

Integrated Hardened DSP on DAC/ADC ICs Improves Wideband Multichannel Systems

Mike Jones, Electrical Design Engineer,
Travis Collins, Software Support Engineer, and
Chas Frick, Applications Engineer

Introduction

Over the past several decades, wireless system channel counts and bandwidths have steadily increased. The driving factors for these modern telecommunication, radar, and instrumentation systems are their data rate and overall system performance requirements. However, these requirements have also increased power envelopes and system complexities, making power density and component-level features more important.

To help address some of these limitations, the semiconductor industry has integrated more channels on the same silicon footprint, thereby reducing watt per channel requirements. Additionally, semiconductor companies are integrating more complex features into digital front ends that ease the off-chip hardware design historically achieved in an application-specific integrated circuit (ASIC) or field programmable gate array (FPGA) fabric. These features can range from generic components like filters, downconverters, or numerically controlled oscillators (NCOs), to more complex application-specific operations.

Signal conditioning and calibration problems only become more compounded when developing high channel count systems. This architecture may require unique filters or other digital signal processing (DSP) blocks per channel, thereby making the shift to hardened DSP more important for power savings.

This article presents experimental results utilizing a 16-channel transmit and 16-channel receive subarray in which all transmit and receive channels are calibrated using hardened DSP blocks within the digitizer integrated circuit (IC). The resulting multichannel system provides performance improvements in size, weight, and power when compared to other architectures. When comparing the resource utilization of an FPGA for the system, it becomes clear that the hardened DSP blocks solve significant challenges for designers of multichannel platforms.

Digital Signal Processing Blocks

Real-world signals, whether used for synthesis or reception, require some amount of analysis or processing to collectively achieve the performance required for any application. A common method to compensate for signal chain amplitude droop or flatness is to leverage compensation filters. Figure 1 is an example of a gain and flatness compensation filter that was designed and used to correct imperfections across a given frequency band, thereby creating a more ideal response for downstream applications.

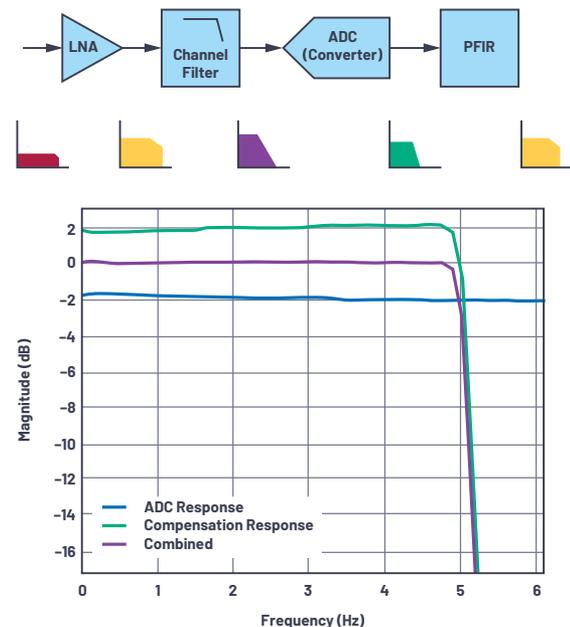


Figure 1. An ADC's amplitude flatness response across frequency can be improved with digital filtering.

For multichannel systems, this processing must allow independent control on a per-channel basis to isolate each channel's performance with respect to another. Therefore, separate DSP blocks are utilized in this system to achieve channel phase and amplitude alignment while also attaining gain flattening within the pass band of interest. Since each channel and system is unique, the DSP must be tuned specifically for that configuration, environment, and hardware lot.

Digital Up/Downconverter Blocks

The results of this article highly utilize digital upconverter (DUC) DSP blocks and digital downconverter (DDC) DSP blocks collocated within monolithic DACs and ADCs. An example DUC and DDC block diagram revealing the often-used innards of these datapaths is shown in Figure 2. These DUC and DDC blocks can serve many useful purposes:

- ▶ Interpolate (DUC) and decimate (DDC) the converter sample rate as compared to the digital interface's data rate.
- ▶ Translate the frequency of the to-be-synthesized DAC data (DUC) and the digitized ADC data (DDC).
- ▶ Channelize the digital data transmitting on the interface to the baseband processor (BBP).
- ▶ Enable digital gain for each channel to generate code values closer to the system's full-scale value.
- ▶ Allow injection of simple digital tones to ease system bring-up without the need for digital data links.
- ▶ Align the phases of each channel with respect to a common reference.

It is often desired that the digital data rate which is offloaded to or from a converter be different than that of the converter's sample rate to save on system power and improve overall system flexibility. As such, digital upconverter and downconverter blocks are often implemented. DUC blocks allow the transmit waveform data from a BBP to be transmitted at a lower rate than the DAC's sample rate, and thereby allow for interpolated waveform data to be synthesized by the DAC at this higher rate, as shown within the interpolation subblock at the top of Figure 2. Similarly, DDC blocks allow the receive input to be digitized at a higher speed ADC sample rate prior to being decimated and then sent to the BBP at a lower data rate, as shown within the decimation subblock at the bottom of Figure 2.

Additionally, frequency translation is often desired within the digital domain to synthesize or analyze higher frequency analog signals when compared with those signals sent over the digital interface to or from the BBP. Many systems utilize complex-valued NCOs within DUCs and DDCs in order to achieve this frequency translation, as shown in Figure 2. NCOs can be regarded as digital signal generators that can serve as a local oscillator (LO)-equivalent signal that, when sent into a digital mixer also contained within the DUC/DDCs, can increase the transmit waveform's frequency sent into the DAC (as in the DUC case) or decrease the receive waveform's frequency sent out of the ADC (as in the DDC case). Often, when digital frequency translation occurs, the output of these digital mixers inside a DDC becomes complex valued such that both in-phase (I) and quadrature-phase (Q) signals can propagate along a single digital channel ultimately attached to a sole ADC sampling real-valued data. Similarly, the input complex-valued signal to the digital mixer of the DUC's digital gain block becomes real-valued at its output prior to being issued to a sole DAC synthesizing a real-valued signal.

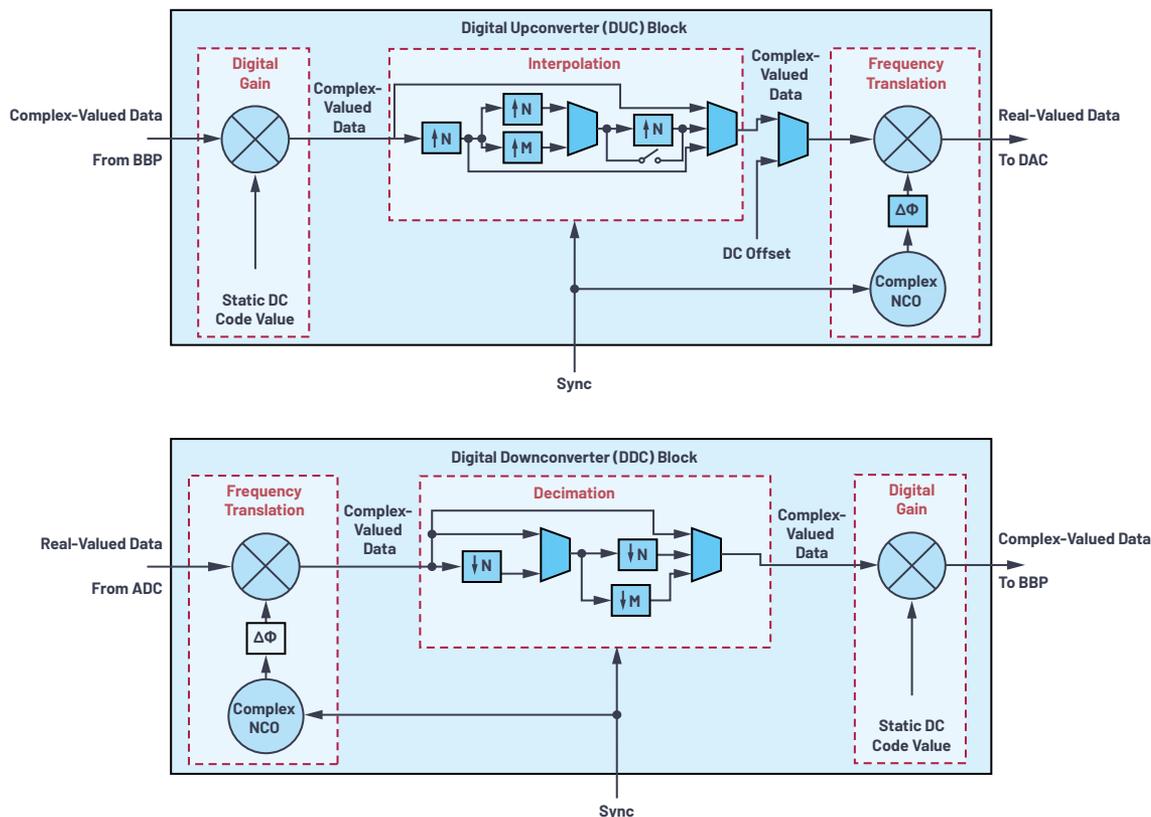


Figure 2. DUC and DDC blocks provide many useful DSP features now within converter ICs.

Furthermore, DUCs and DDCs allow users to achieve multiple digital channels within the converter's instantaneous bandwidth. This results in more data streams capable of being synthesized and/or analyzed by the BBP than the number of converters in the subarray itself. The result is a system that can provide improved signal synthesis or analysis for the case in which two narrow channels separated far apart are desired.

As can be observed in Figure 2, digital gain blocks are also often present in DUCs and DDCs. The digital gain is enabled by providing a static digital code value to the input of another digital mixer in the subblock. Using this feature allows the user to achieve code values closer to the full-scale value provided by the number of bits used for the digital interface. Similarly, DC offset continuous-wave (CW) tones can be injected instead of baseband data by simply providing a continuous static code value into one port of the digital mixer. This allows the user to easily synthesize transmit CW tones via the DAC into the analog domain without the need to establish JESD204B or JESD204C data links with the BBP.

Additionally, phase offset blocks are often implemented at the output of the NCOs, as shown in Figure 2. These phase offsets can be employed to correct channel-to-channel phase anomalies with respect to a common baseline reference present in the system. Since each DUC and DDC contain their own NCO, this allows a method to achieve phase alignment for each channel in the system simply by offsetting the phase of the NCO by a determined amount for a given NCO frequency. The result, when used in conjunction with available multichip synchronization algorithms, is a deterministic phase relationship between all channels that can be corrected with these NCO phase offsets.¹ Figure 3 shows the experimental results of 16 simultaneous receive I/Q data captures before and after achieving phase alignment strictly by setting the required NCO phase offset values for each receive datapath. Note that these digital corrections also correct the RF and microwave impairments located in the front-end networks for each channel.

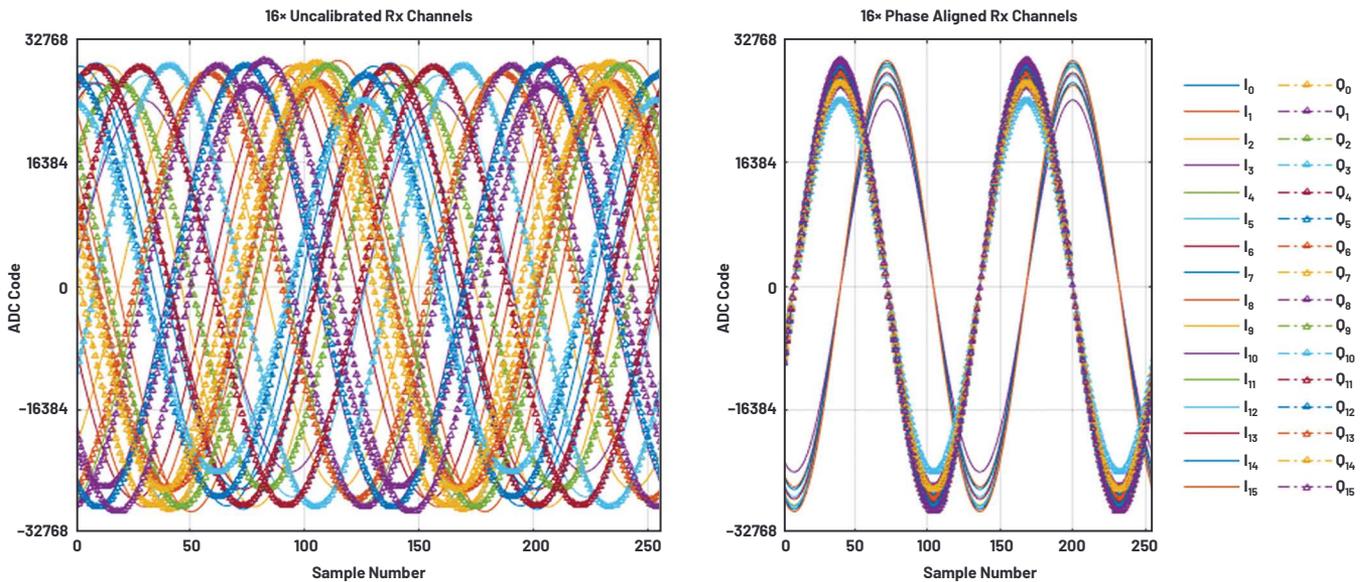


Figure 3. Experimental results showing the simultaneous I/Q capture of 16 receive channels that have been phase aligned (but not amplitude aligned) using strictly the complex NCO phase offsets provided in DDC blocks located on the digitizer IC.

Programmable Finite Impulse Response Filters

While phase offset blocks located at NCO outputs can be employed to create phase alignment at a single frequency, subarray calibrations often require phase alignment throughout the full frequency band of interest. Additionally, amplitude equalization, in which all channels possess nominally identical amplitudes with respect to a common reference channel, and amplitude gain flattening, in which all channels possess a nonchanging amplitude response with respect to frequency, are desired.

To obtain broadband phase and amplitude correction, another DSP block is often employed. These blocks are called finite impulse response (FIR) filters.² A FIR filter is a type of digital filter that is heavily used in DSP, and its coefficients dictate the amplitude and phase response of the input digital signal. Systems that allow these coefficients to be changed are regarded as programmable FIR (pFIR) filters and allow users to generate their own desired magnitude and phase response for each channel.

Achieving Channel Amplitude Alignment and Gain Flattening Using pFIRs

A high-level block diagram of the system used to demonstrate broadband amplitude and phase alignment, as well as gain flattening, is shown in Figure 4. This system employs the use of four digitizing ICs, each containing four transmit and four receive analog channels, or eight transmit and eight receive digital channels. When using all four digitizing ICs in the system, a total of 16 transmit and 16 receive analog channels, or 32 transmit and 32 receive digital channels, are realized. Separate phase-locked loop (PLL) synthesizer ICs are used to provide converter sample clock signals to each digitizing IC. Additionally, a clock buffer IC is used to provide the digital reference and system reference clocks needed for multichip synchronization algorithms.¹ The system is first configured to operate within the S-band and sets the NCO frequencies such that the analog signal for all transmit and all receive channels is the same at 2.7 GHz. The DAC sample rate used is 12 GSPS and the platform synthesizes transmit channels in the first Nyquist. The ADC sample rate is 4 GSPS and the platform captures receive channels in the second Nyquist.

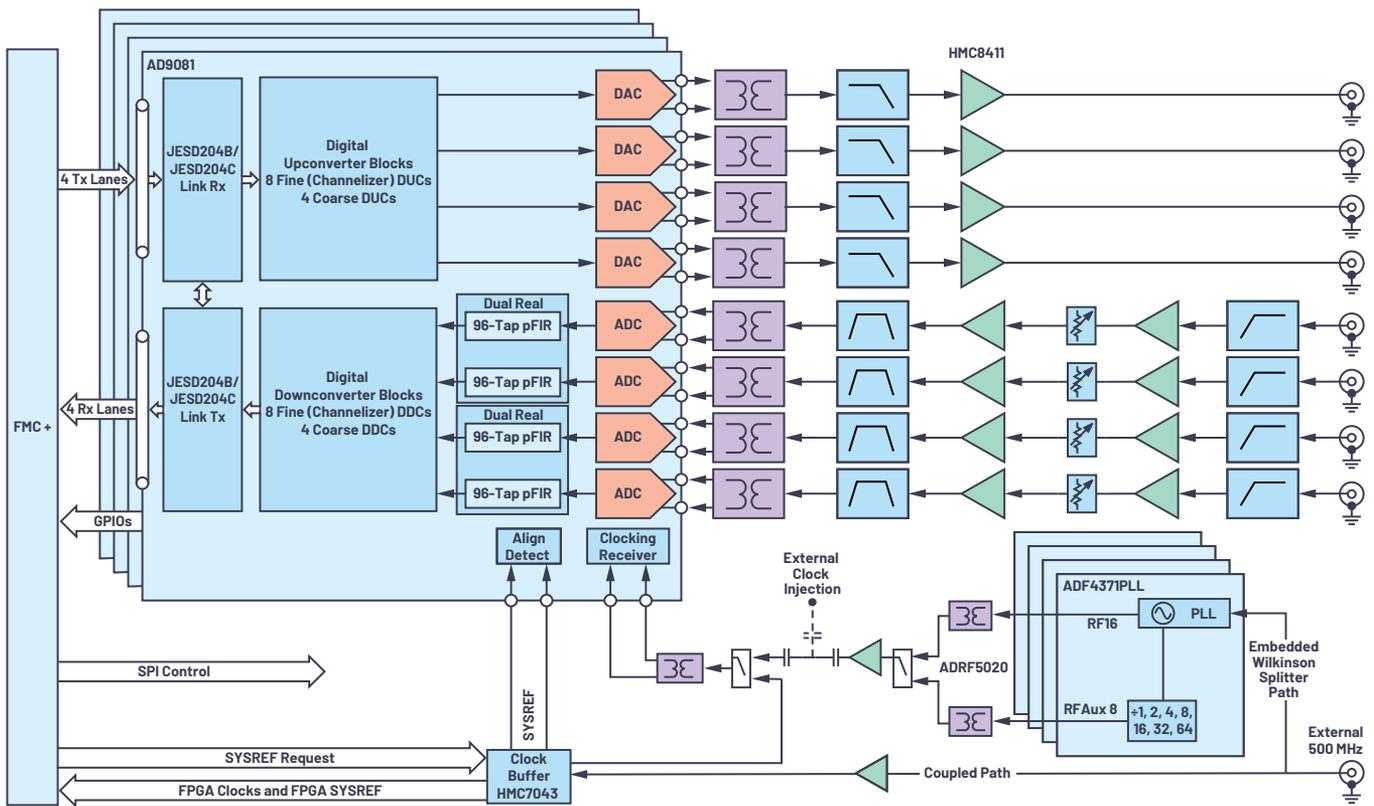


Figure 4. This high-level system block diagram is used to demonstrate multichannel phase and amplitude equalization/flatness.

As shown in Figure 5, an attached 16 transmit/16 receive calibration board is used to concisely loop back combined channel transmit signals into each individual receive channel such that a simultaneous capture of all receive channels can be obtained. The system's PLL synthesizers are then aligned using their own phase adjustment blocks, and the transmit channels and receive channels are coarsely aligned using the NCO phase offset blocks provided in the DUCs and DDCs, respectively. This results in the subsystem being roughly phase aligned at the calibration frequency as in the plot shown in Figure 3, but no amplitude alignment is yet achieved. While the 16 transmit/16 receive calibration board is used for this article to electrically align the system, a similar configuration could be obtained over the air with the use of system calibration reflectors that also help to correct any antenna channel-to-channel anomalies.

As shown in Figure 4, 96-tap pFIR filters are located at the output of each ADC, such that the phase and amplitude response of each ADC channel can be aligned to each other throughout the frequency span of the full ADC sample rate. Therefore this places the pFIR between the ADCs and the DDC blocks. As such, the digital interface's data rate is different from that of the pFIR rate, and so knowledge of the amount of frequency translation and rate decimation in the system is required to use the pFIRs for channel amplitude alignment. Since real data is being sampled at the input of each ADC for this article, the pFIR inputs are real valued. Additionally, the system design is configurable such that one pFIR block per ADC pair is the implemented solution, as shown by the dual real block in Figure 4. This alternatively allows the use of I/Q complex inputs into two separate ADCs to enable system alignment.

To amplitude align and amplitude flatten the channels in the system, a wideband chirp waveform is loaded into each transmit channel such that it encompasses all frequencies within the I/Q bandwidth of the system. This allows the user to determine the frequency error response for all frequencies within the system's data rate. Following this, a baseline data capture is obtained at the decimated I/Q data rate. For the results in this article, an ADC sample rate of 4 GSPS and an I/Q data rate of 250 MSPS is employed. As such, each transmit NCO frequency is set to 2.7 GHz and each receive NCO frequency is set to 1.3 GHz due to frequency folding from the second Nyquist down to the first Nyquist. The baseline data is captured using a MATLAB® system interface, and magnitude and phase error responses are calculated for each channel with respect to a gain flattened Rx0 such that the maximum value received for all receive channels is the ideal received input throughout the full I/Q band. Figure 6 shows the phase and amplitude error response of four of the 16 receive channels in the system. Note from the left of Figure 6 that the NCO phase offsets tend to mostly correct the phase errors of each receive channel but, as can be observed from the right of Figure 6, amplitude errors remain in the system. The remainder of the 12 receive channels have similar error responses. Also note that not only are the receive amplitudes unidentical to Rx0, but there is also poor amplitude flatness without the use of additional calibration techniques. These anomalies are intentionally introduced using analog filters in the ADC front-end networks such that amplitude flatness and equalization can be demonstrated.

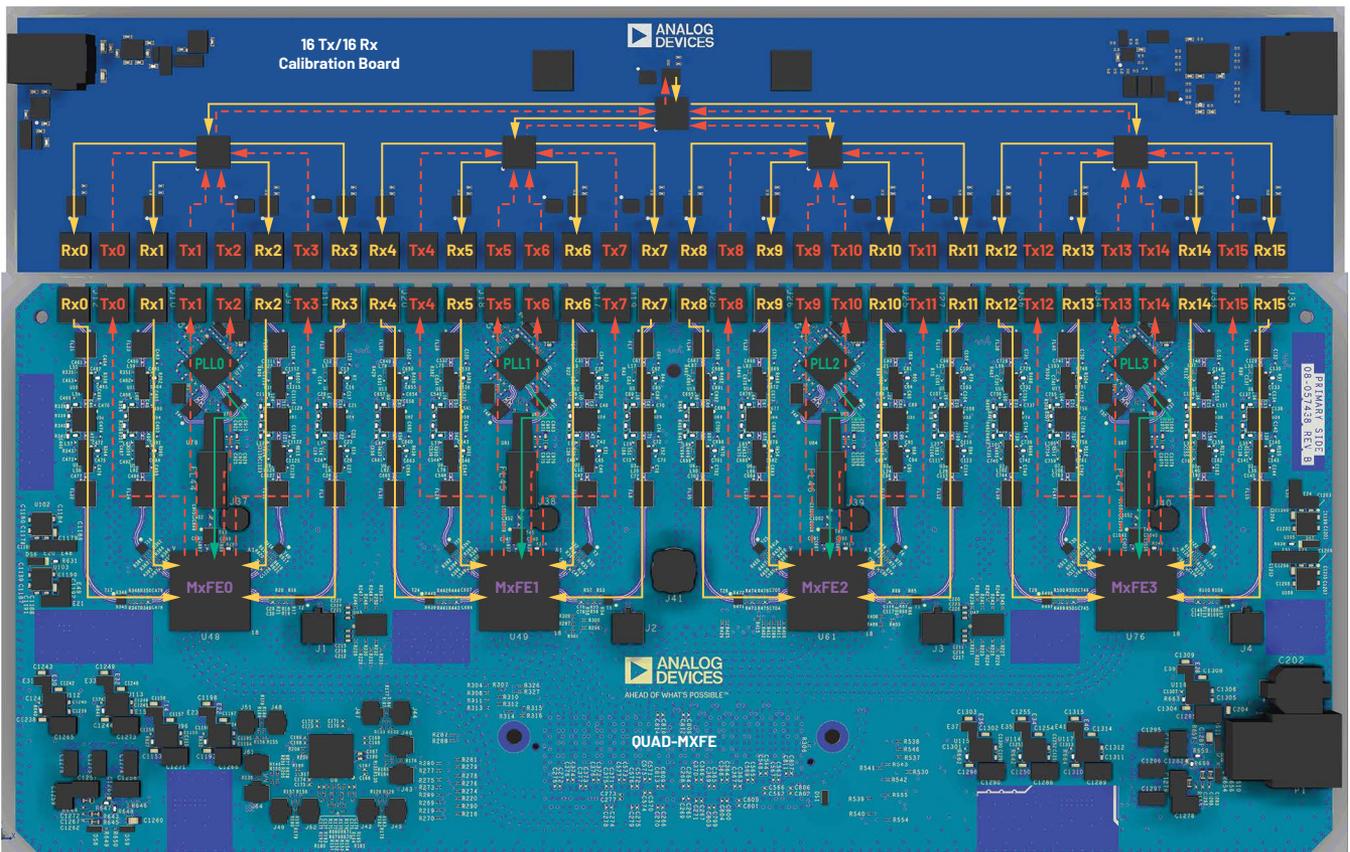


Figure 5. The test setup outputs transmit (red) signals, then combines all transmit signals using an attached 16 transmit/16 receive calibration board. This combined signal is then evenly split and loops back into each receive (orange) channel. The 16 transmit/16 receive calibration board is at the top of the image and mates to the separate Quad-MxFE™ platform used for this article. PLL/synthesizer signals (green) can be adjusted to correct for thermal impairments intentionally introduced on the platform.

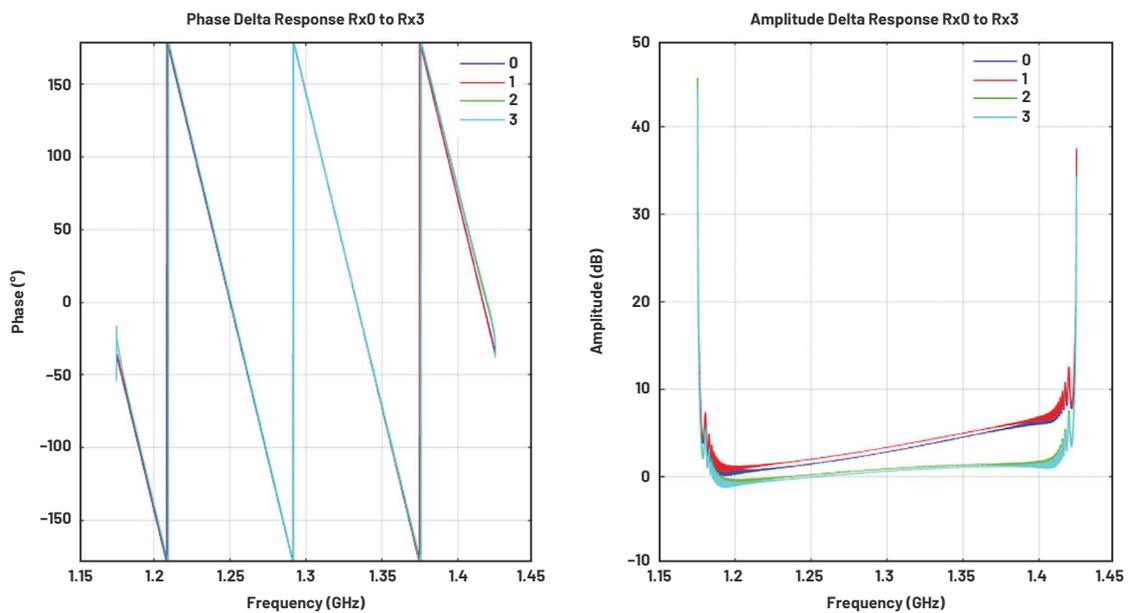


Figure 6. The phase/amplitude error response of each channel with respect to a gain flattened Rx0 helps determine the pFIR filter design.

Therefore, to improve amplitude alignment and amplitude flatness, real-valued 96-tap arbitrary magnitude and phase pFIRs are designed based on the complex-valued error response of each channel with respect to the gain flattened Rx0. Care is taken such that the pFIR design algorithm more heavily weighs the error response of the narrower I/Q band of interest. However, the full pFIR design covers the wider full-rate ADC Nyquist zone, and the regions outside of the 250 MHz sub-band are forced to a unity pass band response. Therefore, for this article, a sub-band of 250 MHz centered at the receive NCO frequency (1.3 GHz) is more heavily weighted for the pFIR design than the remainder of the Nyquist zone. These pFIRs are designed using filter design functions within the DSP System Toolbox in MATLAB, but similar algorithms can alternatively be employed in hardened digital circuitry for fielded systems. Figure 7 shows the designed 96-tap pFIR filters for two of the 16 receive channels used in the example for this article. The pFIR design for the remaining 14 receive channels is similar. Figure 8 shows the designed pFIR magnitude and phase response across the full Nyquist zone for all 16 receive channels in the subarray.

It is important to note that pFIR design algorithms typically design over continuously valued coefficient space between a value of zero and one. However, hardware requires that these continuously valued coefficients be quantized and forced within a certain bit width available on the system. This system uses a varying bit width for the pFIR coefficient space such that some coefficients are 16 bits, others are 12 bits, and still others are only 6 bits. Additionally, the 12-bit coefficients must reside next to the 16-bit coefficients. As can be observed from the coefficient values in Figure 7, only the larger valued coefficients require 16 bits, whereas the smaller valued coefficients only require 6 bits. However, any time one quantizes ideal filter coefficients, a quantization error is introduced, and care is taken to minimize this quantization error for this article such that the designed coefficients still fit within the available coefficient space.

After quantization is performed, the pFIR coefficients are loaded into each channel with the aid of application programming interface (API) functions provided with the digitizer IC. This article uses serial peripheral interface (SPI) communication via the API to modify each channel's coefficients. However, dedicated general-purpose input/output (GPIO) signals can alternatively be used to switch between different coefficient banks more quickly if needed.

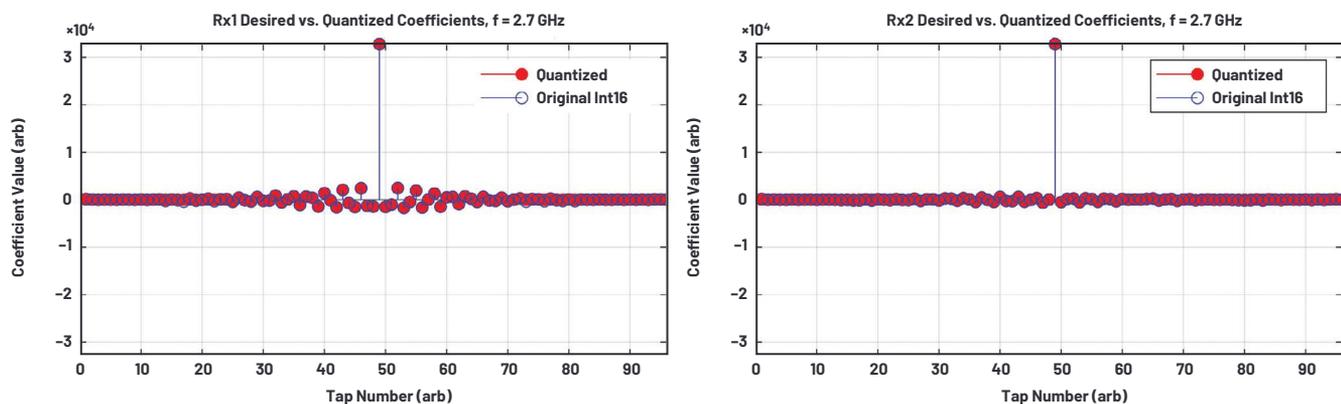


Figure 7. Separate 96-tap pFIRs are designed to provide gain flattening and amplitude alignment across the subarray.

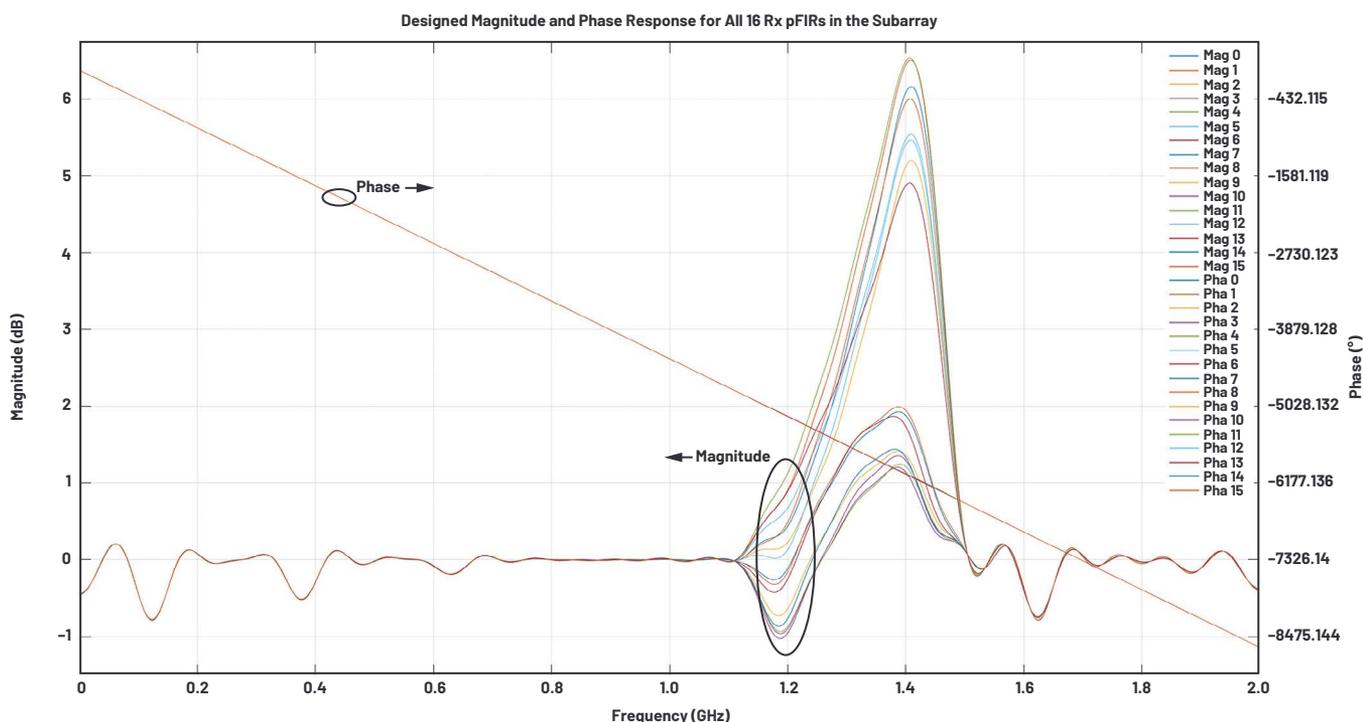


Figure 8. The designed pFIR frequency response for all receive channels shows the applied calibration response for each channel.

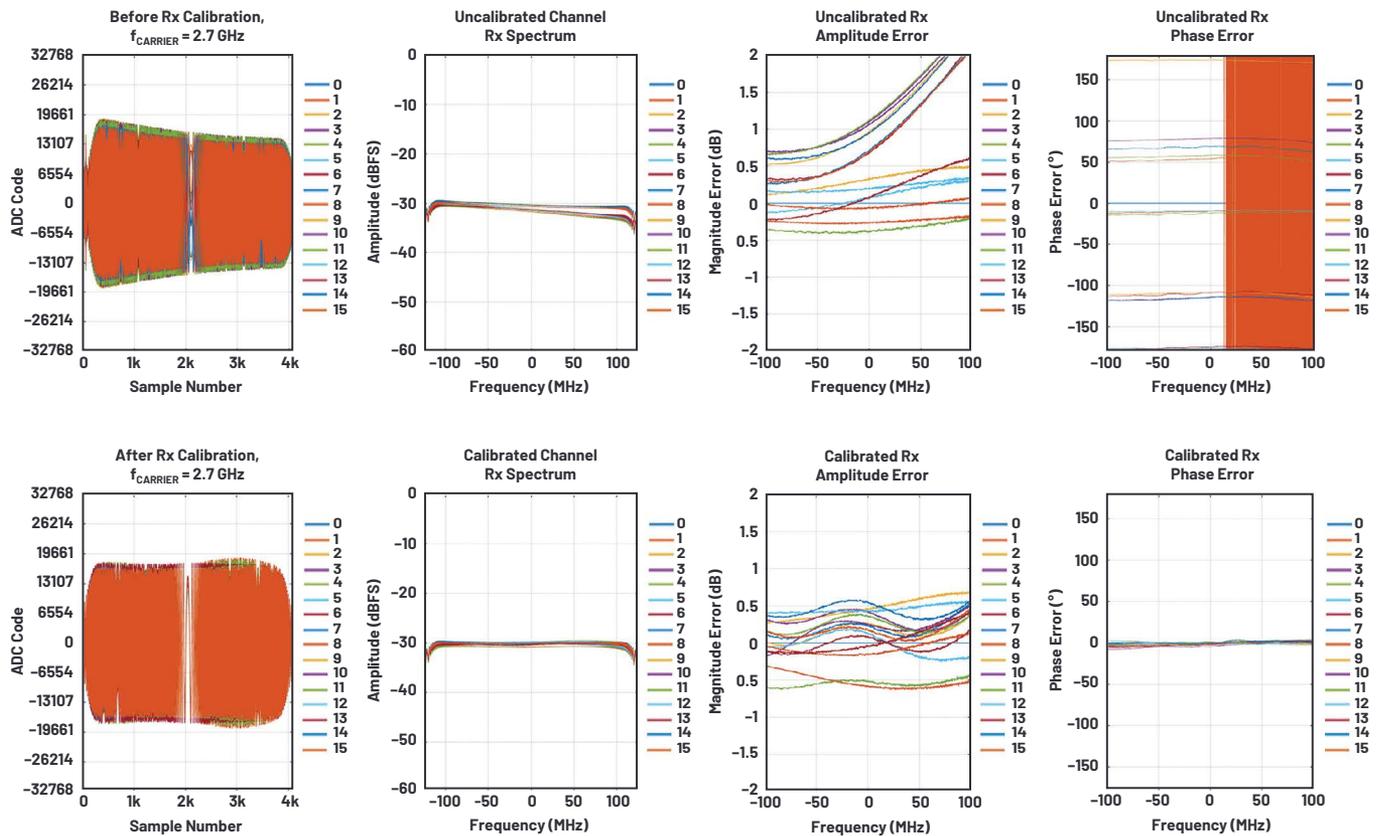


Figure 9. Implementing pFIRs for each receive channel provides improved amplitude equalization and amplitude flatness with respect to Rx0.

Finally, a subsequent receive data capture is obtained while the pFIRs are enabled to analyze the pFIR design's effectiveness. The top of Figure 9 shows the results prior to enabling the pFIRs. Notice that prior to the amplitude equalization steps, the 16 receive channels have varying amplitude and phases across the frequencies of interest. Also note that eight receive channels have different amplitude flatness responses than do the other eight. However, after designing and enabling the pFIRs for each receive channel, as can be observed from the bottom of Figure 9, the amplitudes of all receive channels are nominally amplitude equalized, amplitude flattened, and phase aligned across the I/Q bandwidth. Additional amplitude and phase equalization improvements can be made with more refined pFIR designs, but that was beyond the scope of this article.

Digitizing Element Resource Consumption vs. FPGA Resource Consumption

As discussed, on-chip hardened pFIRs exist within the ADC datapath before the decimation stages. These pFIRs present users with significant application flexibility as demonstrated, but the pFIRs also allow developers to significantly reduce FPGA resources due to the feature offload to the digitizing IC itself. The obvious question becomes: why use the hardened pFIRs on the digitizing IC instead of within hardware description language (HDL) fabric on an FPGA? This can be answered in several parts: resource reduction, design complexity, and power consumption.

Resource reduction is an important topic regardless of focus area. In the case of the digitizing IC, hardened pFIR blocks are already created and placed. In an FPGA, a FIR filter can be built from DSP slices that contain specific FPGA fabric components intended for DSP functions. FPGA DSP slices are different than traditional logic gates, such as flip-flops, and count toward FPGA resource utilization

separately. To determine if the pFIRs should be used on either the digitizing IC or the FPGA, the utilization of the FPGA—specifically the DSP slice utilization percentage—becomes paramount. As a point of comparison, the chosen VCU118 platform contains an XCVU9P Virtex® Ultrascale+® Xilinx® FPGA consisting of 6840 DSP slices. While this is a relatively large number of DSP slices, the number of channels must also be considered when determining how many filters are to be placed in the fabric.

For this, the desired input sample rate of the filter must be known. Table 1 shows an estimated number of resources required when synthesizing an FIR design on the FPGA for several use cases that map to potential digitizing IC datapath configurations. These estimated number of resources for each filter comes from the Xilinx LogiCORE™ IP FIR Compiler 7.2 block summary. To view this summary, a filter was added to a simplified MicroBlaze® design, seen in Figure 10, that was created in Xilinx Vivado™ Design Suite 2018.2. The 250 MSPS and 1 GSPS rates are situations where the FIRs would be operating with decimated data from the converters, while the 4 GSPS case assumes that the data is input undecimated directly from the converter. Each FIR filter runs at 250 MHz to mimic the speed at which an FIR filter would operate if it were in the baseband datapath and contains 96 16-bit reloadable coefficients.

Given the percentage utilization of the XCVU9P FPGA, it becomes clear that a larger FPGA such as the XCVU13P (with 12,288 DSP slices) must be utilized to contain all the filters required. In the case of 4 GSPS FIR filters, at least two of the XCVU13P devices are required to share the resource loading of all the filters, which in turn drives the cost of the design up. In contrast, all filters required for all 16 channels used on the hardened DSP pFIR implementation discussed in the earlier sections of this article are completely contained within the digitizing ICs themselves to achieve a less complicated system design approach.

Table 1. Increased FIR Sample Rates Lead to Increased FPGA Resource Utilization Beyond Existing Capabilities and Drastically Increased System Power Consumption

FIR Input Sample Rate	DSP Slices per Filter in FPGA	Filters Required in FPGA	Total DSP Slices for All Filters in FPGA	Utilization of XCVU9P (6840 Total DSP Slices) (%)
250 MHz	96	32	3072	45
1 GHz	384	32	12288	180
4 GHz	1536	16	24576	359

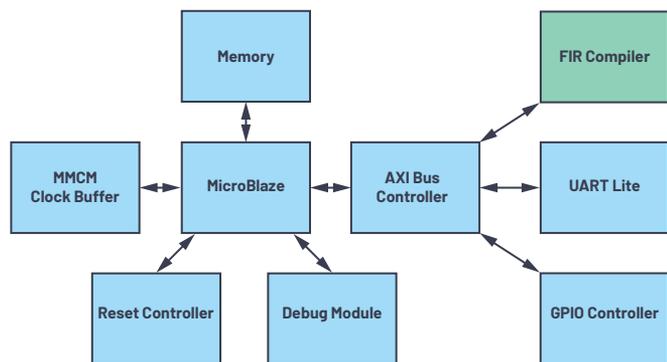


Figure 10. An example MicroBlaze design with one FIR filter is implemented in the FPGA to determine resource utilization.

Another major concern with the FIRs inside an FPGA is the design complexity associated with the high DSP slice resource utilization. Consider how to build the filter. On silicon, the design of the filter is fixed in a single location in the chip, but the coefficients and weightings can be digitally altered, resulting in a relatively static implementation. In FPGA fabric, the FIR filter design routes those DSP slices in various regions of the chip. This means that as the filter grows or changes, more area of the FPGA is consumed and routing connections between DSP slices becomes more and more challenging. Secondly, the routing of the rest of the FPGA design can be impaired by expanding the FIR filter design, which can make timing-critical routing difficult, if not impossible, in certain situations.

Digitizing Element Power Consumption vs. FPGA Power Consumption

The general industry trend of increased converter sample rates and multichannel integration often leads a system architect to analyze the system power consumption when implementing DSP blocks in the overall design. Historically, these DSP blocks have been implemented with the use of programmable logic, such as is found within an FPGA. However, implementing configurable blocks within the FPGA can often create excess overall system power consumption.

Table 2. Increased FIR Sample Rates Lead to Increased System Power Consumption

FIR Input Sample Rate	Filters Required in FPGA	Measured Power per Filter in FPGA (W)	Calculated Power for All Filters in FPGA (W)	Power Burn per Filter in FPGA (W) (From XPE Tool)	Worse-Case Total Power for All Filters in FPGA (W) (From XPE Tool)	Measured Power per Filter Using Hardened DSP in Digitizer IC (W)	Measured Power for All Filters Using Hardened DSP in Digitizer IC (W)
250 MHz	32	0.075	2.40	0.391	13	X	X
1 GHz	32	0.22	7.04	1.564	50	X	X
4 GHz	16	0.81	12.96	6.254	100	0.405	6.48

To try to directly compare both systems, several simplistic reference designs were created for the VCU118 to determine the relative difference in power consumed by the FPGA-based filter approach in a realistic scenario. The VCU118 was chosen since, at the time, it had the most DSPs of any evaluation system directly provided and supported by Xilinx. Based on the VCU118, two Vivado projects were created for each FIR input sample rate: one with filters and one without. For both the 250 MHz and 1 GHz cases, eight FIR filters were inserted into the design similar to that shown in Figure 10. In the 4 GHz case, only two FIR filters were inserted into the design due to high resource utilization. Each filter was fed using the output Xilinx LogiCORE DDS Compiler 6.0 block to ensure valid data was used. It was also important to note that the RTL was examined after synthesis to verify that filters remained in the design, making sure they were not optimized away. In the second design for each sample rate, the filters were removed, but all the other IP blocks remained in place.

Once implemented, the designs were booted, and current measurements were taken to create a relative power delta to isolate the additional power required by the filters. The current draw of the filters can be seen in Table 2 under the measured power per filter column. The total power draw for all the filters was then extrapolated using the data collected for a limited number of filters in the design (eight filters for 250 MHz and 1 GHz and two filters for 4 GHz). This delta was the basic unit of comparison which was used to scale to different configurations that are not implementable with the VCU118, but possible with the digitizer IC. The authors believe this is relatively fair or possibly advantageous toward the FPGA since a real system's power draw is unlikely to scale linearly. Lastly, the results were compared against the power estimates for various filter implementations generated from the Xilinx Power Estimator (XPE) tool.³ The power estimates are much higher than the extrapolated results, but this could account for the nonlinear increase in power because of the increased utilization.

To compare the power draw of the FIRs in the FPGA vs. the hardened pFIRs in the digitizer IC, the measurements from the simple filter designs were compared to the actual current draw of the multichannel system, which uses hardened pFIR DSP blocks on the digitizer ICs. Including all the front-end networks and clocking circuitry, the total system power consumption using the digitizer IC platform without the hardened pFIRs enabled is approximately 98.40 W. When all 16 hardened pFIRs are enabled, the total system power consumption using the digitizer IC platform is approximately 104.88 W. Therefore, the power consumption delta of the hardened pFIRs used in the multichannel platform is around 6.48 W total for all 16 receive channels on the digitizing IC system. The hardened pFIRs are receiving data from the ADCs directly and must run at the ADC sample rate (4 GSps) for this present generation.

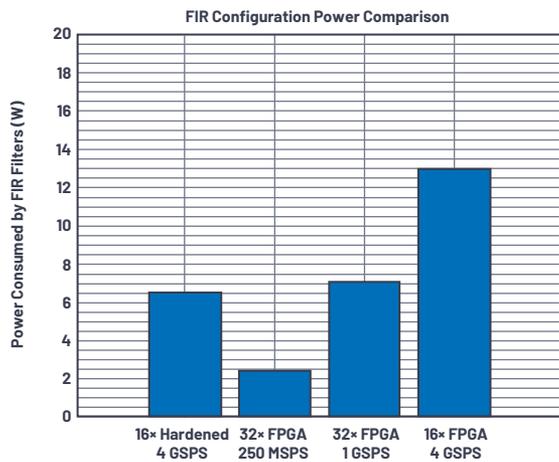


Figure 11. The hardened DSP blocks within the digitizer IC lead to system-level power consumption improvements.

However, comparing this power consumption to that consumed as if there were 16 4 GSPS FPGA FIRs is a bit unreasonable since the resource utilization is impossibly high for a single Virtex Ultrascale+ series FPGA. Therefore, the 250 MSPS rate FPGA FIRs are compared to the hardened 4 GSPS pFIRs, with Table 2 and Figure 11 showing that the power consumption of the 32 FPGA FIRs (16 I FIRs and 16 Q FIRs) is 2.40 W. The filters in the FPGA are running 16 \times more slowly than those in the hardened digitizer IC DSP blocks, but the FPGA still consumes 0.37 \times as much power as the hardened digitizer ICs. Comparing the 32 1 GSPS FPGA FIRs to the hardened 4 GSPS pFIRs, the FPGA FIRs draw about 7.04 W (which is already higher power consumption than the hardened pFIRs) while operating 4 \times slower than the hardened pFIRs. Comparing the 16 4 GSPS FPGA FIRs to the 16 hardened 4 GSPS pFIRs, the FPGA consumes 2 \times as much power with this system configuration. In summary, Figure 11 shows that the power consumption of the hardened pFIRs in the digitizing IC is less than the corresponding FPGA FIR filters. Additionally, the hardened pFIRs reduce the utilization of the FPGA DSP slices, which in turn reduces the complexity of the design and brings down the total power consumption. Using the higher rate filters also opens more wideband use cases when a reduction in data rate to 250 MSPS filters may not be possible.

One final factor to consider is the scalability of leveraging a hardened DSP in a device like the digitizer IC, the AD9081, over relying upon FPGA resources. Utilizing 16 channels in many applications may just be a small subarray of a final system. For system integrators who leverage hardened DSPs, like in the AD9081, they will have a more flexible solution at scale, as well as a much simpler signal chain when compared to expanding the back-end processing by adding FPGA resources. The authors have primarily considered systems that possess a central processing model, where all data must converge to a single FPGA eventually, for this argument. In this case, adding more data converters with built-in filtering will require more SERDES lanes as you scale channels, but it is architecturally simple to manage since more FPGA resources are not required. Without these hardened DSP features, a system integrator would be required to connect multiple FPGAs together to have the necessary resources for the same application, which introduces a significant amount of complexity.

Conclusion

A system that uses DSP blocks integrated within monolithic digitizing element ICs is shown, with the specific example demonstrating that these digital blocks can provide multichannel amplitude and phase equalization required for phased array, radar, satellite communication, and electronic warfare applications. A method using pFIR digital filters and DUC/DDC NCO phase offsets shows that multichannel broadband equalization can be achieved without the need for these DSP blocks being synthesized in the FPGA. The system used for this demonstration is shown in Figure 12, and is called the Quad-MxFE Platform⁴ and is available for purchase from Analog Devices. Specifically, AD9081 MxFE ICs have been used as the backbone of the subarray design. Example HDL, MATLAB scripts, and user documentation can be found on the ADQUADMxFE1EBZ product wiki page (Analog Devices 2020). The 16 transmit/16 receive calibration board (ADQUADMxFE-CAL) is also available for purchase. Instrumentation and 5G markets may also have interest in performing these techniques for subarray test and measurement or base station development.

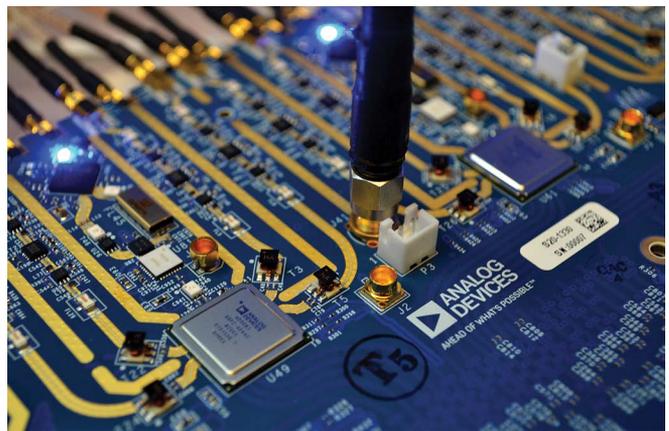


Figure 12. The Quad-MxFE platform is available for purchase from Analog Devices.

References

- ¹ Michael Jones, Michael Hennerich, and Peter Delos. "Power-Up Phase Determinism Using Multichip Synchronization Features in Integrated Wideband DACs and ADCs." Analog Devices, Inc., January 2021.
- ² Mixed-Signal and DSP Design Techniques, Digital Filters. Analog Devices, Inc.
- ³ Xilinx Power Estimator Tool. Xilinx.
- ⁴ Peter Delos, Charles Frick, and Michael Jones. "Multichannel RF-to-Bits Development Platform Enables Rapid Prototyping for Phased Arrays." Analog Devices, Inc., July 2020.

Quad-MxFE Prototyping Platform User Guide. Analog Devices, Inc.

About the Authors

Mike Jones is a principal electrical design engineer with Analog Devices working in the Aerospace and Defense Business Unit in Greensboro, North Carolina. He joined Analog Devices in 2016. From 2007 until 2016 he worked at General Electric in Wilmington, North Carolina, as a microwave photonics design engineer working on microwave and optical solutions for the nuclear industry. He received his B.S.E.E. and B.S.C.P.E. degrees from North Carolina State University in 2004 and his M.S.E.E. degree from North Carolina State University in 2006. He can be reached at michael.jones@analog.com.

Travis Collins holds Ph.D. and M.S. degrees in electrical and computer engineering from WPI. His research focused on small cell interference modeling, phased array direction finding, and high performance computation for software-defined radio. He currently works in the System Development Group at Analog Devices, focusing on applications in communications, radar, and general signal processing. He can be reached at travis.collins@analog.com.

Charles (Chas) Frick is a system applications engineer within the Aerospace and Defense Group in Greensboro, North Carolina. Prior to ADI, Chas received two Bachelor of Science degrees in robotics and electrical engineering from Worcester Polytechnic Institute in 2016. Since joining ADI in 2016, he has worked on PCB design, embedded C code, MATLAB GUIs, Python™ test automation, and version control systems. Outside of work, Chas can most likely be found at the rock climbing gym, the roller skating rink, or helping out with FIRST® robotics. He can be reached at charles.frick@analog.com.

Engage with the ADI technology experts in our online support community. Ask your tough design questions, browse FAQs, or join a conversation.



Visit ez.analog.com



AHEAD OF WHAT'S POSSIBLE™

For regional headquarters, sales, and distributors or to contact customer service and technical support, visit analog.com/contact.

Ask our ADI technology experts tough questions, browse FAQs, or join a conversation at the EngineerZone Online Support Community. Visit ez.analog.com.

©2021 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.

VISIT ANALOG.COM

TA22825-5/21