Improving ADC Results

Through Oversampling and Post-Processing of Data

January 2007

Table of Contents

List of Figures

Actel Fusion™ mixed-signal FPGAs include a configurable 8-,10-, or 12-bit successive approximation register (SAR) analog to digital converter (ADC). This resolution is sufficient for many applications, but for those cases where increased accuracy is needed and bandwidth is not of primary concern, digital post-processing techniques can be used to increase the effective resolution or the effective sample rate of your measurements. This paper introduces these concepts and provides examples to help you enhance the capabilities of the Fusion ADC.

ADC Basics

An analog to digital converter is used to capture discrete samples of a continuous analog voltage and provide a discrete binary representation of the signal. ADC converters are generally characterized in three ways: input voltage range, resolution, and bandwidth (conversion rate).

The input voltage range of an ADC is determined by its reference voltage (V_{REF}). Fusion devices include an internal 2.56 V reference, or you can supply an external reference up to 3.3 V. For the following examples, assume that the internal 2.56 V reference is used, so the full-scale input range of the ADC is 2.56 V. For input signal ranges less than or greater than V_{REF}, an analog scaling function such as that built into the Fusion Analog Quads may be used to amplify or attenuate the input signal, thus matching the input voltage range of the ADC.

Resolution of the ADC is a function of the number of binary bits in the converter. The SAR ADC approximates the value of the input voltage using 2^n steps where n is the number of bits in the converter. Each step therefore represents $V_{REF}/2^n$ volts. For a Fusion ADC configured for 12-bit operation, the least significant bit (LSB) = 2.56 V/4,096 = 0.625 mV.

Finally, bandwidth is an indication of the maximum number of conversions the ADC can perform each second. The bandwidth of an ADC is constrained by its architecture and several key performance characteristics. In 12-bit mode, the Fusion ADC is capable of up to 600 kilo-samples per second (ksps).

There are several popular ADC architectures, each with its own advantages and limitations. The analog to digital converter in an Actel Fusion device is a switched-capacitor SAR ADC. This type of ADC contains N capacitors with binary-weighted values (Figure 1).

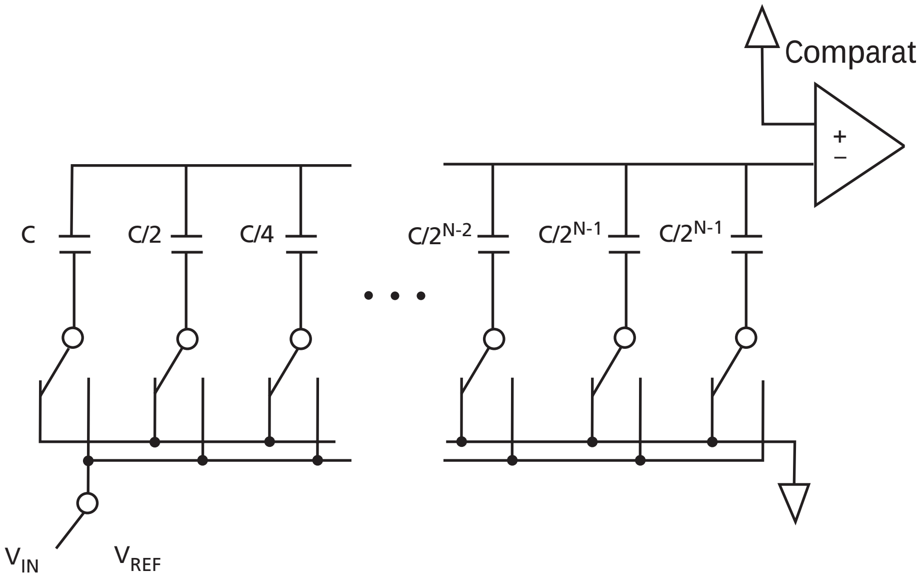


Figure 1: **Example SAR ADC Architecture**

To begin a conversion, all of the capacitors are quickly discharged. Then V_{IN} is applied to all of the capacitors for a period of time (acquisition time) during which the capacitors are charged to a value very close to V_{IN}. Next, all of the capacitors are switched to ground and thus $–V_{IN}$ is applied across the comparator.

Now the conversion process begins. First, capacitor C is switched to V_{REF}. Because of the binary weighting of the capacitors, the voltage at the input of the comparator is $–V_{IN} + ½V_{REF}$. If V_{IN} is greater than $V_{REF}/2$, then the comparator output is 1; otherwise the comparator output is 0. A register is clocked to retain this value as the most significant bit (MSB) of the result.

Next, if the MSB is 0, capacitor C is switched back to ground. Otherwise, it remains connected to $V_{REF.}$ Then capacitor C/2 is connected to V_{REF}. The result at the comparator input is either $–V_{IN} + ¼V_{REF}$ or $–V_{IN} + ¾V_{REF}$ (depending on the state of the MSB), and the comparator output now indicates the value of the next most significant bit. This bit is registered, and the process continues for each subsequent bit until a conversion is completed. The conversion process requires some acquisition time plus N + 1 ADC clock cycles to complete.

This process results in a binary approximation of V_{IN}. Generally, there is a fixed interval (T, the sampling period) between the samples. The inverse of the sampling period is often referred to as the sampling frequency (fs = 1/T).

Figure 2 demonstrates that if the signal changes faster than the sampling rate can accommodate, or if the actual value of V_{IN} falls between counts in the result, information is lost during the conversion. There are several techniques that may be used to address these issues.

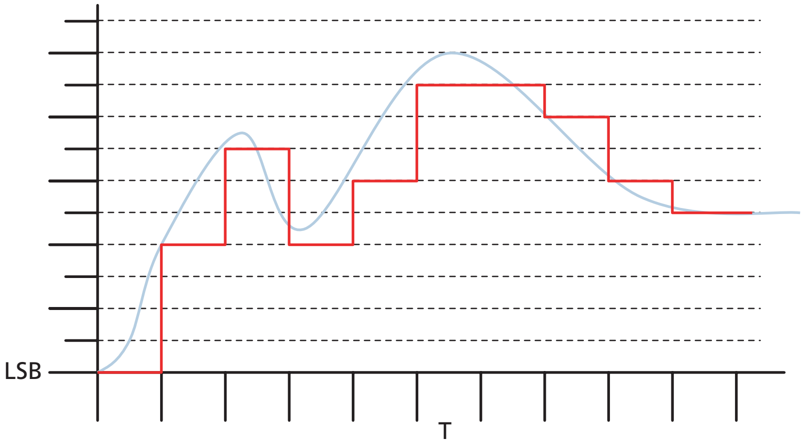


Figure 2: **Analog to Digital Conversion Example**

First, the sampling rate (T) must be high enough to provide enough samples to adequately represent the input signal. Based on the Nyquist-Shannon Sampling Theorem, the minimum sampling rate must be at least twice the frequency of the highest frequency component in the target signal (Nyquist Frequency). For example, to recreate the frequency content of an audio signal with up to 22 kHz bandwidth, you must sample it at a minimum of 44 ksps. However, as you can see from Figure 2, post-processing of the data is required to obtain values of the waveform during the time between each sample. This interpolation may require considerable effort.

Similarly, to faithfully recreate the amplitude variation of a signal, you need to sample the signal with adequate resolution. Continuing with our audio example, the dynamic range of the human ear (the ratio of amplitude of the threshold of hearing to the threshold of pain) is generally accepted to be 135 dB, and the dynamic range of a typical symphony orchestra performance is around 85 dB. It should therefore come as no surprise that most commercial recording media provide about 96 dB of dynamic range using 16-bit sample resolution. A 16-bit ADC is not necessarily required for 16-bit fidelity, however. As long as the input is sampled at or above the Nyquist Frequency, post-processing techniques can be used to interpolate intermediate values and reconstruct the original input signal to within desired tolerances.

If sophisticated digital signal processing (DSP) capabilities are available, the best results are obtained by implementing a reconstruction filter. A reconstruction filter is used to interpolate many intermediate values (increase the effective sample rate) with higher resolution (increase the effective number of bits in the sample) than the original data. In many cases however, it is not cost effective or necessary to implement a sophisticated reconstruction algorithm. For applications that do not require extremely fine reproduction of the input signal, there are alternative methods that can enhance digital sampling results with relatively simple post-processing. The following pages explore some of these techniques, including a brief revisit of reconstruction.

Oversampling

The first step for improving the results of analog to digital conversions is called oversampling. As the name implies, oversampling simply refers to sampling the signal at a rate significantly higher than the Nyquist Frequency. Figure 3 shows that doubling the sampling frequency yields an improved first-order representation of the input signal relative to that obtained with the conversion in Figure 2 on page 4.

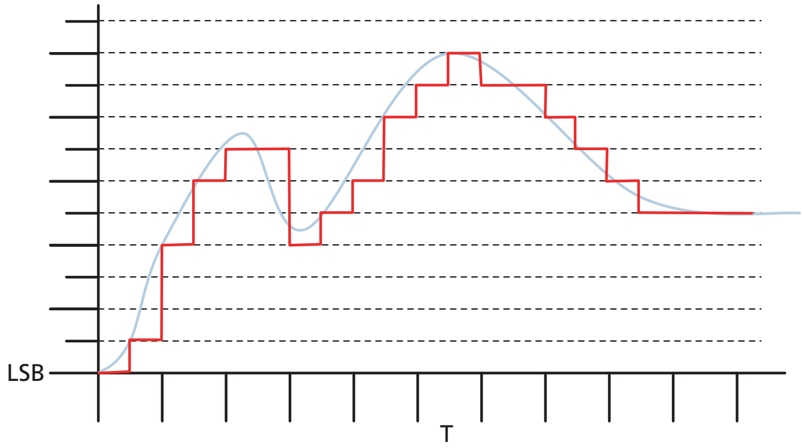


Figure 3: **Input Waveform Sampled at 2x Rate**

Note that the increased sampling rate does not directly improve ADC resolution, but by providing more samples, this technique more accurately tracks the input signal by better utilizing the existing ADC dynamic range. It should be clear that oversampling by itself improves the digital representation of the signal only down to the physical dynamic range limit (minimum step size) of the ADC.

Figure 4 shows that doubling the sampling frequency again results in a series of samples that utilize the available dynamic range of the ADC even more fully.

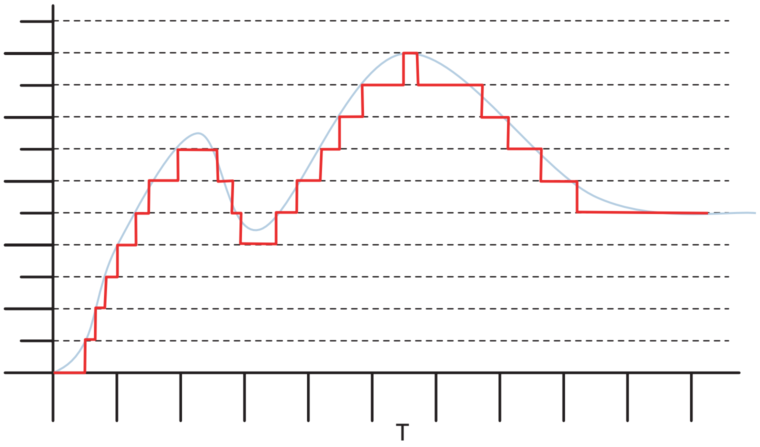


Figure 4: **Input Waveform Sampled at 8x Rate**

Increasing the sampling rate further without additional post-processing simply results in multiple samples of the same value during each step in the waveform, yielding no real improvement in the basic digital representation of the signal.

The high maximum sampling rate of the Fusion ADC supports oversampling of multiple analog inputs under user control. By controlling the system clock rate, sample acquisition times, and the sampling sequence of the Fusion analog subsystem, or by implementing your own state machine to control sequencing and timing of the analog block, you can independently control the sample interval for each analog input in your design.

Oversampling alone is sufficient for many applications, but this technique is often combined with additional post-processing to further improve the digital representation of an input signal without the need for a high-performance DSP.

Averaging

One common post-processing technique is digital low-pass filtering, also called windowing, rolling averaging, or simply averaging. Unlike the other post-processing techniques covered here, averaging is not intended to improve the resolution of the result. Rather, it is a simple and effective way to smooth the input waveform and reduce sensor noise or provide damping for a control input. Averaging is typically implemented with EQ 1 and EQ 2:

$$S(t) = S(t-1) - \frac{S(t-1)}{N} + V_{\text{IN}}$$

EQ 1

$$A(t) = \frac{S(t)}{N}$$

EQ 2

where *S* is an accumulator, *N* is the length of the filter (i.e., the number of samples included in the rolling average), and *A* is the result of the averaging function.

The time constant of the averaging equation is given by $\tau = N/F_S$, where N is again the length of the filter and F_S is the sampling frequency.

Figure 5 shows the step response of the averaging equation with $N = 4$. Also shown (as the dotted line) is the familiar waveform associated with the step response of a single-pole RC low-pass filter with time constant τ = RC (shown equal to the averaging equation's time constant in this case).

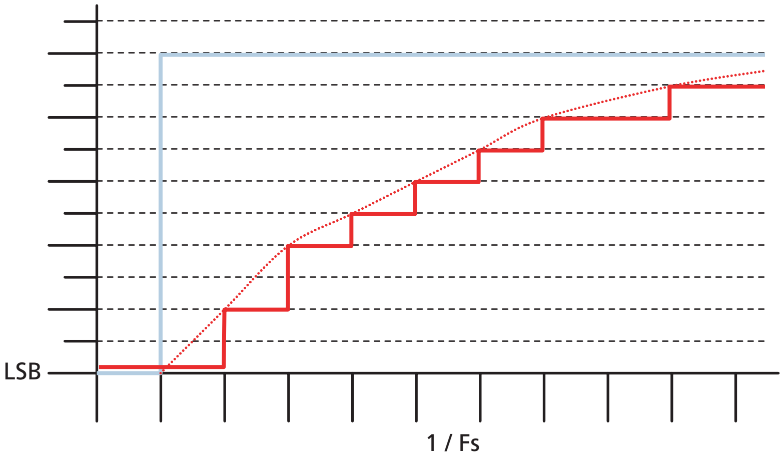


Figure 5: **Averaging Function Response to Step Input**

The green values in Figure 6 illustrate the result when the averaging equations are applied to the samples shown in Figure 4 on page 6 (red).

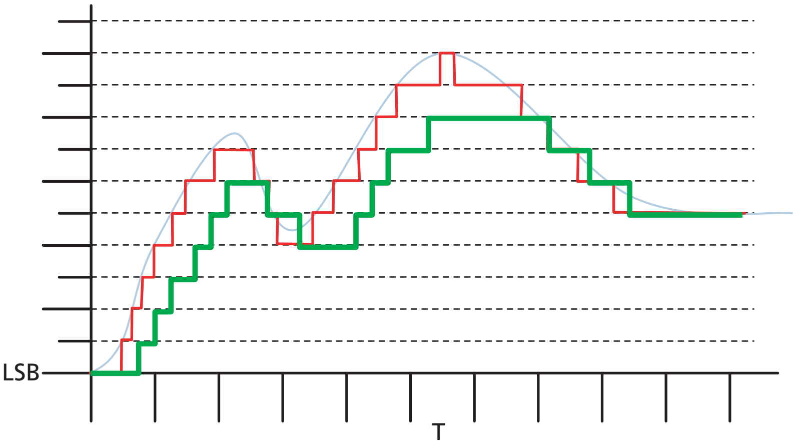
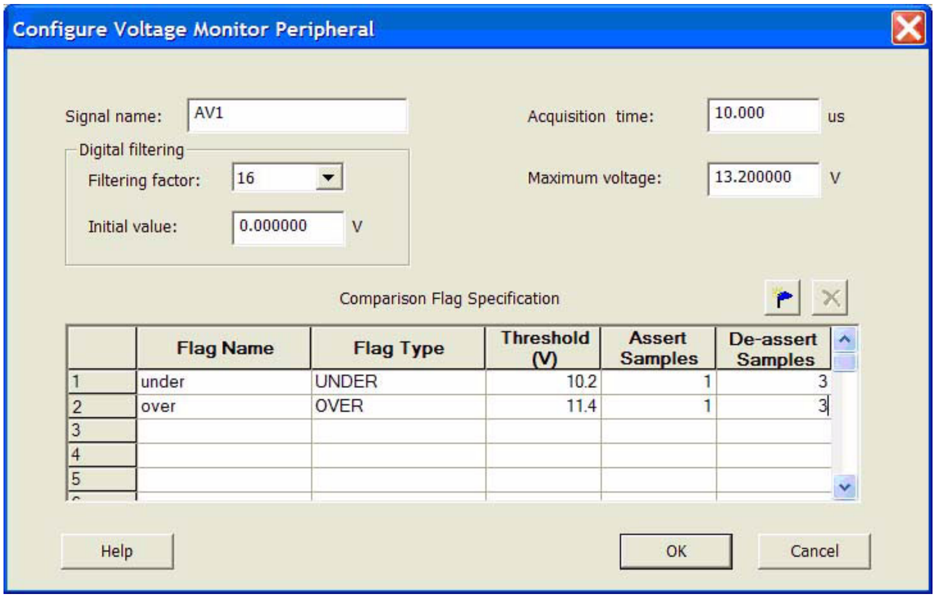


Figure 6: **Digital Post-Filtering of Optimally Sampled Input**

In this example, the result of the calculations is truncated to maintain the original resolution of the digital representation of the input.

The Analog System Builder wizard in the Actel Libero$^{®}$ Integrated Design Environment (IDE) provides easy access to the averaging function. For each analog input that is included in your design, you may choose to include digital averaging simply by selecting this feature in the peripheral configuration window. For example, Figure 7 on page 8 shows the dialog box that configures features for an analog voltage measurement. In this case, digital low-pass filtering of the input is selected and the filter length (*N*) is set to 16 samples. The result is read from the analog subsystem's internal RAM user interface.



Configure Voltage Monitor Peripheral

Signal name: AV1

Acquisition time: 10.000 us

Digital filtering

Filtering factor: 16

Initial value: 0.000000 V

Maximum voltage: 13.200000 V

Comparison Flag Specification

	Flag Name	Flag Type	Threshold (V)	Assert Samples	De-assert Samples
1	under	UNDER	10.2	1	3
2	over	OVER	11.4	1	3
3					
4					
5					

Figure 7: **Analog System Builder Peripheral Configuration Dialog Box**

Alternatively, averaging and other post-processing methods discussed herein can be performed by user-created functions implemented in the FPGA fabric. Samples for each analog input can be captured in real time directly from the ADC_RESULT interface of the analog subsystem and these data can then be post-processed using state machines, digital signal processing blocks, an embedded microcontroller, or any combination thereof.

Enhancing ADC Performance through Digital Post-Processing

Optimal oversampling and digital low-pass filtering are not always enough. For some applications, it is desirable to enhance the sampling rate or resolution of the Fusion ADC in order to represent signal content that might normally require a more costly ADC plus an FPGA or microcontroller. In practice, simple post-processing techniques can increase the effective sampling rate by up to four times and the effective resolution of the ADC by two to four bits. Beyond these limits, it is generally best to consider upgrading the hardware, or moving to a high-end digital signal processor using the Whittaker-Shannon interpolation formula (see the "Reconstruction" section on page 14 for more information).

Interpolation

Anything done to arithmetically enhance the effective sample rate or resolution of the sampling results is, by definition, interpolation; however, this term is generally applied only to techniques that use data from two or more real samples to create a new data series with an increased effective sample rate.

There are many ways to perform an interpolation. The simplest interpolation duplicates each real sample N times to increase the effective sample rate to $N \times F_S$. If logic resources are at a minimum and audio quality is not of paramount concern, then you might use this simple technique to adapt an audio signal that was sampled at 8 ksps to a recording device that expects data at 11 ksps (by duplicating 3 of every 8 samples).

Figure 8 shows a more common use of interpolation. Using the real samples from Figure 2 on page 4, we see the effect of simple two-point linear interpolation to double the number of samples and smooth the resulting digital representation of the input signal.

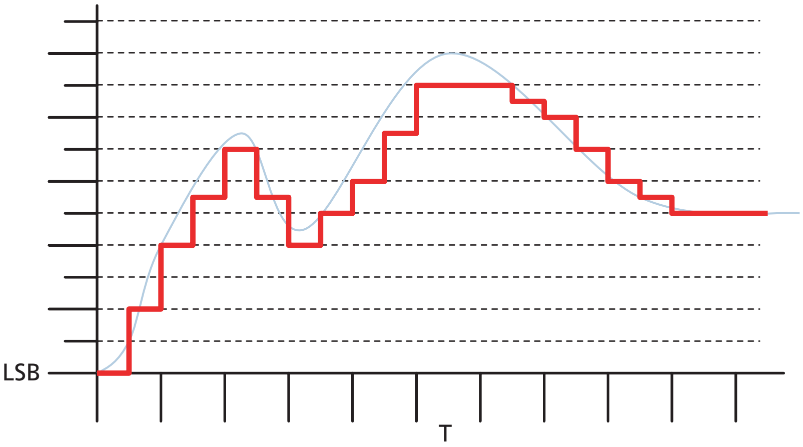


Figure 8: **Linear Interpolation Example**

In Figure 8, each tick on the T axis represents a real sample value. The intermediate values were calculated by summing two adjacent real samples and then dividing this result by two. This doubles the effective sample rate and simultaneously increases the effective resolution by one bit. Performing the linear interpolation again will result in four times the original number of samples, and an increase to 0.25 LSB resolution (an effective two-bit increase).

Implementing a simple linear predictive coding (LPC) algorithm in FPGA logic gates is relatively straightforward. Figure 9 contains a block diagram of the logic required to implement an LPC expansion that enhances the sampling rate by four times and the ADC resolution from 12 to 14 effective bits.

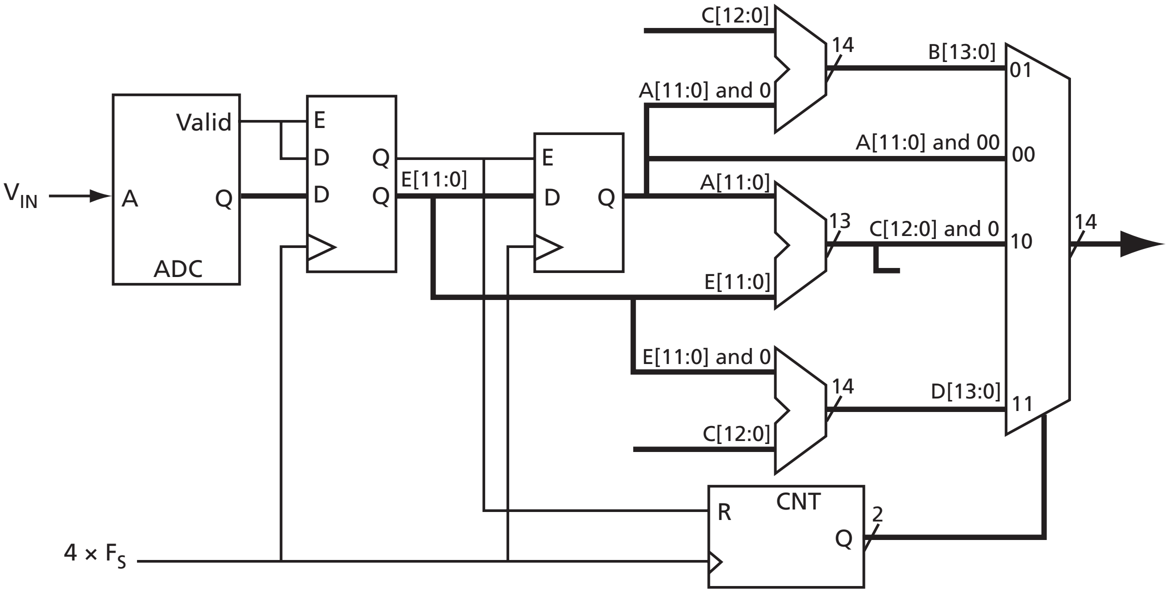


Figure 9: **Implementation of a 2-Bit LPC Expansion**

Figure 9 on page 9 shows that for each pair of 12-bit real samples from V_{IN}, four 14-bit intermediate samples are computed using simple arithmetic and padding of the original samples. In this example, the computations are performed in parallel and the 14-bit output data is multiplexed to the rest of the processing chain at four times the ADC sample rate.

Note that linear interpolation increases both the effective sample rate and the effective resolution of the samples by predicting the value of V_{IN} for the times between real samples, but it can not recover any lost information. For example, Figure 8 on page 9 illustrates the loss of information about the peak of the input signal between T– and T+. Whereas the Whittaker-Shannon interpolation formula (see the "Reconstruction" section on page 14) would recover some information about this peak, interpolating values by simply averaging the two adjacent samples did not.

Decimation

Another method for increasing the effective resolution of the ADC is oversampling and decimation. This technique involves oversampling of the input signal so that a number of samples can be used to compute a virtual result with greater accuracy than a single real sample can provide.

To understand how oversampling and decimation works, consider oversampling the signal in Figure 2 on page 4 by a factor of 16 (i.e., the new sample rate is 16 × F_S). A portion of the signal is shown in Figure 10. This area is the major peak in the signal from time T– to T+ that was lost in both our original sampling and in the linear interpolation. The red line shows the original result using the raw samples from the 12-bit ADC; the green, orange, and blue vertical lines illustrate the sample results at the higher sampling frequency.

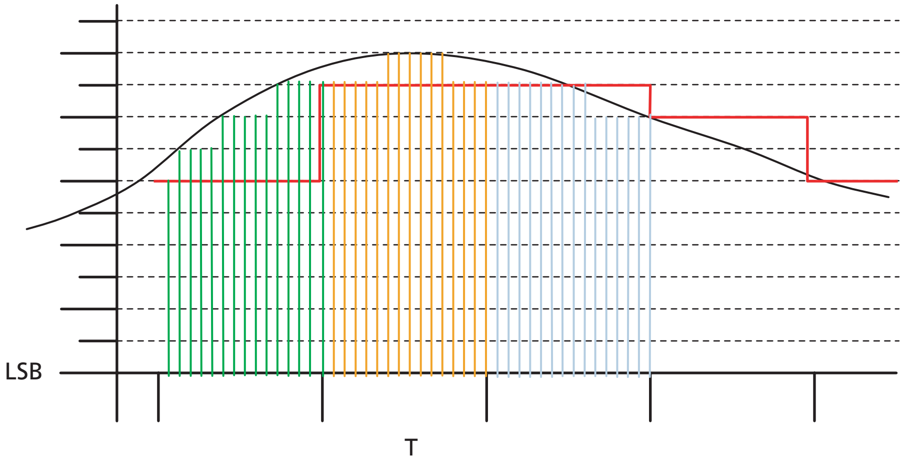


Figure 10: **Example of 16x Oversampling**

Once the oversampling is done, we can simply sum the samples during a given sampling interval to derive a value that represents the value of the input during that sampling interval. In this example, when we sum sixteen 12-bit values, the result is a 16-bit (decimated) result. During the green interval for example, we calculate a sum of 119, which represents an intermediate value of 7.4375 (i.e., 119/16). At this point, we can 1) choose to retain one or more extra bits of effective resolution, 2) truncate the result by using the upper 12 bits, or 3) round off the result by adding bit 3 of the 16-bit result (the 13th bit) to the upper 12 bits. With any of these strategies, the resulting value more closely approximates the actual value of the signal during this interval than did the original data. We can also choose to retain the original sample rate of the data by using only the decimated data points (decimation), or we can combine the decimated values with the raw data to effectively double the sample rate, yielding a decimated interpolation.

In addition to yielding a more accurate approximation of the signal value during a given sampling interval, decimation also helps to improve the signal-to-noise ratio (SNR) of the input signal. By spreading the effects of random noise over multiple samples and computing a sum, decimation allows the noise to at least be partially cancelled from the final result. For every two samples taken during the sampling interval, we cut the noise-floor in half, so with 16x oversampling, we can reduce the noise floor by –12 dB ($1/16^{th}$).

Using Decimation to Increase Measurement Accuracy

For low-bandwidth signals such as power supply voltages, static pressure, and temperature measurements, oversampling and decimation can be used to increase the accuracy of your measurements. There are several criteria that must be met:

1. During the sampling interval required to perform the desired number of ADC conversions, the signal of interest must not change more than 0.5 effective LSB of the end result. For example, if you are using this technique to increase the effective resolution of a 12-bit measurement to 16 bits, the signal of interest must not vary during the sampling interval by more than 1/32 LSB of the ADC.
2. During the sample interval, the ADC must convert the signal 4^n times, where n is the number of virtual bits desired in the result.
3. There must be some noise on the input signal. This noise must have an amplitude greater than 1 LSB of the ADC; it must have an mean value of zero, and it must be randomly distributed (white).
4. The result is truncated by shifting right 2^n places to yield the desired resolution.

Consider the example of controlling the atmospheric pressure in a reaction chamber for a chemical process. The pressure transducer on the reaction vessel is instrumented to provide an input voltage from 0.3 V to 2.2 V for the expected pressure range of 100 to 800 kilo-Pascals (kPa). The processing system is set up to allow a solenoid to increase or decrease the pressure in the reaction vessel at a rate of 25 Pascals (Pa) per millisecond. During a critical phase of the process we will need to control the pressure in the reaction vessel to within 250 Pa. Therefore, we would like to measure the pressure to within ±60 Pa so that our digital control loop has some room to dither within the 250 Pa target range.

Using the direct input of the Fusion Analog Quad and 12-bit resolution with the 2.56 V internal reference voltage, we get a full-scale dynamic range of 0 to 800 kPa × 2.56 V / 2.2 V = 931 kPa. This yields an LSB = 931 kPa / 4096 = 227 Pa. We need to enhance the resolution of our measurements by 2 bits to meet the requirement of 60 Pa resolution.

This can be accomplished by oversampling and decimating the transducer value at $4^2 / 0.00025 = 64$ ksps to yield one decimated sample every 0.25 millisecond. The resulting sum contains 16-bit values that we truncate by using only the upper 14 bits to yield a measurement with an effective LSB = 57 Pa.

To be certain that this scheme will work, we check the required criteria. First, the input will change at a maximum rate of 25 Pa per millisecond or 6.25 Pa per 0.25-millisecond sampling interval, well below the 57 Pa effective resolution that we are seeking. Second, the Fusion ADC can easily provide oversampling at the 64 ksps; in fact, it can monitor up to 8 pressure vessels simultaneously. Next, we need some random noise on the sensor signal. It is worth lingering here to explain this. Figure 11 illustrates the input signal over three sample intervals while the solenoid control is set to increase the pressure in the vessel.

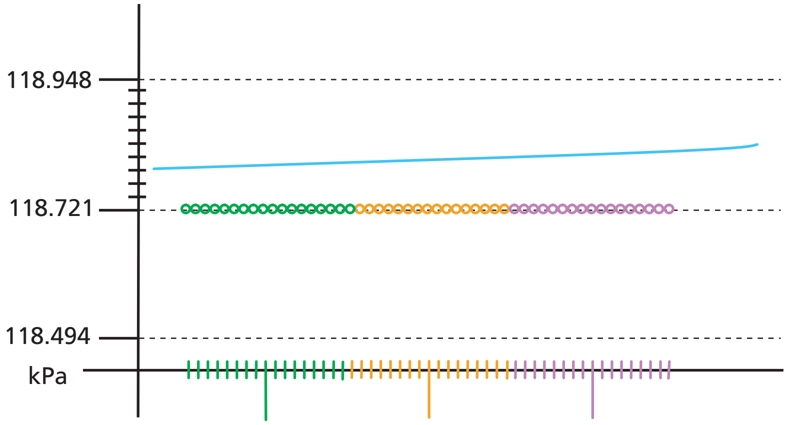


Figure 11: **Slowly Rising Pressure Input with No Noise**

In this example, the actual voltage falls between two steps of the 12-bit ADC resolution and there is no noise riding on the signal. It is easy to see the problem. With no noise on the signal, the ADC result for each conversion will be the same. Averaging produces no effective gain in resolution, and we do not have satisfactory control over the pressure in the vessel.

If, on the other hand, we have some random high-frequency noise of sufficient amplitude to dither the LSB of the conversion, as shown in Figure 12, we see that the decimated sum (red) is weighted by the statistical average of the noise plus the previously undetectable change in the sensor input, yielding an increased sensitivity of effectively two bits.

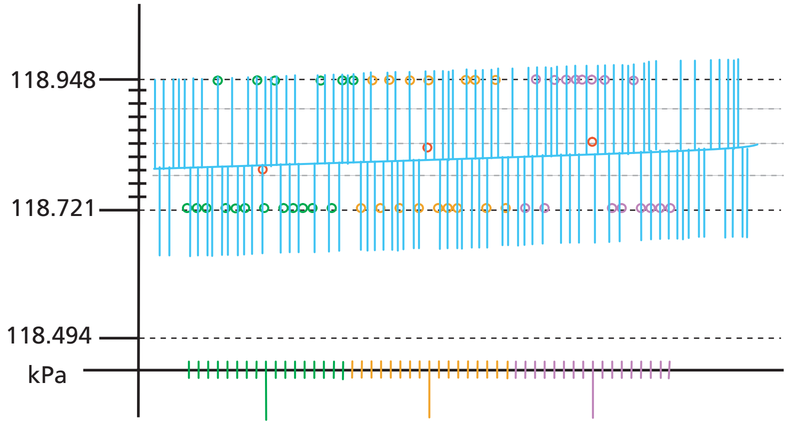


Figure 12: **Slowly Rising Pressure Input With Greater than 1 LSB Noise**

It is important to note that the noise amplitude on the signal must exceed the resolution of the LSB of the ADC. There is no upper limit on the noise amplitude as long as the frequency content represents broad-spectrum (white) noise.

Most systems have an abundance of electrical noise. Electrical noise is radiated from building lights, nearby electrical motors, radio stations, and the sun. Electrical noise is generated on your circuit board from switching power regulators, oscillator chips, and switching digital signals. In many cases, the sum of these noise sources is a good approximation for board-spectrum white noise. However, users of this technique should be aware that any patterns in the noise can lead to an offset in results. If a particular component of the input noise occurs at the ADC conversion frequency (or a harmonic frequency thereof), the sample results will show a DC offset consistent with the non-zero average value of the noise when sampled by the ADC.

In rare cases, environmental noise in the system is not sufficient to effectively dither the input signal to the ADC. In these cases, you may need to inject a noise signal onto your sensor input. This is accomplished with a white noise source and a summing amplifier, as shown in Figure 13.

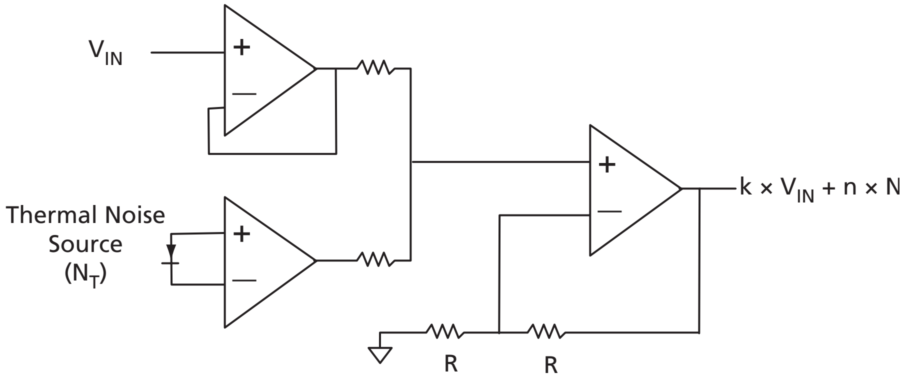


Figure 13: **Circuit to Inject Noise onto a Sensor Input**

One interesting side note concerning Fusion devices is an alternative method for generating a white noise source. A Fusion FPGA device can provide a noise source using a long-count linear feedback shift register (LFSR) to generate a random sequence of pulses, which in turn is fed into a low-pass RC filter to create a randomly fluctuating voltage. This scheme is illustrated in Figure 14.

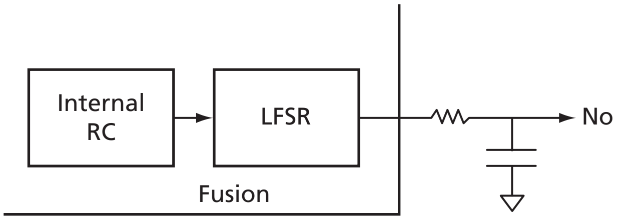


Figure 14: **Generating a Random Noise Source with an LFSR**

To further refine this scheme, the LFSR output can be combined with a polynomial generator or a coding block (i.e., 8b/10b encoder) to ensure a minimum number of transitions during a sampling interval. This yields a predictable DC level that can be used to bias the summing circuit and can then be cancelled out either with an analog offset voltage, or during arithmetic post-processing of the sampled data.

Reconstruction

As mentioned earlier, the best results in faithfully reconstructing the input signal are obtained by implementing a reconstruction filter. The reconstruction filter is used to interpolate many intermediate values to effectively achieve higher resolution than the original data using the Whittaker-Shannon interpolation formula, as shown in EQ 3.

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \times \mathrm{sinc}((t-nT)/T)$$

EQ 3

where T is the sampling interval and sinc is the normalized sinc function given in EQ 4.

$$\mathrm{sinc}(x) = \mathrm{sine}(px) = \frac{\sin(\pi x)}{\pi x}$$

EQ 4

This technique requires significant arithmetic capability and either time/storage capacity, or extremely fast real-time digital signal processing hardware.

The Actel Fusion FPGA can provide either option. A detailed example of this is beyond the scope of this paper, but with both embedded SRAM and nonvolatile memory, a CoreMP7 soft processor, or Synplify® DSP synthesis support—or any combination of these—Actel Fusion devices allow you to blend various digital signal processing techniques to create a customized post-processing solution that fits your needs like a glove.

Summary

The designer of a Fusion mixed-signal FPGA can choose from a powerful array of tools to enhance the capabilities of the Fusion analog to digital converter.

The Actel Libero IDE design suite provides control over acquisition time, sampling sequence, and clock rates, allowing optimization of the sampling rate for each signal and maximizing the potential accuracy of these measurements.

Beyond the built-in features of the design tools, the FPGA logic in Actel Fusion devices offers a wide range of post-processing possibilities. Designers can implement logic or embedded-processor-based functions to adjust, scale, and filter the sampled results. In addition, they can use digital signal processing techniques to reconstruct or interpolate values of the original input signal, or even enhance the effective sampling rate or resolution of the conversions. Actel Fusion mixed-signal FPGAs give you more flexibility than any other ADC solution.

To further refine this scheme, the LFSR output can be combined with a polynomial generator or a coding block (i.e., 8b/10b encoder) to ensure a minimum number of transitions during a sampling interval. This yields a predictable DC level that can be used to bias the summing circuit and can then be cancelled out either with an analog offset voltage, or during arithmetic post-processing of the sampled data.

For more information, visit our website at **www.actel.com**

www.actel.com

Actel Corporation
2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.
River Court, Meadows Business Park
Station Approach, Blackwatery
Camberley Surrey GU17 9AB
United Kingdom
Phone +44 (0) 1276 609 300
Fax +44 (0) 1276 607 540

Actel Japan
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668
www.jp.actel.com

Actel Hong Kong
Suite 2114, Two Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488
www.actel.com.cn