz clock by the clock divisor (set to 100) to obtain a 1 MHz convert clock. An internal trigger waits 100 clock cycles from when the FPGA starts to trigger the beginning of ADC conversion.

The SPI module outputs the convert clock signal, and shifts in and deserializes data from the ADC's. It also divides the 100 MHz clock to get the 50 MHz SPI clock.

The logic of the module ensures that the SPI follows the timing depicted above. The rising edge of the CNV signal initiates the conversion sequence. After a specified amount of time, the serial data out (SDO) is stored in shift registers, least significant bit first, on the rising edge of each SPI clock tick. After the completion of a shift, these shift registers are output from the module as parallel data.

3.2 Digital-to-Analog Interface

D-TACQ did not provide VHDL code for the DAC interface, so the following designs were developed as part of this project.

The DAC modules run off of the same 1 MHz convert clock as the ADC modules (i.e. all conversions trigger together on the same clock). A separate internal trigger also waits 100 clock cycles from when the FPGA starts to trigger the beginning of DAC conversion.

The SPI module outputs the Load DAC signal, Sync signal, and a serial stream of data to the DAC's. It divides the 100 MHz clock to get a 25 MHz SPI clock (note the difference in SPI clock speeds between the ADC's and DAC's).

The module utilizes a Moore machine to control the timing of the two control signals as depicted in Figure 2. Two modes of operation are shown. This project uses the synchronous DAC update mode. The LDAC and SYNC signals are high when conversion is inactive. To initiate the process, SYNC is brought low, and DAC is brought low a minimum of 10-13 ns later. The serial data in (SDI) is sent to the DAC, most significant bit first, on the falling edge of each

SPI clock tick. When the shifts are complete, SYNC is brought high to register and output the data. Afterwards, LDAC can be brought high any time before the next conversion cycle begins.

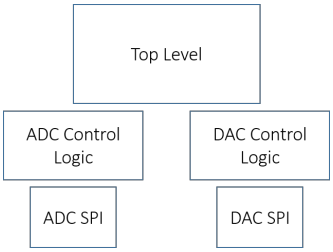

Figure 4: Hierarchy of VHDL Modules

3.3 Top-Level

A VHDL reference design provided by VadaTech was incorporated into the code to ensure correct pin assignments and board initialization. This reference design sets up a blinking LED as an indicator of clock functionality. It also sets up the necessary buffers for all of the I/O ports including a differential clock buffer for the 100 MHz clock. It then instantiates a Mixed-Mode Clock Manager (MMCM) to generate 200 MHz, 50 MHz, and 8 MHz clocks from the 100 MHz. Another set of buffers divides the 8 MHz clock down to 1 MHz. The design also sets up Gigabit Ethernet, a Microblaze CPU, and routing for four clocks from the Quad PLL to the M-LVDS Crossbar Switch [4]. A reset module puts the system into an initial state and holds off execution for 20 ns.

Figure 5 shows the flow of data through the system for a single channel and how the FPGA interacts with the ADC and DAC. Signal processing can be done with the parallel data between the ADC and DAC modules.

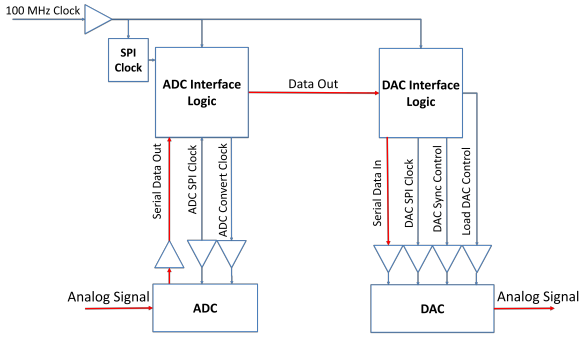

Figure 5: Top-Level Block Diagram of the FPGA, ADC, DAC Interface

Lastly, the top-level design routes and accounts for each of the 160 pins in the FMC connector. Though many of these are grounded, the rest of the pins need to be either connected to the correct location on the FPGA or set to floating. This process must be very carefully and meticulously handled, as any mistakes could cause damage to the FMC or carrier board.

3.4 Debugging

The ADC and DAC modules were simulated both separately and together on ModelSim. An Integrated Logic Analyzer (ILA) was instantiated in both the top-level module and the DAC SPI module for debugging the actual synthesized design. Internal signals from the ILA were read to the laboratory computer via JTAG connection.

4 ADC Data

A function generator was used to input a sinusoidal signal to the front panel of the ADC. The ADC data was read and downloaded from the ILA interface in Vivado. The data was imported into MATLAB, where we applied a Fast Fourier Transform (FFT) to decompose the signal into its component frequencies and their relative amplitudes. Figure 6 shows a peak at 10 kHz as expected. The noise at the higher harmonics for channel 1 results from the signal input source, and not from the ADC. Also note that, with a 10 V signal, we expect the peak at 10 kHz to reach 20 dBV. It is short of this quantity due to the effects of windowing.

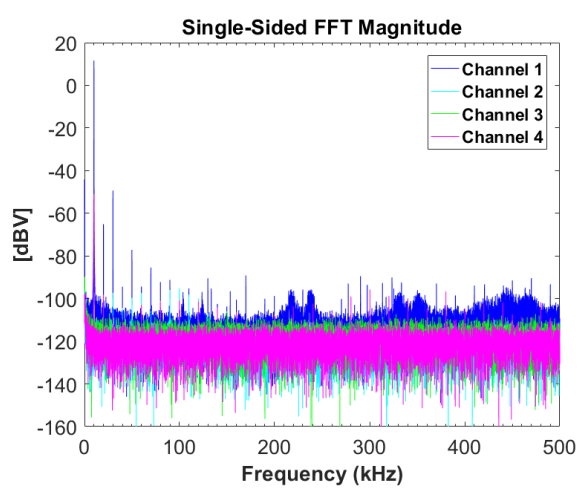

Figure 6: 32K point FFT with Blackman window. 10 kHz full-scale input on channel 1

With the same set of data we plotted the power

spectral density, a measure of the frequency distribution of a signal's power [5]. The signal source was not synchronized with the clocks driving the conversion logic. In conjunction with FFT windowing effects, this results in power leaking from the 10 kHz peak to other frequencies as seen in Figure 7. The peak at 10 kHz for channels 2, 3, and 4 is a result of crosstalk.

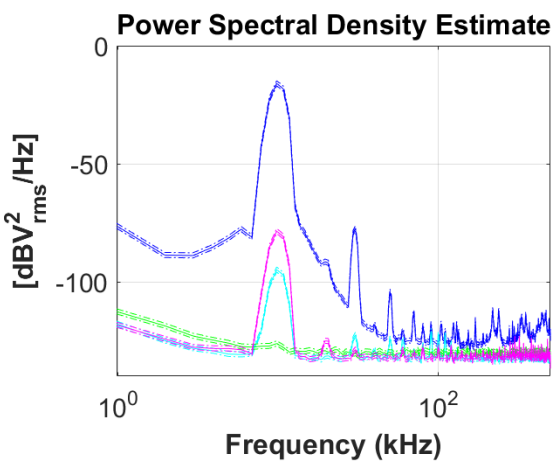

Figure 7: Power Spectral Density of a 10 kHz full-scale input on channel 1 using the Welch method with a Blackman window. Dotted lines denote 95% confidence intervals.

The noise-floor settles to about $-133\frac{\mathrm{dBV}^2_{\mathrm{rms}}}{\mathrm{Hz}}$. The total channel bandwidth is 500 kHz. The noise in the system can thus be approximated by:

$$\text{Noise} = \sqrt{\text{Noise Floor} * \text{Bandwidth}} \approx 158\mu\mathrm{V}_{\mathrm{rms}}$$

Since the input was approximately 7.07 $\mathrm{V}_{\mathrm{rms}}$ the signal-to-noise ratio (SNR) at full-scale input is:

$$\mathrm{SNR} = 20\log\frac{\text{Signal Voltage}}{\text{Noise Voltage}} \approx 93\ \mathrm{dB}$$

5 DAC Data

Functionality of the DAC was confirmed by looking at its output on an oscilloscope as shown in Figure 8. The amplitude difference between the input to the ADC and the DAC output results from gain settings in the ADC. The phase difference results from the signal processing delay through the hardware.

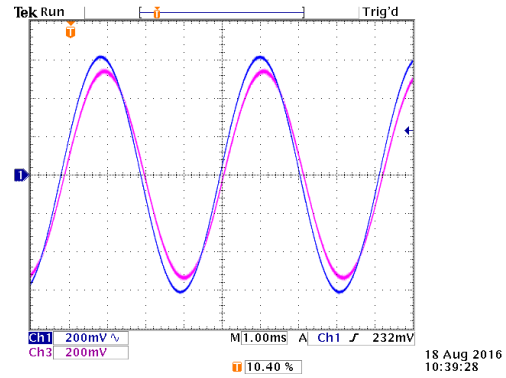

Figure 8: The ADC input is shown in blue, and the DAC output is shown in pink.

An FFT analysis of the DAC output shows a near replica of the signal source driving the input to the ADC as shown in Figure 9. Note that the 10 kHz harmonics originate from the signal source and are not caused by the ADC or DAC.

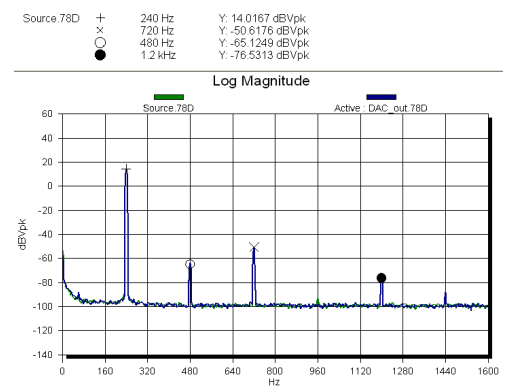

Figure 9: FFT of DAC Output (blue) and signal source (green) using Stanford Research Systems SR785 FFT Analyzer

6 AM/PM Noise Suppression

A two-weight adaptive notch filter as shown in Figure 10 was implemented to demonstrate the filter used in [1] for AM/PM noise suppression. The system in Figure 10 strives to match the DAC output to the noise source such that their sum at the error signal, ϵ, goes to zero.

The full system of [1] with Hilbert quadrature filters and narrow band pass filters was not implemented; instead a reduced version was used in order to demonstrate proof of principle. A sine and cosine noise reference signal was applied to ADC 2 and ADC 3 respectively. These signals are used for digital down-conversion of the error signal at ADC 1

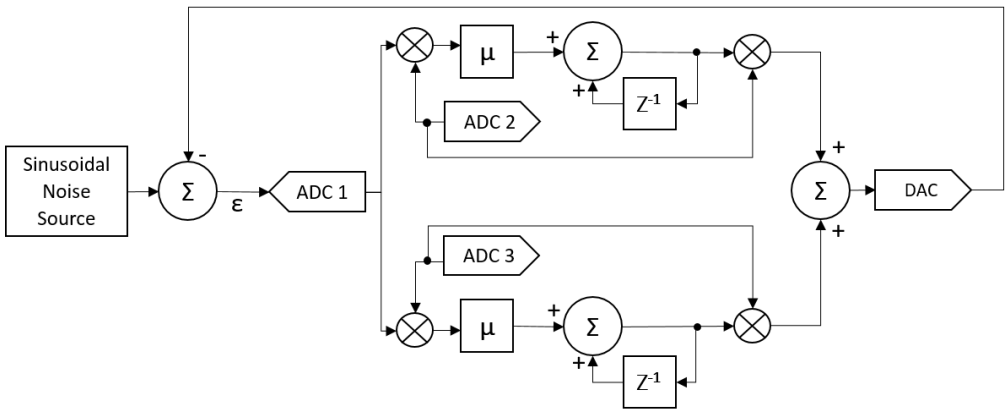

Figure 10: Two-Weight Adaptive Notch Filter

before the integrators (i.e. Z^{-1}) and the subsequent up-conversion after. The gain, μ, controls the notch bandwidth and convergence time. We used a sinusoidal function generator to produce a 240 Hz "noise" signal and the quadrature references to ADC 2 and 3. Figure 11 shows the noise suppression starting from the initialization of the filter with $\mu = 2^{-12}$. It takes about 60 ms for the DAC output to converge and cancel the noise. This time interval can be adjusted; a higher gain μ would yield a faster convergence time.

Figure 12 shows the suppression of 240 Hz in the frequency domain. The adaptive notch filter does appear to introduce noise at other frequency levels. We speculate that this noise is likely related to interference and noise between modules.

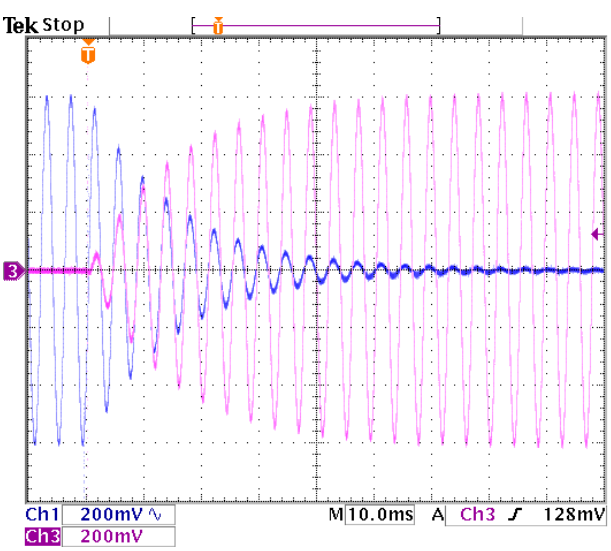

Figure 11: Noise Suppression in Time Domain. The blue shows the signal being suppressed, and the pink shows the output of the filter

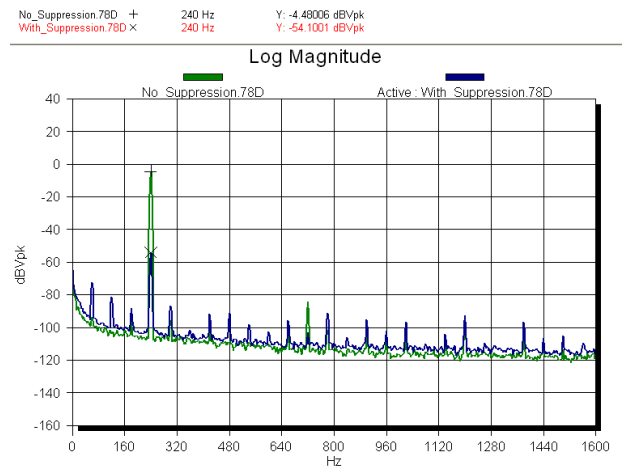

Figure 12: Noise Suppression in Frequency Domain. Comparison of FFT's with and without noise suppression

7 Future Work

This project has successfully demonstrated the use of ADC and DAC modules on a μTCA carrier board to perform baseband signal acquisition. Furthermore, the noise suppression algorithm that past Lee Teng projects have implemented on proprietary National Instruments hardware and design tools has in principle been successfully demonstrated on μTCA platform. Future work would entail full implementation of the AM/PM noise suppression system with the hardware described here.

In the long run, high speed digitizer FMC cards could be used with the same AMC502 FMC carrier board for full digital LLRF.

8 References

References

[1] K. Cook and T. Berenc, "Rf noise suppression for the aps storage ring," Sept. 2014, unpublished. [Online]. Available: http://www.illinoisacceleratorinstitute.org/2014 Program/student_papers/Kevin_Cook.pdf

[2] *LTC2380-16*, Linear Technology.

[3] *Datasheet AD5780*, Analog Devices, Nov. 2011.

[4] *VadaTech AMC502 Hardware Reference Manual*, VadaTech, Apr. 2015.

[5] E. P. Cunningham, *Digital Filtering: An Introduction*. Boston, MA: Houghton Mifflin Company, 1992.