Implementation of a Polyphase Filter Bank Channelizer on a Zynq FPGA

L. H. Arnaldi
Centro Atómico Bariloche and Instituto Balseiro
Comisión Nacional de Energía Atómica
S. C. de Bariloche, Argentina
arnaldi@cab.cnea.gov.ar

***Abstract*—This paper describes the design and implementation of a 16-channel polyphase filter bank (PFB) channelizer. The PFB channelizer structure implements a resource-efficient multichannel digital receiver for a set of frequency division multiplexed (FDM) signals that exist in a single sampled data stream. The implementation is based on the Zynq® field programmable gate array (FPGA) and aims to exploit the potential for data reuse and flexibility offered by the PFB channelizer structure. General design criteria are summarized for the 16-channel polyphase filter bank channelizer. Python and Vivado simulation results of the performance and operation for the design are also presented.**

***Index Terms*—Multirate signal processing, Polyphase Filter Bank, Discrete Fourier Transform, FIR Filtering.**

I. Introduction

Within the particle physics and astrophysics communities, superconducting detectors have been widely used for more than three decades in applications that require exceptionally high levels of performance. In this group of detectors, superconducting micro-resonators especially stand out for being simple and versatile devices. Interest in these devices has grown especially fast over the last decade as a result of the demonstration of transmission line micro-resonators as sensitive photon detectors [1] and the subsequent adoption of this type of resonator for reading quantum state of superconducting qubits [2]. At present, superconducting micro-resonators are under intensive study and are intended for a wide variety of applications, including dark matter search experiments, neutrino mass experiments, frequency multiplexed reading of cryogenic detector arrays, quantum circuits that include qubit reading and coupling to nano-mechanical systems.

In astronomy, transition edge sensors (TES) and superconducting tunnel junctions (STJ) are used for photon counting spectroscopy with X-ray time resolution [3], [4], and at submillimeter wavelengths, TES are used for high performance photometric observations [5], [6]. Long wavelength TES have revolutionized experimental cosmology, and these devices are now being designed in sophisticated image arrays and polarimeters [7]–[9]. The current challenge is to manufacture image matrices in extremely large format to achieve wide fields of view in the instruments and place image matrices in space. Associated with these challenges is the challenge of achieving the manufacture of electronic excitation and reading systems capable of working together with these large detector arrays. These electronic systems must have a large computational capacity and be able to process large amounts of data in relatively short times. This letter focuses on the design and characterization of an excitation and reading system for superconducting detectors working at high frequencies. In particular, experiments that use superconducting micro-resonators as photon detecting elements and characterization studies of the cosmic microwave background polarization (CMB) are of interest, because it is one of the fields that has most advanced the technology in recent years in the areas of particle physics and astronomy.

As stated above, this paper is focused in the development and implementation of a filter bank, to extract a set of frequency division multiplexing (FDM) channels that exist in a single sampled data stream. This data stream can be, for instance, a consequence of passing the FDM signal through a matrix of superconducting micro-resonators, working as photo-detectors. So, the filter bank will act as a monitoring and analysis system for the superconducting detector. The application of this structure to a particular hardware platform is also investigated, in order to test its performance in term of speed and resource utilization. The design steps and some simulation results are shown for a polyphase filter bank (PFB) to be used in a STEMLab RedPitaya 125-14 board [10], but it can be easily adapted to another platform. The entire structure for the PFB, supporting 16 channels, is implemented in a single FPGA, fitting into the Zynq® (xc7z010clg400-1) device of the STEMLab RedPitaya 125-14 board.

II. Preliminaries

The problem of changing the sampling frequency of a digitized signal appears in many applications of modern digital signal processing (DSP). The main advantage of multirate systems is based on the computational efficiency achieved, thanks to the ability to use different sampling rates simultaneously in the same system [11]. In some cases, for example telemetry, radio astronomy or frequency division multiplexing, it may be desirable to receive several digital channels simultaneously. One way to accomplish this task is to provide a separate filter for each channel. This process requires a great deal of computation if a large number of channels are involved. An alternative may be to use a polyphase filter that can accomplish the task much more efficiently [12], [13].

978-1-7281-6195-2/20/$31.00 ©2020 IEEE

IEEE Catalog Number 978-1-7281-6195-2

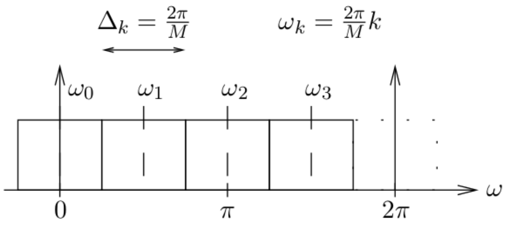


Fig. 1. Partitioning of the frequency spectrum in a uniform DFT filter bank. The frequencies, w_k, are obtained by uniformly shifting the frequency response of the prototype filter in integer multiples of $2\pi/M$, where M is the total number of channels.

A multirate polyphase filter can, for instance, perform the task of a multi-channel receiver (channelizer). Such task is equivalent to the down-conversion, filtering, and re-sampling of multiple narrowband signals [14]. The discrete Fourier transform polyphase filter bank (DFT-PFB) [12], [15] is one of these filter banks that provides high computational efficiency. This digital structure makes use of a polyphase filter to isolate and decimate the various input channels and then employs a fast Fourier transform (FFT) algorithm to efficiently convert each channel to baseband. The inverse procedure is also possible, interchanging the order of the processing blocks. Although this technique is limited to channel structures consisting of equally spaced channels, it requires only a single finite impulse response (FIR) filtering structure and a FFT block working together, increasing both the efficiency in size and speed. These aspects are desirable and well suited for reconfigurable logic, such as the field-programmable gate arrays (FPGA) devices. Combining the advantages of the quadrature-sampling scheme [16] and the polyphase filtering, it is possible to reach even increased resource optimization.

Taking into account the above advantages and thinking in the idea of expanding the design to a great number of channels ($\geq$1000), this paper will focus on the design of a DFT-PFB.

III. Digital Filter Banks

Filter banks are generally categorized as two types, *analysis* and *synthesis* filter banks. One is basically the inverse of the other. Both involve first the design and then the partition of a *prototype* filter, where the partitioning occurs over a bank of smaller filters. An analysis filter bank consists of a set of filters, with system functions $H_k(z)$, arranged in parallel. The frequency response characteristics of this filter bank splits the input signal into a corresponding number of sub-bands. On the other hand, a synthesis filter bank consists of a set of filters with system functions $G_k(z)$, also arranged in a parallel bank. The output of these filters are summed to form the synthesized signal $x(n)$. A deep treatment on the subject is beyond the scope of this article, but the interested reader can found detailed analysis of the filter bank structures in [12], [14], [15] and [17].

If a filter bank is used in the computation of the discrete Fourier transform of the sequence $x(n)$, that filter is called a DFT filter bank.

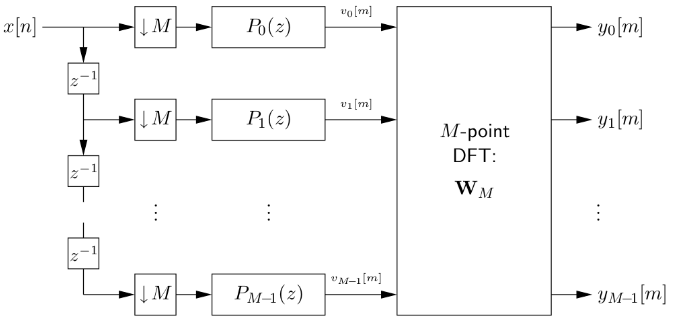


Fig. 2. Structure of an analysis DFT filter bank [12]. The $x[n]$ signal is the FDM signal containing the channels of interest. The combination of delays, plus down-sampling ($\downarrow M$) can be seen as a commutator, delivering one input sample to each of the filters ($P_i(z)$) at a time and at a low sampling rate (m). Every output channel $y_i[m]$ is recovered at the output of a M-point DFT block. Here, the n and m indexes denote a signal working at the input and output sampling rate, respectively. In this case $n > m$.

An analysis filter bank consisting of M filters $H_k(z), \quad k = 0, 1, ..., M-1$ is called a *uniform* DFT filter bank if $H_k(z)$ are derived from a prototype filter $H_0(z)$, where

$$H_k(w) = H_0\Big(w - \frac{2\pi k}{M}\Big), \quad k = 1, 2, ..., M-1. \quad (1)$$

Hence, the frequency response characteristics of the filters $H_k(z)$ are simply obtained by uniformly shifting the frequency response of the prototype filter in integer multiples of $2\pi/M$ [12]. The frequency spectrum is then partitioned in a uniform manner, as illustrated in Fig. 1. The sub-band width $\Delta_k = \frac{2\pi}{M} = \frac{F_s}{M}$ is identical for each sub-band and the band centers are uniformly spaced at intervals of $\frac{2\pi}{M} = \frac{F_s}{M}$, where F_s is the sampling frequency of the system and is the input sampling rate in this DFT-PFB. In general, the prototype filter can be either a FIR or IIR filter, but this selection depends on the application's requirements. One of these requirements, common in radio astronomy and communications, where the signal is supposed to pass without distortions, is the linear phase of the digital processing chain and is another key point for choosing this particular architecture in this paper.

IV. Polyphase structure of DFT filter banks

It is of particular interest the case where the down-sampling factor D is selected to be equal to the number M of frequency bands in which to split the input signal. When $D = M$, it is said that the filter bank is critically sampled. For the analysis filter bank, it can be defined a set of $M = D$ polyphase filters with impulse responses

$$p_k(n) = h_0(nM - k), \quad k = 0, 1, ..., M-1 \quad (2)$$

and the corresponding set of down-sampled input sequences for each channel k

$$x_k(n) = x(nM + k), \quad k = 0, 1, ..., M-1 \quad (3)$$

The structure of the analysis filter bank based on the use of

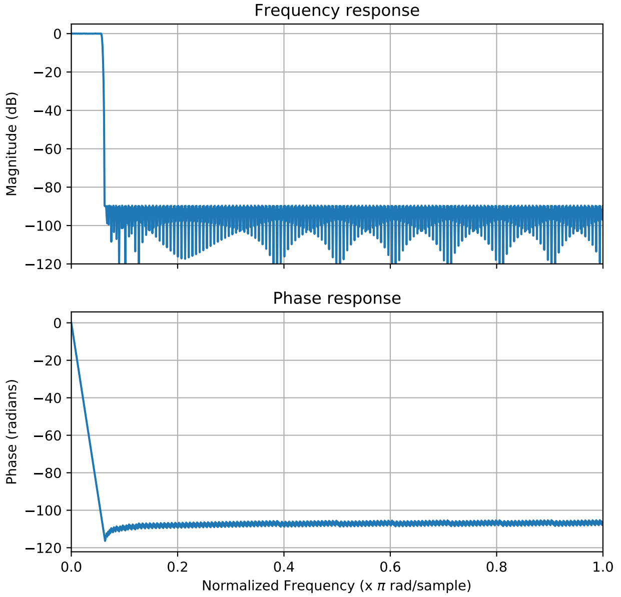


Fig. 3. Magnitude and phase response of the prototype filter. The normalized end of the pass-band frequency is $f_{pb} = 0.05625\,\text{rad/sample}$ and the normalized start of the stop-band frequency is $f_{sb} = 0.0625\,\text{rad/sample}$. The maximum pass-band ripple is set to $0.1\,\text{dB}$ and the minimum attenuation in the stop-band is $90\,\text{dB}$.

these polyphase filters can be obtained as

$$Y_k(m) = \sum_{r=0}^{M-1} \left[\sum_{l} p_r(l) x_r(m-l) \right] e^{-j2\pi rk/M}, \qquad k = 0, 1, ..., M-1. \tag{4}$$

Where m represents the new sampling rate and $n > m$ for this case. Note that the inner summation represents the convolution of $p_r(l)$ and $x_r(l)$. The outer summation represents the M-point DFT of the filter outputs. The filter bank structure corresponding to this computation is illustrated in Fig. 2. Every M samples at the input results in M outputs, denoted as $v_n[m], \quad n = 0, 1, ..., M-1$ from the M polyphase filters. The M-point DFT of this sequence yields the spectral samples $Y_k(m)$. The quantities $Y_k(m)$ and $x(n)$ are assumed to be complex. For large values of M, the FFT algorithm provides an efficient mean for computing the DFT and makes this approach attractive for the implementation in programmable logic.

V. The Prototype Filter

As stated above, the design of a DFT-PFB starts by specifying the prototype filter. Then, the focus is now on the understanding and development of the prototype filter for the polyphase filter bank. In an analysis (or channelizer) filter bank, the sample rate for the design process is the input rate, which turns out to be the highest sample rate the system will ever operate at. This frequency is $F_s = 125\,\text{MHz}$ for each one of the two input channels in the STEMLab RedPitaya 125-14 board. In the channelizer, each channel will be $1/M^{th}$ of the total received bandwidth. By partitioning the filter, what is done is to split the original filter into M pieces. So building the prototype means building a filter capable of filtering a single channel at the full bandwidth of all M channels.

TABLE I
Prototype Filter Specifications

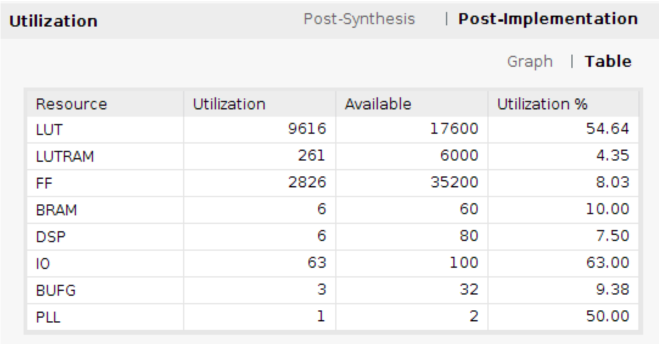
Parameter	Value
Type	FIR
Style	Low Pass
Sample Rate	125 MHz
Start of Stop Band	3.515625 MHz
End of Pass Band	3.90625 MHz
Band Pass Ripple	0.1 dB
Minimum Stop Band Atten.	80 dB

The design begins with the selection of the type of filter we want for the application, which in this case will be an equiripple low-pass FIR filter [18] for testing purposes. Then we set the sample rate (F_s), the end of the pass-band frequency (F_{pb}), and the start of the stop-band frequency (F_{sb}), along with the desired stop-band attenuation and the maximum pass-band ripple in dB. The pass-band and stop-band parameters are specified in units relative to the normalized sample rate. In this case, a pass-band frequency of $F_{pb} = 3.515625\,\text{MHz}$ and a stop-band frequency of $F_{sb} = 3.906250\,\text{MHz}$ have been selected, since our input FDM signal will have a full bandwidth of $125\,\text{MHz}$ for the in-phase and quadrature components (I/Q).

These design parameters have been expressed in normalized frequencies, where $F_s/2 = 62.5\,\text{MHz} \equiv 1\,\text{rad/sample}$, $f_{pb} = 0.9 \times (1/16)\,\text{rad/sample}$ and $f_{sb} = 1/16\,\text{rad/sample}$. The design parameters were changed to lowercase, indicating they are normalized parameters. Table I shows the design specifications for this filter and the Fig. 3 shows the magnitude and phase responses of the designed prototype filter.

Each channel will have a $7.8125\,\text{MHz}$ total bandwidth, given the full-spectrum bandwidth for the FDM signal is $125\,\text{MHz}$. The design of the low-pass prototype filter is based on these parameters. Each channel is $7.8125\,\text{MHz}$ wide and separated by $7.8125\,\text{MHz}$. Instead of filtering and moving each channel separately from his place in the high-frequency RF spectrum to baseband, the channelizer process all sixteen channels at the same time.

The resulting filter consists of a 1168-taps low-pass FIR filter, which seems quite long. However, since this filter is separated among a filter bank of 16 channels, each channel will actually consist of FIR filters with only 73 taps each. Computationally, these become very light-weight filters, especially if we can parallelize the computations, as can be done in an FPGA. A bigger concern may be memory usage, depending on the platform on which the filter bank is implemented.



Resource	Utilization	Available	Utilization %
LUT	9616	17600	54.64
LUTRAM	261	6000	4.35
FF	2826	35200	8.03
BRAM	6	60	10.00
DSP	6	80	7.50
IO	63	100	63.00
BUFG	3	32	9.38
PLL	1	2	50.00

Fig. 4. Resource utilization for the 16-channel channelizer, as reported by Vivado. This design consist of a 16-channel DFT filter bank (analysis filter bank), plus a tone generator for the FDM signal generation.

VI. Simulations and Testing

Simulations were carried out using Python and Vivado [19], in order to test the design. The DFT-PFB was implemented using the version 2018.3 of Vivado. It was used the block design methodology, and the resource usage is reported in Fig. 4. As can be seen, the design has a low DSP resource usage (less than 8% of total available), enabling the use of the remainder for other signal processing tasks. The design use standard Xilinx Intellectual Property (IP) cores for the FIR filter and for the FFT processing unit, allowing to reach the optimum approach in term of resource usage.

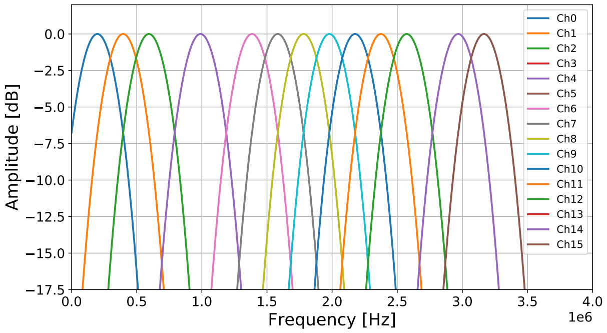


Fig. 5. Spectrum of the input tones. It is shown the distribution of the tones inside the $\sim 3.90625\,\text{MHz}$ pass-band bandwidth of the prototype filter. The tones for channels 4, 6 and 14 were removed for testing purposes.

An FDM test signal was created using Python. The test signal is composed of a group of complex tones at frequencies inside the bandwidth of the channels in the channelizer. Some of the probe tones were removed for testing purposes. The frequency spectrum of these signals can be seen in Fig. 5. The signal for channels 4, 6 and 14 were removed in order to test the channelizer response. The tones were generated at different frequencies, but all inside the bandwidth of the prototype filter. They are equi-spaced, with the first one located at $F_1 = 197.892,93\,\text{Hz}$ and the last one at $F_{16} = 3.166.286,88\,\text{Hz}$. All the tones are relatively low-frequency signals and they are all within the bandwidth of a channel in the channelizer, but spread in the input FDM signal. This signal was loaded into a tone generator block inside the FPGA, in order to have a testing setup formed by a loopback path between the digital-to-analog converter (DAC) and the analog-to-digital converter (ADC) system of the STEMLab Redpitaya board.

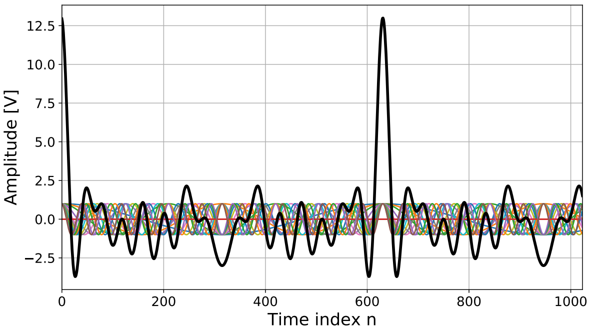


Fig. 6. FDM signal formed by the sum of the tones (tick line). It is shown the signal as a result of the sum of these probe tones in time. The crest factor of the FDM signal can be improved by assigning random phases and/or amplitudes to the probe tones that compose it.

Fig. 6 shows the FDM signal formed by the sum of the input tones. As can be seen in this figure, there are a peak in the FDM signal which may be a problem for the dynamic range of the system. This is system dependent and in general it is relatively easy to correct this issue playing around the phases of the tones. To maximize the signal-to-noise ratio (SNR) in each of the resonators working as a particle or photon detector, it is necessary to use the greatest possible carrier amplitude with the restriction that its total sum will not exceed the dynamic range of the DAC nor generate nonlinear distortions. That implies that the crest factor - that is, the relationship between the maximum amplitude and the mean quadratic amplitude (RMS) of the signal - must be as low as possible, improvement that can be achieved in the crest factor of an FDM signal by assigning random phases and/or amplitudes to each of the tones that compose it. For this testings, this is not a matter and the signal was used as it is.

The design was synthesized and implemented in the Zynq FPGA of the STEMLab Redpitaya board and the output of the testings can be seen in Fig. 7, Fig. 8 and Fig. 9. These last two figures show the recovering of the input tones in the time domain, also showing the proper functioning of the design.

VII. Conclusions

This paper presented the design and implementation of a 16-channel polyphase filter bank (DFT-PFB), a resource-efficient algorithm for multirate digital signal processing. It has been shown, through some simulations and after implementation into an FPGA, its usage and its capability of processing complex signals. It was created aiming to provide the criteria to take into account and the design guidelines when implementing this kind of hardware in a reconfigurable platform.

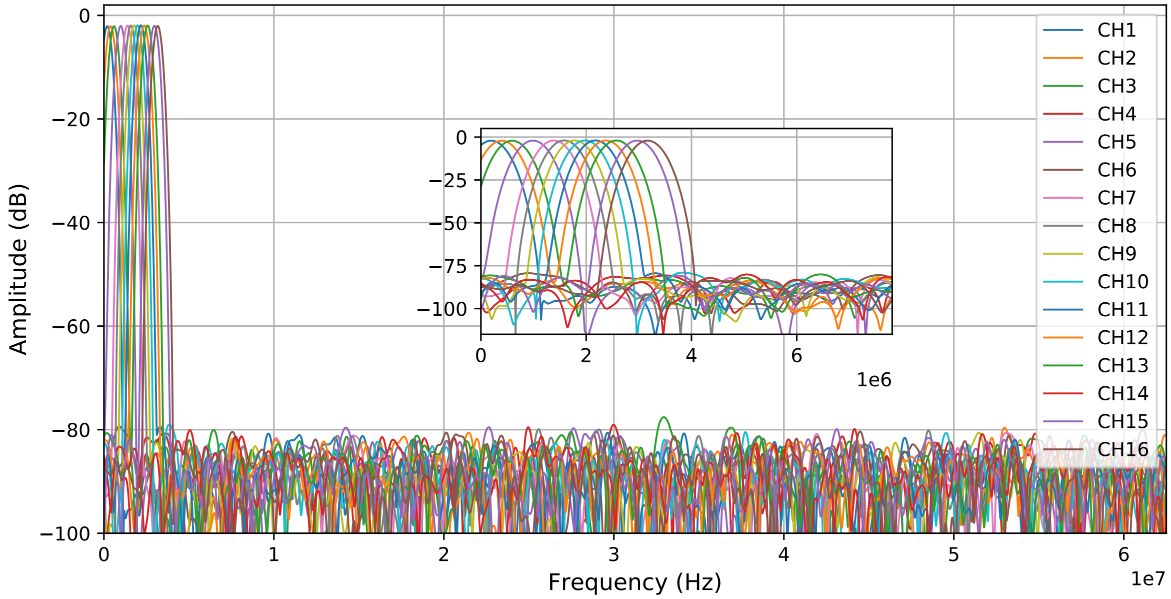


Fig. 7. Output spectrum after the channelizer processing. This spectrum is obtained after simulating the design in Vivado. A testbench was designed and the input were the tones generated in Python. The spectrum show the recovering of the tones after the channelizer processing. The SNR ratio of the recovered spectrum is in accordance with the prototype filter design.

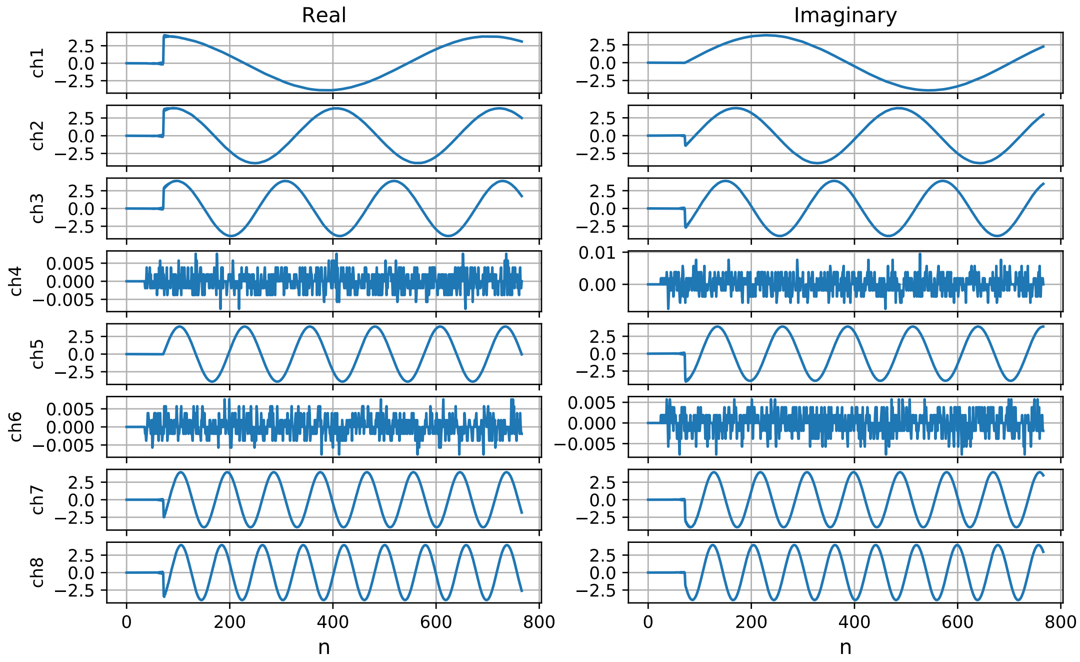


Fig. 8. Output of the channels 1 to 8. In order to test the performance of the channelizear, channels 4 and 6 are not receiving data. The vertical scale of the plots is in arbitrary units, as the intention here is to show just the recovery of the signals after passing the channelizer.

The implementation of the system showed to be highly configurable and flexible, thanks to the use of the FPGA present in the STEMLab RedPitaya board. The design was targeted to a specific platform and frequencies, but can be

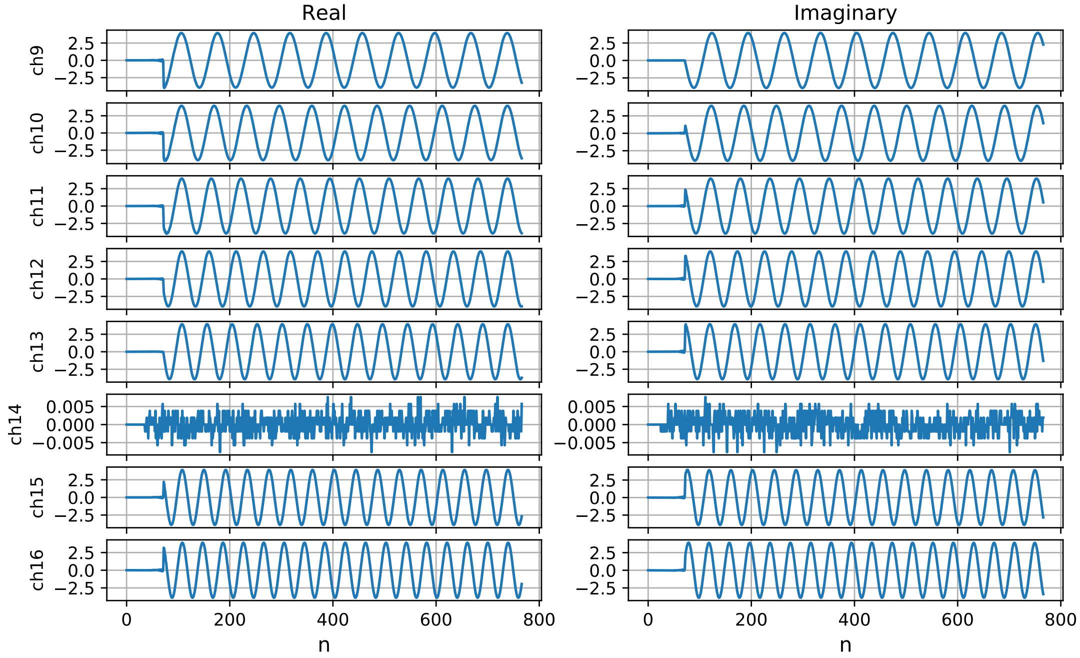


Fig. 9. Output of the channels 9 to 16. As stated in the text, for testing purposes, the channel 14 does not have an input signal.

easily adapted to another range of frequencies and hardware platform.

References

[1] P. K. Day, H. G. LeDuc, B. A. Mazin, A. Vayonakis, and J. Zmuidzinas, "A broadband superconducting detector suitable for use in large arrays." *Nature*, vol. 425, no. 6960, pp. 817–21, oct 2003.

[2] Andreas Wallraff, David I Schuster, Alexandre Blais, Luigi Frunzio, R-S Huang, Johannes Majer, Sameer Kumar, Steven M Girvin, and Robert J Schoelkopf, "Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics," *Nature*, vol. 431, no. 7005, pp. 0–5, 2004.

[3] P. Verhoeve, R. H. den Hartog, D. D. E. Martin, N. Rando, A. J. Peacock, and D. J. Goldie, "Development of distributed readout imaging detectors based on superconducting tunnel junctions for UV/optical astronomy," *Opt. IR Telesc. Instrum. Detect.*, vol. 4008, p. 683, 2000.

[4] T. R. Stevenson, F. A. Pellerano, C. M. Stahle, K. Aidala, and R. J. Schoelkopf, "Multiplexing of radio-frequency single-electron transistors," *Appl. Phys. Lett.*, vol. 80, no. 16, pp. 3012–3014, apr 2002.

[5] K. D. Irwin, "SQUID multiplexers for transition-edge sensors," *Phys. C Supercond. its Appl.*, vol. 368, no. 1-4, pp. 203–210, 2002.

[6] A. T. Lee, "SQUID readout multiplexers for transition-edge sensor arrays," *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 559, no. 2, pp. 786–789, 2006.

[7] A. Taylor, A. Challinor, D. Goldie, K. Grainge, M. Jones, A. Lasenby, S. Withington, G. Yassin, W. Gear, L. Piccirillo, P. Ade, P. Mauskopf, B. Maffei, and G. Pisano, "CLOVER - A new instrument for measuring the B-mode polarization of the CMB," *ArXiv Astrophys. e-prints*, 2004.

[8] W. Holland, M. MacIntosh, A. Fairley, D. Kelly, D. Montgomery, D. Gostick, E. Atad-Ettedgui, M. Ellis, I. Robson, M. Hollister, A. Woodcraft, P. Ade, I. Walker, K. Irwin, G. Hilton, W. Duncan, C. Reintsema, A. Walton, W. Parkes, C. Dunare, M. Fich, J. Kycia, M. Halpern, D. Scott, A. Gibb, J. Molnar, E. Chapin, D. Bintley, S. Craig, T. Chylek, T. Jenness, F. Economou, and G. Davis, "SCUBA-2: a 10,000-pixel submillimeter camera for the James Clerk Maxwell Telescope," *Millim. Submillim. Detect. Instrum. Astron. III*, vol. 6275, p. 62751E, 2006.

[9] M. Piat, E. Battistelli, A. Baù, D. Bennett, L. Bergé, J.-P. Bernard, P. de Bernardis, M.-A. Bigot-Sazy, G. Bordier, A. Bounab, E. Bréelle, E. F. Bunn, M. Calvo, R. Charlassier, S. Collin, A. Cruciani, G. Curran, L. Dumoulin, A. Gault, M. Gervasi, A. Ghribi, M. Giard, C. Giordano, Y. Giraud-Héraud, M. Gradziel, L. Guglielmi, J.-C. Hamilton, V. Haynes, J. Kaplan, A. Korotkov, J. Landé, B. Maffei, M. Maiello, S. Malu, S. Marnieros, J. Martino, S. Masi, L. Montier, A. Murphy, F. Nati, C. O'Sullivan, F. Pajot, C. Parisel, A. Passerini, S. Peterzen, F. Piacentini, L. Piccirillo, G. Pisano, G. Polenta, D. Prêle, D. Romano, C. Rosset, M. Salatino, A. Schillaci, G. Sironi, R. Sordini, S. Spinelli, A. Tartari, P. Timbie, G. Tucker, L. Vibert, F. Voisin, R. A. Watson, and M. Zannoni, "QUBIC: the Q&U Bolometric Interferometer for Cosmology," *J. Low Temp. Phys.*, vol. 167, no. 5-6, pp. 872–878, jun 2012.

[10] STEMLab RedPitaya, "RedPitaya Open Source Instrument." [Online]. Available: http://www.redpitaya.com/

[11] G. J. Dolecek, *Advances in Multirate Systems*. Springer, sep 2017, vol. 1.

[12] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*. Pearson, abr 2006, vol. 1.

[13] M. Frerking, *Digital Signal Processing in Communications Systems*. Springer Science & Business Media, mar 2013, vol. 1.

[14] F. J. Harris, *Multirate Signal Processing for Communication Systems*. Prentice Hall, may 2004, vol. 1.

[15] N. Fliege, *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets*. Wiley, ago 1994, vol. 1.

[16] R. G. Lyons, *Understanding Digital Signal Processing*. Prentice Hall/PTR, ago 2004, vol. 1.

[17] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice Hall, oct 1993, vol. 1.

[18] T. Parks and J. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Trans. Circuit Theory*, vol. 19, no. 2, pp. 189–194, 1972. [Online]. Available: http://ieeexplore.ieee.org/document/1083419/

[19] Xilinx Inc., "Vivado Design Suite." [Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html