Journal of Low Temperature Physics (2018) 193:485–497
https://doi.org/10.1007/s10909-018-2012-2

A Scalable Readout for Microwave SQUID Multiplexing of Transition-Edge Sensors

J. D. Gard[1] · D. T. Becker[1] · D. A. Bennett[2] · J. W. Fowler[1,2] · G. C. Hilton[2] · J. A. B. Mates[1] · C. D. Reintsema[2] · D. R. Schmidt[2] · D. S. Swetz[2] · J. N. Ullom[1,2]

Received: 6 November 2017 / Accepted: 25 June 2018 / Published online: 9 July 2018

Abstract
The readout requirements for instruments based on transition-edge sensors (TESs) have dramatically increased over the last decade as demand for systems with larger arrays and faster sensors has grown. Emerging systems are expected to contain many thousands of sensors and/or sensors with time constants as short as 100 ms. These requirements must be satisfied while maintaining low noise, high dynamic range, and low crosstalk. A promising readout candidate for future TES arrays is the microwave SQUID multiplexer, which offers several gigahertz of readout bandwidth per pair of coaxial cables. In microwave SQUID multiplexing, sensor signals are coupled to RF-SQUIDs embedded in superconducting microwave resonators, which are probed via a common microwave feedline and read out using gigahertz signals. This form of SQUID multiplexing moves complexity from the cryogenic stages to room temperature hardware and digital signal processing firmware which must synthesize the microwave tones and process the information contained within them. To demultiplex signals from the microwave SQUID multiplexer, we have implemented an FPGA-based firmware architecture that is flexible enough to read out a variety of differently optimized TESs. A gamma-ray spectrometer targeted at nuclear materials accounting applications, known as SLEDGEHAMMER, is an early adopter of microwave SQUID multiplexing and is driving our current firmware development effort. This instrument utilizes 300 kHz full-width half-maximum resonators with 256 channels in a one gigahertz wide band. We have recently demonstrated undegraded readout of 128 channels using two ROACH2s on a single pair of coaxial cables. This manuscript describes the firmware implementation for the readout electronics of these early array-scale demonstrations.

Keywords Transition-edge sensors · Firmware · Multiplexing

Extended author information available on the last page of the article

1 Introduction

Microwave SQUID multiplexing has shown significant promise as a scalable readout for the next generation of cryogenic bolometers and microcalorimeters. Small-scale demonstrations have shown that microwave SQUID multiplexing can reach sufficiently low readout noise levels to achieve intrinsic energy resolutions in transition-edge sensor (TES) microcalorimeters [1] and detector-noise-limited noise equivalent power in TES bolometers [2]. These early demonstrations utilized a homodyne readout architecture that is difficult and expensive to scale beyond a handful of channels. Two projects, MUSTANG-2 and SLEDGEHAMMER [3,4], have developed scalable room temperature electronics capable of reading out small arrays of transition-edge sensors using microwave SQUID multiplexing. MUSTANG-2 is an instrument on the Green Bank Telescope consisting of an array of 215 TES bolometers measuring 90 GHz radiation [5]. SLEDGEHAMMER is a prototype high-resolution energy-dispersive gamma-ray spectrometer consisting of an array of 512 TES microcalorimeters. Both instruments utilize the same multiplexer chip and similar room temperature electronics. The MUSTANG hardware and firmware, described elsewhere, are optimized for lower bandwidth TES bolometers. The SLEDGEHAMMER hardware and firmware are optimized to read out the pulses from TES microcalorimeters while maintaining their intrinsic energy resolution. This manuscript describes the optimization and implementation of room temperature electronics for the readout of the SLEDGEHAMMER instrument.

In simplest terms, a single microwave SQUID multiplexer channel consists of an RF-SQUID coupled to a superconducting resonator with a resonance frequency in the microwave regime. Changes of the magnetic flux in the input of the SQUID change the Josephson inductance of the RF-SQUID and therefore the inductance loading the resonator. The changing inductive load of the RF-SQUID causes a shift in the resonant frequency. The multiplexer is formed by coupling many individual resonator channels to a common microwave feedline. The room temperature electronics must perform the task of simultaneously probing and recording the frequency shift of each resonator. In order to linearize the response of the RF-SQUIDs and boost the signals above the $1/f$ noise in the superconducting resonator [6], the signal from each channel is modulated using flux-ramp modulation [7]. By providing a linear current ramp of an integer number of flux quanta (n_Φ) to a common flux bias line, all the channels can be modulated to a frequency which is the product of the flux-ramp rate and n_Φ.

Tracking the frequency shift of the microwave SQUID multiplexers is similar to tracking the changes to the resonant response of microwave kinetic inductance detectors (MKIDs). The readout described in this work, along with other early prototype microwave SQUID multiplexer readouts [5,8], leveraged hardware and firmware developed by the MUSIC [9] and ARCONS [10] collaborations for different MKID-based instruments. These platforms all use the Reconfigurable Open Architecture Computing Hardware (ROACH) platform that was developed and is maintained by the Collaboration for Astronomy Signal Processing for Electronics Research (CASPER) [11]. The readout described here uses the MKID DAC/ADC board with two ADC channels capable of 550 MS/s and two DAC channels capable of 1000 MS/s and the

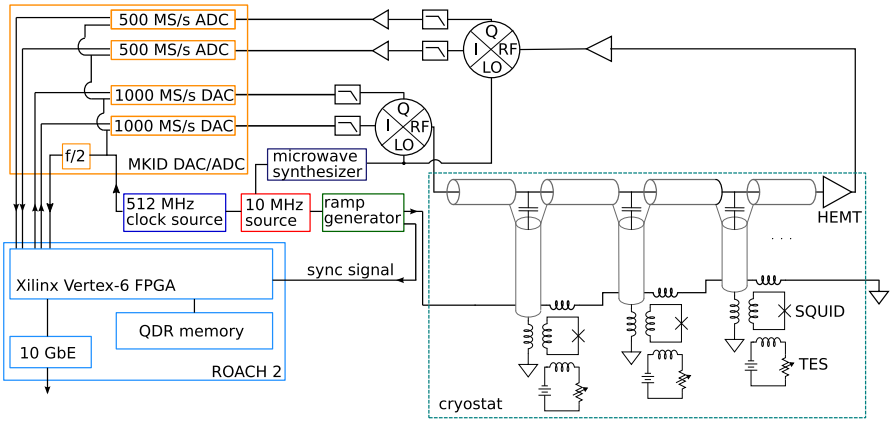

Fig. 1 System schematic of the scalable microwave SQUID multiplexing readout electronics used for validation of the firmware described in this manuscript and in an array-scale TES microcalorimeter demonstration [3]. The portion inside the cryostat, indicated by the cyan dashed box, has been simplified for clarity (Color figure online)

second-generation ROACH board (ROACH2) [11] that features a Xilinx Virtex-6 FPGA.[1]

2 Hardware Setup

The system setup for validation of the hardware and firmware is shown in Fig. 1. The cyan box labeled "cryostat" contains a simplified schematic of the microwave SQUID multiplexer chip and the cryogenic HEMT amplifier. A detailed description of the cryogenic setup and details of the multiplexer chip are beyond the scope of this manuscript and can be found elsewhere [2]. Instead, the focus of this manuscript is the firmware and related system hardware. This section gives a broad overview of how the different hardware parts relate to each other to clarify detailed discussion in the following sections.

The measurement system uses an IQ modulator to mix the digital tones generated by the DAC up to 4–8 GHz and an IQ demodulator to mix the signal transmitted through the multiplexer back down to frequencies below the Nyquist frequency of the ADCs. The microwave synthesizer provides the local oscillator (LO) to the IQ modulator and demodulator. The amplifiers between the demodulator and the ADCs are necessary to efficiently utilize the full input range of the ADC. The low-pass filters are used to remove out of band mixing products.

The ROACH2 platform is schematically indicated by the light blue box. The components relevant to the firmware discussion in the next section are indicated by the smaller light blue boxes. The MKID DAC/ADC card, shown in orange, is connected to the ROACH2 board though 2 connectors each with 40 differential pairs (Z-DOK +).

[1] Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

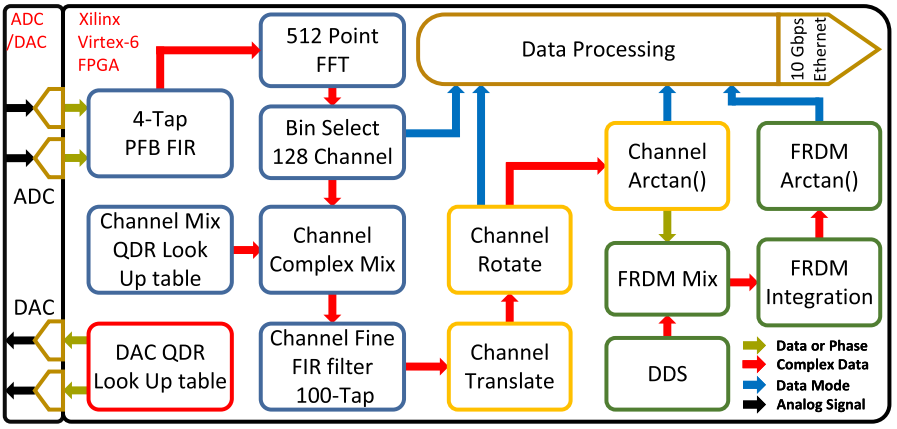

Fig. 2 The firmware flow diagram highlighting the major functional blocks. The box color for the major functional blocks indicates to which firmware task each block belongs; Sect. 3.1 tone generation (red), Sect. 3.2 channelization (blue), Sect. 3.3 arc-angle measurement (yellow), Sect. 3.4 flux-ramp demodulation (FRDM) (green), and Sect. 3.5 data handler (brown) (Color figure online)

A 512 MHz clock generator is used to provide the clock for the DACs and ADCs and is divided down to 256 MHz on the DAC/ADC board before it is routed to the FPGA on the ROACH2. An external arbitrary waveform generator (ramp generator) provides the flux-ramp modulation signal [7]. The synchronization signal from the ramp generator is routed to the FPGA and used by the firmware for synchronization of the flux-ramp demodulation algorithm. The clock source, microwave synthesizer providing the LO, and the ramp generator are all phase locked to a 10 MHz source, usually a Rubidium frequency standard.

3 Firmware Implementation

In this manuscript, we describe our firmware implementation of software-defined radio readout for the microwave SQUID multiplexer. We lay out the algorithm: from tone generation to channelization, to arc-angle measurement, and finally to flux-ramp demodulation. We specifically emphasize our 128-channel implementation, which has been used to demonstrate 128-channel readout of the SLEDGEHAMMER experiment [3], and was designed to be scalable to larger numbers of channels. Figure 2 shows a block diagram of the 128-channel firmware. The following description of the firmware implementation is divided into major tasks as called out by the box color groups in Fig. 2.

3.1 Tone Generation

The first task required of the firmware is the generation of a large number of tones at baseband that can then be mixed up to microwave frequencies and used to measure the microwave SQUIDs. This can be accomplished many ways. Two methods are

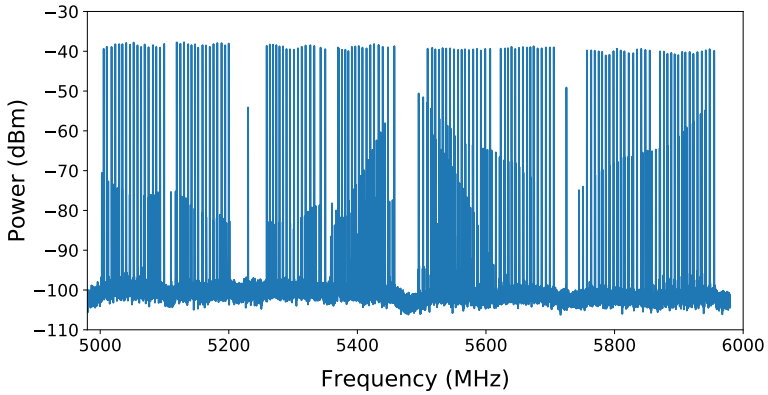

Fig. 3 128 tones produced by two separate ROACH2s each making 64 tones. The 128-channel firmware is producing the tones, but due to DAC/ADC bandwidth limitations two sets of hardware were used to fill 1 GHz of bandwidth. Newer multiplexer chips have been demonstrated that contain 128 resonators in 500 MHz of bandwidth allowing one set of hardware to read out 128 resonators. Looking closely we can see a series of image tones that were not fully nulled out by phase adjustments, below about -50dBm. The leakage of the two separate local oscillator tones can be seen at $\sim$ 5250 and $\sim$ 5725 MHz (Color figure online)

described here: 1) direct digital synthesis (DDS) and 2) playback of a pre-synthesized waveform out of the quad data rate (QDR) memory.

Direct digital synthesis utilizes standard firmware blocks to generate sine waves as a function of input angle. It can therefore be rapidly adjusted by updating the register that stores the angular increment between clock cycles. The bit depth of this register determines the frequency resolution of tone placement.

In order to produce clean single-sideband tones using an IQ mixer, the relative phase between the I and Q channels must be adjusted on a per tone basis to compensate for the frequency-dependent deviations from the ideal phase difference of 90 degrees at the mixer. Without the ability to compensate for the non-ideal phases to at least 1 degree (0.1 degrees), more than -35 dB (-55 dB) of the tone power can end up in an image tone on the other side of the local oscillator (LO), where it has the potential to interfere with other channels. This than imposes the requirement that the phase between I and Q be adjustable for every tone. Four DDS blocks per tone are required to allow phase adjustment between the I and Q channels, as well as to provide two values per FPGA clock cycle (256 MHz) to each DAC (512 MHz). The phase adjustment that is applied comes from a separate calibration procedure [12].

QDR playback is less expensive in terms of FPGA resources and is scalable to larger numbers of channels. To generate a large number of tones, we simply calculate the sum of those tones on a separate computer and then load the resulting waveform into the QDR memory. The drawback to this method is the slower update time for changing a single tone, since the entire waveform must be recalculated and transferred. The length of the waveform determines the maximum frequency resolution of tone placement. Because the I and Q streams are inherently independent, the QDR playback method satisfies the requirement for phase calibration of generated tones.

The 128-channel firmware uses the QDR playback technique to generate its full spread of tones. The current implementation uses two blocks of QDR memory, one for the I DAC and one for the Q DAC, which yields a lookup table length of 2^{21} or about

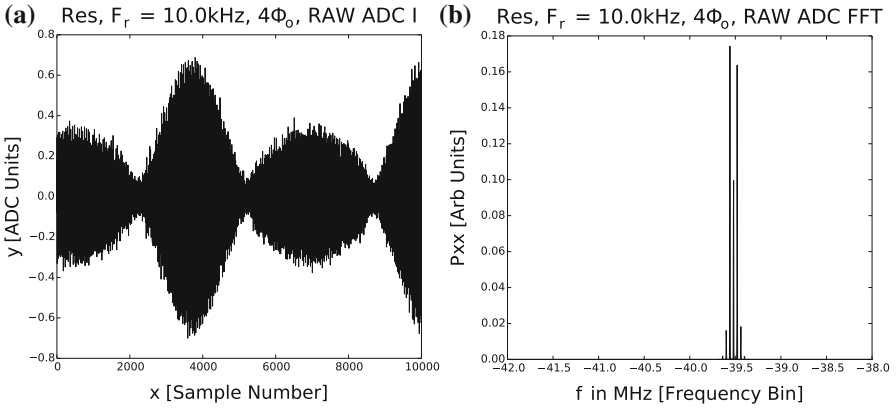

Fig. 4 Signal from an example probe tone being modulated by a resonator in a flux-ramped microwave SQUID multiplexer, shown as **a** the time stream from one ADC and **b** its Fourier transform with the sideband tones produced by the resonator modulation

two million entries. However, it maintains a set of DDS blocks that can be switched in place of the QDR to generate a single probe tone for rapid tuning and calibration procedures. Figure 3 shows an example spectrum of 128 tones generated from this firmware after applying this phase adjustment. Looking closely we can see a series of image tones that were not fully nulled out by phase adjustments. Further reduction in image tone power requires fine-tuning the amplitude of the I and Q signals. With careful tuning it is possible to reduce the image tones below the noise floor of the spectrum analyzer for every generated tone but the process can be very time-consuming.

3.2 Channelization

The data streams from the I and Q ADCs contain information on all of the interrogated resonators in the ADC bandwidth. Before we attempt to demodulate them to extract the detector signals, we must first split them into different data channels in the firmware. This process is called channelization and can be performed by multiple techniques. Two different methods that are described here are: (1) direct digital downconversion (DDC) and (2) using a polyphase filter bank (PFB) in a two-stage process. Figure 4 shows the time stream and Fourier transform of a single probe tone that has been modulated by a flux-ramped resonator and digitized by the ADCs. Since only one probe tone is turned on, the modulation can be clearly seen in the time series.

Direct digital downconversion takes the data stream and downmixes it with various reference tones generated by DDS blocks, two per channel to account for I and Q (actually four because the ADC operates at twice the clock rate of the FPGA). Each channel, now mixed to DC, is low-pass filtered with a decimating finite impulse response (FIR) filter to reject all other channels and then multiplexed into a serial stream. This technique is rapidly updatable but expensive in resources due to the resources needed by each filter.

The process of decimating the data stream can cause tones to be aliased into the passband of other channels. The level to which the stopband of the FIR filter needs

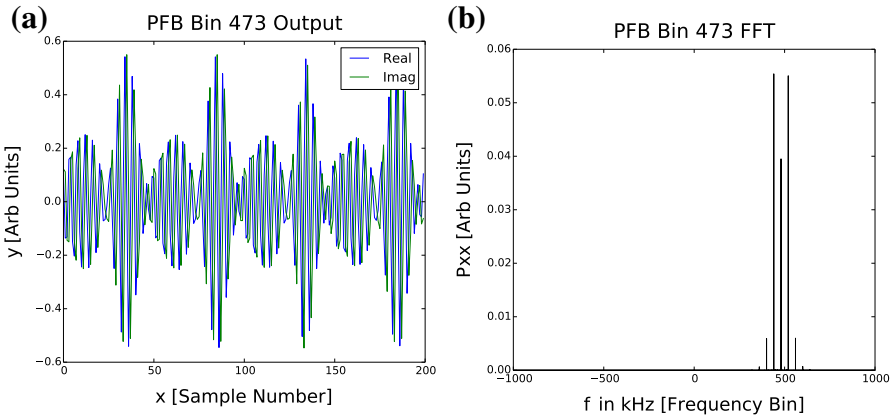

Fig. 5 Same signal as Fig. 4 after the coarse channelization stage of the PFB in **a** time domain and **b** frequency domain representations (Color figure online)

to suppress these out of band aliased tones depends on the details of the microwave multiplexer and the desired application. For the SLEDGEHAMMER instrument, simulations showed that this crosstalk mechanism would give a pulse height error of less than a part in ten thousand if the digital FIR filter had at least 60 dB of suppression out of band. In order to achieve this suppression while maintaining a sharp stop edge at 1.6 MHz, 256 taps would need to be used in the filter. A decimation factor of 64 yields a sampling rate of 4 MS/s per channel. The internal decimation within the FIR filter plays a key role in decreasing the required number of FPGA resources needed relative to the number of taps in the filter.

PFB channelization consists of two steps: a coarse channelization using the polyphase filter bank to divide the data stream into Fourier bins, and a fine channelization that applies a direct digital downconversion to center each measurement tone in its Fourier bin. Our 128-channel firmware uses this technique to channelize the data from the multiplexer. The specific implementation is modeled after the PFB channelizers of ARCONS [10] and CASPER [13].

For a given Fourier transform size, this technique first applies a window function over a period of data that is a multiple of the number of samples required for that Fourier transform size. The multiple sections are then summed and transformed to Fourier space using the fast Fourier transform (FFT). This is a computationally efficient way to apply both the Fourier transform and a filter/window to reduce scalloping and leakage between the bins. Depending on the window being applied, the PFB technique also has the effect of widening the pass band of a bin. In order to prevent aliasing, a second FFT is operated in parallel and out of phase by half of the FFT size. This increases the effective sampling speed of a single bin by two to resolve the bin's full passband.

Once the data from the ADCs have been processed by the PFB, bins must be mapped onto channels. Bin width is set so that no two resonances will fall within the same bin, and thus, the majority of bins contain no tones to read out. A bin selector block which contains a user-supplied map of bins to channels multiplexes the selected channels

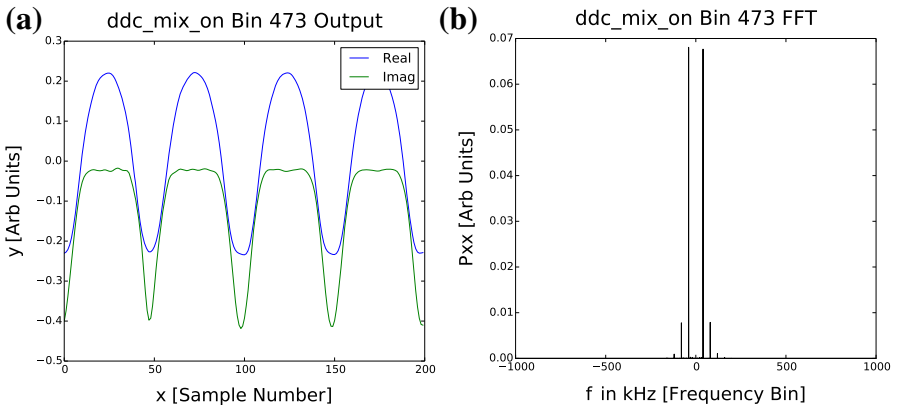

Fig. 6 The same signal as Fig. 4, but after the full channelization chain has mixed it down to dc shown in both **a** the time domain and **b** the frequency domain (Color figure online)

from each FFT into a serial data stream. A 2 MHz sampling rate after the PFB allows 128 bins to be mapped onto channels to form a serial pipeline for later stages of the firmware. Figure 5 shows the result of this process and shows the same probe tone from Fig. 4 in its Fourier bin after passing through the PFB.

To complete the channelization for the PFB method, the tones selected for readout must be brought to the centers of their respective bins. This is accomplished by direct digital downconversion, mixing with reference signals loaded in the another piece of QDR on a per channel basis. The baseband data are then filtered to reject spurious peaks outside the bandwidth of the resonator. Even though the windowed response of the PFB technique relaxes the requirement of 60 dB of stopband attenuation or more, this attenuation was carried into the fine FIR filter. In order to achieve a 60 dB stopband attenuation with a sharp cutoff edge, 100 taps were used for the fine FIR filter. Figure 6 shows the result of the full channelization chain.

If you recall, the generation and upload time of the QDR memory is relatively slow. The firmware therefore retains a set of DDS blocks to take the place of the QDR memory for DAC tone generation and the QDR memory for the fine channelization. This allows the readout of a single channel through all steps of the firmware but with the ability to quickly adjust probe tone frequencies and tone power and is used for calibration procedures.

3.3 Arc-Angle Measurement

Magnetic flux in a microwave SQUID causes its microwave signal to traverse an arc on a resonance circle, as shown in Fig. 7. We therefore transform into coordinates of radius and angle from the center of the resonance circle. We can discard the radius because the amplitude carries minimal information. To perform this coordinate transformation, we first translate the center of the circle to the origin, then rotate the arc to the positive x-axis, and finally take the arctangent of the result using a Coordinate Rotation Digital

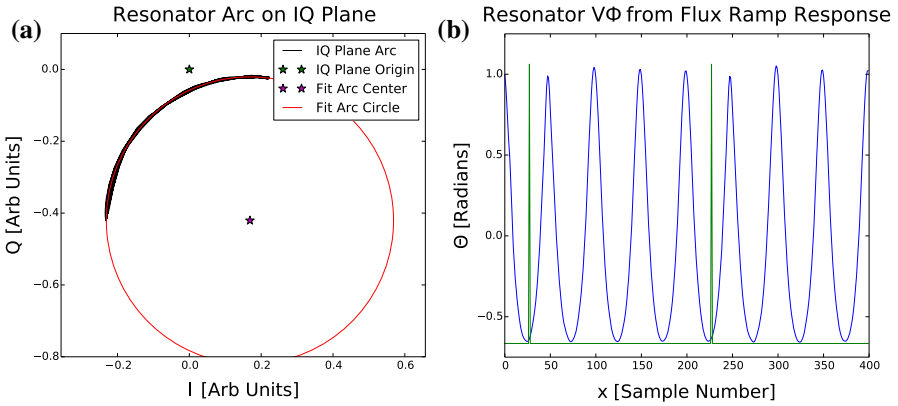

Fig. 7 **a** The resonator arc in the I/Q plane. The I/Q origin, calculated arc center, and fitted circle are shown for reference. **b** The extracted arc angle of the resonator, often called the SQUID V-Φ curve or flux-ramp response. The green spikes are the sync signal from the flux-ramp generator which will be used later in the firmware (Color figure online)

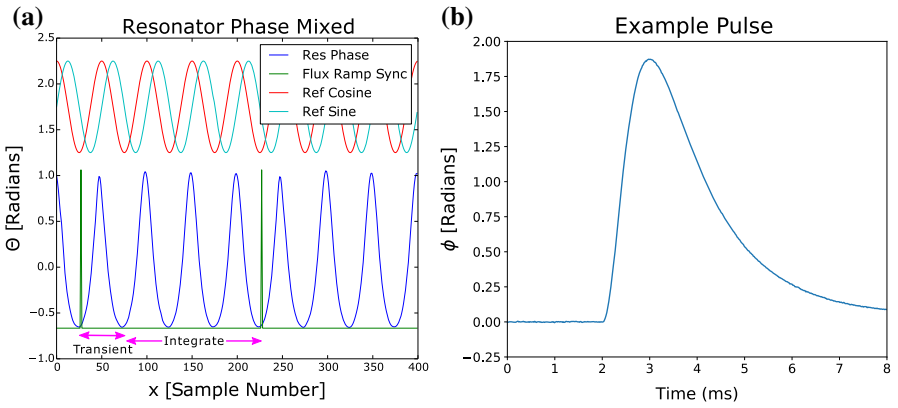

Fig. 8 **a** Two representative flux-ramp periods and the sinusoids that will be used to demodulate the flux-ramp modulation. The sync signal from the flux-ramp generator is shown in green. **b** The output of the flux-ramp demodulation corresponding to a photon absorbed in the TES. The phase angle has been unwrapped in software to account for wrapping at integer values of a Φ_0 (Color figure online)

Computer (CORDIC) algorithm [14]. The resulting angle is analogous to the voltage of a conventional SQUID V-Φ curve.

In the 128-channel firmware, each channel has a pair of registers that specify the (x_0, y_0)-coordinates of the center of its resonance circle, determined by a software fitting algorithm. The following non-unitary transform performs the translation and rotation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -x_0 & y_0 \\ -y_0 & -x_0 \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) \tag{1}$$

To measure the angle, we use a standard Xilinx implementation of a CORDIC, a highly efficient way to compute standard trigonometric functions like the arctangent in an FPGA.

3.4 Flux-Ramp Demodulation

The last task of the firmware is to demodulate the phase-modulation imparted by the common flux-ramp signal. We use a simple algorithm that mixes the arc angle from the previous step with an integral number of oscillations of sine and cosine waveforms at the modulation frequency. These products are then accumulated over a portion of the flux-ramp period to produce one point of scientific data for each flux-ramp period.

The demodulation process is a firmware implementation of the algorithm detailed in Mates et al. [7]. This algorithm detects the phase of the sinusoidal component of the SQUID response with a simple Fourier method:

$$\phi = \arctan\left(\frac{-\sum \theta_t \sin \omega_c t}{\sum \theta_t \cos \omega_c t}\right) \tag{2}$$

$$\frac{\Phi}{\Phi_0} = \frac{\phi}{2\pi} \tag{3}$$

Figure 8a shows the resonator arc angle, θ and the DDS generated sinusoids that it will be mixed with. The sine and cosine waveforms must be identical for every flux-ramp period for each channel. This is accomplished using DDS blocks which are reset by the sync signal from the flux-ramp generator every flux-ramp period. Since the flux-ramp reset causes a transient deformation in θ, we do not want to include that portion of the mixing products in the accumulation.

A state machine that controls a BRAM-based accumulator performs the integration. This state machine controls what section of the flux-ramp response is being integrated over, sends the accumulated terms to the next firmware block, and zeros the accumulator for the next flux ramp. A second CORDIC arctangent computes the phase angle, effectively measuring the magnetic flux imparted by the input coil modulo Φ_0. The demodulated phase angle for a gamma-ray photon absorption in the TES is shown in Fig. 8b. Since this pulse excursion goes beyond the Φ_0 boundary, it has been unwrapped in software.

3.5 Data Handler

The firmware has been designed to allow the data to be tapped off at a variety of points for setup, characterization, and debugging. Each data mode can be streamed continuously over the 10 gigabit Ethernet. The six data modes for the 128-channel firmware are: 1) co-added raw ADC, 2) bin to channel selection 3) post-FIR filter, translation, and rotation, 4) arc-angle (first arctangent), 5) flux-ramp demodulation mix, and 6) flux-ramp demodulation result (the second arctangent). The ability to inspect the data at multiple points in the algorithm has proven critical for development of the readout and for characterization of the microwave SQUID multiplexer. The possible data rates range from one gigabyte per second for the co-added ADC stream down to the flux-ramp repetition rate multiplied by the number of channels and two bytes per sample for the final science data rate. In typical operation only three data modes are used: co-added ADC to determine how much the input is clipping on the

Table 1 FPGA resource usage: comparison of the number of resources used inside the FPGA of the two firmware architectures, DDC-based 32-channel firmware versus PFB-based 128-channel firmware

	32-Chan		128-Chan		Available
Register	128.9 k	21.65%	45.8 k	7.69%	595.2 k
LUT	105.3 k	35.45%	31.8k	10.70%	297.0 k
BRAM	332	15.60%	365	17.15%	2128
DSP	526	26.09%	339	16.81%	2016

ADC, fine FIR output to view the resonator arcs and fit them for center and rotation information, and the flux-ramp demodulation arctangent which is the scientific data.

3.6 Synchronization

Two signals pass throughout the firmware and are necessary for proper synchronization of the various blocks. The first is a channel index which matches the channel number to the appropriate data sample and is used for memory indexing in multiple parts of the firmware. In the 128-channel firmware it is sourced from a counter in the bin selection block and is synchronized to the FFT frame. The second signal is the flux-ramp synchronization signal, acquired from the external function generator that is generating the flux ramp. The flux-ramp synchronization signal can be delayed within the firmware to align to the beginning of the flux ramp as observed in the data stream; Fig. 7 shows this alignment. This allows the blanking period in the FRDM integration to overlap with the transient to the flux-ramp reset.

4 Resource Utilization

FPGA resource utilization is important for future scalability to larger numbers of channels. Table 1 shows a comparison of a 32-channel firmware using DDS to generate the tones and DDC for channelization versus the 128-channel firmware that uses playback from a QDR memory to generate the tones and a combined PFB and DDC for channelization. The resource types available and noted in the table are lookup tables (LUT), registers, internal block memory (BRAM), and digital signal processing slices (DSP). Lookup tables implement the desired logic operations inside the FPGA: and, or, nor, xor, and so forth. Registers, also known as flip-flops, are used to synchronize logic signals inside an FPGA and one or more are tied to the output of a single lookup table. Internal block memory is a small section of random access memory inside the FPGA. In the case of the FPGA on the ROACH2, it contains 18 kilo-bit blocks of memory to be used to hold information. The last form of resource, and one of the most important for scalability, is the digital signal processing block. This block holds a dedicated chain of components; pre-adder, multiplier, and adder that is heavily used in and optimized for FIR filters and Fourier transforms.

The scaling for the 32-channel DDC architecture is fairly straightforward. The resource usage largely depends on the number of taps in the FIR filter and how much decimation is present in the FIR filter. Additional resources are used by the later steps

in the firmware but are minor in comparison. The resource use of the PFB technique is substantially different in comparison. There are three dominant resource use factors in the PFB-based firmware (128-channel): the number of periods in the FFT window, the number of FFT bins, and the number of taps in the fine FIR filter. The number of FFT bins determines how much block memory is needed for the PFB FIR and the FFT itself. In the current implementations of the PFB firmware, each tap in the fine FIR takes one multiplier and one block RAM. By subtracting the DSP slices being used by the fine FIR filter, we can get a general idea of how many resources are used by the PFB. The filter uses a DSP slice for each tap and uses 100 taps for the I data stream and 100 taps for Q data stream. Since the firmware uses a total of 339 DSP slices, only 139 of the DSP slices are needed for the PFB and a few other steps in the firmware. This yields a substantially better ratio of number of channels processed to DSP slices used compared to the 32-channel DDC-based firmware. On the Virtex-6 FPGA, the most stringent limit on the number of channels that can be processed is set by the number of DSP slices occupied by the firmware.

The width of the various data buses routed around the firmware is also important factor for resource utilization and achieving timing closure. The goal at each stage was to use enough bits to not add noise to the sensor signals but not more bits than necessary. Since most mathematical operations in the firmware add less than a single bit of quantization noise, a bit or two was added to the data at each major step in the firmware. The input to the PFB FIR block is the raw 12 bits from the ADC, while the output going to the FFT is 14 bits. Four bits were added for the FFT, one bit for the complex multiply that mixes the signal down to DC, and one bit for fine FIR filter, bringing the total to 20 bits at the output of the channelizer. A bit each was added for the translate and rotate block. Therefore, the input to the first arctan CORDIC was 22 bits each for the I and Q channels. Simulation showed that once the phase angle is calculated, only 13 effective bits are needed to fully resolve the angle of the resonator.

The flux-ramp demodulation has data coming in at 2 MHz per channel but going out at the flux-ramp rate. Therefore, special attention needs to be paid to the bit widths in the demodulation firmware. Two bits are added for the mixing with the sine and cosine. Due to the nature of the operation of the block RAM accumulator, the bit depth is increased significantly to 42 bits. The final output from the second arctan is only 16 bits. This ratio of input bits to output bits on the CORDIC requires us to enforce the number of iterations done by the CORDIC. In this case the number of iterations was matched to the input bit width, 42. Both, 32-channel and 128-channel versions of firmware followed this methodology for the bit widths. In the case of the 32-channel firmware using only DDC blocks, the output was matched to the output of the fine FIR filter, 20 bits. Between the two firmwares, everything from the output of channelization to the final arctan is identical.

5 Conclusion

We have successfully implemented firmware for microwave SQUID multiplexed readout of a 128 channels per ROACH2 unit. This firmware has enabled early array-scale demonstration of microwave SQUID multiplexing in the SLEDGEHAMMER instru-

ment. Given the current resource utilization, the firmware should be scalable to larger numbers of channels. The current limiting factor to increasing the multiplexing factor is the available bandwidth of the DACs and ADCs. We are currently pursuing hardware architectures that will provide 4 GHz of total bandwidth and allow over a thousand channels per pair of coaxial cables.

Acknowledgements We gratefully acknowledge support from the U.S. Department of Energy Nuclear Energy University Program and the Basic Energy Science Sciences Accelerator and Detector Research Program. We would also like to thank the National Institute of Standards and Technology Innovations in Measurement Science Program.

References

1. O. Noroozian, J.A. Mates, D.A. Bennett, J.A. Brevik, J.W. Fowler, J. Gao, G.C. Hilton, R.D. Horansky, K.D. Irwin, Z. Kang et al., Appl. Phys. Lett. **103**, 202602 (2013)
2. J.A.B. Mates, *The Microwave SQUID Multiplexer*, Ph.D. thesis, School University of Colorado (2011)
3. J. Mates, D. Becker, D. Bennett, B. Dober, J. Gard, J. Hays-Wehle, J. Fowler, G. Hilton, C. Reintsema, D. Schmidt et al., Appl. Phys. Lett. **111**, 062601 (2017)
4. D.A. Bennett, J.A. Mates, J.D. Gard, A.S. Hoover, M.W. Rabin, C.D. Reintsema, D.R. Schmidt, L.R. Vale, J.N. Ullom, IEEE Trans. Appl. Supercond. **25**, 1 (2015)
5. S. Stanchfield, P. Ade, J. Aguirre, J. Brevik, H. Cho, R. Datta, M. Devlin, S. Dicker, B. Dober, D. Egan et al., J. Low Temp. Phys. **184**, 460 (2016)
6. J. Gao, J. Zmuidzinas, B.A. Mazin, H.G. LeDuc, P.K. Day, Appl. Phys. Lett. **90**, 102507 (2007)
7. J. Mates, K. Irwin, L. Vale, G. Hilton, J. Gao, K. Lehnert, J. Low Temp. Phys. **167**, 707 (2012)
8. T.J. Madden, T.W. Cecil, L.M. Gades, O. Quaranta, D. Yan, A. Miceli, D.T. Becker, D.A. Bennett, J.P. Hays-Wehle, G.C. Hilton et al., IEEE Trans. Appl. Supercond. **27**, 1 (2017)
9. R. Duan, S. McHugh, B. Serfass, B.A. Mazin, A. Merrill, S.R. Golwala, T.P. Downes, N.G. Czakon, P.K. Day, J. Gao et al., Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy V **7741**, 77411V (2010)
10. S. McHugh, B.A. Mazin, B. Serfass, S. Meeker, K. O'Brien, R. Duan, R. Raffanti, D. Werthimer, Rev. Sci. Instrum. **83**, 044702 (2012)
11. J. Hickish, Z. Abdurashidova, Z. Ali, K.D. Buch, S.C. Chaudhari, H. Chen, M. Dexter, R.S. Domagalski, J. Ford, G. Foster et al., J. Astron. Instrum. **5**, 1641001 (2016)
12. E. Nash, AN-1039, Application Note, Analog Devices (2009)
13. D.C. Price, arXiv preprint arXiv:1607.03579 (2016)
14. J.E. Volder, I.R.E. Trans, Electron. Comput. **3**, 330 (1959)

Affiliations

J. D. Gard[1] · D. T. Becker[1] · D. A. Bennett[2] · J. W. Fowler[1,2] · G. C. Hilton[2] · J. A. B. Mates[1] · C. D. Reintsema[2] · D. R. Schmidt[2] · D. S. Swetz[2] · J. N. Ullom[1,2]

✉ J. D. Gard
johnathon.gard@colorado.edu

[1] Department of Physics, University of Colorado, Boulder, CO 80309, USA

[2] Quantum Sensors Group, National Institute of Standards and Technology, Boulder, CO 80305, USA