INTRODUCTION

You don't have to deal with ADCs or DACs for long before running across this often quoted formula for the theoretical signal-to-noise ratio (SNR) of a converter. Rather than blindly accepting it on face value, a fundamental knowledge of its origin is important, because the formula encompasses some subtleties which if not understood can lead to significant misinterpretation of both data sheet specifications and converter performance.

Remember that this formula represents the theoretical performance of a perfect N-bit ADC. You can compare the actual ADC SNR with the theoretical SNR and get an idea of how the ADC stacks up.

This tutorial first derives the theoretical quantization noise of an N-bit analog-to-digital converter (ADC). Once the rms quantization noise voltage is known, the theoretical signal-to-noise ratio (SNR) is computed. The effects of oversampling on the SNR are also analyzed.

DERIVATION

The maximum error an ideal converter makes when digitizing a signal is $\pm \frac{1}{2} \text{LSB}$ as shown in the transfer function of an ideal N-bit ADC (Figure 1). The quantization error for any ac signal which spans more than a few LSBs can be approximated by an uncorrelated sawtooth waveform having a peak-to-peak amplitude of $q$, the weight of an LSB. Another way to view this approximation is that the actual quantization error is equally probable to occur at any point within the range $\pm \frac{1}{2} q$. Although this analysis is not precise, it is accurate enough for most applications.

W. R. Bennett of Bell Laboratories analyzed the actual spectrum of quantization noise in his classic 1948 paper (Reference 1). With the simplifying assumptions previously mentioned, his detailed mathematical analysis simplifies to that of Figure 1. Other significant papers and books on converter noise followed Bennett's classic publication (References 2-6).
The quantization error as a function of time is shown in more detail in Figure 2. Again, a simple sawtooth waveform provides a sufficiently accurate model for analysis. The equation of the sawtooth error is given by

\[
e(t) = st, \quad -q/2s < t < +q/2s.
\]  

Eq. 1

The mean-square value of \(e(t)\) can be written:

\[
e^2(t) = \frac{q^2}{12}.
\]

Eq. 2

Performing the simple integration and simplifying,

\[
e^2(t) = \frac{q^2}{12}.
\]

Eq. 3

The root-mean-square quantization error is therefore

\[
\text{rms quantization noise} = \sqrt{e^2(t)} = \frac{q}{\sqrt{12}}.
\]

Eq. 4

The sawtooth error waveform produces harmonics which extend well past the Nyquist bandwidth of dc to \(f_s/2\). However, all these higher order harmonics must fold (alias) back into the Nyquist bandwidth and sum together to produce an rms noise equal to \(q/\sqrt{12}\).

As Bennett points out (Reference 1), the quantization noise is approximately Gaussian and spread more or less uniformly over the Nyquist bandwidth dc to \(f_s/2\). The underlying assumption here is that the quantization noise is uncorrelated to the input signal. Under certain conditions where the sampling clock and the signal are harmonically related, the quantization noise becomes correlated, and the energy is concentrated in the harmonics of the signal – however, the rms value remains approximately \(q/\sqrt{12}\). The theoretical signal-to-noise ratio can now be calculated assuming a full-scale input sinewave:

\[
\text{Input FS Sinewave} = v(t) = \frac{q2^N}{2} \sin(2\pi ft).
\]

Eq. 5

The rms signal of the input signal is therefore

\[
\text{rms value of FS input} = \frac{q2^N}{2\sqrt{2}}.
\]

Eq. 6

The rms signal-to-noise ratio for an ideal N-bit converter is therefore
Bennett's paper shows that although the actual spectrum of the quantization noise is quite complex to analyze, the simplified
analysis which leads to Eq. 9 is accurate enough for most purposes. However, it is important to emphasize again that the rms
quantization noise is measured over the full Nyquist bandwidth, dc to $f_s/2$.

**OVERSAMPLING AND UNDERSAMPLING**

In many applications, the actual signal of interest occupies a smaller bandwidth, BW, which is less than the Nyquist bandwidth
(see Figure 3). If digital filtering is used to filter out noise components outside the bandwidth BW, then a correction factor (called
process gain) must be included in the equation to account for the resulting increase in SNR as shown in Eq. 10.

$$\text{SNR} = 6.02N + 1.76\text{dB} + 10\log_{10}\left(\frac{f_s}{2 \cdot \text{BW}}\right), \quad \text{over the bandwidth BW.} \quad \text{Eq. 10}$$

The process of sampling a signal at a rate which is greater than twice its bandwidth is referred to as oversampling. Oversampling
in conjunction with quantization noise shaping and digital filtering are the key concepts in sigma-delta converters, although
oversampling can be used with any ADC architecture.

The significance of process gain can be seen from the following example. In many digital basestations or other wideband
receivers the signal bandwidth is composed of many individual channels, and a single ADC is used to digitize the entire
bandwidth. For instance, the analog cellular radio system (AMPS) in the U.S. consists of 416 30-kHz wide channels, occupying a
bandwidth of approximately 12.5 MHz. Assume a 65-MSPS sampling frequency, and that digital filtering is used to separate the
individual 30-kHz channels. The process gain due to oversampling for these conditions is given by:

$$\text{SNR} = 20\log_{10}\frac{\text{rms value of FS input}}{\text{rms value of quantization noise}}$$

$$\text{SNR} = 20\log_{10}\left(\frac{2^N/\sqrt{2}}{\sqrt{12}}\right) = 20\log_{10}2^N + 20\log_{10}\frac{\sqrt{3}}{2}$$

$$\text{SNR} = 6.02N + 1.76\text{dB}, \quad \text{over the dc to } f_s/2 \text{ bandwidth.} \quad \text{Eq. 9}$$
The process gain is added to the ADC SNR specification to yield the SNR in the 30-kHz bandwidth. In the above example, if the ADC SNR specification is 65 dB (dc to \( f_s/2 \)), then it is increased to 95.3 dB in the 30-kHz channel bandwidth (after appropriate digital filtering).

Figure 4 shows an application which combines oversampling and undersampling. The signal of interest has a bandwidth \( BW \) and is centered around a carrier frequency \( f_c \). The sampling frequency can be much less than \( f_c \) and is chosen such that the signal of interest is centered in its Nyquist zone. Analog and digital filtering removes the noise outside the signal bandwidth of interest, and therefore results in process gain per Eq. 10.

\[
\text{Process Gain} = 10 \log_{10} \frac{f_s}{2 \cdot \text{BW}} = 10 \log_{10} \frac{65 \times 10^6}{2 \times 30 \times 10^3} = 30.3 \text{ dB.} \tag{11}
\]

Figures 5 shows an application which combines oversampling and undersampling. The signal of interest has a bandwidth \( BW \) and is centered around a carrier frequency \( f_c \). The sampling frequency can be much less than \( f_c \) and is chosen such that the signal of interest is centered in its Nyquist zone. Analog and digital filtering removes the noise outside the signal bandwidth of interest, and therefore results in process gain per Eq. 10.

\[
\text{SNR} = 6.02N + 1.76 \text{dB} + 10 \log_{10} \left( \frac{f_s}{2 \cdot \text{BW}} \right) \quad \text{FOR FS SINEWAVE}
\]

**Figure 4: Undersampling and Oversampling Combined Results in Process Gain**

**CORRELATION BETWEEN QUANTIZATION NOISE AND INPUT SIGNAL YIELDS MISLEADING RESULTS**

Although the rms value of the noise is accurately approximated by \( q/\sqrt{12} \), its frequency domain content may be highly correlated to the ac-input signal under certain conditions. For instance, there is greater correlation for low amplitude periodic signals than for large amplitude random signals. Quite often, the assumption is made that the theoretical quantization noise appears as white noise, spread uniformly over the Nyquist bandwidth dc to \( f_s/2 \). Unfortunately, this is not true in all cases. In the case of strong correlation, the quantization noise appears concentrated at the various harmonics of the input signal, just where you don’t want them.

In most practical applications, the input to the ADC is a band of frequencies (always summed with some unavoidable system noise), so the quantization noise tends to be random. In spectral analysis applications (or in performing FFTs on ADCs using spectrally pure sinewaves as inputs, however, the correlation between the quantization noise and the signal depends upon the ratio of the sampling frequency to the input signal.

This is demonstrated in Figure 5, where the output of an ideal 12-bit ADC is analyzed using a 4096-point FFT. In the left-hand FFT plot (A), the ratio of the sampling frequency (80,000 MSPS) to the input frequency (2,000 MHz) was chosen to be exactly 40, and the worst harmonic is about 77 dB below the fundamental. The right hand diagram (B) shows the effects of slightly offsetting the input frequency to 2.111 MHz, showing a relatively random noise spectrum, where the SFDR is now about 93 dBC.
and is limited by the spikes in the noise floor of the FFT. In both cases, the rms value of all the noise components is approximately $q/\sqrt{12}$ (yielding a theoretical SNR of 74 dB) but in the first case, the noise is concentrated at harmonics of the fundamental because of the correlation.

![Figure 5: Effect of Ratio of Sampling Clock to Input Frequency on Quantization Noise Frequency Spectrum for Ideal 12-bit ADC, 4096-Point FFT. (A) Correlated Noise, (B) Uncorrelated Noise](image)

Note that this variation in the apparent harmonic distortion of the ADC is an artifact of the sampling process caused by the correlation of the quantization error with the input frequency. In a practical ADC application, the quantization error generally appears as random noise because of the random nature of the wideband input signal and the additional fact that there is a usually a small amount of system noise which acts as a *dither* signal to further randomize the quantization error spectrum.

It is important to understand the above point, because single-tone sinewave FFT testing of ADCs is one of the universally accepted methods of performance evaluation. In order to accurately measure the harmonic distortion of an ADC, steps must be taken to ensure that the test setup truly measures the ADC distortion, not the artifacts due to quantization noise correlation. This is done by properly choosing the frequency ratio and sometimes by summing a small amount of noise (dither) with the input signal. The exact same precautions apply to measuring DAC distortion with an analog spectrum analyzer.

**THEORETICAL NOISE FLOOR OF THE FFT OUTPUT**

Figure 6 shows the FFT output for an ideal 12-bit ADC. Note that the average value of the noise floor of the FFT is approximately 107 dB below full-scale, but the theoretical SNR of a 12-bit ADC is 74 dB. The FFT noise floor is *not* the SNR of the ADC, because the FFT acts like an analog spectrum analyzer with a bandwidth of $f_s/M$, where $M$ is the number of points in the FFT. The theoretical FFT noise floor is therefore $10\log_{10}(M/2)$ dB below the quantization noise floor due to the *processing gain* of the FFT.

In the case of an ideal 12-bit ADC with an SNR of 74 dB, a 4096-point FFT would result in a processing gain of $10\log_{10}(4096/2) = 33$ dB, thereby resulting in an overall FFT noise floor of $74 + 33 = 107$ dBc. In fact, the FFT noise floor can be reduced even further by going to larger and larger FFTs; just as an analog spectrum analyzer's noise floor can be reduced by narrowing the bandwidth. When testing ADCs using FFTs, it is therefore important to ensure that the FFT size is large enough so that the distortion products can be distinguished from the FFT noise floor itself. Averaging a number of FFTs does not further reduce the noise floor, it simply reduces the variations between the individual noise spectral component amplitudes.
Figure 6: Noise Floor for an Ideal 12-bit ADC Using 4096-point FFT

REFERENCES

INTRODUCTION

A quick reading of Harry Nyquist's classic Bell System Technical Journal article of 1924 (Reference 1) does not reveal the true significance of the criterion which bears his name. Nyquist was working on the transmission of telegraph signals over a channel that was bandwidth limited. A thorough understanding of the modern interpretation of Nyquist's criterion is mandatory when dealing with sampled data systems. This tutorial explains in easy to understand terms how the Nyquist criterion applies to baseband sampling, undersampling, and oversampling applications.

A block diagram of a typical real-time sampled data system is shown in Figure 1. Prior to the actual analog-to-digital conversion, the analog signal usually passes through some sort of signal conditioning circuitry which performs such functions as amplification, attenuation, and filtering. The lowpass/bandpass filter is required to remove unwanted signals outside the bandwidth of interest and prevent aliasing.

![Figure 1: Typical Sampled Data System](image)

The system shown in Figure 1 is a real-time system, i.e., the signal to the ADC is continuously sampled at a rate equal to $f_s$, and the ADC presents a new sample to the DSP at this rate. In order to maintain real-time operation, the DSP must perform all its required computation within the sampling interval, $1/f_s$, and present an output sample to the DAC before arrival of the next sample from the ADC. An example of a typical DSP function would be a digital filter.

Note that the DAC is required only if the DSP data must be converted back into an analog signal (as would be the case in a voiceband or audio application, for example). There are many applications where the signal remains entirely in digital format after the initial A/D conversion. Similarly, there are applications where the DSP is solely responsible for generating the signal to the
DAC. If a DAC is used, it must be followed by an analog anti-imaging filter to remove the image frequencies. Finally, there are slower speed industrial process control systems where sampling rates are much lower—regardless of the system, the fundamentals of sampling theory still apply.

There are two key concepts involved in the actual analog-to-digital and digital-to-analog conversion process: discrete time sampling and finite amplitude resolution due to quantization. This tutorial discusses discrete time sampling.

THE NEED FOR A SAMPLE-AND-HOLD AMPLIFIER (SHA) FUNCTION

The generalized block diagram of a sampled data system shown in Figure 1 assumes some type of ac signal at the input. It should be noted that this does not necessarily have to be so, as in the case of modern digital voltmeters (DVMs) or ADCs optimized for dc measurements, but for this discussion assume that the input signal has some upper frequency limit $f_a$.

Most ADCs today have a built-in sample-and-hold function, thereby allowing them to process ac signals. This type of ADC is referred to as a sampling ADC. However many early ADCs, such as Analog Devices’ industry-standard AD574, were not of the sampling type, but simply encoders as shown in Figure 2. If the input signal to a SAR ADC (assuming no SHA function) changes by more than 1 LSB during the conversion time (8 µs in the example), the output data can have large errors, depending on the location of the code. Most ADC architectures are subject to this type of error—some more, some less—with the possible exception of flash converters having well-matched comparators.

\[
\begin{align*}
\text{ANALOG INPUT} \\
v(t) &= q \frac{2^n}{2} \sin(2\pi f t) \\
\frac{dv}{dt} &= q \frac{2^{N}}{2} 2\pi f \cos(2\pi f t) \\
\left| \frac{dv}{dt} \right|_{\text{max}} &= q \frac{2^{N}}{2} 2\pi f \\
f_{\text{max}} &= \frac{\left| \frac{dv}{dt} \right|_{\text{max}}}{2\pi q} \\
&= \frac{dv}{dt} \left| \text{max} \right. \\
&= \frac{q}{\pi 2^n}
\end{align*}
\]

**EXAMPLE:**
\[
\begin{align*}
dv &= 1\text{ LSB} - q \\
dt &= 8\mu s \\
N &= 12, \quad 2^N = 4096 \\
f_{\text{max}} &= 9.7\text{ Hz}
\end{align*}
\]

**Figure 2: Input Frequency Limitations of Non-Sampling ADC (Encoder)**

Assume that the input signal to the encoder is a sinewave with a full-scale amplitude (q$2^N/2$), where $q$ is the weight of 1 LSB.

\[
v(t) = q \frac{2^{N}}{2} \sin \left( 2\pi f t \right). \\
\text{Eq. 1}
\]

Taking the derivative:

\[
\frac{dv}{dt} = 2\pi f q \frac{2^{N}}{2} \cos (2\pi f t). \\
\text{Eq. 2}
\]

The maximum rate of change is therefore:

\[
\left| \frac{dv}{dt} \right|_{\text{max}} = 2\pi f q \frac{2^{N}}{2}. \\
\text{Eq. 3}
\]

Solving for $f$: 
If \( N = 12 \), and 1 LSB change (\( dv = q \)) is allowed during the conversion time (\( dt = 8 \mu s \)), then the equation can be solved for \( f_{\text{max}} \), the maximum full-scale signal frequency that can be processed without error:

\[
f_{\text{max}} = \frac{\frac{dv}{dt}}{2q} = \frac{9.7}{12} = 0.808\text{ Hz}.
\]

This implies any input frequency greater than 0.808 Hz is subject to conversion errors, even though a sampling frequency of 100 kSPS is possible with the 8-µs ADC (this allows an extra 2-µs interval for an external SHA to re-acquire the signal after coming out of the hold mode).

To process ac signals, a sample-and-hold (SHA) function is added as shown in Figure 3. The ideal SHA is simply a switch driving a hold capacitor followed by a high input impedance buffer. The input impedance of the buffer must be high enough so that the capacitor is discharged by less than 1 LSB during the hold time. The SHA samples the signal in the sample mode, and holds the signal constant during the hold mode. The timing is adjusted so that the encoder performs the conversion during the hold time. A sampling ADC can therefore process fast signals – the upper frequency limitation is determined by the SHA aperture jitter, bandwidth, distortion, etc., not the encoder. In the example shown, the sample-and-hold acquires the signal in 2 µs, the encoder converts the signal in 8 µs, yielding a total sampling period of 10 µs. This yields a sampling frequency of 100 kSPS, and the capability of processing input frequencies up to 50 kHz.

It is important to understand a subtle difference between a true sample-and-hold amplifier (SHA) and a track-and-hold amplifier (T/H, or THA). Strictly speaking, the output of a sample-and-hold is not defined during the sample mode, however the output of a track-and-hold tracks the signal during the sample or track mode. In practice, the function is generally implemented as a track-and-hold, and the terms track-and-hold and sample-and-hold are often used interchangeably. The waveforms shown in Figure 3 are those associated with a track-and-hold.

**Figure 3: Sample-and-Hold Function Required for Digitizing AC Signals**

THE NYQUIST CRITERION

A continuous analog signal is sampled at discrete intervals, \( t_s = 1/f_s \), which must be carefully chosen to ensure an accurate
representation of the original analog signal. It is clear that the more samples taken (faster sampling rates), the more accurate the
digital representation, but if fewer samples are taken (lower sampling rates), a point is reached where critical information about
the signal is actually lost. The mathematical basis of sampling was set forth by Harry Nyquist of Bell Telephone Laboratories in
two classic papers published in 1924 and 1928, respectively. (See References 1 and 2 as well as Chapter 2 of Reference 6).
Nyquist’s original work was shortly supplemented by R. V. L. Hartley (Reference 3). These papers formed the basis for the PCM
work to follow in the 1940s, and in 1948 Claude Shannon wrote his classic paper on communication theory (Reference 4).

Simply stated, the Nyquist criterion requires that the sampling frequency be at least twice the highest frequency contained in the
signal, or information about the signal will be lost. If the sampling frequency is less than twice the maximum analog signal
frequency, a phenomenon known as aliasing will occur.

In order to understand the implications of aliasing in both the time and frequency domain, first consider the case of a time domain
representation of a single tone sinewave sampled as shown in Figure 4. In this example, the sampling frequency $f_s$ is not at least
$2f_a$, but only slightly more than the analog input frequency $f_a$ – the Nyquist criterion is violated. Notice that the pattern of the
actual samples produces an aliased sinewave at a lower frequency equal to $f_s - f_a$.

Figure 4: Aliasing in the Time Domain

The corresponding frequency domain representation of this scenario is shown in Figure 5B. Now consider the case of a single
frequency sinewave of frequency $f_a$ sampled at a frequency $f_s$ by an ideal impulse sampler (see Figure 5A). Also assume that $f_s$
$> 2f_a$ as shown. The frequency-domain output of the sampler shows aliases or images of the original signal around every multiple
of $f_s$, i.e. at frequencies equal to $\pm Kf_s \pm f_a$, $K = 1, 2, 3, 4, .....$
Figure 5: Analog Signal $f_a$ Sampled @ $f_s$ Using Ideal Sampler
Has Images (Aliases) at $|\pm Kf_s \pm f_a|$, $K = 1, 2, 3, ...$

The Nyquist bandwidth is defined to be the frequency spectrum from dc to $f_s/2$. The frequency spectrum is divided into an infinite number of Nyquist zones, each having a width equal to $0.5f_s$ as shown. In practice, the ideal sampler is replaced by an ADC followed by an FFT processor. The FFT processor only provides an output from dc to $f_s/2$, i.e., the signals or aliases which appear in the first Nyquist zone.

Now consider the case of a signal which is outside the first Nyquist zone (Figure 5B). The signal frequency is only slightly less than the sampling frequency, corresponding to the condition shown in the time domain representation in Figure 4. Notice that even though the signal is outside the first Nyquist zone, its image (or alias), $f_s - f_a$, falls inside. Returning to Figure 5A, it is clear that if an unwanted signal appears at any of the image frequencies of $f_a$, it will also occur at $f_a$, thereby producing a spurious frequency component in the first Nyquist zone.

This is similar to the analog mixing process and implies that some filtering ahead of the sampler (or ADC) is required to remove frequency components which are outside the Nyquist bandwidth, but whose aliased components fall inside it. The filter performance will depend on how close the out-of-band signal is to $f_s/2$, and the amount of attenuation required.

BASEBAND ANTIALIASING FILTERS

Baseband sampling implies that the signal to be sampled lies in the first Nyquist zone. It is important to note that with no input filtering at the input of the ideal sampler, any frequency component (either signal or noise) that falls outside the Nyquist bandwidth in any Nyquist zone will be aliased back into the first Nyquist zone. For this reason, an antialiasing filter is used in almost all sampling ADC applications to remove these unwanted signals.

Properly specifying the antialiasing filter is important. The first step is to know the characteristics of the signal being sampled. Assume that the highest frequency of interest is $f_a$. The antialiasing filter passes signals from dc to $f_a$ while attenuating signals above $f_a$.

Assume that the corner frequency of the filter is chosen to be equal to $f_a$. The effect of the finite transition from minimum to maximum attenuation on system dynamic range is illustrated in Figure 6A.
Figure 6: Oversampling Relaxes Requirements on Baseband Anti-aliasing Filter

Assume that the input signal has full-scale components well above the maximum frequency of interest, \( f_a \). The diagram shows how full-scale frequency components above \( f_s - f_a \) are aliased back into the bandwidth \( dc \) to \( f_a \). These aliased components are indistinguishable from actual signals and therefore limit the dynamic range to the value on the diagram which is shown as \( DR \).

Some texts recommend specifying the anti-aliasing filter with respect to the Nyquist frequency, \( f_s/2 \), but this assumes that the signal bandwidth of interest extends from \( dc \) to \( f_s/2 \) which is rarely the case. In the example shown in Figure 6A, the aliased components between \( f_s \) and \( f_s/2 \) are not of interest and do not limit the dynamic range.

The anti-aliasing filter transition band is therefore determined by the corner frequency \( f_a \), the stopband frequency \( f_s - f_a \), and the desired stopband attenuation, \( DR \). The required system dynamic range is chosen based on the requirement for signal fidelity.

Filters become more complex as the transition band becomes sharper, all other things being equal. For instance, a Butterworth filter gives 6-dB attenuation per octave for each filter pole (as do all filters). Achieving 60-dB attenuation in a transition region between 1 MHz and 2 MHz (1 octave) requires a minimum of 10 poles – not a trivial filter, and definitely a design challenge.

Therefore, other filter types are generally more suited to applications where the requirement is for a sharp transition band and in-band flatness coupled with linear phase response. Elliptic filters meet these criteria and are a popular choice. There are a number of companies which specialize in supplying custom analog filters. TTE is an example of such a company (Reference 5). As an example, the normalized response of the TTE, Inc., LE1182 11-pole elliptic anti-aliasing filter is shown in Figure 7. Notice that this filter is specified to achieve at least 80 dB attenuation between \( f_c \) and \( 1.2f_c \). The corresponding passband ripple, return loss, delay, and phase response are also shown in Figure 7.
From this discussion, we can see how the sharpness of the antialiasing transition band can be traded off against the ADC sampling frequency. Choosing a higher sampling rate (oversampling) reduces the requirement on transition band sharpness (hence, the filter complexity) at the expense of using a faster ADC and processing data at a faster rate. This is illustrated in Figure 6B which shows the effects of increasing the sampling frequency by a factor of K, while maintaining the same analog corner frequency, $f_a$, and the same dynamic range, DR, requirement. The wider transition band ($f_a$ to $Kf_a - f_a$) makes this filter easier to design than for the case of Figure 6A.

The antialiasing filter design process is started by choosing an initial sampling rate of 2.5 to 4 times $f_a$. Determine the filter specifications based on the required dynamic range and see if such a filter is realizable within the constraints of the system cost and performance. If not, consider a higher sampling rate which may require using a faster ADC. It should be mentioned that sigma-delta ADCs are inherently highly oversampled converters, and the resulting relaxation in the analog anti-aliasing filter requirements is therefore an added benefit of this architecture.

The antialiasing filter requirements can also be relaxed somewhat if it is certain that there will never be a full-scale signal at the stopband frequency $f_s - f_a$. In many applications, it is improbable that full-scale signals will occur at this frequency. If the maximum signal at the frequency $f_s - f_a$ will never exceed X dB below full-scale, then the filter stopband attenuation requirement can be reduced by that same amount. The new requirement for stopband attenuation at $f_s - f_a$ based on this knowledge of the signal is now only DR - X dB. When making this type of assumption, be careful to treat any noise signals which may occur above the maximum signal frequency $f_a$ as unwanted signals which will also alias back into the signal bandwidth.

**UNDERSAMPLING (HARMONIC SAMPLING, BANDPASS SAMPLING, IF SAMPLING, DIRECT IF-TO-DIGITAL CONVERSION)**

Thus far we have considered the case of baseband sampling, i.e., all the signals of interest lie within the first Nyquist zone. Figure 8A shows such a case, where the band of sampled signals is limited to the first Nyquist zone, and images of the original band of frequencies appear in each of the other Nyquist zones.

Consider the case shown in Figure 8B, where the sampled signal band lies entirely within the second Nyquist zone. The process of sampling a signal outside the first Nyquist zone is often referred to as undersampling, or harmonic sampling. Note that the image which falls in the first Nyquist zone contains all the information in the original signal, with the exception of its original location (the order of the frequency components within the spectrum is reversed, but this is easily corrected by re-ordering the output of the FFT).
Figure 8C shows the sampled signal restricted to the third Nyquist zone. Note that the image that falls into the first Nyquist zone has no spectral reversal. In fact, the sampled signal frequencies may lie in any unique Nyquist zone, and the image falling into the first Nyquist zone is still an accurate representation (with the exception of the spectral reversal which occurs when the signals are located in even Nyquist zones). At this point we can restate the Nyquist criterion as it applies to broadband signals:

A signal of bandwidth BW must be sampled at a rate equal to or greater than twice its bandwidth \((2BW)\) in order to preserve all the signal information.

Notice that there is no mention of the absolute location of the band of sampled signals within the frequency spectrum relative to the sampling frequency. The only constraint is that the band of sampled signals be restricted to a single Nyquist zone, i.e., the signals must not overlap any multiple of \(f_s/2\) (this, in fact, is the primary function of the antialiasing filter).

Sampling signals above the first Nyquist zone has become popular in communications, because the process is equivalent to analog demodulation. It is becoming common practice to sample IF signals directly and then use digital techniques to process the signal, thereby eliminating the need for an IF demodulator and filters. Clearly, however, as the IF frequencies become higher, the dynamic performance requirements on the ADC become more critical. The ADC input bandwidth and distortion performance must be adequate at the IF frequency, rather than only baseband. This presents a problem for most ADCs designed to only process signals in the first Nyquist zone – an ADC suitable for undersampling applications must maintain dynamic performance into the higher order Nyquist zones.

**ANTIALIASING FILTERS IN UNDERSAMPLING APPLICATIONS**

Figure 9 shows a signal in the second Nyquist zone centered around a carrier frequency, \(f_c\), whose lower and upper frequencies are \(f_1\) and \(f_2\). The antialiasing filter is a bandpass filter. The desired dynamic range is DR, which defines the filter stopband attenuation. The upper transition band is \(f_2\) to \(2f_s - f_2\), and the lower is \(f_1\) to \(f_s - f_1\). As in the case of baseband sampling, the antialiasing filter requirements can be relaxed by proportionally increasing the sampling frequency, but \(f_c\) must also be changed so that it is always centered in the second Nyquist zone.
Two key equations can be used to select the sampling frequency, $f_s$, given the carrier frequency, $f_c$, and the bandwidth of its signal, $\Delta f$. The first is the Nyquist criteria:

$$f_s > 2\Delta f.$$  \hspace{1cm} \text{Eq. 5}

The second equation ensures that $f_c$ is placed in the center of a Nyquist zone:

$$f_s = \frac{4f_c}{2NZ - 1},$$ \hspace{1cm} \text{Eq. 6}

where $NZ = 1, 2, 3, 4, \ldots$ and $NZ$ corresponds to the Nyquist zone in which the carrier and its signal fall (see Figure 10).

NZ is normally chosen to be as large as possible thereby allowing high IF frequencies. Regardless of the choice for NZ, the Nyquist criterion requires that $f_s > 2\Delta f$. If NZ is chosen to be odd, then $f_c$ and its signal will fall in an odd Nyquist zone, and the image frequencies in the first Nyquist zone will not be reversed.
As an example, consider a 4-MHz wide signal centered around a carrier frequency of 71 MHz. The minimum required sampling frequency is therefore 8 MSPS. Solving Eq. 6 for NZ using $f_c = 71$ MHz and $f_s = 8$ MSPS yields $NZ = 18.25$. However, NZ must be an integer, so we round 18.25 to the next lowest integer, 18. Solving Eq. 6 again for $f_s$ yields $f_s = 8.1143$ MSPS. The final values are therefore $f_s = 8.1143$ MSPS, $f_c = 71$ MHz, and $NZ = 18$.

Now assume that we desire more margin for the antialiasing filter, and we select $f_s$ to be 10 MSPS. Solving Eq. 6 for NZ, using $f_c = 71$ MHz and $f_s = 10$ MSPS yields $NZ = 14.7$. We round 14.7 to the next lowest integer, giving $NZ = 14$. Solving Eq. 6 again for $f_s$ yields $f_s = 10.519$ MSPS. The final values are therefore $f_s = 10.519$ MSPS, $f_c = 71$ MHz, and $NZ = 14$.

The above iterative process can also be carried out starting with $f_s$ and adjusting the carrier frequency to yield an integer number for NZ.

**SUMMARY**

This tutorial has covered the basics of the Nyquist criterion and the effects of aliasing in both the time and frequency domain. A working knowledge of the criterion was used to show how to adequately specify the antialiasing filter. Oversampling and undersampling examples were shown in relationship to modern applications in communications systems.

**REFERENCES**

MT-003: Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don’t Get Lost in the Noise Floor

by Walt Kester

REV. 0, 10-03-2005

INTRODUCTION

Six popular specifications for quantifying ADC dynamic performance are SINAD (signal-to-noise-and-distortion ratio), ENOB (effective number of bits), SNR (signal-to-noise ratio), THD (total harmonic distortion), THD + N (total harmonic distortion plus noise), and SFDR (spurious free dynamic range). Although most ADC manufacturers have adopted the same definitions for these specifications, some exceptions still exist. Because of their importance in comparing ADCs, it is important not only to understand exactly what is being specified, but the relationships between the specifications.

There are a number of ways to quantify the distortion and noise of an ADC. All of them are based on an FFT analysis using a generalized test setup such as shown in Figure 1.

![Figure 1: Generalized Test Setup for FFT Analysis of ADC Output](image)

The spectral output of the FFT is a series of M/2 points in the frequency domain (M is the size of the FFT—the number of samples stored in the buffer memory). The spacing between the points is \( f_s/M \), and the total frequency range covered is dc to \( f_s/2 \), where \( f_s \) is the sampling rate. The width of each frequency “bin” (sometimes called the resolution of the FFT) is \( f_s/M \). Figure 2 shows an FFT output for an ideal 12-bit ADC using the Analog Devices’ ADIsimADC® program. Note that the theoretical noise floor of the FFT is equal to the theoretical SNR plus the FFT process gain, \( 10 \times \log(M/2) \). It is important to remember that the value for noise used in the SNR calculation is the noise that extends over the entire Nyquist bandwidth (dc to \( f_s/2 \)), but the FFT acts as a narrowband spectrum analyzer with a bandwidth of \( f_s/M \) that sweeps over the spectrum. This has the effect of pushing the noise down by an amount equal to the process gain—the same effect as narrowing the bandwidth of an analog spectrum analyzer.

The FFT data shown in Figure 2 represents the average of 5 individual FFTs. Note that averaging a number of FFTs does not affect the average noise floor, it only acts to “smooth” the random variations in the amplitudes contained in each frequency bin.
The FFT output can be used like an analog spectrum analyzer to measure the amplitude of the various harmonics and noise components of a digitized signal. The harmonics of the input signal can be distinguished from other distortion products by their location in the frequency spectrum. Figure 3 shows a 7-MHz input signal sampled at 20 MSPS and the location of the first 9 harmonics. Aliased harmonics of $f_a$ fall at frequencies equal to $|\pm Kf_s \pm nf_a|$, where $n$ is the order of the harmonic, and $K = 0, 1, 2, 3, \ldots$. The second and third harmonics are generally the only ones specified on a data sheet because they tend to be the largest, although some data sheets may specify the value of the worst harmonic.

Harmonic distortion is normally specified in dBc (decibels below carrier), although in audio applications it may be specified as a percentage. It is the ratio of the rms signal to the rms value of the harmonic in question. Harmonic distortion is generally specified with an input signal near full-scale (generally 0.5 to 1 dB below full-scale to prevent clipping), but it can be specified at any level. For signals much lower than full-scale, other distortion products due to the differential nonlinearity (DNL) of the converter—not direct harmonics—may limit performance.

**Total harmonic distortion (THD)** is the ratio of the rms value of the fundamental signal to the mean value of the root-sum-square
of its harmonics (generally, only the first 5 harmonics are significant). THD of an ADC is also generally specified with the input signal close to full-scale, although it can be specified at any level.

Total harmonic distortion plus noise (THD + N) is the ratio of the rms value of the fundamental signal to the mean value of the root-sum-square of its harmonics plus all noise components (excluding dc). The bandwidth over which the noise is measured must be specified. In the case of an FFT, the bandwidth is dc to \( f_s/2 \). (If the bandwidth of the measurement is dc to \( f_s/2 \) (the Nyquist bandwidth), THD + N is equal to SINAD—see below). Be warned, however, that in audio applications the measurement bandwidth may not necessarily be the Nyquist bandwidth.

Spurious free dynamic range (SFDR) is the ratio of the rms value of the signal to the rms value of the worst spurious signal regardless of where it falls in the frequency spectrum. The worst spur may or may not be a harmonic of the original signal. SFDR is an important specification in communications systems because it represents the smallest value of signal that can be distinguished from a large interfering signal (blocker). SFDR can be specified with respect to full-scale (dBFS) or with respect to the actual signal amplitude (dBc). The definition of SFDR is shown graphically in Figure 4.

![Figure 4: Spurious Free Dynamic Range (SFDR)](image)

The Analog Devices' ADIsimADC® ADC modeling program allows various high performance ADCs to be evaluated at various operating frequencies, levels, and sampling rates. The models yield an accurate representation of actual performance, and a typical FFT output for the AD9444 14-bit, 80-MSPS ADC is shown in Figure 5. Note that the input frequency is 95.111 MHz and is aliased back to 15.111 MHz by the sampling process. The output also displays the locations of the first five harmonics. In this case, all the harmonics are aliases. The program also calculates and tabulates the important performance parameters as shown in the left-hand data column.

![Figure 5: AD9444 14-Bit, 80MSPS ADC fin = 95.111MHz, fs = 80MSPS,](image)
SIGNAL-TO-NOISE-AND-DISTORTION RATIO (SINAD), SIGNAL-TO-NOISE RATIO (SNR), AND EFFECTIVE NUMBER OF BITS (ENOB)

SINAD and SNR deserve careful attention, because there is still some variation between ADC manufacturers as to their precise meaning. Signal-to-Noise-and-Distortion (SINAD, or $S/(N+D)$) is the ratio of the rms signal amplitude to the mean value of the root-sum-square (rss) of all other spectral components, including harmonics, but excluding dc. SINAD is a good indication of the overall dynamic performance of an ADC because it includes all components which make up noise and distortion. SINAD is often plotted for various input amplitudes and frequencies. For a given input frequency and amplitude, SINAD is equal to THD + N, provided the bandwidth for the noise measurement is the same for both (the Nyquist bandwidth). A typical plot for the AD9226 12-bit, 65-MSPS ADC is shown in Figure 6.

Figure 6: AD9226 12-bit, 65-MSPS ADC SINAD and ENOB for Various Input Full-Scale Spans (Range)

The SINAD plot shows that the ac performance of the ADC degrades due to high-frequency distortion and is usually plotted for frequencies well above the Nyquist frequency so that performance in undersampling applications can be evaluated. SINAD plots such as these are very useful in evaluating the dynamic performance of ADCs. SINAD is often converted to effective-number-of-bits (ENOB) using the relationship for the theoretical SNR of an ideal N-bit ADC: $SNR = 6.02N + 1.76$ dB. The equation is solved for N, and the value of SINAD is substituted for SNR:

$$ENOB = \frac{SINAD - 1.76 \text{ dB}}{6.02}$$  \hspace{1cm} \text{Eq. 1}

Note that Equation 1 assumes a full-scale input signal. If the signal level is reduced, the value of SINAD decreases, and the ENOB decreases. It is necessary to add a correction factor for calculating ENOB at reduced signal amplitudes as shown in Equation 2:

$$ENOB = \frac{SINAD_{\text{MEASURED}} - 1.76 \text{ dB} + 20 \log \left( \frac{\text{Fullscale Amplitude}}{\text{Input Amplitude}} \right)}{6.02}$$  \hspace{1cm} \text{Eq. 2}

The correction factor essentially "normalizes" the ENOB value to full-scale regardless of the actual signal amplitude.

Signal-to-noise ratio (SNR, or sometimes called SNR-without-harmonics) is calculated from the FFT data the same as SINAD, except that the signal harmonics are excluded from the calculation, leaving only the noise terms. In practice, it is only necessary...
to exclude the first 5 harmonics, since they dominate. The SNR plot will degrade at high input frequencies, but generally not as rapidly as SINAD because of the exclusion of the harmonic terms.

A few ADC data sheets somewhat loosely refer to SINAD as SNR, so you must be careful when interpreting these specifications and understand exactly what the manufacturer means.

**THE MATHEMATICAL RELATIONSHIPS BETWEEN SINAD, SNR, AND THD**

There is a mathematical relationship between SINAD, SNR, and THD (assuming all are measured with the same input signal amplitude and frequency. In the following equations, SNR, THD, and SINAD are expressed in dB, and are derived from the actual numerical ratios S/N, S/D, and S/(N+D) as shown below:

\[
\begin{align*}
\text{SNR} &= 20 \log \left( \frac{S}{N} \right), \\
\text{THD} &= 20 \log \left( \frac{S}{D} \right), \\
\text{SINAD} &= 20 \log \left( \frac{S}{N+D} \right).
\end{align*}
\]

Eq. 3, Eq. 4, and Eq. 5 can be solved for the numerical ratios N/S, D/S, and (N+D)/S as follows:

\[
\begin{align*}
\frac{N}{S} &= 10^{-\text{SNR} / 20} \\
\frac{D}{S} &= 10^{-\text{THD} / 20} \\
\frac{N+D}{S} &= 10^{-\text{SINAD} / 20}
\end{align*}
\]

Because the denominators of Eq. 6, Eq. 7, and Eq. 8 are all equal to S, the root sum square of N/S and D/S is equal to (N+D)/S as follows:

\[
\frac{N+D}{S} = \left[ \left( \frac{N}{S} \right)^2 + \left( \frac{D}{S} \right)^2 \right]^{1/2} = \left[ \left( 10^{-\text{SNR} / 20} \right)^2 + \left( 10^{-\text{THD} / 20} \right)^2 \right]^{1/2},
\]

\[
\frac{N+D}{S} = \left[ 10^{-\text{SNR} / 10} + 10^{-\text{THD} / 10} \right]^{1/2}.
\]

Therefore, S/(N+D) must equal:

\[
\frac{S}{N+D} = \left[ 10^{-\text{SNR} / 10} + 10^{-\text{THD} / 10} \right]^{-1/2}.
\]

and hence,
\[
\text{SINAD} = 20 \log \left( \frac{S}{S + D} \right) = -10 \log \left[ 10^{-\text{SNR}/10} + 10^{-\text{THD}/10} \right]. \quad \text{Eq. 12}
\]

Eq. 12 gives us SINAD as a function of SNR and THD.

Similarly, if we know SINAD and THD, we can solve for SNR as follows:

\[
\text{SNR} = 20 \log \left( \frac{S}{N} \right) = -10 \log \left[ 10^{-\text{SINAD}/10} - 10^{-\text{THD}/10} \right]. \quad \text{Eq. 13}
\]

Similarly, if we know SINAD and SNR, we can solve for THD as follows:

\[
\text{THD} = 20 \log \left( \frac{S}{D} \right) = -10 \log \left[ 10^{-\text{SINAD}/10} - 10^{-\text{SNR}/10} \right]. \quad \text{Eq. 14}
\]

Equations 12, 13, and 14 are implemented in an easy to use design tool on the Analog Devices' website. It is important to emphasize again that these relationships hold true only if the input frequency and amplitude are equal for all three measurements.

**SUMMARY**

Because SINAD, SNR, ENOB, THD, THD + N, and SFDR are common measures of ADC dynamic performance, a complete understanding of them in the context of the manufacturers' data sheet is critical. This tutorial has defined the quantities and derived the mathematical relationship between SINAD, SNR, and THD.

**REFERENCES**

INTRODUCTION

All analog-to-digital converters (ADCs) have a certain amount of "input-referred noise" – modeled as a noise source connected in series with the input of a noise-free ADC. Input-referred noise is not to be confused with quantization noise which only occurs when an ADC is processing an ac signal. In most cases, less input noise is better, however there are some instances where input noise can actually be helpful in achieving higher resolution. This probably doesn't make sense right now, so you will just have to read further into this tutorial to find out how SOME noise can be GOOD noise.

INPUT-REFERRED NOISE (CODE TRANSITION NOISE)

Practical ADCs deviate from ideal ADCs in many ways. Input-referred noise is certainly a departure from the ideal, and its effect on the overall ADC transfer function is shown in Figure 1. As the analog input voltage is increased, the "ideal" ADC (shown in Figure 1A) maintains a constant output code until the transition region is reached, at which point the output code instantly jumps to the next value and remains there until the next transition region is reached. A theoretically perfect ADC has zero "code transition" noise, and a transition region width equal to zero. A practical ADC has certain amount of code transition noise, and therefore a transition region width that depends on the amount of input-referred noise present (shown in Figure 1B). Figure 1B shows a situation where the width of the code transition noise is approximately one least significant bit (LSB) peak-to-peak.

![Figure 1: Code Transition Noise (Input-referred Noise) and its Effect on ADC Transfer Function](image)

All ADC internal circuits produce a certain amount of rms noise due to resistor noise and "kT/C" noise. This noise is present even for dc-input signals, and accounts for the code transition noise. Today, code transition noise is most often referred to as input-referred noise rather than code transition noise. The input-referred noise is most often characterized by examining the histogram of a number of output samples when the input to the ADC is a dc value. The output of most high speed or high resolution ADCs is a distribution of codes, centered around the nominal value of the dc input (see Figure 2). To measure its value, the input of the ADC is either grounded or connected to a heavily decoupled voltage source, and a large number of output samples are collected and plotted as a histogram (sometimes referred to as a grounded-input histogram). Since the noise is approximately Gaussian, the standard deviation of the histogram, \( \sigma \), can be calculated, corresponding to the effective input rms noise. Reference 1 provides a detailed description of how to calculate the value of \( \sigma \) from the histogram data. It is common practice to express this rms noise in terms of LSBs rms, although it can be expressed as an rms voltage referenced to the ADC full-scale input range.
Although the inherent differential nonlinearity (DNL) of the ADC may cause some minor deviations from an ideal Gaussian distribution (some DNL is shown in Figure 2, for instance), it should be at least approximately Gaussian. If there is significant DNL, the value of $s$ should be calculated for several different dc input voltages, and the results averaged. If the code distribution has large and distinct peaks and valleys, for instance—this indicates either a poorly designed ADC or, more likely, a bad PC board layout, poor grounding techniques, or improper power supply decoupling (see Figure 3). Another indication of trouble is when the width of the distribution changes drastically as the dc input is swept over the ADC input voltage range.

**NOISE-FREE (FLICKER-FREE) CODE RESOLUTION**

The *noise-free code resolution* of an ADC is the number of bits beyond which it is impossible to distinctly resolve individual codes. The cause is the effective input noise (or input-referred noise) associated with all ADCs and described above. This noise can be expressed as an rms quantity, usually having the units of $LSBs \text{ rms}$. Multiplying by a factor of 6.6 converts the rms noise...
into peak-to-peak noise (expressed in LSBs peak-to-peak). The total range (or span) of an N-bit ADC is $2^N$ LSBs. The total number of noise-free counts is therefore equal to:

$$\text{Noise - Free Counts} = \frac{2^N}{\text{Peak - Peak Input Noise (LSBs)}}.$$  \hspace{1cm} \text{Eq. 1}

The number of noise-free counts can be converted into noise-free code resolution by taking the base-2 logarithm as follows:

$$\text{Noise Free Code Resolution} = \log_2 \left( \frac{2^N}{\text{Peak - Peak Input Noise (LSBs)}} \right).$$  \hspace{1cm} \text{Eq. 2}

The noise-free code resolution specification is generally associated with high-resolution sigma-delta measurement ADCs. It is most often a function of sampling rate, digital filter bandwidth, and programmable gain amplifier (PGA) gain. Figure 4 shows a typical table taken from the AD7730 sigma-delta measurement ADC.

The ratio of the FS range to the $\text{rms}$ input noise (rather than peak-to-peak noise) is sometimes used to calculate resolution. In this case, the term effective resolution is used. Note that under identical conditions, effective resolution is larger than noise-free code resolution by $\log_2(6.6)$, or approximately 2.7 bits.

$$\text{Effective Resolution} = \log_2 \left( \frac{2^N}{\text{RMS Input Noise (LSBs)}} \right).$$  \hspace{1cm} \text{Eq. 3}

$$\text{Effective Resolution} = \text{Noise Free Code Resolution} + 2.7 \text{ Bits}$$  \hspace{1cm} \text{Eq. 4}

Some manufacturers prefer to specify effective resolution rather than noise-free code resolution because it results in a higher number of bits—the user should check the data sheet closely to make sure which is actually specified.

**INCREASING ADC “RESOLUTION” AND REDUCING NOISE BY DIGITAL AVERAGING**

The effects of input-referred noise can be reduced by digital averaging. Consider a 16-bit ADC which has 15 noise-free bits at a sampling rate of 100 kSOPS. Averaging two samples per output sample reduces the effective sampling rate to 50 kSOPS and increases the SNR by 3 dB, and the noise-free bits to 15.5. Averaging four samples per output sample reduces the sampling rate to 25 kSOPS, increases the SNR by 6 dB, and increases the noise-free bits to 16.

In fact, if we average sixteen samples per output, the output sampling rate is reduced to 6.25 kSOPS, the SNR increases by
another 6 dB, and the noise-free bits increase to 17. The arithmetic in the averaging must be carried out to the larger number of significant bits in order to take advantage of the extra "resolution."

The averaging process also helps "smooth" out the DNL errors in the ADC transfer function. This can be illustrated for the simple case where the ADC has a missing code at quantization level "k". Even though code "k" is missing because of the large DNL error, the average of the two adjacent codes, k−1 and k+1 is equal to k.

This technique can therefore be used effectively to increase the dynamic range of the ADC, at the expense of overall output sampling rate and extra digital hardware. It should be noted, however, that averaging will not correct the inherent integral nonlinearity of the ADC.

Now, consider the case of an ADC that has extremely low input-referred noise, and the histogram shows a solid code all the time. What will digital averaging do for this ADC? This answer is simple—it will do nothing! No matter how many samples we average, we will still get the same answer. However, as soon as we add enough noise to the input signal so that there is more than one code in the histogram, the averaging method starts working again. Some small amount of noise is therefore good (at least with respect to the averaging method), but the more noise present at the input, the more averaging is required to achieve the same resolution.

DON'T CONFUSE EFFECTIVE NUMBER OF BITS (ENOB) WITH "EFFECTIVE RESOLUTION" OR "NOISE-FREE CODE RESOLUTION"

Because of the similarity of the terms, effective number of bits and effective resolution are often assumed to be equal. This is not the case.

Effective number of bits (ENOB) is derived from an FFT analysis of the ADC output when the ADC is stimulated with a full-scale sinewave input signal. The rss value of all noise and distortion terms is computed, and the ratio of the signal to the noise and distortion is defined as SINAD, or S/(N+D). The theoretical SNR of a perfect N-bit ADC is given by:

\[ \text{SNR} = 6.02N + 1.76\text{dB} \]  
Eq. 5

The calculated value of SINAD for the ADC is substituted for SNR in Eq. 5, and the equation solved for N, yielding the ENOB:

\[ \text{ENOB} = \frac{\text{SINAD} - 1.76\text{dB}}{6.02} \]  
Eq. 6

The noise and distortion used to calculate SINAD and ENOB include not only the input-referred noise but also the quantization noise and the distortion terms. SINAD and ENOB are used to measure the dynamic performance of an ADC, while effective resolution and noise-free code resolution are used to measure the noise of the ADC under dc input conditions where there is no quantization noise.

USING NOISE DITHER TO INCREASE ADC SPURIOUS FREE DYNAMIC RANGE

There are two fundamental limitations to maximizing SFDR in a high speed ADC. The first is the distortion produced by the front-end amplifier and the sample-and-hold circuit. The second is that produced by non-linearity in the actual transfer function of the encoder portion of the ADC. The key to high SFDR is to minimize the non-linearity of each.

There is nothing that can be done externally to the ADC to significantly reduce the inherent distortion caused by the ADC front end. However, the differential nonlinearity in the ADC encoder transfer function can be reduced by the proper use of dither (external noise which is summed with the analog input signal to the ADC).

Dithering improves ADC SFDR under certain conditions (References 2-5). For example, even in a perfect ADC, there is some correlation between the quantization noise and the input signal. This can reduce the SFDR of the ADC, especially if the input signal is an exact sub-multiple of the sampling frequency. Summing broadband noise (about 1/2 LSB rms in amplitude) with the input signal tends to randomize the quantization noise and minimize this effect (see Figure 5A). In most systems, however, there is enough noise riding on top of the signal so that adding additional dither noise is not required. The input-referred noise of the ADC may also be enough to produce the same effect. Increasing the wideband rms noise level beyond approximately one LSB will proportionally reduce the ADC SNR and results in no additional improvement.
Other schemes have been developed which use larger amounts of dither noise to randomize the transfer function of the ADC. Figure 5B also shows a dither noise source comprised of a pseudo-random number generator which drives a DAC. This signal is subtracted from the ADC input signal and then digitally added to the ADC output, thereby causing no significant degradation in SNR. An inherent disadvantage of this technique is that the allowable input signal swing is reduced as the amplitude of the dither signal is increased. This reduction in signal amplitude is required to prevent overdriving the ADC. It should be noted that this scheme does not significantly improve distortion created by the front-end of the ADC, only that produced by the non-linearity of the ADC encoder transfer function.

Another method which is easier to implement, especially in wideband receivers, is to inject a narrowband dither signal outside the signal band of interest as shown in Figure 6. Usually, there are no signal components located in the frequency range near dc, so this low-frequency region is often used for such a dither signal. Another possible location for the dither signal is slightly below $f_s/2$. Because the dither signal occupies only a small bandwidth relative to the signal bandwidth (usually a bandwidth of a few hundred kHz is sufficient), there is no significant degradation in SNR, as would occur if the dither was broadband.
A subranging pipelined ADC, such as the AD6645 14-bit, 105 MSPS ADC, (see Figure 7), has very small differential non-linearity errors that occur at specific code transition points across the ADC range. The AD6645 uses a 5-bit ADC (ADC1) followed by a 5-bit ADC2 and a 6-bit ADC3. The only significant DNL errors occur at the ADC1 transition points—the second and third stage ADC DNL errors are minimal. There are $2^5 = 32$ decision points associated with ADC1, and they occur every 68.75-mV ($2^9 = 512$ LSBs) for a 2.2-V fullscale input range. Figure 8 shows a greatly exaggerated representation of these nonlinearities.

The distortion components produced by the front end of the AD6645 up to about 200-MHz analog input are negligible compared to those produced by the encoder. That is, the static non-linearity of the AD6645 transfer function is the chief limitation to SFDR.

The goal is to select the proper amount of out-of-band dither so that the effect of these small DNL errors are randomized across the ADC input range, thereby reducing the average DNL error. Experimentally, it was determined that making the peak-to-peak dither noise cover about two ADC1 transitions gives the best improvement in DNL. The DNL is not significantly improved with higher levels of noise. Two ADC1 transitions cover 1024 LSBs peak-to-peak, or approximately 155 LSBs rms (peak-to-peak gaussian noise is converted to rms by dividing by 6.6).

The first plot shown in Figure 9 shows the undithered DNL over a small portion of the input signal range. The horizontal axis has
been expanded to show two of the subranging points which are spaced 68.75 mV (512 LSBs) apart. The second plot shows the DNL after adding 155 LSBs rms dither. This amount of dither corresponds to approximately \(-20.6 \text{ dBm}\). Note the dramatic improvement in the DNL.

![Figure 9: AD6645 Undithered and Dithered DNL](image)

Dither noise can be generated in a number of ways. Noise diodes can be used, but simply amplifying the input voltage noise of a wideband bipolar op amp provides a more economical solution. This approach has been described in detail (References 3, 4, and 5) and will not be repeated here.

The dramatic improvement in SFDR obtained with out-of-band dither is shown in Figure 10 using a deep (1,048,576-point) FFT, where the AD6645 is sampling a \(-35\text{-dBm}, 30.5\text{-MHz}\) signal at 80 MSPS. Note that the SFDR without dither is approximately 92 dBFS compared to 108 dBFS with dither, representing a 16-dB improvement!

![Figure 10: AD6645 Undithered and Dithered SFDR FFT Plot](image)

The AD6645 ADC was introduced by Analog Devices in 2000. Until recently, it represented the ultimate in SFDR performance. Since its introduction, improvements in both process technology and circuit design have resulted in even higher performance ADCs such as the AD9444 (14-bits @ 80 MSPS), AD9445 (14-bits @ 105/125 MSPS), and the AD9446 (16-bits @ 80/100 MSPS). These ADCs have very high SFDR (typically greater than 90 dBc for a 70-MHz full-scale input signal) and low DNL. Still, the addition of an appropriate out-of-band dither signal can improve the SFDR under certain input signal conditions.

Figure 11 shows the AD9444 (14-bits @ 80MSPS) FFT with and without dither. Note that under these input conditions, the
addition of dither improves the SFDR by 25 dB. The data was taken using the ADIsimADC program and the AD9444 model.

Figure 11: AD9444, 14-Bit, 80MSPS ADC

Even though the results shown in Figures 10 and 11 are fairly dramatic, it should not be assumed that the addition of out-of-band noise dither will always improve the SFDR of the ADC or under all conditions. As mentioned earlier, dither will not improve the linearity of the front end circuits of the ADC. Even with a nearly ideal front end, the effects of dither will be highly dependent upon the amplitude of the input signal as well as the amplitude of the dither signal itself. For example, when signals are near the fullscale input range of the ADC, the integral nonlinearity of the transfer function may become the limiting factor in determining SFDR, and dither will not help. In any event, the data sheet should be studied carefully—in some cases dithered and undithered data may be shown along with suggestions for the amplitude and bandwidth. Dither may be a built-in feature of newer IF sampling ADCs.

SUMMARY

In this discussion we have shown that all ADCs have some amount of input-referred noise. In precision, low frequency measurement applications, this noise can be reduced by digitally averaging the ADC output data at the expense of lower sampling rates and additional hardware. The resolution of the ADC can actually be increased by this averaging process, however integral nonlinearity errors are not improved. A small amount of input-referred noise is needed to increase the resolution by the averaging technique, however too much noise requires a large number of samples in the average, and a point of diminishing returns is reached.

In certain high speed ADC applications, the addition of the proper amount of out-of-band noise dither can improve the DNL of the ADC and increase the spurious free dynamic range (SFDR). However, the effectiveness of dither on the SFDR is highly dependent upon the characteristics of the particular ADC under consideration.

ACKNOWLEDGEMENTS

The author would like to thank Bonnie Baker of Microchip Technology and Alain Guery of Analog Devices for their thoughtful inputs to this article.

REFERENCES

INTRODUCTION

The concept of Noise Power Ratio (NPR) has been around since the early days of frequency division multiplexed (FDM) telephone systems. The NPR is simply a measure of the "quietness" of an unused channel in a multi-channel system when there is random activity on the others. Noise and intermodulation distortion products fall into the unused channel causing less than ideal performance. Originally used to check 4-kHz wide voice channels in FDM links, the same concept is useful today in characterizing multichannel wideband communication systems – but there are some important differences in the modern measurement techniques.

HISTORY OF NPR

Noise power ratio testing has been used since the early days of Frequency Division Multiplexed (FDM) communication systems. In a typical FDM system, 4-kHz wide voice channels are "stacked" in frequency bins for transmission over coaxial, microwave, or satellite equipment. The number of channels depends on the system. A group is composed of 12 voice channels and occupies a bandwidth of 48 kHz. Similarly, a supergroup has 60 channels and occupies a bandwidth of 240 kHz, and a mastergroup has 300 channels and occupies a bandwidth of approximately 1.3 MHz. Supergroups and mastergroups are often combined to make up even higher capacity systems. For instance, an 1800-channel system occupies a bandwidth of approximately 8 MHz.

At the receiving end of the transmission link, the FDM data is demultiplexed and converted back to 4-kHz individual voiceband channels. The FDM signal is therefore composed of many individual voice channels and passes through amplifiers, repeaters, channel banks, etc., which add noise and distortion to the signal. Early studies at Bell Telephone Labs (Reference 1) led to the conclusion that the composite signal in an FDM system having more than approximately 100 channels can be approximated by Gaussian noise having a bandwidth equal to the bandwidth of the combined FDM signal. For instance, a 1800-channel FDM signal is approximated by Gaussian noise with a bandwidth of 8.2 MHz.

The "quality" of an individual voice channel is then measured by first assuming that there are random "talkers" on all channels except the specific 4-kHz channel under test. An individual 4-kHz channel can therefore be measured for "quietness" using a narrow-band notch (bandstop) filter and a specially tuned receiver which measures the noise power inside the 4-kHz notch as shown in Figure 1A.
Noise Power Ratio (NPR) measurements are straightforward in an analog transmission system (Figure 1A). With the notch filter out, the rms noise power of the signal inside the notch is measured by the narrowband receiver. The notch filter is then switched in, and the residual noise inside the notch is measured. The ratio of these two readings expressed in dB is the NPR. Several notch frequencies across the noise bandwidth (low, midband, and high) are tested to characterize the system adequately. Details of early NPR test equipment and the measurements can be found in Reference 4. NPR measurements on ADCs are made in a similar manner, except the analog receiver is replaced by a buffer memory and an FFT processor which performs the calculations as shown in Figure 1B. There are some cases where the combined FDM signal is converted to digital with an ADC, transmitted, and then converted back to analog using a DAC at the receiver. In this case, the analog method shown in Figure 1A is utilized in performing the NPR test.

In a 1939 article (Reference 1), Holbrook and Dixon performed an analysis of FDM systems in an effort to determine the optimum channel "loading" levels. Their work led to the fundamental theory of multichannel noise loading. The goal is to set the signal level (or "loading") to a value which will give the highest NPR. The NPR is plotted as a function of rms noise level referred to the peak range of the system. For very low noise loading levels, the undesired noise (in non-digital systems) is primarily thermal noise and is independent of the input noise level. Over this region of the curve, a 1-dB increase in noise loading level causes a 1-dB increase in NPR. As the noise loading level is increased, the amplifiers and repeaters in the system begin to overload, creating intermodulation products which cause the noise floor of the system to increase. As the input noise continues to increase, the effects of "overload" noise predominate, and the NPR is reduced dramatically. FDM systems are usually operated at a noise loading level a few dB below the point of maximum NPR to allow headroom for peak busy hours.

System NPR recommendations for FDM systems were formalized in 1966 by the CCITT/CCIR to measure the transmission characteristics of Frequency Division Multiple (FDM) communications links (see Reference 4).

In a digital system containing an ADC, the noise within the notch is primarily quantization noise when low levels of input noise are applied. However, for very low amplitude signals (less than 1-LSB peak-to-peak), the resulting noise reverts to the input-referred noise of the ADC. For signals that exercise several LSBs of the ADC, the NPR curve is linear, and quantization noise predominates. As the noise level increases, there is a one-for-one correspondence between the noise level and the NPR. At some level, however, "clipping" noise caused by the hard-limiting action of the ADC begins to dominate.

The ADC hard-limiting "clipping" noise is somewhat different from the soft-limiting "overload" noise of an analog FDM and results in a "steeper" downward slope in the clipping region.
THEORETICAL NPR FOR A DIGITAL SYSTEM

Several papers have been written over the years deriving the theoretical NPR of an ideal n-bit ADC (for example, References 5, 6, and 7). Reference 6 is the most complete, and shows the derivation for both uniformly distributed noise and Gaussian noise. However, Gaussian noise is much more relevant to NPR testing. The derivation is not difficult but does involve some partial integration. Since the “clipping” noise component not have a closed-form solution, numerical methods must be used to actually compute the theoretical NPR numbers.

A theoretical curve for 10, 12, 14, and 16-bit ADCs is shown in Figure 2. Understanding the definitions of the terms \( V_0 \), \( \sigma \), \( k \), and the rms loading level \( -20\log_{10}(k) \) are very important in order to avoid confusion.

It is important to understand that these curves are based on an ideal ADC where the only noise is the quantization noise and the clipping noise. In practice, the actual level of performance will be less than theoretical, depending upon the particular ADC under test.

![Image showing the theoretical NPR for 10, 12, 14 and 16-bit ADCs](image)

**Figure 2: Theoretical NPR for 10, 12, 14 and 16-bit ADCs**

The ADC input range is bipolar, and is \( \pm V_0 \) full-scale (hence \( 2V_0 \) peak-to-peak). The input rms noise level is \( \sigma \), and the noise-loading factor \( k \) (also called the crest factor) is defined as \( V_0/\sigma \). The value of \( k \) is therefore the peak-signal-to-rms-noise-ratio, where \( k \) is expressed as a numerical ratio. Again, it is important to note that a peak signal of \( V_0 \) implies a peak-to-peak full-scale input of \( 2V_0 \). This can become a point of confusion. Another way to put it is that a full-scale sinewave given by \( v(t) = V_0 \cdot \sin \omega t \) exactly fills the ADC input range. This is why \( V_0 \) is referred to as the peak amplitude.

The reciprocal of \( k \) is the rms-noise-to-peak-signal-ratio, and the rms noise loading level is defined as \( 1/k \) expressed in dB:

\[
\text{RMS Noise Loading Level} = 20\log_{10} \left( \frac{1}{k} \right) = -20\log(k). \tag{Eq. 1}
\]

The derivation for theoretical NPR can be broken into two parts. The first part derives the theoretical quantization noise power of an ideal n-bit ADC. The second part derives the “clipping noise” power due to the limiting action of the ADC. The total noise power is the sum of the two noise powers. The complete error waveform showing the two regions is shown in Figure 3.
The theory is based on several assumptions. First, the quantization error signal is not correlated to the input signal. This is valid provided the signal amplitude is at least several LSBs in amplitude and the resolution of the ADC is at least 6-bits. Second, the sampling rate is twice the input noise bandwidth. Third, the ADC acts as an ideal limiter for out-of-range signals. These three assumptions are valid for most practical systems and lead to a relatively straightforward solution.

The quantization noise component (expressed as the square of the actual quantization noise voltage to yield noise power), has been shown to be (see Reference 2, for example):

\[ N_Q = \frac{q^2}{12} \]  

Eq. 2

where \( q \) is the weight of the least significant bit (LSB). It should be noted that this is the quantization noise power measured over the full Nyquist bandwidth dc to \( f_s/2 \). If the signal bandwidth is reduced, the noise in the reduced bandwidth is proportionally less, and a correction factor must be added (discussed later in this paper).

Continuing with the derivation we know that, \( q = 2V_O/2^n \). Therefore, from Equation 2:

\[ N_Q = \frac{q^2}{12} = \left( \frac{2V_O}{2^n} \right)^2 = \frac{V_O^2}{3 \cdot 2^{2n}}. \]  

Eq. 3

However, \( k = V_O/\sigma \), therefore \( V_O = k\sigma \), and substituting for \( V_O \) in Equation 3 yields:

\[ N_Q = \frac{k^2 \sigma^2}{3 \cdot 2^{2n}}. \]  

Eq. 4

Now, refer to Figure 3 for the derivation of the clipping noise power, \( N_C \).

The clipping noise power is given by the following general equation:
From Figure 3B,

\[ e(x) = x - V_O, \quad \text{for } x > V_O, \quad \text{and therefore} \]

\[ N_C = 2 \int_{V_O}^{\infty} (x - V_O)^2 P(x) \, dx, \]  

Eq. 7

where \( P(x) \) is the Gaussian probability density function and is given by:

\[ P(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-x^2 / 2\sigma^2}. \]  

Eq. 8

Substituting \( V_O = k\sigma \), and combining Equation 8 with Equation 7 yields:

\[ N_C = 2 \int_{k\sigma}^{\infty} (x - k\sigma)^2 \frac{1}{\sigma \sqrt{2\pi}} e^{-x^2 / 2\sigma^2} \, dx \]  

Eq. 9

The final results of the integration (see Appendix for complete derivation) yields:

\[ N_C = 2\sigma^2 (k^2 + 1)[1 - N(k)] - k\sigma^2 \sqrt{\frac{2}{\pi}} e^{-k^2 / 2} \]  

Eq. 10

Where \( N(k) \) is the Normal Distribution Function:

\[ N(k) = \int_{-\infty}^{k} \frac{1}{\sqrt{2\pi}} e^{-t^2 / 2} \, dt. \]  

Eq. 11

For calculation purposes, the function \([1 - N(k)]\) can be approximated by the following expression:

\[ 1 - N(k) \approx \frac{1}{k\sqrt{2\pi}} e^{-k^2 / 2} \left[ 1 - \frac{1}{k^2 + 2} + \frac{1}{(k^2 + 2)(k^2 + 4)} - \frac{5}{(k^2 + 2)(k^2 + 4)(k^2 + 6)} \right. \]

\[ + \frac{9}{(k^2 + 2)(k^2 + 4)(k^2 + 6)(k^2 + 8)} - \frac{129}{(k^2 + 2)(k^2 + 4)(k^2 + 6)(k^2 + 8)(k^2 + 10)} \]  

Eq. 12

The total noise, \( N_T \), can now be calculated by adding Equation 4 and Equation 10:
Figure 4: Theoretical Maximum NPR for 8 to 20-bit ADCs

\[ N_T = N_Q + N_C = \frac{k^2 \sigma^2}{3 \cdot 2^{2n}} + 2\sigma^2 (k^2 + 1)[1 - N(k)] - k \sigma^2 \sqrt{\frac{2}{\pi}} e^{-k^2/2} , \quad \text{Eq. 13} \]

\[ \frac{N_T}{\sigma^2} = \frac{k^2}{3 \cdot 2^{2n}} + 2(k^2 + 1)[1 - N(k)] - k \sqrt{\frac{2}{\pi}} e^{-k^2/2} . \quad \text{Eq. 14} \]

\[ \text{NPR} = 10 \log \left( \frac{\sigma^2}{N_T} \right) = -10 \log \left( \frac{N_T}{\sigma^2} \right) \quad \text{Eq. 15} \]

Figure 4 shows the theoretical peak value of NPR and the corresponding value of k for ADCs having resolutions between 8 and 20 bits. The vertical axis is NPR (expressed in dB per Equation 15). The horizontal axis is the Gaussian noise loading level with respect to the peak signal level, \(\sigma/V_O\), expressed in dB.

<table>
<thead>
<tr>
<th>BITS</th>
<th>k OPTIMUM</th>
<th>k(dB)</th>
<th>MAX NPR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3.92</td>
<td>11.87</td>
<td>40.60</td>
</tr>
<tr>
<td>9</td>
<td>4.22</td>
<td>12.50</td>
<td>46.05</td>
</tr>
<tr>
<td>10</td>
<td>4.50</td>
<td>13.06</td>
<td>51.56</td>
</tr>
<tr>
<td>11</td>
<td>4.76</td>
<td>13.55</td>
<td>57.12</td>
</tr>
<tr>
<td>12</td>
<td>5.01</td>
<td>14.00</td>
<td>62.71</td>
</tr>
<tr>
<td>13</td>
<td>5.26</td>
<td>14.41</td>
<td>68.35</td>
</tr>
<tr>
<td>14</td>
<td>5.49</td>
<td>14.79</td>
<td>74.01</td>
</tr>
<tr>
<td>15</td>
<td>5.72</td>
<td>15.15</td>
<td>79.70</td>
</tr>
<tr>
<td>16</td>
<td>5.94</td>
<td>15.47</td>
<td>85.40</td>
</tr>
<tr>
<td>18</td>
<td>6.34</td>
<td>16.04</td>
<td>96.88</td>
</tr>
<tr>
<td>20</td>
<td>6.78</td>
<td>16.62</td>
<td>108.41</td>
</tr>
</tbody>
</table>

ADC Range = \( \pm V_0 \)
\[ k = V_0 / \sigma \]
\[ \sigma = \text{RMS Noise Level} \]

Figure 4: Theoretical Maximum NPR for 8 to 20-bit ADCs

Again it is important to remember that this is the NPR obtained when the input signal noise occupies the full Nyquist bandwidth, dc to \(f_s/2\). For the case of oversampling, where the signal bandwidth, BW, is less than \(f_s/2\), the correction factor of \(10 \log_{10}[f_s/(2\cdot BW)]\), often referred to as process gain, must be added to the NPR given in Equation 15:

\[ \text{NPR} = 10 \log \left( \frac{\sigma^2}{N_T} \right) + 10 \log \left( \frac{f_s}{2 \cdot BW} \right) \quad \text{Eq. 16} \]

In multi-channel high frequency communication systems, where there is little or no phase correlation between channels, NPR can be used to measure the distortion and noise caused by a large number of individual channels, similar to an FDM system. A notch filter is placed between the noise source and the ADC, and an FFT output is used in place of an analog receiver. The width of the notch filter is set for about 500 kHz to 2 MHz as shown in Figure 5 for the AD9229 12-bit 65-MSPS ADC. The sampling rate is 65 MSPS, the notch is centered at 18 MHz, and the NPR is the “depth” of the notch. An ideal ADC will only generate the theoretical value of quantization noise, however a practical one has additional noise components due to additional noise and
intermodulation distortion caused by ADC imperfections. Notice that the NPR is about 60.8 dB compared to 62.7-dB theoretical.

Making NPR measurements digitally requires that the FFT have a sufficient number of samples such that there are at least 25 to 50 samples within the filter notch. There are obviously tradeoffs between the width of the notch and the FFT size. However, the notch width should not be wider than about 10% of the noise bandwidth, or the test results may not be valid.

In the example shown in Figure 5 for the AD9229, the FFT size was 16,384 which gives a frequency resolution of 65 MSPS/16,384 = 3.97 kHz. Since the notch filter width is approximately 1 MHz at the bottom of the notch, approximately 250 samples fall within the notch. Due to the specific requirements regarding the center frequency, width, and band-stop rejection, custom-made notch filters are generally required in order to implement NPR tests on ADCs. Achieving good results is difficult using just a simple filter and wideband noise source. Wideband Gaussian noise generators, such as the NoiseCom DNG7500, are available that allow the user to custom shape the noise according to their application. Using a combination of a Gaussian noise shaping generator and notch filter makes this test easier to implement. The results of several FFTs must then be averaged in order to reduce the variation in the NPR results from run to run since there are only a limited number of samples which fall inside the notch itself. The data shown in Figure 5 represents the average of the NPR results for 5 individual FFT runs.

NPR should be measured at several different frequencies across the noise bandwidth, thereby requiring several notch filters. Some degradation will occur at the higher frequencies – very similar to the degradation in other ADC ac specifications such as SNR and SFDR.

![Figure 5: AD9229 12-bit, 65-MSPS ADC NPR Measures 60.8 dB (62.7 dB Theoretical)](image)

**SUMMARY**

We have shown how NPR is used in standard FDM systems to characterize the noise and intermodulation distortion of multi-channel system where the voice channel width is 4 kHz. It can also be used to determine the optimum signal level to give maximum dynamic range. This 65-year old concept is still useful today in modern multichannel wireless systems. Bandwidths and channel spacings are higher, but the same concepts still apply. In many cases, NPR is a good approximation to complicated multi-tone testing and embodies the specific features of many applications when testing your system's dynamic range (Reference 7).

Although the single tone or two-tone sinewave signal is by far the most popular method for testing ADCs for wideband applications, NPR testing offers a relatively easy method using a Gaussian noise input to simulate a broadband multitone signal without the need for generating a large number of single tone sinewaves.

***********************

**REFERENCES**
APPENDIX

In this appendix, we will show how to evaluate the following integral from Equation 9.

\[ N_C = 2 \int_{k\sigma}^{\infty} \frac{(x-k\sigma)^2}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \, dx. \quad \text{Eq. A1} \]

This integral is of the form:

\[ N_C = C \int_{A}^{\infty} (x-A)^2 e^{-Bx^2} \, dx \quad \text{Eq. A2} \]

where

\[ A = k\sigma, \quad B = \frac{1}{2\sigma^2}, \quad C = \frac{\sqrt{2}}{\sigma\sqrt{\pi}}. \quad \text{Eq. A3} \]

Integrating Eq. A2:

\[ N_C = C \int_{A}^{\infty} (x-A)^2 e^{-Bx^2} \, dx = C \int_{A}^{\infty} x^2 e^{-Bx^2} \, dx - 2AC \int_{A}^{\infty} xe^{-Bx^2} \, dx + CA^2 \int_{A}^{\infty} e^{-Bx^2} \, dx. \quad \text{Eq. A4} \]

Now, evaluate the first integral using partial integration:

\[ \int_{A}^{\infty} x^2 e^{-Bx^2} \, dx = - \frac{1}{2B} \int_{A}^{\infty} x(-2Bx) e^{-Bx^2} \, dx. \quad \text{Eq. A5} \]

The fundamental equation for partial integration is:
\[ \int uv = uv - \int vdu. \quad \text{Eq. A6} \]

Let \( u = x \) and \( dv = -2Bxe^{-Bx^2} \, dx \).

Then

\[
\int_{-\infty}^{\infty} x e^{-Bx^2} \, dx = -\frac{1}{2B} \left\{ xe^{-Bx^2}\right|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} e^{-Bx^2} \, dx \} \quad \text{Eq. A7}
\]

\[
= \frac{A}{2B} e^{-BA^2} + \frac{1}{2B} \int_{A}^{\infty} e^{-Bx^2} \, dx \quad \text{Eq. A8}
\]

Evaluating the second integral in Equation A4:

\[
\int_{A}^{\infty} e^{-Bx^2} \, dx = -\frac{1}{2B} \int_{A}^{\infty} (-2Bx) \, dx = -\frac{1}{2B} e^{-Bx^2}\right|_{A}^{\infty} = \frac{1}{2B} e^{-BA^2} \quad \text{Eq. A9}
\]

Substituting Equation A8 and A9 into Equation A4:

\[
N_C = C \int_{A}^{\infty} (x - A)^2 e^{-Bx^2} \, dx
\]

\[
= \frac{CA}{2B} e^{-BA^2} + \frac{C}{2B} \int_{A}^{\infty} e^{-Bx^2} \, dx - \frac{2AC}{2B} e^{-BA^2} + A^2 C \int_{A}^{\infty} e^{-Bx^2} \, dx
\]

\[
= \left( \frac{AC}{2B} - \frac{2AC}{2B} \right) e^{-BA^2} + \left( \frac{C}{2B} + A^2 C \right) \int_{A}^{\infty} e^{-Bx^2} \, dx
\]

\[
= -\frac{AC}{2B} e^{-BA^2} + C \left( \frac{1}{2B} + A^2 \right) \int_{A}^{\infty} e^{-Bx^2} \, dx \quad \text{Eq. A10}
\]

Now from Eq. A3, substituting \( A = k\sigma \), \( B = \frac{1}{2\sigma^2} \), and \( C = \frac{\sqrt{2}}{\sigma \sqrt{\pi}} \) into Eq. A10:

\[
N_C = -k\sigma^2 \sqrt{\frac{2}{\pi}} e^{-k^2/2} + \frac{2\sigma}{\sqrt{2\pi}} \left( 1 + k^2 \right) \int_{k\sigma}^{\infty} e^{-x^2/2\sigma^2} \, dx \quad \text{Eq. A11}
\]

Define \( t = \frac{x}{\sigma} \), \( x = t \sigma \), \( dx = \sigma \, dt \).
Then substituting into Equation A11 and rearranging yields:

\[ N_C = 2\sigma^2 (1 + k^2) \left[ 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{k} e^{-t^2/2} \, dt \right] - k\sigma^2 \sqrt{\frac{2}{\pi}} e^{-k^2/2} \quad \text{Eq. A13} \]

\[ N_C = 2\sigma^2 (1 + k^2) \left[ 1 - N(k) \right] - k\sigma^2 \sqrt{\frac{2}{\pi}} e^{-k^2/2} \quad \text{Eq. A14} \]

Where \[ N(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{k} e^{-t^2/2} \, dt \], the Normal Distribution Function \quad \text{Eq. A15}
INTRODUCTION

Noise figure (NF) is a popular specification among RF system designers. It is used to characterize the noise of RF amplifiers, mixers, etc., and widely used as a tool in radio receiver design. Many excellent textbooks on communications and receiver design treat noise figure extensively (see Reference 1, for example) – it is the purpose of this discussion to focus on how the specification applies to data converters.

Since many wideband operational amplifiers and ADCs are now being used in RF applications, the day has come where the noise figure of these devices becomes important. Reference 2 discusses the proper methods for determining the noise figure of an op amp. One must not only know the op amp voltage and current noise, but the exact circuit conditions—closed-loop gain, gain-setting resistor values, source resistance, bandwidth, etc. Calculating the noise figure for an ADC is even more of a challenge, as will be seen shortly.

When an RF engineer first calculates the noise figure of even the best low-noise high-speed ADC, the result may appear relatively high compared to the noise figure of typical RF gain blocks, low-noise amplifiers, etc. An understanding of where the ADC is positioned in the signal chain is needed to properly interpret the results. A certain amount of caution is therefore needed when dealing with ADC noise figures.

ADC NOISE FIGURE DEFINITION

Figure 1 shows the basic model for defining the noise figure of an ADC. The noise factor, $F$, is simply defined as the ratio of the total effective input noise power of the ADC to the amount of that noise power caused by the source resistance alone. Because the impedance is matched, the square of the voltage noise can be used instead of noise power. The noise figure, $NF$, is simply the noise factor expressed in dB, $NF = 10 \log_{10} F$.
This model assumes the input to the ADC comes from a source having a resistance, \( R \), and that the input is band-limited to \( f_s/2 \) with a filter having a noise bandwidth equal to \( f_s/2 \). It is also possible to further band-limit the input signal which results in oversampling and process gain — this condition will be discussed shortly.

It is also assumed that the input impedance to the ADC is equal to the source resistance. Many ADCs have a high input impedance, so this termination resistance may be external to the ADC or used in parallel with the internal resistance to produce an equivalent termination resistance equal to \( R \).

**ADC NOISE FIGURE DERIVATION**

The full-scale input power is the power of a sinewave whose peak-to-peak amplitude exactly fills the ADC input range. The full-scale input sinewave given by the following equation has a peak-to-peak amplitude of \( 2V_o \), corresponding to the peak-to-peak input range of the ADC:

\[
v(t) = V_o \sin 2\pi ft
\]

The full scale power of this sinewave is given by:

\[
P_{FS} = \frac{(V_o / \sqrt{2})^2}{R} - \frac{V_o^2}{2R}
\]

It is customary to express this power in dBm (referenced to 1mW) as follows:

\[
P_{FS(dBm)} = 10 \log_{10} \left[ \frac{P_{FS}}{1 \text{mW}} \right]
\]

Some more discussion is required regarding the noise bandwidth of filter, \( B \). The *noise bandwidth* of a non-ideal brick wall filter is defined as the bandwidth of an ideal brick wall filter which will pass the same noise power as the non-ideal filter. Therefore, the noise bandwidth of a filter is always greater than the 3-dB bandwidth of the filter by a factor which depends upon the sharpness of the cutoff region of the filter. Figure 2 shows the relationship between the noise bandwidth and the 3-dB bandwidth for Butterworth filters up to 5 poles. Note that for two poles, the noise bandwidth and 3-dB bandwidth are within 11% of each other, and beyond that the two quantities are essentially equal.

<table>
<thead>
<tr>
<th>NUMBER OF POLES</th>
<th>NOISE BW : 3dB BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.57</td>
</tr>
<tr>
<td>2</td>
<td>1.11</td>
</tr>
<tr>
<td>3</td>
<td>1.05</td>
</tr>
<tr>
<td>4</td>
<td>1.03</td>
</tr>
<tr>
<td>5</td>
<td>1.02</td>
</tr>
</tbody>
</table>

**Figure 2: Relationship Between Noise Bandwidth and 3-dB Bandwidth for a Butterworth Filter**

The first step in the NF calculation is to calculate the effective input noise of the ADC from its SNR. The SNR of the ADC is shown on the data sheet for a variety of input frequencies, so be sure and use the value corresponding to the IF input frequency.
of interest. Also, make sure that the harmonics of the fundamental signal are not included in the SNR number – some ADC data sheets may confuse SINAD with SNR. Once the SNR is known, the equivalent input rms voltage noise can be calculated starting from the equation:

\[
\text{SNR} = 20 \log_{10} \left( \frac{V_{\text{FS RMS}}}{V_{\text{NOISE RMS}}} \right) \quad \text{Eq. 4}
\]

Solving for \( V_{\text{NOISE RMS}} \):

\[
V_{\text{NOISE RMS}} = V_{\text{FS RMS}} \cdot 10^{-\text{SNR} / 20} \quad \text{Eq. 5}
\]

This is the total effective input rms noise voltage measured over the Nyquist bandwidth, dc to \( f_s/2 \). Note that this noise includes the source resistance noise.

The next step is to actually calculate the noise figure. In Figure 3 notice that the amount of the input voltage noise due to the source resistance is the voltage noise of the source resistance \( \sqrt{4kTBR} \) divided by two, or \( \sqrt{(kTBR)} \), because of the 2:1 attenuator formed by the ADC input termination resistor.

The expression for the noise factor \( F \) can be written

\[
F = \frac{V_{\text{NOISE RMS}}^2}{kTRB} = \left[ \frac{V_{\text{FS RMS}}^2}{R} \right] \left[ \frac{1}{kT} \right] \left[ 10^{-\text{SNR} / 10} \right] \left[ \frac{1}{B} \right] \quad \text{Eq. 6}
\]

The noise figure is obtained by converting \( F \) into \( \text{dB} \) and simplifying:

\[
\text{NF} = 10 \log_{10} F = P_{FS(dBm)} + 174 \text{dBm} - \text{SNR} - 10 \log_{10} B, \quad \text{Eq. 7}
\]

Where \( \text{SNR} \) is in \( \text{dB} \), \( B \) in Hz, \( T = 300 \text{K} \), \( k = 1.38 \times 10^{-23} \text{J/K} \).
Figure 3: ADC Noise Figure in Terms of SNR, Sampling Rate, and Input Power

Oversampling and digital filtering can be used to decrease the noise figure as a result of the process gain as has been previously discussed. In the case of oversampling, the signal bandwidth $B$ is less than $f_s/2$. Figure 4 shows the correction factor which results in the following equation:

$$NF = 10 \log_2 \left( \frac{P_{\text{out}}(\text{dBm})}{P_{\text{in}}(\text{dBm})} \right) + 174 \text{ dBm} - \text{SNR} - 10 \log_{10} \left( \frac{f_s}{2B} \right) - 10 \log_{10} \left( \frac{f_s}{2} \right).$$  \text{Eq. 8}

![Figure 4: Effect of Oversampling and Process Gain on ADC Noise Figure]

**EXAMPLE CALCULATION FOR AD9446 16-BIT 80/100-MSPS ADC**

Figure 5 shows an example NF calculation for the AD9446 16-bit, 80/105-MSPS ADC. A 52.3-Ω resistor is added in parallel with the AD9446 input impedance of 1 kΩ to make the net input impedance 50 Ω. The ADC is operating under Nyquist conditions, and the SNR of 82 dB is the starting point for the calculations using Eq. 8 above. A noise figure of 30.1 dB is obtained.
**USING RF TRANSFORMERS TO INCREASE ADC NOISE FIGURE**

Figure 6 shows how using an RF transformer with voltage gain can improve the noise figure. Figure 6A shows a 1:1 turns ratio, and the noise figure (from Figure 5) is 30.1 dB. Figure 6B shows a transformer with a 1:2 turns ratio. The 249-Ω resistor in parallel with the AD9446 internal resistance results in a net input impedance of 200 Ω. The noise figure is improved by 6 dB because of the "noise-free" voltage gain of the transformer. Figure 6C shows a transformer with a 1:4 turns ratio. The AD9446 input is paralleled with a 4.02-kΩ resistor to make the net input impedance 800 Ω. The noise figure is improved by another 6 dB.

In theory, even higher turns ratios will yield further improvement, but transformers with higher turns ratios are not generally practical because of bandwidth and distortion limitations.

**Figure 5: Example Calculation of Noise Figure Under Nyquist Conditions for AD9446 16-Bit, 80/100 MSPS ADC**

**Figure 6: Using RF Transformers to Improve Overall ADC Noise Figure**
CASCADED NOISE FIGURE

Even with the 1:4 turns ratio transformer, the overall noise figure for the AD9446 is 18.1 dB, still relatively high by RF standards. It should be noted that the 82-dB SNR of the AD9446 ADC represents excellent noise performance, and the "solution" for system applications is to provide low-noise high-gain stages ahead of the ADC. In a typical receiver there will be at least one low noise amplifier (LNA) and mixing stage ahead of the ADC which provides sufficient signal gain to minimize the degradation in overall system noise figure due to the ADC.

This can be explained in Figure 7 which shows how the Friis equation is used to calculate the noise factor for cascaded gain stages. Notice that high gain in the first stage reduces the contribution of the noise factor of the second stage – the noise factor of the first stage dominates the overall noise factor.

![Figure 7: Cascade Noise Figure Using the Friis Equation](image)

Figure 8 shows the effects of a high-gain (25 dB) low-noise (NF = 4 dB) stage placed in front of a relatively high NF stage (30 dB) – the noise figure of the second stage is typical of high performance ADCs. The overall noise figure is 7.53 dB, only 3.53 dB higher than the first stage noise figure of 4 dB.
SUMMARY

Applying the noise figure concept to characterize wideband ADCs must be done with extreme caution to prevent misleading results. Simply trying to minimize the noise figure by manipulating the values in the equations can actually increase the total circuit noise.

For instance, NF decreases with increasing source resistance according to the equations, but increased source resistance increases circuit noise. Another example relates to the ADC input bandwidth $B$. Increasing $B$ reduces NF according to the equations; however, this is contradictory, because increasing the ADC input bandwidth actually increases the effective input noise. In both these examples, the total circuit noise increases, but the NF decreases.

The reason NF decreases is because when either the source resistance or bandwidth increases, the source noise simply makes up a larger component of the total noise. However, the total noise remains relatively constant because the noise due to the ADC is much greater than the source noise; therefore according to the equations, NF decreases, but actual circuit noise increases.

For these reasons, NF must be used with some caution when dealing with ADCs. The equations in this article will give valid results, but can be misleading without a full understanding of the noise principles involved.

It is true that on a stand-alone basis even low-noise ADCs have relatively high noise figures compared to other RF parts such as LNAs or mixers. In an actual system application, however, the ADC is always preceded by at least one low-noise gain block which reduces the overall ADC noise contribution to a very small level per the Friis equation (see Figure 8).

REFERENCES

MT-007: Aperture Time, Aperture Jitter, Aperture Delay Time – Removing the Confusion

by Walt Kester

INTRODUCTION

Perhaps the most misunderstood and misused ADC and sample-and-hold (or track-and-hold) specifications are those that include the word aperture. A simple model is shown in Figure 1, and the most essential dynamic property of a SHA is its ability to disconnect quickly the hold capacitor from the input buffer amplifier. Historically, the short (but non-zero) interval required for this action is called aperture time (or sampling aperture), \( t_a \). The actual value of the voltage that is held at the end of this interval is a function of both the input signal slew rate and the errors introduced by the switching operation itself. Figure 1 shows what happens when the hold command is applied with an input signal of two arbitrary slopes labeled as 1 and 2. For clarity, the sample-to-hold pedestal and switching transients are ignored. The value that is finally held is a delayed version of the input signal, averaged over the aperture time of the switch. The first-order model assumes that the final value of the voltage on the hold capacitor is approximately equal to the average value of the signal applied to the switch over the interval during which the switch changes from a low to high impedance (\( t_a \)).

The model shows that the finite time required for the switch to open (\( t_a \)) is equivalent to introducing a small delay \( t_e \) in the sampling clock driving the SHA. This delay is constant, and can be either positive or negative. The diagram shows that the same value of \( t_e \) works for the two signals, even though the slopes are different. This delay is called effective aperture delay time, aperture delay time, or simply aperture delay, \( t_e \). In an ADC, the aperture delay time is referenced to the input of the converter, and the effects of the analog propagation delay through the input buffer, \( t_{da} \), and the digital delay through the switch driver, \( t_{dd} \), must be considered. Referenced to the ADC inputs, aperture time, \( t_e' \), is defined as the time difference between the analog propagation delay of the front-end buffer, \( t_{da} \), and the switch driver digital delay, \( t_{dd} \), plus one-half the aperture time, \( t_a/2 \).

The effective aperture delay time is usually positive, but may be negative if the sum of one-half the aperture time, \( t_a/2 \), and the switch driver digital delay, \( t_{dd} \), is less than the propagation delay through the input buffer, \( t_{da} \). The aperture delay specification thus establishes when the input signal is actually sampled with respect to the sampling clock edge.
Aperture delay time can be measured by applying a bipolar sinewave signal to the ADC and adjusting the synchronous sampling clock delay such that the output of the ADC is mid-scale (corresponding to the zero-crossing of the sinewave). The relative delay between the input sampling clock edge and the actual zero-crossing of the input sinewave is the aperture delay time as shown in Figure 2.

![Figure 2: Effective Aperture Delay Time Measured with Respect to ADC Input](image)

Aperture delay produces no errors (assuming it is relatively short with respect to the hold time), but acts as a fixed delay in either the sampling clock input or the analog input (depending on its sign). However, in "interleaved" ADCs, simultaneous sampling applications, or in direct I/Q demodulation, where two or more ADCs must be well matched; variations in the aperture delay between converters can produce errors on fast slewing signals. In these applications, the aperture delay mismatches must be removed by properly adjusting the phases of the individual sampling clocks to the various ADCs.

If, however, there is sample-to-sample variation in aperture delay (aperture jitter), then a corresponding voltage error is produced as shown in Figure 3. This sample-to-sample variation in the instant the switch opens is called aperture uncertainty, or aperture jitter and is usually measured in rms picoseconds. The amplitude of the associated output error is related to the rate-of-change of the analog input. For any given value of aperture jitter, the aperture jitter error increases as the input dv/dt increases. The effects of phase jitter on the external sampling clock (or the analog input for that matter) produce exactly the same type of error. For this reason, the total amount of jitter is the root-sum-square of the external sampling clock jitter and the ADC aperture jitter.
EFFECT OF APERTURE JITTER AND SAMPLING CLOCK JITTER ON ADC SIGNAL-TO-NOISE RATIO (SNR)

The effects of aperture and sampling clock jitter on an ideal ADCs SNR can be predicted by the following simple analysis.

Assume an input signal given by

\[ v(t) = V_O \sin 2\pi ft. \quad \text{Eq. 1} \]

The rate of change of this signal is given by:

\[ \frac{dv}{dt} = 2\pi f V_O \cos 2\pi ft. \quad \text{Eq. 2} \]

The rms value of \( \frac{dv}{dt} \) can be obtained by dividing the amplitude \( 2\pi fV_O \) by \( \sqrt{2} \):

\[ \frac{dv}{dt}_{\text{rms}} = \frac{2\pi V_O}{\sqrt{2}}. \quad \text{Eq. 3} \]

Now let \( \Delta v_{\text{rms}} \) = the rms voltage error and \( \Delta t \) = the rms aperture jitter \( t_j \) and substitute these values into Eq. 3:

\[ \frac{\Delta v_{\text{rms}}}{t_j} = \frac{2\pi V_O}{\sqrt{2}}. \quad \text{Eq. 4} \]

Solving Eq. 4 for \( \Delta v_{\text{rms}} \):

\[ \Delta v_{\text{rms}} = \frac{2\pi V_O t_j}{\sqrt{2}}. \quad \text{Eq. 5} \]

The rms value of the full-scale input sinewave is \( V_O \sqrt{2} \), therefore the rms signal to rms noise ratio (expressed in dB) is given by

\[ 20 \log \left( \frac{V_O \sqrt{2}}{\frac{2\pi V_O t_j}{\sqrt{2}}} \right) \]

For small values of \( t_j \):

\[ 20 \log \left( \frac{V_O \sqrt{2}}{\frac{2\pi V_O t_j}{\sqrt{2}}} \right) \approx 20 \log \left( \frac{V_O \sqrt{2}}{\frac{2\pi V_O \cdot 0.1}{\sqrt{2}}} \right) \approx 20 \log \left( \frac{V_O \sqrt{2}}{0.636 V_O} \right) \approx 20 \log(1.59) \approx 20 \times 0.176 \approx 3.52 \text{ dB} \]
This equation assumes an infinite resolution ADC where aperture jitter is the only factor in determining the SNR. This equation is plotted in Figure 4 and shows the serious effects of aperture and sampling clock jitter on SNR and ENOB, especially at higher input/output frequencies. For instance, in order to achieve 14-bit SNR performance when sampling a 100-MHz IF signal, the aperture jitter must be less than 0.1 ps. ADCs are currently available with typical aperture jitter specifications of 60-fs rms (AD9445 14-bits @ 125 MSPS and AD9446 16-bits @ 100 MSPS). Extreme care must be taken to minimize phase noise in the sampling/reconstruction clock so as not to degrade the inherent performance of the ADC itself.

![Figure 4: Theoretical Data Converter SNR and ENOB Due to Jitter vs. Fullscale Sinewave Input Frequency](image)

This care must extend to all aspects of the clock signal: the oscillator itself (for example, a 555 timer is absolutely inadequate, but even a quartz crystal oscillator can give problems if it uses an active device which shares a chip with noisy logic); the transmission path (these clocks are very vulnerable to interference of all sorts), and phase noise introduced in the ADC or DAC. As discussed, a very common source of phase noise in converter circuitry is aperture jitter in the integral sample and hold (SHA) circuitry, however the total rms jitter will be composed of a number of components—the actual SHA aperture jitter often being the least of them.

Before the 1980s, most sampling ADCs were generally built from a separate SHA and ADC. Interface design was complex, and accurately predicting the performance of the combination was difficult. Today, almost all sampled data systems use sampling ADCs which contain an integral SHA. The aperture jitter of the SHA may not be specified as such, but this is not a cause of concern if the SNR or ENOB is clearly specified over frequency, since a guarantee of a specific SNR at a specific input frequency is an implicit guarantee of an adequate aperture jitter specification.

**MEASURING ADC APERTURE JITTER USING FFT TECHNIQUES**

The FFT test routine for measuring ADC SNR, SFDR, etc., is an excellent indirect method for measuring aperture jitter. The caveat in this test is that the measurement includes the jitter of the sampling clock generator as well as the ADC internal aperture jitter. Therefore, a generator should be selected with an rms jitter specification which is several times less than the specified aperture jitter of the ADC under test, since individual jitter components combine on an rss basis. The basic test setup for the
Aperture jitter test is shown in Figure 5 along with the key calculations.

Figure 5: Measuring Aperture Jitter Based on Degradation in SNR at High Frequencies

There are two SNR measurements required, and both utilize a full-scale input sinewave having a frequency $f_L$ and $f_H$. The first measurement, SNRL, is made at a relatively low frequency, $f_L$, where the noise is primarily the combination of the ADC input-referred noise and the quantization noise. It should be possible to vary the low input frequency quite a bit and still measure the same SNR value. The sampling frequency is generally set for the maximum allowable. The second measurement, SNRH, is made using a high frequency input, $f_H$, where the effects of aperture jitter on the ADC SNR are noticeable. Depending on the ADC, this frequency may be as high as $f_s/2$. We have shown that relationship between the signal-to-noise ratio due to aperture jitter alone is given by:

$$SNRA = 20 \log_{10} \left[ \frac{1}{2\pi f_H t_a} \right]$$

where SNRA is the SNR (dB) due to aperture jitter, and $f_H$ is the input frequency. Solving for $t_a$:

$$t_a = \frac{1}{2\pi f_H} \cdot \frac{1}{10^{SNRA/20}}$$

The next step is to calculate SNRA based on SNRH and SNRL. Since the SNRs are in dB, they must first be converted to ratios, and their reciprocals can then be combined on an rss basis:

$$\left( \frac{1}{10^{SNRH/20}} \right)^2 = \left( \frac{1}{10^{SNRL/20}} \right)^2 + \left( \frac{1}{10^{SNRA/20}} \right)^2$$

Re-arranging Eq. 9:
Substituting Eq. 10 into Eq. 8:

\[
\left( \frac{1}{10^{\text{SNRA} / 20}} \right) = \sqrt{\left( \frac{1}{10^{\text{SNRH} / 20}} \right)^2 - \left( \frac{1}{10^{\text{SNRL} / 20}} \right)^2}.
\]

**SUMMARY**

It should be emphasized that all the measurements required for this test use SNR and not SINAD (signal-to-noise and distortion). It is extremely important that the 2\(^{nd}\), 3\(^{rd}\), 4\(^{th}\), 5\(^{th}\), and 6\(^{th}\) harmonics (as well as the dc components) be removed when making the SNR calculation from the FFT output. Otherwise, the measurement will not give an accurate measure of aperture jitter.

As a final note, measuring rms aperture jitter less than 10-ps rms is extremely difficult, simply because of unwanted jitter which may occur on the input signal or the ADC sampling clock, or layout-induced jitter and noise. Obtaining this level of accuracy requires frequency synthesizers with extremely low jitter, as well as detailed attention to layout, signal routing, grounding, and decoupling.

**REFERENCES**


INTRODUCTION

A low aperture jitter specification of an ADC is critical to achieving high levels of signal-to-noise ratios (SNR). (See References 1, 2, and 3). ADCs are available with aperture jitter specifications as low as 60-fs rms (AD9445 14-bits @ 125 MSPS and AD9446 16-bits @ 100 MSPS). Extremely low jitter sampling clocks must therefore be utilized so that the ADC performance is not degraded, because the total jitter is the root-sum-square of the internal converter aperture jitter and the external sampling clock jitter. However, oscillators used for sampling clock generation are more often specified in terms of phase noise rather than time jitter. The purpose of this discussion is to develop a simple method for converting oscillator phase noise into time jitter.

PHASE NOISE DEFINED

First, a few definitions are in order. Figure 1 shows a typical output frequency spectrum of a non-ideal oscillator (i.e., one that has jitter in the time domain, corresponding to phase noise in the frequency domain). The spectrum shows the noise power in a 1-Hz bandwidth as a function of frequency. Phase noise is defined as the ratio of the noise in a 1-Hz bandwidth at a specified frequency offset, $f_m$, to the oscillator signal amplitude at frequency $f_O$.

![Figure 1: Oscillator Power Spectrum due to Phase Noise](image)

The sampling process is basically a multiplication of the sampling clock and the analog input signal. This is multiplication in the time domain, which is equivalent to convolution in the frequency domain. Therefore, the spectrum of the sampling clock oscillator is convolved with the input and shows up on the FFT output of a pure sinewave input signal (see Figure 2).
The "close-in" phase noise will "smear" the fundamental signal into a number of frequency bins, thereby reducing the overall spectral resolution. The "broadband" phase noise will cause a degradation in the overall SNR as predicted by Eq. 1 (Reference 1 and 2):

\[
\text{SNR} = 20 \log_{10} \left[ \frac{1}{2\pi f_{m} f_{j}} \right].
\]

It is customary to characterize an oscillator in terms of its single-sideband phase noise as shown in Figure 3, where the phase noise in dBc/Hz is plotted as a function of frequency offset, \(f_{m}\), with the frequency axis on a log scale. Note the actual curve is approximated by a number of regions, each having a slope of \(1/f^{x}\), where \(x = 0\) corresponds to the "white" phase noise region (slope = 0 dB/decade), and \(x = 1\) corresponds to the "flicker" phase noise region (slope = -20 dB/decade). There are also regions where \(x = 2, 3, 4\), and these regions occur progressively closer to the carrier frequency.

Note that the phase noise curve is somewhat analogous to the input voltage noise spectral density of an amplifier. Like amplifier
voltage noise, low 1/f corner frequencies are highly desirable in an oscillator.

We have seen that oscillators are typically specified in terms of phase noise, but in order to relate phase noise to ADC performance, the phase noise must be converted into jitter. In order to make the graph relevant to modern ADC applications, the oscillator frequency (sampling frequency) is chosen to be 100 MHz for discussion purposes, and a typical graph is shown in Figure 4. Notice that the phase noise curve is approximated by a number of individual line segments, and the end points of each segment are defined by data points.

![Figure 4: Calculating Jitter from Phase Noise](image)

**CONVERTING PHASE NOISE TO JITTER**

The first step in calculating the equivalent rms jitter is to obtain the integrated phase noise power over the frequency range of interest, i.e., the area of the curve, A. The curve is broken into a number of individual areas (A1, A2, A3, A4), each defined by two data points. Generally speaking, the upper frequency range for the integration should be twice the sampling frequency, assuming there is no filtering between the oscillator and the ADC input. This approximates the bandwidth of the ADC sampling clock input.

Selecting the lower frequency for the integration also requires some judgment. In theory, it should be as low as possible to get the true rms jitter. In practice, however, the oscillator specifications generally will not be given for offset frequencies less than 10 Hz, or so - however, this will certainly give accurate enough results in the calculations. A lower frequency of integration of 100 Hz is reasonable in most cases, if that specification is available. Otherwise, use either the 1-kHz or 10-kHz data point.

One should also consider that the "close-in" phase noise affects the spectral resolution of the system, while the broadband noise affects the overall system SNR. Probably the wisest approach is to integrate each area separately as explained below and examine the magnitude of the jitter contribution of each area. The low frequency contributions may be negligible compared to the broadband contribution if a crystal oscillator is used. Other types of oscillators may have significant jitter contributions in the low frequency area, and a decision must be made regarding their importance to the overall system frequency resolution.

The integration of each individual area yields individual power ratios. The individual power ratios are then summed and converted back into dBc. Once the integrated phase noise power is known, the rms phase jitter in radians is given by the equation (see References 3-7 for further details, derivations, etc.),
RMS Phase Jitter (radians) = \sqrt{2 \cdot 10^{A/10}}. \quad \text{Eq. 2}

and dividing by 2\pi f_o converts the jitter in radians to jitter in seconds:

RMS Phase Jitter (seconds) = \frac{\sqrt{2 \cdot 10^{A/10}}}{2\pi f_o}. \quad \text{Eq. 3}

It should be noted that computer programs and spreadsheets are available online to perform the integration by segments and calculate the rms jitter, thereby greatly simplifying the process (References 8, 9).

Figure 5 shows a sample calculation which assumes only broadband phase noise. The broadband phase noise chosen of -150 dBc/Hz represents a reasonably good signal generator specification, so the jitter number obtained represents a practical situation. The phase noise of -150 dBc/Hz (expressed as a ratio) is multiplied by the bandwidth of integration (200 MHz) to obtain the integrated phase noise of -67 dBc. Note that this multiplication is equivalent to adding the quantity 10 \log_{10}(200 MHz - 0.01 MHz) to the phase noise in dBc/Hz. In practice, the lower frequency limit of 0.01 MHz can be dropped from the calculation, as it does not affect the final result significantly. A total rms jitter of approximately 1 ps is obtained using Eq. 3.

Crystal oscillators generally offer the lowest possible phase noise and jitter, and some examples are shown for comparison in Figure 6. All the oscillators shown have a typical 1/f corner frequency of 20 kHz, and the phase noise therefore represents the white phase noise level. The two Wenzel oscillators are fixed-frequency and represent excellent performance (Reference 9). It is difficult to achieve this level of performance with variable frequency signal generators, as shown by the -150 dBc specification for a relatively high quality generator.
At this point, it should be noted that there is a theoretical limit to the noise floor of an oscillator determined by the thermal noise of a matched source: $-174 \text{ dBm/Hz}$ at $+25^\circ \text{C}$. Therefore, an oscillator with a +13-dBm output into 50 $\Omega$ (2.82-V p-p) with a phase noise of $-174 \text{ dBc/Hz}$ has a noise floor of $-174 \text{ dBc} + 13 \text{ dBm} = -161 \text{ dBm}$. This is the case for the Wenzel ULN series as shown in Figure 6.

Figure 7 shows the jitter calculations from the two Wenzel crystal oscillators. In each case, the data points were taken directly for the manufacturer's data sheet. Because of the low 1/f corner frequency, the majority of the jitter is due to the “white” phase noise area. The calculated values of 64 fs (ULN-Series) and 180 fs represent extremely low jitter. For informational purposes, the individual jitter contributions of each area have been labeled separately. The total jitter is the root-sum-square of the individual jitter contributors.
In system designs requiring low jitter sampling clocks, the costs of low noise dedicated crystal oscillators is generally prohibitive. An alternative solution is to use a phase-locked-loop (PLL) in conjunction with a voltage-controlled oscillator to "clean up" a noisy system clock as shown in Figure 8. There are many good references on PLL design (see References 10-13, for example), and we will not pursue that topic further, other than to state that using a narrow bandwidth loop filter in conjunction with a voltage-controlled crystal oscillator (VCXO) typically gives the lowest phase noise. As shown in Figure 8, the PLL tends to reduce the "close-in" phase noise while at the same time, reducing the overall phase noise floor. Further reduction in the white noise floor can be obtained by following the PLL output with an appropriate bandpass filter.

The effect of enclosing a free-running VCO within a PLL is shown in Figure 9. Notice that the "close-in" phase noise is reduced significantly by the action of the PLL.
Analog Devices offers a wide portfolio of frequency synthesis products, including DDS systems, N, and fractional-N PLLs. For example, the ADF4360 family are fully integrated PLLs complete with an internal VCO. With a 10-kHz bandwidth loop filter, the phase noise of the ADF4360-1 2.25-GHz PLL is shown in Figure 10, and the line-segment approximation and jitter calculations shown in Figure 11. Note that the rms jitter is only 1.57 ps, even with a non-crystal VCO.
Historically, PLL design relied heavily on textbooks and application notes to assist in the design of the loop filter, etc. Now, with Analog Devices free downloadable ADIsimPLL® software, PLL design is much easier. To start, choose a circuit by entering the desired output frequency range, and select a PLL, VCO, and a crystal reference. Once the loop filter configuration has been selected, the circuit can be analyzed and optimized for phase noise, phase margin, gain, spur levels, lock time, etc., in both the frequency and time domain. The program also performs the rms jitter calculation based on the PLL phase noise, thereby allowing the evaluation of the final PLL output as a sampling clock.

**SUMMARY**

Sampling clock jitter can be disastrous to the SNR performance of high performance ADCs. Although the relationship between SNR and jitter is well known, most oscillators are specified in terms of their phase noise. This article has shown how to convert phase noise into jitter so that the SNR degradation can be easily calculated.

Although not as good as relatively expensive stand alone crystal oscillators, modern PLLs using crystal VCOs (along with suitable filtering) can achieve jitter performance suitable for all but the most demanding requirements.

The entire problem of clock distribution has become much more critical because of low jitter requirements. Analog Devices is now offering a line of clock distribution ICs to serve these needs (www.analog.com/clocks).

**REFERENCES**


8. Raltron Electronics Corporation, 10651 Northwest 19th Street, Miami, Florida 33172, Tel: (305) 593-6033, [http://www.raltron.com](http://www.raltron.com), (see "Convert SSB Phase Noise to Jitter" under "Engineering Design Tools").


INTRODUCTION

Analog-to-digital converters (ADCs) translate analog quantities, which are characteristic of most phenomena in the "real world," to digital language, used in information processing, computing, data transmission, and control systems. Digital-to-analog converters (DACs) are used in transforming transmitted or stored data, or the results of digital processing, back to "real-world" variables for control, information display, or further analog processing. The relationships between inputs and outputs of DACs and ADCs are shown in Figure 1.

Analog input variables, whatever their origin, are most frequently converted by transducers into voltages or currents. These electrical quantities may appear (1) as fast or slow "dc" continuous direct measurements of a phenomenon in the time domain, (2) as modulated ac waveforms (using a wide variety of modulation techniques), (3) or in some combination, with a spatial configuration of related variables to represent shaft angles. Examples of the first are outputs of thermocouples, potentiometers on dc references, and analog computing circuitry; of the second, "chopped" optical measurements, ac strain gage or bridge outputs, and digital signals buried in noise; and of the third, synchros and resolvers.

The analog variables to be dealt with in this article are those involving voltages or currents representing the actual analog phenomena. They may be either wideband or narrowband. They may be either scaled from the direct measurement, or subjected to some form of analog pre-processing, such as linearization, combination, demodulation, filtering, sample-hold, etc.

As part of the process, the voltages and currents are "normalized" to ranges compatible with assigned ADC input ranges. Analog output voltages or currents from DACs are direct and in normalized form, but they may be subsequently post-processed (e.g., scaled, filtered, amplified, etc.).

Information in digital form is normally represented by arbitrarily fixed voltage levels referred to "ground," either occurring at the outputs of logic gates, or applied to their inputs. The digital numbers used are all basically binary; that is, each "bit," or unit of
information has one of two possible states. These states are "off," "false," or "0," and "on," "true," or "1." It is also possible to represent the two logic states by two different levels of current, however this is much less popular than using voltages. There is also no particular reason why the voltages need be referenced to ground - as in the case of emitter-coupled-logic (ECL), positive-emitter-coupled-logic (PECL) or low-voltage-differential-signaling logic (LVDS), for example.

Words are groups of levels representing digital numbers; the levels may appear simultaneously in parallel, on a bus or groups of gate inputs or outputs, serially (or in a time sequence) on a single line, or as a sequence of parallel bytes (i.e., "byte-serial") or nibbles (small bytes). For example, a 16-bit word may occupy the 16 bits of a 16-bit bus, or it may be divided into two sequential bytes for an 8-bit bus, or four 4-bit nibbles for a 4-bit bus.

Although there are several systems of logic, the most widely used choice of levels are those used in TTL (transistor-transistor logic) and, in which positive true, or 1, corresponds to a minimum output level of +2.4 V (inputs respond unequivocally to "1" for levels greater than 2.0 V); and false, or 0, corresponds to a maximum output level of +0.4 V (inputs respond unequivocally to "0" for anything less than +0.8 V). It should be noted that even though CMOS is more popular today than TTL, CMOS logic levels are generally made to be compatible with the older TTL logic standard.

A unique parallel or serial grouping of digital levels, or a number, or code, is assigned to each analog level which is quantized (i.e., represents a unique portion of the analog range). A typical digital code would be this array:

\[ a_7 \ a_6 \ a_5 \ a_4 \ a_3 \ a_2 \ a_1 \ a_0 = 1 \ 0 \ 1 \ \ 1 \ \ 1 \ \ 0 \ \ 0 \ \ 1 \]

It is composed of eight bits. The "1" at the extreme left is called the "most significant bit" (MSB, or Bit 1), and the one at the right is called the "least significant bit" (LSB, or bit N: 8 in this case). The meaning of the code, as either a number, a character, or a representation of an analog variable, is unknown until the code and the conversion relationship have been defined. It is important not to confuse the designation of a particular bit (i.e., Bit 1, Bit 2, etc.) with the subscripts associated with the "a" array. The subscripts correspond to the power of 2 associated with the weight of a particular bit in the sequence.

The best-known code (other than base 10) is natural or straight binary (base 2). Binary codes are most familiar in representing integers; i.e., in a natural binary integer code having N bits, the LSB has a weight of 2^0 (i.e., 1), the next bit has a weight of 2^1 (i.e., 2), and so on up to the MSB, which has a weight of 2^{N-1} (i.e., 2^N/2). The value of a binary number is obtained by adding up the weights of all non-zero bits. When the weighted bits are added up, they form a unique number having any value from 0 to 2^N - 1. Each additional trailing zero bit, if present, essentially doubles the size of the number.

In converter technology, full-scale (abbreviated FS) is independent of the number of bits of resolution, N. A more useful coding is fractional binary which is always normalized to full-scale. Integer binary can be interpreted as fractional binary if all integer values are divided by 2^N. For example, the MSB has a weight of ½ (i.e., 2^{(N-1)/2^N} = 2^{-1}), the next bit has a weight of ¼ (i.e., 2^{-2}), and so forth down to the LSB, which has a weight of 1/2^N (i.e., 2^{-N}). When the weighted bits are added up, they form a number with any of 2^N values, from 0 to (1 - 2^{-N}) of full-scale. Additional bits simply provide more fine structure without affecting full-scale range. The relationship between base-10 numbers and binary numbers (base 2) are shown in Figure 2 along with examples of each.
In data conversion systems, the coding method must be related to the analog input range (or span) of an ADC or the analog output range (or span) of a DAC. The simplest case is when the input to the ADC or the output of the DAC is always a unipolar positive voltage (current outputs are very popular for DAC outputs, much less for ADC inputs). The most popular code for this type of signal is straight binary and is shown in Figure 3 for a 4-bit converter. Notice that there are 16 distinct possible levels, ranging from the all-zeros code 0000, to the all-ones code 1111. It is important to note that the analog value represented by the all-ones code is not full-scale (abbreviated FS), but FS - 1 LSB. This is a common convention in data conversion notation and applies to both ADCs and DACs. Figure 3 gives the base-10 equivalent number, the value of the base-2 binary code relative to full-scale (FS), and also the corresponding voltage level for each code (assuming a +10 V full-scale converter. The Gray code equivalent is also shown, and will be discussed shortly. 

<table>
<thead>
<tr>
<th>BASE 10 NUMBER</th>
<th>SCALE</th>
<th>+10V FS</th>
<th>BINARY</th>
<th>GRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>+15</td>
<td>+15/16 FS</td>
<td>9.375</td>
<td>1111</td>
<td>1000</td>
</tr>
<tr>
<td>+14</td>
<td>+7/8 FS</td>
<td>8.750</td>
<td>1110</td>
<td>1001</td>
</tr>
<tr>
<td>+13</td>
<td>+13/16 FS</td>
<td>8.125</td>
<td>1101</td>
<td>1011</td>
</tr>
<tr>
<td>+12</td>
<td>+3/4 FS</td>
<td>7.600</td>
<td>1100</td>
<td>1010</td>
</tr>
<tr>
<td>+11</td>
<td>+11/16 FS</td>
<td>6.875</td>
<td>1011</td>
<td>1110</td>
</tr>
<tr>
<td>+10</td>
<td>+5/8 FS</td>
<td>6.250</td>
<td>1010</td>
<td>1111</td>
</tr>
<tr>
<td>+9</td>
<td>+9/16 FS</td>
<td>5.625</td>
<td>1001</td>
<td>1101</td>
</tr>
<tr>
<td>+8</td>
<td>+1/2 FS</td>
<td>5.000</td>
<td>1000</td>
<td>1100</td>
</tr>
<tr>
<td>+7</td>
<td>+7/16 FS</td>
<td>4.375</td>
<td>0111</td>
<td>0100</td>
</tr>
<tr>
<td>+6</td>
<td>+3/8 FS</td>
<td>3.750</td>
<td>0110</td>
<td>0101</td>
</tr>
<tr>
<td>+5</td>
<td>+5/16 FS</td>
<td>3.125</td>
<td>0101</td>
<td>0111</td>
</tr>
<tr>
<td>+4</td>
<td>+1/4 FS</td>
<td>2.500</td>
<td>0100</td>
<td>0110</td>
</tr>
<tr>
<td>+3</td>
<td>+3/16 FS</td>
<td>1.875</td>
<td>0011</td>
<td>0010</td>
</tr>
<tr>
<td>+2</td>
<td>+1/8 FS</td>
<td>1.250</td>
<td>0010</td>
<td>0011</td>
</tr>
<tr>
<td>+1</td>
<td>1 LSB</td>
<td>0.625</td>
<td>0001</td>
<td>0001</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>0.000</td>
<td>0000</td>
<td>0000</td>
</tr>
</tbody>
</table>

Figure 4 shows the transfer function for an ideal 3-bit DAC with straight binary input coding. Notice that the analog output is zero for the all-zeros input code. As the digital input code increases, the analog output increases 1 LSB (1/8 scale in this example) per
code. The most positive output voltage is \( \frac{7}{8} \) FS, corresponding to a value equal to FS - 1 LSB. The mid-scale output of \( \frac{1}{2} \) FS is generated when the digital input code is 100.

The transfer function of an ideal 3-bit ADC is shown in Figure 5. There is a range of analog input voltage over which the ADC will produce a given output code; this range is the quantization uncertainty and is equal to 1 LSB. Note that the width of the transition regions between adjacent codes is zero for an ideal ADC. In practice, however, there is always transition noise associated with these levels, and therefore the width is non-zero. It is customary to define the analog input corresponding to a given code by the code center which lies halfway between two adjacent transition regions (illustrated by the black dots in the diagram). This requires that the first transition region occur at \( \frac{1}{2} \) LSB. The full-scale analog input voltage is defined by \( \frac{7}{8} \) FS, (FS - 1 LSB).

![Figure 4: Transfer Function for Ideal Unipolar 3-bit DAC](image)
Another code worth mentioning at this point is the Gray code (or reflective-binary) which was invented by Elisha Gray in 1878 (Reference 1) and later re-invented by Frank Gray in 1949 (see Reference 2). The Gray code equivalent of the 4-bit straight binary code is also shown in the last column of Figure 3. Although it is rarely used in computer arithmetic, it has some useful properties which make it attractive to A/D conversion. Notice that in Gray code, as the number value changes, the transitions from one code to the next involve only one bit at a time. Contrast this to the binary code where all the bits change when making the transition between 0111 and 1000. Some ADCs make use of it internally and then convert the Gray code to a binary code for external use.

As mentioned above, the Gray code has the property that adjacent levels differ by only one digit in the corresponding Gray-coded word. Therefore, if there is an error in a bit decision for a particular level, the corresponding error after conversion to binary code is only one least significant bit (LSB). In the case of mid-scale, note that only the MSB changes. It is interesting to note that this same phenomenon can occur in modern comparator-based flash converters due to comparator metastability. With small overdrive, there is a finite probability that the output of a comparator will generate the wrong decision in its latched output, producing the same effect if straight binary decoding techniques are used. In many cases, Gray code, or "pseudo-Gray" codes are used to decode the comparator bank. The Gray code output is then latched, converted to binary, and latched again at the final output.

Other examples where Gray code is often used in the conversion process to minimize errors are shaft encoders (angle-to-digital) and optical encoders.

ADCs which use the Gray code internally almost always convert the Gray code output to binary for external use. The conversion from Gray-to-binary and binary-to-Gray is easily accomplished with the exclusive-or logic function as shown in Figure 8.
BIPOLAR CODES

In many systems, it is desirable to represent both positive and negative analog quantities with binary codes. Either offset binary, twos complement, ones complement, or sign magnitude codes will accomplish this, but offset binary and twos complement are by far the most popular. The relationships between these codes for a 4-bit systems is shown in Figure 9. Note that the values are scaled for a ±5-V full-scale input/output voltage range.

For offset binary, the zero signal value is assigned the code 1000. The sequence of codes is identical to that of straight binary. The only difference between a straight and offset binary system is the half-scale offset associated with analog signal. The most negative value (-FS + 1 LSB) is assigned the code 0001, and the most positive value (+FS - 1 LSB) is assigned the code 1111. Note that in order to maintain perfect symmetry about mid-scale, the all-zeros code (0000) representing negative full-scale (-FS) is not normally used in computation. It can be used to represent a negative off-range condition or simply assigned the value of the 0001 (-FS + 1 LSB).
The relationship between the offset binary code and the analog output range of a bipolar 3-bit DAC is shown in Figure 10. The analog output of the DAC is zero for the zero-value input code 100. The most negative output voltage is generally defined by the 001 code \((-\text{FS} + 1 \text{ LSB})\), and the most positive by 111 \((+\text{FS} - 1 \text{ LSB})\). The output voltage for the 000 input code is available for use if desired, but makes the output non-symmetrical about zero and complicates the mathematics.

![Figure 10: Transfer Function for Ideal Bipolar 3-bit DAC](image)

The offset binary output code for a bipolar 3-bit ADC as a function of its analog input is shown in Figure 11. Note that zero analog input defines the center of the mid-scale code 100. As in the case of bipolar DACs, the most negative input voltage is generally defined by the 001 code \((-\text{FS} + 1 \text{ LSB})\), and the most positive by 111 \((+\text{FS} - 1 \text{ LSB})\). As discussed above, the 000 output code is available for use if desired, but makes the output non-symmetrical about zero and complicates the mathematics.
Twos complement is identical to offset binary with the most-significant-bit (MSB) complemented (inverted). This is obviously very easy to accomplish in a data converter, using a simple inverter or taking the complementary output of a “D” flip-flop. The popularity of twos complement coding lies in the ease with which mathematical operations can be performed in computers and DSPs. Twos complement, for conversion purposes, consists of a binary code for positive magnitudes (0 sign bit), and the twos complement of each positive number to represent its negative. The twos complement is formed arithmetically by complementing the number and adding 1 LSB. For example, -3/8 FS is obtained by taking the twos complement of +3/8 FS. This is done by first complementing +3/8 FS, 0011 obtaining 1100. Adding 1 LSB, we obtain 1101.

Twos complement makes subtraction easy. For example, to subtract 3/8 FS from 4/8 FS, add 4/8 to -3/8, or 0100 to 1101. The result is 0001, or 1/8, disregarding the extra carry.

Ones complement can also be used to represent negative numbers, although it is much less popular than twos complement and rarely used today. The ones complement is obtained by simply complementing all of a positive number’s digits. For instance, the ones complement of 3/8 FS (0011) is 1100. A ones complemented code can be formed by complementing each positive value to obtain its corresponding negative value. This includes zero, which is then represented by either of two codes, 0000 (referred to as 0+) or 1111 (referred to as 0-). This ambiguity must be dealt with mathematically, and presents obvious problems relating to ADCs and DACs for which there is a single code which represents zero.

Sign-magnitude would appear to be the most straightforward way of expressing signed analog quantities digitally. Simply determine the code appropriate for the magnitude and add a polarity bit. Sign-magnitude BCD is popular in bipolar digital voltmeters, but has the problem of two allowable codes for zero. It is therefore unpopular for most applications involving ADCs or DACs.

Figure 12 summarizes the relationships between the various bipolar codes: offset binary, twos complement, ones complement, and sign-magnitude and shows how to convert between them.
The last code to be considered in this section is *binary-coded-decimal (BCD)*, where each base-10 digit (0 to 9) in a decimal number is represented as the corresponding 4-bit straight binary word as shown in Figure 13. The minimum digit 0 is represented as 0000, and the digit 9 by 1001. This code is relatively inefficient, since only 10 of the 16 code states for each decade are used. It is, however, a very useful code for interfacing to decimal displays such as in digital voltmeters.

![Figure 12: Relationships Among Bipolar Codes](image)

The last code to be considered in this section is *binary-coded-decimal (BCD)*, where each base-10 digit (0 to 9) in a decimal number is represented as the corresponding 4-bit straight binary word as shown in Figure 13. The minimum digit 0 is represented as 0000, and the digit 9 by 1001. This code is relatively inefficient, since only 10 of the 16 code states for each decade are used. It is, however, a very useful code for interfacing to decimal displays such as in digital voltmeters.

![Figure 13: Binary Coded Decimal (BCD) Code](image)

**COMPLEMENTARY CODES**

Some forms of data converters (for example, early DACs using monolithic NPN quad current switches), require standard codes such as natural binary or BCD, but with all bits represented by their complements. Such codes are called **complementary codes**. All the codes discussed thus far have complementary codes which can be obtained by this method. A *complementary code*...
should not be confused with a ones complement or a twos complement code.

In a 4-bit complementary-binary converter, 0 is represented by 1111, half-scale by 0111, and FS - 1 LSB by 0000. In practice, the complementary code can usually be obtained by using the complementary output of a register rather than the true output, since both are available.

Sometimes the complementary code is useful in inverting the analog output of a DAC. Today many DACs provide differential outputs which allow the polarity inversion to be accomplished without modifying the input code. Similarly, many ADCs provide differential logic outputs which can be used to accomplish the polarity inversion.

ACKNOWLEDGEMENT

The author would like to thank Dan Sheingold of Analog Devices for the permission to use material directly from his classic 1986 Analog-Digital Conversion Handbook (Reference 3).

REFERENCES


INTRODUCTION

In the 1950s and 1960s, dc performance specifications such as integral nonlinearity, differential nonlinearity, monotonicity, no missing codes, gain error, offset error, and drift, etc., were primarily used to characterize data converters. These specifications were adequate during this era, because most early applications (with the exception of PCM and radar, for example) dealt with dc or low frequency signals such as those encountered in industrial measurement and process control. With the advent of microprocessors and digital signal processing (DSP) in the 1970s and 1980s, dynamic performance specifications, such as signal-to-noise ratio (SNR), spurious free dynamic range (SFDR), etc., were required in order to adequately characterize converters for more sophisticated signal processing applications.

Modern data converter applications cover the frequency spectrum from low frequency industrial measurement to wideband radio receivers. Even though the importance of dc specifications generally decreases with increasing signal frequency, they are still important in many applications. For instance, significant gain and/or offset errors in IF sampling applications can cause signal clipping and thereby degrade SNR and SFDR. In applications requiring matched converters, such as interleaving, simultaneous sampling, and I/Q signal processing, the relative gain and offset matching between individual converters is critical.

This article explains the dc performance specifications of data converters so that you will be equipped to understand this important part of an ADC or DAC data sheet.

DATA CONVERTER RESOLUTION AND QUANTIZATION

It is important to remember that for both DACs and ADCs, either the input or output is digital, and therefore the signal is quantized. That is, an N-bit word represents one of $2^N$ possible states, and therefore an N-bit DAC (with a fixed reference) can have only $2^N$ possible analog outputs, and an N-bit ADC can have only $2^N$ possible digital outputs. As previously discussed, the analog signals will generally be voltages or currents.

The resolution of data converters may be expressed in several different ways: the weight of the Least Significant Bit (LSB), parts per million of full-scale (ppm FS), millivolts (mV), etc. Different devices (even from the same manufacturer) will be specified differently, so converter users must learn to translate between the different types of specifications if they are to compare devices successfully. The size of the least significant bit for various resolutions is shown in Figure 1.
Figure 1: Quantization: The Size of a Least Significant Bit (LSB)

TRANSFER FUNCTIONS OF IDEAL DATA CONVERTERS

Figure 2 shows the ideal transfer characteristics for a 3-bit unipolar DAC and a 3-bit unipolar ADC. In a DAC, both the input and the output are quantized, and the graph consists of eight points – while it is reasonable to discuss the line through these points, it is very important to remember that the actual transfer characteristic is not a line, but a number of discrete points.

The input to an ADC is analog and is not quantized, but its output is quantized. The transfer characteristic therefore consists of eight horizontal steps. When considering the offset, gain and linearity of an ADC we consider the line joining the midpoints of these steps – often referred to as the code centers.

For both DACs and ADCs, digital full-scale (all "1"s) corresponds to 1 LSB below the analog full-scale (FS). The (ideal) ADC
transitions take place at ½ LSB above zero, and thereafter every LSB, until 1½ LSB below analog full-scale. Since the analog input to an ADC can take any value, but the digital output is quantized, there may be a difference of up to ½ LSB between the actual analog input and the exact value of the digital output. This is known as the quantization error or quantization uncertainty also shown in Figure 2. In ac (sampling) applications this quantization error gives rise to quantization noise which is discussed in detail in other articles.

There are many possible digital coding schemes for data converters: straight binary, offset binary, 1’s complement, 2’s complement, sign magnitude, gray code, BCD and others. This article, being devoted mainly to the analog issues surrounding data converters, will use simple binary and offset binary in its examples and will not consider the merits and disadvantages of these, or any other forms of digital code. For more details on data converter coding, refer to References 1 and 2.

The examples in Figure 2 use unipolar converters, whose analog port has only a single polarity. These are the simplest type, but bipolar converters are generally more useful in real-world applications. There are two types of bipolar converters: the simpler is merely a unipolar converter with an accurate 1 MSB of negative offset (and many converters are arranged so that this offset may be switched in and out so that they can be used as either unipolar or bipolar converters at will), but the other, known as a sign-magnitude converter is more complex, and has N bits of magnitude information and an additional bit which corresponds to the sign of the analog signal. Sign-magnitude DACs are quite rare, and sign-magnitude ADCs are found mostly in digital voltmeters (DVMs). The unipolar, offset binary, and sign-magnitude representations are shown in Figure 3.

DATA CONVERTER GAIN AND OFFSET ERRORS

The four dc errors in a data converter are offset error, gain error, and two types of linearity error (differential and integral). Offset and gain errors are analogous to offset and gain errors in amplifiers as shown in Figure 4 for a bipolar input range. (Though offset error and zero error, which are identical in amplifiers and unipolar data converters, are not identical in bipolar converters and should be carefully distinguished.)
The transfer characteristics of both DACs and ADCs may be expressed as a straight line given by \( D = K + GA \), where \( D \) is the digital code, \( A \) is the analog signal, and \( K \) and \( G \) are constants. In a unipolar converter, the ideal value of \( K \) is zero; in an offset bipolar converter it is -1 MSB. The offset error is the amount by which the actual value of \( K \) differs from its ideal value.

The gain error is the amount by which \( G \) differs from its ideal value, and is generally expressed as the percentage difference between the two, although it may be defined as the gain error contribution (in mV or LSB) to the total error at full-scale. These errors can usually be trimmed by the data converter user. Note, however, that amplifier offset is trimmed at zero input, and then the gain is trimmed near to full-scale. The trim algorithm for a bipolar data converter is not so straightforward.

**DATA CONVERTER LINEARITY ERRORS**

The integral linearity error of a converter is also analogous to the linearity error of an amplifier, and is defined as the maximum deviation of the actual transfer characteristic of the converter from a straight line, and is generally expressed as a percentage of full-scale (but may be given in LSBs). For an ADC, the most popular convention is to draw the straight line through the mid-points of the codes, or the code centers. There are two common ways of choosing the straight line: end point and best straight line as shown in Figure 5.
In the *end point* system, the deviation is measured from the straight line through the origin and the full-scale point (after gain adjustment). This is the most useful integral linearity measurement for measurement and control applications of data converters (since error budgets depend on deviation from the ideal transfer characteristic, not from some arbitrary "best fit"), and is the one normally adopted by Analog Devices, Inc.

The *best straight line*, however, does give a better prediction of distortion in ac applications, and also gives a lower value of "linearity error" on a data sheet. The best fit straight line is drawn through the transfer characteristic of the device using standard curve fitting techniques, and the maximum deviation is measured from this line. In general, the integral linearity error measured in this way is only 50% of the value measured by end point methods. This makes the method good for producing impressive data sheets, but it is less useful for error budget analysis. For ac applications it is better to specify distortion than dc linearity, so it is rarely necessary to use the best straight line method to define converter linearity.

The other type of converter nonlinearity is *differential nonlinearity* (DNL). This relates to the linearity of the code transitions of the converter. In the ideal case, a change of 1 LSB in digital code corresponds to a change of exactly 1 LSB of analog signal. In a DAC, a change of 1 LSB in digital code produces exactly 1 LSB change of analog output, while in an ADC there should be exactly 1 LSB change of analog input to move from one digital transition to the next. Differential linearity error is defined as the maximum amount of deviation of any quantum (or LSB change) in the entire transfer function from its ideal size of 1 LSB.

Where the change in analog signal corresponding to 1 LSB digital change is more or less than 1 LSB, there is said to be a DNL error. The DNL error of a converter is normally defined as the maximum value of DNL to be found at any transition across the range of the converter. Figure 6 shows the non-ideal transfer functions for a DAC and an ADC and shows the effects of the DNL error.

![Figure 6: Transfer Functions for Non-Ideal 3-Bit DAC and ADC](image)

The DNL of a DAC is examined more closely in Figure 7. If the DNL of a DAC is less than -1 LSB at any transition, the DAC is non-monotonic i.e., its transfer characteristic contains one or more localized maxima or minima. A DNL greater than +1 LSB does not cause non-monotonicity, but is still undesirable. In many DAC applications (especially closed-loop systems where non-monotonicity can change negative feedback to positive feedback), it is critically important that DACs are monotonic. DAC monotonicity is often explicitly specified on data sheets, although if the DNL is guaranteed to be less than 1 LSB (i.e., $|\text{DNL}| \leq 1$ LSB) then the device must be monotonic, even without an explicit guarantee.
In Figure 8, the DNL of an ADC is examined more closely on an expanded scale. ADCs can be non-monotonic, but a more common result of excess DNL in ADCs is missing codes. Missing codes in an ADC are as objectionable as non-monotonicity in a DAC. Again, they result from DNL < -1 LSB.

Not only can ADCs have missing codes, they can also be non-monotonic as shown in Figure 9. As in the case of DACs, this can present major problems—especially in servo applications.
In a DAC, there can be no missing codes – each digital input word will produce a corresponding analog output. However, DACs can be non-monotonic as previously discussed. In a straight binary DAC, the most likely place a non-monotonic condition can develop is at mid-scale between the two codes: 011…11 and 100…00. If a non-monotonic conditions occurs here, it is generally because the DAC is not properly calibrated or trimmed. A successive approximation ADC with an internal non-monotonic DAC will generally produce missing codes but remain monotonic. However it is possible for an ADC to be non-monotonic – again depending on the particular conversion architecture. Figure 9 shows the transfer function of an ADC which is non-monotonic and has a missing code.

ADCs which use the subranging architecture divide the input range into a number of coarse segments, and each coarse segment is further divided into smaller segments – and ultimately the final code is derived. This process is described in more detail in Chapter 4 of this book. An improperly trimmed subranging ADC may exhibit non-monotonicity, wide codes, or missing codes at the subranging points as shown in Figure 10A, B, and C, respectively. This type of ADC should be trimmed so that drift due to aging or temperature produces wide codes at the sensitive points rather than non-monotonic or missing codes.
Defining missing codes is more difficult than defining non-monotonicity. All ADCs suffer from some inherent transition noise as shown in Figure 11 (think of it as the flicker between adjacent values of the last digit of a DVM). As resolutions and bandwidths become higher, the range of input over which transition noise occurs may approach, or even exceed, 1 LSB. High resolution wideband ADCs generally have internal noise sources which can be reflected to the input as effective input noise summed with the signal. The effect of this noise, especially if combined with a negative DNL error, may be that there are some (or even all) codes where transition noise is present for the whole range of inputs. There are therefore some codes for which there is no input which will guarantee that code as an output, although there may be a range of inputs which will sometimes produce that code.

For low resolution ADCs, it may be reasonable to define no missing codes as a combination of transition noise and DNL which guarantees some level (perhaps 0.2 LSB) of noise-free code for all codes. However, this is impossible to achieve at the very high resolutions achieved by modern sigma-delta ADCs, or even at lower resolutions in wide bandwidth sampling ADCs. In these cases, the manufacturer must define noise levels and resolution in some other way. Which method is used is less important, but the data sheet should contain a clear definition of the method used and the performance to be expected. A complete discussion of effective input noise is given in References 2 and 3.
REFERENCES


MT-011: Find Those Elusive ADC Sparkle Codes and Metastable States

by Walt Kester

INTRODUCTION

A major concern in the design of digital communications systems is the bit error rate (BER). The effect of the ADC noise on system BER can be analyzed, provided the noise is Gaussian. Unfortunately, ADCs may have non-Gaussian error codes which contribute to the BER in ways that are not predictable by simple analysis. Bit error rate may also be a concern in such instrumentation applications as digital oscilloscopes, especially when operating in the "single-shot" mode or when trying to capture infrequent transient pulses. An error code can be misinterpreted as a transient pulse, thereby giving a false result. This tutorial describes the basic mechanisms within ADCs that can contribute to the error rate, ways to minimize the problem, and methods for measuring the BER.

SPARKLE CODES, ERROR CODES, RABBITS, OR FLYERS

Random noise, regardless of the source, creates a finite probability of errors (deviations from the expected output). Before describing the error code sources, however, it is important to define what constitutes an ADC error or "sparkle" code. Noise generated prior to, or inside the ADC can be analyzed in the traditional manner. In most cases, the ADC noise has a Gaussian distribution and is a function of the resolution of the ADC (quantization noise) and additional noise created within the ADC (input-referred noise). An ADC sparkle code is any deviation from the expected output that is not attributable to the effective Gaussian noise of the ADC. Figure 1 illustrates an exaggerated output of a low-amplitude sinewave applied to an ADC that has error codes. The figure does not show the Gaussian noise of the ADC.

![Figure 1: Exaggerated Output of ADC Showing Error Codes](image)

The large errors are more significant than those due to the ADC Gaussian noise and are not expected. These errors are random and are usually so infrequent that an FFT-based SNR test of the ADC will rarely detect them. These types of errors plagued some of the early ADCs for video applications in the 1970s, and were given the name sparkle codes because of their appearance on a TV screen as small white dots or "sparkles" under certain test conditions. These errors have also been called rabbits or flyers. In digital communications applications, this type of error increases the overall system bit error rate (BER). However, it should be noted that most communications systems have built-in error detection and correction codes which correct some of the error codes; therefore, the analysis of the actual degradation in overall system bit error rate due to the ADC sparkle codes becomes quite difficult. On the other hand, the effect of the ADC Gaussian noise on overall system BER has been well
SPARKLE CODES IN FLASH CONVERTERS

In order to understand the causes of the error codes, we will first consider the case of a simple flash converter. Comparators used as building blocks in flash ADCs need good resolution which implies high gain. This can lead to uncontrolled oscillation when the differential input approaches zero. In order to prevent this, hysteresis is often added to comparators using a small amount of positive feedback. Figure 2 shows the effects of hysteresis on the overall transfer function. Many comparators have a millivolt or two of hysteresis to encourage "snap" action and to prevent local feedback from causing instability in the transition region. Note that the resolution of the comparator can be no less than the hysteresis, so large values of hysteresis are generally not useful in ADC applications.

![Figure 2: Latched Comparator](image)

Early comparators were designed with vacuum tubes and were often used in radio receivers – where they were called discriminators, not comparators. Most modern comparators used in ADCs include a built-in latch which makes them sampling devices suitable for data converters. A typical structure is shown in Figure 3 for the AM685 ECL (emitter-coupled-logic) latched comparator introduced in 1972 by Advanced Micro Devices, Inc. (see Reference 1).
The input stage preamplifier drives a cross-coupled latch. The latch locks the output in the logic state it was in at the instant when the latch was enabled. The latch thus performs a track-and-hold function, allowing short input signals to be detected and held for further processing. Because the latch operates directly on the input stage, the signal suffers no additional delays – signals only a few nanoseconds wide can be acquired and held. The latched comparator is also less sensitive to instability caused by local feedback than an unlatched one.

Where comparators are incorporated into IC ADCs, their design must consider resolution, speed, overload recovery, power dissipation, offset voltage, bias current, and the chip area occupied by the architecture which is chosen. There is another subtle but troublesome characteristic of comparators which can cause large errors (sparkle codes) in ADCs if not understood and dealt with effectively. This error mechanism is the occasional inability of a comparator to resolve a small differential input into a valid output logic level. This phenomenon is known as metastability – the ability of a comparator to balance right at its threshold for an extended period of time.

The metastable state problem is illustrated in Figure 4. Three conditions of differential input voltage are illustrated: (1) large differential input voltage, (2) small differential input voltage, and (3) zero differential input voltage.
The approximate equation which describes the output voltage, $V_O(t)$, is given by:

$$V_O(t) = \Delta V_{IN} A e^{t/\tau}, \quad \text{Eq. 1}$$

where $\Delta V_{IN}$ is the differential input voltage at the time of latching, $A$ is the gain of the preamp at the time of latching, $\tau$ is the regeneration time constant of the latch, and $t$ is the time that has elapsed after the comparator output is latched (see References 2 and 3).

For small differential input voltages, the output takes longer to reach a valid logic level. If the output data is read when it lies between the "valid logic 1" and the "valid logic 0" region, the data can be in error. If the differential input voltage is exactly zero, and the comparator is perfectly balanced at the time of latching, the time required to reach a valid logic level can be quite long (theoretically infinite). However, comparator hysteresis and input noise makes this condition highly unlikely. The effects of invalid logic levels out of the comparator are different depending upon how the comparator is used in the actual ADC.

From a design standpoint, comparator metastability can be minimized by making the gain ($A$) high, minimizing the regeneration time constant ($\tau$) by increasing the gain-bandwidth of the latch, and allowing sufficient time ($t$), for the output of the comparator to settle to a valid logic level. It is not the purpose of this discussion to analyze the complex tradeoffs between speed, power, and circuit complexity when optimizing comparator designs, but an excellent treatment of the subject can be found in References 2 and 3.

It is easy to see how metastability in a comparator can produce sparkle codes in a flash converter. If simple binary decoding logic is used to decode the thermometer code, a metastable comparator output may result in a large output code error. Consider the case of a simple 3-bit flash converter shown in Figure 5. Assume that the input signal is exactly at the threshold of Comparator 4 and random noise is causing the comparator to toggle between a "1" and a "0" output each time a latch strobe is applied. The corresponding binary output should be interpreted as either 011 or 100. If, however, the comparator output is in a metastable state, the simple binary decoding logic shown may produce binary codes 000, 011, 100, or 111. The codes 000 and 111 represent a one-half scale departure from the expected codes and will appear as sparkle codes.
Figure 5: Metastable Comparator Output States
May Cause Error Codes in Data Converters

The probability of errors due to metastability increases as the sampling rate increases because less time is available for a metastable comparator to settle.

Various measures have been taken in flash converter designs to minimize the metastable state problem. Initially decoding the comparator outputs in Gray code or "pseudo-Gray" code (followed by conversion to binary) is one method. This and other decoding schemes described in References 2 to 6 can minimize the magnitude of these errors. Optimizing comparator designs for regenerative gain and small time constants is another way to reduce these problems – usually at the expense of additional power.

Metastable state errors (and the resulting sparkle codes) may also appear in successive approximation and pipelined subranging ADCs which make use of flash converters and comparators as building blocks. The same concepts apply, although the magnitudes and locations of the error codes may be different.

The test system shown in Figure 6 may be used to test for BER in an ADC. The analog input to the ADC is provided by a high stability low noise sinewave generator. The analog input level is set slightly greater than full-scale, and the frequency such that there is always slightly less than 1-LSB change between samples as shown in Figure 7.
The test set uses two series buffer registers to acquire successive codes A and B. A logic circuit determines the absolute difference between A and B. This difference is then compared to the error limit, chosen to allow for the expected random Gaussian noise spikes due to the normal ADC noise. Errors which cause the difference to be larger than the error limit will increment the counters. The number of errors, E, are counted over a period of time, T. The error rate is then calculated as BER = E/(2Tf_s). The factor of 2 in the denominator is required because the hardware records a second error when the output returns to the correct code after making the initial error. The error counter is therefore incremented twice for each error. It should be noted that the same function can be accomplished in software if the ADC outputs are stored in a memory and analyzed by a computer program.

The input frequency must be carefully chosen such that there is at least one sample taken per code as shown in Figure 7. Assume a full-scale input sinewave having an amplitude of 2^N/2:

\[ v(t) = \frac{2^N}{2} \sin \ 2\pi f t \]  

Eq. 2

The maximum rate of change of this signal is

\[ \frac{dv}{dt}|_{\text{max}} \leq 2^N \pi f. \]  

Eq. 3

Letting dv = 1 LSB, dt = 1/f_s, and solving for the input frequency:

\[ f_{in} \leq \frac{f_s}{2^N \pi}. \]  

Eq. 4

Choosing an input frequency less than this value will ensure that there is at least one sample per code as shown in Figure 7.
The same test can be conducted at high frequencies by applying an input frequency slightly offset from $f_s/2$ as shown in Figure 8. This causes the ADC to slew full-scale between conversions. Every other conversion is compared, and the “beat” frequency is chosen such that there is slightly less than 1 LSB change between alternate samples.

The equation for calculating the proper frequency for the high frequency BER test is derived as follows.

Assume an input full-scale sinewave of amplitude $2^N/2$ whose frequency is slightly less than $f_s/2$ by a frequency equal to $\Delta f$. 

$$f_{\text{in}} = \frac{f_s}{2} - \Delta f$$

$$\Delta f \leq \frac{f_s}{2} \left[ 1 - \frac{1}{2^{N+1}} \right]$$
The maximum rate of change of this signal is

\[
\frac{dv}{dt}_{\text{max}} \leq 2^N \pi \left( \frac{f_s}{2} - \Delta f \right).
\]  

Eq. 6

Letting \( dv = 1 \) LSB and \( dt = 2/f_s \), and solving for the input frequency \( \Delta f \):

\[
\Delta f \leq \frac{f_s}{2} \left( 1 - \frac{1}{2 \cdot 2^N \pi} \right).
\]  

Eq. 7

Establishing the BER of a well-behaved ADC is a difficult, time-consuming task – a single converter can sometimes be tested for days without an error. For example, tests on the AD9002 8-bit 150-MSPS flash converter operating at a sampling rate of 75 MSPS yielded a BER of approximately \( 3.7 \times 10^{-12} \) (1 error per hour), with an error limit of 4 LSBs. Meaningful tests for long periods of time require special attention to EMI/RFI effects (requiring a shielded screen room), isolated power supplies, isolation from soldering irons with mechanical thermostats, isolation from other bench equipment, etc.

Figure 9 shows the average time between errors as a function of BER for a sampling frequency of 75 MSPS. This illustrates the difficulty in measuring low BER because the long measurement times increase the probability of power supply transients, noise, etc. causing an error.

<table>
<thead>
<tr>
<th>Bit Error Rate (BER)</th>
<th>Average Time Between Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1 \times 10^{-8} )</td>
<td>1.3 seconds</td>
</tr>
<tr>
<td>( 1 \times 10^{-9} )</td>
<td>13.3 seconds</td>
</tr>
<tr>
<td>( 1 \times 10^{-10} )</td>
<td>2.2 minutes</td>
</tr>
<tr>
<td>( 1 \times 10^{-11} )</td>
<td>22 minutes</td>
</tr>
<tr>
<td>( 1 \times 10^{-12} )</td>
<td>3.7 hours</td>
</tr>
<tr>
<td>( 1 \times 10^{-13} )</td>
<td>1.5 days</td>
</tr>
<tr>
<td>( 1 \times 10^{-14} )</td>
<td>15 days</td>
</tr>
</tbody>
</table>

Figure 9: Average Time Between Errors Versus BER when Sampling at 75 MSPS

SUMMARY

From a user standpoint the effect of comparator metastability (if it affects the ADC performance at all) is in the bit error rate (BER) – which is not usually specified on most ADC data sheets. As can be seen from this tutorial, completely specifying ADC BER requires extensive characterization of devices under a variety of input conditions and error limits, resulting in a formidable task.

Bit error rate should not be a problem in a properly designed ADC in most applications, however the system designer should be aware that the phenomenon can exist. An application example where it can be a problem is when the ADC is used in a digital oscilloscope to detect small-amplitude single-shot randomly occurring events. The ADC can give false indications if its BER is not sufficiently small. Excessive ADC bit error rates in communications applications can degrade the overall system bit error rate.
REFERENCES


INTRODUCTION

Intermodulation distortion (IMD) is a popular measure of the linearity of amplifiers, gain blocks, mixers, and other RF components. The second and third-order intercept points (IP2 and IP3) are figures of merit for these specifications and allow distortion products to be computed for various signal amplitudes. RF engineers are quite familiar with these specifications, but confusion can arise when applying them to ADCs.

This tutorial first defines IMD with respect to ADCs and then points out the precautions which must be taken when applying the definitions of IP2 and IP3 to ADCs.

TWO TONE INTERMODULATION DISTORTION (IMD)

Two tone IMD is measured by applying two spectrally pure sinewaves to the ADC at frequencies $f_1$ and $f_2$, usually relatively close together. The amplitude of each tone is set slightly more than 6 dB below full-scale so that the ADC does not clip when the two tones add in-phase. The location of the second and third-order products are shown in Figure 1. Notice that the second-order products fall at frequencies which can be removed by digital filters. However, the third-order products $2f_2 - f_1$ and $2f_1 - f_2$ are close to the original signals and are more difficult to filter. Unless otherwise specified, two-tone IMD refers to these "close-in" third-order products. It is customary to specify the value of the IMD product in dBc relative to the value of either of the two original tones, and not to their sum.

![Figure 1: Second and Third-Order Intermodulation Products for $f_1 = 5$ MHz, $f_2 = 6$ MHz](image)

Note, however, that if the two tones are close to $f_s/4$, then the aliased third harmonics of the fundamentals can make the identification of the actual $2f_2 - f_1$ and $2f_1 - f_2$ products difficult. This is because the third harmonic of $f_s/4$ is $3f_s/4$, and the alias occurs at $f_s - 3f_s/4 = f_s/4$. Similarly, if the two tones are close to $f_s/3$, the aliased second harmonics may interfere with the measurement. The same reasoning applies here; the second harmonic of $f_s/3$ is $2f_s/3$, and its alias occurs at $f_s - 2f_s/3 = f_s/3$. 
SECOND AND THIRD-ORDER INTERCEPT POINTS (IP2, IP3), 1-dB COMPRESSION POINT

Third-order IMD products are especially troublesome in multi-channel communications systems where the channel separation is constant across the frequency band. Third-order IMD products can mask out small signals in the presence of larger ones.

In amplifiers, mixers, and other RF components, it is common practice to specify the third-order IMD products in terms of the third order intercept point (IP3) as is shown by Figure 2. Two spectrally pure tones are applied to the system. The output signal power in a single tone (in dBm) as well as the relative amplitude of the third-order products (referenced to a single tone) are plotted as a function of input signal power. The fundamental is shown by the slope = 1 curve in the diagram. If the system nonlinearity is approximated by a power series expansion, it can be shown that second-order IMD (IMD2) amplitudes increase 2 dB for every 1 dB of signal increase, as represented by the slope = 2 curve in the diagram.

Similarly, the third-order IMD (IMD3) amplitudes increase 3 dB for every 1-dB of signal increase, as indicated by the slope = 3 plotted line. With a low level two-tone input signal, and two data points, one can draw the second and third order IMD lines as they are shown in Figure 2 (using the principle that a point and a slope define a straight line).

Once the input reaches a certain level however, the output signal begins to soft-limit, or compress. A parameter of interest here is the 1-dB compression point. This is the point where the output signal is compressed 1 dB from an ideal input/output transfer function. This is shown in Figure 2 within the region where the ideal slope = 1 line becomes dotted, and the actual response exhibits compression (solid).

Nevertheless, both the second- and third-order intercept lines may be extended, to intersect the (dotted) extension of the ideal output signal line. These intersections are called the second and third-order intercept points, respectively, or IP2 and IP3. These power level values are usually referenced to the output power of the device delivered to a matched load (usually, but not necessarily 50 Ω) expressed in dBm.

It should be noted that IP2, IP3, and the 1-dB compression point are all a function of frequency, and as one would expect, the distortion is worse at higher frequencies. For a given frequency, knowing the third order intercept point allows calculation of the approximate level of the third-order IMD products as a function of output signal level.
The concept of second- and third-order intercept points is not valid for an ADC, because the distortion products do not vary in a predictable manner (as a function of signal amplitude). The ADC does not gradually begin to compress signals approaching full-scale (there is no 1-dB compression point); it acts as a hard limiter as soon as the signal exceeds the ADC input range, thereby suddenly producing extreme amounts of distortion because of clipping. On the other hand, for signals much below full-scale, the distortion floor remains relatively constant and is independent of signal level. This is shown graphically in Figure 3.

![Figure 3: Intercept Points for Data Converters Have No Practical Significance](image)

The IMD curve in Figure 3 is divided into three regions. For low level input signals, the IMD products remain relatively constant regardless of signal level. This implies that as the input signal increases 1 dB, the ratio of the signal to the IMD level will increase 1 dB also. When the input signal is within a few dB of the ADC full-scale range, the IMD may start to increase (but it might not in a very well-designed ADC). The exact level at which this occurs is dependent on the particular ADC under consideration – some ADCs may not exhibit significant increases in the IMD products over their full input range, however most will. As the input signal continues to increase beyond full-scale, the ADC should function acts as an ideal limiter, and the IMD products become very large.

For these reasons, the 2nd and 3rd order IMD intercept points are not specified for ADCs. It should be noted that essentially the same arguments apply to DACs. In either case, the single- or multi-tone SFDR specification is the most accepted way to measure data converter distortion.

**MULTI-TONE SPURIOUS FREE DYNAMIC RANGE**

Two-tone and multi-tone SFDR are often measured in communications applications. The larger number of tones more closely simulates the wideband frequency spectrum of cellular telephone systems such as AMPS or GSM. Figure 4 shows the 2-tone intermodulation performance of the AD9444 14-bit, 80-MSPS ADC. The input tones are at 69.3 MHz and 70.3 MHz and are located in the second Nyquist Zone.
The aliased tones therefore occur at 9.7 MHz and 10.7 MHz in the first Nyquist Zone. Figure 4 also shows the location of all the aliased IMD products. High SFDR increases the receiver's ability to capture small signals in the presence of large ones, and prevents the small signals from being masked by the intermodulation products of the larger ones. Figure 5 shows the AD9444 two-tone SFDR as a function of input signal amplitude for the same input frequencies.

**SUMMARY**

Intermodulation distortion (IMD2, IMD3) and intercept points (IP2, IP3) are well understood specifications for RF components such as mixers, LNAs, gain blocks, and amplifiers. The nonlinearity of these devices is modeled by a power series expansion, and distortion levels for various signal amplitudes can be predicted from the intercept points IP2 and IP3. Unlike amplifiers and
mixers, ADC distortion (especially for low level signals) does not follow the simple power series expansion model, so the intercept points IP2 and IP3 are not useful in predicting distortion performance. In addition, ADCs act as ideal limiters when input signals exceed full-scale, while amplifiers and mixers generally act as soft limiters.

In spite of these differences, however, it is critical to know the two-tone IMD performance of ADCs used in communications applications. A good data sheet will present this data for a large number of input signal frequencies and amplitudes. In addition, the ADIsimADC™ program can be very useful in evaluating various ADCs at the specific frequencies and amplitudes required in the system application. The ADIsimADC program acts as a virtual evaluation board and can be downloaded from the Analog Devices' website along with the most recent models for IF-sampling ADCs. The program uses an FFT engine and gives accurate SNR, SFDR, and IMD results for both single and dual-tone input signals.

REFERENCES

INTRODUCTION

Unlike an ADC which requires an FFT processor to evaluate spectral purity, a DAC produces an analog output which can be examined directly using a traditional analog spectrum analyzer. A challenge in DAC evaluation is generating the digital input that can range from a single-tone sinewave to a complex wideband CDMA signal. Direct digital synthesis techniques can be used to generate digital sinewaves, but more sophisticated and expensive word generators are needed to produce the more complex digital signals.

The ac specifications which are the most important in evaluating high speed DACs are settling time, glitch impulse area, distortion, spurious free dynamic range (SFDR), and signal-to-noise ratio (SNR). Time domain specifications will be addressed first, followed by frequency domain specifications.

DAC SETTLING TIME

The precise settling time of a DAC may or may not be of interest depending upon the application. It is especially important in high speed DACs used in video displays because of the high pixel rates associated with high resolution monitors. The DAC must be capable of making the transition from all "0"s (black level) to all "1"s (white level) in 5% to 10% of a pixel interval, which can be quite short. For instance, even the relatively common 1024 x 768, 60-Hz refresh-rate monitor has a pixel interval of approximately 16 ns. This implies a required settling time of less than 2 ns to at least 8-bit accuracy (for an 8-bit system).

The fundamental definitions of full-scale settling time are shown in Figure 1. The definition is quite similar to that of the settling time of an op amp. Notice that settling time can be defined in two acceptable ways. The more traditional definition is the amount of time required for the output to settle with the specified error band measured with respect to the 50% point of either the data strobe to the DAC (if it has a parallel register driving the DAC switches) or the time when the input data to the switches changes (if there is no internal register).

Another equally valid definition is to define the settling time with respect to the time the output leaves the initial error band. This effectively removes the "dead time" from the measurement. In video DAC applications, for instance, settling time with respect to the output is the key specification – the fixed delay (dead time) is of little interest.

The error band is usually defined in terms of an LSB or % full-scale. It is customary, but not mandatory, to define the error band as 1 LSB. However, measuring full-scale settling time to 1 LSB at the 12-bit level (0.025% FS) is possible with care, but measuring it to 1 LSB at the 16-bit level (0.0015% FS) presents a real instrumentation challenge. For this reason, high-speed DACs such as the TxDAC® family specify 14- and 16-bit settling time to the 12-bit level, 0.025% FS (typically less than 11 ns).
Mid-scale settling time is also of interest, because in a binary-weighted DAC, the transition between the 0111⋯1 code and the 1000⋯0 code produces the largest transient. In fact, if there is significant bit skew, the transient amplitude can approach full-scale. Figure 2 shows a waveform along with the two acceptable definitions of mid-scale settling time. As in the case of full-scale settling time, mid-scale settling time can either be referred to the output or to the latch strobe (or the bit transitions if there is no internal latch).

GLITCH IMPULSE AREA

Ideally, when a DAC output changes it should move from one value to its new one monotonically. In practice, the output is likely
to overshoot, undershoot, or both. This uncontrolled movement of the DAC output during a transition is known as a **glitch**. It can arise from two mechanisms: capacitive coupling of digital transitions to the analog output, and the effects of some switches in the DAC operating more quickly than others and producing temporary spurious outputs.

Capacitive coupling frequently produces roughly equal positive and negative spikes (sometimes called a **doublet glitch**) which more or less cancel in the longer term. The glitch produced by switch timing differences is generally unipolar, much larger, and of greater concern.

Glitches can be characterized by measuring the **glitch impulse area**, sometimes inaccurately called **glitch energy**. The term glitch energy is a misnomer, since the unit for glitch impulse area is volt-seconds (or more probably µV-sec or pV-sec). The **peak glitch area** is the area of the largest of the positive or negative glitch areas.

Glitch impulse area is easily estimated from the mid-scale settling time waveform as shown in Figure 3. The areas of the four triangles are used to calculate the net glitch area. Recall that the area of a triangle is one-half the base times the height. If the total positive area equals the total negative glitch area, then the net area is zero. The specification given on most data sheets is the net glitch area, although in some cases, the peak area may be specified instead.

**Figure 3: Glitch Impulse Area**

**OSCILLOSCOPE MEASUREMENT OF SETTLING TIME AND GLITCH IMPULSE AREA**

A wideband fast-settling oscilloscope is crucial to accurate settling time measurements. There are several considerations in selecting the proper scope. The required bandwidth can be calculated based on the rise/fall time of the DAC output, for instance, a 1-ns output risetime and falltime corresponds to a bandwidth of $0.35/t_r = 350$ MHz. A scope of at least 500-MHz bandwidth would be required. Preferably, the scope bandwidth should be at least three times the signal bandwidth to include the second and third harmonic components for a more accurate representation of the waveform.

Modern digital storage scopes (DSOs) and digital phosphor scopes (DPOs) are popular and offer an excellent solution for performing settling time measurements as well as many other waveform analysis functions (see Reference 3). These scopes offer real-time sampling rates of several GHz and are much less sensitive to overdrive than older analog scopes or traditional sampling scopes. Overdrive is a serious consideration in measuring settling time, because the scope is generally set to maximum sensitivity when measuring a full-scale DAC output change. For instance, measuring 12-bit settling for a 1-V output (20 mA into 50 Ω) requires the resolution of a signal within a 0.25-mV error band riding on the top of a 1-V step function.
From a historical perspective, older analog oscilloscopes were sensitive to overdrive and could not be used to make accurate step function settling time without adding additional circuitry. Quite a bit of work was done during the 1980s on circuits to cancel out portions of the step function using Schottky diodes, current sources, etc. References 4, 5, and 6 are good examples of various circuits which were used during that time to mitigate the oscilloscope overdrive problems.

Even with modern DSOs and DPOs, overdrive should still be checked by changing the scope sensitivity by a known factor and making sure that all portions of the waveform change proportionally. Measuring the mid-scale settling time can also subject the scope to considerable overdrive if there is a large glitch. The sensitivity of the scope should be sufficient to measure the desired error band. A sensitivity of 1-mV/division allows the measurement of a 0.25-mV error band if care is taken (one major vertical division is usually divided into five smaller ones, corresponding to 0.2 mV/small division). If the DAC has an on-chip op amp, the fullscale output voltage may be larger, perhaps 10 V, and the sensitivity required in the scope is relaxed proportionally.

Although there is a well-known relationship between the risetime and the settling time in a single-pole system, it is inadvisable to extrapolate DAC settling time using risetime alone. There are many higher order nonlinear effects involved in a DAC which dominate the actual settling time, especially for DACs of 12-bits or higher resolution.

When making settling time measurements, it is generally better to make a direct connection between the DAC output and the 50-Ω scope input and avoid the use of probes. FET probes are notorious for giving misleading settling time results. If probes must be used, compensated passive ones are preferable, but they should be used with care. Skin effect associated with even short lengths of properly terminated coaxial cable can give erroneous settling time results. In making the connection between the DAC and the scope, it is mandatory that a good low impedance ground be maintained. This can be accomplished by soldering the ground of a BNC connector to the ground plane on the DAC test board and using this BNC to connect to the scope's 50-Ω input. A manufacturer's evaluation board can be of great assistance in interfacing to the DAC and should be used if available.

Finally, if the DAC output is specifically designed to drive the virtual ground of an external current-to-voltage converter and does not have enough compliance to develop a measurable voltage across a load resistor, then an external op amp is required, and the test circuit measures the settling time of the DAC/op amp combination. In this case, select an op amp that has a settling time which is at least 3 to 5 times smaller than the DAC under test. If the settling time of the op amp is comparable to that of the DAC, the settling time of the DAC can be determined, because the total settling time of the combination is the root-sum-square of the DAC settling time and the op amp settling time. Solving the equation for the DAC settling time yields:

\[ DAC \text{ Settling Time} = \sqrt{(Total \ Settling \ Time)^2 - (Op \ Amp \ Settling \ Time)^2}. \quad \text{Eq. 1} \]

**DAC DISTORTION**

If we consider the spectrum of a waveform reconstructed by a DAC from digital data, we find that in addition to the expected spectrum (which will contain one or more frequencies, depending on the nature of the reconstructed waveform), there will also be noise and distortion products.

Code-dependent glitches will produce both out-of-band and in-band harmonics when the DAC is reconstructing a digitally generated sinewave as in a Direct Digital Synthesis (DDS) system. For instance, the mid-scale glitch occurs twice during a single cycle of a reconstructed sinewave (at each mid-scale crossing), and will therefore produce a second harmonic of the sinewave, as shown in Figure 4. Note that the higher order harmonics of the sinewave, which also alias back into the Nyquist bandwidth (dc to \( f_s/2 \)), cannot be filtered.
Although segmented DAC architectures can be used to greatly minimize the distortion caused by code-dependent glitches, the distortion can never be completely eliminated.

It is difficult to predict the harmonic distortion or SFDR from the glitch area specification alone. Other factors, such as the overall linearity of the DAC, also contribute to distortion. In addition, integer ratios between the DAC sampling clock and the DAC output frequency and the cause the quantization noise to concentrate at harmonics of the fundamental thereby increasing the apparent distortion at these points.

Because so many DAC applications are in communications and frequency analysis systems, practically all modern DACs are now specified in the frequency domain. The basic ac specifications include harmonic distortion, total harmonic distortion (THD), signal-to-noise ratio (SNR), total harmonic distortion plus noise (THD + N), spurious free dynamic range (SFDR), etc. In order to test a DAC for these specifications, a proper digitally-synthesized signal must be generated to drive the DAC (for example, a single or multi-tone sinewave).

In the early 1970s, when ADC and DAC frequency domain performance first became important, "back-to-back" testing was popular. An ADC and its companion DAC were connected together, and the appropriate analog signal source was selected to drive the ADC. An analog spectrum analyzer was then used to measure the distortion and noise of the DAC output. This approach was logical, because ADCs and DACs were often used in conjunction with a digital signal processor placed between them to perform various functions. Obviously, it was impossible to determine exactly how the total ac errors were divided between the ADC and the DAC. Today, however, ADCs and DACs are used quite independently of one another, so they must be completely tested on their own.

Figure 5 shows a typical test setup for measuring the distortion and noise of a DAC. The first consideration, of course, is the generation of the digital signal to drive the DAC. To achieve this, modern arbitrary waveform generators (for example Tektronix AWG2021 with Option 4) or word generators (Tektronix DG2020) allow almost any waveform to be synthesized digitally in software, and are mandatory in serious frequency domain testing of DACs (see Reference 3). In most cases, these generators have standard waveforms pre-programmed, such as sinewaves and triangle waves, for example. In many communications applications, however, more complex digital waveforms are required, such as two-tone or multi-tone sinewaves, QAM, GSM, and CDMA test signals, etc. In many cases, application-specific hardware and software exists for generating these types of signals and can greatly speed up the evaluation process.
Analog Devices and other manufacturers of high performance DACs furnish evaluation boards which greatly simplify interfacing to the test equipment. Because many communications DACs (such as the TxDAC®-family) have quite a bit of on-chip control logic, their evaluation boards have interfaces to PCs via the SPI, USB, parallel, or serial ports, as well as Windows™-compatible software to facilitate setting the various DAC options and modes of operation.

Testing DACs which are part of a direct-digital-synthesis (DDS) system is somewhat easier because the DDS portion of the IC acts as the digital signal generator for the DAC. Testing these DACs often requires no more than the manufacturer's evaluation board, a PC, a stable clock source, and a high performance spectrum analyzer.

The spectrum analyzer chosen to measure the distortion and noise performance of the DAC should have at least 10-dB more dynamic range than the DAC being tested. The "maximum intermodulation-free range" specification of the spectrum analyzer is an excellent indicator of distortion performance (see Reference 7). However, spectrum analyzer manufacturers may specify distortion performance in other ways. Modern communications DACs such as the TxDAC®-series require high performance spectrum analyzers such as the Rhode and Schwartz FSEA30 (Reference 7). As in the case of oscilloscopes, the spectrum analyzer must not be sensitive to overdrive. This can be easily verified by applying a signal corresponding to the full-scale DAC output, measuring the level of the harmonic distortion products, and then attenuating the signal by 6 dB or so and verifying that both the signal and the harmonics drop by the same amount. If the harmonics drop more than the fundamental signal drops, then the analyzer is distorting the signal.

In some cases, an analyzer with less than optimum overdrive performance can still be used by placing a bandstop filter in series with the analyzer input to remove the frequency of the fundamental signal being measured. The analyzer looks only at the remaining distortion products. This technique will generally work satisfactorily, provided the attenuation of the bandstop filter is taken into account when making the distortion measurements. Obviously, a separate bandstop filter is required for each individual output frequency tested, and therefore multi-tone testing is cumbersome.

Finally, there are a variety of application-specific analyzers for use in communications, video, and audio. In video, the Tektronix VM-700 and VM-5000 series are widely used (Reference 3). In measuring the performance of DACs designed for audio applications, special signal analyzers designed specifically for audio are preferred. The industry standard for audio analyzers is the Audio Precision, System Two (see Reference 8). There are, of course, many other application-specific analyzers available which may be preferred over the general-purpose types. In addition, software is usually available for generating the various
digital test signals required for the applications.

Once the proper analyzer is selected, measuring the various distortion and noise-related specifications such as SFDR, THD, SNR, SINAD, etc., is relatively straightforward. The analyzer resolution bandwidth must be set low enough so that the harmonic products can be resolved above the noise floor. Figure 6 shows a typical spectral output where the SFDR is measured.

![Figure 6: Measuring DAC Spurious Free Dynamic Range (SFDR)](image)

Figure 7 shows how to measure the various harmonic distortion components with a spectrum analyzer. The first nine harmonics are shown. Notice that aliasing causes the 6th, 7th, 8th, 9th, and 10th harmonic to fall back inside the f_c/2 Nyquist bandwidth.

![Figure 7: Measuring DAC Distortion and SNR with an Analog Spectrum Analyzer](image)

The harmonics of the input signal can be distinguished from other distortion products by their location in the frequency spectrum. Figure 8 shows a 7-MHz input signal sampled at 20 MSPS and the location of the first 9 harmonics. Aliased harmonics of f_c fall at frequencies equal to |±Kf_c ± nf_o|, where n is the order of the harmonic, and K = 0, 1, 2, 3,.... The second and third harmonics
are generally the only ones specified on a data sheet because they tend to be the largest, although some data sheets may specify the value of the worst harmonic. An interactive Harmonic Image Calculator applet is available on the Analog Devices’ Design Center website which shows the locations of the second and third harmonics as a function of output frequency and DAC update rate. In addition, the tool shows the attenuation effects of the sin x/x rolloff and the output anti-imaging filter.

![Figure 8: Non-Monotonic ADC with Missing Code](image)

**DAC NOISE**

The spectrum analyzer can also be used to measure SNR if the proper correction factors are taken into account. Figure 7 shows the analyzer sweep bandwidth, BW, which in most cases will be considerably less than \( f_c/2 \). First, measure the noise floor level with respect to the signal level at a point in the frequency spectrum which is relatively free of harmonics. This corresponds to the value "\( S/(\text{NOISE FLOOR}) \)" in the diagram. The actual SNR over the dc to \( f_c/2 \) bandwidth is obtained by subtracting the process gain, \( 10\log_{10}(f_c/2\cdot\text{BW}) \), from the \( S/(\text{NOISE FLOOR}) \).

\[
\text{SNR} = \frac{S}{(\text{NOISE FLOOR})} - 10\log_{10}(f_c/2\cdot\text{BW}).
\]

Eq. 2

In order for this SNR result to be accurate, one must precisely know the analyzer bandwidth. The bandwidth characteristics of the analyzer should be given out in the manufacturer’s documentation. Also, if there is any signal averaging used in the analyzer, that may affect the net correction factor.

In order to verify the process gain calculation, several LSBs can be disabled under these conditions, the SNR performance of the DAC should approach ideal. For instance, measuring the 8-bit SNR of a low distortion, low noise 12-, 14-, or 16-bit DAC should produce near theoretical results. The theoretical 8-bit SNR, calculated using the formula \( \text{SNR} = 6.02N + 1.76 \text{ dB} \), is 50 dB. The process gain can then be calculated using the formula:

\[
\text{PROCESS GAIN} = \frac{S}{(\text{NOISE FLOOR})} - \text{SNR}.
\]

Eq. 3

The accuracy of this measurement should be verified by enabling the 9th bit of the DAC and ensuring that the analyzer noise floor drops by 6 dB. If the noise floor does not drop by 6 dB, the measurement should be repeated using only the first 6 bits of the DAC. If near theoretical SNR is not achieved at the 6-bit level, the DAC under consideration is probably not suitable for ac applications where noise and distortion are important.
The relationship between SINAD, SNR, and THD can be derived as follows. THD is defined as the ratio of the signal to the root-sum-square (rss) of a specified number harmonics of the fundamental signal. IEEE Std. 1241-2000 (Reference 9) suggests that the first 10 harmonics be included. Various manufacturers may choose to include fewer than 10 harmonics in the calculation. Analog Devices defines THD to be the root-sum-square of the first 6 harmonics (2\textsuperscript{nd}, 3\textsuperscript{rd}, 4\textsuperscript{th}, 5\textsuperscript{th}, and 6\textsuperscript{th}) for example. In practice, the difference in dB between THD measured with 10 versus 6 harmonics is less than a few tenths of a dB, unless there is an extreme amount of distortion. The various harmonics, V2 through V6, are measured with respect to the signal level, S, in dBc. They are then converted into a ratio, combined on an rss basis, and converted back into dB to obtain the THD.

The signal-to-noise-and-distortion, SINAD, can then be calculated by combining SNR and THD as a root-sum-square:

$$\text{SINAD} = 20\log_{10} \sqrt{\left(10^{-\text{SNR} / 20}\right)^2 + \left(10^{-\text{THD} / 20}\right)^2}$$  \hspace{1cm} \text{Eq. 4}

An SNR/THD/SINAD Calculator applet is available on the Analog Devices' Design Center website to assist in these conversions.

One of the most important factors in obtaining accurate distortion measurements is to ensure that the DAC output frequency, f_o, is not a sub-harmonic of the update rate, f_c. If \(f_c/f_o\) is an integer, then the quantization error is not random, but is correlated with the output frequency. This causes the quantization noise energy to be concentrated at harmonics of the fundamental output frequency, thereby producing distortion which is an artifact of the sampling process rather than nonlinearity in the DAC. It should be noted that these same artifacts can occur in evaluating ADCs.

To illustrate this point, Figure 9 shows simulated results for an ideal 12-bit DAC where 9A shows the output frequency spectrum for the case of \(f_c/f_o = 40\). Notice that the SFDR is approximately 77 dBc. The right-hand spectral output (9B) shows the case where the \(f_c/f_o\) ratio is a non-integer—the quantization noise is now random, and the SFDR is 93 dBc.

Figure 9: Correlated (A) and Uncorrelated (B) Quanization Noise Spectrum of an Ideal 12-Bit DAC

Because of the wide range of possible clock and output frequencies, Analog Devices offers special fast-turnaround measurements on TxDACs for specific customer test vectors. This important service allows system designers to do advance frequency planning to ensure optimum distortion performance for their application.

In lieu of specific frequency measurements, the SFDR performance of a DAC is often plotted as a function of the output frequency at fixed clock rates. This data is usually taken for sinewave outputs of various amplitudes as shown in Figure 10 for the AD9777 16-bit TxDAC. Note that this plot does not include data points where there is strong correlation between the quantization noise and the signal (i.e., where the ratio of the clock frequency to the output frequency is an integer number).
There is another useful test method that gives a good overall indicator of the DAC performance at various combinations of output and clock frequencies. Specifically, this involves testing distortion for output frequencies, $f_o$, equal to $f_c/3$ and $f_c/4$. In practice, the output frequency is slightly offset by a small amount, $\Delta f$, where $\Delta f$ is a non-integer fraction of $f_c$, i.e., $\Delta f = kf_c$, where $k \ll 1$. For an output frequency of $f_c/3 - \Delta f$, the even-order harmonics are spaced at intervals of $\Delta f$ around the fundamental $f_o$ output frequency as shown in Figure 11. The worst even-order harmonic is measured at various clock frequencies up to the maximum allowable while maintaining this same ratio. The same procedure should be repeated for an output frequency $f_c/4 - \Delta f$, in which case the odd-order harmonics are uniformly spaced around the output frequency as shown in Figure 12.

---

Figure 10: AD9777 16-bit TxDAC™ SFDR, Data Update Rate = 160 MSPS

Figure 11: Location of Even Harmonics for $f_o = f_c/3 - \Delta f$
These measurements are relatively easy to make, since once the ratio of \( f_o \) to \( f_c \) is established by the DDS or digital waveform generator, it is preserved as the master clock frequency is changed. Figure 13 shows a typical plot of SFDR versus clock frequency for a low distortion DAC with two output frequencies \( f_c/3 \) and \( f_c/4 \). In most cases, the \( f_c/3 \) distortion represents a worst case condition and is good for comparing various DACs.

DAC OUTPUT SPECTRUM AND SIN (X)/X Frequency Rolloff

The output of a reconstruction DAC can be represented as a series of rectangular pulses whose width is equal to the reciprocal of the clock rate as shown in Figure 14. Note that the reconstructed signal amplitude is down 3.92 dB at the Nyquist frequency, \( f_c/2 \). An inverse \( \sin(x)/x \) filter can be used to compensate for this effect in most cases and is usually designed as part of the anti-imaging filter. The images of the fundamental signal occur as a result of the sampling function and are also attenuated by the \( \sin(x)/x \) function.
If there is no compensation for the \( \sin(x)/x \) rolloff, it must be considered when making bandwidth measurements on the DAC output. The effect of the rolloff on distortion and SNR measurements is negligible over the Nyquist bandwidth, \( dc \) to \( f_c/2 \).

An interactive Harmonic Image Calculator applet is available on the Analog Devices' Design Center website which shows the locations of the second and third harmonics as a function of output frequency and DAC update rate. In addition, the tool shows the attenuation effects of both the \( \sin(x)/x \) rolloff and the output anti-imaging filter.

REFERENCES

3. Tektronix, Inc., 14200 SW Karl Braun Drive, P. O. Box 500, Beaverton, OR 97077, Phone: (800) 835-9433, http://www.tek.com. (the website contains a wealth of information on oscilloscopes, measurement techniques, probing, etc., as well as complete specifications on products).
MT-014: Basic DAC Architectures I: String DACs and Thermometer (Fully Decoded) DACs

by Walt Kester

INTRODUCTION

Rather than simply treating DACs as black boxes having a digital input and an analog output, it is much more useful to understand the fundamental DAC architectures in use today. This can also aid in the selection process which can be somewhat daunting considering the sheer number of DACs currently on the market.

This tutorial examines the most fundamental DAC architectures, the "string" DAC and the "thermometer" DAC. String DACs had their origin with Lord Kelvin, who invented the Kelvin divider in the mid-1800s. String DACs are popular today, especially in applications such as digital potentiometers where resolutions of 6 to 8 bits are typical. Because of their relative freedom from code-dependent switching glitches, thermometer DACs are popular building blocks in low distortion segmented DACs as well as in pipelined ADCs.

THE SWITCH: A SIMPLE 1-BIT DAC

It is reasonable to consider a changeover switch (a single-pole, double-throw, SPDT switch), switching an output between a reference and ground or between equal positive and negative reference voltages, as a 1-bit DAC as shown in Figure 1. Such a simple device is a component of many more complex DAC structures, and is used, with oversampling, as the basic analog component in many of the sigma-delta DACs we shall discuss later. The simple switch is also very easy to implement in standard CMOS processes. Nevertheless it is a little too simple to require detailed discussion, and it is more rewarding to consider more complex structures.

THE KELVIN DIVIDER (STRING DAC)

The simplest DAC structure of all, after the changeover switch mentioned above, is the Kelvin divider or string DAC as shown in Figure 2. An N-bit version of this DAC simply consists of $2^N$ equal resistors in series and $2^N$ switches (usually CMOS), one between each node of the chain and the output. The output is taken from the appropriate tap by closing just one of the switches (there is some slight digital complexity involved in decoding to 1 of $2^N$ switches from N-bit data, but digital circuitry is cheap). The origins of this DAC date back to Lord Kelvin in the mid-1800s, and it was first implemented using resistors and relays, and later with vacuum tubes in the 1920s (See References 1, 2, 3).
Figure 2: Simplest Voltage-Output Thermometer DAC: The Kelvin Divider ("String DAC")

This architecture is simple, has a voltage output (but a code-dependent output impedance) and is inherently monotonic – even if a resistor is accidentally short-circuited, output \( n \) cannot exceed output \( n + 1 \). It is linear if all the resistors are equal, but may be made deliberately nonlinear if a nonlinear DAC is required. Since only two switches operate during a transition, it is a low-glitch architecture. Also, the switching glitch is not code-dependent, making it ideal for low distortion applications. Because the glitch is relatively constant regardless of the code transition, the frequency content of the glitch is at the DAC update rate and its harmonics – not at the harmonics of the fundamental DAC output frequency. The major drawback of the string DAC is the large number of resistors and switches required for high resolution, and as a result it was not commonly used as a simple DAC architecture until the recent advent of very small IC feature sizes made it very practical for low and medium resolution DACs. Today the architecture is quite widely used in simple DACs, such as digital potentiometers and, as we shall see later, its current-output version, the thermometer DAC, is also used as a component in more complex high resolution segmented DAC structures.

The output of a DAC for an all "1"s code is 1 LSB below the reference, so a string DAC intended for use as a general purpose DAC has a resistor between the reference terminal and the first switch as shown in Figure 2.

In an ideal potentiometer, on the other hand, all "0"s and all "1"s codes should connect the variable tap to one or other end of the string of resistors. So a digital potentiometer, while basically the same as a general purpose string DAC, has one fewer resistor, and neither end of the string has any other internal connection. A simple digital potentiometer is shown in Figure 3.
The simplest digital potentiometers are no more complex than this, and none of the potentiometer terminals may be at a potential outside the 5-V or 3-V logic supply. But others have more complex decoders with level shifters and additional high voltage supply terminals, so that while the logic control levels are low (3 V or 5 V), the potentiometer terminals have a much greater range – up to ±15 V in some cases. Digital potentiometers frequently incorporate nonvolatile logic so that their settings are retained when they are turned off.

It is evident that string DACs have a large number of resistors ($2^N$ for an N-bit DAC as we have already seen). It is not practical to trim every resistor in a string DAC to obtain perfect DNL and INL, partly because they are too many, and partly because they are too small to trim, and mainly because it's too costly. Because of the physical size, pure string DACs are primarily limited to resolutions of 8 to 10 bits.

**CURRENT OUTPUT THERMOMETER (FULLY DECODED) DACs**

There is a current-output DAC analogous to a string DAC that consists of $2^N - 1$ switchable current sources (which may be resistors and a voltage reference or may be active current sources) connected to an output terminal, which must be at, or close to, ground. This architecture is commonly referred to as a “thermometer” or “fully decoded” DAC. Figure 4 shows such a thermometer DAC which uses resistors connected to a reference voltage to generate the currents.
If active current sources are used as shown in Figure 5, the output may have more compliance, and a resistive load used to develop an output voltage. The load resistor must be chosen so that at maximum output current the output terminal remains within its rated compliance voltage.

Once a current in a thermometer DAC is switched into the circuit by increasing the digital code, any further increases do not switch it out again. The structure is thus inherently monotonic, irrespective of inaccuracies in the currents. Again, like the Kelvin divider, only the advent of high density IC processes has made this architecture practical for general purpose medium resolution DACs, although a slightly more complex version – shown in the next diagram – is quite widely used in high speed applications. Unlike the Kelvin divider, this type of current-mode DAC does not have a unique name, although both types may be referred to as thermometer DACs or fully-decoded DACs.

A DAC where the currents are switched between two output lines – one of which is often grounded, but may, in the more general case, be used as the inverted output – is more suitable for high speed applications because switching a current between two
outputs is far less disruptive, and so causes a far lower glitch than simply switching a current on and off. This architecture is shown in Figure 6.

![Figure 6: High Speed Thermometer DAC with Complementary Current Outputs](image)

But the settling time of this DAC still varies with initial and final code, giving rise to intersymbol interference (ISI). This can be addressed with even more complex switching where the output current is returned to zero before going to its next value. Note that although the current in the output is returned to zero it is not "turned off" – the current is dumped when it is not being used, rather than being switched on and off. The techniques involved are too complex to discuss in detail here but can be found in Reference 4.

In the normal (linear) version of this DAC, all the currents are nominally equal. Where it is used for high speed reconstruction, its linearity can be improved by dynamically changing the order in which the currents are switched by ascending code. Instead of code 001 always turning on current A; code 010 always turning on currents A & B, code 011 always turning on currents A, B & C; etc., the order of turn-on relative to ascending code changes for each new data point. This can be done quite easily with a little extra logic in the decoder. The simplest way of achieving it is with a counter which increments with each clock cycle so that the order advances: ABCDEFG, BCDEFGA, CDEFGAB, etc., but this algorithm may give rise to spurious tones in the DAC output. A better approach is to set a new pseudo-random order on each clock cycle – this requires a little more logic, but, as we have pointed out, even complex logic is now very cheap and easily implemented on CMOS processes. There are other, even more complex, techniques which involve using the data itself to select bits and thus turn current mismatch into shaped noise. Again they are too complex for a tutorial of this sort. (See References 4 and 5 for a more detailed discussion).

REFERENCES


INTRODUCTION

While the string DAC and thermometer DAC architectures are by far the simplest, they are certainly not the most efficient when high resolutions are required. Binary-weighted DACs utilize one switch per bit and were first developed in the 1920s (see References 1, 2, and 3). Since then, the architecture has remained popular and forms the backbone for modern precision as well as high-speed DACs.

BINARY-WEIGHTED DACS

The voltage-mode binary-weighted resistor DAC shown in Figure 1 is usually the simplest textbook example of a DAC. However, this DAC is not inherently monotonic and is actually quite hard to manufacture successfully at high resolutions. In addition, the output impedance of the voltage-mode binary DAC changes with the input code.

Current-mode binary DACs are shown in Figure 2A (resistor-based), and Figure 2B (current-source based). An N-bit DAC of this type consists of N weighted current sources (which may simply be resistors and a voltage reference) in the ratio $1:2:4:8:...:2^{N-1}$. The LSB switches the $2^{N-1}$ current, the MSB the 1 current, etc. The theory is simple but the practical problems of manufacturing an IC of an economical size with current or resistor ratios of even 128:1 for an 8-bit DAC are significant, especially as they must have matched temperature coefficients.
If the MSB current is slightly low in value, it will be less than the sum of all the other bit currents, and the DAC will not be monotonic (the differential non-linearity of most types of DACs is worst at major bit transitions). This architecture is virtually never used on its own in integrated circuit DACs, although, again, 3- or 4-bit versions have been used as components in more complex structures.

However, there is another binary-weighted DAC structure that has recently become widely used. This uses binary-weighted capacitors as shown in Figure 3. The problem with a DAC using capacitors is that leakage causes it to lose its accuracy within a few milliseconds of being set. This may make capacitive DACs unsuitable for general purpose DAC applications, but it is not a problem in successive approximation ADCs, since the conversion is complete in a few µs or less - long before leakage has any appreciable effect.

The successive approximation ADC has a very simple structure, low power, and reasonably fast conversion times. It is probably the most widely used general-purpose ADC architecture, but in the mid-1990s the subranging ADC was starting to overtake the successive approximation type in popularity because the R-2R thin-film resistor DAC in the successive approximation ADC made the chip larger and more expensive than that of a subranging ADC, even though the subranging types tend to use more power. The development of sub-micron CMOS processes made possible very small (and therefore cheap), and very accurate switched capacitor DACs. These enabled a new generation of successive approximation ADCs to be made small, cheap, low-power and precise, and thus to regain their popularity (such as the Analog Devices’ PulSAR® family, for example).
The use of capacitive charge redistribution DACs offers another advantage as well – the DAC itself behaves as a sample-and-hold circuit (SHA), so neither an external SHA nor allocation of chip area for a separate integral SHA are required.

**R-2R DACs**

One of the most common DAC building-block structures is the R-2R resistor ladder network shown in Figure 4. It uses resistors of only two different values, and their ratio is 2:1. An N-bit DAC requires 2N resistors, and they are quite easily trimmed. There are also relatively few resistors to trim.

There are two ways in which the R-2R ladder network may be used as a DAC – known respectively as the **voltage mode** and the **current mode**. They are sometimes called "normal" mode and "inverted" mode, respectively, but as there is no consensus on whether the voltage mode or the current mode is the "normal" mode for a ladder network, so this nomenclature can be misleading. Each mode has its advantages and disadvantages.

In the voltage mode R-2R ladder DAC shown in Figure 5, the "rungs" or arms of the ladder are switched between \(V_{REF}\) and ground, and the output is taken from the end of the ladder. The output may be taken as a voltage, but the output impedance is independent of code, so it may equally well be taken as a current into a virtual ground. As mentioned earlier, this architecture
was proposed by B. D. Smith in 1953 (Reference 3).

Figure 5: Voltage-Mode R-2R Ladder Network DAC

The voltage output is an advantage of this mode, as is the constant output impedance, which eases the stabilization of any amplifier connected to the output node. Additionally, the switches switch the arms of the ladder between a low impedance \( V_{\text{REF}} \) connection and ground, which is also, of course, low impedance, so capacitive glitch currents tend not to flow in the load. On the other hand, the switches must operate over a wide voltage range (\( V_{\text{REF}} \) to ground). This is difficult from a design and manufacturing viewpoint, and the reference input impedance varies widely with code, so that the reference input must be driven from a very low impedance. In addition, the gain of the DAC cannot be adjusted by means of a resistor in series with the \( V_{\text{REF}} \) terminal.

In the current-mode R-2R ladder DAC shown in Figure 6, the gain of the DAC may be adjusted with a series resistor at the \( V_{\text{REF}} \) terminal, since in the current mode, the end of the ladder, with its code-independent impedance, is used as the \( V_{\text{REF}} \) terminal; and the ends of the arms are switched between ground (or, sometimes, an "inverted output" at ground potential) and an output line which must be held at ground potential. The normal connection of a current-mode ladder network output is to an op amp configured as current-to-voltage (I/V) converter, but stabilization of this op amp is complicated by the DAC output impedance variation with digital code. As was previously mentioned, this architecture is sometimes referred to as an "inverted R-2R" DAC.
Current-mode operation has a larger switching glitch than voltage mode since the switches connect directly to the output line(s). However, since the switches of a current-mode ladder network are always at ground potential, their design is less demanding and, in particular, their voltage rating does not affect the reference voltage rating. If switches capable of carrying current in either direction (such as CMOS devices) are used, the reference voltage may have either polarity, or may even be ac. Such a structure is one of the most common types used as a multiplying DAC (MDAC).

Since the switches are always at, or very close to, ground potential, the maximum reference voltage may greatly exceed the logic voltage, provided the switches are make-before-break – which they are in this type of DAC – and the resistors must be thin film. It is not uncommon for a CMOS MDAC to accept a $\pm 30$ V reference (or even a 60-V peak-to-peak ac reference) while working from a single 5-V supply.

In all DACs, the output is the product of the reference voltage and the digital code, so in that sense, all DACs are multiplying DACs. But some DACs use an external reference voltage which may be varied over a wide range. These are "Multiplying DACs" or MDACs where the analog output is the product of the analog input and the digital code. They are extremely useful in many different applications. A strict definition of an MDAC is that it will continue to work correctly as its reference is reduced to zero, but sometimes the term is used less stringently for DACs which work with a reference range of 10:1 or even 6:1 – a better name for devices of this type might be "semi-multiplying" DACs.

While some types of multiplying DACs will work only with references of one polarity (two quadrant) others handle bipolar (positive or negative) references, and can work with an ac signal as a reference as well. A bipolar DAC that will work with bipolar reference voltages is known as a four-quadrant multiplying DAC. Some types of MDACs are so configured that they can work with reference voltages substantially greater than their supply voltage.

Current-mode ladder networks and CMOS switches permit positive, negative, and ac $V_{\text{REF}}$ as previously shown in Figure 6. While this is a simple implementation of an MDAC, several others are possible.

Another popular form of R-2R DAC switches equal currents into the R-2R network as shown in Figure 7. This architecture was first implemented by Bernard M. Gordon at EPSCO (now Analogic, Inc.) in a vacuum tube 11-bit, 50-kSPS successive approximation ADC. Gordon’s 1955 patent application (Reference 5) describes the ADC, which was the first commercial offering of a complete converter. In this architecture the output impedance of the DAC is equal to $R$, and this structure is often used in high-speed video DACs. A distinct advantage is that only a 2:1 resistor ratio is required regardless of the resolution. In some applications, however, the relatively low output impedance of the R-2R network can be a disadvantage.
Figure 7: Equal Current Sources Switched into an R-2R Ladder Network

Figure 8 shows a DAC using binary-weighted currents switched into a load. The output impedance is high, and this architecture generally has a volt or so of output compliance. The main problem with all of the binary-weighted DACs discussed thus far is that high resolutions require large resistor ratios, making manufacture very difficult.

SOME HISTORICAL PERSPECTIVE ON MONOLITHIC DACs

In 1970 Analog Devices introduced the AD550 “µDAC” monolithic quad (4-bit) current switch building block IC shown in Figure 9. Notice that the binary-weighted currents were generated using an external thin film network – on-chip laser trimmed thin film resistor technology was not developed until several years later. The transistor areas are scaled (8:4:2:1), thereby ensuring equal
current densities in all the transistors for optimum $V_{BE}$ matching.

![Figure 9: Binary-Weighted 4-Bit DAC, the AD550 "µDAC" Quad Switch](image)

An alternative method of developing the binary-weighted currents in the quad switch is shown in Figure 10, where an R-2R ladder network connected to the transistor emitters accomplishes the binary current division.

![Figure 10: Binary-Weighted 4-Bit DAC: R/2R Ladder Network Current Setting Resistors](image)
Figure 11 shows how three AD550 quad switches with 16:1 inter-stage attenuators are connected to form a 12-bit current-output DAC. Note that the maximum required resistor ratio of 16:1 is manageable. This monolithic “quad switch” (AD550 µDAC) along with a thin film resistor network (AD850), voltage reference, and an op amp formed the popular building blocks for 12-bit DACs in the early 1970s before the complete function was available in IC form several years later. The concept for the quad switch was patented by James J. Pastoriza (1970 filing, Reference 6).

The complete 1970-vintage 12-bit DAC solution, shown in Figure 12, consists of three monolithic quad switches, a thin-film resistor network, an op amp, and a voltage reference. The matching provided by the monolithic quad switches along with the accuracy and tracking of the external thin film network provided 12-bit performance without the need for additional trimming. An interesting and complete analysis of this 12-bit DAC based on the quad switches can be found in Reference 7.
One of the problems in implementing a completely monolithic 12-bit DAC using the quad switch approach is that each 4-bit DAC requires emitter areas scaled 8:4:2:1. This requires a total of 15 unit emitter areas, and consumes a fairly large chip area. A few years after the introduction of the quad switch building block, Paul Brokaw of Analog Devices invented a technique in which only the first two current sources have an emitter scaling of 2:1. Subsequent current sources have the same unit emitter area but operate at different current densities—while still maintaining stable currents over temperature. Paul Brokaw’s classic patent (filed in 1975) describes this technique in detail, and this particular patent is probably the most referenced and cited patent relating to data converters (Reference 8).

It should be noted that the basic circuit principles set forth in these early IC DACs are still widely used today.

REFERENCES

1. Paul M. Rainey, "Facsimile Telegraph System," U.S. Patent 1,608,527, filed July 20, 1921, issued November 30, 1926. (the first PCM patent. Also shows a relay-based 5-bit electro-mechanical flash converter and a binary DAC using relays and multiple resistors).


3. B. D. Smith, "Coding by Feedback Methods," Proceedings of the I. R. E., Vol. 41, August 1953, pp. 1053-1058. (Smith uses an internal binary weighted DAC and also points out that a non-linear transfer function can be achieved by using a DAC with non-uniform bit weights, a technique which is widely used in today's voiceband ADCs with built-in companding. He was also one of the first to propose using an R/2R ladder network within the DAC core).


SEGMENTED DACS

When we are required to design a DAC with a specific performance, it may well be that no single architecture is ideal. In such cases, two or more DACs may be combined in a single higher resolution DAC to give the required performance. These DACs may be of the same type or of different types and need not each have the same resolution.

In principle, one DAC handles the MSBs, another handles the LSBs, and their outputs are added in some way. The process is known as "segmentation," and these more complex structures are called "segmented DACs". There are many different types of segmented DACs and some, but by no means all, of them will be illustrated in this tutorial.

Figure 1 shows two varieties of segmented voltage-output DAC. The architecture in Figure 1A is sometimes called a Kelvin-Varley Divider and is composed of two or more "string DACs." Since there are buffers between the first and second stages, the second string DAC does not load the first, and the resistors in this string do not need to have the same value as the resistors in the other one. All the resistors in each string, however, do need to be equal to each other or the DAC will not be linear. The examples shown have 3-bit first and second stages but for the sake of generality, let us refer to the first (MSB) stage resolution as M-bits and the second (LSB) as K-bits for a total of N = M + K bits. The MSB DAC has a string of $2^M$ equal resistors, and a string of $2^K$ equal resistors in the LSB DAC.

Buffer amplifiers have offset, of course, and this can cause non-monotonicity in a buffered segmented string DAC. In the simpler configuration of a buffered Kelvin-Varley divider buffer (Figure 1A), buffer A is always "below" (at a lower potential than) buffer B, and the extra tap labeled "A" on the LSB string DAC is not necessary. The data decoding is just two priority encoders. In this configuration, however, buffer offset can cause non-monotonicity.

But if the decoding of the MSB string DAC is made more complex so that buffer A can only be connected to the taps labeled "A"
in the MSB string DAC, and buffer B to the taps labeled “B,” then it is not possible for buffer offsets to cause non-monotonicity. Of course, the LSB string DAC decoding must change direction each time one buffer “leapfrogs” the other, and taps A and B on the LSB string DAC are alternately not used – but this involves a fairly trivial increase in logic complexity and is justified by the increased performance.

Rather than using a second string of resistors, a binary DAC can be used to generate the three LSBs as shown in Figure 1B. It is quite hard to manufacture very high resolution R-2R ladder networks – to be more accurate, it is hard to trim them to monotonicity. So it is quite common to make high resolution DACs with a ladder network for the LSBs, and some other structure for two to five of the MSBs. This voltage-output DAC (Figure 1B) consists of a 3-bit string DAC followed by a 3-bit buffered voltage-mode ladder network.

An unbuffered version of the segmented string DAC architecture is shown in Figure 2. This version is more clever in concept (and, of course, can be manufactured on CMOS processes which make resistors and switches but not amplifiers, so it may be cheaper as well). It is intrinsically monotonic. Here, the resistors in the two strings must be equal, except that the top resistor in the MSB string must be smaller \( (1/2^K) \) of the value of the others, and the LSB string has \( 2^K - 1 \) resistors rather than \( 2^K \). Because there are no buffers, the LSB string appears in parallel with the resistor in the MSB string that it is switched across and loads it. This drops the voltage across that MSB resistor by 1 LSB of the LSB DAC – which is exactly what is required. The output impedance of this DAC, being unbuffered, varies with changing digital code.

![Figure 2: Segmented Unbuffered String DACs Use Patented Architecture](image)

In order to understand this clever concept better, the actual voltages at each of the taps has been worked out and labeled for the 6-bit segmented DAC composed of two 3-bit string DACs shown in Figure 2. The reader is urged to go through this simple analysis with the second string DAC connected across any other resistor in the first string DAC and verify the numbers. A detailed mathematical analysis of the unbuffered segmented string DAC can be found in the relevant patent filed by Dennis Dempsey and Christopher Gorman of Analog Devices in 1997 (Reference 1).

Very high speed DACs for video, communications, and other HF reconstruction applications are often built with arrays of fully decoded current sources. The two or three LSBs may use binary-weighted current sources. It is extremely important that such
DACs have low distortion at high frequency, and there are several important issues to be considered in their design.

First of all, currents are never turned on and off – they are steered to one place or another. Turning a current off at high speed frequently involves inductive spikes and, in general, because of capacitance charging, it takes longer than current steering.

Secondly, it is important that the voltage change on the chip required to switch the current should be kept as small as possible. A voltage change causes more charge to flow in stray capacitances and a larger charge-coupled glitch.

Finally, the decoding must be done before the new data is applied to the DAC so that all the data is ready and can be applied simultaneously to all the switches in the DAC. This is generally implemented by using separate parallel latches for the individual switches in a fully decoded array. If all switches were to change state instantaneously and simultaneously there would be no skew glitch – by very careful design of propagation delays around the chip and time constants of switch resistance and stray capacitance the update synchronization can be made very good, and hence the glitch-related distortion is very small.

Two examples of segmented current-output DAC structures are shown in Figure 3. Figure 3A shows a resistor-based approach for the 7-bit DAC where the 3 MSBs are fully decoded, and the 4 LSBs are derived from an R-2R network. Figure 3B shows a similar implementation using current sources. The current source implementation is by far the most popular for today’s high-speed reconstruction DACs.

![Figure 3: Segmented Current-Output DACs: (A) Resistor-Based, (B) Current-Source Based](image)

It is also often desirable to utilize more than one fully-decoded thermometer section to make up the total DAC. Figure 4 shows a 6-bit DAC constructed from two fully-decoded 3-bit DACs. As previously discussed, these current switches must be driven simultaneously from parallel latches in order to minimize the output glitch.
The AD9775 14-bit, 160-MSPS (input)/400-MSPS (output) TxDAC® uses three sections of segmentation as shown in Figure 5. Other members of the AD977x-family and the AD985x-family also use this same basic core.

The first 5 bits (MSBs) are fully decoded and drive 31 equally weighted current switches, each supplying 512 LSBs of current. The next 4 bits are decoded into 15 lines which drive 15 current switches, each supplying 32 LSBs of current. The 5 LSBs are latched and drive a traditional binary-weighted DAC which supplies 1 LSB per output level. A total of 51 current switches and latches are required to implement this ultra low glitch architecture.

The basic current switching cell in the TxDAC family is made up of a differential PMOS transistor pair as shown in Figure 6. The differential pairs are driven with low-level logic to minimize switching transients and time skew. The DAC outputs are symmetrical differential currents, which help to minimize even-order distortion products (especially when driving a differential output such as a transformer or an op amp differential current-to-voltage converter).

The overall architecture of the AD977x TxDAC® family and the AD985x-DDS family is an excellent tradeoff between power/performance, and allows the entire DAC function to be implemented on a standard CMOS process with no thin-film resistors.
Figure 6: PMOS Transistor Current Switches

REFERENCES


INTRODUCTION

Oversampling and digital filtering eases the requirements on the antialiasing filter which precedes an ADC. The concept of oversampling and interpolation can be used in a similar manner with a reconstruction DAC. For instance, oversampling is common in digital audio CD players, where the basic update rate of the data from the CD is 44.1 kSPS. Early CD players used traditional binary DACs and inserted “zeros” into the parallel data, thereby increasing the effective update rate to 4-times, 8-times, or 16-times the fundamental throughput rate. The $4 \times$, $8 \times$, or $16 \times$ data stream is passed through a digital interpolation filter which generates the extra data points. The high oversampling rate moves the image frequencies higher, thereby allowing a less complex lower cost filter with a wider transition band. In addition, there is an increase in the SNR within the signal bandwidth because of the process gain. The sigma-delta DAC architecture uses a much higher oversampling rate and represents the ultimate extension of this concept and has become popular in modern CD players.

The same concept of oversampling and interpolation is also utilized in high speed DACs used in communications applications, relaxing the requirements on the output filter as well as increasing the SNR due to process gain.

OUTPUT SPECTRUM OF A RECONSTRUCTION DAC

The output of a reconstruction DAC can be represented as a series of rectangular pulses whose width is equal to the reciprocal of the clock rate as shown in Figure 1. Note that the reconstructed signal amplitude is down 3.92 dB at the Nyquist frequency, $f_c/2$. An inverse $\sin(x)/x$ filter can be used to compensate for this effect if required. The images of the fundamental signal occur as a result of the sampling function and are also attenuated by the $\sin(x)/x$ function.

![Unfiltered DAC Output Showing Images and sin(x)/x Roll Off](image)

Figure 1: Unfiltered DAC Output Showing Images and sin (x)/x Roll Off

OVERSAMPLING INTERPOLATING DACS
The basic concept of an oversampling/interpolating DAC is shown in Figure 2. The N-bit words of input data are received at a rate of $f_c$. The digital interpolation filter is clocked at an oversampling frequency of $Kf_c$, and inserts the extra data points. The effects on the output frequency spectrum are shown in Figure 2. In the Nyquist case (A), the requirements on the analog anti-imaging filter can be quite severe. By oversampling and interpolating, the requirements on the filter are greatly relaxed as shown in (B). Also, since the quantization noise is spread over a wider region with respect to the original signal bandwidth, an improvement in the signal-to-noise ratio is also achieved. By doubling the original sampling rate ($K = 2$), an improvement of 3 dB is obtained, and by making $K = 4$, an improvement of 6 dB is obtained. Early CD players took advantage of this, and generally carried the arithmetic in the digital filter to more than N-bits. Today, most DACs in CD players are sigma-delta types.

One of the earliest publications on the oversampling/interpolating DAC concept was by Ritchie, Candy, and Ninke in 1974 (Reference 1) and followed by a 1981 patent (filing date) by Mussman and Korte (Reference 2).

The following example illustrates the concept of oversampling using some actual numbers. Assume a traditional DAC is driven at an input word rate of 30 MSPS (see Figure 3A). Assume the DAC output frequency is 10 MHz. The image frequency component at $30 - 10 = 20$ MHz must be attenuated by the analog antialiasing filter, and the transition band of the filter starts at 10 MHz and ends at 20 MHz. Assume that the image frequency must be attenuated by 60 dB. The filter must therefore go from a passband corner frequency of 10 MHz to 60 dB of stopband attenuation over the transition band between 10 and 20 MHz (one octave). A filter gives approximately 6-dB attenuation per octave for each pole. Therefore, a minimum of 10 poles is required to provide the desired attenuation. Filters become even more complex as the transition band becomes narrower.
Assume that we increase the DAC update rate to 60 MSPS and insert a "zero" between each original data sample. The parallel data stream is now 60 MSPS, but we must now determine the value of the zero-value data points. This is done by passing the 60-MSPS data stream with the added zeros through a digital interpolation filter which computes the additional data points. The response of the digital filter relative to the $2 \times$ oversampling frequency is shown in Figure 3B. The analog antialiasing filter transition zone is now 10 to 50 MHz (the first image occurs at $2f_c - f_0 = 60 - 10 = 50$ MHz). This transition zone is a little greater than 2 octaves, implying that a 5- or 6-pole filter is sufficient.

The AD9773/AD9775/AD9777 (12-/14-/16-bit) series of Transmit DACs (TxDAC®) are selectable $2 \times$, $4 \times$, or $8 \times$ oversampling interpolating dual DACs, and a simplified block diagram is shown in Figure 4. These devices are designed to handle 12-/14-/16-bit input word rates up to 160 MSPS. The output word rate is 400 MSPS maximum. For an output frequency of 50 MHz, an input update rate of 160 MHz, and an oversampling ratio of $2 \times$, the image frequency occurs at 320 MHz - 50 MHz = 270 MHz. The transition band for the analog filter is therefore 50 MHz to 270 MHz. Without $2 \times$ oversampling, the image frequency occurs at 160 MHz - 50 MHz = 110 MHz, and the filter transition band is 50 MHz to 110 MHz.
Figure 4: Oversampling Interpolating TxDAC® Simplified block Diagram

Notice also that an oversampling interpolating DAC allows both a lower frequency input clock and input data rate, which are much less likely to generate noise within the system.

SIGMA-DELTA DACS

Sigma-delta DACs operate very similarly to sigma-delta ADCs, however in a sigma-delta DAC, the noise shaping function is accomplished with a digital modulator rather than an analog one.

A $\Sigma$-$\Delta$ DAC, unlike the $\Sigma$-$\Delta$ ADC, is mostly digital (see Figure 5A). It consists of an "interpolation filter" (a digital circuit which accepts data at a low rate, inserts zeros at a high rate, and then applies a digital filter algorithm and outputs data at a high rate), a $\Sigma$-$\Delta$ modulator (which effectively acts as a low pass filter to the signal but as a high pass filter to the quantization noise, and converts the resulting data to a high speed bit stream), and a 1 bit DAC whose output switches between equal positive and negative reference voltages. The output is filtered in an external analog LPF. Because of the high oversampling frequency, the complexity of the LPF is much less than the case of traditional Nyquist operation.
It is possible to use more than one bit in the $\Sigma$-$\Delta$ DAC, and this leads to the multibit architecture shown in Figure 5B. The concept is similar to that of interpolating DACs previously discussed, with the addition of the digital sigma-delta modulator. In the past, multibit DACs have been difficult to design because of the accuracy requirement on the n-bit internal DAC (this DAC, although only n-bits, must have the linearity of the final number of bits, N). The AD195x-series of audio DACs, however use a proprietary data scrambling technique (called data directed scrambling) which overcomes this problem and produces excellent performance with respect to all audio specifications.

The AD1955 multibit sigma-delta audio DAC is shown in Figure 6. The AD1955 also uses data directed scrambling, supports a multitude of DVD audio formats and has an extremely flexible serial port. THD + N is typically 110 dB.
**SUMMARY**

Oversampling used in conjunction with digital filtering is a powerful tool in modern sampled data systems. We have seen how the same fundamental theory is applicable to both ADCs and reconstruction DACs. A primary advantage is the relaxation of the requirements on the antialiasing/anti-imaging filter. Another advantage is the increase in SNR which occurs because of the process gain.

The $\Sigma\Delta$ ADC and DAC architecture is the ultimate extension of the oversampling concept and is the architecture of choice for most voiceband and audio signal processing data converter applications.

**REFERENCES**


INTRODUCTION

Usually, we have emphasized the importance of maintaining good differential and integral linearity in data converters. However, there are situations where ADCs and DACs which have been made intentionally nonlinear (but maintaining good differential linearity) are useful, especially when processing signals having a wide dynamic range.

TELECOMMUNICATIONS APPLICATIONS OF NON-LINEAR DACs AND ADCs

One of the earliest uses of nonlinear data converters was in the digitization of voiceband signals for pulse code modulation (PCM) systems. Major contributions were made at Bell Labs during the development of the T1 carrier system. The motive for the nonlinear ADCs and DACs was to reduce the total number of bits (and therefore the serial transmission rate) required to digitize voice channels. Straight linear encoding of a voice channel required 11- or 12-bits and a sampling rate of 8 kSPS. In the 1960s Bell Labs determined that 7-bit nonlinear encoding was sufficient; and later in the 1970s they went to 8-bit nonlinear encoding for better performance (References 1-6).

The nonlinear transfer function allocates more quantization levels out of the total range for small signals and fewer for large amplitude signals. In effect, this reduces the quantization noise associated with small signals (where it is most noticeable) and increases the quantization noise for larger signals (where it is less noticeable). The term "companding" is generally used to describe this form of encoding.

The logarithmic transfer function chosen is referred to as the "Bell μ-255" standard, or simply "μ-law." A similar standard developed in Europe is referred to as "A-law." The Bell μ-law allows a dynamic range of about 4000:1 using 8 bits, whereas an 8-bit linear data converter provides a range of only 256:1.

The first generation channel bank (D1) generated the logarithmic transfer function using temperature controlled resistor-diode networks for "compressors" ahead of a 7-bit linear ADC in the transmitter. Corresponding resistor-diode "expandors" having an inverse transfer function followed the 7-bit linear DAC in the receiver. The next generation D2 channel banks used nonlinear ADCs and DACs to accomplish the compression/expansion functions in a much more reliable and cost-effective manner, and eliminated the need for the temperature-controlled diode networks.

In his 1953 classic paper, B. D. Smith proposed that the transfer function of a successive approximation ADC utilizing a nonlinear internal DAC in the feedback path would be the inverse transfer function of the DAC (Reference 7). The same basic DAC could therefore be used in the ADC and also for the reconstruction DAC. Later in the 1960s and early 1970s, nonlinear ADC and DAC technology using piecewise linear approximations of the desired transfer function allowed low cost, high volume implementations (References 1-6). These nonlinear 8-bit, 8-kSPS data converters became popular telecommunications building blocks.

The nonlinear transfer function of the 8-bit DAC is first divided into 16 segments (chords) of different slopes — the slopes are determined by the desired nonlinear transfer function. The 4 MSBs determine the segment containing the desired data point, and the individual segment is further subdivided into 16 equal quantization levels by the 4 LSBs of the 8-bit word. This is shown in Figure 1 for a 6-bit DAC, where the first 3 bits identify one of the 8 possible chords, and each chord is further subdivided into 8 equal levels defined by the 3 LSBs. The 3 MSBs are generated using a nonlinear string DAC, and the 3 LSBs are generated using a 3-bit binary R-2R DAC.
In 1982, Analog Devices introduced the LOGDAC® AD7111 monolithic multiplying DAC featuring wide dynamic range using a logarithmic transfer function. The basic DAC in the LOGDAC is a linear 17-bit current-mode "inverted" R-2R DAC preceded by an 8-bit input decoder (see Figure 2). The LOGDAC can attenuate an analog input signal, VIN, over the range 0 dB to 88.5 dB in 0.375 dB steps. The degree of attenuation across the DAC is determined by an nonlinear-coded 8-bit word applied to the onboard decode logic. This 8-bit word is mapped into the appropriate 17-bit word, which is then applied to a 17-bit R-2R ladder. A functional diagram of the LOGDAC is shown in Figure 2. In addition to providing the logarithmic transfer function, the LOGDAC also acts as a full four-quadrant multiplying DAC.

With the introduction of high resolution linear ADCs and DACs, the method used in the LOGDAC® is widely used today to implement various nonlinear transfer functions such as the µ-law and A-law companding functions required for telecommunications and other applications. Figure 3 shows a general block diagram of the modern approach.
The μ-law or A-law companded input data is mapped into data points on the transfer function of a high resolution DAC. This mapping can be easily accomplished by a simple lookup table in either hardware, software, or firmware. A similar nonlinear ADC can be constructed by digitizing the analog input signal using a high resolution ADC and mapping the data points into a shorter word using the appropriate transfer function. A big advantage of this method is that the transfer curve does not have to be approximated with straight line segments as in the earlier method, thereby providing more accuracy.

REFERENCES


7. B. D. Smith, "Coding by Feedback Methods," *Proceedings of the I. R. E.*, Vol. 41, August 1953, pp. 1053-1058. *(Smith uses an internal binary weighted DAC and also points out that a non-linear transfer function can be achieved by using a DAC with non-uniform bit weights, a technique which is widely used in today’s voiceband ADCs with built-in companding. He was also one of the first to propose using an R/2R ladder network within the DAC core)*.
INTRODUCTION

This tutorial outlines some important issues regarding DAC interface circuitry including the voltage reference, analog output, data input, and clock driver. Because ADCs require references and clocks also, most of the concepts presented in this tutorial regarding these subjects apply equally to ADCs.

DAC REFERENCE VOLTAGE

There is a tendency to regard DACs simply as devices with digital inputs and an analog output. But the analog output depends on the presence of that analog input known as the reference, and the accuracy of the reference is almost always the limiting factor on the absolute accuracy of a DAC. Design tools such as the Voltage Reference Wizard are useful in matching references to data converters. These tools and others are available on the Design Center portion of the Analog Devices' website.

Some ADCs and DACs have internal references, while others do not. Some ADCs use the power supply as a reference. Unfortunately, there is little standardization with respect to ADC/DAC voltage references. In some cases, the dc accuracy of a converter with an internal reference can often be improved by overriding or replacing the internal reference with a more accurate and stable external one. In other cases, the use of an external low-noise reference will also increase the noise-free code resolution of a high-resolution ADC.

Various ADCs and DACs provide the capability to use external references in lieu of internal ones in various ways. Figure 1 shows some of the popular configurations (but certainly not all). Figure 1A shows a converter which requires an external reference. It is generally recommended that a suitable decoupling capacitor be added close to the ADC/DAC REF IN pin. The appropriate value is usually specified in the voltage reference data sheet. It is also important that the reference be stable with the required capacitive load (more on this to come).

Figure 1B shows a converter that has an internal reference, where the reference is also brought out to a pin on the device. This allows it to be used other places in the circuit, provided the loading does not exceed the rated value. Again, it is important to place the capacitor close to the converter pin. If the internal reference is pinned out for external use, its accuracy, stability, and temperature coefficient is usually specified on the ADC or DAC data sheet.
If the reference output is to be used other places in the circuit, the data sheet specifications regarding fanout and loading must be strictly observed. In addition, care must be taken in routing the reference output to minimize noise pickup. In many cases, a suitable op amp buffer should be used directly at the REF OUT pin before fanning out to various other parts of the circuit.

Figure 1C shows a converter which can use either the internal reference or an external one, but an extra package pin is required. If the internal reference is used, as in Figure 1C, REF OUT is simply externally connected to REF IN, and decoupled if required. If an external reference is used as shown in Figure 1D, REF OUT is left floating, and the external reference decoupled and applied to the REF IN pin. This arrangement is quite flexible for driving similar ADCs or DACs with the same reference in order to obtain good tracking between the devices.

Figure 1E shows an arrangement whereby an external reference can override the internal reference using a single package pin. The value of the resistor, R, is typically a few kΩ, thereby allowing the low impedance external reference to override the internal one when connected to the REF OUT/IN pin. Figure 1F shows how the external reference is connected to override the internal reference.

The arrangements shown in Figure 1 are by no means the only possible configurations for ADC and DAC references, and the individual data sheets should be consulted in all cases for details regarding options, fanout, decoupling, etc.

Although the reference element itself can be either a bandgap, buried zener, or XFET™, practically all references have some type of output buffer op amp. The op amp isolates the reference element from the output and also provides drive capability. However, this op amp must obey the general laws relating to op amp stability, and that is what makes the topic of reference decoupling relevant to the discussion.

Note that a reference input to an ADC or DAC is similar to the analog input of an ADC, in that the internal conversion process can inject transient currents at that pin. This requires adequate decoupling to stabilize the reference voltage. Adding such decoupling might introduce instability in some reference types, depending on the output op amp design. Of course, a reference data sheet may not show any details of the output op amp, which leaves the designer in somewhat of a dilemma concerning whether or not it will be stable and free from transient errors. In many cases, the ADC or DAC data sheet will recommend appropriate external references and the recommended decoupling network.
A well-designed voltage reference is stable with heavy capacitive decoupling. Unfortunately, some are not, and larger capacitors actually increases the amount of transient ringing. Such references are practically useless in data converter applications, because some amount of local decoupling is almost always required at the converter.

A suitable op amp buffer might be added between the reference and the data converter. However, there are many good references available which are stable with an output capacitor. This type of reference should be chosen for a data converter application, rather than incurring the further complication and expense of an op amp.

**DAC ANALOG OUTPUT CONSIDERATIONS**

The analog output of a DAC may be a voltage or a current. In either case it may be important to know the output impedance. If the voltage output is buffered, the output impedance will be low. Both current outputs and unbuffered voltage outputs will be high (er) impedance and may well have a reactive component specified as well as a purely resistive one. Some DAC architectures have output structures where the output impedance is a function of the digital code on the DAC – this should be clearly noted on the data sheet.

In theory, current outputs should be connected to zero ohms at ground potential. In real life they will work with non-zero impedances and voltages. Just how much deviation they will tolerate is defined under the heading "compliance" and this specification should be heeded when terminating current-output DACs.

Most high-speed DACs suitable for video, RF, or IF, have current outputs which are designed to drive source and load-terminated cables directly. For instance, a 20-mA current output DAC can develop 0.5 V across a 25-Ω load (the equivalent dc resistance of a 50-Ω source and load terminated cable). In most cases, single-supply high-speed CMOS DACs have a positive output compliance of at least +1 V and a negative output compliance of a few hundred millivolts.

In many cases, such as the TxDAC® family, both true and complementary current outputs are available. The differential outputs can drive the primary winding of a transformer directly, and a single-ended signal can be developed at the secondary winding by grounding one side of the output winding. This method will often give better distortion performance at high frequencies than simply taking the output signal directly from one of the DAC current outputs and grounding the other.

Modern current output DACs usually have differential outputs, to achieve high common-mode rejection and reduce the even-order distortion products. Fullscale output currents in the range of 2 mA to 30 mA are common.

In many applications, it is desirable to convert the differential output of the DAC into a single-ended signal, suitable for driving a coax line. This can be readily achieved with an RF transformer, provided low frequency response is not required. Figure 2 shows a typical example of this approach. The high impedance current output of the DAC is terminated differentially with 50 Ω, which defines the source impedance to the transformer as 50 Ω.

The resulting differential voltage drives the primary of a 1:1 RF transformer, to develop a single-ended voltage at the output of the secondary winding. The output of the 50-Ω LC filter is matched with the 50-Ω load resistor \( R_L \), and a final output voltage of 1-Vp-p is developed.
The transformer not only serves to convert the differential output into a single-ended signal, but it also isolates the output of the DAC from the reactive load presented by the LC filter, thereby improving overall distortion performance.

An op amp connected as a differential to single-ended converter can be used to obtain a single-ended output when frequency response to dc is required. In Figure 3 the AD8055 op amp is used to achieve high bandwidth and low distortion. The current output DAC drives balanced 25-Ω resistive loads, thereby developing an out-of-phase voltage of 0 to +0.5 V at each output. This technique is used in lieu of a direct I/V conversion to prevent fast slewing DAC currents from overloading the amplifier and introducing distortion. Care must be taken so that the DAC output voltage is within its compliance rating.

The AD8055 is configured for a gain of 2, to develop a final single-ended ground-referenced output voltage of 2-V p-p. Note that because the output signal swings above and below ground, a dual-supply op amp is required.
A modified form of the Figure 3 circuit can be operated on a single supply, provided the common-mode voltage of the op amp is set to mid-supply (+2.5 V). This is shown in Figure 4, where the AD8061 op amp is used. The output voltage is 2-Vp-p centered around a common-mode voltage of +2.5 V. This common-mode voltage can be either developed from the +5 V supply using a resistor divider, or directly from a +2.5 V voltage reference. If the +5 V supply is used as the common-mode voltage, it must be heavily decoupled to prevent supply noise from being amplified.

Figure 4: Differential DC Coupled Output Using a Single-Supply Op Amp

SINGLE-ENDED CURRENT-TO-VOLTAGE CONVERSION

Single-ended current-to-voltage conversion is easily performed using a single op amp as an I/V converter, as shown in Figure 5. The 10-mA full scale DAC current from the AD768 develops a 0 to +2 V output voltage across the 200-Ω \( R_F \) resistor.
Driving the virtual ground of the AD8055 op amp minimizes any distortion due to nonlinearity in the DAC output impedance. In fact, most high resolution DACs of this type are factory trimmed using an I/V converter.

It should be recalled, however, that using the single-ended output of the DAC in this manner will cause degradation in the common-mode rejection and increased second-order distortion products, compared to a differential operating mode.

The \( C_F \) feedback capacitor should be optimized for best pulse response in the circuit. The equations given in the diagram should only be used as guidelines.

An R-2R based current-output DAC has a code-dependent output impedance – therefore, its output must drive the virtual ground of an op amp in order to maintain linearity. The AD5545/AD5555 16-/14-bit DAC is an excellent example of this architecture. A suitable interface circuit is shown in Figure 6 where the ADR03 is used as a 2.5-V voltage reference, and the AD8628 chopper-stabilized op amp is used as an output I/V converter.
The external 2.5-V references determines the fullscale output current, 0.5 mA. Note that a 5-kΩ feedback resistor is included in the DAC, and using it will enhance temperature stability as opposed to using an external resistor. The fullscale output voltage from the op amp is therefore -2.5 V. The $C_F$ feedback capacitor compensates for the DAC output capacitance and should be selected to optimize the pulse response, with 20 pF a typical starting point.

**DIFFERENTIAL CURRENT-TO-DIFFERENTIAL VOLTAGE CONVERSION**

If a buffered differential voltage output is required from a current output DAC, the AD813x-series of differential amplifiers can be used as shown in Figure 7.

---

**Figure 6: AD5545/AD5555 Dual 16-/14-Bit R-2R Current Output DAC Interface**

**Figure 7: Buffering High Speed DACs Using the AD8138 Differential Amplifier**
The DAC output current is first converted into a voltage that is developed across the 25-Ω resistors. The voltage is amplified by a factor of 5 using the AD8138. This technique is used in lieu of a direct I/V conversion to prevent fast slewing DAC currents from overloading the amplifier and introducing distortion. Care must be taken so that the DAC output voltage is within its compliance rating.

The $V_{OCM}$ input on the AD8138 can be used to set a final output common-mode voltage within the range of the AD8138. Adding a pair of 75-Ω series output resistors will allow transmission lines to be driven.

**DAC DATA INPUT CONSIDERATIONS**

The earliest monolithic DACs contained little, if any, logic circuitry, and parallel data had to be maintained on the digital input to maintain the digital output. Today almost all DACs are latched, and data need only be written to them, not maintained. Some even have nonvolatile latches and remember settings while turned off.

There are innumerable variations of DAC input structure, which will not be discussed here, but nearly all are described as "double-buffered." A double-buffered DAC has two sets of latches. Data is initially latched in the first rank and subsequently transferred to the second as shown in Figure 8. There are several reasons why this arrangement is useful.

![Figure 8: Double-Buffered DAC Permits Complex Input Structures and Simultaneous Update](image)

The DAC DATA INPUT CONSIDERATIONS

The first is that it allows data to enter the DAC in many different ways. A DAC without a latch, or with a single latch, must be loaded in parallel with all bits at once, since otherwise its output during loading may be totally different from what it was, or what it is to become. A double-buffered DAC, on the other hand, may be loaded with parallel data, or with serial data, with 4-bit or 8-bit words, or whatever, and the output will be unaffected until the new data is completely loaded and the DAC receives its update instruction.

A second advantage of a double-buffered DAC is that the time skew between the individual switches is minimized by driving all the switches in parallel with a single latch which is updated at the DAC output data rate. This minimizes the glitch impulse and improves distortion performance.

The third convenience of the double-buffered structure is that many DACs may be updated simultaneously. Data is loaded into the first rank of each DAC in turn, and when all is ready, the output buffers of all the DACs are updated at once. There are many DAC applications where the output of a number of DACs must change simultaneously, and the double-buffered structure allows this to be done very easily.

Most early monolithic high resolution DACs had parallel or byte-wide data ports and tended to be connected to parallel data buses and address decoders and addressed by microprocessors as if they were very small write-only memories. (Some parallel DACs are not write-only, but can have their contents read as well – this is convenient for some applications, but is not very common.) A DAC connected to a data bus is vulnerable to capacitive coupling of logic noise from the bus to the analog output, and therefore many DACs today have serial data structures. These are less vulnerable to such noise (since fewer noisy pins are
involved), use fewer pins and therefore take less board space, and are frequently more convenient for use with modern microprocessors, most of which have serial data ports. Some, but not all, of such serial DACs have both data outputs and data inputs so that several DACs may be connected in series, with data clocked to all of them from a single serial port. This arrangement is often referred to as "daisy-chaining."

Serial DACs can be used at voiceband and audio frequency update rates. For instance, 24-bit digital audio updated at 192 kSPS requires a serial port transfer rate of at least $24 \times 192 \text{ kSPS} = 46.08 \text{ MSPS}$, which is easily handled by CMOS logic. However where high update rates are involved, parallel DACs must be used since the required transfer rate of the serial data would be too high.

For parallel data rates greater than approximately 100 MSPS, low-level current-mode differential logic (PECL, reduced-level PECL, LVDS, etc.) is often used because it is much less likely to generate transient glitches than CMOS logic levels (see Figure 9). This helps minimize distortion generated by code-dependent glitches. For instance, the AD9734/AD9735/AD9736 DAC family operates at 1.2 GSPS and accepts LVDS input logic levels. Special circuitry is included on-chip to ensure the proper timing of the input data with respect to the DAC clock.

**Figure 9: LVDS Driver**

**DAC CLOCK CONSIDERATIONS**

It was shown in Tutorial MT-007 that the relationship between ADC broadband aperture jitter, $t_j$, converter SNR, and fullscale sinewave analog input frequency, $f$, is given by

$$\text{SNR} = 20 \log_{10} \left[ \frac{1}{2\pi f t_j} \right].$$

**Eq. 1**

The same relationship is applicable to reconstruction DACs. The equation assumes an ideal ADC/DAC, where the only error source is clock jitter. The bandwidth for the SNR measurement is the Nyquist bandwidth, dc to $f_c/2$, where $f_c$ is the DAC update rate. Note that Eq. 1 also assumes a fullscale sinewave output. The error due to jitter is proportional to the slew rate of the output
signal – lower amplitude sinewaves with proportionally lower slew rate yield higher values of SNR (with respect to fullscale).

It should be noted that $t_j$ in Eq. 1 is the combined jitter of the sampling clock, $t_{jc}$, and the ADC internal aperture jitter, $t_{ja}$ – these terms are not correlated and therefore combine on an root-sum-square (rss) basis:

$$t_j = \sqrt{t_{jc}^2 + t_{ja}^2}.$$  
Eq. 2

High-speed reconstruction DACs, on the other hand, do not have specifications for internal aperture jitter because they have no internal sample-and-hold amplifier. Although there is an internal clock jitter component in a DAC, it is generally not measured or specified, because the jitter of the external clock is the dominant jitter source.

Figure 10 plots Eq. 1 and graphically illustrates how SNR is degraded by jitter for various fullscale analog output frequencies (note that we assume $t_j$ includes all jitter sources, including the internal DAC jitter). For instance, maintaining 12-bit SNR (74 dB) for a 70 MHz IF output frequency requires the clock jitter to be less than 0.45 ps (using Eq. 1).

From Tutorial MT-001, it was shown that there is a very useful relationship between effective number of bits (ENOB) and the signal-to-noise-plus-distortion ratio (SINAD) given by:

$$\text{ENOB} = \frac{\text{SINAD} - 1.76 \text{ dB}}{6.02 \text{ dB}}.$$  
Eq. 3

For the purposes of this discussion, assume that the DAC has no distortion, and therefore SINAD = SNR, so Eq. 3 becomes:

$$\text{ENOB} = \frac{\text{SNR} - 1.76 \text{ dB}}{6.02 \text{ dB}}.$$  
Eq. 4
The SNR values on the left-hand vertical axis of Figure 10 have been converted into ENOB values on the right-hand vertical axis using Eq. 4.

In order to illustrate the significance of these jitter numbers, consider the typical rms jitter associated with a selection of logic gates shown in Figure 11. The values for the 74LS00, 74HCT00, and 74ACT00 were measured with a high performance ADC (aperture jitter less than 0.2-ps rms) using the method described in Chapter 5 of Reference 1, where the jitter was calculated from FFT-based SNR degradation due to several identical gates connected in series. The jitter due to a single gate was then calculated by dividing by the square root of the total number of series-connected gates. The jitter for the MC100EL16 and NBSG16 was specified by the manufacturer.

![Figure 11: RMS Jitter of Typical Logic Gates](image)

Figure 12 shows the same data as Figure 10 but plots maximum allowable jitter as a function of analog output frequency for various resolution requirements. This graph should serve as an approximate guideline for selecting the type of sampling clock generator based upon the maximum output frequency and the required resolution in ENOB. The PLL approach with a standard VCO is an excellent one for generating sampling clocks where the rms jitter requirement is approximately 1 ps or greater. However, sub-picosecond jitter requires either a VCXO-based PLL or a dedicated low noise crystal oscillator. Tutorial MT-008 explains how to convert oscillator phase noise into jitter.
This section has described the effects of jitter on SNR, assuming that the jitter is solely a combination of the internal DAC jitter and the external clock jitter. However, improper layout, grounding, and decoupling techniques can create additional clock jitter which can drastically degrade dynamic performance, regardless of the specifications of the DAC or sampling clock oscillator.

Routing the sampling clock signal in parallel with noisy digital signals is sure to degrade performance due to stray coupling. In fact, coupling high speed data from parallel output ADCs into the sampling clock not only increases noise, but is likely to create additional harmonic distortion, because the energy contained in the digital output transient currents is signal dependent. For further discussion of these and other critical hardware design techniques, the reader is referred to Chapter 9 of Reference 1.

REFERENCE

INTRODUCTION

Commercial flash converters appeared in instruments and modules of the 1960s and 1970s and quickly migrated to integrated circuits during the 1980s. The monolithic 8-bit flash ADC became an industry standard in digital video applications of the 1980s. Today, the flash converter is primarily used as a building block within subranging "pipeline" ADCs. The lower power, lower cost pipeline architecture is capable of 8- to 10-bits of resolution at sampling rates of several hundred MHz. Therefore, higher power stand-alone flash converters are primarily used in 6- or 8-bit ADCs requiring sampling rates greater than 1 GHz. These converters are usually designed on Gallium Arsenide processes.

Because of their importance as building blocks in high resolution pipeline ADCs, it is important to understand the fundamentals of the basic flash converter. This tutorial begins with a brief discussion of the comparator which is the basic building block for flash converters.

THE COMPARATOR: A 1-BIT ADC

As a changeover switch is a 1 bit DAC, so a comparator is a 1 bit ADC (see Figure 1). If the input is above a threshold, the output has one logic value, below it has another. Moreover, there is no ADC architecture which does not use at least one comparator of some sort.

Figure 1: The Comparator: A 1-Bit ADC
The most common comparator has some resemblance to an operational amplifier in that it uses a differential pair of transistors or FETs as its input stage, but unlike an op amp, it does not use external negative feedback, and its output is a logic level indicating which of the two inputs is at the higher potential. Op amps are not designed for use as comparators—they may saturate if overdriven and recover slowly. Many op amps have input stages which behave in unexpected ways when used with large differential voltages, and their outputs are rarely compatible with standard logic levels. There are cases, however, when it may be desirable to use an op amp as a comparator, and an excellent treatment of this subject can be found in Reference 1.

Comparators used as building blocks in ADCs need good resolution which implies high gain. This can lead to uncontrolled oscillation when the differential input approaches zero. In order to prevent this, "hysteresis" is often added to comparators using a small amount of positive feedback. Figure 1 shows the effects of hysteresis on the overall transfer function. Many comparators have a millivolt or two of hysteresis to encourage "snap" action and to prevent local feedback from causing instability in the transition region. Note that the resolution of the comparator can be no less than the hysteresis, so large values of hysteresis are generally not useful.

Early comparators were designed with vacuum tubes and were often used in radio receivers—where they were called "discriminators," not comparators. Most modern comparators used in ADCs include a built-in latch which makes them sampling devices suitable for data converters. A typical structure is shown in Figure 2 for the AM685 ECL (emitter-coupled-logic) latched comparator introduced in 1972 by Advanced Micro Devices, Inc. (see Reference 2). The input stage preamplifier drives a cross-coupled latch. The latch locks the output in the logic state it was in at the instant when the latch was enabled. The latch thus performs a track-and-hold function, allowing short input signals to be detected and held for further processing. Because the latch operates directly on the input stage, the signal suffers no additional delays—signals only a few nanoseconds wide can be acquired and held. The latched comparator is also less sensitive to instability caused by local feedback than an unlatched one.

![Figure 2: The AM685 ECL Comparator (1972)](image)

Where comparators are incorporated into IC ADCs, their design must consider resolution, speed, overload recovery, power dissipation, offset voltage, bias current, and the chip area occupied by the architecture which is chosen. There is another subtle but troublesome characteristic of comparators which can cause large errors in ADCs if not understood and dealt with effectively. This error mechanism is the occasional inability of a comparator to resolve a small differential input into a valid output logic level. This phenomenon is known as "metastability"—the ability of a comparator to balance right at its threshold for a short period of
The metastable state problem is illustrated in Figure 3. Three conditions of differential input voltage are illustrated: (1) large differential input voltage, (2) small differential input voltage, and (3) zero differential input voltage. The approximate equation which describes the output voltage, $V_O(t)$ is given by:

$$V_O(t) = \Delta V_{IN} A e^{t/\tau}, \quad \text{Eq. 1}$$

Where $\Delta V_{IN}$ = the differential input voltage at the time of latching, $A$ = the gain of the preamp at the time of latching, $\tau$ = regeneration time constant of the latch, and $t$ = the time that has elapsed after the comparator output is latched (see References 3 and 4).

For small differential input voltages, the output takes longer to reach a valid logic level. If the output data is read when it lies between the “valid logic 1” and the “valid logic 0” region, the data can be in error. If the differential input voltage is exactly zero, and the comparator is perfectly balanced at the time of latching, the time required to reach a valid logic level can be quite long (theoretically infinite). However, hysteresis and noise on the input makes this condition highly unlikely. The effects of invalid logic levels out of the comparator are different depending upon how the comparator is used in the actual ADC.

From a design standpoint, comparator metastability can be minimized by making the gain, $A$, high, minimizing the regeneration time constant, $\tau$, by increasing the gain-bandwidth of the latch, and allowing sufficient time, $t$, for the output of the comparator to settle to a valid logic level. It is not the purpose of this discussion to analyze the complex tradeoffs between speed, power, and circuit complexity when optimizing comparator designs, but an excellent treatment of the subject can be found in References 3 and 4.

From a user standpoint, the effect of comparator metastability (if it affects the ADC performance at all) is in the “bit error rate” (BER)—which is not usually specified on most ADC data sheets. The resulting errors are often referred to as “sparkle codes”, "rabbits", or "flyers."

Bit error rate should not be a problem in a properly designed ADC in most applications, however the system designer should be aware that the phenomenon exists. An application example where it can be a problem is when the ADC is used in a digital oscilloscope to detect small-amplitude single-shot randomly occurring events. The ADC can give false indications if its BER is not sufficiently small. More discussion of sparkle codes can be found in Tutorial MT-011.
FLASH CONVERTERS

Flash ADCs (sometimes called "parallel" ADCs) are the fastest type of ADC and use large numbers of comparators. An N-bit flash ADC consists of $2^N$ resistors and $2^N - 1$ comparators arranged as in Figure 4. Each comparator has a reference voltage from the resistor string which is 1 LSB higher than that of the one below it in the chain. For a given input voltage, all the comparators below a certain point will have their input voltage larger than their reference voltage and a "1" logic output, and all the comparators above that point will have a reference voltage larger than the input voltage and a "0" logic output. The $2^N - 1$ comparator outputs therefore behave in a way analogous to a mercury thermometer, and the output code at this point is sometimes called a "thermometer" code. Since $2^N - 1$ data outputs are not really practical, they are processed by a decoder to generate an N-bit binary output.

![3-bit All-Parallel (Flash) Converter](image)

The input signal is applied to all the comparators at once, so the thermometer output is delayed by only one comparator delay from the input, and the encoder N-bit output by only a few gate delays on top of that, so the process is very fast. In addition, the individual comparators provide an inherent "sample-and-hold" function, so theoretically a flash converter does not need a separate SHA, provided the comparators are perfectly dynamically matched. In practice, however, the addition of a proper external sample-and-hold usually enhances the dynamic performance of most flash converters because of the inevitable slight timing mismatches which occur between comparators.

Because the flash converter uses large numbers of resistors and comparators and is limited to low resolutions, and if it is to be fast, each comparator must run at relatively high power levels. Hence, the problems of flash ADCs include limited resolution, high power dissipation because of the large number of high speed comparators (especially at sampling rates greater than 50 MSPS), and relatively large (and therefore expensive) chip sizes. In addition, the resistance of the reference resistor chain must be kept low to supply adequate bias current to the fast comparators, so the voltage reference has to source quite large currents (typically > 10 mA).

TYPICAL FLASH CONVERTER TIMING

Simplified timing for a typical commercial flash converter (AD9048 8-bit, 35 MSPS) is shown in Figure 5. The input comparators...
are in the "track" or "transparent" mode when the sampling clock is low. The rising edge of the sampling clock places the comparators in the "hold" or "latched" mode. During the "hold" time, the decoding logic makes its decision based on the comparator outputs. The falling edge of the sampling clock latches the decoded data into an intermediate latch. The next rising edge of the sampling clock transfers the decoded data into an output latch. Note that this results in one cycle of "pipeline delay" in the output data with respect to the corresponding sampling clock edge. The intermediate latch allows for more sophisticated two-stage decoding methods. For instance, the comparator output data might first be decoded as a Gray code, latched on the falling edge of the sampling clock, and converted to binary during the "track" interval. The two-stage decoding is often used to minimized "sparkle codes" which are due to incorrectly interpreting a comparator output. (See Tutorial MT-011 for a complete discussion of sparkle codes and metastable state errors). Some flash converters use even more sophisticated decoding and therefore have more than one clock cycle of pipeline delay.

Figure 5: Data Timing for Typical Flash Converter (AD9048 8-bit, 35 MSPS)

If simple priority decoding is used, it would be possible to eliminate both the output latch and the intermediate latch and take the binary data directly from the output of the decoding logic. If this were the case, however, the output data is constantly changing during the "track" interval, thereby limiting the "DATA VALID" interval to one-half of the sampling clock period. It is therefore customary to use at least one latch so that the output data stays constant during the entire sampling period, with the exception of the small amount of "DATA CHANGING" time shown in Figure 5.

FLASH CONVERTER HISTORICAL PERSPECTIVE

The first documented flash converter was part of Paul M. Rainey's electro-mechanical PCM facsimile system described in a relatively ignored patent filed in 1921 (Reference 5). In the ADC, a current proportional to the intensity of light drives a galvanometer which in turn moves another beam of light which activates one of 32 individual photocells, depending upon the amount of galvanometer deflection. Each individual photocell output activates part of a relay network which generates the 5-bit binary code as shown in Figure 6.
A significant development in high speed ADC technology during the 1940s was the electron beam coding tube developed at Bell Labs and shown in Figure 7. The tube described by R. W. Sears in Reference 6 was capable of sampling at 96 kSPS with 7-bit resolution. The basic electron beam coder concepts are shown in Figure 6 for a 4-bit device. The tube used a fan-shaped beam creating a "flash" converter delivering a parallel output word.

Figure 6: A 5-Bit Flash ADC Proposed by Paul Rainey
Adapted from Paul M. Rainey, "Facsimile Telegraph System," U.S. Patent 1,608,527, Filed July 20, 1921, Issued November 30, 1926
Early electron tube coders used a binary-coded shadow mask (Figure 7A), and large errors can occur if the beam straddles two adjacent codes and illuminates both of them. The errors associated with binary shadow masks were later eliminated by using a Gray code shadow mask as shown in Figure 7B. This code was originally called the "reflected binary" code, and was invented by Elisha Gray in 1878, and later re-invented by Frank Gray in 1949 (see Reference 7). The Gray code has the property that adjacent levels differ by only one digit in the corresponding Gray-coded word. Therefore, if there is an error in a bit decision for a particular level, the corresponding error after conversion to binary code is only one least significant bit (LSB). In the case of midscale, note that only the MSB changes. It is interesting to note that this same phenomenon can occur in modern comparator-based flash converters due to comparator metastability. With small overdrive, there is a finite probability that the output of a comparator will generate the wrong decision in its latched output, producing the same effect if straight binary decoding techniques are used. In many cases, Gray code, or "pseudo-Gray" codes are used to decode the comparator bank output before finally converting to a binary code output.

In spite of the many mechanical and electrical problems relating to beam alignment, electron tube coding technology reached its peak in the mid-1960s with an experimental 9-bit coder capable of 12-MSPS sampling rates (Reference 8). Shortly thereafter, however, advances in all solid-state ADC techniques made the electron tube technology obsolete.

It was soon recognized that the flash converter offered the fastest sampling rates compared to other architectures, but the problem with this approach is that the comparator circuit itself is quite bulky using discrete transistor circuits and very cumbersome using vacuum tubes. Constructing a single latched comparator cell using either technology is quite a task, and extending it to even 4-bits of resolution (15 comparators required) makes it somewhat unreasonable. Nevertheless, work was done in the mid 1950s and early 1960s as shown in Robert Staffin and Robert D. Lohman's patent which describes a subranging architecture using both tube and transistor technology (Reference 9). The patent discusses the problem of the all-parallel approach and points out the savings by dividing the conversion process into a coarse conversion followed by a fine conversion.

Tunnel (Esaki) diodes were used as comparators in several experimental early flash converters in the 1960s as an alternative to a latched comparator based solely on tubes or transistors (see References 10-13).

In 1964 Fairchild introduced the first IC comparators, the µA711/712, designed by Bob Widlar. The same year, Fairchild also introduced the first IC op amp, the µA709—another Widlar design. Other IC comparators soon followed including the Signetics
With the introduction of these building block comparators and the availability of TTL and ECL logic ICs, 6-bit rack-mounted discrete flash converters were introduced by Computer Labs, Inc., including the VHS-630 (6-bit, 30 MSPS in 1970) and the VHS-675 (6-bit, 75 MSPS in 1975). The VHS-675 shown in Figure 8 used 63 AM685 ECL comparators preceded by a high-speed track-and-hold, ECL decoding logic, contained a built-in linear power supply (ac line powered), and dissipated a total of 130 W (sale price was about $10,000 in 1975). Instruments such as these found application in early high speed data acquisition applications including military radar receivers.

The AM685 comparator was also used as a building block in the 4-bit 100-MSPS board-level flash ADC, the MOD-4100, introduced in 1975 and shown in Figure 9.
The first integrated circuit 8-bit video-speed 30-MSPS flash converter, the TDC1007J, was introduced by TRW LSI division in 1979 (References 14 and 15). A 6-bit version of the same design, the TDC1014J followed shortly. Also in 1979, Advanced Micro Devices, Inc. introduced the AM6688, a 4-bit 100-MSPS IC flash converter.

Monolithic flash converters became very popular in the 1980s for high speed 8-bit video applications as well as building blocks for higher resolution subranging card-level, modular, and hybrid ADCs. Examples from Analog Devices included the popular AD9048 (8-bit, 35 MSPS) and the AD9002 (8-bit, 150 MSPS). Many flash converters were fabricated on CMOS processes for lower power dissipation. Recently, however, the subranging pipeline architecture has become popular for 8-bit ADCs up to about 250 MSPS. For instance, the AD9480 8-bit 250-MSPS ADC is fabricated on a high speed BiCMOS process and dissipates less than 400mW compared to the several watts required for a full flash implementation on a similar process.

In practice, IC flash converters are currently available up to 10 bits, but more commonly they have 6- or 8 bits of resolution. Their maximum sampling rate can be as high as 1 GHz (these are generally made on Gallium Arsenide processes with several watts of power dissipation), with input full power bandwidths in excess of 300 MHz.

But as mentioned earlier, full power bandwidths are not necessarily full resolution bandwidths. Ideally, the comparators in a flash converter are well matched both for dc and ac characteristics. Because the sampling clock is applied to all the comparators simultaneously, the flash converter is inherently a sampling converter. In practice, there are delay variations between the comparators and other ac mismatches which cause a degradation in the effective number of bits (ENOBs) at high input frequencies. This is because the inputs are slewing at a rate comparable to the comparator conversion time. For this reason, track-and-holds are often required ahead of flash converters to achieve high SFDR on high frequency input signals.

The input to a flash ADC is applied in parallel to a large number of comparators. Each has a voltage-variable junction capacitance, and this signal-dependent capacitance results in most flash ADCs having reduced ENOB and higher distortion at high input frequencies. For this reason, most flash converters must be driven with a wideband op amp which is tolerant to the capacitive load presented by the converter as well as high speed transients developed on the input.

Comparator metastability in a flash converter can severely impact the bit error rate (BER). Figure 10 shows a simple flash converter with one stage of binary decoding logic. The two-input AND gates convert the thermometer code output of the parallel comparators into a "one-hot out of 7" code. The decoding logic is simply a "wired-or" array, a technique popular with emitter-
coupled logic (ECL). Assume that the comparator labeled "X" has metastable outputs labeled "X". The desired output code should be either 011 or 100, but note that the 000 code (both gate outputs high) and the 111 code (both gate outputs low) are also possible due to the metastable states, representing a ½ FS error.

Figure 10: Metastable Comparator Output States May Cause Error Codes in Data Converters

Metastable state errors in flash converters can be reduced by several techniques, one of which involves decoding the comparator outputs in Gray code followed by a Gray-to-binary conversion as in the Bell Labs electron beam encoder previously described. The advantage of Gray code decoding is that a metastable state in any of the comparators can produce only a 1-LSB error in the Gray code output. The Gray code is latched and then converted into a binary code which, in turn, will only have a maximum of 1-LSB error as shown in Figure 11.

The same principles have been applied to several modern IC flash converters to minimize the effects of metastable state errors as described in References 3, 16, 17, for example.
Power dissipation is always a big consideration in flash converters, especially at resolutions above 8 bits. A clever technique was used in the AD9410 10-bit, 210-MSPS ADC called "interpolation" to minimize the number of preamplifiers in the flash converter comparators and also reduce the power. The method is shown in Figure 12 (see Reference 18).
The preamplifiers (labeled "A1", "A2", etc.) are low-gain $g_m$ stages whose bandwidth is proportional to the tail currents of the differential pairs. Consider the case for a positive-going ramp input which is initially below the reference to AMP A1, V1. As the input signal approaches V1, the differential output of A1 approaches zero (i.e., $A = A'$), and the decision point is reached. The output of A1 drives the differential input of LATCH 1. As the input signals continues to go positive, A continues to go positive, and B' begins to go negative. The interpolated decision point is determined when $A = B'$. As the input continues positive, the third decision point is reached when $B = B'$. This novel architecture reduces the ADC input capacitance and thereby minimizes its change with signal level and the associated distortion. The AD9410 also uses an input sample-and-hold circuit for improved ac linearity.

**SUMMARY**

The flash converter still maintains its position as the fastest possible ADC architecture for a given IC process. However, power and real estate considerations generally limit the resolution to 6 or 8 bits. Commercial Gallium Arsenide flash converters are available with sampling rates over 1 GHz, however cost and power dissipation limit their popularity. Higher resolution, lower power, lower cost ADCs can be implemented at lower sampling rates (up to a few hundred MSPS) using the "pipeline" architecture. This technique makes use of low resolution flash converters as building blocks and is discussed in Tutorial MT-023.

**REFERENCES**


5. Paul M. Rainey, "Facsimile Telegraph System," U.S. Patent 1,608,527, filed July 20, 1921, issued November 30, 1926. (although A. H. Reeves is generally credited with the invention of PCM, this patent discloses an electro-mechanical PCM system complete with A/D and D/A converters. The 5-bit electro-mechanical ADC described is probably the first documented flash converter. The patent was largely ignored and forgotten until many years after the various Reeves' patents were issued in 1939-1942).


award for this product in 1988 and was responsible for the initial marketing and applications support for the device).


INTRODUCTION

The successive approximation ADC has been the mainstay of data acquisition systems for many years. Recent design improvements have extended the sampling frequency of these ADCs into the megahertz region with 18-bit resolution. The Analog Devices PulSAR® family of SAR ADCs uses internal switched capacitor techniques along with auto calibration and offers 18-bits at 2 MSPS (AD7641) on CMOS processes without the need for expensive thin-film laser trimming.

The basic successive approximation ADC is shown in Figure 1. It performs conversions on command. In order to process ac signals, SAR ADCs must have an input sample-and-hold (SHA) to keep the signal constant during the conversion cycle.

![Figure 1: Basic Successive Approximation ADC (Feedback Subtraction ADC)](image)

On the assertion of the CONVERT START command, the sample-and-hold (SHA) is placed in the hold mode, and the internal DAC is set to midscale. The comparator determines whether the SHA output is above or below the DAC output, and the result (bit 1, the most significant bit of the conversion) is stored in the successive approximation register (SAR). The DAC is then set either to ¼ scale or ¾ scale (depending on the value of bit 1), and the comparator makes the decision for bit 2 of the conversion. The result is stored in the register, and the process continues until all of the bit values have been determined. When all the bits have been set, tested, and reset or not as appropriate, the contents of the SAR correspond to the value of the analog input, and the conversion is complete. These bit "tests" form the basis of a serial output version SAR-based ADC. Note that the acronym "SAR" actually stands for Successive Approximation Register (the logic block that controls the conversion process), but is universally accepted as the acronym for the architecture itself.

SAR ADC TIMING

The fundamental timing diagram for a typical SAR ADC is shown in Figure 2. The end of conversion is generally indicated by an
end-of-convert (EOC), data-ready (DRDY), or a busy signal (actually, not-BUSY indicates end of conversion). The polarities and name of this signal may be different for different SAR ADCs, but the fundamental concept is the same. At the beginning of the conversion interval, the signal goes high (or low) and remains in that state until the conversion is completed, at which time it goes low (or high). The trailing edge is generally an indication of valid output data, but the data sheet should be carefully studied—in some ADCs additional delay is required before the output data is valid.

![Figure 2: Typical SAR ADC Timing](image)

An N bit conversion takes N steps. It would seem on superficial examination that a 16 bit converter would have twice the conversion time of an 8 bit one, but this is not the case. In an 8 bit converter, the DAC must settle to 8 bit accuracy before the bit decision is made, whereas in a 16 bit converter, it must settle to 16 bit accuracy, which takes a lot longer. In practice, 8 bit successive approximation ADCs can convert in a few hundred nanoseconds, while 16 bit ones will generally take several microseconds.

While there are some variations, the fundamental timing of most SAR ADCs is similar and relatively straightforward. The conversion process is generally initiated by asserting a CONVERT START signal. This signal is a negative-going pulse whose positive-going edge actually initiates the conversion. The internal sample-and-hold (SHA) amplifier is placed in the hold mode on this edge, and the various bits are determined using the SAR algorithm. The negative-going edge of the pulse causes the BUSY line to go high. When the conversion is complete, the BUSY line goes low, indicating the completion of the conversion process. In most cases the trailing edge of the BUSY line can be used as an indication that the output data is valid and can be used to strobe the output data into an external register. However, because of the many variations in terminology and design, the individual data sheet should always be consulted when using a specific ADC. An important characteristic of a SAR ADC is that at the end of the conversion time, the data corresponding to the sampling clock edge is available with no "pipeline" delay. This makes the SAR ADC especially easy to use in "single-shot" and multiplexed applications.

It should also be noted that some SAR ADCs require an external high frequency clock in addition to the CONVERT START command. In most cases there is no need to synchronize the CONVERT START command to the high frequency clock. The frequency of the external clock, if required, generally falls in the range of 1 MHz to 30 MHz depending on the conversion time and resolution of the ADC. Other SAR ADCs have an internal oscillator which is used to perform the conversions and only require the CONVERT START command. Because of their architecture, SAR ADCs generally allow single-shot conversion at any repetition rate from dc to the converter's maximum conversion rate—however, there are some exceptions, so the data sheet should always be consulted.

Notice that the overall accuracy and linearity of the SAR ADC is determined primarily by the internal DAC. Until recently, most precision SAR ADCs used laser-trimmed thin-film DACs to achieve the desired accuracy and linearity. The thin-film resistor trimming process adds cost, and the thin-film resistor values may be affected when subjected to the mechanical stresses of packaging.
For these reasons, switched capacitor (or charge-redistribution) DACs have become popular in newer SAR ADCs. The advantage of the switched capacitor DAC is that the accuracy and linearity is primarily determined by high-accuracy photolithography, which in turn controls the capacitor plate area and the capacitance as well as matching. In addition, small capacitors can be placed in parallel with the main capacitors which can be switched in and out under control of autocalibration routines to achieve high accuracy and linearity without the need for thin-film laser trimming. Temperature tracking between the switched capacitors can be better than 1 ppm/°C, thereby offering a high degree of temperature stability. Modern fine-line CMOS processes are ideal for the switched capacitor SAR ADC, and the cost is therefore low.

A simple 3-bit capacitor DAC is shown in Figure 3. The switches are shown in the track, or sample mode where the analog input voltage, \( A_{IN} \), is constantly charging and discharging the parallel combination of all the capacitors. The hold mode is initiated by opening \( S_{IN} \), leaving the sampled analog input voltage on the capacitor array. Switch \( S_C \) is then opened allowing the voltage at node A to move as the bit switches are manipulated. If S1, S2, S3, and S4 are all connected to ground, a voltage equal to \(-A_{IN}\) appears at node A. Connecting S1 to \( V_{REF} \) adds a voltage equal to \( V_{REF}/2 \) to \(-A_{IN}\). The comparator then makes the MSB bit decision, and the SAR either leaves S1 connected to \( V_{REF} \) or connects it to ground depending on the comparator output (which is high or low depending on whether the voltage at node A is negative or positive, respectively). A similar process is followed for the remaining two bits. At the end of the conversion interval, S1, S2, S3, S4, and \( S_{IN} \) are connected to \( A_{IN} \); \( S_C \) is connected to ground, and the converter is ready for another cycle.

![Figure 3: 3-Bit Switched Capacitor DAC](image)

Note that the extra LSB capacitor (C/4 in the case of the 3-bit DAC) is required to make the total value of the capacitor array equal to 2C so that binary division is accomplished when the individual bit capacitors are manipulated.

The operation of the capacitor DAC (cap DAC) is similar to an R-2R resistive DAC. When a particular bit capacitor is switched to \( V_{REF} \), the voltage divider created by the bit capacitor and the total array capacitance (2C) adds a voltage to node A equal to the weight of that bit. When the bit capacitor is switched to ground, the same voltage is subtracted from node A.

**HISTORICAL PERSPECTIVES ON SAR ADCS**

The basic algorithm used in the successive approximation (initially called feedback subtraction) ADC conversion process can be traced back to the 1500s relating to the solution of a certain mathematical puzzle regarding the determination of an unknown weight by a minimal sequence of weighing operations (Reference 1). In this problem, as stated, the object is to determine the
least number of weights which would serve to weigh an integral number of pounds from 1 lb to 40 lb using a balance scale. One solution put forth by the mathematician Tartaglia in 1556, was to use the series of weights 1 lb, 2 lb, 4 lb, 8 lb, 16 lb, and 32 lb. The proposed weighing algorithm is the same as used in modern successive approximation ADCs. (It should be noted that this solution will actually measure unknown weights up to 63 lb rather than 40 lb as stated in the problem). The algorithm is shown in Figure 4 where the unknown weight is 45 lbs. The balance scale analogy is used to demonstrate the algorithm.

![Figure 4: Successive Approximation ADC Algorithm](image)

Early implementations of the successive approximation ADC did not use either DACs or successive approximation registers but implemented similar functions in a variety of ways. In fact, early SAR ADCs were referred to as *sequential coders*, *feedback coders*, or *feedback subtractor coders*. The term *SAR ADC* came about in the 1970s when commercial successive approximation register logic ICs such as the 2503 and 2504 became available from National Semiconductor and Advanced Micro Devices. These devices were designed specifically to perform the register and control functions in successive approximation ADCs and were standard building blocks in many modular and hybrid data converters.

From a data conversion standpoint, the successive approximation ADC architecture formed the building block for the T1 PCM carrier system and is still a popular architecture today, but the exact origin of this architecture is not clear. Although countless patents have been granted relating to refinements and variations on the successive approximation architecture, they do not claim the fundamental principle.

The first mention of the successive approximation ADC architecture (actually a *sequential coder*) in the context of PCM was by J. C. Schelleng of Bell Telephone Laboratories in a patent filed in 1946 (Reference 2). The design does not use an internal DAC, but implements the approximation process in a somewhat novel manner involving the addition of binary weighted reference voltages. Details of this vacuum tube design are discussed in the patent.

A much more elegant implementation of the successive approximation ADC is described by Goodall of Bell Telephone Labs in a 1947 article (Reference 3). This ADC has 5-bit resolution and samples the voice channel at a rate of 8 kSPS. The voice signal is first sampled, and the corresponding voltage stored on a capacitor. It is then compared to a reference voltage which is equal to ½ the full-scale voltage. If it is greater than the reference voltage, the MSB is registered as a "1," and an amount of charge equal to ½ scale is subtracted from the storage capacitor. If the voltage on the capacitor is less than ½ scale, then no charge is removed, and the bit is registered as a "0". After the MSB decision is completed, the cycle continues for the second bit, but with
the reference voltage now equal to ¼ scale. The process continues until all bit decisions are completed. This concept of charge redistribution is similar to modern switched-capacitor DACs.

Both the Schelleng and the Goodall ADCs use a process of addition/subtraction of binary weighted reference voltages to perform the SAR algorithm. Although the DAC function is there, it is not performed using a traditional binary weighted DAC. The ADCs described by H. R. Kaiser et. al. (Reference 4) and B. D. Smith (Reference 5) in 1953 use an actual binary weighted DAC to generate the analog approximation to the input signal, similar to modern SAR ADCs. Smith also points out that non-linear ADC transfer functions can be achieved by using a non-uniformly weighted DAC. This technique formed the basis of companding voiceband codecs used in early PCM systems. (See Tutorial MT-018, "Intentionally Nonlinear DACs.") Before this non-linear ADC technique was developed, linear ADCs were used, and the compression and expansion functions were performed by diode/resistor networks which had to be individually calibrated and held at a constant temperature to prevent drift errors (Reference 6).

Of course, no discussion on ADC history would be complete without crediting the truly groundbreaking work of Bernard M. Gordon at EPSCO (now Analogic, Incorporated). Gordon's 1955 patent application (Reference 7) describes an all-vacuum tube 11-bit, 50-kSPS successive approximation ADC—representing the first commercial offering of a complete converter (see Figure 5). The DATRAC was offered in a 19" × 26" × 15" housing, dissipated several hundred watts, and sold for approximately $8000.00.

In a later patent (Reference 8), Gordon describes the details of the logic block required to perform the successive approximation algorithm. The SAR logic function was later implemented in the 1970s by National Semiconductor and Advanced Micro Devices—the popular 2502/2503/2504 family of IC logic chips. These chips were to become an integral building block of practically all modular and hybrid successive approximation ADCs of the 1970s and 1980s.

![Figure 5: 1954 "DATRAC" 11-Bit, 50-kSPS SAR ADC Designed by Bernard M. Gordon at EPSCO](image)

ANALOG DEVICES ENTERS THE DATA CONVERTER ARENA IN 1969

In 1965, Ray Stata and Matt Lorber founded Analog Devices, Inc. (ADI) in Cambridge, MA. The initial product offerings were high performance modular op amps, but in 1969 ADI acquired Pastoriza Electronics, a leader in data converter products, thereby making a solid commitment to both data acquisition and linear products.

Pastoriza had a line of data acquisition products, and Figure 6 shows a photograph of a 1969 12-bit, 10-µs general purpose successive approximation ADC, the ADC-12U, that sold for approximately $800.00. The architecture was successive
approximation, and the ADC-12U utilized a µA710 comparator, a modular 12-bit "Minidac," and 14 7400-series logic packages to perform the successive approximation conversion algorithm.

The "Minidac" module was actually constructed from "quad switch" ICs (AD550) and a thin film network (AD850). These early DAC building blocks are discussed further in Tutorial MT-015, "DAC Architectures II: Binary DACs."

Notice that in the ADC-12U, the implementation of the successive approximation algorithm required 14 logic packages. In 1958, Bernard M. Gordon had filed a patent on the logic to perform the successive approximation algorithm (Reference 19), and in the early 1970s, Advanced Micro Devices and National Semiconductor introduced commercial successive approximation register logic ICs: the 2502 (8-bit, serial, not expandable), 2503 (8-bit, expandable) and 2504 (12-bit, serial, expandable). These were designed specifically to perform the register and control functions in successive approximation ADCs. These became standard building blocks in many modular and hybrid data converters.

Analog Devices continued to pioneer in data conversion after 1969. Modules gradually evolved into hybrid circuits during the 1970s. Hybrids generally utilize ceramic substrates with either thick or thin film conductors. Individual die are bonded to the substrate (usually with epoxy), and wire bonds make the connections between the bond pads and the conductors. The hybrid is usually hermetically sealed in some sort of ceramic or metal package. Accuracy was achieved by trimming thick or thin film resistors after assembly and interconnection, but before sealing. Manufacturers used thin film networks, discrete thin film resistors, deposited thick or thin film resistors, or some combination of the above.

An excellent example of hybrid technology was the AD572 12-bit, 25-µs SAR ADC introduced by Analog Devices in 1977. The AD572 was complete with internal clock, voltage reference, comparator, and input buffer amplifier. The SAR register was the popular 2504. The internal DAC was comprised of a 12-bit switch chip and an actively trimmed thin film ladder network (separately packaged as the two-chip AD562 DAC). The AD572 was the first military-approved 12-bit ADC processed to MIL-STD-883B, and specified over the full operating temperature range of −55°C to +125°C. A photograph of the AD572 is shown in Figure 7.
Analog Devices also pioneered in monolithic data converters. Probably the most significant SAR ADC ever introduced was the 12-bit, 35-µs AD574 in 1978. The AD574 represents a complete solution, including buried Zener reference, timing circuits, and three-state output buffers for direct interfacing to an 8-, 12-, or 16-bit microprocessor bus. In its introductory form, the AD574 was manufactured using compound monolithic construction, based on two chips—one an AD565 12-bit current-output DAC, including reference and thin film scaling resistors; and the other containing the successive approximation register (SAR) and microprocessor interface logic functions as well as a precision latching comparator. The AD574 soon emerged as the industry-standard 12-bit ADC in the early 1980s. In 1985, the device became available in single-chip monolithic form for the first time; thereby making low-cost commercial plastic packaging possible. A simplified block diagram of the AD574 is shown in Figure 8.
MODERN SAR ADCs

Because of their popularity, successive approximation ADCs are available in a wide variety of resolutions, sampling rates, input and output options, package styles, and costs. Many SAR ADCs now offer on-chip input multiplexers, making them the ideal choice for multichannel data acquisition systems. It would be impossible to attempt to discuss all types of SAR ADCs in this tutorial, so we will only give a few highlights of modern breakthrough products.

An example of modern charge redistribution successive approximation ADCs is Analog Devices' PulSAR® series. The AD7641 is a 18-bit, 2-MSPS, fully differential, ADC that operates from a single 2.5 V power supply (see Figure 9). The part contains a high-speed 18-bit sampling ADC, an internal conversion clock, error correction circuits, internal reference, and both serial and parallel system interface ports. The AD7641 is hardware factory calibrated and comprehensively tested to ensure such ac parameters as signal-to-noise ratio (SNR) and total harmonic distortion (THD), in addition to the more traditional dc parameters of gain, offset, and linearity.
PROCESSING INDUSTRIAL-LEVEL SIGNALS

Many low voltage single-supply SAR ADCs have been introduced over the last few years, however their input range is usually limited to less than or equal to the supply voltage. In many situations this is not a problem; but there still exist many industrial applications which require digitization of bipolar signals (for example, ±5 V or ±10 V). This requires external circuitry when interfacing to single-supply ADCs. Figure 10 shows two possible approaches. An external op amp can be used to perform the level shifting and attenuation required to match the ±10 V signal to the 0 to +2.5 V input range of the ADC (Figure 10A). An alternative is to utilize a resistor network to perform the attenuation and level shifting (Figure 10B). Both methods require external components.

A much better solution available from Analog Devices uses a proprietary industrial CMOS (\textit{CMOS™}) process which allows the input circuitry to operate on standard industrial ±15 V supplies, while operating the ADC core on the low voltage supply (5 V or less). Figure 11 shows the AD7328 13-bit 8-channel input ADC.
The AD7328 is designed on the \textit{iCMOS} (industrial CMOS) process. \textit{iCMOS} is a process combining high voltage CMOS and low voltage CMOS. It enables the development of a wide range of high performance analog ICs capable of 33 V operation in a footprint that no previous generation of high voltage parts could achieve. Unlike analog ICs using conventional CMOS processes, \textit{iCMOS} components can accept bipolar input signals while providing increased performance, dramatically reducing power consumption, and having a reduced package size. The AD7328 can accept true bipolar analog input signals. The AD7328 has four software-selectable input ranges, $\pm$10 V, $\pm$5 V, $\pm$2.5 V, and 0 V to 10 V. Each analog input channel can be independently programmed to one of the four input ranges. The analog input channels on the AD7328 can be programmed to be single-ended, true differential, or pseudo differential. The ADC contains a 2.5 V internal reference. The AD7328 also allows for external reference operation. If a 3 V external reference is applied to the REFIN/OUT pin, the AD7328 can accept a true bipolar $\pm$12 V analog input. Minimum $\pm$12 V V\textsubscript{DD} and V\textsubscript{SS} supplies are required for the $\pm$12 V input range.

The low voltage core of the AD7328 operates on the V\textsubscript{CC} supply which should be 5 V nominal (4.75 V to 5.5 V) for specified performance. For V\textsubscript{CC} between 2.7 V and 4.75 V, the AD7328 will meet its typical specifications. The AD7328 has a separate V\textsubscript{DRIVE} pin which sets the I/O logic interface voltage (2.7 V to 5.5 V). The V\textsubscript{DRIVE} voltage should not exceed V\textsubscript{CC} by more than 0.3 V.

The AD7328 has a high speed serial interface that can operate at throughput rates up to 1 MSPS.

**SUMMARY**

The SAR ADC architecture is elegant, efficient, easy to understand, and ideally suited to modern fine-line CMOS processes. The lack of "pipeline" delay (or latency) makes it ideal for single-shot and multiplexed data acquisition applications. CMOS processes allows the addition of a variety of digital functions, such as automatic channel sequencing, auto-calibration, etc. In addition, many SAR ADCs have on-chip temperature sensors and voltage references. Although the SAR ADC had its origins in mathematical puzzles of the 1500s, it is still the converter of choice for modern multichannel data acquisition systems.

**REFERENCES**

2. John C. Schelleng, "Code Modulation Communication System," U.S. Patent 2,453,461, filed June 19, 1946, issued November 9, 1948. (an interesting description of a rather cumbersome successive approximation ADC based on vacuum tube technology. This converter was not very practical, but did illustrate the concept. Also in the patent is a description of a corresponding binary DAC).


5. B. D. Smith, "Coding by Feedback Methods," Proceedings of the I. R. E., Vol. 41, August 1953, pp. 1053-1058. (Smith uses an internal DAC and also points out that a non-linear transfer function can be achieved by using a DAC with non-uniform bit weights, a technique which is widely used in today's voiceband ADCs with built-in companding).


INTRODUCTION

The sigma-delta (Σ-Δ) ADC is the converter of choice for modern voiceband, audio, and high-resolution precision industrial measurement applications. The highly digital architecture is ideally suited for modern fine-line CMOS processes, thereby allowing easy addition of digital functionality without significantly increasing the cost. Because of its widespread use, it is important to understand the fundamental principles behind this converter architecture.

Due to the length of the topic, the discussion of Σ-Δ ADCs requires two tutorials, MT-022 and MT-023. This first tutorial (MT-022) first discusses the history of Σ-Δ and the fundamental concepts of oversampling, quantization noise shaping, digital filtering, and decimation. Tutorial MT-023 discusses more advanced topics related to Σ-Δ, including idle tones, multi-bit Σ-Δ ADCs, multistage noise shaping Σ-Δ ADCs (MASH), bandpass Σ-Δ ADCs, as well as some example applications.

HISTORICAL PERSPECTIVE

The Σ-Δ ADC architecture had its origins in the early development phases of pulse code modulation (PCM) systems—specifically, those related to transmission techniques called delta modulation and differential PCM. (An excellent discussion of both the history and concepts of the Σ-Δ ADC can be found by Max Hauser in Reference 1). Delta modulation was first invented at the ITT Laboratories in France by E. M. Deloraine, S. Van Mierlo, and B. Derjavitch in 1946 (References 2, 3). The principle was "rediscovered" several years later at the Phillips Laboratories in Holland, whose engineers published the first extensive studies both of the single-bit and multi-bit concepts in 1952 and 1953 (References 4, 5). In 1950, C. C. Cutler of Bell Telephone Labs in the U.S. filed an important patent on differential PCM which covered the same essential concepts (Reference 6).

The driving force behind delta modulation and differential PCM was to achieve higher transmission efficiency by transmitting the changes (delta) in value between consecutive samples rather than the actual samples themselves.

In delta modulation, the analog signal is quantized by a one-bit ADC (a comparator) as shown in Figure 1A. The comparator output is converted back to an analog signal with a 1-bit DAC, and subtracted from the input after passing through an integrator. The shape of the analog signal is transmitted as follows: a "1" indicates that a positive excursion has occurred since the last sample, and a "0" indicates that a negative excursion has occurred since the last sample.
If the analog signal remains at a fixed dc level for a period of time, an alternating pattern of "0s" and "1s" is obtained. It should be noted that differential PCM (see Figure 1B) uses exactly the same concept except a multibit ADC is used rather than a single comparator to derive the transmitted information.

Since there is no limit to the number of pulses of the same sign that may occur, delta modulation systems are capable of tracking signals of any amplitude. In theory, there is no peak clipping. However, the theoretical limitation of delta modulation is that the analog signal must not change too rapidly. The problem of slope clipping is shown in Figure 2. Here, although each sampling instant indicates a positive excursion, the analog signal is rising too quickly, and the quantizer is unable to keep pace.
Slope clipping can be reduced by increasing the quantum step size or increasing the sampling rate. Differential PCM uses a multibit quantizer to effectively increase the quantum step sizes at the increase of complexity. Tests have shown that in order to obtain the same quality as classical PCM, delta modulation requires very high sampling rates, typically $20 \times$ the highest frequency of interest, as opposed to Nyquist rate of $2\times$. 

For these reasons, delta modulation and differential PCM have never achieved any significant degree of popularity, however a slight modification of the delta modulator leads to the basic $\Sigma\Delta$ architecture, one of the most popular ADC architectures in use today.

In 1954 C. C. Cutler of Bell Labs filed a very significant patent which introduced the principle of oversampling and noise shaping with the specific intent of achieving higher resolution (Reference 7). His objective was not specifically to design a Nyquist ADC, but to transmit the oversampled noise-shaped signal without reducing the data rate. Thus Cutler’s converter embodied all the concepts in a $\Sigma\Delta$ ADC with the exception of digital filtering and decimation which would have been too complex and costly at the time using vacuum tube technology.

Occasional work continued on these concepts over the next several years, including an important patent of C. B. Brahm filed in 1961 which gave details of the analog design of the loop filter for a second-order multibit noise shaping ADC (Reference 8). Transistor circuits began to replace vacuum tubes over the period, and this opened up many more possibilities for implementation of the architecture.

In 1962, Inose, Yasuda, and Murakami elaborated on the single-bit oversampling noise-shaping architecture proposed by Cutler in 1954 (Reference 9). Their experimental circuits used solid state devices to implement first and second-order $\Sigma\Delta$ modulators. The 1962 paper was followed by a second paper in 1963 which gave excellent theoretical discussions on oversampling and noise-shaping (Reference 10). These two papers were also the first to use the name delta-sigma to describe the architecture. The name delta-sigma stuck until the 1970s when AT&T engineers began using name sigma-delta. Since that time, both names have been used; however, sigma-delta may be the more correct of the two.

It is interesting to note that all the work described thus far was related to transmitting an oversampled digitized signal directly rather than the implementation of a Nyquist ADC. In 1969 D. J. Goodman at Bell Labs published a paper describing a true
Nyquist Σ-Δ ADC with a digital filter and a decimator following the modulator (Reference 11). This was the first use of the Σ-Δ architecture for the explicit purpose of producing a Nyquist ADC. In 1974 J. C. Candy, also of Bell Labs, described a multibit oversampling Σ-Δ ADC with noise shaping, digital filtering, and decimation to achieve a high resolution Nyquist ADC (Reference 12).

The IC Σ-Δ ADC offers several advantages over the other architectures, especially for high resolution, low frequency applications. First and foremost, the single-bit Σ-Δ ADC is inherently monotonic and requires no laser trimming. The Σ-Δ ADC also lends itself to low cost foundry CMOS processes because of the digitally intensive nature of the architecture. Examples of early monolithic Σ-Δ ADCs are given in References 13-21. Since that time there have been a constant stream of process and design improvements in the fundamental architecture proposed in the early works cited above.

Modern CMOS Σ-Δ ADCs (and DACs, for that matter) are the converters of choice for voiceband and audio applications. The highly digital architectures lend themselves nicely to fine-line CMOS. In addition, high resolution (up to 24 bits) low frequency Σ-Δ ADCs have virtually replaced the older integrating converters in precision industrial measurement applications.

**BASICS OF Σ-Δ ADCs**

There have been innumerable descriptions of the architecture and theory of Σ-Δ ADCs, but most commence with a maze of integrals and deteriorate from there. Some engineers who do not understand the theory of operation of Σ-Δ ADCs are convinced, from study of a typical published article, that it is too complex to comprehend easily.

There is nothing particularly difficult to understand about Σ-Δ ADCs, as long as you avoid the detailed mathematics, and this section has been written in an attempt to clarify the subject. A Σ-Δ ADC contains very simple analog electronics (a comparator, voltage reference, a switch, and one or more integrators and analog summing circuits), and quite complex digital computational circuitry. This digital circuitry consists of a digital signal processor (DSP) which acts as a filter (generally, but not invariably, a low pass filter). It is not necessary to know precisely how the filter works to appreciate what it does. To understand how a Σ-Δ ADC works, familiarity with the concepts of oversampling, quantization noise shaping, digital filtering, and decimation is required.

Let us consider the technique of oversampling with an analysis in the frequency domain. Where a dc conversion has a quantization error of up to ½ LSB, a sampled data system has quantization noise. A perfect classical N bit sampling ADC has an rms quantization noise of \( q/\sqrt{12} \) uniformly distributed within the Nyquist band of dc to \( f_s/2 \) (where \( q \) is the value of an LSB and \( f_s \) is the sampling rate) as shown in Figure 3A. Therefore, its SNR with a full scale sinewave input will be \((6.02N + 1.76)\ dB\) (Refer to Tutorial MT-001 for the derivation). If the ADC is less than perfect, and its noise is greater than its theoretical minimum quantization noise, then its effective resolution will be less than N bits. Its actual resolution (often known as its Effective Number of Bits or ENOB) will be defined by

\[
ENOB = \frac{SNR - 1.76\ dB}{6.02\ dB}.
\]  
Eq. 1

If we choose a much higher sampling rate, \( Kf_s \) (see Figure 3B), the rms quantization noise remains \( q/\sqrt{12} \), but the noise is now distributed over a wider bandwidth dc to \( Kf_s/2 \). If we then apply a digital low pass filter (LPF) to the output, we remove much of the quantization noise, but do not affect the wanted signal—so the ENOB is improved. We have accomplished a high resolution A/D conversion with a low resolution ADC. The factor \( K \) is generally referred to as the oversampling ratio. It should be noted at this point that oversampling has an added benefit in that it relaxes the requirements on the analog anti-aliasing filter. This is a big advantage of Σ-Δ, especially in consumer audio applications where the cost of a sharp cutoff linear phase filter can be significant.
Figure 3: Oversampling, Digital Filtering, Noise Shaping, and Decimation

Since the bandwidth is reduced by the digital output filter, the output data rate may be lower than the original sampling rate ($Kf_s$) and still satisfy the Nyquist criterion. This may be achieved by passing every $M$th result to the output and discarding the remainder. The process is known as "decimation" by a factor of $M$. Despite the origins of the term (decem is Latin for ten), $M$ can have any integer value, provided that the output data rate is more than twice the signal bandwidth. Decimation does not cause any loss of information (see Figure 3B).

If we simply use oversampling to improve resolution, we must oversample by a factor of $2^{2N}$ to obtain an $N$ bit increase in resolution. The $\Sigma\Delta$ converter does not need such a high oversampling ratio because it not only limits the signal passband, but also shapes the quantization noise so that most of it falls outside this passband as shown in Figure 3C.

If we take a 1 bit ADC (a comparator), drive it with the output of an integrator, and feed the integrator with an input signal summed with the output of a 1 bit DAC fed from the ADC output, we have a first-order $\Sigma\Delta$ modulator as shown in Figure 4. Add a digital low pass filter (LPF) and decimator at the digital output, and we have a $\Sigma\Delta$ ADC—the $\Sigma\Delta$ modulator shapes the quantization noise so that it lies above the passband of the digital output filter, and the ENOB is therefore much larger than would otherwise be expected from the oversampling ratio.
Intuitively, a $\Sigma$-$\Delta$ ADC operates as follows. Assume a dc input at $V_{IN}$. The integrator is constantly ramping up or down at node A. The output of the comparator is fed back through a 1-bit DAC to the summing input at node B. The negative feedback loop from the comparator output through the 1-bit DAC back to the summing point will force the average dc voltage at node B to be equal to $V_{IN}$. This implies that the average DAC output voltage must equal the input voltage $V_{IN}$. The average DAC output voltage is controlled by the ones-density in the 1-bit data stream from the comparator output. As the input signal increases towards $+V_{REF}$, the number of "ones" in the serial bit stream increases, and the number of "zeros" decreases. Similarly, as the signal goes negative towards $-V_{REF}$, the number of "ones" in the serial bit stream decreases, and the number of "zeros" increases. From a very simplistic standpoint, this analysis shows that the average value of the input voltage is contained in the serial bit stream out of the comparator. The digital filter and decimator process the serial bit stream and produce the final output data.

For any given input value in a single sampling interval, the data from the 1-bit ADC is virtually meaningless. Only when a large number of samples are averaged, will a meaningful value result. The $\Sigma$-$\Delta$ modulator is very difficult to analyze in the time domain because of this apparent randomness of the single-bit data output. If the input signal is near positive full-scale, it is clear that there will be more "1"s than "0"s in the bit stream. Likewise, for signals near negative full-scale, there will be more "0"s than "1"s in the bit stream. For signals near midscale, there will be approximately an equal number of "1"s and "0"s. Figure 5 shows the output of the integrator for two input conditions. The first is for an input of zero (midscale). To decode the output, pass the output samples through a simple digital lowpass filter that averages every four samples. The output of the filter is $2/4$. This value represents bipolar zero. If more samples are averaged, more dynamic range is achieved. For example, averaging 4 samples gives 2 bits of resolution, while averaging 8 samples yields $4/8$, or 3 bits of resolution. In the bottom waveform of Figure 5, the average obtained for 4 samples is $3/4$, and the average for 8 samples is $6/8$. 

Figure 4: First-Order Sigma-Delta ADC
For an interactive tutorial on the time domain characteristics of the Σ-Δ modulator, refer to the Sigma-Delta Tutorial located in the Analog Devices’ Design Center which gives a graphical illustration of the behavior of an idealized Σ-Δ ADC.

The Σ-Δ ADC can also be viewed as a synchronous voltage-to-frequency converter followed by a counter. If the number of “1”s in the output data stream is counted over a sufficient number of samples, the counter output will represent the digital value of the input. Obviously, this method of averaging will only work for dc or very slowly changing input signals. In addition, $2^N$ clock cycles must be counted in order to achieve N-bit effective resolution, thereby severely limiting the effective sampling rate.

It should be noted that because the digital filter is an integral part of the Σ-Δ ADC, there is a built-in "pipeline" delay (sometimes called "latency") primarily determined by the number of taps in the digital filter. Digital filters in Σ-Δ ADCs can be quite large (several hundred taps), so the latency may become an issue in multiplexed applications where the appropriate amount of settling time must be allowed after switching channels.

**FREQUENCY DOMAIN ANALYSIS OF A SIGMA-DELTA ADC AND NOISE SHAPING**

Further time-domain analysis is not productive, and the concept of noise shaping is best explained in the frequency domain by considering the simple Σ-Δ modulator model in Figure 6.
The integrator in the modulator is represented as an analog lowpass filter with a transfer function equal to \( H(f) = \frac{1}{f} \). This transfer function has an amplitude response which is inversely proportional to the input frequency. The 1-bit quantizer generates quantization noise, \( Q \), which is injected into the output summing block. If we let the input signal be \( X \), and the output \( Y \), the signal coming out of the input summer must be \( X - Y \). This is multiplied by the filter transfer function, \( \frac{1}{f} \), and the result goes to one input of the output summer. By inspection, we can then write the expression for the output voltage \( Y \) as:

\[
Y = \frac{1}{f} (X - Y) + Q
\]

REARRANGING, SOLVING FOR \( Y \):

\[
Y = \frac{X}{f + 1} + \frac{Q \cdot f}{f + 1}
\]

SIGNAL TERM NOISE TERM

This expression can easily be rearranged and solved for \( Y \) in terms of \( X \), \( f \), and \( Q \):

\[
Y = \frac{X}{f + 1} + \frac{Q \cdot f}{f + 1}
\]

Note that as the frequency \( f \) approaches zero, the output voltage \( Y \) approaches \( X \) with no noise component. At higher frequencies, the amplitude of the signal component approaches zero, and the noise component approaches \( Q \). At high frequency, the output consists primarily of quantization noise. In essence, the analog filter has a lowpass effect on the signal, and a highpass effect on the quantization noise. Thus the analog filter performs the noise shaping function in the \( \Sigma-\Delta \) modulator model. For a given input frequency, higher order analog filters offer more attenuation. The same is true of \( \Sigma-\Delta \) modulators, provided certain precautions are taken.

By using more than one integration and summing stage in the \( \Sigma-\Delta \) modulator, we can achieve higher orders of quantization noise shaping and even better ENOB for a given oversampling ratio as is shown in Figure 7 for both a first and second-order \( \Sigma-\Delta \) modulator.
The block diagram for the second-order Σ-Δ modulator is shown in Figure 8. Third, and higher, order Σ-Δ ADCs were once thought to be potentially unstable at some values of input—recent analyses using finite rather than infinite gains in the comparator have shown that this is not necessarily so, but even if instability does start to occur, the DSP in the digital filter and decimator can be made to recognize incipient instability and react to prevent it.

Figure 9 shows the relationship between the order of the Σ-Δ modulator and the amount of oversampling necessary to achieve a particular SNR. For instance, if the oversampling ratio is 64, an ideal second-order system is capable of providing an SNR of
about 80 dB. This implies approximately 13 effective number of bits (ENOB). Although the filtering done by the digital filter and decimator can be done to any degree of precision desirable, it would be pointless to carry more than 13 binary bits to the outside world. Additional bits would carry no useful signal information, and would be buried in the quantization noise unless post-filtering techniques were employed. Additional resolution can be obtained from the 1-bit system by increasing the oversampling ratio and/or by using a higher-order modulator. Other methods are often used to achieve higher resolution, such as the multi-bit Σ-Δ architecture, and are discussed in Tutorial MT-023.

![Graph showing SNR versus oversampling ratio for first, second, and third-order loops](image)

**Figure 9: SNR Versus Oversampling Ratio for First, Second, and Third-Order Loops**

**SUMMARY**

This tutorial has covered the basics of Σ-Δ ADCs from a historical perspective including the important concepts of oversampling, digital filtering, noise shaping, and decimation. Tutorial MT-023 covers some of the more advanced concepts and applications of Σ-Δ ADCs, such as idle tones, multi-bit Σ-Δ, MASH, and bandpass Σ-Δ.

**REFERENCES**


7. C. C. Cutler, "Transmission Systems Employing Quantization," U.S. Patent 2,927,962, filed April 26, 1954, issued March 8, 1960. (a ground-breaking patent describing oversampling and noise shaping using first and second-order loops to increase effective resolution. The goal was transmission of oversampled noise shaped PCM data without decimation, not a Nyquist-type ADC).


INTRODUCTION

Tutorial MT-022 discussed the basics of Σ-Δ ADCs. In this tutorial, we will look at some of the more advanced concepts including idle tones, multi-bit Σ-Δ, MASH, bandpass Σ-Δ, as well as some example applications.

IDLE TONE CONSIDERATIONS

In our discussion of Σ-Δ ADCs up to this point, we have made the assumption that the quantization noise produced by the Σ-Δ modulator (see Figure 1) is random and uncorrelated with the input signal. Unfortunately, this is not entirely the case, especially for the first-order modulator. Consider the case where we are averaging 16 samples of the modulator output in a 4-bit Σ-Δ ADC.

Figure 1: First-Order Sigma-Delta ADC

Figure 2 shows the bit pattern for two input signal conditions: an input signal having the value 8/16, and an input signal having the value 9/16. In the case of the 9/16 signal, the modulator output bit pattern has an extra "1" every 16th output. This will produce energy at Kfs/16, which translates into an unwanted tone. If the oversampling ratio (K) is less than 8, this tone will fall into the passband. In audio, the idle tones can be heard just above the noise floor as the input changes from negative to positive fullscale.
Figure 2: Repetitive Bit Pattern in Sigma-Delta Modulator Output

<table>
<thead>
<tr>
<th>16 SAMPLES OF SIGMA-DELTA MODULATOR DATA OUTPUT STREAM</th>
<th>BINARY EQUIVALENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0... = 8/16</td>
<td>1000</td>
</tr>
<tr>
<td>1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1... = 9/16</td>
<td>1001</td>
</tr>
</tbody>
</table>

REPEATS EVERY 16 SAMPLES

Figure 3 shows the correlated idling pattern behavior for a first order Σ-Δ modulator, and Figure 4 shows the relatively uncorrelated pattern for a second-order modulator. For this reason, virtually all Σ-Δ ADCs contain at least a second-order modulator loop, and some use up to fifth-order loops.

Figure 3: Idling Patterns for First-Order Sigma-Delta Modulator (Integrator Output)
HIgher Order Loop Considerations

In order to achieve wide dynamic range, Σ-Δ modulator loops greater than second-order are necessary, but present real design challenges. First of all, the simple linear models previously discussed are no longer fully accurate. Loops of order greater than two are generally not guaranteed to be stable under all input conditions. The instability arises because the comparator is a nonlinear element whose effective "gain" varies inversely with the input level. This mechanism for instability causes the following behavior: if the loop is operating normally, and a large signal is applied to the input that overloads the loop, the average gain of the comparator is reduced. The reduction in comparator gain in the linear model causes loop instability. This causes instability even when the signal that caused it is removed. In actual practice, such a circuit would normally oscillate on power-up due to initial conditions caused by turn-on transients. The AD1879 dual audio ADC released in 1994 by Analog Devices used a 5th order loop. Extensive nonlinear stabilization techniques were required in this and similar higher-order loop designs (References 1-5).

Multi-Bit Sigma-Delta Converters

So far we have considered only Σ-Δ converters which contain a single-bit ADC (comparator) and a single-bit DAC (switch). The block diagram of Figure 5 shows a multi-bit Σ-Δ ADC which uses an n-bit flash ADC and an n-bit DAC. Obviously, this architecture will give a higher dynamic range for a given oversampling ratio and order of loop filter. Stabilization is easier, since second-order loops can generally be used. Idling patterns tend to be more random thereby minimizing tonal effects.
The real disadvantage of this technique is that the linearity depends on the DAC linearity, and thin film laser trimming is required to approach 16-bit performance levels. This makes the multi-bit architecture extremely impractical to implement on mixed-signal ICs using traditional binary DAC techniques.

However, fully decoded thermometer DACs (see Tutorial MT-014) coupled with proprietary data scrambling techniques as used in a number of Analog Devices' audio ADCs and DACs, including the 24-bit stereo AD1871 (see References 6 and 7) can achieve high SNR and low distortion using the multi-bit architecture. The multi-bit data scrambling technique both minimizes idle tones and ensures better differential linearity. A simplified block diagram of the AD1871 ADC is shown in Figure 6.

![Figure 6: AD1871 24-Bit 96-kSPS Stereo Audio Multi-Bit Sigma-Delta ADC](image)

The AD1871's analog Σ-Δ modulator section comprises a second order multi-bit implementation using Analog Device's proprietary technology for best performance. As shown in Figure 7, the two analog integrator blocks are followed by a flash ADC section that generates the multi-bit samples.

The output of the flash ADC, which is thermometer encoded, is decoded to binary for output to the filter sections and is scrambled for feedback to the two integrator stages. The modulator is optimized for operation at a sampling rate of 6.144 MHz (which is 128×f_s at 48-kHz sampling and 64×f_s at 96-kHz sampling). The A-weighted dynamic range of the AD1871 is typically 105 dB.

![Figure 7: Details of the AD1871 Second-Order Modulator and Data Scrambler](image)

**DIGITAL FILTER IMPLICATIONS ON MULTIPLEXED APPLICATIONS**

The digital filter is an integral part of all Σ-Δ ADCs—there is no way to remove it. The settling time of this filter affects certain
applications especially when using Σ-Δ ADCs in multiplexed applications. The output of a multiplexer can present a step function input to an ADC if there are different input voltages on adjacent channels. In fact, the multiplexer output can represent a full-scale step voltage to the Σ-Δ ADC when channels are switched. Adequate filter settling time must be allowed, therefore, in such applications. This does not mean that Σ-Δ ADCs shouldn’t be used in multiplexed applications, just that the settling time of the digital filter must be considered. Some newer Σ-Δ ADCs such are actually optimized for use in multiplexed applications.

For example, the group delay through the AD1871 digital filter is 910 µs (sampling at 48 kSPS) and 460 µs (sampling at 96 kSPS)—this represents the time it takes for a step function input to propagate through one-half the number of taps in the digital filter. The total settling time is therefore approximately twice the group delay time. The input oversampling frequency is 6.144 MSPS for both conditions. The frequency response of the digital filter in the AD1871 ADC is shown in Figure 8. This filter uses a finite impulse response (FIR) design, and therefore has linear phase over the audio passband. Duplicating this performance using an analog filter would require considerable design effort as well as rather costly components.

![Figure 8: AD1871 24-Bit, 96-kSPS Stereo Sigma-Delta ADC Digital Filter Characteristics](image)

In other applications, such as low frequency, high resolution 24-bit measurement Σ-Δ ADCs (such as the AD77xx-series), other types of digital filters may be used. For instance, the SINC$^3$ response is popular because it has zeros at multiples of the throughput rate. For instance a 10-Hz throughput rate produces zeros at 50 Hz and 60 Hz which aids in ac power line rejection.

Regardless of the type of digital filter, Σ-Δ ADCs require that sufficient settling time is allowed after the application of a step function input.

**MULTISTAGE NOISE SHAPING (MASH) SIGMA-DELTA CONVERTERS**

As has been discussed, nonlinear stabilization techniques can be difficult for 3rd order loops or higher. In many cases, the multi-bit architecture is preferable. An alternative approach to either of these, called multistage noise shaping (MASH), utilizes cascaded stable first-order loops (see References 8 and 9). Figure 9 shows a block diagram of a three-stage MASH ADC. The output of the first integrator is subtracted from the first DAC output to yield the first stage quantization noise, Q1. Q1 is then quantized by the second stage. The output of the second integrator is subtracted from the second DAC output to yield the second stage quantization noise which is in turn quantized by the third stage.
The output of the first stage is summed with a single digital differentiation of the second stage output and a double differentiation of the third stage output to yield the final output. The result is that the quantization noise $Q_1$ is suppressed by the second stage, and the quantization noise $Q_2$ is suppressed by the third stage yielding the same suppression as a third-order loop. Since this result is obtained using three first-order loops, stable operation is assured.

![Multi-Stage Noise Shaping Sigma-Delta ADC (MASH)](image)

**Figure 9: Multi-Stage Noise Shaping Sigma-Delta ADC (MASH)**

**HIGH RESOLUTION MEASUREMENT SIGMA-DELTA ADCS**

While older integrating architectures such as dual-slope are still used in digital voltmeters, CMOS $\Sigma$-$\Delta$ ADCs are the dominant converter for today's industrial measurement applications. These converters offer excellent 50-Hz/60-Hz power line common-mode rejection and resolutions up to 24 bits with various digital features, such as on-chip calibration. Many have programmable gain amplifiers (PGAs) which allow the direct digitization of small signals from bridge and thermocouple transducers without the need for additional external signal conditioning circuits.

In order to better understand the capability of $\Sigma$-$\Delta$ measurement ADCs and the power of the technique, a modern example, the 24-bit AD7799, will be examined in detail. The AD7799 is a member of the AD77xx family and is shown in Figure 10. This ADC was specifically designed to interface directly to low-level sensor outputs such as bridges in weigh scale applications. The device accepts low-level signals directly from a bridge and outputs a serial digital word. There are three multiplexed and buffered differential inputs which drive an internal instrumentation amplifier. The in-amp can be programmed for eight different gains: 1, 2, 4, 8, 16, 32, 64, and 128.
Figure 10: **AD7799** Sigma-Delta Single-Supply Bridge ADC

Figure 11 shows a direct connection between a bridge-based load cell and a high resolution ΣΔ ADC, the AD7799. The fullscale bridge output of 10 mV is digitized to approximately 16 noise-free bits by the ADC at a throughput rate of 4.17 Hz. Ratiometric operation eliminates the need for a precision voltage reference. The AD7799 can be operated at throughput rates from 4.17 Hz to 500 Hz. The part operates with a power supply from 2.7 V to 5.25 V and consumes 380 µA typical.

Figure 11: Load Cell Conditioning Using a High Resolution Sigma-Delta ADC

**BANDPASS SIGMA-DELTA CONVERTERS**

The Σ-Δ ADCs that we have described so far contain integrators, which are low pass filters, whose passband extends from dc. Thus, their quantization noise is pushed up in frequency. At present, most commercially available Σ-Δ ADCs are of this type (although some which are intended for use in audio or telecommunications applications contain bandpass rather than lowpass digital filters to eliminate any system dc offsets). But there is no particular reason why the filters of the Σ-Δ modulator should be LPFs, except that traditionally ADCs have been thought of as being baseband devices, and that integrators are somewhat easier to construct than bandpass filters. If we replace the integrators in a Σ-Δ ADC with bandpass filters (BPFs) as shown in Figure 12, the quantization noise is moved up and down in frequency to leave a virtually noise-free region in the pass band (see References 10, 11, and 12). If the digital filter is then programmed to have its pass band in this region, we have a Σ-Δ ADC with a bandpass, rather than a lowpass characteristic. Such devices would appear to be useful in direct IF-to-digital conversion, digital radios, ultrasound, and other undersampling applications. However, the modulator and the digital BPF must be designed for the specific set of frequencies required by the system application, thereby somewhat limiting the flexibility of this approach.
In an undersampling application of a bandpass $\Sigma$-$\Delta$ ADC, the minimum sampling frequency must be at least twice the signal bandwidth, $BW$. The signal is centered around a carrier frequency, $f_c$. A typical digital radio application using a 455-kHz center frequency and a signal bandwidth of 10 kHz is described in Reference 11. An oversampling frequency $Kf_s = 2$ MSPS and an output rate $f_s = 20$ kSPS yielded a dynamic range of 70 dB within the signal bandwidth.

Another example of a bandpass is the AD9870 IF Digitizing Subsystem having a nominal oversampling frequency of 18 MSPS, a center frequency of 2.25 MHz, and a bandwidth of 10 kHz to 150 kHz (see details in Reference 12).

**SUMMARY**

Sigma-delta ADCs and DACs have proliferated into many modern applications including measurement, voiceband, audio, etc. The technique takes full advantage of low cost CMOS processes and therefore makes integration with highly digital functions such as DSPs practical. Modern techniques such as the multi-bit data scrambled architecture minimize problems with idle tones which plagued early $\Sigma$-$\Delta$ products. Resolutions up to 24-bits are currently available, and the requirements on analog antialiasing/anti-imaging filters are greatly relaxed due to oversampling. The internal digital filter in audio $\Sigma$-$\Delta$ ADCs can be designed for linear phase, which is a major requirement in those applications. For high resolution $\Sigma$-$\Delta$ ADCs designed for measurement applications, the digital filter is generally designed so that zeros occur at the mains frequencies of 50 Hz and 60 Hz.

Many $\Sigma$-$\Delta$ converters offer a high level of user programmability with respect to output data rate, digital filter characteristics, and self-calibration modes. Multi-channel $\Sigma$-$\Delta$ ADCs are now available for data acquisition systems, and most users are well-educated with respect to the settling time requirements of the internal digital filter in these applications.

**REFERENCES**


INTRODUCTION

The pipelined subranging ADC architecture dominates today's applications where sampling rates of greater than 5 MSPS to 10 MSPS are required. Although the flash (all-parallel) architecture (see Tutorial MT-020) dominated the 8-bit video IC ADC market in the 1980s and early 1990s, the pipelined architecture has largely replaced the flash ADC in modern applications. There are a small number of high power Gallium Arsenide (GaAs) flash converters with sampling rates greater than 1 GHz, but resolution is limited to 6 or 8 bits. However, the flash converter still remains a popular building block for higher resolution pipelined ADCs.

Applications for pipelined ADCs include video, image processing, communications, and a myriad of others. The architecture lends itself to a variety of relatively low cost IC processes, CMOS and BiCMOS being the most popular. Current technology yields 12- to 16-bit resolution at sampling rates greater than 100 MSPS.

BASIC SUBRANGING ADC ARCHITECTURE

The pipelined ADC had its origins in the subranging architecture which was first used in the 1950s as a means to reduce the component count and power in tunnel diode and vacuum tube flash ADCs (see References 1, 2). A block diagram of the subranging architecture is shown in Figure 1, where a 6-bit, two-stage ADC is shown.

The output of the input sample-and-hold (SHA) is digitized by the first-stage 3-bit sub-ADC (SADC)—a flash converter. The coarse 3-bit MSB conversion is then converted back to an analog signal using a 3-bit sub-DAC (SDAC). The SDAC output is then subtracted from the SHA output, amplified, and applied to a second-stage 3-bit SADC. The "residue signal" is then digitized by the 3-bit second-stage SADC, thereby generating the three LSBs of the total 6-bit output word. This type of ADC is generally referred to as "subranging" because the input range is subdivided into a number of smaller ranges (subranges) which are, in turn, further subdivided.
This subranging ADC can best be analyzed by examining the residue waveform at the input to the second-stage ADC as shown in Figure 2. This waveform assumes a low frequency ramp input signal to the overall ADC. In order for there to be no missing codes, the residue waveform must exactly fill the input range of the second-stage ADC, as shown in the ideal case of Figure 2A. This implies that both the $N_1$ SADC and the $N_1$ SDAC must be better than $N_1 + N_2$ bits accurate—in the example shown, $N_1 = 3$, $N_2 = 3$, and $N_1 + N_2 = 6$. This architecture, as shown, is useful for resolutions up to about 8 bits ($N_1 = N_2 = 4$), however maintaining better than 8-bit alignment between the two stages (over temperature variations, in particular) can be difficult. The situation shown in Figure 2B will result in missing codes when the residue waveform goes outside the range of the $N_2$ SADC, "R", and falls within the "X" or "Y" regions—caused by a nonlinear $N_1$ SADC or interstage gain and/or offset mismatch.

![Figure 2: Residue Waveform at Input of Second-Stage SADC](image)

When the interstage alignment is not correct, missing codes will appear in the overall ADC transfer function as shown in Figure 3. If the residue signal goes into positive overrange (the "X" region), the output first "sticks" on a code and then "jumps" over a region leaving missing codes. The reverse occurs if the residue signal is negative overrange.
At this point it is worth noting that there is no particular reason—other than certain design issues beyond the scope of this discussion—why there must be an equal number of bits per stage in the subranging architecture. In addition, there can be more than two stages. Regardless, the architecture as shown in Figure 1 is limited to approximately 8-bit resolution unless some form of error correction is added.

Figure 4 shows a popular 8-bit 15-MSPS subranging ADC manufactured by Computer Labs, Inc. in the mid-1970s. This converter was a basic two-stage subranging ADC with two 4-bit flash converters—each composed of 8 dual AM687 high speed comparators. The interstage offset adjustment potentiometer allowed the transfer function to be optimized in the field. This ADC was popular in early digital video products such as frame stores and time base correctors.
In order to reliably achieve higher than 8-bit resolution using the subranging approach, a technique generally referred to as digitally corrected subranging, digital error correction, overlap bits, redundant bits, etc. is utilized. This method was referred to in literature as early as 1964 by T. C. Verster (Reference 3) and quickly became widely known and utilized (References 4-7). The fundamental concept is illustrated using the residue waveform shown in Figure 5.

![Figure 5: Error Correction Using Added Quantization Levels for N1 = 3](image)

The residue waveform is shown for the specific case where N1 = 3 bits. In a standard subranging ADC, the residue waveform must exactly fill the input range of the N2 SADC—it must stay within the region designated R. The missing code problem is solved by adding extra quantization levels in the positive overrange region X and the negative overrange region Y. These additional levels require additional comparators in the basic N2 flash SADC. The scheme works as follows. As soon as the residue enters the X region, the N2 SADC should return to all-zeros and start counting up again. Also, the code 001 must be added to the output of the N1 SADC to make the MSBs read the correct code. The figure labels the uncorrected MSB regions on the lower part of the waveform and the corrected MSB regions on the upper part of the waveform. A similar situation occurs when the residue waveform enters the negative overrange region Y. Here, the first quantization level in the Y region should generate the all-ones code, and the additional overrange comparators should cause the count to decrease. In the Y region, the code 001 must be subtracted from the MSBs to produce the corrected MSB code. It is important to understand that in order for this correction method to work properly, the N1 SDAC must be more accurate than the total resolution of the ADC. Nonlinearity or gain errors in the N1 SDAC affect the amplitude of the vertical "jump" portions of the residue waveform and therefore can produce missing codes in the output.

Horna in a 1972 paper (Reference 6) describes an experimental 8-bit 15-MSPS error corrected subranging ADC using Motorola MC1650 dual ECL comparators as the flash converter building blocks. Horna adds additional comparators in the second flash converter and describes this procedure in more detail. He points out that the correction logic can be greatly simplified by adding an appropriate offset to the residue waveform so that there is never a negative overrange condition. This eliminates the need for the subtraction function—only an adder is required. The MSBs are either passed through unmodified, or 1 LSB (relative to the N1 SADC) is added to them, depending on whether the residue signal is in range or overrange.

Modern digitally corrected subranging ADCs generally obtain the additional quantization levels by using an internal ADC with higher resolution for the N2 SADC. For instance, if one additional bit is added to the N2 SADC, its range is doubled—then the residue waveform can go outside either end of the range by ½ LSB referenced to the N1 SADC. Adding two extra bits to N2 allows the residue waveform to go outside either end of the range by 1¼ LSBs referenced to the N1 SADC. The residue...
waveform is offset using Horna’s technique such that only a simple adder is required to perform the correction logic. The details of how all this works are not immediately obvious, and can best be explained by going through an actual example of a 6-bit ADC with a 3-bit MSB SADC and a 4-bit LSB SADC providing one bit of error correction. The block diagram of the example ADC is shown in Figure 6.

![Block Diagram](image)

**Figure 6: A 6-Bit Subranging Error Corrected ADC, N1 = 3, N2 = 4**

After passing through an input sample-and-hold, the signal is digitized by the 3-bit SADC, reconstructed by a 3-bit SDAC, subtracted from the held analog signal and then amplified and applied to the second 4-bit SADC. The gain of the amplifier, G, is chosen so that the residue waveform occupies ½ the input range of the 4-bit SADC. The 3 LSBs of the 6-bit output data word go directly from the second SADC to the output register. The MSB of the 4-bit SADC controls whether or not the adder adds 001 to the 3 MSBs. The carry output of the adder is used in conjunction with some simple overrange logic to prevent the output bits from returning to the all-zeros state when the input signal goes outside the positive range of the ADC.

The residue waveform for a full-scale ramp input will now be examined in more detail to explain how the correction logic works. Figure 7 shows the ideal residue waveform assuming perfect linearity in the first ADC and perfect alignment between the two stages. Notice that the residue waveform occupies exactly ½ the range of the N2 SADC. The 4-bit digital output of the N2 SADC are shown on the left-hand side of the figure. The regions defined by the 3-bit uncorrected N1 SADC are shown at the top of the figure. The regions defined by the 3-bit corrected N1 ADC are shown are shown at the top of the figure.
Following the residue waveform from left-to-right—as the input first enters the overall ADC range at $-FS$, the N2 SADC begins to count up, starting at 0000. When the N2 SADC reaches the 1000 code, 001 is added to the N1 SADC output causing it to change from 000 to 001. As the residue waveform continues to increase, the N2 SADC continues to count up until it reaches the code 1100, at which point the N1 SADC switches to the next level, the SDAC switches and causes the residue waveform to jump down to the 0100 output code. The adder is now disabled because the MSB of the N2 SADC is zero, so the N1 SADC output remains 001. The residue waveform then continues to pass through each of the remaining regions until $+FS$ is reached.

This method has some clever features worth mentioning. First, the overall transfer function is offset by $\frac{1}{2}$ LSB referred to the MSB SADC (which is $\frac{1}{16}$th FS referred to the overall ADC analog input). This is easily corrected by injecting an offset into the input sample-and-hold. It is well-known that the points at which the internal N1 SADC and SDAC switch are the most likely to have additional noise and are the most likely to create differential nonlinearity in the overall ADC transfer function. Offsetting them by $\frac{1}{16}$th FS ensures that low level signals (less than $\pm\frac{1}{16}$th FS) near zero volts analog input do not exercise the critical switching points and gives low noise and excellent DNL where they are most needed in communications applications. Finally, since the ideal residue signal is centered within the range of the N2 SADC, the extra range provided by the N2 SADC allows up to a $\pm\frac{1}{16}$th FS error in the N1 SADC conversion while still maintaining no missing codes.

Figure 8 shows a residue signal where there are errors in the N1 SADC. Notice that there is no effect on the overall ADC linearity provided the residue signal remains within the range of the N2 SADC. As long as this condition is met, the error correction method described corrects for the following errors: sample-and-hold droop error, sample-and-hold settling time error, N1 SADC gain error, N1 SADC offset error, N1 SDAC offset error, N1 SADC linearity error, residue amplifier offset error. In spite of its ability to correct all these errors, it should be emphasized that this method does not correct for gain and linearity errors associated with the N1 SDAC or gain errors in the residue amplifier. The errors in these parameters must be kept less than 1 LSB referred to the N-bits of the overall subranging ADC. Another way to look at this requirement is to realize that the amplitude of the vertical "jump" transitions of the residue waveform, corresponding to the N1 SADC and SDAC changing levels, must remain within $\pm\frac{1}{2}$ LSB referenced to the N2 SADC input in order for the correction to prevent missing codes.
The error-corrected subranging ADC shown in Figure 6 does not have "pipeline" delay. The input SHA remains in the hold mode during the time required for the following events occur: the first-stage SADC makes its decision, its output is reconstructed by the first-stage SDAC, the SDAC output is subtracted from the SHA output, amplified, and digitized by the second-stage SADC. After the digital data passes through the error correction logic and the output registers, it is ready for use, and the converter is ready for another sampling clock input.

**PIPELINED SUBRANGING ADCs INCREASE SPEED**

The pipelined architecture shown in Figure 9 is a digitally corrected subranging architecture in which each stage operates on the data for one-half the sampling clock cycle and then passes its residue output to the next stage in the pipeline, prior the next half cycle. The interstage track-and-hold (T/H) serves as an analog delay line— timing is set such that it enters the hold mode when the first stage conversion is complete. This gives more settling time for the internal SADCs, SDACs, and amplifiers, and allows the pipelined converter to operate at a much higher overall sampling rate than a non-pipelined version.

The term "pipelined" architecture refers to the ability of one stage to process data from the previous stage during any given phase of the sampling clock cycle. At the end of each phase of a particular clock cycle, the output of a given stage is passed on to the next stage using the T/H functions, and new data is shifted into the stage. Of course this means that the digital outputs of all but the last stage in the "pipeline" must be stored in the appropriate number of shift registers so that the digital data arriving at the correction logic corresponds to the same sample.
Figure 10 shows a timing diagram of a typical pipelined subranging ADC. Notice that the phases of the clocks to the T/H amplifiers are alternated from stage to stage such that when a particular T/H in the ADC enters the hold mode it holds the sample from the preceding T/H, and the preceding T/H returns to the track mode. The held analog signal is passed along from stage to stage until it reaches the final stage in the pipelined ADC—in this case, a flash converter. When operating at high sampling rates, it is critical that the differential sampling clock be kept at a 50% duty cycle for optimum performance. Duty cycles other than 50% affect all the T/H amplifiers in the chain—some will have longer than optimum track times and shorter than optimum hold times; while others suffer exactly the reverse condition. Many newer pipelined ADCs including the 12-bit, 65-MSPS AD9235 and the 12-bit, 170-/210-MSPS AD9430 have on-chip clock conditioning circuits to control the internal duty cycle and maintain rated performance even if there is some variation in the external clock duty cycle.

Figure 10: Clock Issues in Pipelined ADCs

The effects of the "pipeline" delay (sometimes called "latency") in the output data are shown in Figure 11 for the AD9235 12-bit 65-MSPS ADC where there is a 7-clock cycle pipeline delay.

Figure 11: Typical Pipelined ADC Timing for AD9235 12-Bit, 65-MSPS ADC
Note that the pipeline delay is a function of the number of stages and the particular architecture of the ADC under consideration—the data sheet should always be consulted for the exact details of the relationship between the sampling clock and the output data timing. In many applications the pipeline delay will not be a problem, but if the ADC is inside a feedback loop the pipeline delay may cause instability. The pipeline delay can also be troublesome in multiplexed applications or when operating the ADC in a "single-shot" mode. Other ADC architectures—such as successive approximation—are better suited to these types of applications.

A subtle issue relating to most CMOS pipelined ADCs is their performance at low sampling rates. Because the internal timing generally is controlled by the external sampling clock, very low sampling rates extend the hold times for the internal track-and-holds to the point where excessive droop causes conversion errors. Therefore, most pipelined ADCs have a specification for minimum as well as maximum sampling rate. Obviously, this precludes operation in single-shot or burst-mode applications—where the SAR ADC architecture is more appropriate.

It is often erroneously assumed that all subranging ADCs are pipelined, and that all pipelined ADCs are subranging. While it is true that most modern subranging ADCs are pipelined in order to achieve the maximum possible sampling rate, they don't necessarily have to be pipelined if designed for use at much lower speeds. For instance, the leading edge of the sampling clock could initiate the conversion process, and any additional clock pulses required to continue the conversion could be generated internal to the ADC using an on-chip timing circuit. At the end of the conversion process, an end-of-conversion or data-ready signal could be generated as an external indication that the data corresponding to that particular sampling edge is valid. This "no latency" approach is not often used for the obvious reason that the overall sampling rate is greatly reduced by eliminating the pipelined structure.

Conversely, there are some ADCs which use other architectures than subranging and are pipelined. For instance, most flash converters use an extra set of output latches (in addition to the latch associated with the parallel comparators) which introduces pipeline delay in the output data (see Tutorial MT-020). Another example of a non-subranging architecture which generally has quite a bit of pipeline delay is sigma-delta which is covered in detail in Tutorial MT-022 and Tutorial MT-023. Note, however, that it is possible to modify the timing of a normal sigma-delta ADC, reduce the output data rate, and make a "no latency" sigma-delta ADC.

RECYCLATING SUBRANGING PIPELINED ADC

Another less popular type of error corrected subranging architecture is the recirculating subranging ADC. This is shown in Figure 12 and was proposed in a 1966 paper by Kinniment, et.al. (Reference 5). The concept is similar to the error corrected subranging architecture previously discussed, but in this architecture, the residue signal is recirculated through a single ADC and DAC stage using switches and a programmable gain amplifier (PGA). Figure 12 shows the additional buffer registers required to store the pipelined data resulting from each conversion such that the data into the correction logic (adder) corresponds to the same sample. The recirculating architecture show in Figure 12 is similar to some integrated circuit ADCs introduced in the early 1990s, such as the AD678 (12-bits, 200 kSPS) and AD679 (14-bits, 128 kSPS). Today, ADCs with these resolutions and sampling rates are implemented in a much more efficient and cost effective manner using the successive approximation architecture discussed in Tutorial MT-021.
Figure 12: Kinniment, et. al., 1966 Pipelined 7-bit, 9-MSPS Recirculating ADC Architecture

MODERN MONOLITHIC PIPELINED ADCs FOR VIDEO AND IMAGE PROCESSING

The discussion of error corrected pipelined ADCs concludes with a few examples of modern integrated circuit implementations of the popular architecture. These examples show the flexibility of the technique in optimizing ADC performance at different resolutions, sampling rates, power dissipation, etc.

The video market currently uses ADCs with resolutions of 8 to 12-bits, and sampling rates from 54 MSPS to 140 MSPS. Most of these ADCs are now integrated into chips which perform further digital signal processing, such as the conversion between the various existing video standards (composite, RGB, Y/C, Y/Pb/Pr). These ICs perform a considerable amount of digital processing as shown in the ADV-Series Video Decoders from Analog Devices. The ADC architecture is generally pipelined, the process CMOS, and total package power dissipation ranges from 250 mW to 600 mW. Similar converters are used in other Mixed Signal Front End products (MxFE®) for wireless, networking, set top box, and IF signal processing. Another family of similar products is used in CCD Image Processing applications for cameras and camcorders.

In the current "stand-alone" 8-bit ADC market, the pipelined architecture is implemented in the 8-bit 250 MSPS, AD9480 (LVDS outputs) and AD9481 (demuxed CMOS outputs) which dissipate 700 mW and 600 mW, respectively.

PIPELINED ADCs FOR WIDEBAND COMMUNICATIONS

The demand for wide dynamic range (high SFDR) ADCs suitable for communications applications led to the development of a breakthrough product in 1995, the AD9042 12-bit, 41-MSPS ADC (see Reference 8). A block diagram of the converter is shown in Figure 13.
The AD9042 uses an error corrected subranging architecture composed of a 6-bit MSB ADC/DAC followed by a 7-bit LSB ADC and uses one bit of error correction in the second stage. The AD9042 yields 80-dB SFDR performance over the Nyquist bandwidth at a sampling rate of 41 MSPS. Fabricated on a high speed complementary bipolar process, the device dissipates 600 mW and operates on a single +5 V supply.

In order to meet the need for lower cost, lower power devices, Analog Devices initiated a family of CMOS high performance ADCs such as the AD9225 12-bit, 25-MSPS ADC released in 1998. The AD9225 dissipates 280 mW, has 85-dB SFDR, and operates on a +5 V supply.

The AD9235 12-bit 65-MSPS CMOS ADC released in 2001 shows the progression of CMOS high performance converters. The AD9235 operates on a single +3 V supply, dissipates 300 mW (at 65 MSPS), and has a 90-dB SFDR over the Nyquist bandwidth.

The 12-bit 210-MSPS AD9430 released in 2002 is fabricated on a BiCMOS process, has 80-dB SFDR up to 70-MHz inputs, operates on a single +3 V supply and dissipates 1.3 W at 210 MSPS. Output data is provided on two demuxed ports at 105 MSPS each in the CMOS mode or on a single port at 210 MSPS in the LVDS mode.

Another breakthrough product was the 14-bit 105-MSPS AD6645 ADC released in 2002 and fabricated on a high speed complementary bipolar process (XFCB), has 90-dB SFDR, operates on a single +5 V supply and dissipates 1.5 W.

THE LATEST IN PIPELINED ADCs

The 12-bit high speed ADC segment has seen significant improvements in speed, power, and performance, as reflected by the AD9236 12-bit 80 MSPS CMOS ADC with only 360 mW of power dissipation. The AD9236 is part of the pin-compatible family which includes the AD9215 (10-bit, 105 MSPS), AD9235 (12-bit, 65 MSPS) and the AD9245 (14-bit, 80 MSPS). These pin-compatible devices allow easy migration from 10-bits to 14-bits and sampling rates from 20 MSPS to higher rates.

In addition to single ADCs, dual ADCs and quad ADCs are available, including the AD9229 quad 12-bit 65 MSPS ADC with LVDS outputs. Power dissipation is 1.5 W, and the part is ideally suited for high density applications such as medical ultrasound.

In the 14-bit communications ADC area, the AD9244 14-bit, 65 MSPS is optimized for Nyquist input signals (dc to fs/2), has an SFDR of 86 dB, and dissipates only 550 mW on a CMOS process.

For higher input bandwidths and IF sampling, the AD9445 14-bit, 125 MSPS ADC is available with an SFDR of 95 dB (measured with a 170 MHz input), and a power dissipation of 2.6 W. The AD9445 is designed on a BiCMOS process.
Also for communications applications, the **AD9446** 16-bit, 100 MSPS ADC is optimized for high SNR (84 dB), dissipates 2.8 W, and is also designed on a BiCMOS process.

**SUMMARY**

The pipelined subranging ADC architecture virtually dominates where sampling rates of greater than a few MHz are required. There is some overlap between the SAR architecture and the pipelined architecture in the 2 to 5 MSPS region, but the application should easily dictate which architecture is more appropriate.

Resolutions from 8- to 16-bits are available in a variety of packages and configurations (signals, duals, triples, quads, etc.). Fine-line CMOS is by far the most popular process for these converters, and BiCMOS is used where it is necessary to obtain the ultimate in dynamic performance.

For a given sampling rate and resolution, pipelined ADCs are often differentiated by their dynamic performance. For example, the **AD9244** 14-bit, 65 MSPS ADC is optimized to handle input signals from dc to Nyquist (fs/2) and dissipates only 550 mW. If signals in higher Nyquist zones must be processed, the **AD9445** 14-bit, 125 MSPS ADC is available at 2.6 W on a more expensive BiCMOS process.

Selecting the proper pipelined ADC for a particular application requires a thorough understanding of not only system requirements but also a working knowledge of the architecture and the possible tradeoffs available. Simply treating the ADC as a "black box" often leads to the wrong choice.

**REFERENCES**


5. D. J. Kinniment, D. Aspinall, and D.B.G. Edwards, "High-Speed Analogue-Digital Converter," *IEE Proceedings*, Vol. 113, pp. 2061-2069, Dec. 1966. (a 7-bit 9MSPS three-stage pipelined error corrected converter is described based on recirculating through a 3-bit stage three times. Tunnel (Esaki) diodes are used for the individual comparators. The article also shows a proposed faster pipelined 7-bit architecture using three individual 3-bit stages with error correction. The article also describes a fast bootstrapped diode-bridge sample-and-hold circuit).


INTRODUCTION

The "folding" architecture is one of a number of possible serial or bit-per-stage architectures. Various architectures exist for performing A/D conversion using one stage per bit, and the overall concept is shown in Figure 1. A multistage pipelined subranging ADC with one bit per stage and no error correction is basically a bit-per-stage converter. In practice, this type of pipelined converter generally uses a 1.5 bit per stage approach to provide error correction (this is discussed in more detail in Reference 1).

In the bit-per-stage ADC, the input signal must be held constant during the entire conversion cycle. There are N stages, each of which have a "bit" output and a "residue" output. The residue output of one stage is the input to the next. The last bit is detected with a single comparator as shown.

It is possible to combine the bit-per-stage architecture with other architectures. For example, the residue output of the final stage can be further digitized by a flash converter, thereby providing more resolution.

One of the first references to these architectures appeared in an article by B. D. Smith in 1956 (Reference 2). Smith indicates, however, that previous work had been done at M.I.T. by R. P. Sallen in a 1949 thesis. In the article, Smith describes both the binary and the Gray (or folding) transfer functions required to implement the A/D conversion.

BINARY AND FOLDING BIT-PER-STAGE (SERIAL) ADCs

The basic stage for performing a single binary bit conversion is shown in Figure 2. It consists of a gain-of-two amplifier, a comparator, and a 1-bit DAC (changeover switch). Assume that this is the first stage of the ADC. The MSB is simply the polarity of the input, and that is detected with the comparator which also controls the 1-bit DAC. The 1-bit DAC output is summed with the output of the gain-of-two amplifier. The resulting residue output is then applied to the next stage. In order to better understand how the circuit works, the diagram shows the residue output for the case of a linear ramp input voltage which traverses the entire ADC range, \(-V_R\) to \(+V_R\). Notice that the polarity of the residue output determines the binary bit output of the next stage.
A simplified 3-bit serial-binary bit-per-stage ADC is shown in Figure 3, and the residue outputs are shown in Figure 4. Again, the case is shown for a linear ramp input voltage whose range is between $-V_R$ and $+V_R$. Each residue output signal has discontinuities which correspond to the point where the comparator changes state and causes the DAC to switch. The fundamental problem with this architecture is the discontinuity in the residue output waveforms. Adequate settling time must be allowed for these transients to propagate through all the stages and settle at the final comparator input. As presented here, the prospects of making this architecture operate at high speed are dismal. However, using the 1.5-bit-per-stage pipelined architecture (see Reference 1) makes it much more attractive at high speeds.
Although the binary method is discussed in his paper, B. D. Smith also describes a much preferred bit-per-stage architecture based on absolute value amplifiers (magnitude amplifiers, or simply *MagAMPs™*). This scheme has often been referred to as *serial-Gray* (since the output coding is in Gray code), or *folding* converter because of the shape of the transfer function. Performing the conversion using a transfer function that produces an initial Gray code output has the advantage of minimizing discontinuities in the residue output waveforms and offers the potential of operating at much higher speeds than the binary approach.

The basic folding stage is shown functionally in Figure 5 along with its transfer function. The input to the stage is assumed to be a linear ramp voltage whose range is between $-V_R$ and $+V_R$. The comparator detects the polarity of the input signal and provides the Gray bit output for the stage. It also determines whether the overall stage gain is $+2$ or $-2$. The reference voltage $V_R$ is summed with the switch output to generate the residue signal which is applied to the next stage. The polarity of the residue signal determines the Gray bit for the next stage. The transfer function for the folding stage is also shown in Figure 5.
A 3-bit MagAMP folding ADC is shown in Figure 6, and the corresponding residue waveforms in Figure 7. As in the case of the binary bit-per-stage ADC, the polarity of the residue output signal of a stage determines the value of the Gray bit for the next stage. The polarity of the input to the first stage determines the Gray MSB; the polarity of R1 output determines the Gray bit-2; and the polarity of R2 output determines the Gray bit-3. Notice that unlike the binary ripple ADC, there is no abrupt transition in any of the folding stage residue output waveforms. This makes operation at high speeds quite feasible.
The key to operating this architecture at high speeds is the folding stage. N. E. Chasek of Bell Telephone Labs describes a circuit for generating the folding transfer function using nested diode bridges in a patent filed in 1960 (Reference 3). This circuit made use of solid-state devices, but required different reference voltages for each stage (see Figure 8). Chasek's circuit also suffered from loss of headroom and gain when several stages were cascaded to form higher resolution converters as shown in Figure 9. What is really needed to make the folding ADC work at high resolutions is nearly ideal voltage or current rectification.

Figure 7: Input and Residue Waveforms for 3-Bit Folding ADC

Figure 8: 3-Bit Folding ADC Based on N. E. Chasek's Design
F. D. Waldhaur of Bell Telephone Labs remedied the problems of Chasek’s nested diode bridge circuits in a classic patent filed in 1962 (Reference 4). Figure 10 shows Waldhaur’s elegant implementation of the folding transfer function using solid state op amps with diodes in the feedback loop. The gain-of-two op amps allow the same reference voltages to be used for each stage and maintain the same signal level at each residue output with nearly ideal rectification.
J. O. Edson and H. H. Henning describe the operation and performance of this type of ADC in greater detail in a 1965 *Bell System Technical Journal* article (Reference 5). An operational 9-bit, 6-MSPS ADC of this type was used in experimental studies on 224-Mbit/second PCM terminals. These terminals were supposed to handle data as well as voice signals. The voiceband objective was to digitize an entire 600-channel, 2.4-MHz FDM band, therefore requiring a minimum sampling rate of approximately 6 MSPS.

It is interesting to note that the experimental terminal was also supposed to handle video as well, which required a higher sampling rate of approximately 12-MSPS. For this requirement, the latest (and final) generation Bell Labs’ electron beam coder (see Tutorial MT-020) was needed to meet the ADC requirement, as the solid-state coder based on Waldhaur's patent did not have the necessary accuracy at the higher sampling rates.

The first commercial ADC using Waldhaur's Gray code architecture was the 8-bit, 10-MSPS HS-810 from Computer Labs, Inc., in 1966. The instrument used all discrete transistor circuits (no ICs) and was designed to be mounted in a 19” rack as shown in Figure 11 for an early experimental digital radar receiver application. The 8-bit, 10-MSPS converter contained its own linear power supply, dissipated nearly 150 watts, and sold for approximately $10,000. The same technology was used to produce 9-bit, 5-MSPS and 10-bit 3-MSPS versions. Although the next generation of Computer Labs' designs would take advantage of modular op amps (Computer Labs OA-125 and FS-125), ICs such as the Fairchild µA710/711 comparators, as well as 7400 TTL logic, the first ADCs offered used all discrete devices. These early high speed ADCs produced by Computer Labs were primarily used in research and development projects associated with radar receiver development by companies such as Raytheon, General Electric, and MIT Lincoln Labs.

The folding Gray code architecture was used in a few instrument and modular ADCs in the early 1970s, such as the HS-810, but commercial high speed ADCs primarily used either the flash or the error-corrected subranging architecture in the 1980s. With improvements in IC processes, there was, however, continued interest in the folding architecture in the late 1970s and throughout the 1980s—with quite a number of experimental designs reported in the various journals over the period (References 6-10).

Analog Devices developed the first high speed fully complementary bipolar (CB) process in the mid-1980s, and in 1994 Frank Murden and Carl Moreland filed patents on a significantly improved current-steering architecture for a Gray code MagAMP™-based ADC (References 11-15). The technique was first implemented for building block cores in the AD9042 12-bit, 41-MSPS ADC released in 1995, and refinements of the technique and a higher speed CB process, XFCB, (References 16 and 17) pushed the core technology to 14-bits with the release of the AD6644 14-bit 65-MSPS ADC in 1999, the AD6645 14-bit 80-MSPS ADC in
Modern IC circuit designs implement the transfer function using current-steering open-loop gain techniques which can be made to operate much faster. Fully differential stages (including the SHA) also provide speed, lower distortion, and yield 8-bit accurate folding stages with no requirement for thin film resistor laser trimming.

An example of a fully differential gain-of-two MagAMP folding stage is shown in Figure 12 (see References 11, 12, 14). The differential input signal is applied to the degenerated-emitter differential pair Q1, Q2 and the comparator. The differential input voltage is converted into a differential current which flows in the collectors of Q1, Q2. If \(+IN\) is greater than \(-IN\), cascode-connected transistors Q3, Q6 are on, and Q4, Q6 are off. The differential signal currents therefore flow through the collectors of Q3, Q6 into level-shifting transistors Q7, Q8 and into the output load resistors, developing the differential output voltage between \(+OUT\) and \(-OUT\). The overall differential voltage gain of the circuit is two.

The differential residue output voltage of the stage drives the next stage input, and the comparator output represents the Gray code output for the stage.

The MagAMP architecture offers lower power and can be extended to sampling rates previously dominated by flash converters. For example, the AD9054A 8-bit, 200-MSPS ADC is shown in Figure 13 and was first introduced in 1997. The device is fabricated on a high speed complementary bipolar process, and power dissipation is 500 mW. The first five bits (Gray code) are derived from five differential MagAMP stages. The differential residue output of the fifth MagAMP stage drives a 3-bit flash converter, rather than a single comparator.

The Gray-code output of the five MagAMPs and the binary-code output of the 3-bit flash are latched, all converted into binary,
and latched again in the output data register. Because of the high data rate, a demultiplexed output option is provided.

**Figure 13: AD9054A 8-bit, 200-MSPS ADC Introduced in 1997**

Recent introductions in the 8-bit high speed area have utilized CMOS processes and the pipelined subranging architecture, such as the 8-bit 250 MSPS, AD9480 (LVDS outputs) and AD9481 (demuxed CMOS outputs) which dissipate 700 mW and 600 mW, respectively.

**SUMMARY**

Although initially used in pioneering instrument ADCs at Bell Labs and Computer Labs in the 1960s, the flash the pipelined subranging architectures have dominated the high speed ADC marketplace. Although there have been a number of ICs designed using the folding architecture, it has never attained the popularity of the pipelined subranging ADC. Nevertheless, it is important to know that it exists because it may regain popularity in the future as IC processes evolve.

**REFERENCES**


INTRODUCTION

Although counting-based ADCs are not well suited for high speed applications, they are ideal for high resolution low frequency applications, especially when combined with integrating techniques such as dual-, triple-, and quad-slope.

A. H. REEVES’ 5-BIT COUNTING ADC

Counting ADCs had their origins in early experimental pulse code modulation (PCM) systems during the late 1930s. The first ADC suitable for PCM applications was the one documented by A. H. Reeves in his comprehensive 1939 PCM patent (Reference 1). A simplified diagram of the ADC is repeated here in Figure 1. The early ADCs for PCM typically had 5 to 7 bits of resolution and sampling rates of 6 kSPS to 10 kSPS. Interestingly enough, Reeves’ ADC was based on a counting technique, probably because of his general interests in counters—the Eccles-Jordan bistable multivibrator had been invented only a few years earlier. However other architectures such as flash (Tutorial MT-020), successive approximation (Tutorial MT-021), subranging and pipelined (Tutorial MT-024), and bit-per-stage (Tutorial MT-025) were much more widely used in later PCM applications.

The counting ADC technique basically uses a sampling pulse to take a sample of the analog signal, set an R/S flip-flop, and simultaneously start a controlled ramp voltage. The ramp voltage is compared with the input, and when they are equal, a pulse is generated which resets the R/S flip-flop. The output of the flip-flop is a pulse whose width is proportional to the analog signal at the sampling instant. This pulse width modulated (PWM) pulse controls a gated oscillator, and the number of pulses out of the gated oscillator represents the quantized value of the analog signal. This pulse train can be easily converted to a binary word by driving a counter. In Reeves’ system, a master clock of 600 kHz was used, and a 100:1 divider generated the 6-kHz sampling pulses. The system uses a 5-bit counter, and 31 counts (out of the 100 counts between sampling pulses) therefore represents a full-scale signal. The technique can obviously be extended to higher resolutions.
CHARGE RUN-DOWN ADC

The charge run-down ADC architecture (see Reference 2) shown in Figure 2 first samples the analog input and stores the voltage on a fixed capacitor. The capacitor is then discharged with a constant current source, and the time required for complete discharge is measured using a counter. Notice that in this approach, the overall accuracy is dependent on the quality and magnitude of the capacitor, the magnitude of the current source, as well as the accuracy of the timebase.

RAMP RUN-UP ADC

In the ramp run-up architecture shown in Figure 3 (see Reference 3), a ramp generator is started at the beginning of the conversion cycle. The counter then measures the time required for the ramp voltage to equal the analog input voltage. The counter output is therefore proportional to the value of the analog input. In an alternate version (shown dotted in Figure 3), the ramp voltage generator is replaced by a DAC which is driven by the counter output. The advantage of using the ramp is that the ADC is always monotonic, whereas overall monotonicity is determined by the DAC when it is used as a substitute.

The accuracy of the ramp run-up ADC depends on the accuracy of the ramp generator (or the DAC) as well as the oscillator. In order to process ac signals, a sample-and-hold must be used such that the analog input is stable during the conversion cycle. Note that the ramp run-up architecture is quite similar to the Reeves' counting architecture shown in Figure 1.
The tracking ADC architecture shown in Figure 4 (see References 4 and 5) continually compares the input signal with a reconstructed representation of the input signal. The up/down counter is controlled by the comparator output. If the analog input exceeds the DAC output, the counter counts up until they are equal. If the DAC output exceeds the analog input, the counter counts down until they are equal. It is evident that if the analog input changes slowly, the counter will follow, and the digital output will remain close to its correct value. If the analog input suddenly undergoes a large step change, it will be many hundreds or thousands of clock cycles before the output is again valid. The tracking ADC therefore responds quickly to slowly changing signals, but slowly to a quickly changing one.
The simple analysis above ignores the behavior of the ADC when the analog input and DAC output are nearly equal. This will depend on the exact nature of the comparator and counter. If the comparator is a simple one, the DAC output will cycle by 1 LSB from just above the analog input to just below it, and the digital output will, of course, do the same—there will be 1 LSB of flicker. Note that the output in such a case steps every clock cycle, irrespective of the exact value of analog input, and hence always has unity Mark/Space ratio. In other words, there is no possibility of taking a mean value of the digital output and increasing resolution by oversampling.

A more satisfactory, but more complex arrangement would be to use a window comparator with a window 1-2 LSB wide. When the DAC output is high or low the system behaves as in the previous description, but if the DAC output is within the window, the counter stops. This arrangement eliminates the flicker, provided that the DAC DNL never allows the DAC output to step across the window for 1 LSB change in code.

Tracking ADCs are not very common. Their slow step response makes them unsuitable for many applications, but they do have one asset: their output is continuously available. Most ADCs perform conversions: i.e., on receipt of a "start convert" command (which may be internally generated), they perform a conversion and, after a delay, a result becomes available. Providing that the analog input changes slowly, the output of a tracking ADC is always available. This is valuable in synchro-to-digital and resolver-to-digital converters (SDCs and RDCs), and this is the application where tracking ADCs are most often used (see Tutorial MT-030). Another valuable characteristic of tracking ADCs is that a fast transient on the analog input causes the output to change only one count. This is very useful in noisy environments. Notice the similarity between a tracking ADC and a successive approximation ADC. Replacing the up/down counter with SAR logic yields the architecture for a successive approximation ADC.

REFERENCES
1. Alec Harley Reeves, "Electric Signaling System," U.S. Patent 2,272,070, filed November 22, 1939, issued February 3, 1942. Also French Patent 852,183 issued 1938, and British Patent 538,860 issued 1939. (the ground-breaking patent on PCM. Interestingly enough, the ADC and DAC proposed by Reeves are counting types, and not successive approximation).


INTRODUCTION

Soon after the discovery of the basic counting ADC architectures (see Tutorial MT-026) it was realized that much greater accuracy could be obtained using a combination of integrating and counting techniques. This led to the development of high accuracy dual-slope, triple-slope, and quad-slope ADCs. Although the proliferation of high resolution sigma-delta ADCs has made integrating architectures somewhat less popular, they are still used in a variety of precision applications such as digital voltmeters, etc.

MULTI-SLOPE ADCs

Introduced in the 1950s, the "dual-slope" ADC architecture was truly a breakthrough in ADCs for high resolution applications such as digital voltmeters, etc. (see References 1-4). A simplified diagram is shown in Figure 1, and the integrator output waveforms are shown in Figure 2.

![Figure 1: Dual Slope Integrating ADC](image)

The input signal is applied to an integrator; at the same time a counter is started, counting clock pulses. After a pre-determined amount of time \((T)\), a reference voltage having opposite polarity is applied to the integrator. At that instant, the accumulated charge on the integrating capacitor is proportional to the average value of the input over the interval \(T\). The integral of the reference is an opposite-going ramp having a slope of \(V_{\text{REF}}/RC\). At the same time, the counter is again counting from zero. When the integrator output reaches zero, the count is stopped, and the analog circuitry is reset. Since the charge gained is proportional to \(V_{\text{IN}} \times T\), and the equal amount of charge lost is proportional to \(V_{\text{REF}} \times t_x\), then the number of counts relative to the full scale count is proportional to \(t_x/T\), or \(V_{\text{IN}}/V_{\text{REF}}\). If the output of the counter is a binary number, it will therefore be a binary representation of the input voltage.
Dual-slope integration has many advantages. Conversion accuracy is independent of both the capacitance and the clock frequency, because they affect both the up-slope and the down-slope by the same ratio.

The fixed input signal integration period results in rejection of noise frequencies on the analog input that have periods that are equal to or a sub-multiple of the integration time $T$. Proper choice of $T$ can therefore result in excellent rejection of 50-Hz and 60-Hz line ripple as shown in Figure 3.

Errors caused by bias currents and the offset voltages of the integrating amplifier and the comparator as well as gain errors can be cancelled by using additional charge/discharge cycles to measure "zero" and "full-scale" and using the results to digitally correct the initial measurement, as in the quad-slope architecture discussed in Reference 5.

The triple-slope architecture (see References 6-8) retains the advantages of the dual-slope, but greatly increases the conversion speed at the cost of added complexity. The increase in conversion speed is achieved by accomplishing the reference integration (ramp-down) at two distinct rates: a high-speed rate, and a "vernier" lower speed rate. The counter is likewise divided into two sections, one for the MSBs and one for the LSBs. In a properly designed triple-slope converter, a significant increase in speed can be achieved while retaining the inherent linearity, differential linearity, and stability characteristics associated with dual-slope ADCs.
Figure 3: Frequency Response of Integrating ADC

REFERENCES


MT-028: Voltage-to-Frequency Converters

by Walt Kester and James Bryant

INTRODUCTION

A voltage-to-frequency converter (VFC) is an oscillator whose frequency is linearly proportional to a control voltage. The VFC/counter ADC is monotonic and free of missing codes, integrates noise, and can consume very little power. It is also very useful for telemetry applications, since the VFC, which is small, cheap and low powered can be mounted on the experimental subject (patient, wild animal, artillery shell, etc.) and communicate with the counter by a telemetry link as shown in Figure 1.

There are two common VFC architectures: the current-steering multivibrator VFC and the charge balance VFC (Reference 1). The charge balanced VFC may be made in asynchronous or synchronous (clocked) forms. There are many more VFO (variable frequency oscillator) architectures, including the ubiquitous 555 timer, but the key feature of VFCs is linearity—few VFOs are very linear.

The current-steering multivibrator VFC is actually a current to-frequency converter rather than a VFC, but, as shown in Figure 2, practical circuits invariably contain a voltage to-current converter at the input. The principle of operation is evident: the current discharges the capacitor until a threshold is reached, and when the capacitor terminals are reversed, the half cycle repeats itself. The waveform across the capacitor is a linear triangular wave, but the waveform on either terminal with respect to ground is the more complex waveform shown.
Practical VFCs of this type have linearities around 14 bits, and comparable stability, although they may be used in ADCs with higher resolutions without missing codes. The performance limits are set by comparator threshold noise, threshold temperature coefficient, and the stability and dielectric absorption (DA) of the capacitor, which is generally a discrete component. The comparator/voltage reference structure shown in the diagram is more of a representation of the function performed than the actual circuit used, which is much more integrated with the switching, and correspondingly harder to analyze.

This type of VFC is simple, inexpensive, and low powered, and most run from a wide range of supply voltages. They are ideally suited for low cost medium accuracy ADC and data telemetry applications.

The charge balance VFC shown in Figure 3 is more complex, more demanding in its supply voltage and current requirements, and more accurate. It is capable of 16-18 bit linearity.
The integrator capacitor is charged by the signal as shown in Figure 3. When it passes the comparator threshold, a fixed charge is removed from the capacitor, but the input current continues to flow during the discharge, so no input charge is lost. The fixed charge is defined by the precision current source and the pulse width of the precision monostable. The output pulse rate is thus accurately proportional to the rate at which the integrator charges from the input.

At low frequencies, the limits on the performance of this VFC are set by the stability of the current source and the monostable timing (which depends on the monostable capacitor, among other things). The absolute value and temperature stability of the integration capacitor do not affect the accuracy, although its leakage and dielectric absorption (DA) do. At high frequencies, second order effects, such as switching transients in the integrator and the precision of the monostable when it is retriggered very soon after the end of a pulse, take their toll on accuracy and linearity.

The changeover switch in the current source addresses the integrator transient problem. By using a changeover switch instead of the on/off switch more common on older VFC designs: (a) there are no on/off transients in the precision current source and (b) the output stage of the integrator sees a constant load—most of the time the current from the source flows directly in the output stage; during charge balance, it still flows in the output stage, but through the integration capacitor.

The changeover switch in the current source addresses the integrator transient problem. By using a changeover switch instead of the on/off switch more common on older VFC designs: (a) there are no on/off transients in the precision current source and (b) the output stage of the integrator sees a constant load—most of the time the current from the source flows directly in the output stage; during charge balance, it still flows in the output stage, but through the integration capacitor.

The stability and transient behavior of the precision monostable present more problems, but the issue may be avoided by replacing the monostable with a clocked bistable multivibrator. This arrangement is known as a synchronous VFC or SVFC and is shown in Figure 4.

The difference from the previous circuit is quite small, but the charge balance pulse length is now defined by two successive edges of the external clock. If this clock has low jitter, the charge will be very accurately defined. The output pulse will also be synchronous with the clock. SVFCs of this type are capable of up to 18 bit linearity and excellent temperature stability.

This synchronous behavior is convenient in many applications, since synchronous data transfer is often easier to handle than asynchronous. It does mean, however, that the output of an SVFC is not a pure tone (plus harmonics, of course) like a conventional VFC, but contains components harmonically related to the clock frequency. The display of an SVFC output on an oscilloscope is especially misleading and is a common cause of confusion—a change of input to a VFC produces a smooth change in the output frequency, but a change to an SVFC produces a change in probability density of output pulses N and N + 1 clock cycles after the previous output pulse, which is often misinterpreted as severe jitter and a sign of a faulty device (see Figure 5).
Another problem with SVFCs is nonlinearity at output frequencies related to the clock frequency. If we study the transfer characteristic of an SVFC, we find nonlinearities close to sub harmonics of the clock frequency $F_C$, as shown in Figure 6. They can be found at $F_C/3$, $F_C/4$, and $F_C/6$. This is due to stray capacitance on the chip (and in the circuit layout!) and the coupling of the clock signal into the SVFC comparator which causes the device to behave as an injection locked phase locked loop (PLL). This problem is intrinsic to SVFCs, but is not often serious: if the circuit card is well laid out, and clock amplitude and rate of change kept as low as practical, the effect is a discontinuity in the transfer characteristic of less than 8 LSBs (at 18 bit resolution) at $F_C/3$ and $F_C/4$, and less at other sub harmonics. This is frequently tolerable, since the frequencies where it occurs are known. Of course, if the circuit layout or decoupling is poor, the effect may be much larger, but this is the fault of poor design and not the SVFC itself.

Figure 5: VFC and SVFC Waveforms

Figure 6: SVFC Nonlinearity
It is evident that the SVFC is quantized, while the basic VFC is not. It does NOT follow from this that the counter/VFC ADC has higher resolution (neglecting nonlinearities) than the counter/SVFC ADC, because the clock in the counter also sets a limit to the resolution.

When a VFC has a large input, it runs quickly and (counting for a short time) gives good resolution, but it is hard to get good resolution in a reasonable sample time with a slow running VFC. In such a case, it may be more practical to measure the period of the VFC output (this does not work for an SVFC), but of course the resolution of this system deteriorates as the input (and the frequency) increases. However, if the counter/timer arrangement is made "smart," it is possible to measure the approximate VFC frequency and the exact period of not one, but N cycles (where the value of N is determined by the approximate frequency), and maintain high resolution over a wide range of inputs. The AD1170 modular ADC released in 1986 is an example of this architecture.

VFCs have more applications than as a component in ADCs. Since their output is a pulse stream, it may easily be sent over a wide range of transmission media (PSN, radio, optical, IR, ultrasonic, etc.). It need not be received by a counter, but by another VFC configured as a frequency-to-voltage converter (FVC). This gives an analog output, and a VFC FVC combination is a very useful way of sending a precision analog signal across an isolation barrier. There are a number of issues to be considered in building FVCs from VFCs, and these are considered in References 2-5.

**SUMMARY**

Analog Devices has a wide range of voltage to frequency converters (VFCs) for the instrumentation, industrial and automation markets (see Voltage-to-Frequency Converter Selection Table), including the AD537, AD650, AD652, AD654, and ADVFC32. They are ideally suited for use in analog-to-digital conversion (ADC), long term integration, linear frequency modulation and demodulation, and frequency-to-voltage conversion. Analog Devices’ latest family of VFCs, the AD7740, AD7741, and AD7742, are synchronous VFCs based on sigma-delta technology and provide high linearity in tiny packages at low cost.

**REFERENCES**

3. James M. Bryant, "Voltage-to-Frequency Converters," Application Note AN-361, Analog Devices, Inc. (a good overview of VFCs).
5. Steve Martin, "Using the AD650 Voltage-to-Frequency Converter as a Frequency-to-Voltage Converter," Application Note AN-279, Analog Devices, Inc. (a description of a frequency-to-voltage converter using the popular AD650 VFC).
Among the most popular position measuring sensors, optical encoders find use in relatively low reliability and low resolution applications. An incremental optical encoder (left-hand diagram in Figure 1) is a disc divided into sectors that are alternately transparent and opaque. A light source is positioned on one side of the disc, and a light sensor on the other side. As the disc rotates, the output from the detector switches alternately on and off, depending on whether the sector appearing between the light source and the detector is transparent or opaque. Thus, the encoder produces a stream of square wave pulses which, when counted, indicate the angular position of the shaft. Available encoder resolutions (the number of opaque and transparent sectors per disc) range from 100 to 65,000, with absolute accuracies approaching 30 arc-seconds (1/43,200 per rotation). Most incremental encoders feature a second light source and sensor at an angle to the main source and sensor, to indicate the direction of rotation. Many encoders also have a third light source and detector to sense a once-per-revolution marker. Without some form of revolution marker, absolute angles are difficult to determine. A potentially serious disadvantage is that incremental encoders require external counters to determine absolute angles within a given rotation. If the power is momentarily shut off, or if the encoder misses a pulse due to noise or a dirty disc, the resulting angular information will be in error.

The absolute optical encoder (right-hand diagram in Figure 1) overcomes these disadvantages but is more expensive. An absolute optical encoder's disc is divided up into N sectors (N = 5 for example shown), and each sector is further divided radially along its length into opaque and transparent sections, forming a unique N-bit digital word with a maximum count of $2^N - 1$. The digital word formed radially by each sector increments in value from one sector to the next, usually employing Gray code. Binary coding could be used, but can produce large errors if a single bit is incorrectly interpreted by the sensors. Gray code overcomes this defect: the maximum error produced by an error in any single bit of the Gray code is only 1 LSB after the Gray code is converted into binary code. A set of N light sensors responds to the N-bit digital word which corresponds to the disc's absolute
Industrial optical encoders achieve up to 16-bit resolution, with absolute accuracies that approach the resolution (20 arc seconds). Both absolute and incremental optical encoders, however, may suffer damage in harsh industrial environments.
INTRODUCTION

Machine-tool and robotics manufacturers use resolvers and synchros to provide accurate angular and rotational information. These devices excel in demanding factory and aviation applications requiring small size, long-term reliability, absolute position measurement, high accuracy, and low-noise operation.

SYNCHROS AND RESOLVERS

A diagram of a typical synchro and resolver is shown in Figure 1. Both synchros and resolvers employ single-winding rotors that revolve inside fixed stators. In the case of a simple synchro, the stator has three windings oriented 120° apart and electrically connected in a Y-connection. Resolvers differ from synchros in that their stators have only two windings oriented at 90°.

Because synchros have three stator coils in a 120° orientation, they are more difficult than resolvers to manufacture and are therefore more costly. Today, synchros find decreasing use, except in certain military and avionic retrofit applications.

Modern resolvers, in contrast, are available in a brushless form that employ a transformer to couple the rotor signals from the stator to the rotor. The primary winding of this transformer resides on the stator, and the secondary on the rotor. Other resolvers use more traditional brushes or slip rings to couple the signal into the rotor winding. Brushless resolvers are more rugged than synchros because there are no brushes to break or dislodge, and the life of a brushless resolver is limited only by its bearings. Most resolvers are specified to work over 2 V to 40 V rms and at frequencies from 400 Hz to 10 kHz. Angular accuracies range from 5 arc-minutes to 0.5 arc-minutes. (There are 60 arc-minutes in one degree, and 60 arc-seconds in one arc-minute. Hence, one arc-minute is equal to 0.0167 degrees).
In operation, synchros and resolvers resemble rotating transformers. The rotor winding is excited by an ac reference voltage, at frequencies up to a few kHz. The magnitude of the voltage induced in any stator winding is proportional to the sine of the angle, \( \Theta \), between the rotor coil axis and the stator coil axis. In the case of a synchro, the voltage induced across any pair of stator terminals will be the vector sum of the voltages across the two connected coils.

For example, if the rotor of a synchro is excited with a reference voltage, \( V \sin \omega t \), across its terminals R1 and R2, then the stator's terminal will see voltages in the form:

\[
\begin{align*}
S1 \text{ to } S3 &= V \sin \omega t \sin \Theta \\
S3 \text{ to } S2 &= V \sin \omega t \sin (\Theta + 120^\circ) \\
S2 \text{ to } S1 &= V \sin \omega t \sin (\Theta + 240^\circ),
\end{align*}
\]

where \( \Theta \) is the shaft angle.

In the case of a resolver, with a rotor ac reference voltage of \( V \sin \omega t \), the stator's terminal voltages will be:

\[
\begin{align*}
S1 \text{ to } S3 &= V \sin \omega t \sin \Theta \\
S4 \text{ to } S2 &= V \sin \omega t \sin (\Theta + 90^\circ) = V \sin \omega t \cos \Theta.
\end{align*}
\]

It should be noted that the 3-wire synchro output can be easily converted into the resolver-equivalent format using a Scott-T transformer. Therefore, the following signal processing example describes only the resolver configuration.

**RESOLVER-TO-DIGITAL CONVERTERS (RDCs)**

A typical resolver-to-digital converter (RDC) is shown functionally in Figure 2. The two outputs of the resolver are applied to cosine and sine multipliers. These multipliers incorporate sine and cosine lookup tables and function as multiplying digital-to-analog converters. Begin by assuming that the current state of the up/down counter is a digital number representing a trial angle, \( \phi \). The converter seeks to adjust the digital angle, \( \phi \), continuously to become equal to, and to track \( \Theta \), the analog angle being measured.
The resolver's stator output voltages are written as:

\[ V_1 = V \sin \omega t \sin \theta \]  \hspace{1cm} \text{Eq. 6} \\
\[ V_2 = V \sin \omega t \cos \theta \]  \hspace{1cm} \text{Eq. 7}

where \( \Theta \) is the angle of the resolver's rotor. The digital angle \( \varphi \) is applied to the cosine multiplier, and its cosine is multiplied by \( V_1 \) to produce the term:

\[ V \sin \omega t \sin \theta \cos \varphi \]  \hspace{1cm} \text{Eq. 8}

The digital angle \( \varphi \) is also applied to the sine multiplier and multiplied by \( V_2 \) to produce the term:

\[ V \sin \omega t \cos \theta \sin \varphi \]  \hspace{1cm} \text{Eq. 9}

These two signals are subtracted from each other by the error amplifier to yield an ac error signal of the form:

\[ V \sin \omega t [\sin \theta \cos \varphi - \cos \theta \sin \varphi] \]  \hspace{1cm} \text{Eq. 10}

Using a simple trigonometric identity, this reduces to:

\[ V \sin \omega t [\sin (\theta - \varphi)] \]  \hspace{1cm} \text{Eq. 11}

The detector synchronously demodulates this ac error signal, using the resolver's rotor voltage as a reference. This results in a dc error signal proportional to \( \sin(\Theta - \varphi) \).

The dc error signal feeds an integrator, the output of which drives a voltage-controlled-oscillator (VCO). The VCO, in turn, causes the up/down counter to count in the proper direction to cause:
When this is achieved, and therefore to within one count. Hence, the counter's digital output, $\varphi$, represents the angle $\Theta$. The latches enable this data to be transferred externally without interrupting the loop's tracking.

This circuit is equivalent to a so-called type-2 servo loop, because it has, in effect, two integrators. One is the counter, which accumulates pulses; the other is the integrator at the output of the detector. In a type-2 servo loop with a constant rotational velocity input, the output digital word continuously follows, or tracks the input, without needing externally derived convert commands, and with no steady state phase lag between the digital output word and actual shaft angle. An error signal appears only during periods of acceleration or deceleration.

As an added bonus, the tracking RDC provides an analog dc output voltage directly proportional to the shaft's rotational velocity. This is a useful feature if velocity is to be measured or used as a stabilization term in a servo system, and it makes additional tachometers unnecessary.

Since the operation of an RDC depends only on the ratio between input signal amplitudes, attenuation in the lines connecting them to resolvers doesn't substantially affect performance. For similar reasons, these converters are not greatly susceptible to waveform distortion. In fact, they can operate with as much as 10% harmonic distortion on the input signals; some applications actually use square-wave references with little additional error.

Tracking ADCs are therefore ideally suited to RDCs. While other ADC architectures, such as successive approximation, could be used, the tracking converter is the most accurate and efficient for this application.

Because the tracking converter doubly integrates its error signal, the device offers a high degree of noise immunity (12-dB-per-octave rolloff). The net area under any given noise spike produces an error. However, typical inductively coupled noise spikes have equal positive and negative going waveforms. When integrated, this results in a zero net error signal. The resulting noise immunity, combined with the converter's insensitivity to voltage drops, lets the user locate the converter at a considerable distance from the resolver. Noise rejection is further enhanced by the detector's rejection of any signal not at the reference frequency, such as wideband noise.

The AD2S90 is one of a number of integrated RDCs offered by Analog Devices. The general architecture is similar to that of Figure 2. Further details on synchro and resolver-to-digital converters can be found in References 1, 2, and 3.

**REFERENCES**


2. Dennis Fu, "Circuit Applications of the AD2S90 Resolver-to-Digital Converter," *Application Note AN-230*, Analog Devices. *(applications of the AD2S90 RTD).*

MT-031: Grounding Data Converters and Solving the Mystery of "AGND" and "DGND"

by Walt Kester, James Bryant, and Mike Byrne

INTRODUCTION

Today's signal processing systems generally require mixed-signal devices such as analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) as well as fast digital signal processors (DSPs). Requirements for processing analog signals having wide dynamic ranges increases the importance of high performance ADCs and DACs. Maintaining wide dynamic range with low noise in hostile digital environments is dependent upon using good high-speed circuit design techniques including proper signal routing, decoupling, and grounding.

In the past, "high precision, low-speed" circuits have generally been viewed differently than so-called "high-speed" circuits. With respect to ADCs and DACs, the sampling (or update) frequency has generally been used as the distinguishing speed criteria. However, the following two examples show that in practice, most of today's signal processing ICs are really "high-speed," and must therefore be treated as such in order to maintain high performance. This is certainly true of DSPs, and also true of ADCs and DACs.

All sampling ADCs (ADCs with an internal sample-and-hold circuit) suitable for signal processing applications operate with relatively high speed clocks with fast rise and fall times (generally a few nanoseconds) and must be treated as high speed devices, even though throughput rates may appear low. For example, a medium-speed 12-bit successive approximation (SAR) ADC may operate on a 10-MHz internal clock, while the sampling rate is only 500 kSPS.

Sigma-delta (Σ-Δ) ADCs also require high speed clocks because of their high oversampling ratios. Even high resolution, so-called "low frequency" Σ-Δ industrial measurement ADCs (having throughputs of 10 Hz to 7.5 kHz) operate on 5-MHz or higher clocks and offer resolution to 24-bits (for example, the Analog Devices AD77xx-series).

To further complicate the issue, mixed-signal ICs have both analog and digital ports, and because of this, much confusion has resulted with respect to proper grounding techniques. In addition, some mixed-signal ICs have relatively low digital currents, while others have high digital currents. In many cases, these two types must be treated differently with respect to optimum grounding.

Digital and analog design engineers tend to view mixed-signal devices from different perspectives, and the purpose of this tutorial is to develop a general grounding philosophy that will work for most mixed signal devices, without having to know the specific details of their internal circuits.

GROUND AND POWER PLANES

The importance of maintaining a low impedance large area ground plane is critical to all analog and digital circuits today. The ground plane not only acts as a low impedance return path for decoupling high frequency currents (caused by fast digital logic) but also minimizes EMI/RFI emissions. Because of the shielding action of the ground plane, the circuit's susceptibility to external EMI/RFI is also reduced.

Ground planes also allow the transmission of high speed digital or analog signals using transmission line techniques (microstrip or stripline) where controlled impedances are required.

The use of "buss wire" is totally unacceptable as a "ground" because of its impedance at the equivalent frequency of most logic transitions. For instance, #22 gauge wire has about 20 nH/inch inductance. A transient current having a slew rate of 10 mA/ns created by a logic signal would develop an unwanted voltage drop of 200 mV at this frequency flowing through 1 inch of this wire:
For a signal having a 2-V peak-to-peak range, this translates into an error of about 10% (approximately 3.5-bit accuracy). Even in all-digital circuits, this error would result in considerable degradation of logic noise margins.

Figure 1 shows the classic illustration of a situation where the digital return current modulates the analog return current (top figure). The ground return wire inductance and resistance is shared between the analog and digital circuits, and this is what causes the interaction and resulting error. A possible solution is to make the digital return current path flow directly to the GND REF as shown in the bottom figure. This is the fundamental concept of a "star," or single-point ground system. Implementing the true single-point ground in a system which contains multiple high frequency return paths is difficult because the physical length of the individual return current wires will introduce parasitic resistance and inductance which can make obtaining a low impedance high frequency ground difficult. In practice, the current returns must consist of large area ground planes for low impedance to high frequency currents. Without a low-impedance ground plane, it is therefore almost impossible to avoid these shared impedances, especially at high frequencies.

![Figure 1: Digital Currents Flowing in Analog Return Path Create Error Voltages](image)

All integrated circuit ground pins should be soldered directly to the low-impedance ground plane to minimize series inductance and resistance. The use of traditional IC sockets is not recommended with high-speed devices. The extra inductance and capacitance of even "low profile" sockets may corrupt the device performance by introducing unwanted shared paths. If sockets must be used with DIP packages, as in prototyping, individual "pin sockets" or "cage jacks" may be acceptable. Both capped and uncapped versions of these pin sockets are available (AMP part numbers 5-330808-3, and 5-330808-6). They have spring-loaded gold contacts which make good electrical and mechanical connection to the IC pins. Multiple insertions, however, may degrade their performance.

LOW AND HIGH FREQUENCY DECOUPLING
Each power supply should be decoupled to the low-impedance ground plane with a high quality electrolytic capacitor at the point it enters the PC board. This minimizes low frequency noise on the supply runs. At each individual analog stage, further local, high-frequency-only filtering is required at the individual IC package power pins.

Figure 2 shows this technique, in both correct (left) as well as incorrect example implementations (right). In the left example, a typical 0.1-µF chip ceramic capacitor goes directly to the opposite PCB side ground plane, by virtue of the via, and on to the IC’s GND pin by a second via. In contrast, the less desirable setup at the right adds additional PCB trace inductance in the ground path of the decoupling cap, reducing effectiveness.

![Figure 2: Localized High Frequency Supply Filter(s) Provides Optimum Filtering and Decoupling Via Short Low-Inductance Path (Ground Plane)](image)

All high frequency (i.e., \( \geq 10 \) MHz) ICs should use a bypassing scheme similar to Figure 2 for best performance. The ferrite beads aren’t 100% necessary, but they will add extra high frequency noise isolation and decoupling, which is often desirable. Possible caveats here would be to verify that the beads never saturate, when the ICs are handling high currents. Note that with some ferrites, even before full saturation occurs, some beads can be non-linear, so if a power stage is required to operate with a low distortion output, this should also be checked.

DOUBLE-SIDED VS. MULTILAYER PRINTED CIRCUIT BOARDS

Each PCB in the system should have at least one complete layer dedicated to the ground plane. Ideally, a double-sided board should have one side completely dedicated to ground and the other side for interconnections. In practice, this is not possible, since some of the ground plane will certainly have to be removed to allow for signal and power crossovers, vias, and through-holes. Nevertheless, as much area as possible should be preserved, and at least 75% should remain. After completing an initial layout, the ground layer should be checked carefully to make sure there are no isolated ground “islands,” because IC ground pins located in a ground “island” have no current return path to the ground plane. Also, the ground plane should be checked for “skinny” connections between adjacent large areas which may significantly reduce the effectiveness of the ground plane. Needless to say, auto-routing board layout techniques will generally lead to a layout disaster on a mixed-signal board, so manual intervention is highly recommended.

Systems that are densely packed with surface mount ICs will have a large number of interconnections; therefore multilayer boards are mandatory. This allows at least one complete layer to be dedicated to ground. A simple 4-layer board would have internal ground and power plane layers with the outer two layers used for interconnections between the surface mount components. Placing the power and ground planes adjacent to each other provides additional inter-plane capacitance which helps high frequency decoupling of the power supply. In most systems, 4-layers are not enough, and additional layers are
MULTICARD MIXED-SIGNALS SYSTEMS

The best way of minimizing ground impedance in a multicard system is to use a "motherboard" PCB as a backplane for interconnections between cards, thus providing a continuous ground plane to the backplane. The PCB connector should have at least 30-40% of its pins devoted to ground, and these pins should be connected to the ground plane on the backplane motherboard. To complete the overall system grounding scheme there are two possibilities:

1. The backplane ground plane can be connected to chassis ground at numerous points, thereby diffusing the various ground current return paths. This is commonly referred to as a "multipoint" grounding system and is shown in Figure 3.

2. The ground plane can be connected to a single system "star ground" point (generally at the power supply).

The first approach is most often used in all-digital systems, but can be used in mixed-signal systems provided the ground currents due to digital circuits are sufficiently low and diffused over a large area. The low ground impedance is maintained all the way through the PC boards, the backplane, and ultimately the chassis. However, it is critical that good electrical contact be made where the grounds are connected to the sheet metal chassis. This requires self-tapping sheet metal screws or "biting" washers. Special care must be taken where anodized aluminum is used for the chassis material, since its surface acts as an insulator.

The second approach ("star ground") is often used in high speed mixed-signal systems having separate analog and digital ground systems and warrants further discussion.

SEPARATING ANALOG AND DIGITAL GROUND PLANES

In mixed-signal systems with large amounts of digital circuitry, it is highly desirable to physically separate sensitive analog components from noisy digital components. It may also be beneficial to use separate ground planes for the analog and the digital circuitry. These planes should not overlap in order to minimize capacitive coupling between the two. The separate analog and
Digital ground planes are continued on the backplane using either motherboard ground planes or "ground screens" which are made up of a series of wired interconnections between the connector ground pins. The arrangement shown in Figure 4 illustrates that the two planes are kept separate all the way back to a common system "star" ground, generally located at the power supplies. The connections between the ground planes, the power supplies, and the "star" should be made up of multiple bus bars or wide copper braids for minimum resistance and inductance. The back-to-back Schottky diodes on each PCB are inserted to prevent accidental dc voltage from developing between the two ground systems when cards are plugged and unplugged. This voltage should be kept less than 300 mV to prevent damage to ICs which have connections to both the analog and digital ground planes. Schottky diodes are preferable because of their low capacitance and low forward voltage drop. The low capacitance prevents ac coupling between the analog and digital ground planes. Schottky diodes begin to conduct at about 300 mV, and several parallel diodes in parallel may be required if high currents are expected. In some cases, ferrite beads can be used instead of Schottky diodes, however they introduce dc ground loops which can be troublesome in precision systems.

![Figure 4: Separating Analog and Digital Ground Planes](image)

It is mandatory that the impedance of the ground planes be kept as low as possible, all the way back to the system star ground. DC or ac voltages of more than 300 mV between the two ground planes can not only damage ICs but cause false triggering of logic gates and possible latchup.

**GROUNDING AND DECOUPLING MIXED-SIGNAL ICs WITH LOW DIGITAL CURRENTS**

Sensitive analog components such as amplifiers and voltage references are always referenced and decoupled to the analog ground plane. The ADCs and DACs (and other mixed-signal ICs) with low digital currents should generally be treated as analog components and also grounded and decoupled to the analog ground plane. At first glance, this may seem somewhat contradictory, since a converter has an analog and digital interface and usually has pins designated as analog ground (AGND) and digital ground (DGND). The diagram shown in Figure 5 will help to explain this seeming dilemma.
Inside an IC that has both analog and digital circuits, such as an ADC or a DAC, the grounds are usually kept separate to avoid coupling digital signals into the analog circuits. Figure 5 shows a simple model of a converter. There is nothing the IC designer can do about the wirebond inductance and resistance associated with connecting the bond pads on the chip to the package pins except to realize it's there. The rapidly changing digital currents produce a voltage at point B which will inevitably couple into point A of the analog circuits through the stray capacitance, \( C_{\text{STRAY}} \). In addition, there is approximately 0.2-pF unavoidable stray capacitance between every pin of the IC package! It's the IC designer's job to make the chip work in spite of this. However, in order to prevent further coupling, the AGND and DGND pins should be joined together externally to the analog ground plane with minimum lead lengths. Any extra impedance in the DGND connection will cause more digital noise to be developed at point B; it will, in turn, couple more digital noise into the analog circuit through the stray capacitance. Note that connecting DGND to the digital ground plane applies \( V_{\text{NOISE}} \) across the AGND and DGND pins and invites disaster!

The name "DGND" on an IC tells us that this pin connects to the digital ground of the IC. This does not imply that this pin must be connected to the digital ground of the system.

It is true that this arrangement may inject a small amount of digital noise onto the analog ground plane. These currents should be quite small, and can be minimized by ensuring that the converter output does not drive a large fanout (they normally can't, by design). Minimizing the fanout on the converter's digital port will also keep the converter logic transitions relatively free from ringing and minimize digital switching currents, and thereby reducing any potential coupling into the analog port of the converter. The logic supply pin \( V_{\text{DD}} \) can be further isolated from the analog supply by the insertion of a small lossy ferrite bead as shown in Figure 5. The internal transient digital currents of the converter will flow in the small loop from \( V_{\text{DD}} \) through the decoupling capacitor and to DGND (this path is shown with a heavy line on the diagram). The transient digital currents will therefore not appear on the external analog ground plane, but are confined to the loop. The \( V_{\text{DD}} \) pin decoupling capacitor should be mounted as close to the converter as possible to minimize parasitic inductance. These decoupling capacitors should be low inductance ceramic types, typically between 0.01 \( \mu \text{F} \) and 0.1 \( \mu \text{F} \).

**TREAT THE ADC DIGITAL OUTPUTS WITH CARE**

It is always a good idea (as shown in Figure 5) to place a buffer register adjacent to the converter to isolate the converter's digital lines from noise on the data bus. The register also serves to minimize loading on the digital outputs of the converter and acts as a
Faraday shield between the digital outputs and the data bus. Even though many converters have three-state outputs/inputs, this isolation register still represents good design practice. In some cases it may be desirable to add an additional buffer register on the analog ground plane next to the converter output to provide greater isolation.

The series resistors (labeled "R" in Figure 5) between the ADC output and the buffer register input help to minimize the digital transient currents which may affect converter performance. The resistors isolate the digital output drivers from the capacitance of the buffer register inputs. In addition, the RC network formed by the series resistor and the buffer register input capacitance acts as a lowpass filter to slow down the fast edges.

A typical CMOS gate combined with PCB trace and a through-hole will create a load of approximately 10 pF. A logic output slew rate of 1 V/ns will produce 10 mA of dynamic current if there is no isolation resistor:

\[
\Delta I = C \frac{\Delta V}{\Delta t} = 10 \text{ pF} \times \frac{1 \text{ V}}{\text{ns}} = 10 \text{ mA}.
\]

A 500-Ω series resistors will minimize this output current and result in a rise and fall time of approximately 11 ns when driving the 10-pF input capacitance of the register:

\[
\tau_r = 2.2 \times C = 2.2 \times 500 \text{ } \Omega \times 10 \text{ pF} = 11 \text{ ns}.
\]

TTL registers should be avoided, since they can appreciably add to the dynamic switching currents because of their higher input capacitance.

The buffer register and other digital circuits should be grounded and decoupled to the digital ground plane of the PC board. Notice that any noise between the analog and digital ground plane reduces the noise margin at the converter digital interface. Since digital noise immunity is of the orders of hundreds or thousands of millivolts, this is unlikely to matter. The analog ground plane will generally not be very noisy, but if the noise on the digital ground plane (relative to the analog ground plane) exceeds a few hundred millivolts, then steps should be taken to reduce the digital ground plane impedance, thereby maintaining the digital noise margins at an acceptable level. Under no circumstances should the voltage between the two ground planes exceed 300 mV, or the ICs may be damaged.

Separate power supplies for analog and digital circuits are also highly desirable, even if the voltages are the same. The analog supply should be used to power the converter. If the converter has a pin designated as a digital supply pin (VDD), it should either be powered from a separate analog supply, or filtered as shown in the diagram. All converter power pins should be decoupled to the analog ground plane, and all logic circuit power pins should be decoupled to the digital ground plane as shown in Figure 6.
In some cases it may not be possible to connect $V_D$ to the analog supply. Some of the newer, high speed ICs may have their analog circuits powered by $+5\,\text{V}$, but the digital interface powered by $+3\,\text{V}$ to interface to 3 V logic. In this case, the $+3\,\text{V}$ pin of the IC should be decoupled directly to the analog ground plane. It is also advisable to connect a ferrite bead in series with the power trace that connects the pin to the $+3\,\text{V}$ digital logic supply.

The sampling clock generation circuitry should be treated like analog circuitry and also be grounded and heavily-decoupled to the analog ground plane. Phase noise on the sampling clock produces degradation in system SNR as will be discussed shortly.

**SAMPLING CLOCK CONSIDERATIONS**

In a high performance sampled data system a low phase-noise oscillator should be used to generate the ADC (or DAC) sampling clock because sampling clock jitter modulates the analog input/output signal and raises the noise and distortion floor. The sampling clock generator should be isolated from noisy digital circuits and grounded and decoupled to the analog ground plane, as is true for the op amp and the ADC.

The effect of sampling clock jitter on ADC signal-to-noise ratio (SNR) is given approximately by the equation:

$$\text{SNR} = 20\log_{10}\left[\frac{1}{2\pi f t_j}\right],$$  \hspace{1cm} \text{Eq. 4}$$

where SNR is the SNR of a perfect ADC of infinite resolution where the only source of noise is that caused by the rms sampling clock jitter, $t_j$. Note that $f$ in the above equation is the analog input frequency. Just working through a simple example, if $t_j = 50$ ps rms, $f = 100$ kHz, then SNR = 90 dB, equivalent to about 15-bit dynamic range. This effect of clock jitter on SNR is discussed in much more detail in Tutorial MT-007.

It should be noted that $t_j$ in the above example is the root-sum-square (rss) value of the external clock jitter and the internal ADC clock jitter (called aperture jitter). However, in most high performance ADCs, the internal aperture jitter is negligible compared to the jitter on the sampling clock.
Ideally, the sampling clock oscillator should be referenced to the analog ground plane in a split-ground system. However, this is not always possible because of system constraints. In many cases, the sampling clock must be derived from a higher frequency multi-purpose system clock which is generated on the digital ground plane. It must then pass from its origin on the digital ground plane to the ADC on the analog ground plane. Ground noise between the two planes adds directly to the clock signal and will produce excess jitter. The jitter can cause degradation in the signal-to-noise ratio and also produce unwanted harmonics.

This can be remedied somewhat by transmitting the sampling clock signal as a differential signal using either a small RF transformer as shown in Figure 7 or a high speed differential driver and receiver IC. Many high-speed ADCs have differential sampling clock inputs to facilitate this approach. If an active differential driver and receiver are used, they should be ECL, low-level ECL, or LVDS to minimize phase jitter. In a single +5 V supply system, ECL logic can be connected between ground and +5 V (PECL), and the outputs ac coupled into the ADC sampling clock input. In either case, the original master system clock must be generated from a low phase noise oscillator, and not the clock output of a DSP, microprocessor, or microcontroller.

In order to facilitate system clock management, a family clock generation and distribution products is available from Analog Devices as well as a complete selection of phase-locked loops (PLLs).

![Figure 7: Sampling Clock Distribution From Digital to Analog Ground Planes](image)

THE ORIGINS OF THE CONFUSION ABOUT MIXED-SIGNAL GROUNDING: APPLYING SINGLE-CARD GROUNDING CONCEPTS TO MULTICARD SYSTEMS

Most ADC, DAC, and other mixed-signal device data sheets discuss grounding relative to a single PCB, usually the manufacturer's own evaluation board. This has been a source of confusion when trying to apply these principles to multicard or multi-ADC/DAC systems. The recommendation is usually to split the PCB ground plane into an analog plane and a digital plane. It is then further recommended that the AGND and DGND pins of a converter be tied together and that the analog ground plane and digital ground planes be connected at that same point as shown in Figure 8. This essentially creates the system "star" ground at the mixed-signal device.

All noisy digital currents flow through the digital power supply to the digital ground plane and back to the digital supply; they are isolated from the sensitive analog portion of the board. The system star ground occurs where the analog and digital ground planes are joined together at the mixed signal device. While this approach will generally work in a simple system with a single
PCB and single ADC/DAC, it is not usually optimum for multicard mixed-signal systems. In systems having several ADCs or DACs on different PCBs (or on the same PCB, for that matter), the analog and digital ground planes become connected at several points, creating the possibility of ground loops and making a single-point “star” ground system impossible. For these reasons, this grounding approach is not recommended for multicard systems, and the approach previously discussed should be used for mixed signal ICs with low digital currents.

**Figure 8:**
Grounding Mixed Signal ICs : Single PC Board (Typical Evaluation/Test Board)

**SUMMARY:**
**GROUNDING MIXED SIGNAL DEVICES WITH LOW DIGITAL CURRENTS IN A MULTICARD SYSTEM**

Figure 9 summarizes the approach previously described for grounding a mixed signal device which has low digital currents. The analog ground plane is not corrupted because the small digital transient currents flow in the small loop between $V_D$, the decoupling capacitor, and DGND (shown as a heavy line). The mixed signal device is for all intents and purposes treated as an analog component. The noise $V_N$ between the ground planes reduces the noise margin at the digital interface but is generally not harmful if kept less than 300 mV by using a low impedance digital ground plane all the way back to the system star ground.

However, mixed signal devices such as sigma-delta ADCs, codecs, and DSPs with on-chip analog functions are becoming more and more digitally intensive. Along with the additional digital circuitry come larger digital currents and noise. For example, a sigma-delta ADC or DAC contains a complex digital filter which adds considerably to the digital current in the device. The method previously discussed depends on the decoupling capacitor between $V_D$ and DGND to keep the digital transient currents isolated in a small loop. However, if the digital currents are significant enough and have components at dc or low frequencies, the decoupling capacitor may have to be so large that it is impractical. Any digital current which flows outside the loop between $V_D$ and DGND must flow through the analog ground plane. This may degrade performance, especially in high resolution systems.
It is difficult to predict what level of digital current flowing into the analog ground plane will become unacceptable in a system. All we can do at this point is to suggest an alternative grounding method which may yield better performance.

**SUMMARY:**

GROUNDING MIXED SIGNAL DEVICES WITH HIGH DIGITAL CURRENTS IN A MULTICARD SYSTEM
(USE THIS METHOD WITH CAUTION!)

An alternative grounding method for a mixed signal device with high levels of digital currents is shown in Figure 10. The AGND of the mixed signal device is connected to the analog ground plane, and the DGND of the device is connected to the digital ground plane. The digital currents are isolated from the analog ground plane, but the noise between the two ground planes is applied directly between the AGND and DGND pins of the device. For this method to be successful, the analog and digital circuits within the mixed signal device must be well isolated. The noise between AGND and DGND pins must not be large enough to reduce internal noise margins or cause corruption of the internal analog circuits.

Figure 10 shows optional Schottky diodes (back-to-back) or a ferrite bead connecting the analog and digital ground planes. The Schottky diodes prevent large dc voltages or low frequency voltage spikes from developing across the two planes. These voltages can potentially damage the mixed signal IC if they exceed 300 mV because they appear directly between the AGND and DGND pins. As an alternative to the back-to-back Schottky diodes, a ferrite bead provides a dc connection between the two planes but isolates them at frequencies above a few MHz where the ferrite bead becomes resistive. This protects the IC from dc voltages between AGND and DGND, but the dc connection provided by the ferrite bead can introduce unwanted dc ground loops and may not be suitable for high resolution systems.
Whenever AGND and DGND pins are separated in the special case of ICs with high digital currents, provisions should be made to connect them together if necessary. Jumpers and/or strap options allow both methods to be tried to verify which gives the best overall performance in the system.

GROUNDING SUMMARY

There is no single grounding method which will guarantee optimum performance 100% of the time! This section has presented a number of possible options depending upon the characteristics of the particular mixed signal devices in question. It is helpful, however, to provide for as many options as possible when laying out the initial PC board.

It is mandatory that at least one layer of the PC board be dedicated to ground plane! The initial board layout should provide for non-overlapping analog and digital ground planes, but pads and vias should be provided at several locations for the installation of back-to-back Schottky diodes or ferrite beads, if required. It is also extremely important that pads and vias be provided so that the analog and digital ground planes can be connected together with jumpers if required. It is difficult to predict whether the "multi-point" (single ground plane) or the "star" ground (separate analog and digital ground planes) method will give best overall system performance; therefore, some experimentation with the final PC board using the jumpers may be required.

When in doubt, it is always better to start out with a split analog and digital ground plane and later connect them with jumpers, rather than to start out with a single ground plane and try and later try and split it!

SOME GENERAL PC BOARD LAYOUT GUIDELINES FOR MIXED-SIGNAL SYSTEMS

It is evident that noise can be minimized by paying attention to the system layout and preventing different signals from interfering with each other. High level analog signals should be separated from low level analog signals, and both should be kept away from digital signals. We have seen elsewhere that in waveform sampling and reconstruction systems the sampling clock (which is a digital signal) is as vulnerable to noise as any analog signal, but is as liable to cause noise as any digital signal, and so must be kept isolated from both analog and digital systems. If clock driver packages are used in clock distribution, only one frequency clock should be passed through a single package. Sharing drivers between clocks of different frequencies in the same package will produce excess jitter and crosstalk and degrade performance.
The ground plane can act as a shield where sensitive signals cross. Figure 11 shows a good layout for a data acquisition board where all sensitive areas are isolated from each other and signal paths are kept as short as possible. While real life is rarely as tidy as this, the principle remains a valid one.

There are a number of important points to be considered when making signal and power connections. First of all a connector is one of the few places in the system where all signal conductors must run in parallel – it is therefore imperative to separate them with ground pins (creating a faraday shield) to reduce coupling between them.

Multiple ground pins are important for another reason: they keep down the ground impedance at the junction between the board and the backplane. The contact resistance of a single pin of a PCB connector is quite low (of the order of 10 mΩ) when the board is new—as the board gets older the contact resistance is likely to rise, and the board's performance may be compromised. It is therefore well worthwhile to allocate extra PCB connector pins so that there are many ground connections (perhaps 30-40% of all the pins on the PCB connector should be ground pins). For similar reasons there should be several pins for each power connection, although there is no need to have as many as there are ground pins.

Analog Devices and other manufacturers of high performance mixed-signal ICs offer evaluation boards to assist customers in their initial evaluations and layout. ADC evaluation boards generally contain an on-board low-jitter sampling clock oscillator, output registers, and appropriate power and signal connectors. They also may have additional support circuitry such as the ADC input buffer amplifier and external reference.

The layout of the evaluation board is optimized in terms of grounding, decoupling, and signal routing and can be used as a model when laying out the ADC PC board in the system. The actual evaluation board layout is usually available from the ADC manufacturer in the form of computer CAD files (Gerber files). In many cases, the layout of the various layers appears on the data sheet for the device.

REFERENCES


