Ares: An Oscilloscope Built Using an RFSoC

Bryan Jangeesingh
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA
brytech@mit.edu

Jonas Rajagopal
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA
jrajagop@mit.edu

***Abstract*—**
***Index Terms*—FPGAs, RFSoCs, Oscilloscope**

I. Introduction

Radio-frequency-system-on-chip (RFSoC) systems are field programmable gate arrays (FPGAs) configured to operate with signals in the GHz range. They are used in industry when parallel computation must be done on signals in the radio-frequency range.

Transient Grating Spectroscopy (TGS) is a materials science technique that enables the rapid, non-contact, and non-destructive measurement of the thermal and elastic properties of a material [1]. It's characteristics make it versatile and usable for measuring thin-film or bulk properties in both *in-situ* and *ex-situ* environments. Current TGS systems are lab based because of their size and complexity and a key limitation in shrinking, simplifying, and reducing the cost of TGS systems is the reliance on an oscilloscope.

TGS used highly sensitive analog photodetectors to detect the very weak diffracted probe signals which contain information about the sample. Currently, oscilloscopes are used to interpret and display this signal, after which LabView processes it before sending the data to either Python or MATLAB for fitting and analysis.

This system presented in this paper represents a step towards using an RFSoC for TGS's entire data collection needs. If successful, a completed system would interface with the two detectors and a pump laser trigger, processing all the data into Python to interface with the fitting code. A complete system would have three modes: live reading mode, where averages of 128 traces are sent directly to the screen; collection mode, where $\approx$ 10,000 traces are averaged and saved into a file; and calibration mode, where the system is zoomed out to view two successive triggers to align the chopper.

II. System Overview

The system receives data from an analog detector, and displays it in a Jupyter Notebook. A schematic showing each step can be seen in Figure 1.

A. The Board

A Xilinx Ultrascale+ ZU48R RFSoC was used for this system. Crucially, this board has four 5 Giga-samples per second (Gsps) 14-bit Analog-to-digital converters (ADCs). Vivado version 2024.1 was used to compile the source code and produce the bitstreams. The PYNQ system was used to interface with the onboard CPU and programmable logic with a Jupyter notebook.

B. The Detector

A Hamamatsu C5658 Si-Avalanche photodetector [2] was used to measure the incoming laser signals. It has a 500 µm detection area and a built-in low-noise amplifier. Its detection bandwidth limit is 1 GHz and it outputs data via an SMA cable which can be plugged directly into the ADCs on the RFSoC. It requires a +12V power supply for operation which was supplied with a Keysight E3620A 50 W dual-output DC power supply.

C. ADCs

The native RFSoC ADCs were used to process data from the analog detector. The board has 4 14-bit 5 GSps, and one of these was connected to the detector. This converter was set to "real" mode since incoming data does not follow the I/Q protocol. The converter was also configured to output packets of 12 signals at a time (192 bits) to enable the slowest possible clock.

The reference clock for the ADC was 491.52 MHz. The sampling rate was nine times this value, 4.423 Gsps. This dictated the S-axis clock to be one twelfth of this value, 368.64 MHz – the fast clock for the system. The output clock from the module was 34.56 MHz, so a clocking wizard was used to step this up to the necessary 368.64 MHz.

D. Trigger and Signal Manipulator

The ADC is always reading data from the detector. However, in a TGS system, the pump laser fires at 1 kHz. The relevant data about the sample is contained within roughly the first 1 µs after a pump fire. At a 4 Gsps sample rate, this corresponds to 4000 traces. In a fully functional TGS system, the pump laser trigger would be connected to another ADC, and this value interpreted to ensure the 1 kHz signal aligns with the pump firing. However, in the system presented here, the 1 kHz trigger is internally generated via a counter on the 368 MHz clock.

This module waits for a trigger and then asserts that it is valid for 256 cycles of the 368 MHz clock. This corresponds to 3072 raw traces since each output from the ADC contains

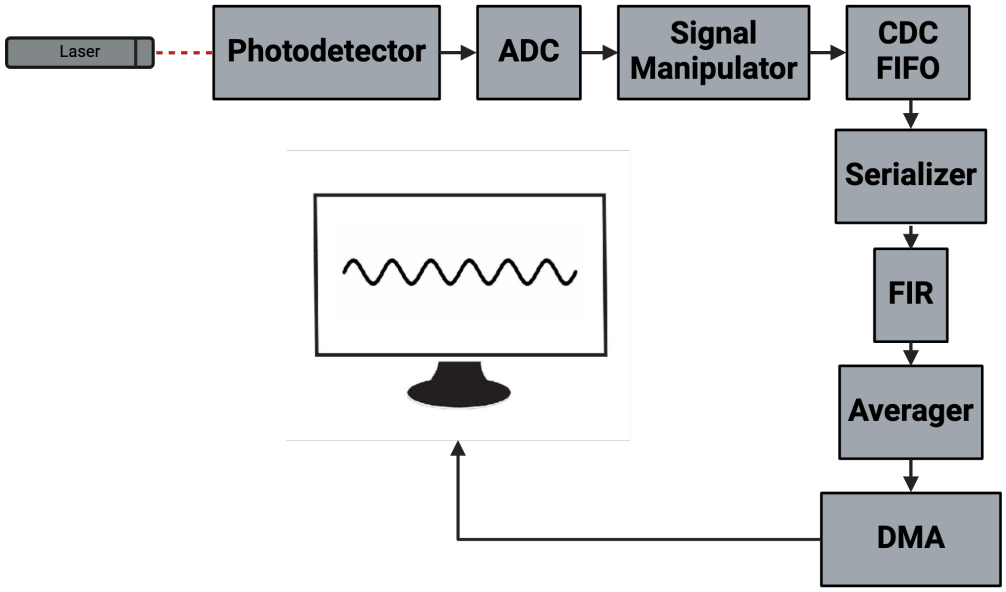

Fig. 1: An schematic overview of the data processing pipeline for the Ares system. The input is a laser and the output is a visual from the photodetector showing the signal.

12 traces. It then asserts a t-last signal to ensure compliance with the AXIS protocol and then goes back into dormant mode waiting for the next trigger. This procedure ensures that the downstream logic is only presented with the relevant data: 3072 traces for each 1 kHz trigger, giving the downstream logic ample clock cycles for any further computation.

E. Dual-clocked FIFO

To avoid issues with clock domain crossings (CDCs), the built-in Vivado AXI-4 Streaming Data FIFO was used to handle the only CDCs. Since the signal manipulator handles the trigger and valid signals, the FIFO should only fill up for the appropriate short time after a trigger when the manipulator asserts a high S-valid signal. The FIFO is sized such that it can hold the entire signal after a trigger to prevent any concerns of it filling up. Its S-axis is clocked on the slower system clock so that it transfers the data from the 368 MHz ADC clock to the 100 MHz PL clock. It does not manipulate the data in any other way.

F. Serializer

The values stored in the FIFO are sill 192 bits wide, as they consist of packets containing 12 14-bit ADC output values, each of which is padded 2 bits to 16 bits. Once the FIFO asserts that it is holding valid data, the serializer asserts M-ready for a single cycle, receiving a single 192-bit value. It then outputs 16-bit values over the next 12 cycles. The serializer reasserts readiness while processing the penultimate value to ensure no dead cycles and repeats this process 256 times to pass all 3072 traces to the FIR. This process does not require additional memory, as it stores only a single 192-bit value at a time.

G. Finite-Impulse-Response (FIR)

The values are sampled at 4.423 GS/s and the ultimate desired sample rate is roughly 1 Gsps. To prevent high-frequency aliasing artifacts, the signal must be passed through a low-pass filter. The cutoff frequency was the Nyquist frequency of 500 MHz. A 15-tap 32-bit FIR filter with 8-bit coefficients was used as the low pass filter with coefficients determined by an online filter-coefficient generating website designed to minimize ripple and create a sharp cutoff as described by the above constraints [3]. The total gain of the filter is 151, so the output value is divided by 128 to ensure the gain is close to 1; otherwise, the output values would be scaled unnecessarily high. The built-in Vivado FIR filter was used to ensure full AXI compliance. After the FIR, the signal can be decimated by a factor of 4 giving data at 1.105 Gsps, roughly the desired amount given the detector limits. After this process, there are 768 traces representing data for 694 µs following the trigger. Future systems may take data for longer values or experiment with keeping higher sample rates.

H. Averager

The above process creates a length-768 time series of the data produced by the detector after a single trigger. A single signal is very noisy, so when reading live data from the oscilloscope, it was necessary to do a small amount of averaging to smoothen the output. In Ares, 128 traces were averaged to enable easy division. The system contains accumulators for each of the 768 timestamps. Currently, these values are stored in an unpacked 2D array. A different design could utilize the native BRAM or Ultra-RAM, however, the onboard resources were not used to anywhere near completion. Data values for

128 successive triggers were summed and each value right shifted by 7 before being sent to the DMA for visualization. Since the trigger rate is 1 kHz, the DMA gets roughly 8 signals per second. Ultimately, the goal is to display this data on a live dashboard with a refresh rate of 8 FPS to facilitate the live alignment of the TGS system.

I. Clock Domain Crossing Challenges

The initial system design combined the signal manipulator, FIFO, and serializer into a single custom module designed to be a dual-clock memory system that receives data on the fast clock and utilizes the trigger to determine when to start writing values to memory. After receiving all values, it begins producing an output of 16 bits at a time on the slower clock. This design required a single Verilog module to have two $always\_ff$ loops, one on the fast clock to feed the memory and one on the slow clock to receive the data. However, no matter what we tried, this version would fail timing. The three failing paths were the clock domain crossings of the signals between the clocks dictating state, but even with modifying the $async\_reg$ parameter to $TRUE$, the system could not meet timing. For this reason, the built-in FIFO was used, which enables a specified length clock domain crossing for dual-clock operation. This required separate modules to manipulate the valid signals such that only values after a trigger are stored and to serialize the 192-bit busses into 16-bit ones.

III. PYNQ Usage

Since there is no I/Q mixing needed, the PYNQ code is very simply. Only 768 spaces are needed in the DMA, however, 1024 are allocated to provide some margin for error in the event of a delayed t-last signal. The setup code is otherwise the similar to Lab 6 without any mixing and with a live plotting system instead of a one-time one.

IV. Real-time data monitoring

Real-time data monitoring was performed using the PYNQ framework, Jupyter Notebook, and IPython. We specifically employed the interactive version of Matplotlib to avoid repeatedly calling plt.show(). A while loop was used to continually fetch data from the FPGA's *out_buffer*. During each iteration, the axes were cleared, new data was plotted, and the canvas was redrawn. This version of the plotting system is still slow: adding roughly 0.13 seconds to visualize each frame. Future system could use purpose-built Python plotting packages designed for real-time data and rapid updating such as Pyqtgraph or Bqplot.

V. Results

A. Waveform detection

Ares displays live data from the detector; though, it should be noted that no ground truth data exists to verify the accuracy of the system at this point, since the system cannot be connected to a TGS system because of the lack of the ADC trigger module and the requirement that the board and laser each remain in their respective rooms for the timebeing. A gain of 1.17 is expected, however, and the absolute magnitude is not critical since the Y-axis scaling is not important for TGS. Eventually, the system would be hooked up to a TGS system and data from a tungsten calibration sample acquired and fit. The fitting parameters could then be compared to the expected values from existing TGS systems.

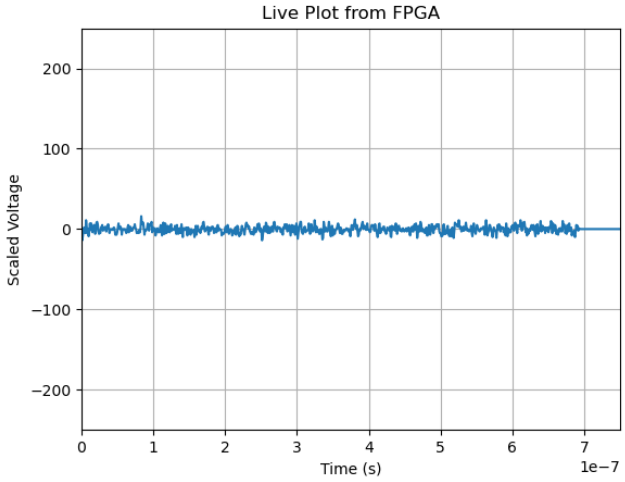

Fig. 2: Detected waveform when the detector is in ambient lighting. Minimal ambient noise is visible.

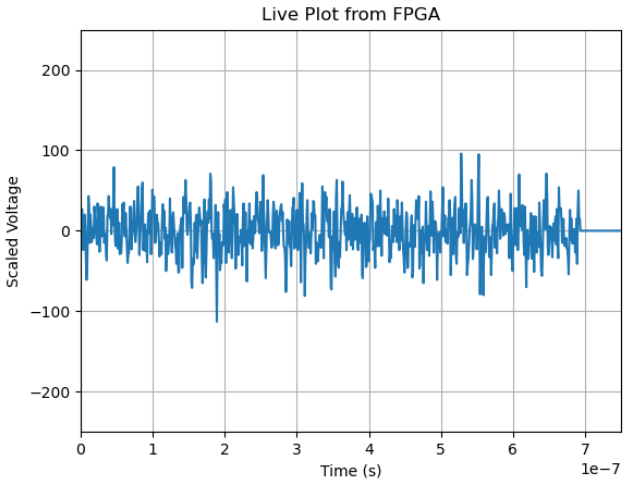

Fig. 3: Waveform when an iPhone flashlight is held over the detector. The larger noise magnitude is the expected behavior for the detector in this scenario.

A snippet of the waveform in ambient lighting can be seen in Figure 2. As expected, the only thing visible is a very weak background noise.

An iPhone flashlight was used to mimic a laser. The waveform shows a visible amplitude increase when this light was held over the detector (as shown in Figure 3). This is expected, since the flashlight should not contain any coherent nanosecond timescale patterns that would be detectable but does increase the in-coherent background night.

B. Timing and Resource Utilization

The presented system had a worst negative slack of 0.754 ns and did not have any failing timing endpoints. In terms of resource utilization, the system used around 2.45% of the total LUTs and 1.11% of the total Registers with none as latches. The system does not come close to using all of the onboard resources. Even with the length 3072 32-bit array used as the accumulator, less than 3% of the total look-up table resources are used. This will enable the sampling time to be doubled from 694 ns after a trigger to 1388 ns and the decimation to potentially be reduced.

VI. Future Extensions

A few future extensions exist to enhance the functionality of the system. One key area is to further optimize its responsiveness. With a 1 kHz trigger, the system outputs values at the correct rate of roughly 8 fps. This level of performance enables a more real-time alignment procedure.

Additionally, a second detector could be added since many TGS systems require two identical detection pipelines. This should not be too difficult, since the system for a single detector is in place.

Another potential extension is a data-acquisition mode. Once a system has been aligned, real-time feedback is no longer needed. However, many more triggers should be averaged to reduce the noise on a signal. This number tends to be closer to 10000, so a separate mode that can be triggered via a switch should be developed where a single 10000 trigger run is conducted and the data saved to a text file for future use (or output as a numpy array directly to other code).

Lastly, proper calibration mode would enable 1000 samples equidistant between two triggers to be displayed so that the chopper - a device that reduces the intensity of the probe beam - can be aligned between the two pump pulses.

VII. Code Availability

The code for Ares can be accessed at https://github.com/bryanjangeesingh/Ares.

VIII. Acknowledgments

The authors would like to thank Joe Steinmeyer for his assistance in developing and debugging the system. They would also like to thank the Short Lab for loaning the detector and power supply for the duration of the project.

References

[1] Felix Hofmann, Michael P. Short, and Cody A. Dennett. "Transient grating spectroscopy: An ultrarapid, nondestructive materials evaluation technique". In: *MRS Bulletin* 44.5 (2019), pp. 392–402. DOI: 10.1557/mrs.2019.104.

[2] Hamamatsu Photonics. *APD Module C5658*. Accessed: 2024-12-11. 2024. URL: https://www.hamamatsu.com/us/en/product/optical-sensors/apd/apd-module/C5658.html.

[3] TFilter. *Free Online FIR Filter Generator*. Accessed: 2024-12-11. 2024. URL: http://t-filter.engineerjs.com.