Application Report
SLAA510–January 2011

High-Speed, Analog-to-Digital Converter Basics

Chris Pearson

ABSTRACT

The goal of this document is to introduce a wide range of theories and topics that are relevant to high-speed, analog-to-digital converters (ADC). This document provides details on sampling theory, data-sheet specifications, ADC selection criteria and evaluation methods, clock jitter, and other common system-level concerns. In addition, some end-users will want to extend the performance capabilities of ADCs by implementing interleaving, averaging, or dithering techniques. The benefits and concerns of interleaving, averaging, and dithering ADCs are discussed in this document.

Contents

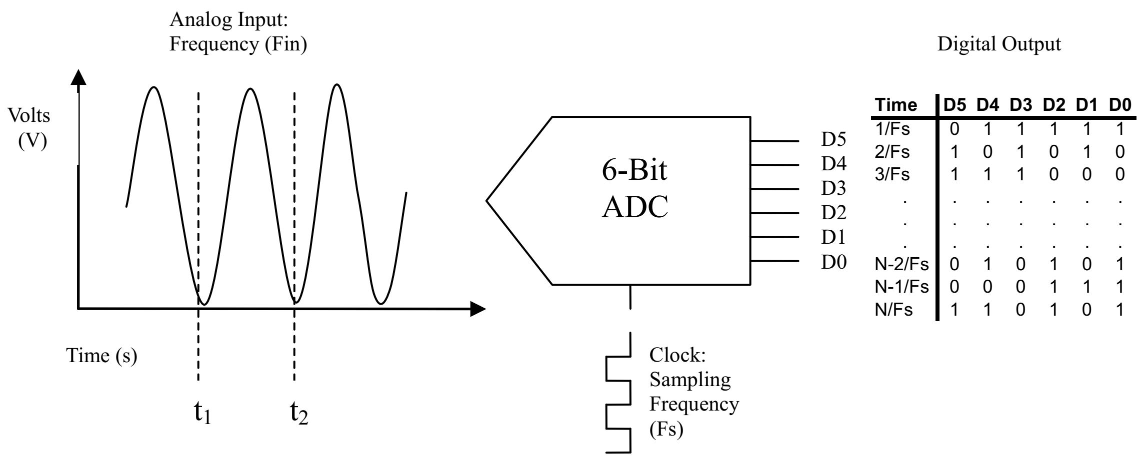
List of Figures

List of Tables

1 Introduction

Analog-to-digital converters (ADC) are devices that sample continuous analog signals and convert them into digital words. ADCs comprise many categories among which are sigma-delta ADCs, high-resolution ADCs, and high-speed ADCs. This application report focuses on high-speed ADCs.

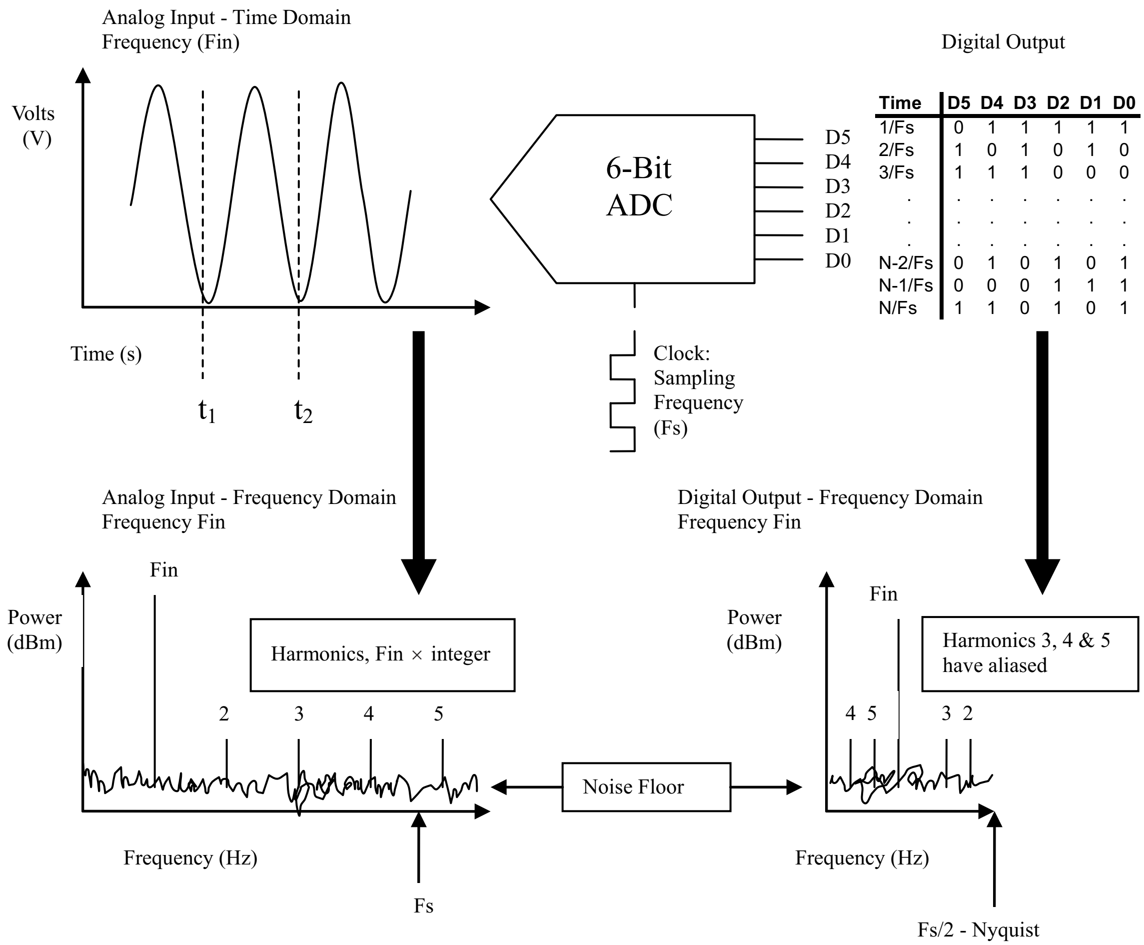
Figure 1 provides the basic block diagram, functionality, and common terminology for ADCs. This figure shows an analog signal applied to the input of the ADC, which then is converted to digital words at the sampling frequency (Fs) applied to the ADC clock. Figure 1 is a time domain representation of the ADC's input and output signals.



Fin: Analog Input Frequency = $1/(t_2 - t_1)$
Fs: Clock Frequency
N: number of digital samples captured
n: number of output bits; in this 6-bit ADC example n = 6

Figure 1. Basic ADC Diagram and Terminology

Time domain representations often are described as real-world signals. In Figure 1, notice that the analog input's amplitude is shown in volts (linear) and seconds (linear). Also notice that the digital output codes are listed with time stamps (1/Fs, 2/Fs, 3/Fs…). Time domain representations are often easy to visualize, and they help with understanding gross concepts. However, time domain representations do an inadequate job of measuring the performance of ADCs and other signal-processing devices. Measuring performance is best accomplished in the frequency domain. Therefore, it is important to understand how the time domain and frequency domain relate.



Fin: Analog Input Frequency = $1/(t_2 - t_1)$
Harmonics: integer multiples of Fin ($2 \times$ Fin, $3 \times$ Fin, $4 \times$ Fin, $5 \times$ Fin…)
Fs: Clock Frequency
Nyquist: Fs/2
Aliasing: the placement of frequencies > Fs/2 (Nyquist) in the Digital Output
Frequency Domain, explained later in more detail

Figure 2. Frequency Domain vs Time Domain

Figure 2 demonstrates a high-level overview of the differences between the time domain and the frequency domain. Frequency domain plots are measured in signal power (log scale) and frequency (linear). In the Figure 2 frequency domain plots, the signal imperfections are labeled as noise and harmonics. Notice the ease with which one can identify and quantify these imperfections. Frequency domain plots also are commonly termed spectrums, spectral plots, or Fast Fourier Transforms (FFT). In Figure 2, the terms Nyquist, harmonics, and aliasing are introduced. These important signal-processing terms are discussed later in more detail.

2 Spectral Performance Terminology

SNR: signal-to-noise ratio. SNR is the ratio of the fundamental (P_S) to the noise floor (P_N), excluding the power at DC and in the first five harmonics. The first five harmonics are labeled 2 to 6 in Figure 3. The fundamental is technically the first harmonic, but is rarely called a harmonic. Some data sheets may exclude the first nine harmonics. Other names for the fundamental tone are signal or carrier. SNR is either given in units of dBc (dB to carrier) when the absolute power of the fundamental is used as the reference, or dBFS (dB to full scale) when the power of the fundamental is extrapolated to the converter full-scale range. See Figure 3 for the SNR equation and illustration.

SFDR: spurious free dynamic range. SFDR is the ratio of the power of the fundamental (P_S) to the next highest spur (P_H). See Figure 3 for the SFDR equation and illustration.

THD: total harmonic distortion. THD is the ratio of the power of the fundamental (P_S) to the power of the first five harmonics (P_D). THD is typically given in units of dBc (dB to carrier). See Figure 3 for the THD equation and illustration. Like SNR, some data sheets may use the first nine harmonics for THD.

SINAD: signal noise and distortion. SINAD is the ratio of the power of the fundamental (P_S) to the power of all the other spectral components including (P_N) and distortion (P_D), but excluding dc. SINAD is either given in units of dBc (dB to carrier) when the absolute power of the fundamental is used as the reference, or dBFS (dB to full scale) Figure 3when the power of the fundamental is extrapolated to the converter full-scale range. See Figure 3 for the SINAD equation and illustration.

ENOB: effective number of bits. ENOB is a measure in units of bits of converter performance as compared to the theoretical ideal SNR limit based on quantization noise (Equation 1). See for the ENOB equation.

$$\mathbf{SNR = 10log_{10}\,(P_S/P_N)}$$
$$\mathbf{SFDR = 10log_{10}\,(P_S/P_H)}$$
$$\mathbf{THD = 10\ log10\,(P_S/P_D)}$$
$$\mathbf{SINAD = 10log_{10}\,P_S/(P_D + P_N)}$$
$$\mathbf{ENOB = (SINAD - 1.76)/6.02}$$

P_S: Signal Power (red)
P_N: Noise Floor Power (blue)
P_D: Power of harmonics 2-6 (black)
P_H: Power of next highest spur (black)

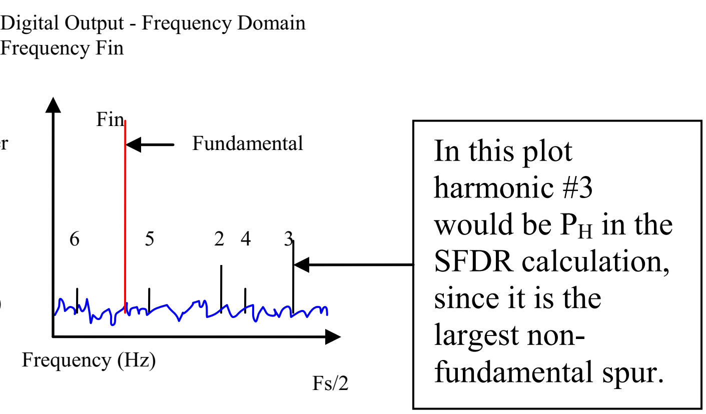


Figure 3. ADC Performance Terminology

Ideal SNR: For a specific converter the ideal SNR can be calculated as shown in Equation 1. This equation is mathematically equivalent to the ENOB calculation, where n = ENOB and Ideal SNR = SNR. Also of importance is that for an ideal converter there is no harmonic content, therefore SINAD=SNR.

$$\text{Ideal SNR} = 6.02 \times n + 1.76$$
$$n = \text{number of bits} \tag{1}$$

As an example, a designer may ask for an ADC with 75-dB SINAD. Using Equation 1,one may assume that the designer requires a 14- or 16-bit ADC (e.g., ENOB = (75 dB – 1.76)/6.02 = 12.2 bits). Other considerations like ADC clock speed, SFDR, bandwidth, and current consumption can further narrow down which 14- or 16-bit ADC the designer requires. These considerations are discussed further.

3 Nyquist, Aliasing, Undersampling, Oversampling, and Bandwidth

The terms Nyquist, aliasing, undersampling, and oversampling are basic ADC terms. They are all closely related, and sometimes this close relationship causes confusion. Once understood, however, the concepts are fairly simple.

The Nyquist-Shannon sampling theorem states that for a true representation of waveform X, greater than two samples per period are required. For an ADC, this means for a true representation of the analog input signal, the clock frequency (Fs) must be two times greater than the analog input frequency (Fin). To state the last sentence in equation form, when Fin < Fs/2, then Fin can be accurately represented. Fs/2 is commonly referred to as the Nyquist Frequency. Figure 4 and Figure 5 illustrate this theorem graphically. In Figure 4, with Fin < Nyquist, notice that the ADC-captured output properly represents the analog input.

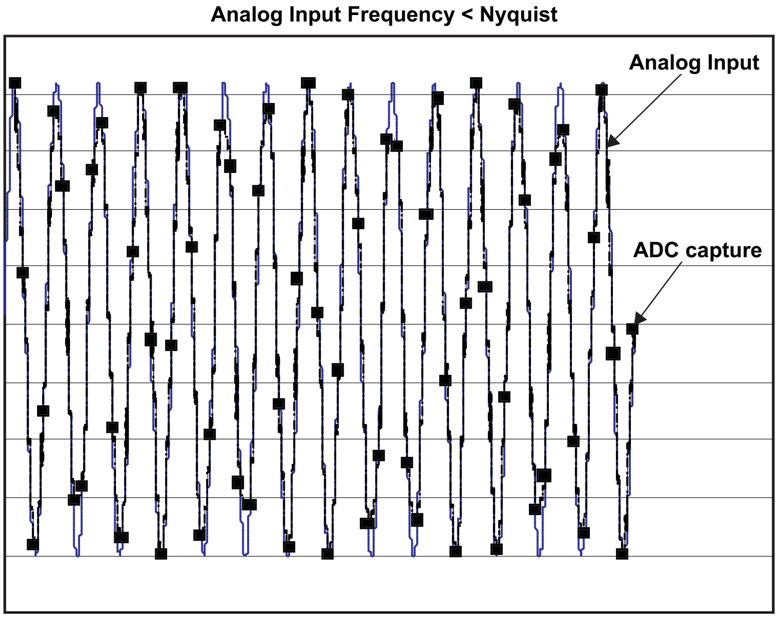


Figure 4. Oversampling Fin < Fs/2

Figure 4 is also an example of ADC oversampling, as more than two samples are captured per the analog inputs period. To state the last sentence in equation form, when Fin < Fs/2, the ADC is oversampling the input waveform.

In Figure 5, with Fin > Nyquist, the ADC capture output has translated the analog input signal at a lower frequency. This frequency translation is called aliasing. As seen in Figure 5, aliasing happens when Fin > Nyquist.

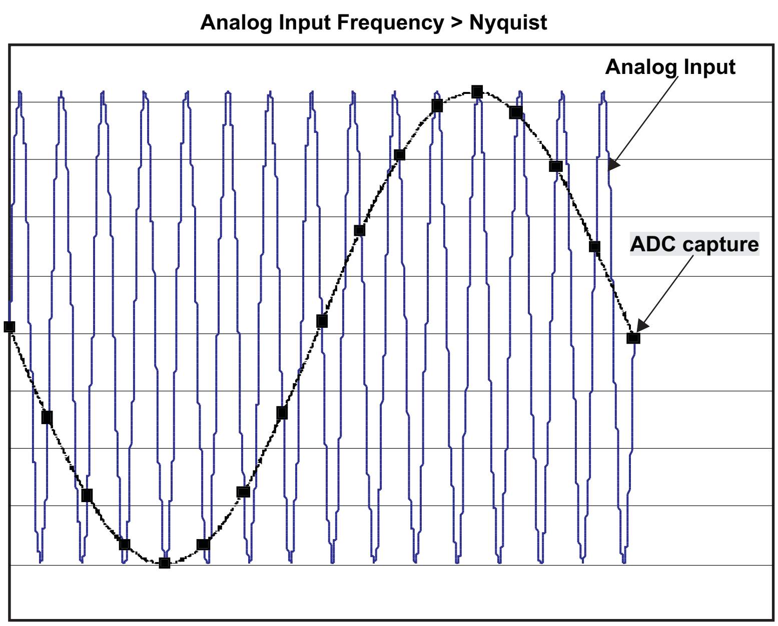


Figure 5. Undersampling Fin > Fs/2

Figure 5 is an example of ADC undersampling (Nyquist < Fin), as less than two samples are captured per period. To state the last sentence in equation form, when Fin > Fs/2, the ADC is undersampling the input waveform. Undersampling allows for an aliased representation of the waveform to be captured.

Compare the ADC capture waveforms in Figure 4 and Figure 5 to understand the effects of the Nyquist frequency (Fs/2), aliasing, oversampling, and undersampling in the time domain. Figure 6 uses the time domain plots in Figure 4 and Figure 5 to describe aliasing, oversampling, and undersampling in the frequency domain.

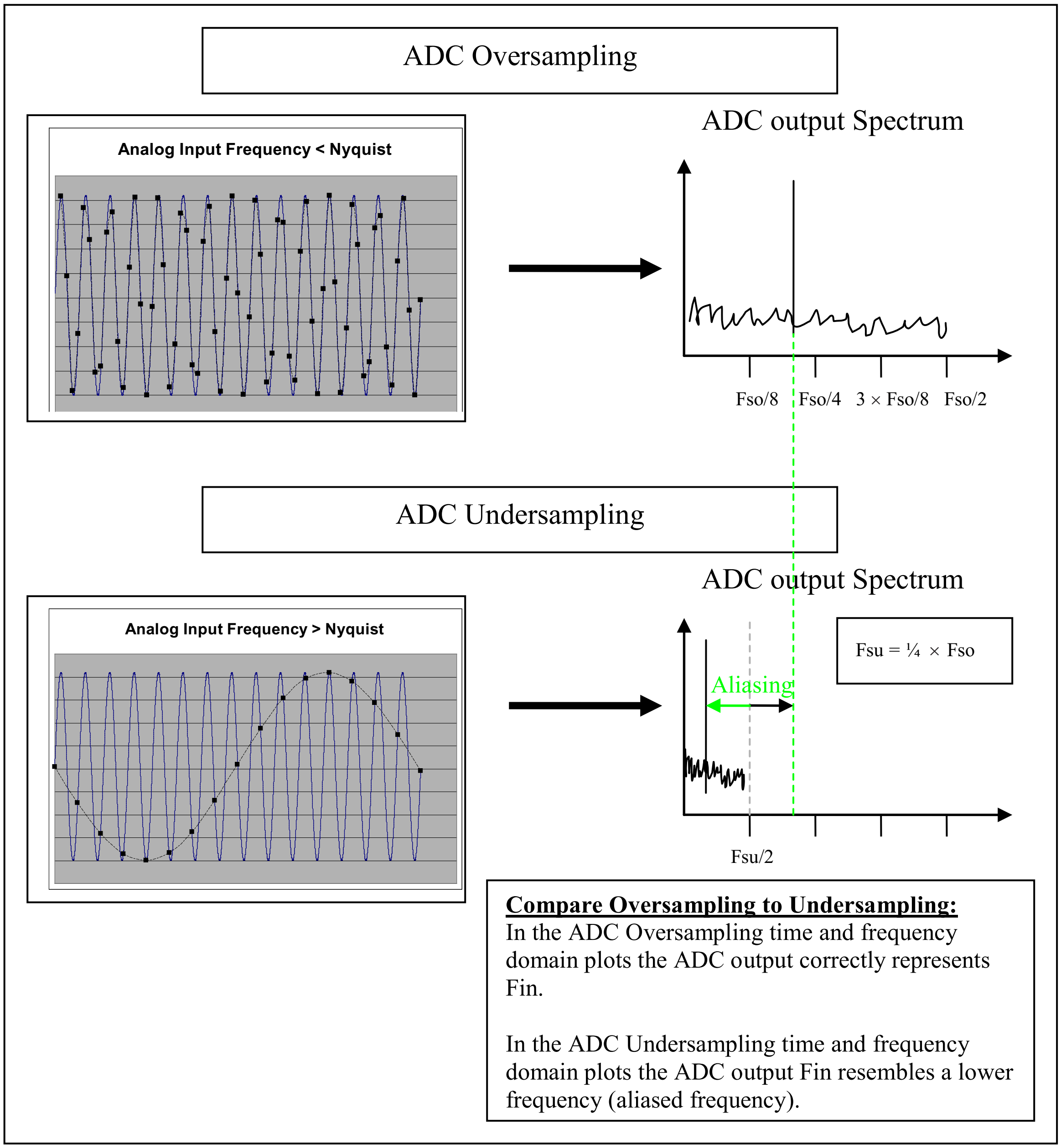


Figure 6. Oversampling and Undersampling in Both Time and Frequency Domains

At first, aliasing may appear to be undesirable. However, it can be very useful. The most useful property is mixing a higher frequency signal to a lower frequency signal. For the system designer, this can translate into cost savings, power savings, or board-space savings by removing the need for an additional mixer in a designer's schematic. To achieve these desirable savings, care must be taken in the frequency planning and ADC selection.

The preceding examples of aliasing were based on single frequency signals. In reality, single frequencies rarely exist in a system. Most systems use wideband signals. A wideband signal definition can simply be stated as many frequencies within a specified bandwidth. Figure 7 shows some examples of desirable and undesirable aliasing with wideband signals. Figure 7a provides the odd-number Nyquist zone example of desirable aliasing example. Figure 7b provides the even-number Nyquist zone example of desirable aliasing example. Even-numbered Nyquist zones alias an inverted frequency response in the ADC's frequency domain output.

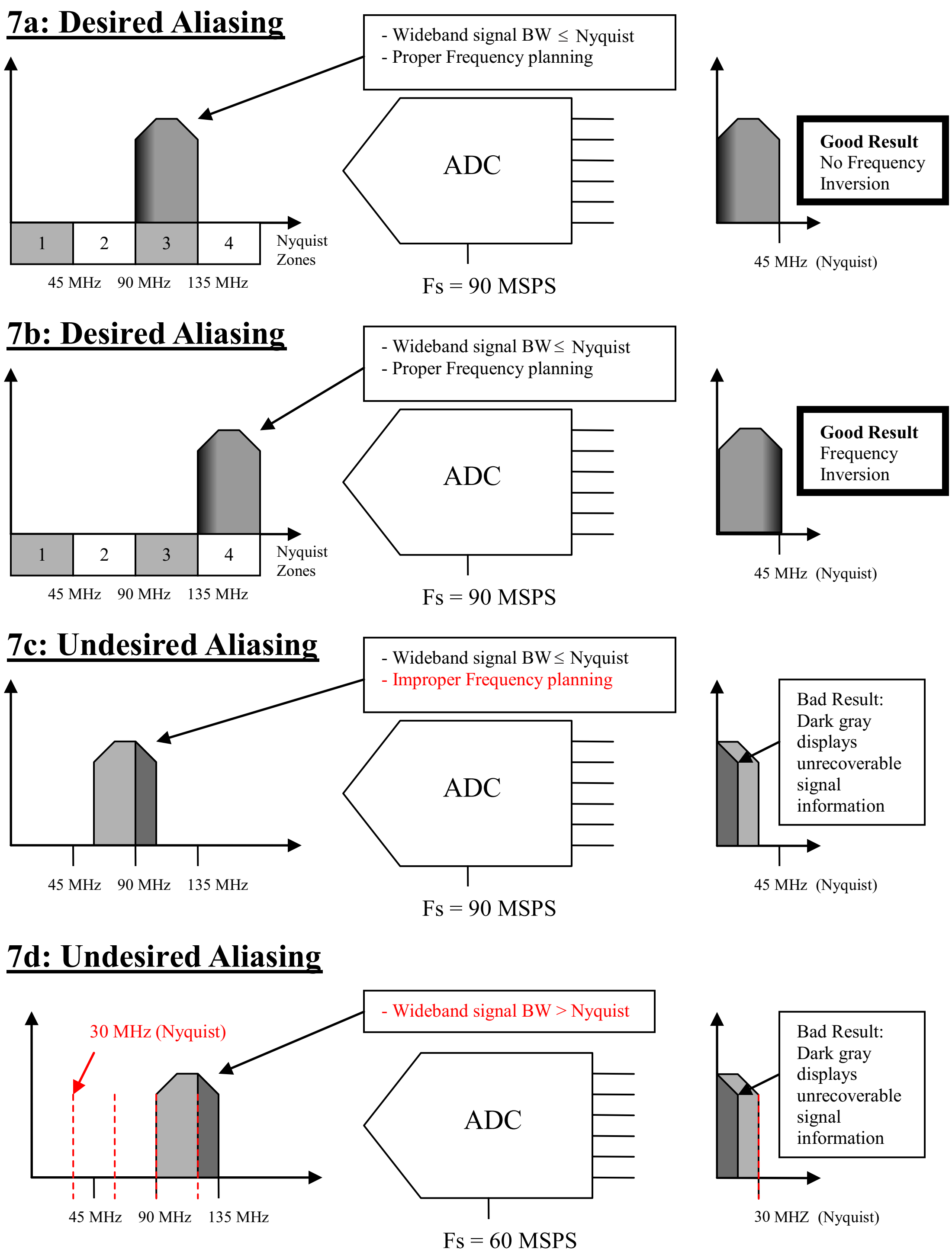


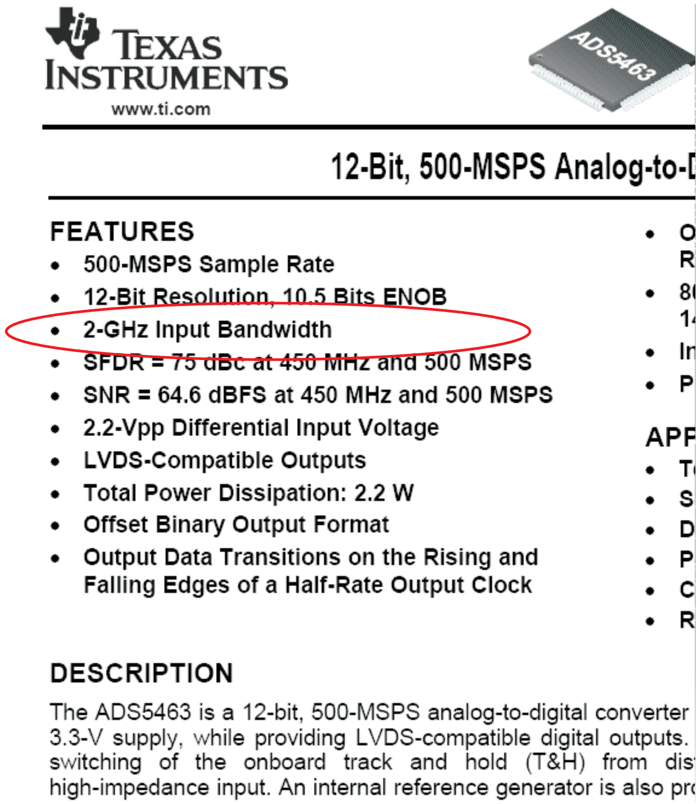
Figure 7. Desired and Undesired Aliasing

To summarize Figure 7, when selecting an ADC it is important to ensure that:

(A) The ADC fits into the designer's desired frequency plan. See Figure 7c

(B) The analog input signal bandwidth is less than the ADC's Nyquist frequency. See Figure 7d

Concerning bandwidth, the other item to consider when selecting an ADC is to ensure that the ADC input bandwidth specification is capable of meeting the customers input frequency requirements. See Figure 8.



www.ti.com

12-Bit, 500-MSPS Analog-to-

FEATURES

- 500-MSPS Sample Rate
- 12-Bit Resolution, 10.5 Bits ENOB
- 2-GHz Input Bandwidth
- SFDR = 75 dBc at 450 MHz and 500 MSPS
- SNR = 64.6 dBFS at 450 MHz and 500 MSPS
- 2.2-Vpp Differential Input Voltage
- LVDS-Compatible Outputs
- Total Power Dissipation: 2.2 W
- Offset Binary Output Format
- Output Data Transitions on the Rising and Falling Edges of a Half-Rate Output Clock

DESCRIPTION

The ADS5463 is a 12-bit, 500-MSPS analog-to-digital converter
3.3-V supply, while providing LVDS-compatible digital outputs.
switching of the onboard track and hold (T&H) from dis
high-impedance input. An internal reference generator is also pr

Figure 8. Analog Input Bandwidth

4 Interfacing With ADC Pins

In general, all ADCs have six pin types.

1. Analog Input
2. Reference/Common Mode
3. Clock Input
4. Digital Output/s
5. Power
6. GND

For high-speed ADCs, the Analog Input, the Reference/Common Mode, and Clock Input are the most critical pin types. These pins have some key interfacing strategies that allow optimum ADC performance. Many of these strategies involve correctly selecting other components that interface with the ADC. These strategies are discussed in the following text.

4.1 Analog Inputs

High-speed ADCs typically have differential analog inputs. The desired signal applied to a differential input is 180 degrees out of phase with respect to each other. This causes the desired signal to add. See Figure 9a.

When compared to a single-ended input, differential signals improve ADC noise performance by cancelling common node noise. Differential signals also reduce even-order harmonics. This is accomplished since the desired signal is shifted by 180 degrees. For the even-order harmonics, this results in 2×180, 4×180, 6×180… (360, 720, 980…) degree shifts. See Figure 9b.

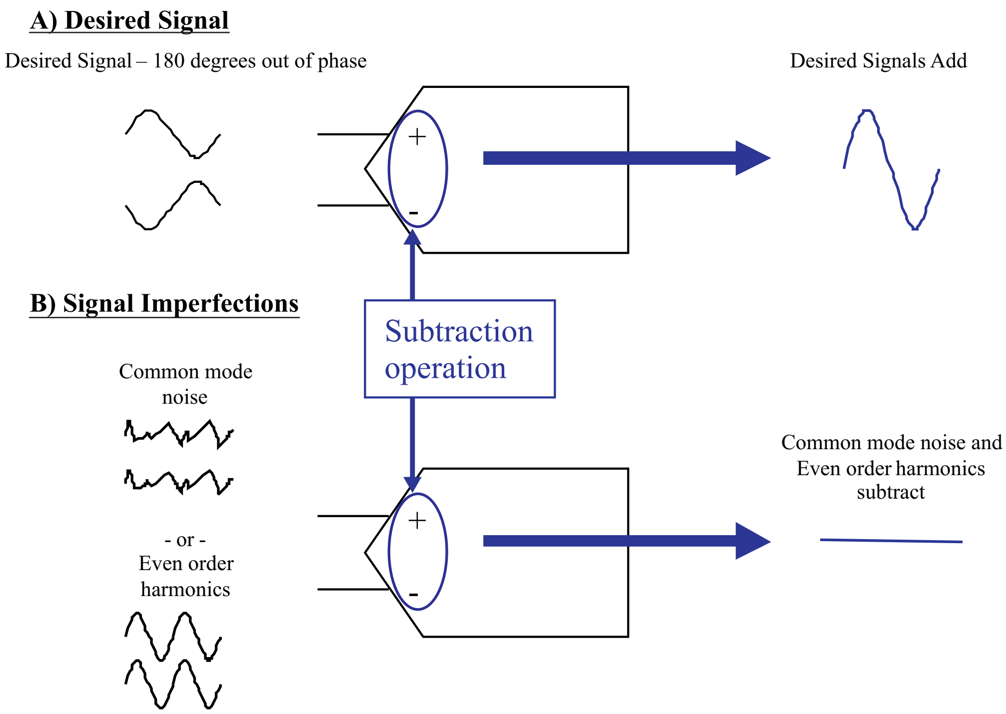


Figure 9. Differential Input Characteristics

When compared to single-ended signals, differential signals are typically superior in harmonic performance as their amplitude is one-half that of an equivalent single-ended signal. These smaller signals allow the differential inputs to operate the ADC and the device driving the ADC with more head room than a single-ended input can. In general, more head room allows the transistors to operate in a more linear device region, thereby reducing nonlinear affects that result in harmonics. These features are shown in Figure 10.

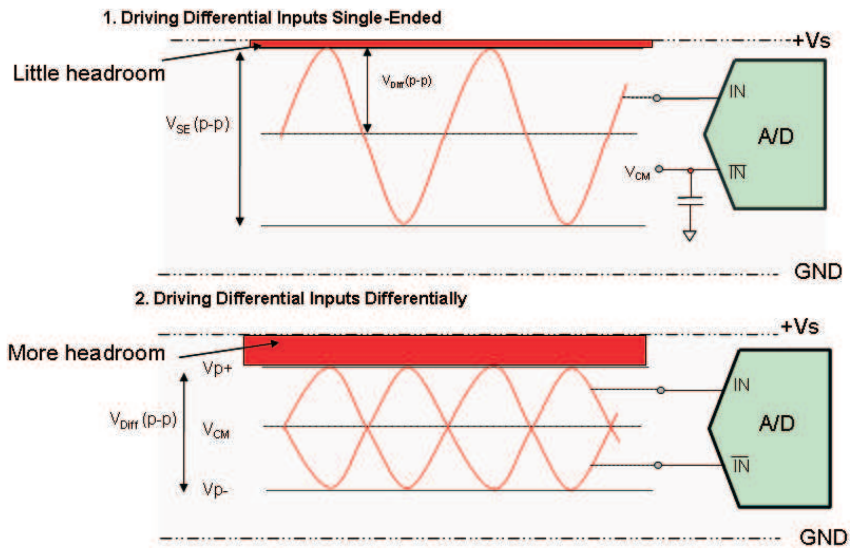
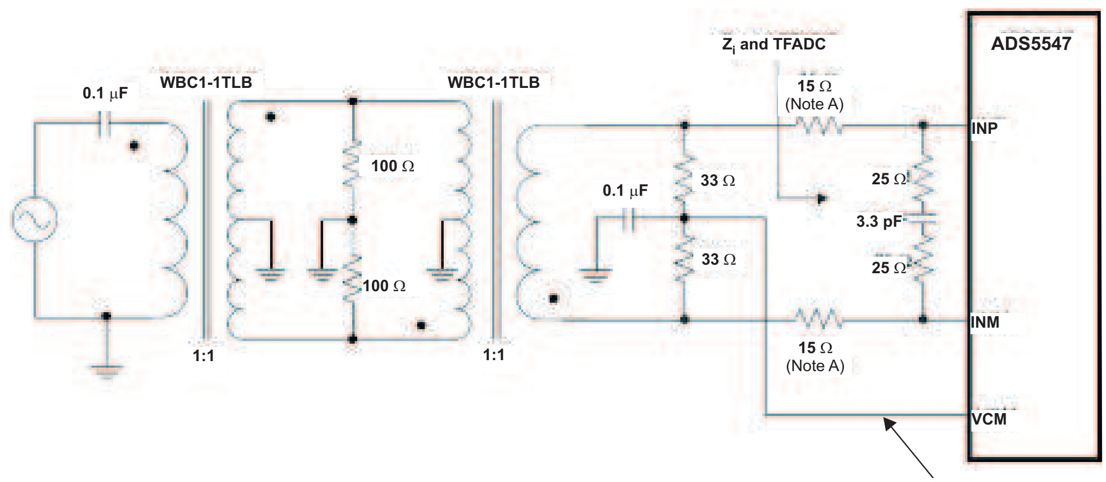


Figure 10. Differential vs Single-Ended Signal Swing

Figure 11 provides an example of the dual-transformer ADC input interface. Transformers are used to convert single-ended signals to differential signals. In this example, it can be noticed that a single transformer (single to differential) has some slight mismatches that result in even-order harmonics. Because the second transformer is differential-to-differential, it corrects this issue and cancels out the even-order harmonics as previously described. Transformers often provide the best performance at higher input frequencies. However, it is preferable to use an operational amplifier to drive the ADC input at baseband frequencies or possibly some lower input frequencies.



A. Use lower series resistance (= 5 Ω to 10 Ω) at high input frequencies (> 100 MHz)

Figure 11. Using Transformers to Drive an ADC Input

4.2 Reference/Common Mode

The reference voltage and common-mode voltage are different functions with respect to each other inside the ADC. In many ADCs, however, these dc voltages are at similar voltage levels, or sometimes an ADC pin may share the reference and common-mode voltage function. For this reason, these signal terminologies are sometimes incorrectly used interchangeably.

The reference voltage determines the dynamic range of the ADC. In data sheets, the relationship between the reference voltage and dynamic range are often provided. See Equation 2 for typical examples. The exact equations for a specific part number can be found in the data sheet.

$$\text{Reference Voltage Sets Dynamic Range}$$

$$\text{Full-scale differential input pp} = \text{Vref} \times 1.33$$

$$\text{or}$$

$$\text{Full-scale differential input pp} = 2 \times \left[\text{Vref}_{\text{Top}} - \text{Vref}_{\text{Bottom}}\right] \quad (2)$$

Reference voltages can be generated internally by the ADC or provided externally to the ADC. It is important to have the correct dc voltages applied to the reference to meet data-sheet performance. It is also important to have a low-noise dc voltage applied to the reference when in external reference mode. Noise on the reference directly impacts the SNR performance of the ADC.

As shown in Figure 11, the common-mode voltage (Vcm) is the dc level applied to the differential analog input signal. Vcm is used to center the differential analog input signal with respect to power and ground. The Vcm can be applied to the differential analog input signal from one of the following sources.

1. Some ADCs have a pin named VCM that provides an internally generated Vcm voltage output. (Figure 11)
2. Some ADCs set Vref equal to the required Vcm level; therefore, Vref can be used to generate the Vcm level.
3. The designer can opt to supply the Vcm level externally.

For options 2 and 3, care must be taken to apply the correct Vcm level as specified by the data sheet. An incorrect level can reduce the dynamic range of the ADC.

4.3 Clock Inputs/Jitter

High-speed ADCs typically have differential clock inputs. Figure 9 in Section 4.1 discusses some of the advantages of differential signals.

From a practical observation, it can be seen that clock jitter and clock slope are the main reason designers can have trouble meeting data-sheet SNR levels. It is critical that a clean clock signal is presented to the high-speed ADC to meet an applications performance requirement. Equation 3 and Figure 12 provide the theoretical explanation for the effects of clock jitter on SNR.

$$\mathrm{SNR} = -20 \times \log_{10}\left(2 \times \mathrm{pi} \times \mathrm{Fin} \times \mathrm{jitter}\right) \tag{3}$$

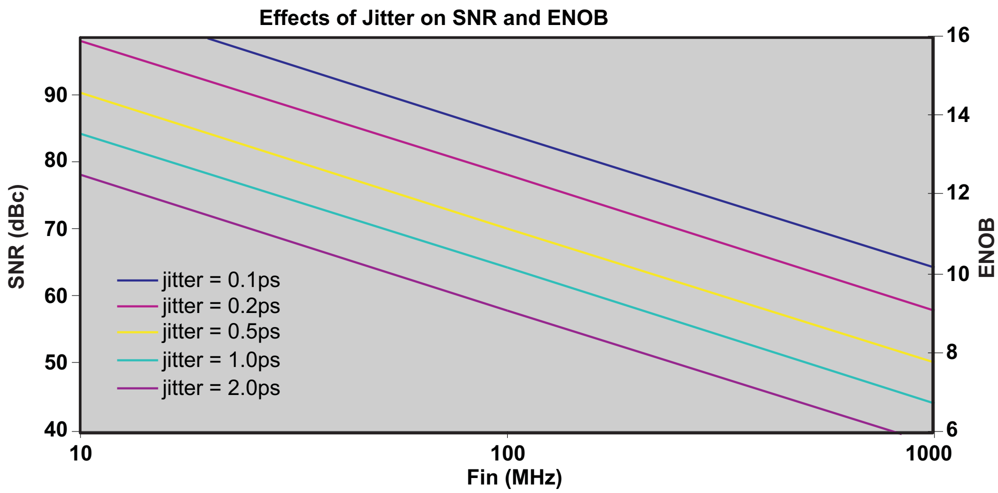


Figure 12. Effects of Jitter on SNR and ENOB

Several key observations can be made about Equation 3 and Figure 12. Observation 1 is that the clock frequency does not affect SNR for an ideal converter. For reasons other than jitter, an actual converter's SNR eventually degrades as the clock frequency approaches the internal ADC's design limits for the digital setup and hold time or the analog inputs settling time. Observation 2 is that for a specified jitter level, SNR degrades as the input frequency increases. Figure 13 visually explains why higher input frequencies (faster slew rates) degrade the SNR for a specified jitter level. Notice that the high-frequency analog input has a larger error with respect to the clock jitter. If this is a random error on the clock signal, it appears as noise in the spectral plots. If this is a deterministic error on the clock signal, then this signal mixes with the ADC input frequency and creates a spur in the spectral plot.

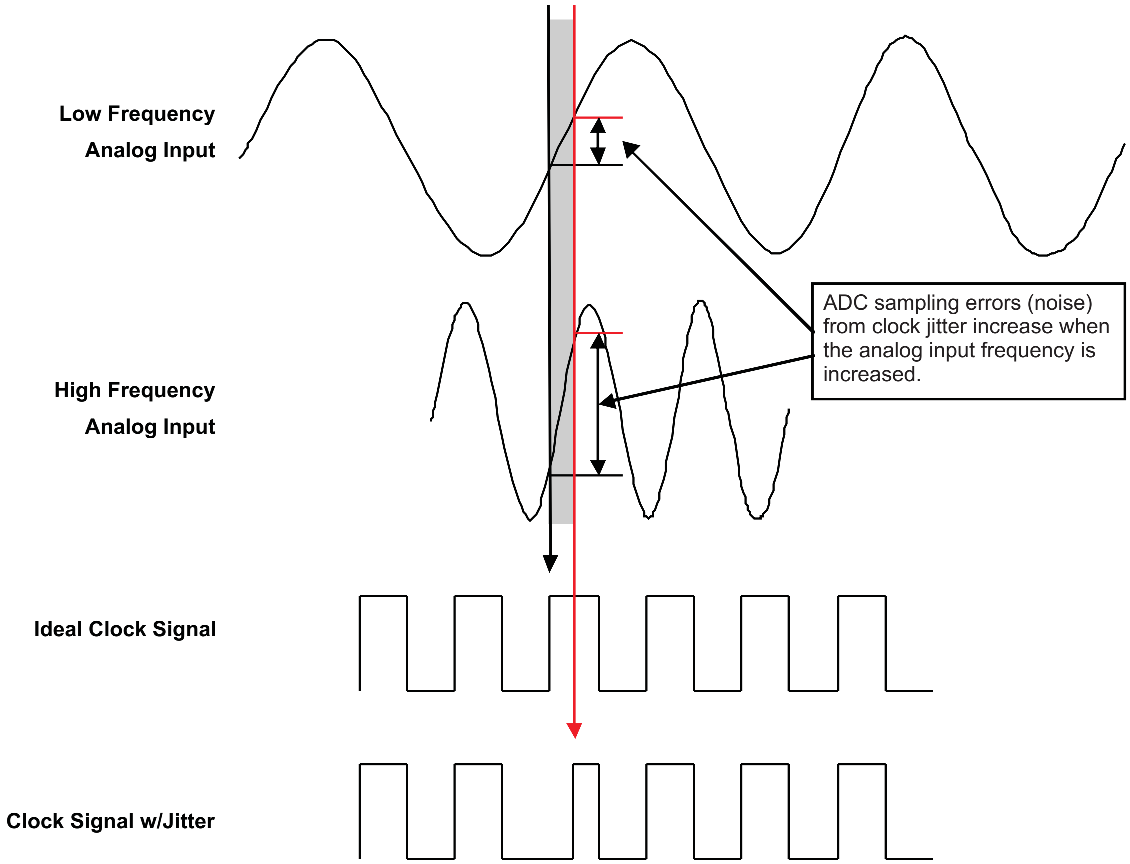


Figure 13. Time Domain Clock Jitter Effects on High- and Low-Frequency Analog Input

The end-user must consider two important components of ADC clock jitter. The first component is the internal clock jitter inherent to the ADC's design. This is commonly known as aperture jitter. Aperture jitter often can be found in the data sheet (see Figure 14). The second component is the external clock jitter that is applied to the ADC. These two components create a total jitter that affects the ADC sampling error. Using Equation 3 and Equation 4, the designer can determine the level of external clock jitter the application requires to meet the desired SNR levels.

$$\text{Jitter}_{\text{Total}}^2 = \text{Jitter}_{\text{Aperture}}^2 + \text{Jitter}_{\text{Ext\_clock}}^2 \quad (4)$$

TIMING CHARACTERISTICS[(1)]

Typical values at T_A = 25°C: minimum and maximum values over full temperature range T_{MIN} = –40°C to T_{MAX} = 85°C, sampling rate = max rated, 50% clock duty cycle, AVDD5 = 5 V, AVDD3 = 3.3 V, DVDD3 = 3.3 V, and 3-V_{PP} differential clock, unless otherwise noted.

	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_a	Aperture delay			200		ps
	Aperture jitter, rms	Internal jitter of the ADC		80		fs
	Latency			5		cycles
t_{CLK}	Clock period	CLK = max rated clock for that part number	1e9/CLK		100	ns
t_{CLKH}	Clock pulse duration, high		0.5e9/CLK		50	ns
t_{CLKL}	Clock pulse duration, low		0.5e9/CLK		50	ns
t_{DRY}	CLK to DRY delay time[(2)]	Zero crossing, 5-pF parasitic to GND	1500	1900	2300	ps
t_{DATA}	CLK to DATA delay time[(2)]		1400	1900	2400	ps
t_{SKEW}	DATA to DRY skew	$t_{DATA} - t_{DRY}$, 5-pF parasitic to GND	–500	0	500	ps
t_{RISE}	DRY/DATA rise time	5-pF parasitic to GND		500		ps
t_{FALL}	DRY/DATA fall time			500		ps

(1) Timing parameters are assured by design or characterization, but not production tested.
(2) DRY and DATA are updated on the rising edge of CLK input. The latency must be added to t_{DATA} to determine the overall propagation delay.

Copyright © 2008–2009, Texas Instruments Incorporated Submit Documentation Feedback 7

Product Folder Link(s): ADS5484 ADS5485

Figure 14. ADC Aperture Jitter From Data Sheet

Example:

A designer has the following requirements and constraints

– ADC SNR of 75 dB
– ADC Fin = 70 MHz
– The customer has selected an ADC with aperture jitter = 80 fs

What is the maximum jitter that the customer's application can tolerate to meet the SNR requirement?

Problem Solving:

A) Using Equation 3, solve for jitter:

1) SNR = $-20 \times \log_{10}(2 \times \text{pi} \times \text{Fin} \times \text{jitter})$
– SNR = 75 dB; provided
– Fin = 70 MHz; provided

2) 75 dB = $-20 \times \log_{10}(2 \times \text{pi} \times 70e6 \times \text{jitter})$
– solve for jitter
– jitter = 405 fs

B) Using Equation 4, solve for $\text{Jitter}_{\text{Ext\_clock}}$:

1) $\text{Jitter}_{\text{Total}}{}^2 = \text{Jitter}_{\text{Aperture2}} + \text{Jitter}_{\text{Ext\_clock2}}$
$\text{Jitter}_{\text{Total}}$ = 405fs; solved for in A2
$\text{Jitter}_{\text{Aperture}}$ = 80fs; provided

2) $(405\text{fs})^2 = (80\text{fs})^2 + \text{Jitter}_{\text{Ext\_clock2}}$
– solve for $\text{Jitter}_{\text{Ext\_clock}}$
– $\text{Jitter}_{\text{Ext\_clock}} < 397$ fs

Answer:

Designer requires an input to the ADC clock to have $\text{Jitter}_{\text{Ext\_clock}} < 397$ fs.

At the start of this section, it was stated that clock jitter and clock slope affect the ADC SNR. So far, only clock jitter has been discussed. Clock slope can also play a critical role, but is a subset of the aperture jitter discussion. Figure 15 demonstrates how a slow clock edge creates a larger aperture jitter internally in the ADC's clock comparator circuit. Figure 15 also shows that for a sine-wave clock, a larger amplitude signal improves the aperture jitter and thereby maximizes the ADC's SNR. Figure 16 provides SNR versus sine-wave amplitude data taken from an actual ADC that illustrates this point.

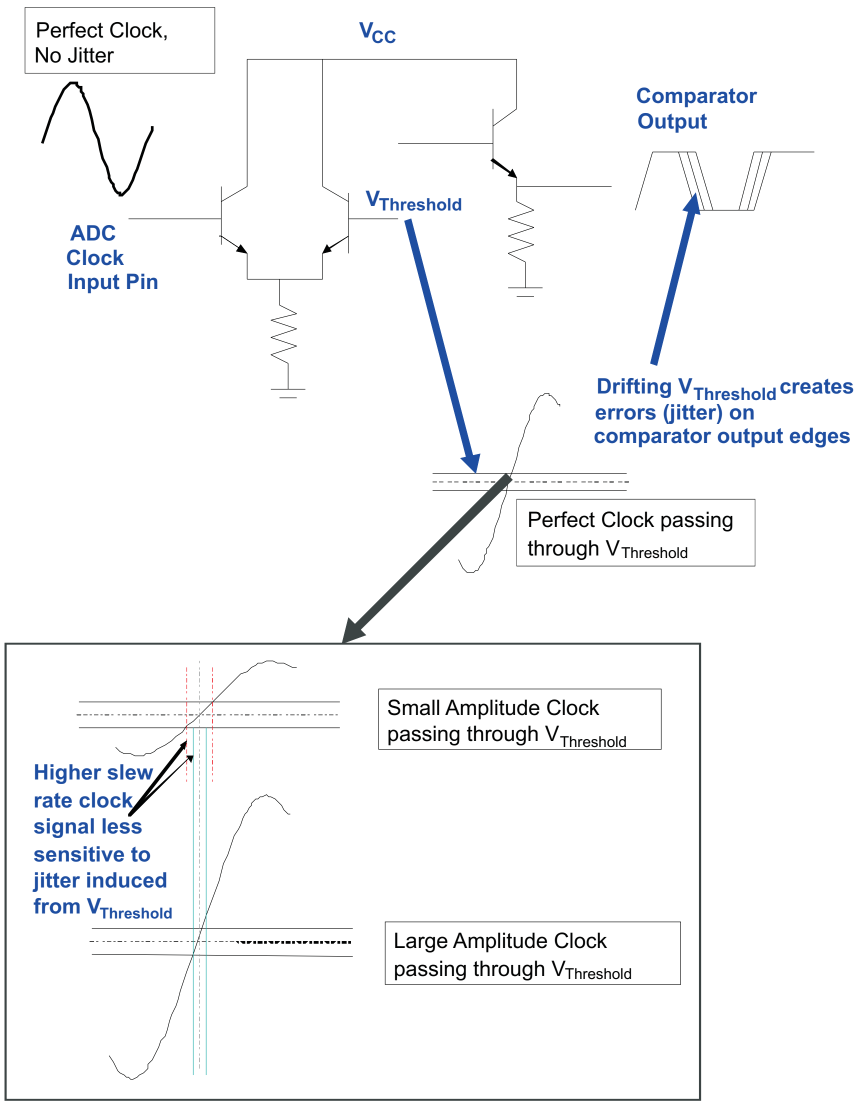


Figure 15. Clock Slope, Aperture Jitter

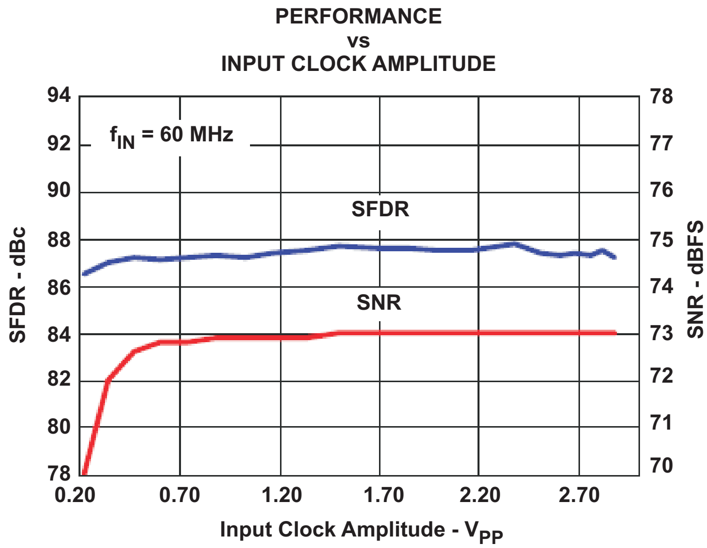


Figure 16. Sine Wave Clock Amplitude vs SNR

One question that arises is, *If the clock slope is of concern, then why not supply the ADC with a square wave clock?* The answer is that square wave clocks also are a viable ADC clock option. The designer must be aware of a couple of trade-offs, however, when considering a sine wave or a square wave clock.

One trade-off is the creating of a low-jitter square wave clock versus the clock frequency range. In many applications, the ADC clock is run through a narrow band saw or crystal filter to improve the close-in phase noise (jitter) of the clock signal. After filtering, the clock is a low-jitter sine wave, which can be applied directly to the ADC clock input. The trade-off of this approach is that the clock frequency range is now limited to the filter range. A few companies have invested in clock jitter cleaner or clock distribution chips. These devices offer improved phase noise performance, square wave outputs, and a wide range of clock frequencies. The phase noise of these devices may be sufficient to meet the required system performance without a clock filter.

The other trade-off with using a square wave clock versus a sine wave clock has to do with signal integrity. Square waves are more harmonically rich than sine waves, and hence, have higher frequency content. It has been argued that this higher frequency content can cause more difficulty during board design due to signal reflections or interference with other signals on the board than a sine wave clock can. Note that regardless of the type of clock applied to the ADC, great care is required during board design to meet the ADC jitter requirements.

4.4 Laboratory Evaluation

The two key aspects in the laboratory evaluation of an ADC are software and hardware. Both of these topics are discussed in this section.

The main software vehicle for laboratory evaluation is the FFT, Fast Fourier Transform. The FFT is a great tool for ADC evaluation due to the speed and accuracy with which it transforms time domain data into frequency domain data. Other transforms perform the same operation, but are slower and/or less accurate when compared to the FFT for ADC evaluation. These other transforms may be preferred in other applications with different requirements than an FFT. Fourier transforms are a very broad and mathematically intensive topic. There are several mathematical variations of FFTs. This document focuses on the input conditions and the output results and not on the mathematics of the operation.

To perform an FFT, one must understand the concepts of coherency, windowing, and spectral leakage. Figure 17 provide a graphical depiction of windowing and spectral leakage. Notice that spectral leakage occurs when the window is not properly defined.

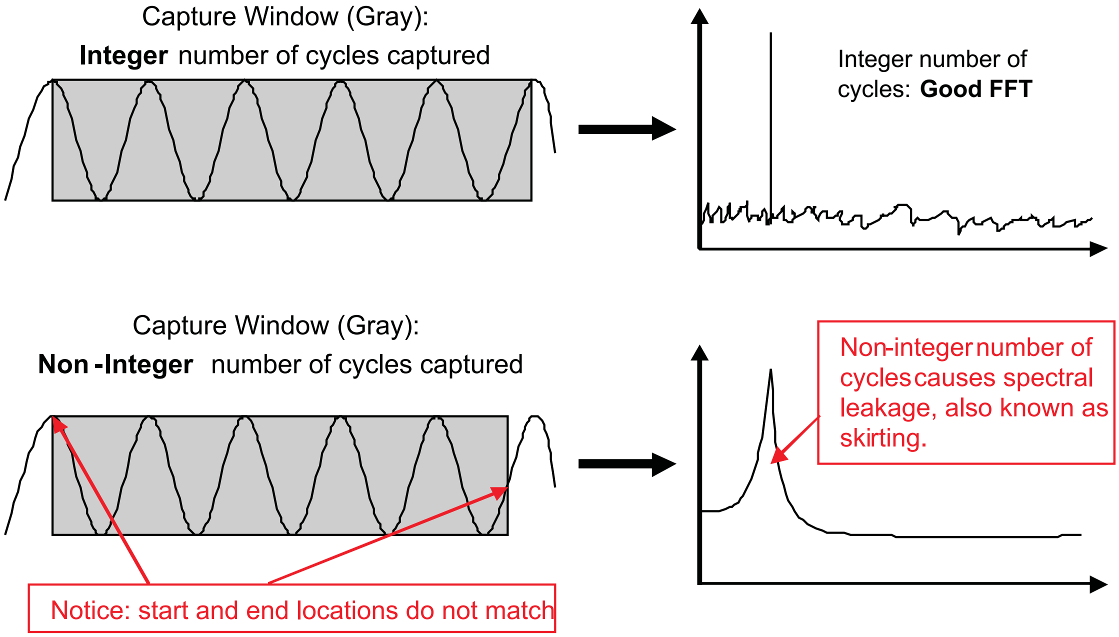


Figure 17. FFT Window and Spectral Leakage

Some designers require a noninteger number of cycles. In these special cases, the FFT cannot be used due to the spectral leakage issue shown in Figure 17. It is recommended to use the Blackman window for Fourier analysis in this case. This method allows for a noninteger number of cycles to be captured, but requires more computation time and has some properties that add a small error to the measurements noise floor and frequency response.

FFT coherency is defined by Equation 5.

$$\begin{aligned} &\mathrm{Fin/Fs} = \mathrm{M/N} \\ &\mathrm{Fin} = \text{input frequency} \\ &\mathrm{Fs} = \text{clock sampling frequency} \\ &\mathrm{M} = \#\text{ of cycles } (\text{odd integer}) \\ &\mathrm{N} = \#\text{ captured samples } (\text{power of } 2) \end{aligned} \quad (5)$$

In addition to Equation 5, observe the following coherency rules when solving this equation.

1. M (number of cycles) is an odd-integer number.
2. N (number of captured samples) is a power of 2
3. M and N are mutually prime numbers
4. Fin and Fs resolutions are greater than the input source minimum resolution specifications.

Rule 1: M (number of cycles) is an odd integer number. The reason for the integer number of samples is to avoid spectral leakage, see Figure 17. The odd-integer number of cycles is a requirement due to Rule 3. The reasons for Rule 3 are discussed in the section for Rule 3.

Rule 2: N (number of captured samples) is a power of 2. This has to do with the mathematical operations of the FFT. The FFT calculations do not work unless this rule is met. Most often, values of 4096, 8192, 16384, 32768, or 65536 are used for N. When selecting N, consider some minor trade-offs between calculation time, measurement repeatability, and the number of unique codes tested for a measurement.

Rule 3: M and N are mutually prime numbers. This rule exists to ensure that a nonrepetitive number of samples are captured. Due to the nature of the FFT calculation, repetitive samples are not useful and only create processing overhead (Figure 18). Because N is a power of 2, making M an odd number ensures that M and N are mutually prime.

Rule 4: Fin and Fs resolutions are greater than the input source minimum resolution specification. This is a practical rule due to hardware limitations. For example, if the analog input and clock source specifications state a minimum resolution of 10 Hz, then they cannot be programmed to < 10-Hz resolution. Therefore, solve Equation 5 with this hardware limitation in mind. If Equation 5 is solved for frequency resolutions less than the input sources, then a noninteger number of cycles are captured and the spectral leakage in Figure 17 is generated to a small degree.

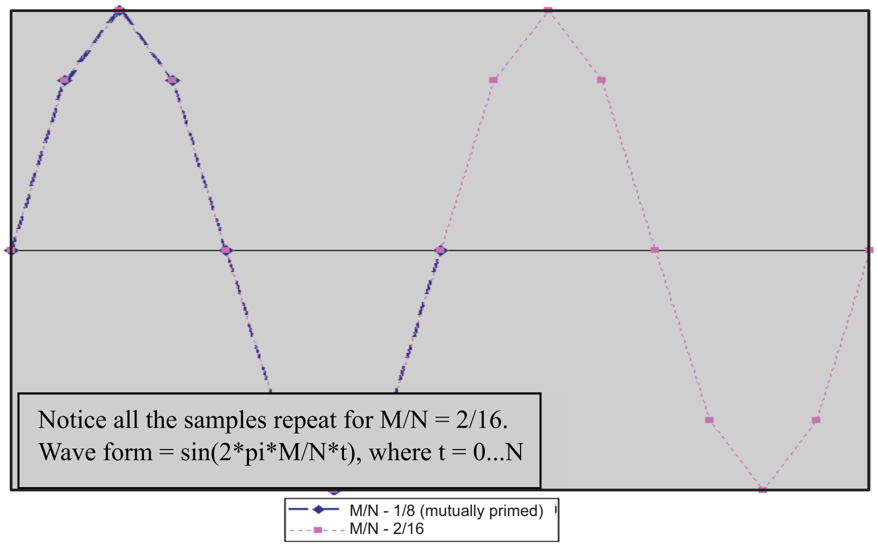


Figure 18. Rule 3 – Mutually Prime

Example:

A designer wishes to evaluate an ADC's performance at

1) Fin = 70 MHz
2) Fs = 125 Msps
3) Both sources have a 1-Hz resolution

What values for M, N, Fin, and Fs can be used for evaluation?

Problem Solving:

1) Use Equation 5: Fin/Fs = M/N
   – Rule 2, select a power of 2 value for N; in this example, use 8192
   – solve for M: M = N × Fin/Fs = 8192 × 70 MHz/125 Msps = 4587.52
   – M = 4587.52 (violates Rule 1 – not an odd integer)
2) Force M to meet Rule 1
   – M = 4587 (odd integer)
3) Solve for a new value of Fs as an integer multiple of N (ensure 1-Hz resolution)
   – X=Fs/N = 125 Msps/8192 = 15258.789
   – Round X to a integer → Xnew = 15258
   – solve for new Fs: Fs = Xnew × N = 15258 × 8192 = 124.993536 Msps
   – Fs = 124.993536 Msps (Rule 4 met for Fs)
4) Solve for new value of Fin using Equation 5e: Fin/Fs = M/N and new value of M, N, and Fs
   – Fin = Fs × M/N = 124.993536 Msps × 4587/8192 = 69.988446 MHz
   – Rule 4 met for Fin

Answer:

Fin = 69988446 Hz
Fs = 124993536 Hz
M = 4587
N = 8192

To meet coherency Fin, Fs is very close to the original requirement. This is an accepted industry standard to have a small variation from original frequency request to meet coherency requirements.

Hardware requirements for evaluating an ADC in the laboratory are:

1. **Clock source:** to achieve the required jitter levels a bandpass filter (BPF) may be required to filter out close-in and wideband noise.
2. **Analog input source:** to achieve the required noise and harmonic levels a bandpass filter (BPF) may be required to filter out noise and harmonics.
3. **Data capture instrument:** ensure that data capture instrument can capture signals at the device speed and store enough samples for FFT process.

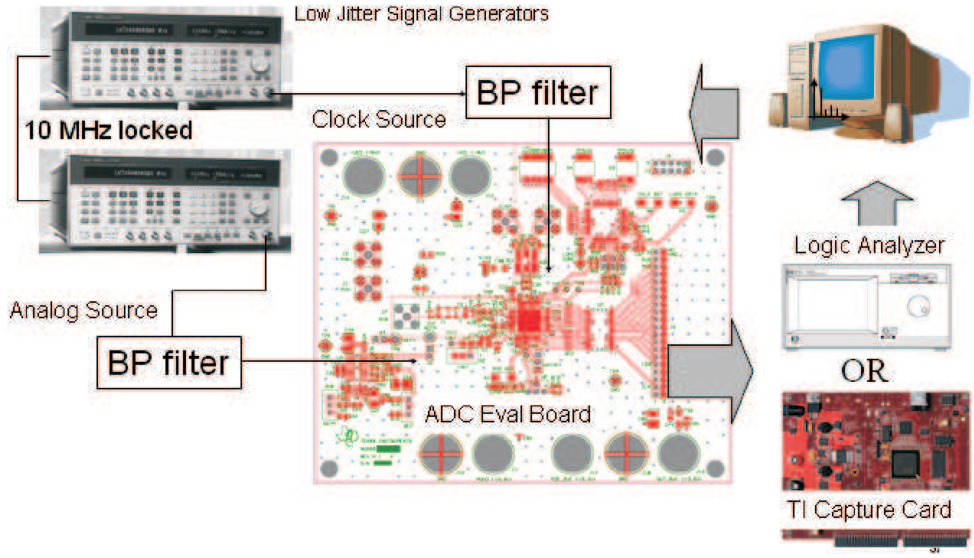


Figure 19. Typical ADC Laboratory Setup

5 Advanced Topic 1: Interleaving ADCs

End-users often push the limits of ADC SNR performance or sampling speeds. Often, when a leading-edge, high-speed data converter enters the market it still does not meet the desired requirements for some applications. Therefore, the topic of interleaving ADCs (also known as ping ponging) comes up for improving SNR or sampling speed. The topic of averaging ADCs is another technique to improve the SNR of a system, and is discussed in the next section.

Figure 20 provides an example of interleaving two ADCs. As in Figure 20, the same analog input is applied to both ADCs, but the sampling clock is delayed half a clock period. As a result, in this example the ADC sampling speed is double.

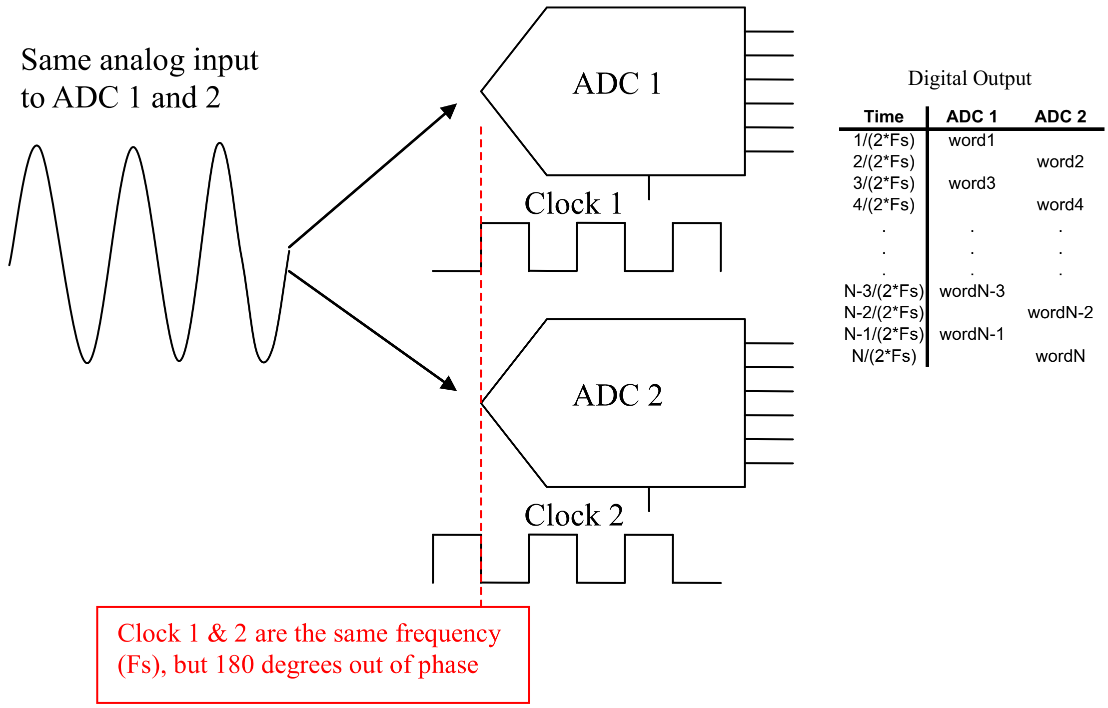


Figure 20. Interleaving ADC Example

Doubling the sampling speed has two benefits. One benefit allows the designer to capture a wider signal bandwidth which can ease frequency planning and/or aliasing filter cost. The second benefit is that interleaving spreads the noise floor across a wider bandwidth, or effectively lowers the noise floor by 3 dB. These benefits are shown in Figure 21.

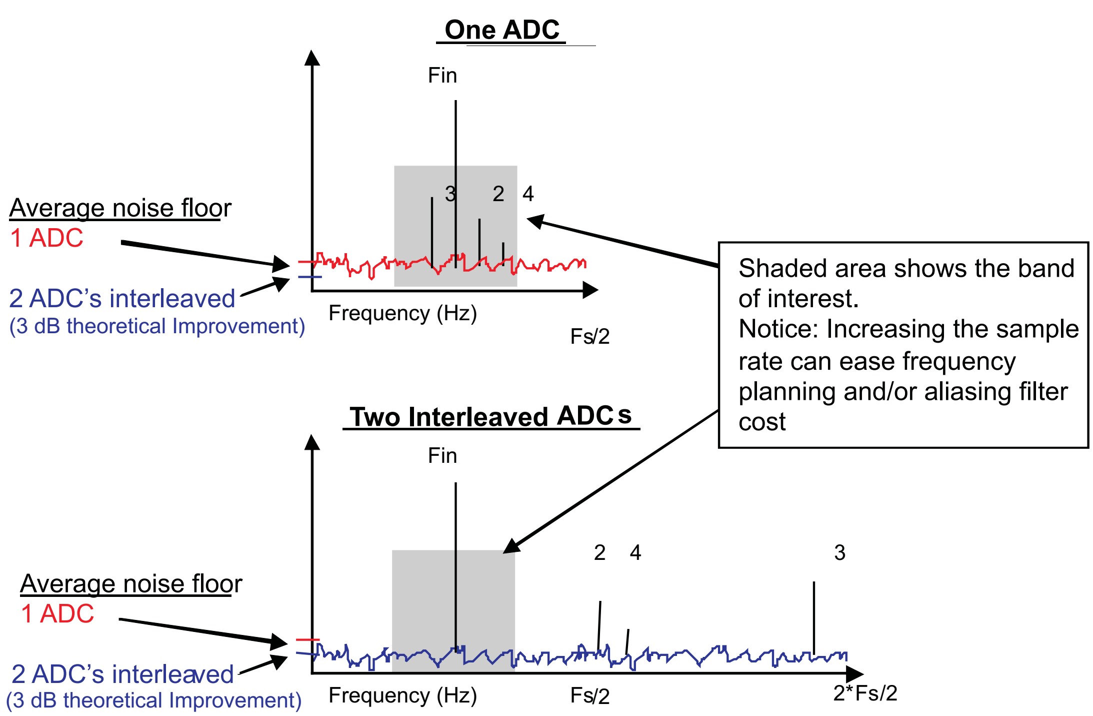


Figure 21. Interleaving Benefits

The noise floor for a given converter is calculated by Equation 6

$$\text{Noise Floor } (-\text{dBHz}) = \text{SNR} + 10 \times \log(\text{Fs}/2)$$

n = number of bits

Fs = clock frequency (6)

The above example discusses interleaving two ADCs, but it is possible to interleave multiple ADCs. When multiple ADCs are interleaved, the noise floor equation can be modified to what is shown in Equation 7.

$$\text{Noise Floor } (-\text{dBHz}) = \text{SNR} + 10 \times \log(x \times \text{Fs}/2)$$

n = number of bits

Fs = clock frequency

x = number of ADC interleaved (7)

Interleaving two or more ADCs does come with its own set of design challenges. DC Offset differences between the interleaved ADCs create a spectral component at specific locations. Gain differences, INL differences, and clock phase errors between the interleaved ADCs create spectral components at clock and analog input signal mixing locations. These error locations are shown in Table 1 and Figure 22.

Table 1. Interleaved ADC Error Spectral Locations

Spectral Error Location for Interleaved ADCs						
Number of ADCs	ADC Offset Error		ADC Gain, Clock Delay, or INL Error			
2	Fs/2		Fs/2–Fin			
3	Fs/3		Fs/3–Fin	Fs/3+Fin		
4	Fs/4	Fs/2	Fs/4–Fin	Fs/4+Fin	2 × Fs/4–Fin	
5	Fs/5	2 × Fs/5	Fs/5–Fin	Fs/5+Fin	2 × Fs/5–Fin	2 × Fs/5+Fin

The good news is that all these errors show up in known locations in the spectrum. However, the bad news is frequency planning becomes increasingly difficult with these errors and the magnitude of these errors varies with temperature. Figure 22 shows FFT plots of 2, 3, 4, and 5 interleaved ADCs. Figure 22 plots were simulated with an ideal 14-bit ADC with forced offset errors <15 LSB and gain errors <0.3%. Even these small errors can create a significant spurious response.

As a side note, Figure 22 shows that errors generated in an interleaved system using a number of ADCs equal to a power of 2 generates coherent frequency errors in an FFT. Conversely, Figure 22 also shows that errors generated in an interleaved system using a number of ADCs equal to a nonpower of 2 generate noncoherent frequency errors in an FFT. This information may benefit the designer if FFTs are used in the final application or in calibrating the interleaved ADCs.

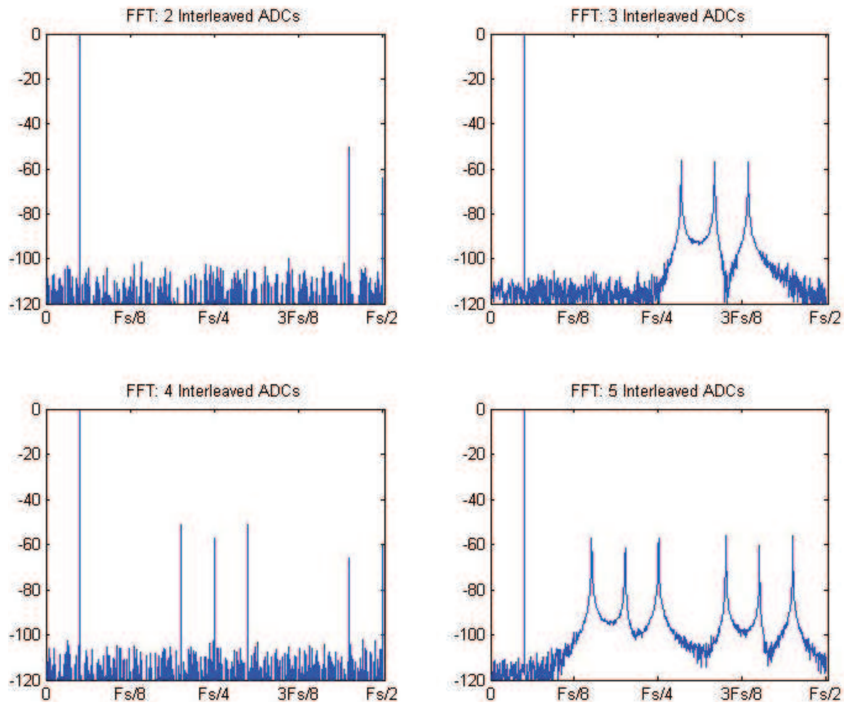


Figure 22. Interleaved ADC Spectral Plots

Most likely, the designer needs to perform some sort of analog or digital calibration across temperature to reduce the errors at these spectral locations. Depending on the type of calibration required, the calibration circuitry or DSP algorithms required can add a significant amount of effort to the design process.

Practically speaking, the interleaved ADC system will suffer some performance penalty. The magnitude of degradation depends on the analog matching and/or the amount of digital signal processing that the designer can afford.

6 Advanced Topic 2: Averaging ADCs

End-users often push the limits of ADC SNR performance. Often, when a leading-edge, high-speed data converter enters the market it still does not meet the desired requirements for some applications.. Therefore, the topic of averaging ADCs arises to improve SNR.

Figure 23 provides an example of averaging two ADCs. As shown Figure 23, the same analog input is applied to both ADCs, and also the same clock frequency and phase are applied to both ADCs. The designer then averages the two ADC output words at each clock cycle. As a result, the SNR performance is improved by 3 dB as shown in Figure 24.

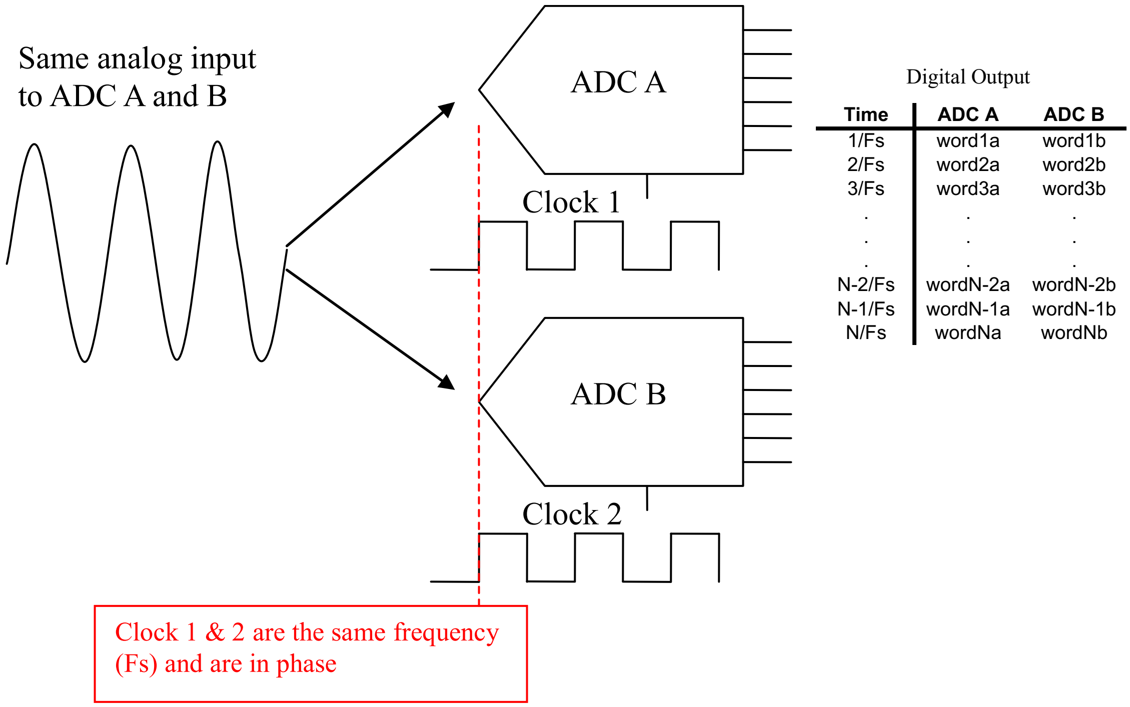


Figure 23. Averaging Two ADCs Example

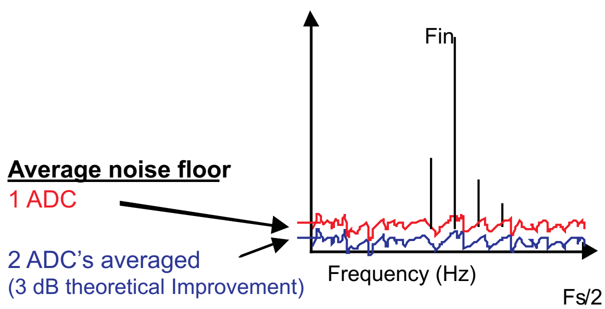


Figure 24. Averaging Two ADCs Benefits

This averaging technique reduces uncorrelated noise sources between the ADCs. Examples of uncorrelated noise sources in an ADC are thermal noise, internal ADC reference noise, or nondeterministic aperture clock jitter (internal ADC clock jitter). Conversely, the averaging technique does not reduce correlated noise sources between the ADCs. Examples of correlated noise sources are distortions inherent to the ADC design or common errors in the analog and clock inputs signals applied externally to the ADC.

To calculate the theoretical SNR of averaging multiple ADCs, see Equation 8. Figure 25 shows the plots resulting from the use of Equation 8 when all ADCs have the same SNR. Figure 25 hints at the practical limitation of diminishing returns, where averaging four ADCs improves the system SNR by 6 dB, but improving the system by 20 dB requires 100 ADCs.

ADC Averaging Equation

$$SNR_{ADC\_AVE} = -10 \times \log\left[\frac{10^{\frac{-SNR_{ADC1}}{10}} + 10^{\frac{-SNR_{ADC2}}{10}} ...+...10^{\frac{-SNR_{ADCN}}{10}}}{N^2}\right]$$

If $SNR_{ADC1} = SNR_{ADC2}... = SNR_{ADCN,}$ then this equation simplifies to

$$SNR_{ADC\_AVE} = SNR_{ADC} + 10 \times \log N$$

N = Number of ADCs (8)

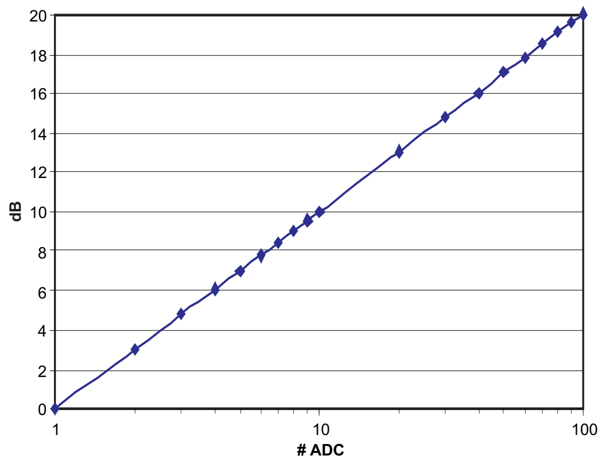


Figure 25. SNR Averaging Improvement vs Number of ADCs

To aid the designer in determining clock jitter requirements, the following is provided. As previously mentioned, aperture clock jitter is an uncorrelated noise source. Assuming all ADCs have a similar and random aperture clock jitter, then use Equation 9 to calculate the maximum external clock jitter that the system can tolerate. See Section 4.3 to relate ADC SNR to the total system jitter.

Jitter calculation for averaging ADCs

$$Jitter_{Total}^2 = Jitter_{Ext\_clock}^2 + \left[\frac{Jitter_{Aperture}}{\sqrt{N}}\right]^2$$

N = Number of ADCs (9)

7 Advanced Topic 3: Dithering

An ADC has deterministic and systematic errors that repeat each time those codes are exercised. In theory, these errors can be minimized by adding a low-level, random noise. This process of adding a low-level, random noise to improve the ADC's distortion beyond its inherent linearity is known as dithering (see Figure 26).

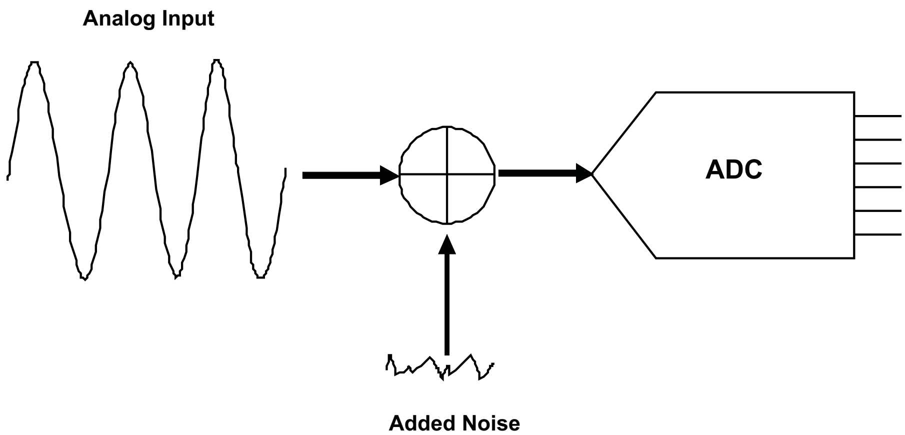
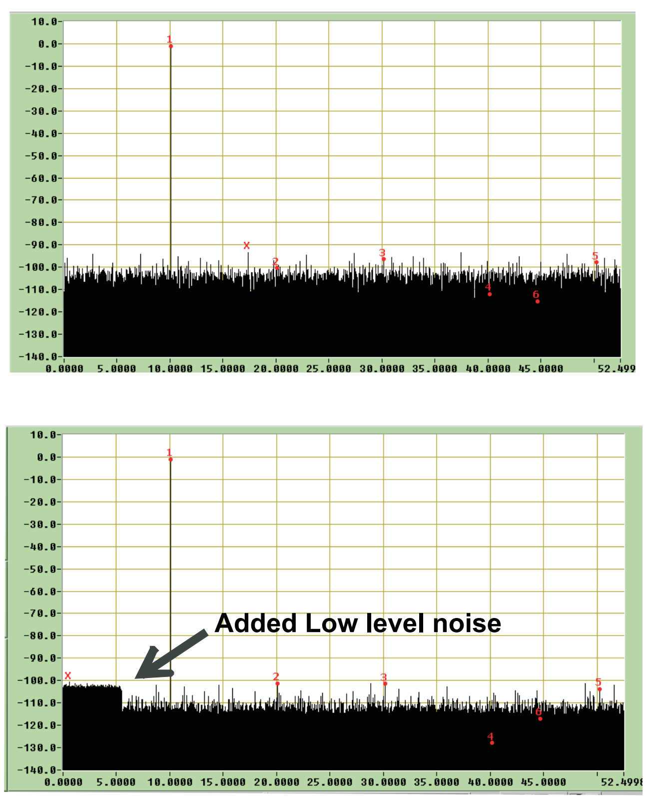


Figure 26. Dithering Diagram

Some of the key points of dithering are:

- Dither can reduce the level of harmonic content, but it can have the adverse affect of increasing the overall noise floor as shown in Figure 27.
- Harmonic improvement can vary with signal type and amplitude. In some cases, little to no improvement is seen.
- Some dither techniques add the noise in areas of the circuit that need to be randomized and then attempt to subtract the noise later so that the degradation in SNR is minimized or not even noticed by the designer.
- Dither also can be applied to the input signal external to the ADC by the designer, or some ADCs have a dithering option internal to the ADC.
- In some cases, real-world signals contain enough noise to behave as dither.



No Dither
SFDR = 92.7dBc
THD = 92.1dBc
SNR = 67.6dB

Dither
SFDR = 99.6dBc
THD = 96.0dBc
SNR = 65.7dB

Figure 27. Example of Affects of Dithering

The designer must determine if dithering is required. Dithering is a complex topic and must be understood thoroughly before implementing. Applying dithering can be difficult due to the nature of distortion behavior varying with gain and signal type. For these reasons, designing with an ADC that meets performance targets without dithering is a more straightforward process than implementing dithering to raise the ADC performance to the required level.