The discrete Fourier transform algorithm for determining decay constants—Implementation using a field programmable gate array

G. Bostrom,[1] D. Atkinson,[2] and A. Rice[1]
[1] *Physics Department, Portland State University, Portland, Oregon 97201, USA*
[2] *Chemistry Department, Portland State University, Portland, Oregon 97201, USA*

(Received 19 November 2014; accepted 21 March 2015; published online 8 April 2015)

Cavity ringdown spectroscopy (CRDS) uses the exponential decay constant of light exiting a high-finesse resonance cavity to determine analyte concentration, typically via absorption. We present a high-throughput data acquisition system that determines the decay constant in near real time using the discrete Fourier transform algorithm on a field programmable gate array (FPGA). A commercially available, high-speed, high-resolution, analog-to-digital converter evaluation board system is used as the platform for the system, after minor hardware and software modifications. The system outputs decay constants at maximum rate of 4.4 kHz using an 8192-point fast Fourier transform by processing the intensity decay signal between ringdown events. We present the details of the system, including the modifications required to adapt the evaluation board to accurately process the exponential waveform. We also demonstrate the performance of the system, both stand-alone and incorporated into our existing CRDS system. Details of FPGA, microcontroller, and circuitry modifications are provided in the Appendix and computer code is available upon request from the authors. © *2015 AIP Publishing LLC.* [http://dx.doi.org/10.1063/1.4916709]

I. INTRODUCTION

Cavity ringdown spectroscopy (CRDS) is one of several cavity-enhanced absorption techniques that make use of the multiple reflections of light inside a high-finesse optical cavity to increase the interaction of the light with gases inside the cavity. If there is a fixed amount of light energy inside the cavity, then an absorbing species will absorb a fraction of that light during each pass given by the Beer-Lambert law,

$$I\left(l\right) = I_0 e^{-\varepsilon l c}, \tag{1}$$

where I_0 and $I(l)$ are the initial intensity and the intensity at a path length of l in cm through an absorber, respectively, $\varepsilon(\nu)$ is the absorption coefficient in 1/(molecules/cm^2) at wavenumber ν in cm^{-1}, and c is the concentration in molecules/cm^3. Literature values for absorption coefficients can be determined using the line intensities catalogued in the HITRAN database.[1] A continuous fractional reduction results in an exponential decay, $I(t) = I_0 \exp(-\beta t)$, where I_0 and $I(t)$ are the initial intensity and the intensity at some time t later, respectively, and β is the decay constant (in s^{-1}). Comparison with Eq. (1) shows that β is proportional to the concentration of the absorber.[2] Continuous-wave CRDS injects light into the cavity at one of the cavity's resonance modes, storing energy inside the cavity. Once sufficient energy is reached, light injection is stopped, and the output of the cavity (which is proportional to the light inside the cavity since the mirror transmits a fraction of the light) is measured as the energy exponentially decays, and the time constant β is determined. Scanning the laser across many cavity modes generates an absorption spectrum. Advantages of CRDS over other absorption techniques include the following: (1) it is nominally unaffected by variations in laser output power and (2) the absorption determination makes use of a large dataset for each data point.

Cavity ringdown spectroscopy offers improved sensitivity over other absorption techniques,[3,4] but significant data processing is typically required to determine the decay constant. Since sensitivity can be improved by averaging many measurements,[3,5] the speed at which data can be processed can limit the performance of CRDS systems. Several algorithms exist for decay constant determination, such as the traditional, but computationally intensive, Levenberg-Marquardt non-linear fit. Alternative algorithms such as corrected successive integration (CSI), the linear regression of the sum (LRS), and Discrete Fourier Transform (DFT) algorithms[6,7] are more efficient, typically orders of magnitude faster than the non-linear fitting. With the use of these fast algorithms, getting the signal from the digitizer to the processor becomes the bottleneck for data collection. Some applications have avoided the conversion stage altogether by using an analog method of determining the decay constant.[8] Others slow down the transfer by determining the decay constant of an average of many ringdowns, rather than averaging decay constants of many individual ringdowns. However, if the latter has an assumed normal distribution, then the former would have a lognormal distribution, with a mean that depends on both the average β and its standard deviation, introducing additional error that can be avoided if individual ringdowns could be processed in real-time.[9,10] Furthermore, the precision becomes limited by the number of bits of the high-speed data acquisition system, and averaging ringdowns has been shown to enhance low-frequency noise on the ringdown signal, requiring additional processing.[11] Finally, processing individual ringdowns enables removal of outliers that can significantly affect the averaged ringdowns.[10]

Improvements in speed, and therefore sensitivity, can be obtained by "moving" the processing stage closer to the analog-to-digital converter (ADC) stage. High-speed Field-Programmable Gate Arrays (FPGAs) provide the possibility of putting a customized, highly efficient processor module immediately after the ADC.[12] Few practitioners of CRDS have applied this approach despite the fact that the ADC and FPGA integrated circuits (ICs) can be relatively inexpensive, presumably because of a perception that building the proper support circuitry around them can be time-intensive and requires a high level of skill in circuit design. Purchasing commercial high-speed data acquisition modules with (or even without) onboard FPGAs can be costly (several thousand dollars). In this work, we pursued a middle ground between designing our own data acquisition and processing (DAP) system from the ground up and buying a commercially available system. Adapting commercially available hardware designed for high-speed communication applications to make it compatible with the exponentially decaying signals encountered in CRDS provides a cost-effective solution for high-speed DAP to acquire decay constants, which we demonstrate at rates up to 4.4 kHz.

II. METHOD

The central components of our system are commercially available evaluation boards from Analog Devices, Inc. (Norwood, MA) designed for the evaluation of their high-speed ADC IC. Additional components used to interface with these boards, as well as the modifications we made to the evaluation boards, are discussed in Secs. II A–II C. To our knowledge, there is only one report of using a FPGA for processing CRDS data,[12] in which an Altera FPGA evaluation board with an ADC daughter card is used to compute the decay constant. In that work, they used a modification of the discrete sum/frequency component algorithm and used a lookup table to determine the time constant from the ratio of two frequency components. Their system was implemented on the FPGA, and they demonstrated its performance at 20 Hz. In addition to the different FPGA and ADC evaluation boards, our approach is different in two fundamental ways: (1) we implement the DFT algorithm described by Everest and Atkinson[7] using a Fast Fourier Transform (FFT) core to determine the decay constant without use of a lookup table and (2) we have modified the input path to the ADC to pass low frequencies, avoiding the signal distortion caused by the transformer seen in the ringdowns of Spence *et al.*, which may have contributed to that system's lack of agreement with the spectrophotometer data.[12]

A. The FFT algorithm

The DFT algorithm for determining CRDS decay constants has been described previously.[6,7] The basis of the algorithm comes from the Fourier transform of an exponential decay function with decay constant β, given by (for t > 0)

$$I(l) = I_0 e^{-\varepsilon l c}. \tag{2}$$

The complex Fourier transform is found to be[13]

$$F(\omega) = \frac{1}{\beta + i\omega} = \frac{\beta}{\beta^2+\omega^2} - i\frac{\omega}{\beta^2+\omega^2}. \tag{3}$$

Here, ω is the angular frequency (rad/s), and the transform has been broken into its real and imaginary parts. Taking the ratio of the real and imaginary parts gives a simple expression for the decay constant,

$$\beta = -\omega\frac{\mathrm{Re}(F(k))}{\mathrm{Im}(F(k))}. \tag{4}$$

Thus, the decay constant for an exponential decay can be found using any frequency component of the Fourier transform. The first non-zero frequency component is typically chosen to maximize the signal-to-noise ratio (SNR), since the real and imaginary parts have their maximum at $\omega = 0$ and $\omega = \beta/2\pi$, respectively, and drop to 10% of their peak value in less than $\omega = 5\beta/2\pi$ and $\omega = 20\beta/2\pi$, respectively. This means that for a 10 μs decay time ($\beta = 10^5\ \mathrm{s}^{-1}$), the SNR for each part is an order of magnitude lower at 80 kHz and 320 kHz, respectively. Note that low frequency distortion of the signal can have a significant impact on the time constant (and therefore the analyte concentration) determined by this method.

In practice, a discrete Fourier transform (or a FFT) is applied to digitized data, which results in a small error when using the simple relationship in Eq. (4), but can be corrected by using the ratio in a slightly more complicated equation[7]

$$\beta = \frac{1}{\Delta t}\ln\left[\frac{\mathrm{Re}(F(k))}{\mathrm{Im}(F(k))}\sin\left(\frac{2\pi}{N}k\right) + \cos\left(\frac{2\pi}{N}k\right)\right], \tag{5}$$

where k is the frequency index, N is the total number of time series samples, Δt is the sampling interval (10 ns for a 100 MHz ADC clock), and the frequency f (in Hz) for any k is given by $f = k/(N\Delta t)$. In this work, in order to conserve FPGA resources and increase the processing speed, we only computed the ratio of the real and imaginary parts on the FPGA. This ratio can then be used in either Eq. (4) or Eq. (5) to calculate the time constant on the computer. For this work, we used Eq. (4), resulting in less than a 0.1% difference from Eq. (5), when using the first frequency component ($k = 1$).

B. Hardware configuration

A block diagram of the data acquisition system is shown in Figure 1. The signal from the photodetector that measures the decay of the light from the cavity enters a circuit that level-shifts and attenuates the signal to levels appropriate for the ADC. The signal is then passed to an AD9255 evaluation board (AD9255-105EBZ, Analog Devices, Inc.) containing a 14-bit pipelined, switched-capacitor ADC that can run at sampling rates up to 125 MSPS. The AD9255 evaluation board is designed with a high-speed interface to FPGA-based buffered memory boards (Analog Devices, Inc.). We chose the HSC-EVALCZ, using a Virtex-4 FPGA (Xilinx, Inc., San Jose, CA) that can read up to 16 bits from the upstream evaluation board, of which our ADC supplies only the upper 14 bits. The ADC continuously supplies data to the FPGA (i.e., it is not triggered to start the ADC process) using a synchronous clock to signal

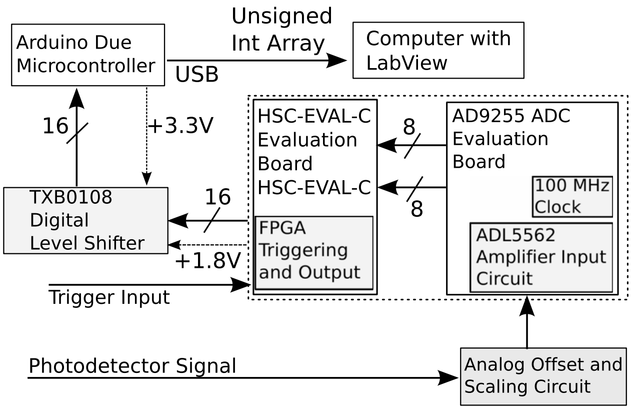

FIG. 1. Block diagram of data acquistion (DAQ) system, whose main components are the AD9255 ADC evaluation board and the co-requisite HSC-EVALC board containing the FPGA. The blocks in gray are the additional circuits or evaluation board modifications discussed in the text.

the FPGA to read in the data. The FPGA board is triggered by a digital signal from the same threshold circuit that induces the ringdown event; in our system, it triggers a pulsed diode laser to shift the main laser off-resonance from the cavity.[14] After acquiring data for a fixed time period (corresponding to ~5 1/e times for the ringdown decay), the FPGA performs a discrete FFT on the ringdown data and divides the first real and imaginary components (i.e., Eq. (4)). The FPGA then sends the 16 most significant bits of the result through a digital level converter to an Arduino Due microcontroller (Arduino, SA, Turin, Italy) for further processing. The Arduino Due stores the values in memory and then transmits them via Universal Serial Bus (USB) to the computer.

The required modifications to the ADC and FPGA evaluation boards are discussed in Secs. I and II, respectively, with circuit details deferred to the Appendix. The description of the digital level shifter is included in Sec. II. The analog scaling and offset circuit are described in Sec. III (with circuit details also in the Appendix).

1. *AD9255 evaluation board modifications*

Most high-speed ADCs require differential inputs (paired signals with equal magnitude and opposite sign). Typically, photo-detectors used for CRDS are single-ended outputs (i.e., referenced to ground). Therefore, in order to make best use of the ADC, a differential signal must be generated from this single-ended signal. The majority of high-speed ADC/FPGA applications are related to communications or signal processing of sinusoidal waves at MHz frequencies and above. We found that as a consequence the evaluation boards are designed for AC-coupled, high frequency input signals, where the key features of a ringdown signal are a relatively low frequency decay followed by a DC level between excitations. The evaluation board's standard input path includes conversion of a single-ended signal to a differential signal using a radio frequency (RF) transformer that is designed to work with alternating signals that, as stated in the AD9255 data sheet, "can saturate at frequencies below a few megahertz."[15] The effect of this is attenuation of low frequencies, or high-pass filtering, and our initial attempts to use the standard input path resulted in ringdown signal distortion similar to that reported by Spence *et al.*[12] Fortunately, the evaluation board contained a secondary input path to the AD9255 with an operational amplifier (ADL5562, Analog Devices, Inc.) capable of converting DC-coupled, low frequency, single-ended signals to the requisite differential signals. Unfortunately, the majority of the alternate input path is unpopulated, and the components that were installed assumed an AC-coupled signal. The modifications needed to make this path functional for ringdown signals are detailed in the Appendix and follow the recommendations given in the data sheets for the AD9255[15] and the ADL5562.[16]

2. *FPGA evaluation board configuration*

In our design, the AD9255 continuously digitizes the input signal, regardless of whether a ringdown has been initiated, and it is the FPGA that is triggered to acquire and store the digitized data and begin the FFT processing. A threshold-triggering circuit enables both the laser unlocking[14] (to initiate the ringdown) and the FPGA data processing. For the FPGA, a solid-state switch and a 1.5-V input created from a simple voltage divider were used since the 5-V TTL levels used for the other parts of our system are incompatible with the 1.8-V logic on the FPGA evaluation board. The FPGA evaluation board has an 18-pin 0.1 in. × 0.1 in. header designated as debug pins which allows access to FPGA inputs. The trigger signal is connected to debug pin 14 (maps to FPGA pin AB7), which is registered as a digital input buffer in the FPGA logic.

The FPGA evaluation board outputs the 16 most significant bits of the ratio of the real and imaginary parts of the first component of the FFT as described in Sec. II C 1. Although the FPGA evaluation board has an onboard USB transceiver for communication with a host computer, a significant number of gates are needed to communicate with the transceiver. Because we opted for a number of quality control/debugging capabilities in this version of the system, those gates were not available, so we chose to use an Arduino Due microcontroller as an interface between the FPGA and the computer. The bottom side of the FPGA board has soldering pads for 3 90-pin 0.05 in. × 0.05 in. dual-line headers, designed to connect to a Digital Signal Processing (DSP) daughter board. Several of these pads are routed to FPGA pins, so we used these to transmit the 16-bit representation of the calculated ratio off the FPGA evaluation board. Additional pins are available on the evaluation board, so it is possible to expand the output by another 8 bits to increase the precision of the ratio, if this approach is adopted. Alternatively, some of the debugging capabilities could be sacrificed to allow for direct connection to a computer via the USB.

The FPGA board uses 1.8 V digital logic while the Arduino Due uses 3.3 V digital logic, so we used two 8-channel bi-directional digital level converters (Adafruit Industries, New York, NY) to provide the Arduino Due with the proper digital levels. These are small daughter boards designed to be mounted in a larger module, so we set them up external to the FPGA board, as shown in Fig. 1. The Arduino and the FPGA evaluation board each supply power for their respective side of the level converter, while the ground is tied to the Arduino. Details of the mapping of the 16 bits of the FPGA calculation result, through the DAQ, and into the Arduino input buffers are given in Table I of the Appendix.

3. Analog offset and scaling circuit

Common mode voltage incompatibility required additional circuitry to avoid distortion of the output signal. Specifically, for "best distortion performance" in DC applications of the ADL5562, the input common mode voltage—the average of the positive (non-inverting) and negative (inverting) inputs—should be in the range 1.0 V–2.3 V.[16] The ringdown is a DC-coupled, ground-referenced signal, and for our system, the maximum voltages were in the range 0.2 V–1.0 V, making the common mode voltage significantly below the recommended range. In addition, as shown in Figure 4 of the Appendix, the output common mode of the ADL5562 amplifier was set to the AD9255 input common mode voltage of 0.9 V, making it also below the recommended range of 1.25 V–1.85 V.[16] Due to these mismatches in input and output common mode voltages, the signal measured by the ADC exhibited distortion, as seen in the ringdown in Figure 2(a). The distortion, most likely attributed to phase reversal[17] of the larger voltages at the beginning of the ringdown, makes a significant portion (the 15% with the largest SNR) of the decay signal unusable. To more properly condition the signal, we first used a simple potentiometer-based voltage divider to reduce the amplitude of our ringdowns to be within the undistorted range (approximately 70 mV or less). We then increased the undistorted range using a subtraction circuit[18] to add a negative offset to the signal, providing a better match to the input signal range of the ADL5562 amplifier. The offset and scaling circuit is described in more detail in the Appendix and can be adjusted to maximize the signal levels and minimize the distortion. A ringdown taken using the circuit is shown in Figure 2(b), and comparison with Figure 2(a) shows the improvement in amplification and digitization fidelity. The resulting ringdown is undistorted, improving the fit (the residual is shown in the inset of each figure), increasing the SNR (since points with larger voltage are available), and increasing the total number of data points used in the fit.

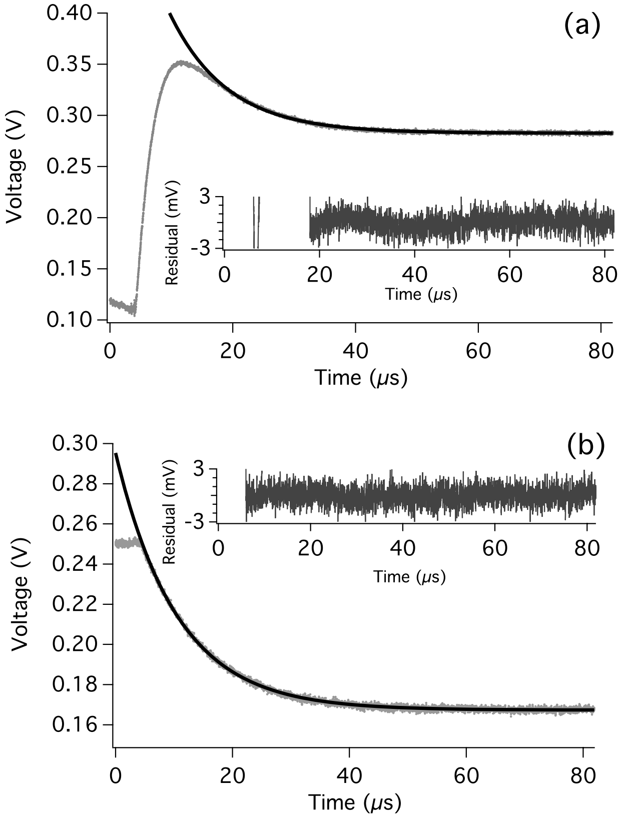

FIG. 2. (a) Comparison of FPGA captured ringdown with and without the ADC front end circuit. (a) Gray line shows the photodetector output (triggered at 440 mV) passed directly to ADC evaluation board, which exhibits distortion due to the low common mode voltage set by the AD9255 amplifier. (b) Gray line shows a signal at the same trigger level passed through the scaling and offset circuit, which eliminates the phase reversal distortion. The front end expands the range of valid data in both voltage and in the number of points available for determining the decay constant. Exponential curve fits (black lines) are shown in both (a) and (b) to highlight the distortion, and the residual to the fit is shown in the inset.

C. Software configuration

1. FPGA coding description

In order to program the FPGA, we used the Xilinx ISE® Design Suite, an integrated development suite for coding, pin assignments, debugging, and loading the FPGA with design. Its use enabled access to the Xilinx LogiCORE™ IP fast Fourier transform core[19] and the LogiCORE™ IP divider generator core.[20] Since the evaluation board had all of the FPGA input/output pins routed, we were constrained by its initial design as to which pins we could use for processing and output. The FPGA code timing was set to the 100 MHz synchronous clock onboard the ADC evaluation board. After the external trigger input buffer is set high (see Sec. II B 2), the code includes a delay from the rising edge of the trigger before starting the data processing. This delay was configured to avoid any transients in the laser switching that may be present in the ringdown signal. In our configuration, an acceptable delay was 1.6 μs. The ADC samples were then loaded directly into the FFT core, and processing begins as soon as the last sample of the ringdown is taken. The FFT core was configured for an 8192-point 16-bit integer transform. At 100 MHz, the 8192 samples capture 81.92 μs of ringdown signal, a collection time long enough to ensure optimal performance of the DFT algorithm (4-5 time constants).[7] To provide maximum transform fidelity, the algorithm was configured as unscaled, meaning that the number of bits at the output expands to 28 bits (due to the multiplications involved) and the least significant bits are not dropped as the information progresses through the algorithm.

Upon completion of the FFT, a DONE flag and a DATA VALID flag are set high, and on each clock cycle, the real and imaginary parts of each frequency component in the transform are output in order. Once the desired component is output ($k = 1$ in this work), the values are latched, and the integer division core is enabled. The division core was configured as a 28-bit by 28-bit division, with the result having a 28-bit integer part and a 14-bit fractional part. The division core required 32 clock cycles to complete, after which the integer and fractional part are latched at FPGA output pins. Given that typical decay constants are on the order of 10^5, we find that the ratio generated should be on the order of 1 (Eq. (4)), meaning that most of the 28 bits of the integer part will be zero. We chose to keep 3 bits of the integer part and 13 bits of the fractional part, providing a maximum time constant of 6.1×10^5 s^{-1} with a resolution of approximately 9 s^{-1} (0.01% for $\beta = 10^5$ s^{-1}). For our CRDS instrument, this is equivalent to a maximum absorption of 0.016 and an absorption sensitivity of 2.6×10^{-8} which, based on the results in Sec. III, do not impact the sensitivity of the rest of the system. Once the division core completes a cycle and calculation, the FFT core is reset, and the FPGA is ready to capture the next ringdown. The ratio result is latched at the output until it is reset by the next processed ringdown, so that the result can be read at a slower rate. The total processing time on the FPGA was measured to be 145 μs. Combining that with the ringdown time of 81.92 μs, the FPGA can process ringdowns at a maximum rate of 4.4 kHz.

2. Arduino Due storage and post-processing

The Arduino Due is an open-source microcontroller with a 32-bit processor, an 84 MHz clock, 54 digital input/output pins, and a USB interface for programming and communication with a computer. We used 16 of the digital input buffers, mapped to the output bits of the FPGA division calculation (see Table I of the Appendix) and conditioned as described in Sec. II C 2. The Arduino reads the digital input as a 16-bit unsigned integer and stores it in memory in approximately 22 μs. This is less than the time to load and process the next ringdown and thus does not form a bottleneck in the processing. The memory limitations on the Arduino restrict the number of stored 16-bit values to approximately 150. The Arduino storage and transmit functions are controlled by commands from the computer via USB. One command initiates the storage of ratio values from the FPGA, and a second command stops the storage and transmits all stored values to the computer (as a series of strings). For this work, we were interested in recording each individual decay constant, but it would be straightforward to have the Arduino Due compute and pass the average to the computer. The 16-bit unsigned integer value, β_{raw}, becomes the ratio required in Eq. (4) by dividing by 2^{13} to move the decimal place and then converting to the decay constant β (in s^{-1}) by multiplying by $\omega_1 = 2\pi/(N\Delta t)$. In our system, $\beta = 2\pi\beta_{raw}/(81.92 \times 10^{-6} \times 2^{13})$, which simplifies to $\beta = 9.3626\beta_{raw}$.

III. PERFORMANCE IN A CRDS SYSTEM

The FPGA processing was incorporated into a CRDS system described in a previous paper,[14] which presents a method of initiating the ringdown by shifting the main laser off cavity-resonance via injection of a pulsed laser ("optical injection unlocking"). The cavity was 85 cm long with a finesse in excess of 10 000. A computer running LabView stepped the main laser current, shifting the laser frequency by 1.06 GHz/mA to the predicted cavity mode and then used a dithering algorithm to find the cavity modes. The LabView code then waited 100 ms (at each cavity mode/main laser frequency) while the FPGA captured and processed ringdowns. At a cell pressure of 500 Torr and temperature of 22 °C, an absorption spectrum of ambient humid air (1.4% H_2O concentration) was obtained by collecting up to 150 samples per frequency point/collection time as shown in Figure 3. The two peaks shown are absorption peaks of H_2O in the near infrared, with the largest centered near 7640.8 cm^{-1}.[21] A Voigt line-shape function was fitted to the two peaks, and the residual to the curve fit is shown. The median relative standard deviation for the fit of the data points was 1.9% (mean RSD was 3.1%). In our system, the instability of the laser-cavity locking resulted in different data collection rates at different cavity modes/frequency positions, but the median number of samples per mode was 150 (mean = 115), meaning that the overall system collected decay constants at a rate of at least 1.5 kHz. This is within a factor of 3 of the maximum capability of the FPGA, limited by the 150 sample Arduino memory maximum. In the figure, the large residuals around the peak are a result of issues with the CRDS system performance resulting from inconsistent cavity locking at higher absorption and not an artifact of the DFT algorithm.[22] Our overall system was designed for high precision measurements at low concentrations of analyte. Therefore, absorption coefficients much larger than the cavity background losses were not expected, which led us to allocate the output bits towards precision and not for maximum detectable absorption. To accommodate larger absorption, modifications to the FPGA code could be made to change the range of the absorption coefficients, increase the number of bits transmitted to the

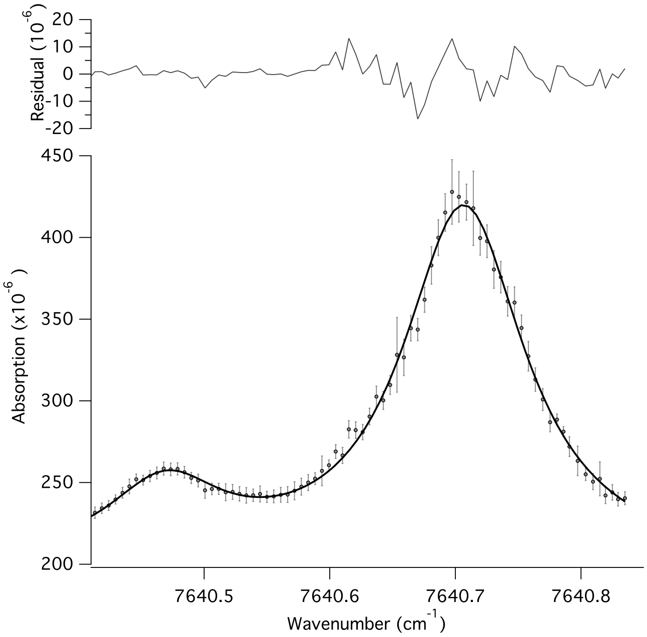

FIG. 3. Near infrared spectrum of H_2O obtained using decay constants calculated on the FPGA. The average of the samples taken at each wavenumber is shown as dots, with error bars indicating ± one standard deviation for that point. Voigt line shapes were applied to each of the two peaks, and the solid line indicates the fit, while the residuals to the fit are shown above the spectrum.

Arduino Due, and/or alter the duration of data collection by reducing the number of data points to reduce overall noise. However, the sampling time of 10 ns has the most impact on the precision for strong absorption (~10 times background) since there would be a significant reduction in the total number of data points collected during the ringdown.[7]

IV. CONCLUSION

We have demonstrated an optimized cost-effective data acquisition system for CRDS that is capable of determining time constants at a maximum rate of 4.4 kHz by modifying a commercial ADC-FPGA evaluation board and programming it to apply the DFT algorithm for determining decay constants. We have provided the details of the modifications with generally available hardware, so that the system could be incorporated into existing CRDS instruments to enable faster data collection and processing. The FPGA code, the Arduino code, and example LabView modules are available upon request from the authors. In this initial version, significant FPGA resources were used for debugging and evaluating/documenting performance, and so, only one FFT component was used for the decay constant. Future implementations would need fewer resources for debugging, which could then be used for improvements to the system. Straightforward modification of the FPGA code to use the average of ratios for the first 5 components would improve the precision, which is designated as the DFT-5 algorithm by Everest and Atkinson.[7] Averaging, pipelining of the FFT core to increase the processing rate, and communication via the on-board USB transceiver could also be implemented with additional FPGA resources.

ACKNOWLEDGMENTS

The authors would like to thank the Xilinx University Program for providing access to the Xilinx ISE Design Suite through a donation to Portland State University.

APPENDIX: HARDWARE MODIFICATION DETAILS

1. ADC evaluation board modifications

A circuit diagram highlighting the ADC board (AD9255-105EBZ) modifications is shown in Figure 4. The rectangles indicate resistors and capacitors that were already included on the board. All others were added based on recommended values for frequencies between 0 and 100 MHz[15] and assume a source impedance of 50 Ω.[16] The names for specific resistors, capacitors, inductors, and jumpers in the following discussion follow the identifications provided in the user's guide for the ADC evaluation board.[23] The resistors and capacitors installed are surface mount devices (SMDs), size 0402 (1 mm × 0.5 mm), and the soldering and rework were done using a hot air rework station. The following pre-installed capacitors were removed: C2, C4, C12, C14, C71, and C96, and 0 Ω resistors were installed at the following capacitor locations: C12, C14, C49, C50, and C51. Capacitor C15 was replaced with an 18 pF capacitor. Resistors R3, R8, R22, and R23 were replaced with 15 Ω resistors. R15 and R16 were removed, and R13 was replaced with a 26 Ω resistor. R54 (0 Ω) and R20 and R21 (1 kΩ) were installed, and locations R18, R19, L8, and L9 were shorted. The 33 Ω resistors in Figure 4 were installed at locations L5 and L6. The board is factory-configured for an external clock input but comes with circuitry for an onboard clock. We installed a compatible 100-MHz clock (OCA3H-C3A-100.000M, Ascend Frequency Devices, Lake Oswego, OR), replaced C70 with a 0 Ω resistor, and removed R25, following the instructions in the user's guide.[23]

On the evaluation board, the solder jumpers (JP1, JP4, and JP8) were shorted to connect the common mode voltage output of the AD9255 to the common mode input of the ADL5562 and the midpoint of its output. JP6 and JP7 were shorted to connect the ADL5562 to the AD9255 input path. Additional

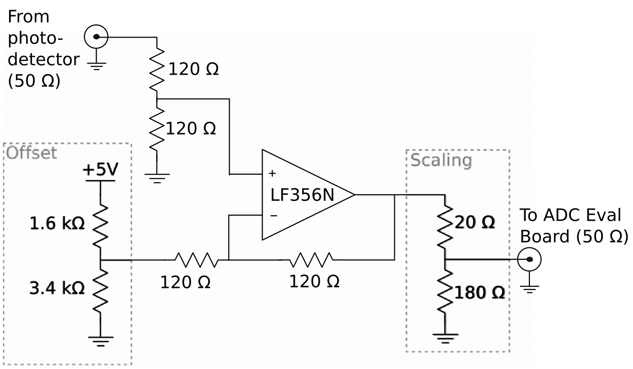

FIG. 5. Diagram of the analog offset and scaling diagram to the AD9255 evaluation board. An adjustable offset is subtracted from the output of the photodetector, and the result is scaled using a voltage divider so that the ringdown is within the voltage limits of the ADC. The resistors in the offset and scaling dashed rectangles are made using the center tap of a 5 kΩ and 200 Ω variable resistor, respectively.

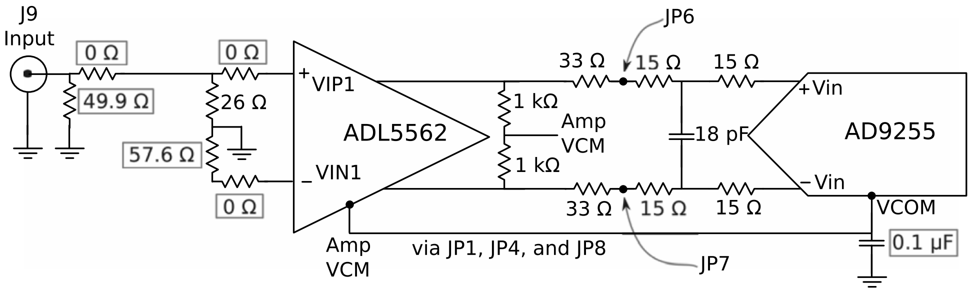

FIG. 4. Circuit diagram of the amplifier input path to the ADC. The rectangles around the labels indicate components that were already on the circuit board and were not modified. The other resistors and the 18 pF capacitors were installed on the board.

TABLE I. Arduino Due to FPGA translation. The names in columns 2-5 correspond to those given in the evaluation board user's guide.[23] The Arduino Due GPIO pins correspond to the labeling on its circuit board.

Re/Im output bit	Evaluation board schematic name	FPGA pin	Evaluation board header pin (P1)	Level shifter pin	Arduino Due GPIO pin
0 (LSB)	DSP_A0	AF15	5	A-A0	2
1	DSP_A2	AF14	7	A-A1	3
2	DSP_A4	Y8	9	A-A2	4
3	DSP_A6	AA13	11	A-A3	5
4	DSP_A8	AA12	13	A-A4	6
5	DSP_A10	AB14	15	A-A5	7
6	DSP_A12	AA10	17	A-A6	8
7	DSP_A14	AB12	19	A-A7	9
8	DSP_A1	AD15	6	B-A0	10
9	DSP_A3	AE15	8	B-A1	11
10	DSP_A5	Y7	10	B-A2	12
11	DSP_A7	AA9	12	B-A3	13
12	DSP_A9	AA14	14	B-A4	22
13	DSP_A11	AC14	16	B-A5	24
14	DSP_A13	AD14	18	B-A6	26
15	DSP_A15	AC13	20	B-A7	28

header pin jumper settings on the ADC board are as follows: P19 left open enables the ADL5562; P5 (SENSE) 1 → 2 sets ADC voltage reference to 0.5 V; installing P6 enables the onboard clock; P4 2 → 3 enables duty cycle stabilizer (DCS); P4 5 → 6 enables two's complement; P14 open enables low voltage CMOS digital output signals to the HSC-EVALC board; and P13 open disables dithering.

2. Analog offset and scaling circuit

The circuit used to scale and provide an offset to the photodetector output (to eliminate the signal distortion due to the common mode voltage phase reversal) is shown in Figure 5. We used a simple subtraction circuit,[18] based on a LF356N operational amplifier. A potentiometer between +5 V and ground creates a voltage divider allowing us to vary the subtracted DC voltage. The scaling is accomplished with another voltage divider, which outputs a fraction of the offset photodetector signal to the ADC evaluation board. The potentiometers are adjusted to ensure the ADC input is in the proper range to maximize fidelity of conversion while preventing signal distortion.

3. FPGA board modifications and pin mapping

To transmit the result computed by the FPGA, we used one of the three solder pads designed for connecting a DSP daughter board through 90-pin 0.05 in. × 0.05 in. dual-line headers. We soldered a 28-position header (FTSH-114-01-L-DV, Samtec, Inc., New Albany, IN) to pins 3 through 32 of the P1 pad.[23] A cable assembly (SFSD-15-28-H-10.00-SR, Samtec, Inc.) with a corresponding 30-position socket on one end and bare wires on the other was then used to connect the FPGA output pins to the 0.1 in. × 0.1 in. headers on the digital level shifter (see Figure 1). Table I maps the data lines corresponding to the bits of the FPGA ratio calculation to the pins of the FPGA board and the Arduino microcontroller. These mappings correspond to our code written for the FPGA and Arduino.

[1] L. Rothman, C. Rinsland, A. Goldman, S. Massie, D. Edwards, J. Flaud, A. Perrin, C. Camy-Peyret, V. Dana, J. Mandin *et al.*, J. Quant Spectrosc. Radiat. Transfer **60**, 665 (1996).

[2] (n.d.).

[3] D. B. Atkinson, Analyst **128**, 117 (2003).

[4] B. A. Paldus and A. A. Kachanov, Can. J. Phys. **83**, 975 (2005).

[5] D. Romanini, A. A. Kachanov, N. Sadeghi, and F. Stoeckel, Chem. Phys. Lett. **264**, 316 (1997).

[6] M. Mazurenka, R. Wada, A. Shillings, T. Butler, J. Beames, and A. Orr-Ewing, Appl. Phys. B **81**, 135 (2005).

[7] M. A. Everest and D. B. Atkinson, Rev. Sci. Instrum. **79**, 023108 (2008).

[8] T. G. Spence, C. C. Harb, B. A. Paldus, R. N. Zare, B. Willke, and R. L. Byer, Rev. Sci. Instrum. **71**, 347 (2000).

[9] A. M. Mood, *Introduction to the Theory of Statistics*, 1st ed. (McGraw-Hill, New York, 1950).

[10] K. K. Lehmann and H. Huang, *Frontiers of Molecular Spectroscopy*, edited by J. Laane (Elsevier, Amsterdam, 2009), pp. 623–658.

[11] B. Paldus, C. Harb, T. Spence, B. Wilke, J. Xie, J. Harris, and R. Zare, J. Appl. Phys. **83**, 3991 (1998).

[12] T. G. Spence, M. E. Calzada, H. M. Gardner, E. Leefe, H. B. Fontenot, L. Gilevicius, R. W. Hartsock, T. K. Boyson, and C. C. Harb, Opt. Express **20**, 8804 (2012).

[13] R. N. Bracewell, *The Fourier Transform and Its Applications*, 2nd ed. (McGraw Hill, New York, 1986).

[14] G. Bostrom, A. Rice, and D. Atkinson, Opt. Lett. **39**, 4227 (2014).

[15] 14-Bit, 125 MSPS/105 MSPS/80 MSPS, 1.8 V Analog-to-Digital Converter, AD9255 Data Sheet, Revision C (Analog Devices Inc., 2013).

[16] 3.3 GHz Ultralow Distortion RF/IF Differential Amplifier, ADL5562, Rev. E (Analog Devices, Inc., 2014).

[17] A. D. I. Engineering, *Linear Circuit Design Handbook* (Newnes, 2011).

[18] J. W. Nilsson and S. Riedel, *Electric Circuits*, 9th ed. (Prentice Hall, Boston, 2010).

[19] *LogiCORE IP Fast Fourier Transform v7.1 Product Specification, DS260* (Xilinx Inc., 2011).

[20] *LogiCORE IP Divider Generator v3.0 Product Specification, DS530* (Xilinx Inc., 2011).

[21] L. S. Rothman, D. Jacquemart, A. Barbe, D. Chris Benner, M. Birk, L. R. Brown, M. R. Carleer, C. Chackerian, K. Chance, L. H. Coudert *et al.*, J. Quant. Spectrosc. Radiat. Transfer **96**, 139 (2005).

[22] G. Bostrom, "Modifications to a Cavity Ringdown Spectrometer to Improve Data Acquisition Rates," Ph.D. thesis (Portland State University, 2015).

[23] *Evaluation Board User Guide: Evaluating the AD9265/AD9255 Analog-to-Digital Converters, UG-074* (Analog Devices Inc., 2011).