

A Comparative Study for Performance and Power Consumption of FPGA Digital Interpolation Filters

Tim Donnelly, Jungu Choi, Alexander V. Kildishev, Matthew Swabey, Mark C. Johnson
Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907

Abstract—The development of FPGA-based digital signal processing devices has been gaining attention. Researchers seek to reduce power consumption and enhance signal processing quality in these devices with given resources and spatial limits. Hence, there is a need to investigate both the capability and the power consumption associated with the various digital filtering schemes commonly used in FPGA-based devices. We carry out a set of performance and power consumption measurements of interpolation filters using an FPGA and other basic signal processing building blocks. We compare the signal processing performance with theoretical prediction, and measure the power consumed by the filters. Our experimental measurements also confirm the accuracy of the numerical tools used for predicting FPGA power consumption. This paper is aimed at providing a framework to accurately test basic signal processing across various interpolation schemes and compare the respective schemes' software-side contributions to power consumption and filtering quality.

Keywords—Digital signal processing; digital interpolation filters; FPGA

I. INTRODUCTION

FPGA filters are widely used in broad applications, including radio-frequency sensors [1]–[3], imaging [4]–[7] and wearable medical devices [8]–[14], due to its flexible programmability. Although versatile, the FPGA-based filters are known to consume considerable power as a draw-back. Previously, researchers have looked into the performance and power estimates of FPGA-based digital filters [15], [16] and these are still relevant issues as consequence of the growing demand for quality signal processing with low power consumption. It is therefore of great interest to understand how the capability of such devices are influenced by the nature of the energy constrained environment in which they operate. In the industry, the effort to optimize power consumption in FPGA-based devices heavily extends to a hardware side such as clock gating [17] and word-length [18]. And power estimation methodology was developed [19] to identify high-power-consuming groups and improve on FPGA power dynamic models.

Taking wearable medical devices as an example, consider how in [20]–[22] remote devices for monitoring a patient's brain-wave were developed and assessed for their efficiency and power consumption. The intra-body sensor in [2] showed clear trade-offs between power consumption and signal processing quality. Likewise, the increasing digital signal processing (DSP) capability of portable heart monitoring devices [23], [24] must be accounted for as part of the available energy budget. Overall, a big challenge biomedical researchers face is to make devices such as these comfortable for patients so as not to compromise their daily activities while still implementing the advanced signal processing algorithms. That is, the design

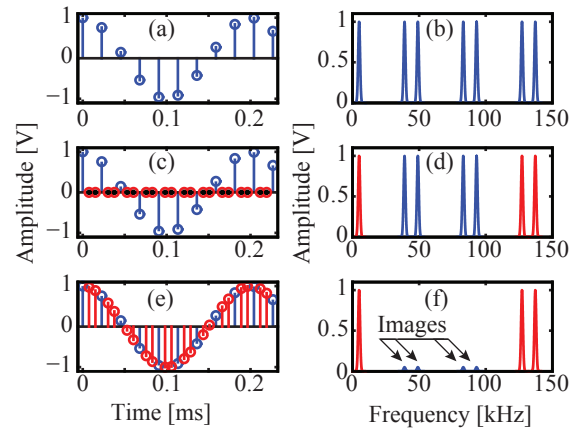


Fig. 1. Depiction of digital interpolation. (a) Original signal with (b) its spectral representation; (c) zeros are added between sampled data and (d) the effective sampling rate increases; (e) interpolation is performed and new data are created (red) and (f) low frequency sampling images are eliminated (significantly attenuated).

of a comfortable device for patients is at odds with the large (heavy) battery needed to meet the power requirements of such a device.

In addition, we note that numerous devices utilize hybrid-filter configurations to optimize performance and reduce power consumption. Most hearing aids contain filter-banks for certain frequency components of sound to serve an individual's hearing needs [8]–[12]. A novel electrocardiogram (ECG) design [14] by Hong et al. contains high-pass FIR filter followed by linear interpolation for quality signal processing with low power. An FPGA-based accelerometer [25] by Ramaesh et al. has FIR filters with cascaded-integrator comb (CIC) filters for highly efficient filtering. In addition, an ultrasound digital transcranial Doppler system (digiTDS) [7] monitors intracranial vessels and update the structure profile in the continuous time domain. Various FPGA-based filters in this system, including the CIC and FIR filters, serve different purposes at the expense of varying degrees of computation resources. These issues motivate us to examine several different digital interpolating filters to compare ideal (theoretical) performance and actual measurements. More specifically, different approaches for the interpolation (upsampling) of signals covering the human audible range have been implemented. Although applicable to wearable medical devices in its own right [26]–[29], the multirate filters used here also serve as a prototypical example for the general study of comparing signal processing performance and digital filter power consumption.

This work considers the power consumption and filter



Fig. 2. Block diagram for our measurement setup. See text for more details.

performance for a range of filter designs, including First Order Hold (FOH), Finite Impulse Response (FIR), and Cascaded Integrator-Comb (CIC). Mathematical analysis of the various interpolation schemes is provided and our empirical results are in good agreement with the analytical solutions. The power and FPGA resource utilization for each interpolation scheme is detailed. The measured power consumptions were also in great agreement with the values predicted by the simulation tools. Our designs outlined in the block diagram in Fig. 2 along with the experimental setup described in Section III-A provides a framework for graduate students and researchers to accurately test signal processing theories. For this purpose, we have also uploaded our HDL files with detailed documentation to an online¹ repository.

II. THEORY

In the following sub-sections, we cover the theory behind several different types of interpolation filters.

A. Interpolation Overview

The well-known sampling theorem states that a continuous band-limited signal can be perfectly reconstructed from its samples, given a sufficiently high sampling rate. This is accomplished by passing the samples through an ideal low-pass filter; the frequency domain transfer function of which has an infinitely sharp cut-off characteristic as in (1).

$$H(f) = \text{rect}\left(\frac{f}{f_s}\right) \quad (1)$$

Here, f_s is the sampling frequency and $\text{rect}(x)$ is a unit function that becomes 1 when the absolute value of x is less than $1/2$ and 0 when the absolute value of x is greater than $1/2$. The corresponding time domain impulse response is a sinc function as in (2). Thus, convolving a sampled signal with (2) reconstructs the original continuous signal.

$$h(t) = \text{sinc}\left(\frac{t}{T}\right) \quad (2)$$

where T is the sampling period.

In order to change the sampling rate of a signal digitally, one must evaluate the result of the convolution of a sampled signal with (2) at the new sampling interval [30]. Of course, (2) cannot be evaluated perfectly, on account of its infinite duration. Thus, the problem of sample rate conversion - or upsampling/interpolation as is the primary concern here - becomes a problem of approximating (1) through filter design. Fig. 1 illustrates how interpolation can be used to reduce the performance requirements of a post-DAC analog reconstruction filter - spacing out the sampling images allows for a looser approximation of (1) to be used as the reconstruction filter.

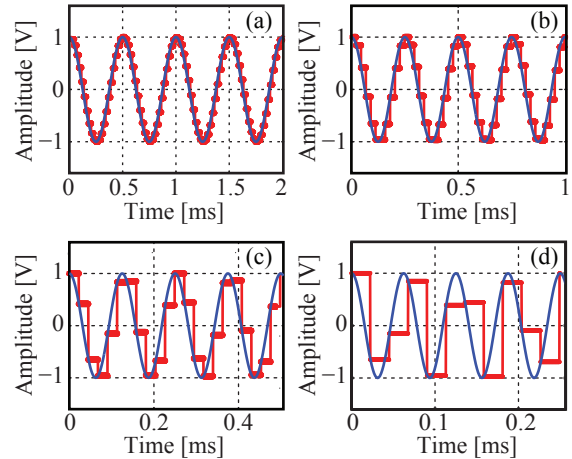


Fig. 3. Various sinusoidal waves at (a) 2 KHz, (b) 4 KHz, (c) 8 KHz, and (d) 16 KHz. The blue curves are the original signals and the red curves represent signals reconstructed via zero-order hold (ZOH) from the sampling at 44.1 KHz of the original signals.

B. Zero-order Hold (ZOH)

It is common for digital-to-analog converters (DAC) to inherently perform a zero-order hold (ZOH) operation on sampled inputs. That is, DAC's typically hold an analog equivalent output of the previous digital input until a new value is presented. Thus, no digital signal processing is required for this simple interpolation scheme. The transfer function of the ZOH operation has the form of a sinc function. This can be shown by considering a ZOH impulse response, as in (3).

$$h_0(t) = u(t) - u(t - T). \quad (3)$$

Here, u is the step function and T represents the sampling period of the discrete signal. The Fourier transform is thus:

$$H_0(f) = e^{-j\pi f T} \text{sinc}(fT) \quad (4)$$

Comparing the frequency domain transfer functions, one finds the sinc function of (4) only loosely approximates the rect function, (1), needed for perfect reconstruction.

A more detailed look at the ZOH operation can be found by considering the case of a sinusoidal input. As shown in Fig. 3, the ZOH scheme is characterized by a stair-case like profile, which becomes more evident as the frequency of the signal increases. Using MATLAB, we evaluated (4) for a sine wave with $f \simeq 5$ KHz. In Fig. 4(a), we show the sampled signal reconstructed via ZOH scheme. In Fig. 4(b), we show the fast Fourier transform (FFT) of the reconstructed signal, and verified that the frequency spectrum of the reconstructed signal matches well with the coefficients obtained from evaluation of (4).

¹https://github.com/donnely92/FPGA_Interpolation

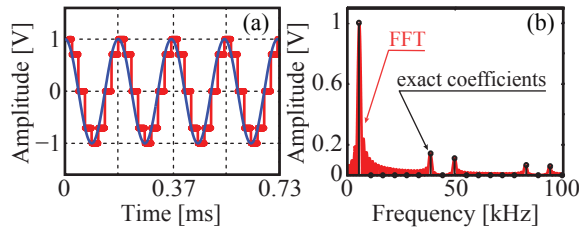


Fig. 4. The original signal (blue) is a 5.5 KHz cosine wave. (a) The reconstructed signal (red) and (b) the FFT spectral representation.

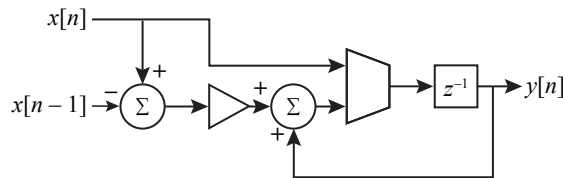


Fig. 5. The block diagram for the linear interpolation scheme. Its structure is relatively simple due to the lack of multiplication operations.

C. Linear Interpolation

Linear interpolation, or first order hold (FOH), is a familiar operation, even outside the realm of digital signal processing. The impulse response of a FOH filter is a triangle (or tent) function [30].

$$h_1(t) = \frac{1}{T} \text{tri} \left(\frac{t}{T} \right) \quad (5)$$

The function $\text{tri}(t)$ is defined as when t is zero, the function becomes 1. As t increases (decreases), the function slopes towards the t axis with unity slope until they intersect at ± 1 . The corresponding transfer function is thus of the form sinc squared as shown in (6).

$$H_1(f) = \text{sinc}^2(fT) \quad (6)$$

While $h_1(t)$ as a continuous impulse response would yield perfect linear interpolation, a digital interpolating filter must operate with discrete samples. This is accomplished by first sampling the impulse response $h_1(t)$ at a rate corresponding to the up-sample factor, then using this now discrete impulse response to convolve a ‘zero-stuffed’ (formally, upsampled) input signal. This is the same method behind any general finite impulse response (FIR) upsampling filter, which we shall discuss later in more detail. Unlike FIR interpolation, which requires multiplication and coefficient storage (see Section II-D), the linear interpolation scheme requires far less computational resources and its architecture is depicted in the block diagram in Fig. 5. Recall that the DAC inherently applies a zero-order-hold between samples, so the discrete linearly interpolated signal is converted into a continuous signal through this operation.

Fig. 6 shows a few examples of signals upsampled 16 times via linear interpolation. With the original signal at relatively low frequencies, 2 KHz and 4 KHz in Fig. 6(a) and (b), respectively, the interpolated signal matches the original reasonably well. At higher frequencies, 8 KHz and 16 KHz in Fig. 6(c) and (d), respectively, there is a noticeable mismatch between the interpolated and original signals. In Fig. 7(a),

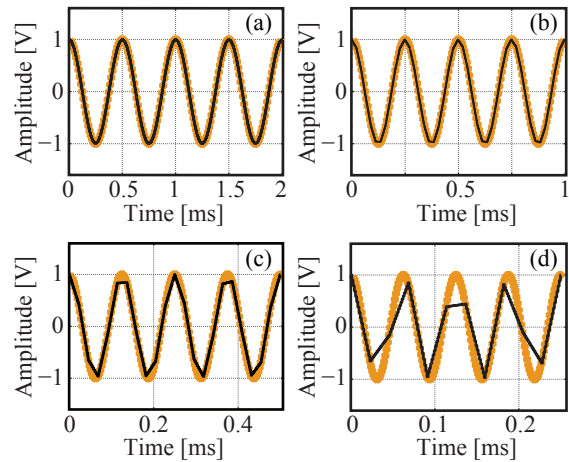


Fig. 6. Various sinusoidal waves at (a) 2 KHz, (b) 4 KHz, (c) 8 KHz, and (d) 16 KHz. The (orange) markers represent the original signals and the black curves represent signals reconstructed via linear interpolation at a 16 time upsampling frequency (i.e. 16×44.1 KHz).

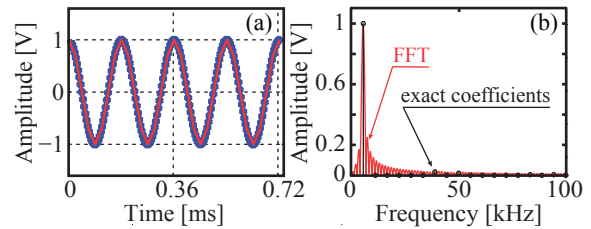


Fig. 7. The original signal (blue) is a 5.5 KHz cosine wave. (a) The interpolated signal (red) obtained via linear interpolation with a upsampling factor of 16 (b) The FFT results of the sampled signal and the Fourier coefficients for the Fourier expansion of the sampled signal.

we plotted linearly interpolated signal using a sine wave of frequency ($f \simeq 5$ KHz). As with ZOH, the FFT routine was performed and is shown in Fig. 7(b). We validate the spectrum with the exact Fourier coefficients obtained by evaluating (6).

D. Finite Impulse Response (FIR) Interpolation

The structure of the finite impulse response scheme is shown in Fig. 8. Here, $x[n]$ represents data samples, which are strung together via unit delays so as to create a sequence of previous values. These samples are then multiplied by coefficients and summed. The coefficients b_0, b_1, \dots, b_n for a low-pass filter are samples of a sinc function. As mentioned, since (2) is infinite, for it to be used as a basis for the FIR filter coefficients it must first be truncated. One method to truncate it is to apply a window such that it has nonzero only in a certain range, as in (7) and (8) for which $n \leq N$ and N is the FIR filter length and corresponds to the number of the b_n coefficients.

$$h[n] = h_{\text{sinc}}(nT)w[n] = \frac{\sin(\omega_c(n - \frac{N-1}{2}))}{\pi(n - \frac{N-1}{2})}, \quad (7)$$

$$H(\omega) = H_{\text{sinc}}(\omega) * W(\omega) \quad (8)$$

Here, $w[n]$ represents a rectangular window that truncates the *sinc* function to a length of N and $W(\omega)$ is the Fourier Transform of said window: $|W(\omega)| = \frac{|\sin(\omega N/2)|}{|\sin(\omega/2)|}$. Other

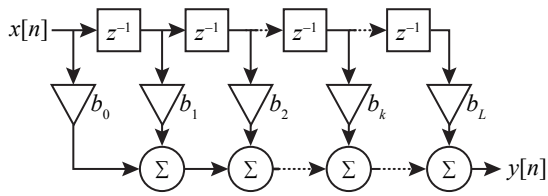


Fig. 8. The block diagram for the finite impulse response (FIR) scheme.

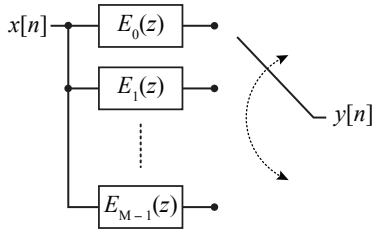


Fig. 9. The block diagram for the polyphase multi-rate signal processing.

truncating windows, such as Hamming, Hanning, or Blackman could alternatively be used for a trade-off between transition width and stop-band ripple (peak sidelobe). The rectangular window has the sharpest transition band - but considerable stopband ripple; whereas the other windows mentioned have wider main lobes in exchange for reduced peak sidelobes.

An alternative to applying a window as a means of truncation is to derive optimal coefficients constrained to a given number of terms [31]. That is, for a given filter length, coefficients can be determined such that the approximation error between the desired frequency response and the actual frequency response is minimized in a least squared error (LSE) fashion. This particular method will be referred to as an LSE FIR scheme.

An interpolation filter can be optimized by using a polyphase structure where sub-filters are multiplexed together. Instead of actually performing upsampling followed by filter-

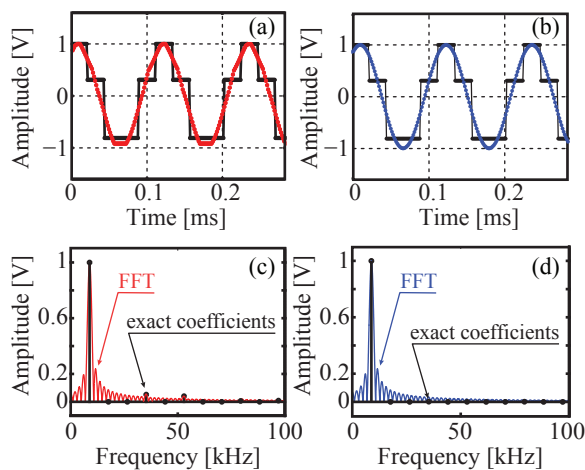


Fig. 10. 9 KHz sine wave sampled at 44.1 KHz with (a) ZOH reconstruction (black), FIR upsampling with the filter length $L = 21$ and (c) its spectral analysis and (b) ZOH reconstruction (black), FIR upsampling with the filter length $L = 201$ with (d) its spectral analysis. Both cases have the upsampling factor $v = 16$.

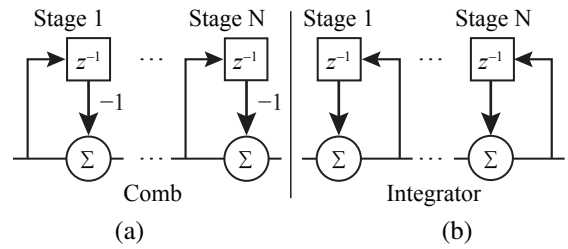


Fig. 11. Block Diagram for (a) N integrator stages and (b) N comb stages.

ing, since many of the samples are known to be zero (and thus multiplication by a coefficient is also zero and would not contribute to the summation tree), the FIR filter can be divided into sub-filters as in Fig. 9. Fig. 10(a) and (c) show, respectively, the time domain interpolation and frequency domain match between the FFT and analytical coefficients (obtained by evaluating (8) with a sine input) for a FIR filter with $N = 21$. Fig. 10(b) and (d) shows the same for a FIR filter with $N = 201$.

E. Cascaded Integrator-Comb (CIC)

An efficient interpolation low-pass filter due to lack of multiplication and coefficient storage can be constructed with a Cascaded Integrator Comb (CIC) Filter [32]. In this section, we apply the Z -transform, where discrete samples are converted to the z -domain. A comb has a transfer function of the form

$$H_{\text{comb}}(z) = 1 - z^{-RM}. \quad (9)$$

The CIC filter integrator stage has a transfer function that can be defined as:

$$H_{\text{integ}}(z) = \frac{1}{1 - z^{-1}} \quad (10)$$

where M is the differential delay per comb stage ($M = 1$ is used throughout) and R is the rate change factor. For an interpolating filter, the comb stage at f_s are followed by the integrator at a higher frequency $R \times f_s$. N sequential integrator and comb stages can be represented as

$$H_{\text{CIC}}(z) = H_{\text{comb}}(z)H_{\text{integ}}(z) = \frac{(1 - z^{-RM})^N}{(1 - z^{-1})^N}. \quad (11)$$

III. RESULTS

In this section, we first compare the transfer function measurements with the analytical expressions. We then discuss the power and FPGA resource utilization for each interpolation scheme.

A. Filter Measurements

Our measurement setup is represented in the block diagram in Fig. 2. To measure the transfer functions, we generated uniformly distributed white noise from the desktop computer and obtained the frequency spectrum of the interpolated white noise using the FFT capability of the oscilloscope. A printed circuit board (PCB) was produced that incorporated the Texas Instrument PCM 2706 and PCM 1702 ICs. The USB interface chip (PCM 2706) connects to a desktop computer, which is responsible for streaming the audio data, and produces a digital

data output in the form of I^2S . An FPGA development board (featuring the Xilinx 7 series Artix FPGA) receives this I^2S output and performs the signal processing (interpolation). The signals are then sent to the digital-to-analog converter (PCM 1702) output of which (after being passed through a trans-impedance amplifier) is measured with a Keysight 3000 X-series oscilloscope.

The transfer functions shown in Fig. 12 illustrate how digital interpolation can be used to reduce the performance requirements of analog reconstruction filters. If no upsampling is performed, as in Fig. 12(a), the analog reconstruction filter must compensate for attenuation in the passband as well as have a steep transition band to remove sampling images. On the other hand, when digital interpolation is used the analog reconstruction filter is only responsible for removing sampling images that appear far away from the passband, centered around the new sampling frequency.

For the FOH scheme in Fig. 12(b), while it has a steeper transition band compared to the ZOH scheme, it too suffers from attenuation in the passband and in general does a poor job of removing the sampling image near the cutoff frequency. However, the computational cost can be quite small.

Fig. 12(c) shows the transfer function of an FIR scheme that used a rectangular window to truncate the ideal low pass filter impulse response. As expected, the transition band appears rather steep but ripples in both the passband and stop band are prominent. In Fig. 12(d) a more preferred transfer function was observed using the least squared error FIR scheme discussed in Section. II-D. In this case, the same computational resources were required as in the rectangularly windowed FIR scheme (i.e., the two schemes have the same filter length). The measured transfer function is rather noisy in the stopband region. After analysis of the setup, we concluded that the noise floor of the DAC on the PCB contributes significantly to the deviation between the measurement and the theoretical yield. This is evident when the filter length is reduced to $L = 64$ in Fig. 12(e). The noise floor no longer appears to affect the interpolation performance, thus producing a transfer function in good agreement with the analytical solution.

Lastly, Fig. 12(f) shows the measured transfer function of the CIC interpolation scheme. The three stage, 16x upsampling CIC filter ($R = 16$, $N = 3$) used here has the benefit of relatively small computational cost (see III-C). However, it suffers from attenuation in the passband, much like as was the case in the ZOH and FOH interpolation schemes, as well as a slow transition band such that low frequency sampling image content is not fully eliminated.

B. FPGA Implementation

For the FOH scheme a single 16 bit full adder was synthesized, along with three 16 bit registers to store intermediate values. No explicit multiplication resources were required, since the scaling block shown in Fig. 5 was implemented by bit-shifting (which was possible in this case because the upsampling factor was a power of two).

In the polyphase FIR implementations, Fig. 8 and 9, multiplication blocks needed to be synthesized. In order to

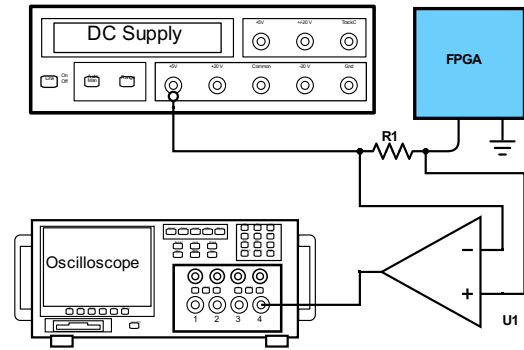


Fig. 13. The power measurement setup consists of a DC power supply, an oscilloscope and the resistor $R1 = 20 \text{ m}\Omega$.

meet timing constraints, pipeline registers were also used. It should be noted that the number of multiplications needed to implement the polyphase scheme is L/R , where L is the filter order (length) and R is the interpolation factor and thus 4 multipliers are needed in the case for $L = 64$, $R = 16$ while 20 multiplications are needed for the case $L = 200$, $R = 10$. Longer filter lengths require a more-than-proportional length adder tree. The adder tree and coefficient storage detail are not shown explicitly in Fig. 8, but a similar implementation was used to what is detailed in [33].

Lastly, the CIC scheme uses six 28 bit adders, three for addition and three for subtraction, and 12 similar width registers. Six of these registers are for the storing previous values for the integrator and comb stages as in Fig. 11, and the remaining six are for pipelining each arithmetic operation (not shown in figure). The adders and registers are wider than the input data (16 bits), to account for bit growth as described in [32].

C. FPGA Resource Usage

In this section, we discuss FPGA resource requirements for each interpolation scheme. The FPGA resources consumed by the interpolation filter designs are of two types: lookup tables (LUT) and flip-flops. Taken together, the power consumed by the FPGA was both predicted and measured. Although total power consumption, including other FPGA resources and external analog circuitry, is an important aspect of signal processing, we omit this measure from the results since all interpolation schemes are equivalent in this regard. Fig. 13 shows the schematic diagram for power measurements. We first isolated all peripherals that may consume power and therefore our measurement is due only to the digital filters in the FPGA. The voltage across a current sensing resistor ($\sim 20 \text{ m}\Omega$) was monitored using the oscilloscope. Since one interpolation filter consumes very little power and its measurement would therefore be sensitive to noise, we implemented a number of filters in parallel inside the FPGA and derived an average power for a single filter from this measurement. We set the clocking frequency and other parameters of the FPGA constant for all filters so that differences in power consumption would be purely due to the complexity of the filters. We used the Xilinx software package Vivado to estimate circuit current and

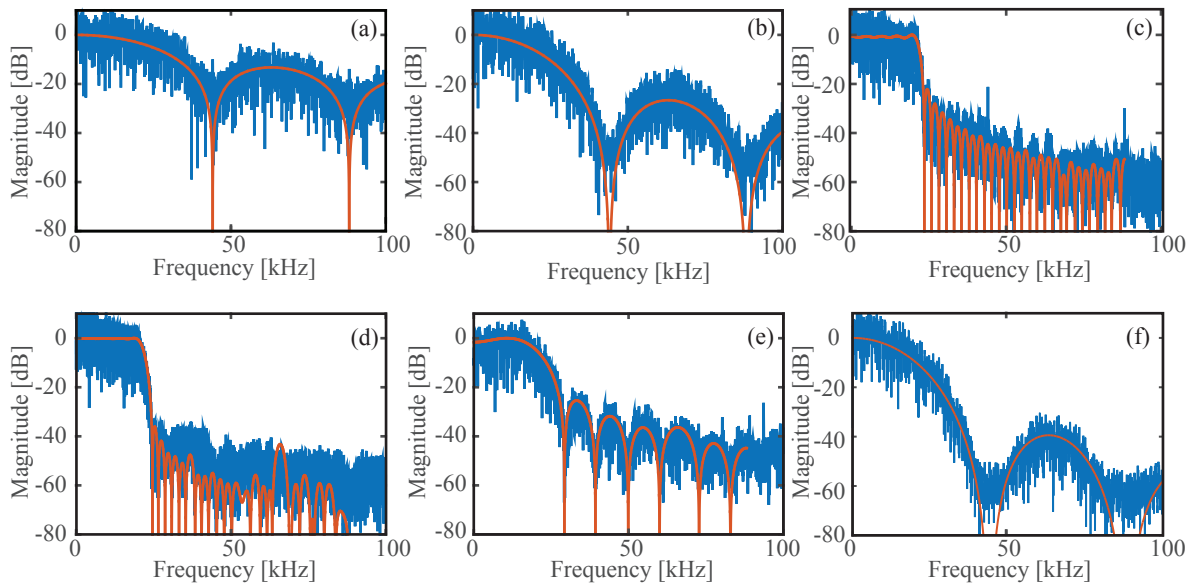


Fig. 12. Measurements (blue) and analytical (red) expressions for the transfer functions. (a) zero-order hold (ZOH), (b) first-order hold (FOH), (c) rectangularly windowed FIR ($L = 200$), (d) least square error (LSE) FIR ($L = 200$), (e) least square error (LSE) FIR ($L = 64$), and (f) CIC ($R = 16$, $N = 3$).

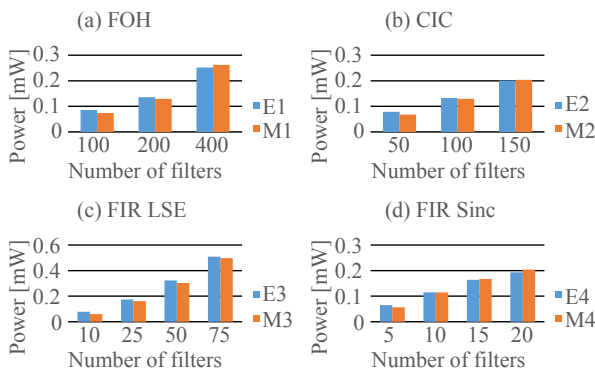


Fig. 14. The estimated (orange) and measured (blue) current for each interpolation scheme. The horizontal axis is the number of filters that are run in parallel in the FPGA.

our measured data shows good agreement with this estimation. Fig. 14 shows the predicted values of FPGA current draw compared to the measurements.

Table I summarizes the FPGA resource requirements for each interpolation scheme. Due to the varying complexity of each scheme, some filters draw considerably more power than others. In the case of the ZOH scheme, all logic LUTs and flip-flops are used to perform data flow control from the input source (I^2S from PCM 2706) to the DAC (PCM 1702). Since resources utilized in this way do not contribute to the signal processing operation, they are not included in the power measurements. All other interpolation schemes require additional resources on top of those required by the ZOH scheme; the FPGA resources and power consumption results listed in Table I only account for these additional resources.

TABLE I. FPGA LOGIC AND POWER UTILIZATION

	LUT	Flip-flops	P_E [mW]	P_M [mW]
ZOH	101	73	N/A	N/A
FOH	71	88	0.66	0.63
CIC	267	315	1.36	1.34
FIR ($L=64$)	761	293	6.64	6.79
FIR ($L=200$)	2670	1336	10.18	9.65

P_E represents the estimated power consumption and P_M represents the actual measurements.

IV. CONCLUSION AND FUTURE WORK

We have studied various digital signal interpolation schemes using an FPGA. Our measurements were in good agreement with analytical expressions, although noise effects contributed to a slight mismatch for some cases. The power measurement confirms the power consumption rate predicted by the simulation tools. We emphasize that studies of power consumption and filter performance across a wide range of FPGA-based devices and applications, are crucial. This paper presents implementation details of various filtering schemes to test digital filter theory in a compact system for graduate students and researchers in the digital signal processing fields. In addition, our work covers some basic hardware-side contributions to power consumption in the digital filtering schemes. In the future, we plan to incorporate more hardware-oriented FPGA performance into the filter designs. This will allow us to explore optimization of filter quality while minimizing power consumption and other resources. Specifically, we will study applications with portable digital signal processing units such as wearable medical devices. Such devices may often hamper patients' daily activity due to the physical dimensions and available battery life. More in-depth work on the hardware contributions to the FPGA-based filter performance will complement our current software-side evaluation in designing the optimization scheme for wearable medical devices.

ACKNOWLEDGMENT

The authors would like to thank C. A. Bouman, N. Conrad, M. Y. Shalaginov, and D. S. Elliott for helpful advice and discussions. Also, technical help from Kyle Woodworth is acknowledged.

REFERENCES

- [1] Z. Szadkowski and D. Gas. Adaptive linear predictor fir filter based on the cyclone v fpga with hps to reduce narrow band rfi in radio detection of cosmic rays. *IEEE Transactions on Nuclear Science*, 63(3):1455–1462, June 2016.
- [2] G. E. Santagati and T. Melodia. Sonar inside your body: Prototyping ultrasonic intra-body sensor networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2679–2687, April 2014.
- [3] R. P. S. Rathore. Reconfigurable digital radar receiver implemented in fpga using under-sampling, direct iq generation, multi-rate filter and pulse compression. In *2014 IEEE International Microwave and RF Conference (IMARC)*, pages 174–177, Dec 2014.
- [4] A. Gabiger-Rose, M. Kube, R. Weigel, and R. Rose. An fpga-based fully synchronized design of a bilateral filter for real-time image denoising. *IEEE Transactions on Industrial Electronics*, 61(8):4093–4104, Aug 2014.
- [5] F. E. Sagcan, A. Bayram, and B. Sen. Fpga implementation of delta sigma modulation based all-digital rf transmitter for parallel magnetic resonance imaging. In *2015 European Microwave Conference (EuMC)*, pages 494–497, Sept 2015.
- [6] K. Hu, X. Wang, Y. Yao, X. Gao, and G. Jin. Real-time data acquisition for single photon imaging detector. *IEEE Transactions on Nuclear Science*, 63(2):1076–1082, April 2016.
- [7] M. Lewandowska, M. Walczaka, P. Karwata, B. Witeka, A. Nowickia, and P. Kar lowiczb. Research and medical doppler platform. In *Acoustics 2012 Nantes Conference*, pages 853–858, 2012.
- [8] N. M. Ghamry. An FPGA implementation of hearing aids based on wavelet-packets. *Journal of Computers*, 7(3), 2012.
- [9] A. R. Buzdar, A. Latif, and L. Sun. FPGA prototype implementation of digital hearing aid from software to complete hardware design. *International Journal of Advanced Computer Science and Applications*, 7(1), 2016.
- [10] V. Hanson and K. Odame. Real-time embedded implementation of the binary mask algorithm for hearing prosthetics. *IEEE Transactions on Biomedical Circuits and Systems*, 8(4):465–473, Aug 2014.
- [11] S. G. Kambalimath, P. C. Pandey, P. N. Kulkarni, S. S. Mahant-Shetti, and S. G. Hiremath. Fpga-based design of a hearing aid with frequency response selection through audio input. In *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, pages 579–580, Jan 2016.
- [12] L. Bendaouia, H. Salhi, S. M. Karabernou, L. Kessal, and F. Ykhlef. Fpga-implementation of a bio-inspired medical hearing aid based dwt-ola. In *2014 International Conference on Audio, Language and Image Processing*, pages 806–811, July 2014.
- [13] Geng Yang, Jian Chen, Ying Cao, Hannu Tenhunen, and Li-Rong Zheng. A novel wearable eeg monitoring system based on active-cable and intelligent electrodes. In *HealthCom 2008 - 10th International Conference on e-health Networking, Applications and Services*, pages 156–159, July 2008.
- [14] Y. Hong, I. Rajendran, and Y. Lian. A new eeg signal processing scheme for low-power wearable eeg devices. In *2011 Asia Pacific Conference on Postgraduate Research in Microelectronics Electronics*, pages 74–77, Oct 2011.
- [15] M. Knieser A. Hayim and M. Rizkalla. Dsp/fpgas comparative study for power consumption, noise cancellation, and real time high speed applications. *Journal of Software Engineering and Applications*, 3(4):391–403, 2010.
- [16] G. C. Cardarilli, A. Del Re, A. Nannarelli, and M. Re. Power characterization of digital filters implemented on fpga. In *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, volume 5, pages V–801–V–804 vol.5, 2002.
- [17] J. P. Oliver, J. Curto, D. Bouvier, M. Ramos, and E. Boemo. Clock gating and clock enable for fpga power reduction. In *2012 VIII Southern Conference on Programmable Logic*, pages 1–5, March 2012.
- [18] K. I. Kum and W. Sung. Word-length optimization for high-level synthesis of digital signal processing systems. In *1998 IEEE Workshop on Signal Processing Systems. SIPS 98. Design and Implementation (Cat. No.98TH8374)*, pages 569–578, Oct 1998.
- [19] R. Jevtic and C. Carreras. Power measurement methodology for fpga devices. *IEEE Transactions on Instrumentation and Measurement*, 60(1):237–247, Jan 2011.
- [20] A. J. Casson and E. Rodriguez-Villegas. Toward online data reduction for portable electroencephalography systems in epilepsy. *IEEE Transactions on Biomedical Engineering*, 56(12):2816–2825, Dec 2009.
- [21] T. J. Chen, C. Jeng, S. T. Chang, H. Chiueh, S. F. Liang, Y. C. Hsu, and T. C. Chien. A hardware implementation of real-time epileptic seizure detector on fpga. In *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 25–28, Nov 2011.
- [22] A. Page, C. Sagedy, E. Smith, N. Attaran, T. Oates, and T. Mohsenin. A flexible multichannel eeg feature extractor and classifier for seizure detection. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(2):109–113, Feb 2015.
- [23] C. I. Jeong, P. I. Mak, C. P. Lam, C. Dong, M. I. Vai, P. U. Mak, S. H. Pun, F. Wan, and R. P. Martins. A 0.83- μ W QRS detection processor using quadratic spline wavelet transform for wireless eeg acquisition in 0.35- μ m cmos. *IEEE Transactions on Biomedical Circuits and Systems*, 6(6):586–595, Dec 2012.
- [24] C. F. Zhang and T. W. Bae. Vlsi friendly eeg qrs complex detector for body sensor networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(1):52–59, March 2012.
- [25] Addanki Purma Ramesh, G.Nagarjuna, and G.Siva Raam. Fpga based design and implementation of higher order fir filter using improved da algorithmr. *International Journal of Computer Applications*, 35(9):0975–8887, 2011.
- [26] A. Tobola, F. J. Streit, C. Espig, O. Korpok, C. Sauter, N. Lang, B. Schmitz, C. Hofmann, M. Struck, C. Weigand, H. Leutheuser, B. M. Eskofier, and G. Fischer. Sampling rate impact on energy consumption of biomedical signal processing systems. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6, June 2015.
- [27] A. Tobola, C. Espig, F. J. Streit, O. Korpok, B. Schmitz, C. Hofmann, M. Struck, C. Weigand, H. Leutheuser, B. M. Eskofier, and G. Fischer. Scalable ECG hardware and algorithms for extended runtime of wearable sensors. In *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on*, pages 255–260, May 2015.
- [28] R. Brennan and T. Schneider. An ultra low-power DSP system with a flexible filterbank. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, volume 1, pages 809–813 vol.1, Nov 2001.
- [29] C. Qian, J. Shi, J. Parramon, and E. Snchez-Sinencio. A low-power configurable neural recording system for epileptic seizure detection. *IEEE Transactions on Biomedical Circuits and Systems*, 7(4):499–512, Aug 2013.
- [30] R. W. Schafer and L. R. Rabiner. A digital signal processing approach to interpolation. *Proceedings of the IEEE*, 61(6):692–702, June 1973.
- [31] T. Parks and J. McClellan. A program for the design of linear phase finite impulse response digital filters. *IEEE Transactions on Audio and Electroacoustics*, 20(3):195–199, Aug 1972.
- [32] E. Hogenauer. An economical class of digital filters for decimation and interpolation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):155–162, Apr 1981.
- [33] C. Chou, S. Mohanakrishnan, and J. B. Evans. FPGA implementation of digital filters. In *In Proceedings of the Fourth International Conference on Signal Processing Applications and Technology*, pages 80–88, 1993.