Chapter

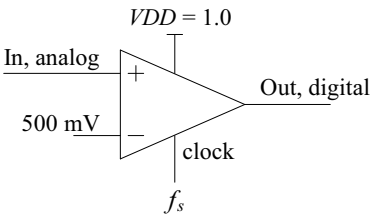
5

Data Converter SNR

After studying Chs. 1-4 we should understand the sampling process, including analog sampling, decimation, and interpolation, the operation of the ideal ADC and DAC, and the basics of filtering. In this chapter we turn our attention towards quantization noise. Quantization noise is the effective noise added to a signal after it passes through an ADC (aka quantizer), a comparator (a 1-bit ADC or quantizer)), or, for digital signals, a circuit that reduces the word size (removes the LSBs of the word). One of the key things we'll focus on is the shape of the quantization noise spectrum and how it's added to the spectrum of an input analog or digital signal.

5.1 Quantization Noise

Examine the clocked comparator seen in Fig. 5.1. In this chapter, like the rest of the book, we'll use a *VDD* of 1 V and a common-mode voltage, V_{CM}, of 500 mV. When the input to the comparator, the non-inverting input, is 600 mV then, since the inverting input is held at V_{CM}, the output goes to 1 V on the rising edge of the clock signal. The difference between the input and output of the comparator is 400 mV. Note that we've gone from an analog signal, the input, to a digital signal, the output. However, while doing this conversion we added noise to our input signal. It's useful to think of this analog-to-digital (quantization) process, from a block diagram point of view, as a S/H followed by an adder that introduces the quantization noise, Fig. 5.2. The S/H response doesn't come into



In	Out	Added noise
600 mV	1 V	+400 mV
800 mV	1 V	+200 mV
200 mV	0 V	-200 mV
450 mV	0 V	-450 mV
950 mV	1 V	+50 mV

Figure 5.1 How a comparator adds noise to an input signal.

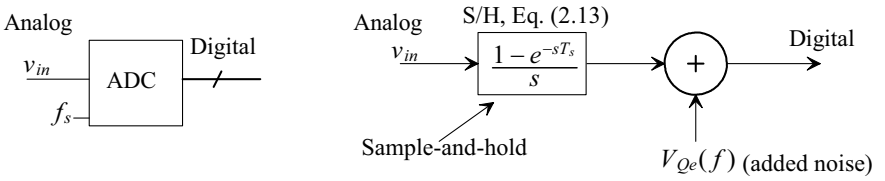


Figure 5.2 Modeling the ADC.

play in switched-capacitor (discrete-time) circuits because the input signal is already sampled (so we don't include it when modeling the ADC), but it does introduce a Sinc response when the ADC input is continuous-time, Sec. 2.1.3. Also notice that we've assumed that the noise we added to our input signal has a spectrum of $V_{Qe}(f)$. Determining the shape and range of this noise spectrum is one of the goals of this chapter. Finally, while a good portion of our studies will focus on the quantization noise introduced during the analog-to-digital process we can also apply the same results when truncating a digital word's size, Fig. 5.3.

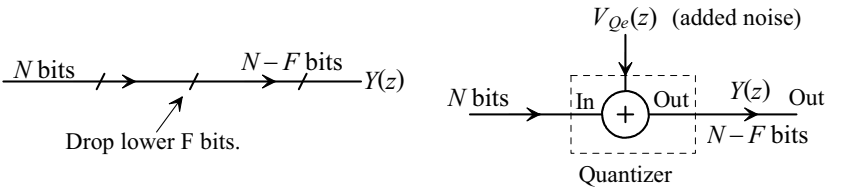


Figure 5.3 Quantizing a digital word and modeling the added noise.

5.1.1 Viewing the Quantization Noise Spectrum Using Simulations

Consider the simple connection of an ideal 8-bit ADC to an ideal 8-bit DAC seen in Fig. 5.4. If we apply a 7 MHz sinewave to the ADC with an amplitude of 0.4 V and an offset of 0.5 V (so the sinewave swings from 100 mV to 900 mV]) and clock the ADC at 100 MHz we get the signals seen in Fig. 5.5. Note that the output of the DAC looks very similar to the output of an ideal S/H (see Fig. 2.14). Now, however, the amplitude of the DAC output signal is quantized, that is, within 1 LSB (= 1.0/256 or 3.906 mV, see Eq. [4.1], for the present simulation) of the ADC input. This quantization is not obvious after looking at Fig. 5.5 (the time domain response). However, looking at the spectrums of the

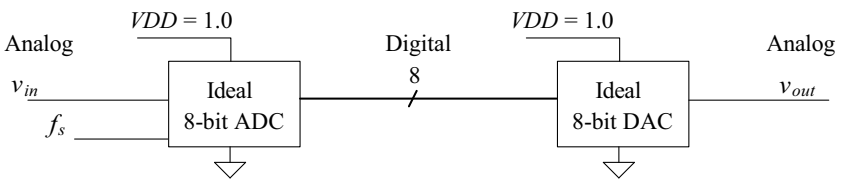
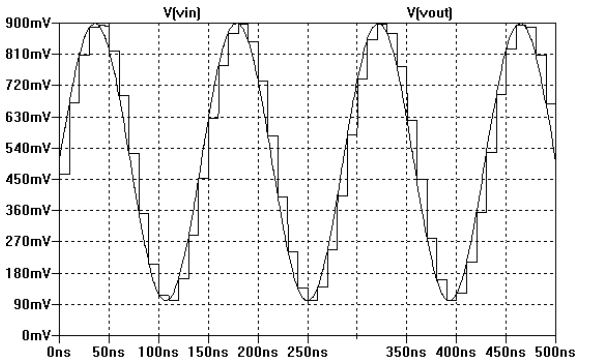


Figure 5.4 Passing a signal through an ADC and then through a DAC.



$$f_{clk} = f_s = 100 \text{ MHz}$$

Figure 5.5 Seven MHz ADC input and the corresponding DAC output.

ADC input and the DAC output reveals the difference in the noise floor between the two, Fig. 5.6. The inherent noise floor in the simulation that is associated with the input signal is approximately −80 dB (0.1 mV, RMS.) The noise floor associated with the DAC's output (the signal + quantization noise) is approximately −70 dB (0.316 mV, RMS). It is desirable to determine what sets this value and its spectral content. Again, note that the ADC quantizes the signal which results in the quantization noise (an ideal S/H and DAC don't introduce quantization noise).

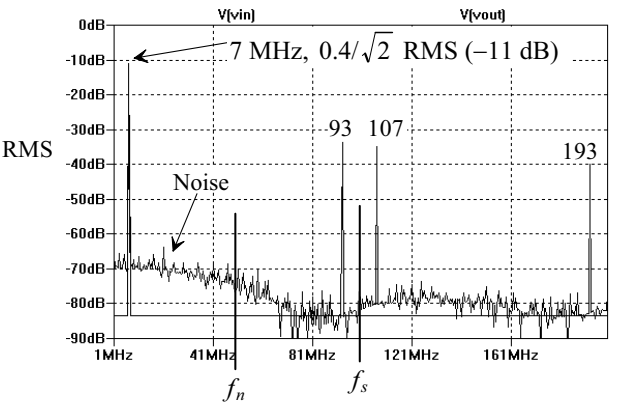


Figure 5.6 ADC input and output spectrums for Fig. 5.4 with signals seen in Fig. 5.5.

Bennett's Criteria

In order to characterize the spectral characteristics of the quantization noise let's make the following assumptions (Bennett's criteria) concerning the signal we are converting:

1. The input (to the ADC) signal's amplitude variation falls between V_{REF+} and V_{REF-} so that no saturation of the digital output code occurs. Exceeding the normal operating range of the ADC affects the quantization noise spectrum by adding spurs or spikes to the output spectrum.

2. The ADC's LSB is much smaller than the input signal amplitude. When this isn't the case, the output of the ADC can appear squarewave-like (when converted back into an analog waveform) and result in a spectrum, once again, that contains spikes or spurs. We'll see later in the book that adding or subtracting a fed-back signal (from the output based on the expected or past quantization noise) to the input modifies this requirement.

3. The input signal is busy (not DC or a low frequency input). We define busy, for the moment, as meaning that no two consecutive outputs of the ADC have the same digital code. For the ideal ADC in Fig. 5.4 1 LSB = 3.906 mV and T_s = 10 ns so that the input must change at least 3.906 mV every 10 ns. We'll see that adding a *high-frequency* dither or pseudorandom noise signal to the input, which can be filtered out later (either using a digital filter or when we pass the output through the reconstruction filter), can make the requirement on the input of being busy practical in an actual circuit. We use these assumptions (Bennett's criteria) in the following discussion unless otherwise indicated.

An Important Note

It's important to note that simply sampling an input waveform, using a S/H, does not result in quantization noise. The amplitude into the ideal S/H, at the sampling instant, is exactly the same as the amplitude out of the ideal S/H. In order to understand why this is important, consider the test setup shown in Fig. 5.7. If we apply a 3 MHz sinewave centered around the common-mode voltage with a 400 mV amplitude we get the outputs seen in Fig. 5.8. Clearly there *is* a difference between the S/H's input and its output. However, this difference has nothing to do with noise, an unwanted signal, since passing the output of the S/H, v_{outsh}, through the ideal reconstruction filter of Fig. 2.19 results in an exact replica of the S/H input v_{in}.

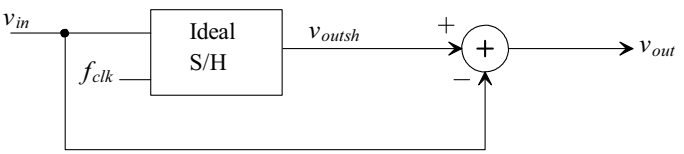


Figure 5.7 Taking the difference between the S/H input and output.

RMS Quantization Noise Voltage

If we were to set up a test configuration similar to that shown in Fig. 5.7 (see Fig. 5.9), where the input to the ADC is subtracted from the DAC output, the resulting output waveform would have little to do, in every case, with the quantization noise. This is true when the input to the ADC contains a broad frequency spectrum extending from DC to the Nyquist frequency, $f_n = f_s/2$. However, if we apply a slow linear ramp to this test setup (to limit the input frequency spectrum) we can (1) see the resulting quantization noise over a wide frequency spectrum and (2) observe the transfer curve, in the time domain. Note that this input violates Bennett's criteria (which, as we'll see, means the noise power spectral density is flat from DC to the Nyquist frequency).

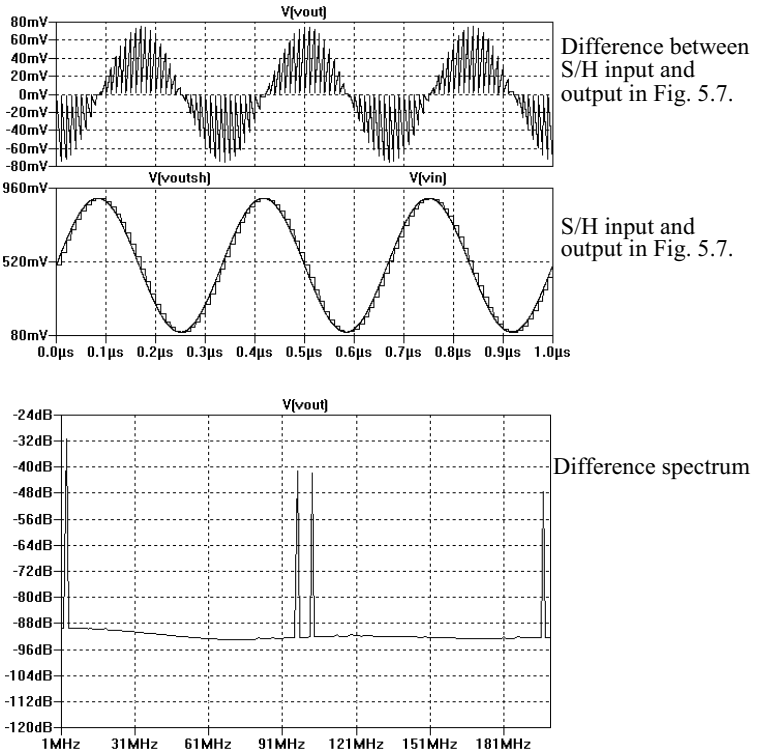


Figure 5.8 Time-domain difference between S/H input and output along with the spectrum.

A section of the input and output, using the test setup of Fig. 5.9, is shown in Fig. 5.10a. It's important to understand the input/output relationship between the ideal ADC and DAC shown in this figure. (Note that clocking the ADC too slow or putting in a ramp that rises too quickly will distort this waveform.) As an example, when the ADC input is slightly above 482 mV, in this figure, the ADC output code (input to the DAC) changes. The ADC output code can be calculated as 482 mV/1 LSB (1 LSB = 1/256 = 3.906 mV for the present simulation) or changing from 123 to 124. Looking at the transfer curves in this figure it appears as though the output changes when the ADC code is 123 or 480.4

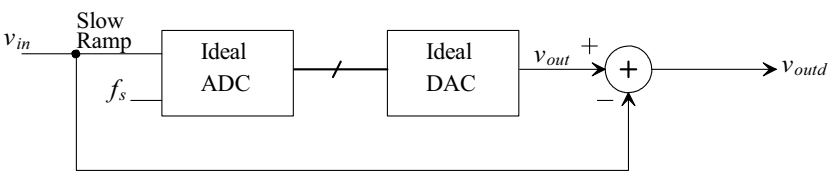


Figure 5.9 Taking the difference between an ADC input and the DAC output.

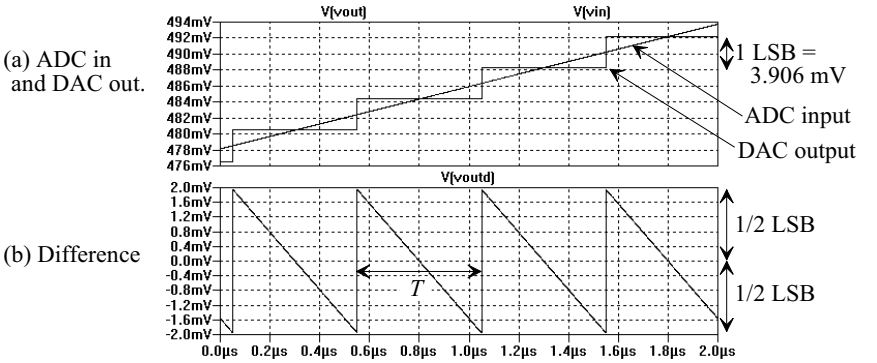


Figure 5.10 Difference between an ADC input and DAC output.

mV/1 LSB. This, as seen in Fig. 5.10b and discussed below, results in centering the quantization error around the input. This is the reason we shifted the ADC transfer curves by 1/2 LSB when we developed our ideal ADC model.

The difference output, between the two signals of Fig. 5.10a, is shown in Fig. 5.10b. Some points to note about this sawtooth waveform are that 1) its average value is zero, 2) the waveform contains an abrupt transition (and so we expect a wideband output spectrum similar to that which occurs after sampling a waveform), and 3) its peak-to-peak amplitude is 1 LSB. Like a sinewave, which also has zero average value, we can characterize this quantization error waveform by looking at its root-mean-square (RMS) value. This value can be calculated using

$$V_{Qe,RMS} = \sqrt{\frac{1}{T}\int_0^T (0.5\ \text{LSB} - \frac{1\ \text{LSB}}{T}\cdot t)^2 dt} = \frac{1\ \text{LSB}}{\sqrt{12}} = \frac{V_{LSB}}{\sqrt{12}} \quad (5.1)$$

This value is the RMS quantization noise voltage for a specific data converter. Note that the value of the period for this sawtooth waveform, T, doesn't appear in the evaluated result of this equation. Also note that the sampling frequency, f_s, isn't present in this equation. For our present discussion where 1 LSB is 3.906 mV, $V_{Qe,RMS}$ = 1.13 mV or −59 dB (RMS).

Treating Quantization Noise as a Random Variable

If Bennett's criteria hold, then the quantization noise voltage can be thought of as a random variable falling in the range of ±0.5 LSB, as seen in Fig. 5.11. The probability that the quantization error is −0.2 LSB is the same as the probability that the error is 0.4 LSB. In other words, there is no reason why the quantization error should have one value more often than another value.

The quantization error noise power is the variance of the probability density function. The RMS quantization error voltage is the square root of the quantization noise power. The variance of the probability density function (the quantization noise power, P_{Qe}) is given, knowing the average of the quantization error, $\overline{Qe}$, is zero, by

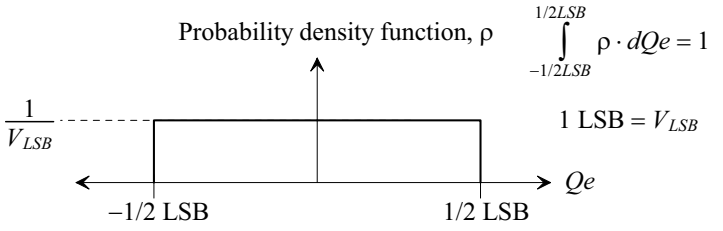


Figure 5.11 Probability density function for the quantization error in an ADC assuming Bennett's criteria hold.

$$P_{Qe} = \int_{-1/2LSB}^{1/2LSB} \rho \cdot (Qe)^2 \cdot dQe = \frac{V_{LSB}^2}{12} \tag{5.2}$$

so that, once again, the RMS quantization noise voltage is

$$V_{Qe,RMS} = \frac{V_{LSB}}{\sqrt{12}} \tag{5.3}$$

Again, if our LSB voltage is 3.096 mV, then, once again, $V_{Qe,RMS}$ = 1.13 mV or −59 dB (RMS). If we look at Fig. 5.6, we see that the RMS noise voltage varies essentially over the entire spectrum (white noise) and has a value ranging from around −70 dB down to less than −80 dB. Note that although the entire spectrum contains quantization noise it is not because of the sampling process used in the ADC (and so quantization noise doesn't experience aliasing). Quantization noise is added to the signal after the sampling process during the analog-to-digital conversion process. In order to qualitatively understand why the quantization error spectrum is white, in Fig. 5.6, we remember that there are abrupt transitions in the DAC output, and if the quantization error is truly random, the times between the changes have varying periods. We might speculate that by simulating a longer time or using a multiple frequency input so as to "exercise" the ADC, the resulting quantization errors are further randomized and the resulting error spectrum will be flatter than what is seen in Fig. 5.6.

5.1.2 Quantization Noise Voltage Spectral Density

If the quantization noise voltage spectrum is truly flat (Bennett's criteria hold) we can determine the noise power spectral density of $V_{Qe,RMS}$, $V_{Qe}^2(f)$ with units of V^2/Hz, or the noise voltage spectral density, $V_{Qe}(f)$ with units of $V/\sqrt{Hz}$ by solving

$$\frac{V_{LSB}^2}{12} = 2\int_0^{f_s/2} V_{Qe}^2(f) \cdot df \tag{5.4}$$

where the factor of 2 accounts for the power in the negative frequencies of the spectrum. Notice that we assumed a spectrum from DC to the Nyquist frequency, $f_s/2$. Assuming all of the quantization noise falls below the Nyquist frequency is the **worse-case** situation. Solving this equation yields

$$V_{Qe}(f) = \frac{V_{LSB}}{\sqrt{12 f_s}} = \frac{V_{REF+} - V_{REF-}}{2^N \sqrt{12 f_s}} \tag{5.5}$$

with units of $V/\sqrt{Hz}$. Note that the quantization noise spectral density is inversely proportional to the sampling frequency. Figure 5.2 shows how we can model the ADC as a summation of the input signal and the quantization noise.

After looking at Eq. 5.5 we might think that by simply increasing the sampling frequency we can reduce the amount of quantization noise an ADC introduces into an analog input signal. While increasing the sampling frequency spreads the quantization noise spectral density out over a wider range of frequencies (see Fig. 5.12) with a corresponding reduction in amplitude, the sampling frequency doesn't affect the total RMS quantization noise voltage. However, bandlimiting the spectrum using a filter reduces the amount of quantization noise introduced into an input signal (**this is important** and the reason mixed-signal design is so powerful). In the simplest case a lowpass filter is used on the output of the ADC to reduce the amount of quantization noise introduced into the signal. We can write the amount of noise introduced into an input signal over a range of frequencies using

$$V_{Qe,RMS}^2 = 2\int_{f_L}^{f_H} V_{Qe}^2(f)\cdot df \text{ where } f_L < f_H \le f_s/2 \tag{5.6}$$

Again, the factor of 2 is used to account for the contributions to $V_{Qe,RMS}$ in the negative frequency spectrum. Let's show that the sampling frequency doesn't affect the quantization noise, assuming Bennett's criteria are valid.

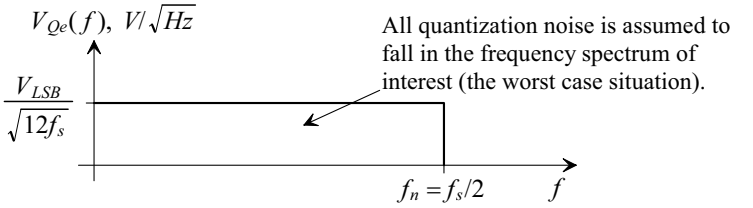


Figure 5.12 Quantization noise spectral density.

Example 5.1
Using the SPICE simulation that was used to generate Fig. 5.10b with sampling frequencies of 100 and 200 MHz calculate and simulate the amount of quantization noise introduced into the input signal.

Doubling the sampling frequency has little effect on the output quantization noise. Our calculated value was 1.13 mV using Eqs. (5.1) and (5.3) while the simulated values are 1.127 mV for both 100 and 200 MHz sampling frequencies. ■

Example 5.2
Suppose a 7 MHz sinewave with a peak amplitude of 400 mV is applied to the topology seen in Fig. 5.13. Calculate and compare to simulations the amount of quantization noise introduced into the sinewave when sampling frequencies of 100 and 200 MHz are used.

Again, the RMS value of the quantization noise added to the input signal is 1.13 mV. The simulated values are 1.12 mV and 1.15 mV for 100 and 200 MHz

sampling frequencies. Figure 5.13 shows the simulation results. Again, note that we assume Bennett's criteria are valid (input busy, large compared to an LSB, etc.)

In order to see the quantization noise introduced by the ADC we have to remove tones (input signal and its aliases) from the DAC's output signal. As seen in Fig. 5.13 the way we do this is by subtracting the S/H input signal (remembering simply sampling a waveform doesn't introduce noise) from the DAC's output. Using ideal components the delay through the system is nearly zero. In most systems we have to adjust the delay to match the delay through the mixed-signal system. This only works if the phase shift through the mixed-signal system is linear (constant delay). A more useful technique is to look at the spectrum of the output signal and remove, using for example Matlab™, the wanted signal and its aliases. ■

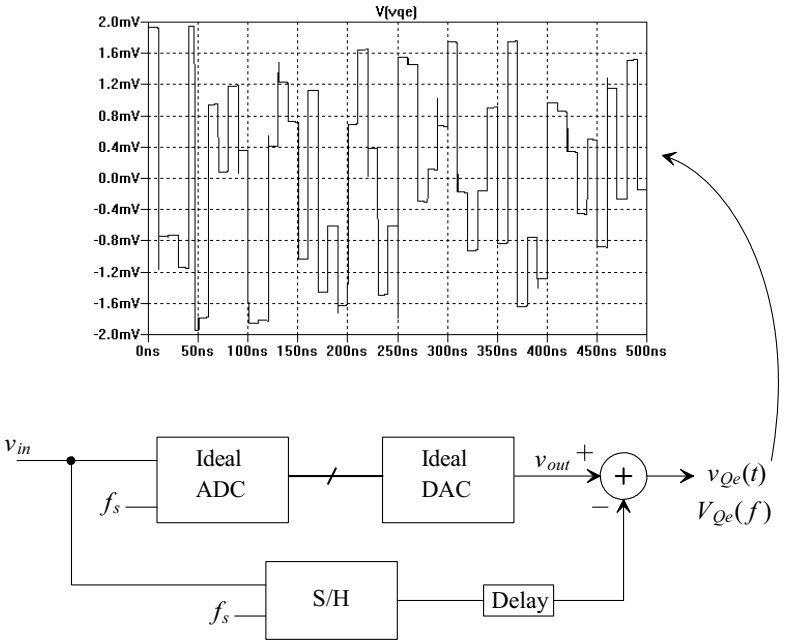


Figure 5.13 Determining quantization noise in a mixed-signal system.

Calculating Quantization Noise from a SPICE Spectrum

The spectrum of a signal in a SPICE simulation is generated using a Fast-Fourier Transform (FFT). We can estimate the RMS value of this spectrum by summing the individual components using

$$V_{Qe,RMS} = \sqrt{\sum_{k=0}^{M-1} V_{FFT}^2(k \cdot f_{res})} \tag{5.7}$$

where M is the number of points used in the FFT and f_{res} is the resolution of the FFT (the distance between adjacent points). Note that here we are assuming that $V_{FFT}(f)$ is an *RMS* value. If it's a peak value then the right-side of Eq. (5.7) should be divided by $\sqrt{2}$.

Figure 5.14 shows a portion of the quantization noise spectrum for the signal seen in Fig. 5.13. We can estimate the RMS noise from this spectrum assuming, as seen in Fig. 5.12, that the noise is bandlimited. While we can look at the specifics of the FFT in the simulation to get exact numbers, let's simply count the number of points seen in the plot (Fig. 5.14) and see how close we get to the exact answer of 1.13 mV. First note that −74 dB is 200 μV RMS and, as seen in Fig. 5.14, we are assuming a constant spectrum (white noise). Looking at the plot there are roughly 32 points from DC to 50 MHz so

$$V_{Qe,RMS} = \sqrt{32 \cdot (0.2\ mV)^2} = 1.13\ mV$$

is an exact match. Note, again, that since we are assuming all of the noise falls between DC and the Nyquist frequency, our estimated value (−74 dB) for the quantization noise is a little above the actual value we see in this range.

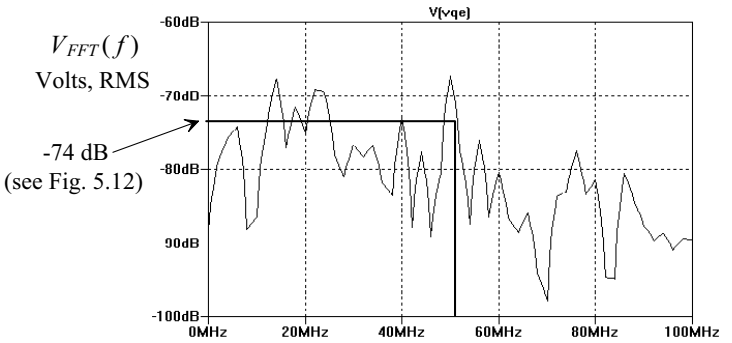


Figure 5.14 Spectrum of the quantization noise seen in Fig. 5.13.

Power Spectral Density

The variable $V_{Qe}^2(f)$ in Eq. (5.6) is a power spectral density (PSD) with units of V^2/Hz. To convert our SPICE plot above (RMS amplitudes) to a power spectral density we use

$$V_{Qe}^2(f) = \frac{V_{FFT}^2(f)}{f_{res}} \text{ with units of } V^2/Hz \tag{5.8}$$

and change the y-axis from $20 \cdot \log y$ to $10 \cdot \log y$. Note that we can then use Eq. (5.6) to calculate $V_{Qe,RMS}$ (without the factor of two since SPICE uses one-sided spectrums). To generate a SPICE plot similar to what is seen in Fig. 5.12 we use

$$V_{Qe}(f) = \frac{V_{FFT}(f)}{\sqrt{f_{res}}} \text{ with units of } V/\sqrt{Hz} \tag{5.9}$$

Again the quantization noise in the simulation would extend beyond $f_s/2$, unlike the assumed shape seen in Fig. 5.12.

5.2 Signal-to-Noise Ratio (SNR)

In the last section we developed the idea of treating an analog-to-digital converter (ADC) as a noisy circuit block where the output of the ADC is the sum of quantization noise and the input signal. Logically, the next step in our development of concepts is to characterize the mixed-signal system in terms of the signal-to-noise ratio (SNR).

If we apply a sinewave with an amplitude of V_p (and thus an RMS value of $V_p/\sqrt{2}$) to an ADC input, then, knowing the RMS quantization noise added to a busy ADC input signal is $V_{LSB}/\sqrt{12}$ (see Eqs. [5.1] and [5.3]), the resulting SNR is

$$SNR_{ideal} = 20 \cdot \log \frac{V_p/\sqrt{2}}{V_{LSB}/\sqrt{12}} \tag{5.10}$$

If we remember that

$$V_{LSB} = 1 \text{ LSB} = \frac{V_{REF+} - V_{REF-}}{2^N} \tag{5.11}$$

and we assume that the largest possible amplitude sinewave is the ADC input (to maximize the SNR), that is,

$$2V_p = V_{REF+} - V_{REF-} \tag{5.12}$$

then Eq. (5.10) can be rewritten as

$$SNR_{ideal} = 20 \cdot \log \frac{2^N \sqrt{12}}{2\sqrt{2}} = 6.02N + 1.76 \text{ (in dB)} \tag{5.13}$$

Effective Number of Bits

Equation (5.13) relates the number of bits used in a data converter to the ideal SNR when the input signal is a sinewave that ranges from V_{REF+} to V_{REF-}. In reality the measured SNR, in most cases, will be different from the ideal value calculated using this equation. When the SNR is measured, we relate it to the *effective number of bits* using

$$N_{eff} = \frac{SNR_{meas} - 1.76}{6.02} \tag{5.14}$$

where the measured SNR (SNR_{meas}) is specified in dB.

Example 5.3

Determine the effective number of bits for an ADC with , $V_{REF+} = 1.0$, $V_{REF-} = 0$ and a measured $V_{Qe,RMS}$ of 2 mV.

If we assume that the input peak amplitude, V_p, is $0.5 \cdot (V_{REF+} - V_{REF-})$ or 500 mV, then the measured SNR is given by

$$SNR_{meas} = \frac{0.5/\sqrt{2}}{2 \text{ mV}} = 177 = 45 \text{ dB}$$

The effective number of bits, N_{eff}, is, from Eq. (5.14), 7.18 bits. Note that to calculate $V_{Qe,RMS}$ we either take the output of the DAC and feed it into a spectrum analyzer or take the FFT of the digital output data of an ADC so that we get a plot, in either case, similar to what is seen in Fig. 5.14. ■

Example 5.4
Using the ideal 8-bit ADC and DAC shown in Fig. 5.4 with a sampling frequency of 100 MHz show, using SPICE, that applying a full-scale sinewave at 24 MHz to this configuration will cause the resulting SNR to approach the ideal value given by Eq. (5.10).

Let's begin by calculating SNR_{ideal}. From Eq. (5.13), SNR_{ideal} is roughly 50 dB, as the data converters have 8-bit resolution.

The time-domain input and output of the circuit and the corresponding DAC output spectrum, are shown in Fig. 5.15. The input to the ADC in Fig. 5.4 is a 24 MHz sinewave with a peak amplitude of 0.5 V centered on a DC voltage of 0.5 V (the peak-to-peak voltage of the input waveform is *VDD* or 1 V). The $V_{Qe,RMS}$ measured with SPICE, in a separate simulation using a S/H to remove the input signal and aliased signals from the DAC's output spectrum, is 1.4 mV. The simulated SNR is then $(0.5/\sqrt{2})/1.4\,\text{mV}$ or 253 (48 dB), which is close to the value, 50 dB, calculated at the beginning of the example. ■

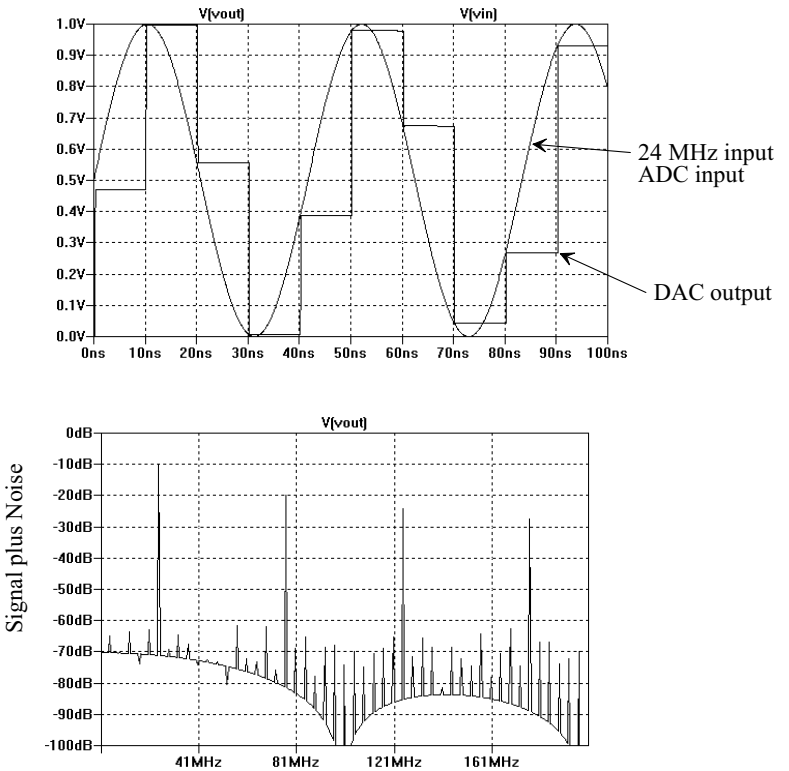


Figure 5.15 Signals from Ex. 5.4, simulating Fig. 5.4 with a 24 MHz sinewave input.

Coherent Sampling

It's important to understand that poor selection of the input frequency can result in an SNR that is different from the ideal value calculated using Eq. (5.13). For example, if our input frequency is 10 MHz while the clock frequency is 100 MHz then the sampling is *coherent*. The sampled points repeat with every cycle of the input signal. The amount of quantization error can then be, repeatedly, near $\pm 1/2$ LSB or much larger than $V_{LSB}/\sqrt{12}$.

Coherent sampling can be useful to minimize the undesired effects of the FFT, namely spectral leakage. In order to understand what is meant by "spectral leakage," consider the sinewave with infinite duration shown in Fig. 5.16a. When an FFT is performed on a time-domain waveform, the first step is to "window" the waveform. The simplest window is the rectangular window. In a simulation the duration of the sinewave is finite and set by the simulation time or transient stop time, T_{stop} . We can think of taking the infinite duration sinewave of (a) and multiplying it by the rectangular waveform of Fig. 5.16b to obtain the waveform used in the simulation, Fig. 5.16c. This multiplication means the resulting waveform is the convolution of the original sinewave spectral response (an impulse) and the frequency domain transform of the squarewave (a Sinc waveform) in the frequency domain. The result is that instead of the sinewave spectral response being an impulse function, as seen in Fig. 5.16d, it is a weighted Sinc waveform, Fig. 5.16e. Note that the FFT spectral response of the sinewave in (e) is spread out or "leaks" into the frequencies around the actual or continuous time response. The large ratio of the peak value of the Sinc pulse to its first sidelobe is usually undesirable. Rather, to minimize these sidelobes, other windowing functions are used. One commonly used

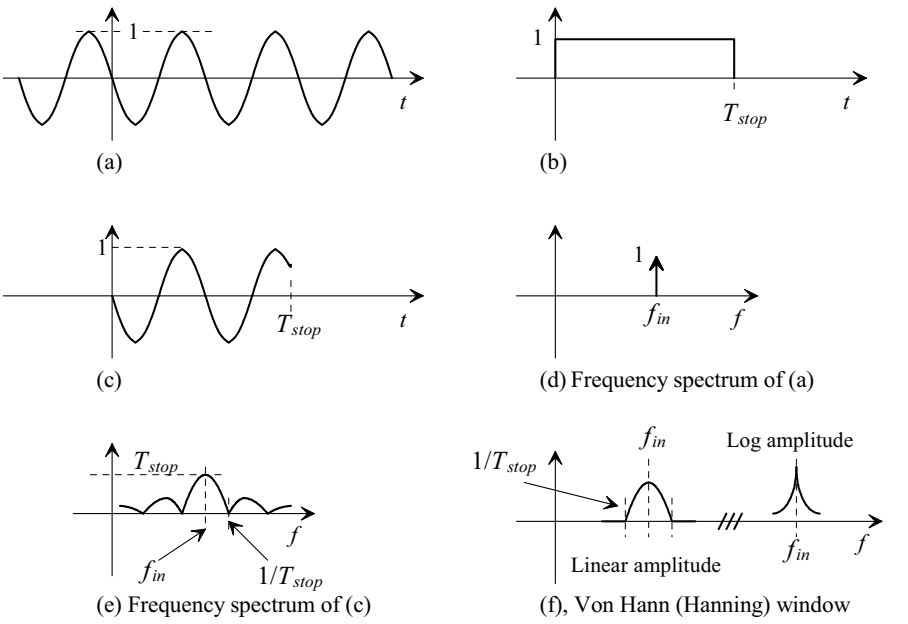


Figure 5.16 Showing how spectral leakage in an FFT affects the spectrum of a waveform.

window is the von Hann (a.k.a. Hanning or Cosine) window represented, without the sidelobes, in Fig. 5.16f. The response is shown on both linear and log amplitude scales and the width of the window is $2/T_{stop}$ at its base.

Selecting an input sinewave frequency, f_{in} , such that f_s /f_{in} is a whole number creates a condition where an integral number of input sinewave cycles fit perfectly into the simulation or measurement time (so the windowing function isn't important). This results in an output spectrum that contains isolated tones (no spectral leakage). Coherent sampling may be used to reduce the effects of spectral leakage when determining the SNR by ensuring $f_{in} << f_s$ (to randomize the quantization noise).

Signal-to-Noise Plus Distortion Ratio

In a practical data converter the output spectrum contains not only quantization noise but distortion resulting from nonlinearities and mismatch in the data converter circuitry. When we calculate the RMS quantization noise voltage using Eq. (5.7) and nonideal components, we are actually calculating the noise plus the distortion in the spectrum. Until this point we have only used ideal components, so that distortion in the output spectrums was absent. We can rewrite Eq. (5.7) to indicate that when it is used with a measured spectrum, both noise and distortion are included in the result as

$$V_{Qe+D,RMS} = \sqrt{\sum_{k=0}^{M-1} V_{FFT}^2(k \cdot f_{RES})} \tag{5.15}$$

The *signal-to-noise plus distortion ratio* is then given by

$$\text{SNDR} = 20 \log \frac{V_p/\sqrt{2}}{V_{Qe+D,RMS}} \tag{5.16}$$

The effective number of bits, from Eq. (5.14), can then be calculated using

$$N_{eff} = \frac{\text{SNDR} - 1.76}{6.02} \tag{5.17}$$

Example 5.5

Suppose that the test setup shown in Fig. 5.9 is used with an input sinewave having a frequency of 7 MHz, a peak amplitude of 0.5 V, and centered around 0.5 V (so that, once again, the sinewave swings from 0 V to 1 V.) Using SPICE simulation, determine the SNDR if there is a gain error in the ideal ADC in Fig. 5.9 (it's no longer ideal) so that each stage in the pipeline algorithm used to implement the ideal SPICE ADC has a gain of 2.1 instead of the ideal 2.0.

The resulting DAC output spectrum is shown in Fig. 5.17. The RMS noise plus distortion voltage, $V_{Qe+D,RMS}$, is 22.58 mV, using SPICE and remembering to remove, with the S/H, the desired terms at DC and 7 MHz as well as the undesired images at 93 MHz, 107 MHz, and 193 MHz. The SNDR is then

$$\text{SNDR} = 20 \log \frac{0.5/\sqrt{2}}{22.58 \text{ mV}} \approx 24 \text{ dB}$$

The effective number of bits is 3.7. In other words, a 5% gain error in the ADC amplifiers results in an effective resolution of less than half of the ideal value of 8-bits. ■

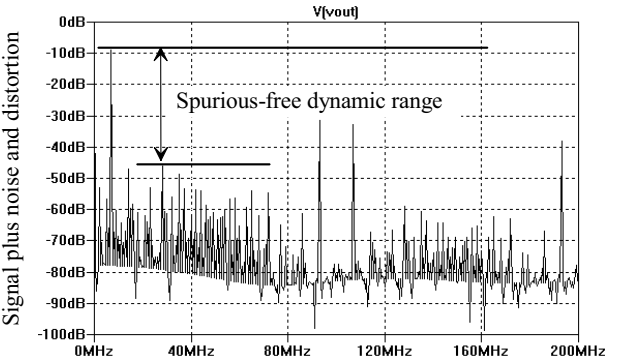


Figure 5.17 Output spectrum with ADC gain error (see Ex. 5.5).

Measuring SNDR requires a spectrum analyzer, when looking at the output of a DAC, or loading digital data into a program that can perform an FFT when looking at the output of an ADC. Trying to measure SNDR using a time domain instrument, such as an oscilloscope, is usually a waste of time because the dynamic range of the instrument is comparable to the dynamic range of the data converter under test. Spectrum analyzers utilize narrow band filtering on their input to reduce the inherent noise measured in a circuit and can have dynamic ranges in excess of 120 dB over a very wide frequency spectrum. Also note that the SNDR is sometimes abbreviated as *SINAD* (signal-to-noise and distortion.)

Spurious Free Dynamic Range

Another specification of interest is the data converter's spurious free dynamic range. This term relates the peak signal in the output spectrum (the input sinewave or carrier) to the largest spike in the output spectrum up to the Nyquist frequency. This can be written using

$$\text{SFDR(dBc)} = \text{input carrier(dB)} - \text{unwanted tone(dB)} \qquad (5.18)$$

For the spectrum shown in Fig. 5.17, the input sinewave (carrier) has an amplitude of 0.5 V (–9 dB RMS), while the largest unwanted tone has an amplitude of –46 dB. The SFDR of this data converter is then 37 dBc.

Dynamic Range

The dynamic range of a data converter can be specified in several ways. One definition is as the ratio of the largest output signal change (e.g., $[V_{REF+} - 1\ LSB] - V_{REF-}$) over the smallest output signal change (1 LSB). Remembering 1 LSB = $(V_{REF+} - V_{REF-})/2^N$ the dynamic range (DR) can be written as

$$\text{DR} = 20\log\frac{V_{REF+} - (V_{REF+} - V_{REF-})/2^N - V_{REF-}}{(V_{REF+} - V_{REF-})/2^N} = 20\log 2^N = 6.02N \qquad (5.19)$$

If a 1,000 to 1 dynamic range (60 dB) is required, then a data converter with at least 10 bits is needed.

Another way to specify DR is as the ratio of the RMS full-scale input sinusoid amplitude, $V_p/\sqrt{2}$, to the input sinusoid amplitude (RMS) that results in an SNDR of 0 dB. (The RMS amplitude of the input signal is equal to the RMS quantization noise plus distortion, $V_{Qe+N,RMS}$, when the SNDR is 0 dB.) This is nothing more than saying that the SNDR can be used to specify DR.

Example 5.6

Determine the DR for the ideal ADC in Ex. 5.4 using Eq. (5.19). Compare the result to the SNDR calculated in Ex. 5.5.

Using Eq. (5.19), the DR is 48.16 dB (the ideal value). The SNDR calculated in Ex. 5.5 was 24 dB. Clearly, the SNDR is a better indication of DR than is the value obtained using Eq. (5.19). ■

Specifying SNR and SNDR

The SNR and the SNDR are usually specified as a function of input sinewave amplitude at a fixed frequency, Fig. 5.18. The x-axis in Fig. 5.18 is normalized so that an input sinewave with a peak-to-peak amplitude of $V_{REF+} - V_{REF-}$ corresponds to 0 dB. We might be wondering how we differentiate between SNR and SNDR as both, up to this point, have been calculated in the same way (Eqs. [5.7] and [5.15]). We continue to calculate SNDR using a data converter output spectrum, remembering to zero out the desired tones and images, and Eq. (5.15) as was done in Ex. 5.5. When we calculate the SNR, we follow the same procedure except that now we also zero out any *spikes* or *spurs* (spurious responses) in the spectrum that are "sticking up" above the noise floor in the spectrum. These spikes come from imperfections in the data converter and result in distortion in the output waveform. Note in Fig. 5.18 how the SNR and the SNDR coincide until the input signal amplitude gets reasonably large (so the distortion tones increase in amplitude above the quantization noise).

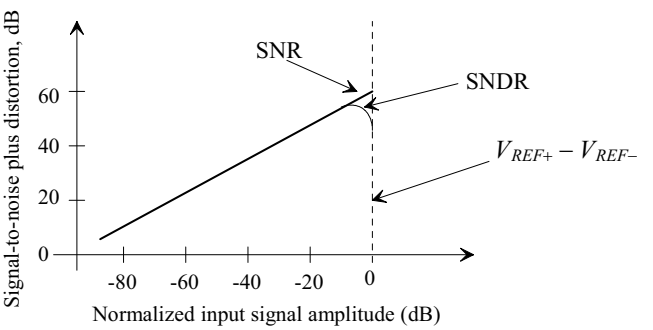


Figure 5.18 Specfying SNR and SNDR for a data converter.

5.2.1 Clock Jitter

We might assume that by using the ideal data converters in a system with "real world" input and clock signals we would get a resolution (number of bits) limited by the resolution of the ideal data converters used. However, if the clock signal isn't ideal, the

resolution will be less than ideal because of a problem known as aperture jitter. In this section we relate the sampling clock's jitter to the data converters SNR and thus the effective number of bits. Clock jitter is the variation in the period of the clock signal around the ideal value.

Figure 5.19 shows the basic problem. In this figure we have assumed the input sinewave frequency is running at the Nyquist frequency f_n (= f_s /2) so that the sampling point (when the sinewave crosses zero in this figure) is seeing the fastest transition in the input signal. We assume the peak amount of jitter in the clock signal is ΔT_s. For example, if the sampling clock frequency is 100 MHz (T_s = 10 ns) and the peak-to-peak clock jitter is 50 ps (= ΔT_s), then the specification of the sampling clock stability is 5,000 ppm (where parts per million [ppm] = 10^{-6} and ΔT_s = (stability, ppm) · ($1/f_s$)).

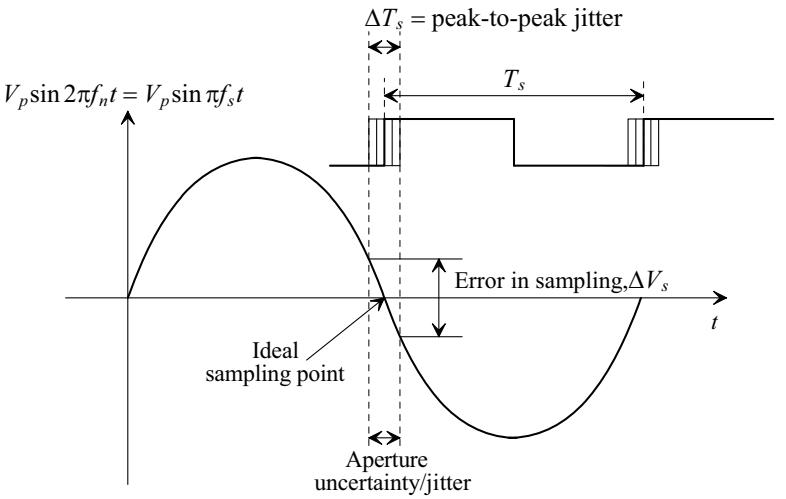


Figure 5.19 Data converter input signal and clock jitter.

The slew rate of the signal in Fig. 5.19, at the sampling point (when the clock signal transitions high), is given by

$$\frac{d}{dt}(V_p \sin \pi f_s t) = \pi f_s V_p \overbrace{\cos \pi f_s t}^{=1} = \pi f_s V_p \tag{5.20}$$

We can relate the uncertainty in the sampling instant, ΔT_s, to the uncertainty in the sampled voltage, ΔV_s, using

$$\frac{\Delta V_s}{\Delta T_s} = \pi f_s V_p, \text{ or } \Delta V_s = \Delta T_s \cdot \pi f_s V_p \tag{5.21}$$

If we require the uncertainty in the sampled voltage, ΔV_s, to be at most 0.5 LSB = (V_{REF+} $-V_{REF-}$)/2^{N+1} and we remember $V_p = (V_{REF+} - V_{REF-})/2$, then our maximum allowable peak-to-peak clock jitter can be determined for a particular data converter using

$$\Delta T_s \le \frac{1}{2^N} \cdot \frac{1}{\pi f_s} \tag{5.22}$$

or in terms of the sampling clock stability

$$\text{Stability, ppm} = \Delta T_s \cdot f_s = \frac{\Delta T_s}{T_s} = \frac{1}{\pi \cdot 2^N} \tag{5.23}$$

Table 5.1 relates the stability requirements placed on a sampling clock for a data converter resolution, N, if less than 0.5 LSBs aperture error or sampling voltage uncertainty is required of the data converter.

Table 5.1 Maximum jitter, ΔT_s, for 0.5 LSB sampling uncertainty.

Resolution, N	Stability, ppm	ΔT_s(max), ps	ΔT_s(max), ns
		If f_s = 100 MHz	If f_s = 44.1 kHz
6	5,000	50	113.4
8	1,250	12.5	28.3
10	312.5	3.13	7.1
12	78.1	0.78	1.77
14	19.5	0.2	0.44
16	4.9	0.05	0.11

Example 5.7

Suppose a phase-locked loop (PLL) is used to generate a clock signal for a data converter. If the resolution of the data converter is 10 bits and the frequency of the sampling clock coming from the PLL is 900 MHz, then specify the maximum jitter allowed in the output of the PLL. Assume that the maximum sampling error allowed is 0.5 LSB and that the data converter is sampling a sinewave with a frequency of 100 MHz.

Because the sinewave being sampled has a frequency below the Nyquist value, Eq. (5.22) cannot be used directly. Instead, after reviewing the derivation of this equation, we can rewrite it in terms of any input signal frequency, f_{in}, as

$$\Delta T_s \le \frac{1}{2^N} \cdot \frac{1}{2\pi \cdot f_{in}} \tag{5.24}$$

noting that when $f_{in} = f_n = f_s/2$, Eq. (5.24) reduces to Eq. (5.22). Using Eq. (5.24) with the numbers in this problem results in a peak-to-peak jitter of 1.56 ps! The reader familiar with PLL design will recognize that this is a very challenging requirement when designing a PLL (that is, to design a PLL with an output frequency of 900 MHz and an output jitter less than 1.56 ps). ■

We're now in a position to answer how, given the peak-to-peak clock jitter, the SNR of a data converter is degraded from the ideal value (given in Eq. [5.13]) when the input sampling clock isn't ideal. Rewriting Eq. (5.24) and assuming

$$\Delta T_s \ge \frac{1}{2^N} \cdot \frac{1}{2\pi \cdot f_{in}} \tag{5.25}$$

(a resolution loss ≥ 0.5 LSB), we get

$$\Delta T_s = \frac{1}{2^{N-N_{Loss}}} \cdot \frac{1}{2\pi \cdot f_{in}} \tag{5.26}$$

where f_{in} is, once again, the frequency of the input sinewave, and N_{Loss} is the number of bits lost due to the excess jitter. Assuming Eq. (5.25) is valid, then when N_{Loss} is zero, the loss in resolution is 0.5 LSB and Eq. (5.26) reduces to the equality condition in Eqs. (5.25) or (5.26). The ideal data converter's SNR, assuming the only non-ideal factor in the system is clock jitter, can be written as

$$\text{SNR} = 6.02 \cdot \overbrace{(N - N_{Loss} - 0.5)}^{\text{effective bits, } N_{eff}} + 1.76 \text{ (in dB)} \tag{5.27}$$

Example 5.8

For an ideal 8-bit ADC clocked at 100 MHz, determine the SNR of the data converter with 100 ps of peak-to-peak jitter in the input sampling clock, ΔT_s, assuming the ADC's input is a full-scale sinewave at 25 MHz.

We can write the number of bits lost by solving Eq. (5.26) as a function of peak-to-peak jitter as

$$N_{Loss} = N + 3.32 \cdot \log(2\pi \cdot f_{in} \cdot \Delta T_s) \text{ assuming } \Delta T_s \geq \frac{1}{2^N} \cdot \frac{1}{2\pi \cdot f_{in}} \tag{5.28}$$

or

$$N_{Loss} = 8 + 3.32 \cdot \log(2\pi \cdot 25\text{MEG} \cdot 100\text{ps}) = 2 \text{ bits}$$

The effective number of bits, N_{eff}, is then 5.5 and the SNR is 34.87 dB. ■

Using Oversampling to Reduce Sampling Clock Jitter Stability Requirements

Suppose we limit the maximum input frequency coming into an ADC to f_{in}, such that the sampling frequency is related to the maximum ADC input frequency by

$$\frac{f_s}{2} = K \cdot f_{in} = f_n \text{ or } f_{in} = \frac{f_s}{2K} \tag{5.29}$$

In other words, we are getting at least 2K samples for every cycle of the input sinewave (we are oversampling the input signal). If we were sampling at twice the Nyquist frequency (f_s), where K = 1, then we would get two samples for every cycle of the input signal. Notice that Eq. (5.24) gives the maximum jitter specification for a given input frequency and data converter resolution, but it doesn't specify the sampling frequency, f_s, or the sampling frequency period, T_s.

For a given maximum jitter, ΔT_s, we can reduce the requirements placed on the stability of the oscillator by increasing the sampling frequency. This can be written as

$$\text{Stability(new), ppm} = [\text{stability(old), ppm}] \cdot K \tag{5.30}$$

If we were sampling at 1 MHz and the stability required was 10 ppm, then the jitter in the sampling clock would be at most 10 ps, peak-to-peak. Increasing the sampling rate to 100 MHz, with 10 ps jitter would require an oscillator stability of 1,000 ppm. If we were to increase the sampling clock frequency to 1 GHz, then the stability of the clock would be at least 10,000 ppm (the period is 1 ns and the jitter is 10 ps or 1% [10,000 ppm]).

Example 5.9

In Table 5.1 we saw that the 16-bit data converter clocked at 44.1 kHz could have at most 111 ps peak-to-peak jitter to limit the sampling uncertainty to 0.5 LSB. We saw that the stability required of the oscillator under these circumstances was 5 ppm at 44.1 kHz. What would happen to the stability requirements of the oscillator generating the sampling clock if we increased the sampling clock frequency to $128 \cdot 44.1$ kHz = 5.645 MHz?

We know that the input bandwidth, prior to increasing the sampling frequency, is limited to 44.1 kHz/2 or 22.05 kHz (= B, the bandwidth of the input signal). We then assume the maximum input frequency, f_{in}, remains at or below 22.05 kHz even after we increase the sampling frequency. We can define the oversampling factor, in this example, as

$$K = \frac{f_s/2}{f_{in}} = \frac{2.822 \times 10^6}{22.05 \times 10^3} = 128$$

The jitter requirement remains 111 ps whether we use a sampling frequency of 44.1 kHz or 5.645 MHz. However, now that the clock frequency has increased to 5.645 MHz, the stability required of the oscillator has gone from approximately 5 ppm to 640 ppm. ■

It's important to note that the oversampling ratio, K, is given by

$$K = \frac{f_s/2}{B} = \frac{f_n}{B} \quad \text{for } f_{in} \le B \tag{5.31}$$

If we desire less than 0.5 LSBs aperture error, and we are using oversampling, then we can use Eqs. (5.22) and (5.31) to write

$$\Delta T_s \le \frac{1}{2^N} \cdot \frac{K}{\pi f_s} = \frac{1}{2^N} \cdot \frac{1}{2\pi B} \tag{5.32}$$

where, once again, B is the bandwidth of the input signal and K is the oversampling ratio. As shown by this equation and in Eq. (5.30), using oversampling reduces the requirements placed on the stability of the sampling clock.

A Practical Note

We need to point out that the effects of clock jitter are possible even if the clock is perfectly stable because of the clock's finite transition times (rise and fall times). If the rise time of the clock signal in Fig. 5.19 is finite, say 50 ps, then the same derivations and discussions concerning jitter in this section can be applied to determine how the SNR of the data converter is affected. We would assume the aperture window is a function of the transition times of the sampling clock signal. The slower the transition times, the larger the sampling uncertainty. In any practical data converter the SNR, and thus the effective number of bits, will be reduced because of the clock jitter and finite transition times as the input signal frequency increases.

5.2.2 A Tool: The Spectral Density

The observant reader may have noticed, in the last section, that we only discussed the peak-to-peak jitter, ΔT_s, and how it effects the data converter's performance. It is very useful, in many situations, to also have an idea how the spectrum or spectral

characteristics of the data converter's output change as a function of the random sampling jitter or a random variable such as noise. In this section we discuss tools useful in describing the spectrum of a random signal.

The Spectral Density of Deterministic Signals: An Overview

Consider the simple sinewave signal of the form

$$v_{in}(t) = V_p \sin 2\pi f_{in} t \quad \text{(units, V)} \tag{5.33}$$

This signal is termed "deterministic" because the signal has a well-defined shape whether it is continuous or sampled. We can find the average power of this signal, as a function of time $R_{in}(t)$, using the *autocorrelation function* (ACF) for continuous signals given by

$$R_{in}(t) = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} v_{in}(\tau) \cdot v_{in}(\tau + t) \cdot d\tau \ \text{(units, V}^2\text{)} \tag{5.34}$$

The average value of Eq. (5.33) as a function of time is then

$$R_{in}(t) = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} [V_p \sin 2\pi f_{in} \tau] \cdot [V_p \sin 2\pi f_{in}(\tau + t)] d\tau \tag{5.35}$$

or knowing

$$\sin A \cdot \sin B = \frac{1}{2}[\cos(A - B) - \cos(A + B)] \tag{5.36}$$

we can write

$$V_p^2 \cdot \sin 2\pi f_{in} \tau \cdot \sin 2\pi f_{in}(t + \tau) = \frac{V_p^2}{2}[\cos 2\pi f_{in} t - \cos 2\pi f_{in}(t + 2\tau)] \tag{5.37}$$

When we integrate this result, the term $\cos[2\pi f_{in}(t + 2\tau)]$ represents a sinusoid with a frequency of $4\pi f_{in}$ (remembering our integration variable is τ) and a phase shift of $2\pi f_{in} t$. Over a long period of time this term averages to zero. Therefore, we can write the average value of Eq. (5.33) as a function of time (the autocorrelation function)

$$R_{in}(t) = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} \frac{V_p^2}{2} \cdot \cos 2\pi f_{in} t \cdot d\tau = \frac{V_p^2}{2} \cdot \cos 2\pi f_{in} t \quad \text{(units, V}^2\text{)} \tag{5.38}$$

The spectrum of the average value of a function can be found by taking the Fourier transform of the autocorrelation function. The result is called the *power spectral density* function (PSD) and is given by

$$P_{in}(f) = \int_{-\infty}^{\infty} R_{in}(t) \cdot e^{-j \cdot 2\pi f \cdot t} \cdot dt \ \left(\text{units, V}^2\text{/Hz or V}^2 \cdot \text{s}\right) \tag{5.39}$$

The power spectral density function of Eq. (5.33) is then, with the help of Eq. (5.38),

$$P_{in}(f) = \frac{V_p^2}{4} \cdot [\delta(f + f_{in}) + \delta(f - f_{in})] \quad \text{(units, V}^2\text{/Hz)} \tag{5.40}$$

This is simply two impulses in the frequency spectrum located at $\pm f_{in}$ with an amplitude of $V_p^2/4$ (V^2/Hz). The *total average power* of this signal is given by

$$P_{AVG} = \int_{-\infty}^{\infty} P_{in}(f) \cdot df = 2 \cdot \int_{0}^{\infty} P_{in}(f) \cdot df \quad \text{(units, } V^2/\Omega \text{ or } watts\text{)} \tag{5.41}$$

assuming a 1-Ω (normalized) load, which, for Eq. (5.33), is $V_p^2/2$ (V^2).

The *voltage spectral density*, with units of $V/\sqrt{Hz}$, is simply the square root of Eq. (5.39) (that is, the square root of the PSD [$= \sqrt{P_{in}(f)}$]). The *root mean square* (RMS) voltage of a signal is given by

$$V_{RMS} = \sqrt{P_{AVG}} = \sqrt{2 \int_{0}^{\infty} P_{in}(f) \cdot df} = \sqrt{2 \int_{0}^{\infty} (voltage\ spectral\ density)^2 \cdot df} \tag{5.42}$$

The RMS value of Eq. (5.33) is simply, as one would expect for a sinewave, $V_p/\sqrt{2}$. Note the similarity between Eq. (5.42) and Eq. (5.7).

Example 5.10

Determine the ACF, PSD, average power, and RMS value of a signal $V(t)$ made up of three sine waves with peak amplitudes of V_1, V_2, and V_3 with frequencies of f_1, f_2, and f_3.

Using Eqs. (5.34) and (5.38), the ACF is

$$R(t) = \frac{V_1^2}{2} \cos 2\pi f_1 t + \frac{V_2^2}{2} \cos 2\pi f_2 t + \frac{V_3^2}{2} \cos 2\pi f_3 t \quad \text{(units, } V^2\text{)}$$

The PSD (positive frequencies) is determined using Eqs. (5.39) and (5.40)

$$P(f) = \frac{V_1^2}{4} \cdot \delta(f - f_1) + \frac{V_2^2}{4} \cdot \delta(f - f_2) + \frac{V_3^2}{4} \cdot \delta(f - f_3) \text{ (units, } V^2/\text{Hz)}$$

The average power, using Eq. (5.41), is

$$P_{AVG} = \frac{V_1^2 + V_2^2 + V_3^2}{2} \quad \text{(units, } watts\text{)}$$

Finally, the RMS value of the signal is given by

$$V_{RMS} = \sqrt{\frac{V_1^2 + V_2^2 + V_3^2}{2}} \quad \text{(units, V)}$$

Note that if we added phase shifts to our signals the results would be the same; the phase shift doesn't change the signal's average value, so we get the same results whether sines or cosines are used in our original spectrum. ■

Next, suppose that the sinewave specified by Eq. (5.33) is sampled at a rate of f_s

$$v_{in}(nT_s) = V_p \sin(2\pi f_{in} \cdot nT_s) \tag{5.43}$$

The ACF for a sampled signal can be written as

$$R_{in}(nT_s) = \lim_{N \to \infty} \frac{1}{(2N+1)} \sum_{k=-N}^{N} v_{in}(kT_s) \cdot v_{in}(kT_s + nT_s) \tag{5.44}$$

which results in

$$R_{in}(nT_s) = \frac{V_p^2}{2}\cos 2\pi f_{in} \cdot nT_s \quad \text{(units, V}^2\text{)} \tag{5.45}$$

The PSD is the Fourier transform of this equation,

$$P_{in}(f) = \frac{V_p^2}{4T_s}\sum_{k=-\infty}^{\infty}[\delta(f - f_{in} + kf_s) + \delta(f + f_{in} + kf_s)] \tag{5.46}$$

The RMS value of the sampled sinewave, Eq. (5.43), assuming we have passed the signal through an ideal reconstruction filter (RCF) with a bandwidth of $f_s/2$, is simply, once again, $V_p/\sqrt{2}$. The PSD of the signal, after passing through the RCF, has an amplitude of $V_p^2/4$ at frequencies of $\pm f_{in}$.

The Spectral Density of Random Signals: An Overview

Let's use our jitter discussion of the last section to illustrate how to look at the spectrum of a random signal. We'll do this in two parts: (1) we'll begin by assuming the jitter is a random variable that falls between two limits and has equal probability of lying anywhere in the region (just as was assumed for the quantization error probability density function when calculating the RMS quantization noise voltage in the last chapter), and (2) then assume the jitter has a Gaussian distribution around some average value (the more practical and realistic situation) and determine how the output of the ADC is affected.

Consider the representations of clock jitter shown in Fig. 5.20. Trace 1 in this figure shows the ideal position of the rising edge of a clock signal. This point is represented on the probability density function (PDF), $\rho(t)$, at time zero. On the next rising edge of the clock, trace 2, the edge is a little too early and is represented on the PDF as shown. We are assuming, probably incorrectly for most practical situations, that the rising edge of the clock is falling within the peak-to-peak boundaries with the equal probability of being in the correct position (as shown in trace 1) or at the edge of a boundary (as shown in trace 4). We also know that the area under the PDF curve in Fig.

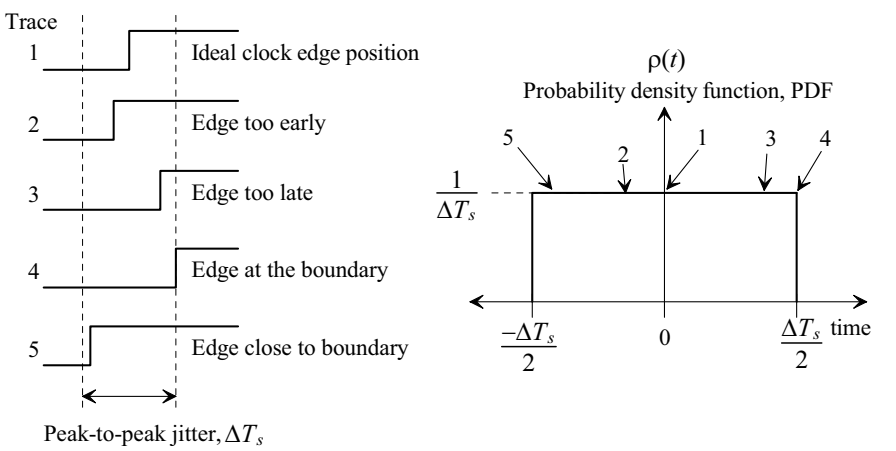


Figure 5.20 Clock jitter assuming the edge falls with the same probability anywhere within the peak-to-peak limits.

5.20 must equal unity, and the average value (also known as the mean or the expected value and denoted by $< y >$ or $\bar{y}$) of a PDF is given by

$$\text{Average value, } \bar{y}, = \int_{-\infty}^{\infty} t \cdot \rho(t) \cdot dt \tag{5.47}$$

Example 5.11
Determine the average value of the jitter with the PDF shown in Fig. 5.20.

We can use Eq. (5.47) to determine the average value of any PDF. Applying this equation to the PDF shown in Fig. 5.20 results in

$$\text{Average value, } \bar{y}, = \int_{-\Delta T_s 2}^{\Delta T_s/2} t \cdot \frac{1}{\Delta T_s} \cdot dt = 0$$

This somewhat obvious result means that the average position of the clock rising edge is the ideal position indicated by trace 1 in Fig. 5.20. Any PDF that is symmetrical about some center point will have an average equal to the center point. ■

The variance of the PDF is defined as the average of the square of the signal's departure from its average value. For a random signal this can be written as

$$\sigma^2 = \overline{(y-\bar{y})^2} = \int_{-\infty}^{\infty} (y-\bar{y})^2 \cdot \rho(y) \cdot dy \tag{5.48}$$

where σ is the standard deviation of the PDF (the square root of Eq. [5.48]). For our purposes, in this book, we can think of variance as the average power of a random (voltage) signal and the standard deviation as the RMS value of the signal (see Eqs. [5.41] and [5.42]). Example random signals include the time difference between the actual edge of a clock and the ideal edge location (jitter), the voltage difference between the input of an ADC and the ADC's reconstructed output (quantization noise), and the random fluctuations of electrons due to thermal motion in a resistor (thermal noise).

Example 5.12
Determine the RMS value of the jitter when the jitter has a probability density function, PDF, as shown in Fig. 5.20.

Using Eq. (5.48) the variance of the jitter PDF is

$$\sigma^2 = \int_{-\Delta T_s/2}^{\Delta T_s/2} t^2 \cdot \frac{1}{\Delta T_s} \cdot dt = \frac{1}{3 \cdot \Delta T_s} \cdot t^3 \Big|_{-\Delta T_s/2}^{\Delta T_s/2} = \frac{(\Delta T_s)^2}{12} \text{ (seconds}^2\text{)}$$

and thus the RMS jitter is

$$\sigma = \frac{\Delta T_s}{\sqrt{12}} \text{ RMS jitter, (seconds)}$$

where ΔT_s is the peak-to-peak jitter in the sampling clock rising edge. Note the similarity to the derivation of $V_{Qe,RMS}$ in Sec. 5.1.1. ■

A more useful discussion of jitter can be constructed if we assume the jitter has a Gaussian PDF, as shown in Fig. 5.21, and attempt to describe how the jitter in the

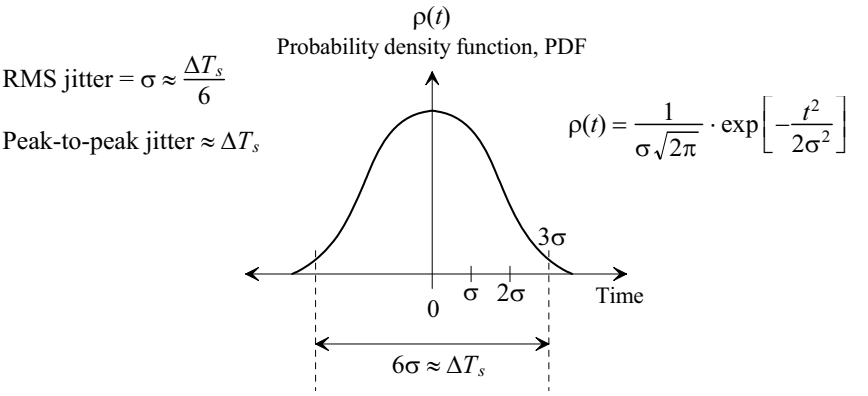


Figure 5.21 Sampling jitter with a Gaussian probability distribution.

sampling clock affects an ADC output spectrum with a single-tone input. Using Eqs. (5.20), (5.21), and (5.22), we can write the sampling error voltage (review Fig. 5.19), at a given time, as

$$\Delta V_s(t) = \delta T_s(t) \cdot V_p \cdot 2\pi f_{in} \cdot \cos 2\pi f_{in} t \tag{5.49}$$

where $\delta T_s(t)$ is a random variable indicating the jitter in the sampling clock at a given time. (The variable $\delta T_s(t)$ is the time difference between the actual clock transition time and the expected transition times that are spaced by T_s [see Fig. 5.20].) The peak-to-peak value of $\delta T_s(t)$ is ΔT_s, while its average value is zero. Again, we assume that the jitter probability distribution function is Gaussian, as seen in Fig. 5.21.

Rewriting Eq. (5.49) using a discrete time step nT_s, the sampling error can be written as

$$\Delta V_s(nT_s) = \overbrace{\delta T_s(nT_s) \cdot V_p \cdot 2\pi f_{in}}^{\text{Sampling error amplitude}} \cdot \overbrace{\cos 2\pi f_{in} nT_s}^{\text{Carrier term}} \tag{5.50}$$

We're interested in the spectrum of this error signal as it will add to our RMS quantization noise plus distortion voltage, effectively lowering the data converter's SNDR. Notice that the spectrum of Eq. (5.50) will have aliased components (and so will the sampled signal) so we need to filter out these components above f_s /2 (with the reconstruction filter.) Also note that multiplying the sampling error by the cosine term in Eq. (5.50) simply shifts the error spectrum to a frequency f_{in}. The cosine terms act like a carrier in an amplitude-modulated signal. This is illustrated in Fig. 5.22.

Example 5.13

Repeat Ex. 5.8 assuming the clock jitter has a Gaussian PDF.

In this example the peak amplitude of the input signal, V_p , is 0.5 V, the input frequency, f_{in} , is 25 MHz, and the peak-to-peak jitter is 100 ps. The average power in the sampling error amplitude spectrum is

$$P_{AVG,jitter} = \sigma^2 \cdot \frac{(V_p \cdot 2\pi f_{in})^2}{2} = \left(\frac{\Delta T_s}{6}\right)^2 \cdot \frac{(V_p \cdot 2\pi f_{in})^2}{2} \tag{5.51}$$

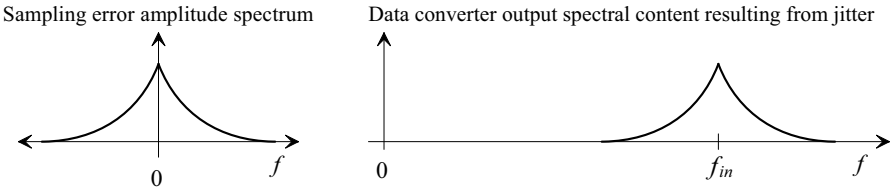


Figure 5.22 Modulating sampling error with an input sinewave frequency.

or

$$P_{AVG,jitter} = \left[\frac{100\,ps}{6}\right]^2 \cdot \frac{(0.5 \cdot 2\pi \cdot 25\text{ MHz})^2}{2} = 0.858 \times 10^{-6}\text{ V}^2$$

while the RMS voltage associated with this error is 0.926 mV. The quantization noise associated with this 8-bit data converter is

$$V_{Qe,RMS} = \frac{V_{LSB}}{\sqrt{12}} = \frac{V_{REF+} - V_{REF-}}{2^N\sqrt{12}} = 1.3\text{ mV}$$

The RMS noise voltage due to clock jitter and quantization effects is then given by

$$\sqrt{0.858 + 1.3^2}\ \text{mV} = 1.6\text{ mV}$$

We can calculate the SNR using

$$\text{SNR} = 20 \cdot \log\frac{0.5/\sqrt{2}}{1.6\text{ mV}} = 46.9\text{ dB}$$

giving an effective number of bits, from Eq. (5.14), equal to 7.53. Note that this is a *significant* improvement over what was calculated in Ex. 5.8, where the jitter variation was always the peak-to-peak value. ■

The PSD of the sampling error amplitude, described by Eq. (5.50), can be determined with the help of Eq. (5.41)

$$\sigma^2 \cdot \frac{(V_p \cdot 2\pi f_{in})^2}{2} = 2\int_0^\infty P_{jitter}(f) \cdot df \tag{5.52}$$

If the spectrum of the phase noise due to jitter is narrow, as seen in Fig. 5.22, then the spectral density of the sampling error, $P_{jitter}(f)$, is concentrated around the frequency of the input sinusoid. However, if we assume the phase noise spectrum is white and evenly distributed throughout the base spectrum (so that we integrate Eq. [5.52] from DC to $f_s/2$), we can write

$$P_{jitter}(f) = \frac{\sigma^2}{f_s} \cdot \frac{(V_p \cdot 2\pi f_{in})^2}{2} \tag{5.53}$$

The power spectral density of the sampling error voltage, assuming even distribution of the noise throughout the base spectrum, is shown in Fig. 5.23.

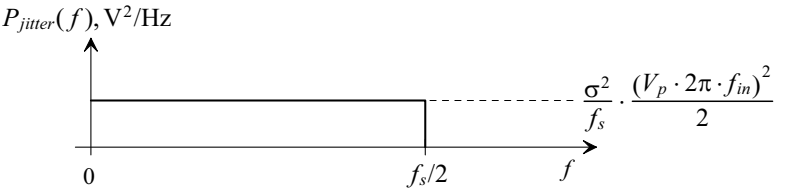


Figure 5.23 Sampling amplitude error PSD assuming sampling error spectrum is white.

Specifying Phase Noise from Measured Data

It's important to note that we have been discussing clock signals that are square waves (that is, have odd order harmonics) and so discussing jitter (a time-domain term) is, generally, more appropriate than discussing phase noise (a frequency domain term). However, because the terms are both widely used to indicate the same, basic, effect (a variation in the period of a periodic waveform), we will briefly discuss phase noise specification from measured oscillator data.

Consider the representation of a measured oscillator spectrum (power spectral density) shown in Fig. 5.24. In general, oscillator noise is specified in terms of the carrier voltage (or power) with units of dBc (decibels with respect to the carrier). The ratio of the power of the fundamental (called the carrier or sampling clock) at f_s is taken to the noise power in a bandwidth at some offset from the fundamental

$$\text{Phase noise, dBc/Hz} = \overbrace{10 \cdot \log\left[\int_{f_{L1}}^{f_{H1}} P_{osc}(f) \cdot df\right]}^{10 \cdot \log\left(\text{V}^2\right)} - \overbrace{10 \cdot \log P(f_s)}^{10 \cdot \log\left(\text{V}^2/\text{Hz}\right)} \qquad (5.54)$$

where the first term is the noise power at an offset from f_s.

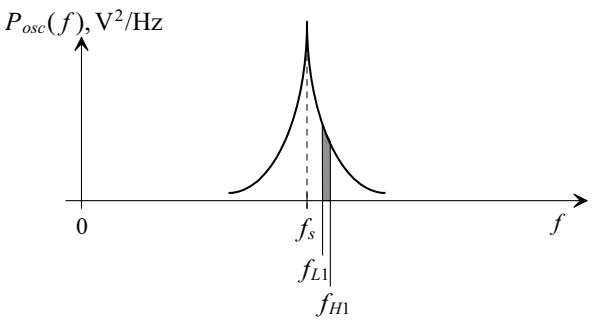


Figure 5.24 Measured oscillator spectrum.

5.3 Improving SNR using Averaging

Examine the two-path combination of ADCs and DACs shown in Fig. 5.25. The top ADC and DAC are the path we had back in Fig. 5.4 clocked at 100 MHz. The bottom path is a mirror image of the top except that its clock signal is inverted. The two resistors are used to sum (or more correctly average) the outputs of the DACs into a single output. This configuration effectively samples the input at 200 MHz (200 Msamples/s [$2 \cdot f_s$]).

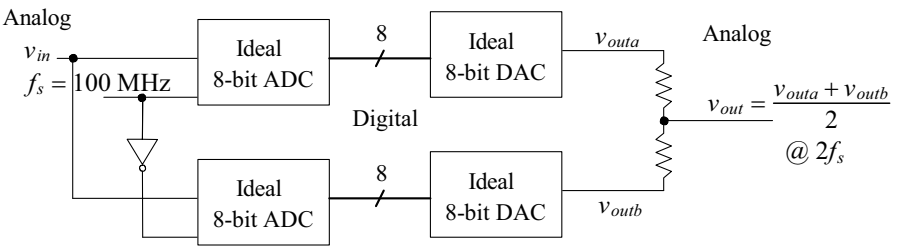


Figure 5.25 Using two paths to average the quantization noise.

Figure 5.26 shows the quantization noise PSD spectrums when the single path ADC/DAC is clocked at f_s and then at $2f_s$. Notice how the quantization noise is spread out over a wider frequency range when a higher sampling frequency is used. As seen in Ex. 5.1 the sampling frequency doesn't affect the RMS value of quantization noise. The area of the spectrums in Fig. 5.26 is equal to $V_{LSB}^2/12$ (negative frequencies are not shown) for both sampling frequencies.

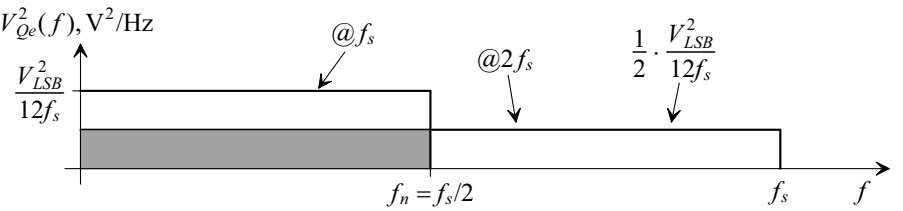


Figure 5.26 Quantization noise power spectral density for two sampling frequencies.

Now if we were to compare the quantization noise added to an input signal using a single path clocked at $2f_s$ to a system using two paths clocked at f_s we would find that the two path system added less noise. In order to understand why, notice that when we take two random uncorrelated variables, the quantization noise added to the input signal from each path in Fig. 5.25, and average (sum the power from each) them we get

$$V_{Qe,RMS,2-path}^2 = \frac{V_{Qe,RMS1}^2 + V_{Qe,RMS2}^2}{2} \tag{5.55}$$

or, generalizing this to *K*-paths

$$V_{Qe,RMS} = \frac{1}{\sqrt{K}} \cdot \frac{V_{LSB}}{\sqrt{12}} \tag{5.56}$$

Further note that if we were to add a lowpass filter to the output of our DAC, we could filter out part of the quantization noise and further decrease $V_{Qe,RMS}$. We use these ideas in this section to improve SNR; however, let's do an example before going any farther.

Example 5.14

Repeat Ex. 5.1 but use, to increase the sampling frequency, the two path topology seen in Fig. 5.25.

As discussed in Ex. 5.1, the RMS value of the quantization noise added to the input signal is 1.13 mV. Using two paths to increase the sampling frequency results in adding, using Eq. (5.56),

$$V_{Qe,RMS} = \frac{1}{\sqrt{2}} \cdot \frac{3.906 \text{ mV}}{\sqrt{12}} = 0.797 \text{ mV}$$

to the input signal. Figure 5.27 shows the simulation results (the quantization noise added to the input signal when two paths are used). The simulated $V_{Qe,RMS}$ is 0.787 mV. ■

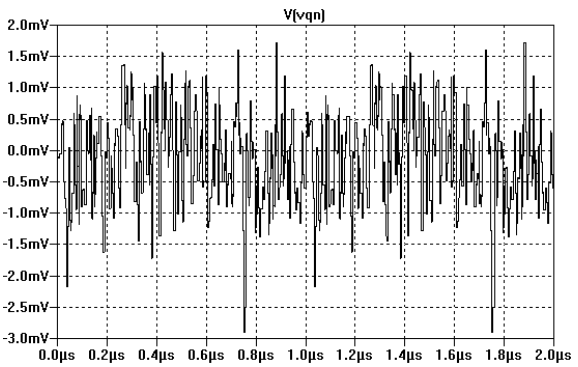


Figure 5.27 Simulated quantization noise using two-paths, see Ex. 5.14.

An Important Note

For averaging to effectively reduce the RMS quantization noise, the ADC and DAC must be linear to the final output resolution. In order to understand this in more detail examine Fig. 5.28. In the ideal situation, two adjacent codes are averaged to give an output code that falls exactly in between the outputs of the data converter. In the case where the data converter has a nonlinearity, the averaged point doesn't necessarily provide an output that is much different from the data converter outputs themselves. If the data converter contains a missing code (an input difference between two inputs at consecutive sampling times of 1-LSB results in the same output), then the averaging does nothing. If the data converter is nonmonotonic (an increase in the data converter's input doesn't result in an

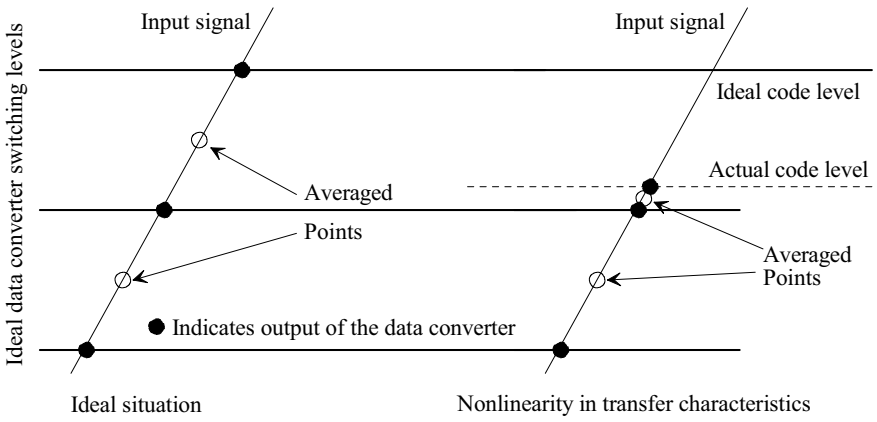


Figure 5.28 How ADC or DAC linearity affects averaging.

increase in its output) then the averaged value is meaningless. Finally, note that an input DC value (a digital code that isn't changing for the DAC, or an analog voltage that isn't changing for the ADC) or a value that isn't "busy" (not changing by at least 1 LSB in between sampling instances) will not benefit from averaging.

5.3.1 Using Averaging to Improve SNR

The averaging topology shown in Fig. 5.25 is not practical in most situations (one notable exception is the parallel combination of noise-shaping converters discussed later in the book). The silicon area required to implement the extra ADC and DAC generally costs more than is gained by the reduction in quantization noise. Figure 5.29 shows how we can add a digital averaging filter (in this figure averaging two ADC's outputs, or $K = 2$) to the output of the ADC to reduce quantization noise. By lowpass filtering the quantization noise PSD seen in Fig. 5.26, we reduce the amount of noise added to our signal. Unfortunately, this lowpass filtering also limits the range of allowable (wanted) signal frequencies. Notice that both the ADC and digital averaging filter are clocked at a rate of f_s. The averaging filter seen in Fig. 5.29 is described using Figs. 1.16 and 1.17.

We might, at this point, assume that we can use a low-resolution ADC, say 6-bits, with a significant amount of averaging ($K >> 2$, see the lowpass Sinc-shaped filters seen in Ch. 4) to attain large resolutions (again the ADC must be linear). Assuming the input to the ADC is busy and we place restrictions on the bandwidth of the signals coming into the ADC then we can increase the resolution by averaging. We have to place restrictions

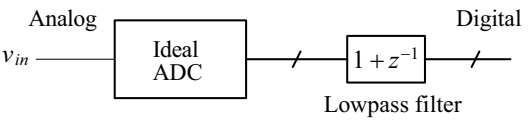


Figure 5.29 Using a digital averaging filter to reduce quantization noise.

on the bandwidth of the signal coming into the ADC because, unlike Fig. 5.25, we haven't increased the sampling rate of the signals. Therefore, the amplitude of the power spectral density seen in Fig. 5.26 remains unchanged. For an averaging of two, we would have to limit our desired input signal bandwidth to $f_s/4$, Fig. 1.17. If this wasn't the case, then an input sinewave at $f_s/2$ would average to zero.

Example 5.15

For the topology seen in Fig. 5.29, sketch the output quantization noise PSD. From this spectrum determine the RMS value of the quantization noise added to input signal. Compare this value to the one obtained using Eq. (5.56).

To begin let's write (see Eq. [1.40])

$$1 + z^{-1} = 2 \cdot \left| \cos \pi \frac{f}{f_s} \right|$$

so the PSD of the quantization noise can be written as

$$V_{Qe}^2(f) = \frac{V_{LSB}^2}{12 f_s} \cdot 4 \cdot \left| \cos^2 \pi \frac{f}{f_s} \right| \text{ for } 0 \le f \le \frac{f_s}{2}$$

This PSD is plotted in Fig. 5.30.

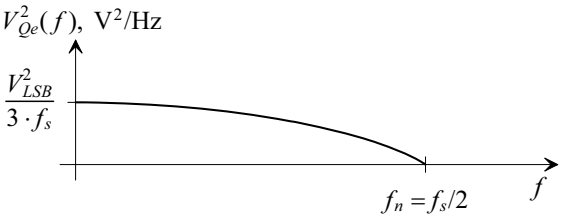


Figure 5.30 Quantization noise power spectral density after averaging two samples.

To determine the RMS quantization noise (see Eq. [1.29])

$$V_{Qe,RMS}^2 = 2 \int_0^{f_s/2} \frac{V_{LSB}^2}{12 f_s} \cdot 2\left(1 + \cos\left[2\pi \frac{f}{f_s}\right]\right) \cdot df$$

or

$$V_{Qe,RMS}^2 = \frac{V_{LSB}^2}{6} + \frac{V_{LSB}^2}{6\pi}\left[\sin 2\pi \frac{f}{f_s}\right]_{f=0}^{f=f_s/2} = \frac{V_{LSB}^2}{6} \neq \frac{1}{K} \cdot \frac{V_{LSB}^2}{12}$$

It would appear that our RMS quantization noise has actually increased! However, after we review the transfer characteristics of our filter seen in Fig. 1.17 we see the gain of the filter at low frequencies (where our desired signal resides) is two. Knowing this we can write the SNR as

$$SNR = 20 \log \frac{2V_p/\sqrt{2}}{V_{LSB}/\sqrt{6}} = 20 \log \frac{V_p/\sqrt{2}}{V_{LSB}/\sqrt{24}}$$

or, once again,

$$V_{Qe,RMS} = \frac{1}{\sqrt{K}} \cdot \frac{V_{LSB}}{\sqrt{12}} \tag{5.57}$$

■

Ideal Signal-to-Noise Ratio

Reviewing the derivation of Eq. (5.13) and using Eq. (5.57) we can write the ideal SNR for a data converter employing a digital (averaging or Sinc-shaped) lowpass filter as

$$SNR_{ideal} = 6.02N + 1.76 + 10\log K \tag{5.58}$$

where N is the number of bits (the resolution) of the data converter whose output is being averaged. Using no averaging, that is $K = 1$, results in this equation simplifying to Eq. (5.13). Averaging two samples causes the SNR_{ideal} to increase by 3 dB or the effective resolution of the data converter to increase by 0.5 bits. The increase in resolution due to averaging can be written as

$$\text{Increased resolution}, N_{Inc} = \frac{10\log K}{6.02} \tag{5.59}$$

Figure 5.31 shows how averaging the output of a data converter changes the effective resolution of the data converter. Again, note that the increase in resolution is based on the following assumptions: a busy input signal, the input signal is bandlimited, and the data converter is linear to the final resolution (data converter resolution, N, + improvement in resolution, N_{Inc}) coming out of the averaging circuit.

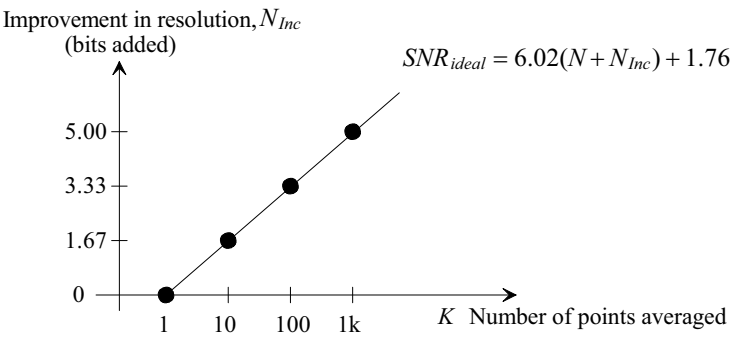


Figure 5.31 Using averaging to improve data converter resolution.

5.3.2 Linearity Requirements

Examine the cases for averaging ADC outputs shown in Fig. 5.32. In part (a) we show the ideal situation where the black dots indicate two consecutive outputs spaced by one LSB (time is not shown in this figure). The ADC outputs in part (a) are located on the ideal levels, while the averaged output falls exactly in the middle of these levels (and hence our increased resolution). Part (b) of this figure shows the situation where the ADC outputs are shifted downwards by 0.5 LSBs from their ideal levels. Following this offset, the averaged point shifts downwards as well. In part (c) the top output of the ADC (the top

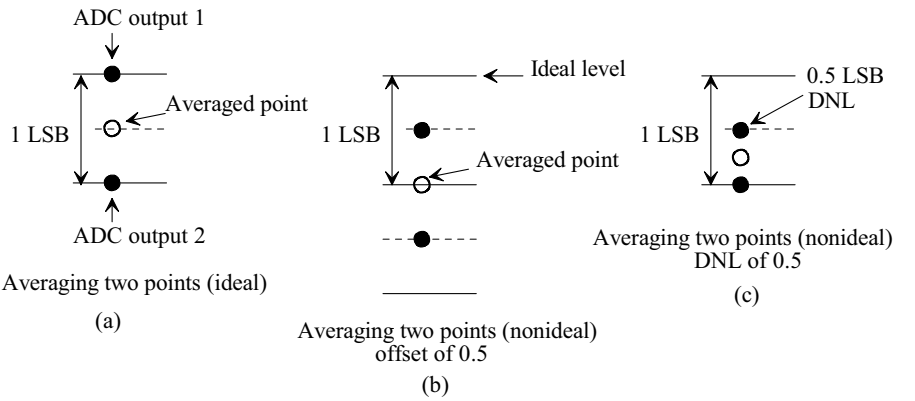


Figure 5.32 Linearity requirements when averaging.

black dot) is shifted downwards by 0.5 LSBs and so the averaged point shows a 0.25 LSB offset from its ideal position. While we used a single LSB difference to show averaging, we could use any number of LSBs to show that the ADC accuracy must be equal to or better than the desired final digital filter output accuracy.

The number of bits in the ADC (its resolution) N, and the number of bits improvement in resolution after filtering, N_{Inc}, are used with the final, total number of bits (the number of bits coming out of the digital filter) to give

$$N_{Final} = N + N_{Inc} \tag{5.60}$$

The ADC output should ideally change in increments of the exact LSB voltage. In reality, the changes will be different from the ideal output levels (as just discussed). In order to achieve an increase in the number of final bits, the output of the ADC must be accurate (its actual levels must be spaced from the ideal levels) to within

$$\pm \frac{V_{REF+} - V_{REF-}}{2^{N_{Final}+1}} = \pm (0.5\ LSB) \cdot \frac{1}{2^{N_{Inc}}} \tag{5.61}$$

where no averaging (N_{Inc} = 0 and K = 1) means the ADC is at least 0.5 LSBs accurate. This is a *significant limitation* when using averaging to increase the resolution of an ADC. This is especially true when a resolution greater than 10 bits is desired with INL and DNL less than ±0.5 LSBs. In the next section, and in the next chapter, we will look at feedback topologies that may relax the accuracy requirements placed on the ADC and allow averaging to more effectively remove quantization noise.

5.3.3 Adding a Noise Dither

Our assumption, when discussing the benefits of averaging or calculating the spectral density of the quantization noise, falls apart for DC or slow-moving signals (the ADC input is not "busy"). In order to help with this problem consider adding a noise signal to the ADC input that has a frequency content that falls within the range

$$\frac{f_s}{2K} \le f < \frac{f_s}{2} \tag{5.62}$$

so that it can be filtered out with the averaging filter (see Fig. 4.10). This noise is often called *dither* (a state of indecision or agitation) because it helps to randomize the spectral content of the quantization noise, making it white.

Figure 5.33 shows the basic idea. In part (a) a DC signal is applied to the ADC that falls halfway between two ADC transition codes spaced apart by 1 LSB. The output code of the ADC remains unchanged with time. In part (b) a noise signal is added to the DC input which has two benefits: (1) the quantization noise (the difference between the input signal and the reconstructed ADC output code) changes with time, and (2) the output of the ADC has some variation which makes it possible to determine the DC voltage after averaging.

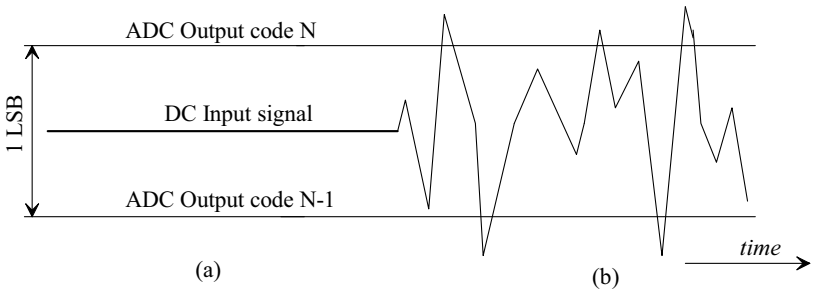


Figure 5.33 (a) DC input signal and (b) DC input signal with dither added.

We can add the noise signal to our desired input signal with a circuit similar to that shown in Fig. 5.34. Simple resistors add and reduce the noise signal applied to the ADC input. The noise signal source is, most easily, derived from some sort of asynchronous logic circuit and has a peak amplitude (before reduction) of *VDD* (= 1 V in this book). In this figure note that we have indicated that the dither signal amplitude should be approximately 0.5 LSB RMS (remembering the signal is, ideally, random and bandlimited as specified by Eq. [5.62]). This number, 0.5 LSB RMS, is subjective, and no exact rules as to its selection can be given other than the desire that the peak-to-peak

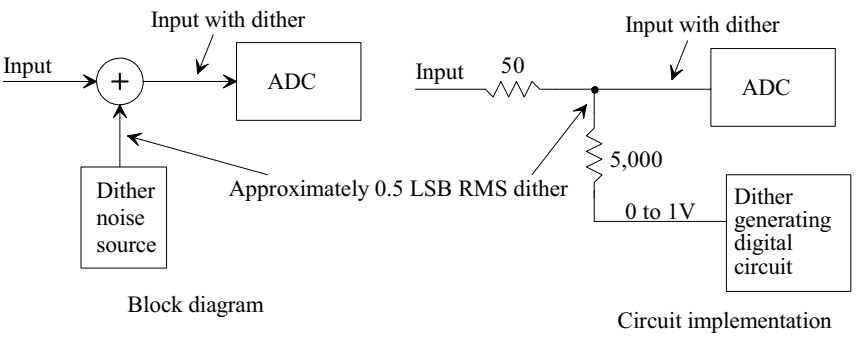


Figure 5.34 Adding dither to an ADC input signal.

amplitude be greater than 1 LSB. One *disadvantage* of adding the dither is that the allowable range of input signals shrinks. A DC signal at *VDD* − 1 LSB will not benefit from dithering since the ADC will be at its full-scale output.

Before we discuss the implementation of a dither source, consider one possibility (a Gaussian PDF) for the desired probability density function (PDF) of the dither signal and DC input shown in Fig. 5.35 (the input to the ADC). If we average this signal over a long time, we get the average or DC input signal since the dither averages to zero. This would also mean that we can have some dither spectral content below $f_s/2K$ as long as we average enough ADC output samples to make its contribution to the SNDR small. It is generally a good idea to use Eq. (5.62) as a guide for allowable dither spectral content. Finally, it's important that any dither signal we generate has a symmetrical PDF (the dither signal must average to *VDD*/2 before amplitude reduction). If not, an unknown DC offset (the known DC offset is the *VDD*/2 attenuated by the resistive divider in Fig. 5.34) in the data converter's (actually the filter's) output will result.

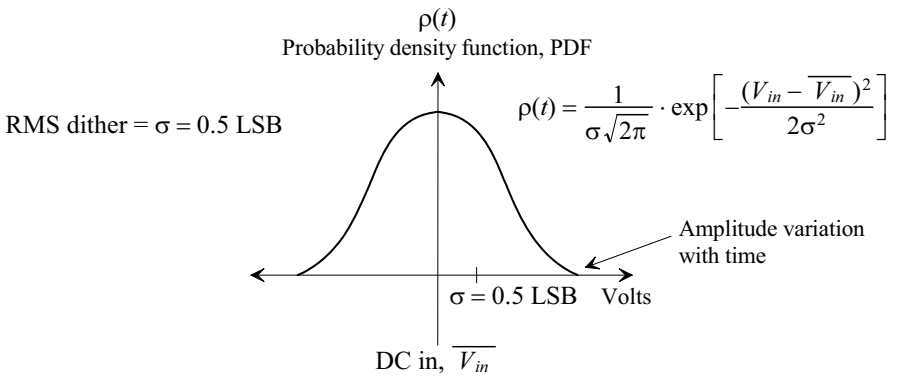


Figure 5.35 Input to the ADC, dither and DC, with a Gaussian probability distribution.

An example of an implementation of a dither noise source is shown in Fig. 5.36. The outputs of the rows of inverters, which are tied together, will occur asynchronously and fight against each other causing the amplitude of the dither signal to occupy levels

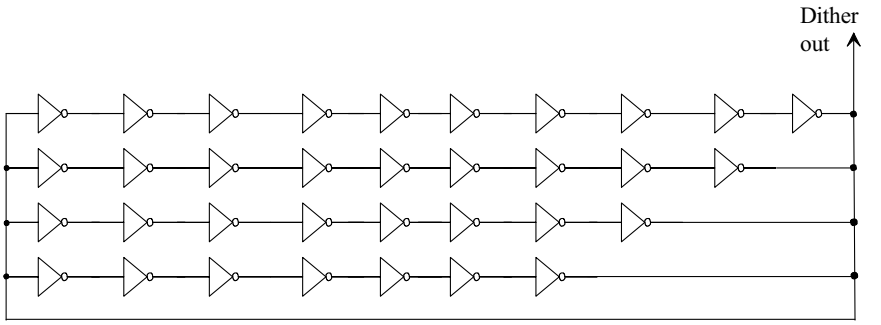


Figure 5.36 One possible implementation of a dither circuit.

other than the normal logic levels of VDD and ground for significant amounts of time. The dither signal can be made more random by adding more rows of inverters. The challenge to this design is setting the number of inverters used in each row so that the spectral content falls within the desired range (which may require a large number of inverters) and keeping the output of the dither circuit uncorrelated with the sampling clock. Other techniques for generating random noise, such as using linear serial feedback registers, can be found in most books covering communication systems.

5.3.4 Jitter

We can apply the averaging discussion just developed directly to the jitter discussion presented earlier in the chapter and answer the question, "How does averaging effect the sampling amplitude error power (resulting from jitter) in a data conversion system?" If we assume that the jitter has a Gaussian PDF, then the average power in the sampling error amplitude, from Ex. 5.13, is

$$P_{AVG,jitter} = \left[\sigma \cdot \frac{V_p}{\sqrt{2}} \cdot 2\pi f_{in} \right]^2 \tag{5.63}$$

where σ is the standard deviation of the jitter (see Fig. 5.21). It may be helpful to rewrite Eq. (5.57) in terms of the quantization error power as

$$P_{Qe,AVG} = (V_{Qe,RMS})^2 = \frac{1}{K} \cdot \frac{V_{LSB}^2}{12} \tag{5.64}$$

and apply the same derivation to Eq. (5.63) to give

$$P_{AVG,jitter} = \frac{1}{K} \cdot \left[\sigma \cdot \frac{V_p}{\sqrt{2}} \cdot 2\pi f_{in} \right]^2 \tag{5.65}$$

This equation shows that the sampling error amplitude power, $P_{AVG,jitter}$, introduced into the data converter's output spectrum decreases with averaging. Averaging two samples causes the sampling error amplitude power to decrease by 3 dB. This effectively reduces the jitter requirements placed on the sampling clock. While this may not appear to be very significant at first glance, consider what happens if, for example, 256 samples are averaged (K = 256). The sampling error power decreases by 48 dB, making clock jitter, when using a reasonably stable oscillator, almost not an issue. Also note that a doubling in the jitter's standard deviation, σ, results in a 6 dB increase in sampling error amplitude power.

5.3.5 Anti-Aliasing Filter

The use of averaging will also lead to relaxed requirements of the anti-aliasing filter (AAF). Figure 5.37a shows the requirements placed on the AAF without averaging. As we saw in Ch. 2, ideally, the transition from the 3 dB frequency to the "stop frequency" or Nyquist frequency should be infinitely sharp (the filter should abruptly change from a gain of unity to a gain of zero [something small]). When using averaging, Fig. 5.37b, we have to limit our desired input signal bandwidth to B. The rolloff of the filter in part (b) of the figure can be much more gradual and in many cases a simple, single pole, RC filter is all that's needed for an AAF. Also, our averaging filter will attenuate the ADC output spectrum, as seen in Fig. 1.17, and help to remove input signal power above $f_s/2K$. The

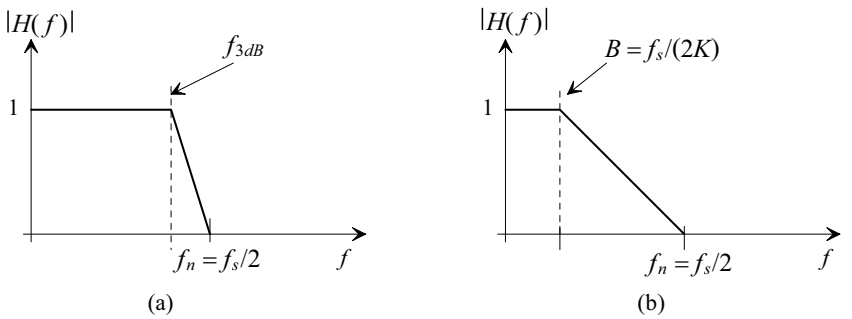


Figure 5.37 (a) AAF requirements without averaging, and (b) AAF requirements with averaging.

significance of this will be easier to see as the number of points averaged increases and our averaging filter's response gets sharper with more attenuation (see Fig. 4.10). Of course, the penalty for the relaxed requirements of the AAF is reduced signal bandwidth for a fixed sampling frequency.

5.4 Using Feedback to Improve SNR

By averaging the outputs of an ADC, or interpolating between inputs of a DAC, the effective data converter resolution can be increased. As specified by Eq. (5.58), every doubling in (octave increase in) K (where K is the number of points averaged) results in a 0.5-bit increase in effective resolution. An effective ADC resolution increase of 6-bits requires averaging 4,096 samples. If a 1 MHz signal bandwidth is of interest, our sampling clock frequency, f_s, will have to be 8.192 GHz!

In this section we briefly introduce the idea that feedback can be used with data converters (ADCs and DACs) to improve overall data conversion system performance (lower the amount of averaging or oversampling needed to attain a given resolution over a certain bandwidth). A topology of this nature is called a *modulator* or *coder* (for analog-to-digital conversion) or a *demodulator* or *decoder* (for digital-to-analog conversion). The complete analog-to-digital interface (a circuit block that functions as an ADC) would be made up of a modulator and a lowpass (decimating) filter, while the digital-to-analog interface (a circuit block that functions as a DAC) would consist of an interpolating filter and a demodulator. This can be confusing since, for example, a modulator will contain a low-resolution ADC in a feedback configuration which, together with the decimating filter, behaves like a high-resolution ADC.

The basic topology of a feedback modulator or coder is shown in Fig. 5.38. Depending on the circuit blocks used for $A(f)$ and $B(f)$ feedback modulators can be separated into two categories: *predictive modulators* and *noise-shaping modulators*.

Predictive modulators (a.k.a. predictive coders), such as delta-modulation, attempt to feed back an analog signal with the same value as the input signal. This drives the output of the summer to zero, reducing the required input range of the ADC and, possibly, the quantization error introduced by the ADC. *Predictive modulators effectively output the change in the input signal over time.* Noise-shaping modulators, an example being delta-sigma-modulation, also known as sigma-delta-modulation, on the other hand,

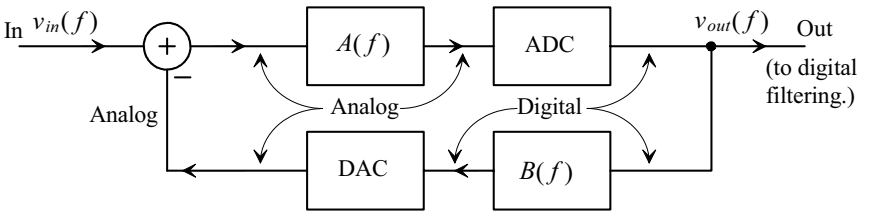


Figure 5.38 Block diagram of a feedback modulator.

feed back, and output, the average value of the input signal. This signal can be filtered (averaged) to reduce the accuracy required of the analog circuit components. *Noise-shaping modulators effectively output the average of the input signal over time.* In a noise-shaping data converter the averaging and decimating filter, as discussed earlier, is connected to the output of the modulator. Because of the averaging used in noise-shaping modulators, the analog components, in the forward path of Fig. 5.38, require less accuracy. However, the DAC's output, in the feedback path (which is subtracted from the input), doesn't experience the averaging so, once again, the DAC must be linear to the final desired resolution of the data converter. DAC linearity concerns have led to the use of a single-bit DAC (an inverter), in many noise-shaping data converter applications. The one-bit DAC is inherently linear. (Two output points determine a line!) Because of the relaxed requirements placed on the analog circuit components, *we will concentrate the next chapters, in detail, on noise-shaping topologies for both ADCs and DACs*. Notice that both predictive and noise-shaping modulators utilize oversampling.

In order to understand these statements in more detail, let's use the additive quantization noise model for the ADC developed in this chapter, Fig. 5.2. Figure 5.39 shows Fig. 5.38 redrawn using this model where the quantization noise is represented in the frequency domain by $V_{Qe}(f)$. We can relate the inputs (the wanted input signal and the unwanted quantization noise) to the output of the feedback modulator by

$$v_{out}(f) = \overbrace{\frac{A(f)}{1+A(f)\cdot B(f)}}^{\text{Signal transfer function, } STF(f)} \cdot v_{in}(f) + \overbrace{\frac{1}{1+A(f)\cdot B(f)}}^{\text{Noise transfer function, } NTF(f)} \cdot V_{Qe}(f) \qquad (5.66)$$

In a predictive modulator the feedback filter, $B(f)$, has a large gain so that, ideally, the fed back signal equals the input signal. If $A(f) = 1$ (a wire), then both the *STF* (signal transfer function) and the *NTF* (noise transfer function) have a value of, approximately, $1/B(f)$. Recovering the input signal requires passing the output of the predictive modulator through an analog filter with a transfer function of precisely $B(f)$ (noting that $B[f]$ is a digital filter in the modulator of Fig. 5.39). The required precision of the analog filter (the matching between the filter in the modulator and the filter in the demodulator) limits the attainable resolution when using predictive modulators. Notice that both the input signal and the quantization noise experience the same spectral shaping (spectral discrimination is absent in a predictive modulator). Also note that the name "predictive" comes from the modulator attempting to predict the input signal in order to drive the output of the summer to zero. If the prediction is perfect, the signal that is fed back exactly matches the input signal.

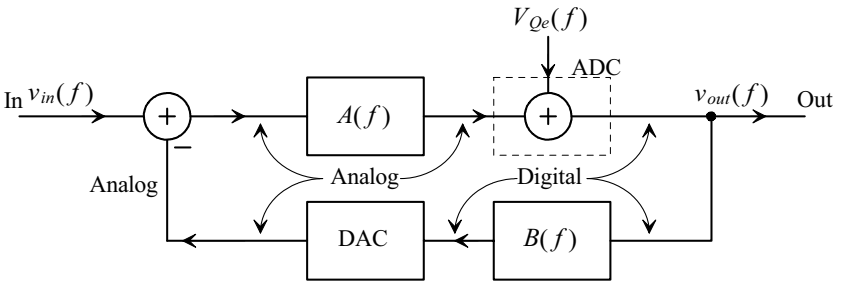


Figure 5.39 Block diagram of a feedback modulator.

In a noise-shaping modulator the gain of the forward path, $A(f)$, is large in the signal bandwidth so that the *STF* is approximately unity (assuming $B[f] = 1$). The *NTF*, on the other hand, will approach zero, ideally, in the bandwidth of interest. Note that the signal spectrum passes through the modulator essentially unchanged, while the quantization noise spectrum is shaped (and thus the name *noise-shaping*). No precision filter or analog components are required, as discussed earlier, except, perhaps, for the DAC in the feedback path of the modulator. We'll see in the next chapter that if $A(f)$ is an integrator, the quantization noise is pushed to higher frequencies so that it can be removed with the averaging filter. This is a very important concept, as a noise-shaping modulator *does not* reduce the quantization noise to attain higher resolutions, but rather pushes the noise to frequencies outside of the signal bandwidth of interest.

ADDITIONAL READING

[1] R. J. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, Springer, 2007. ISBN 978-1402075001

[2] Engineering Staff Analog Devices Inc., *Data Conversion Handbook (Analog Devices)*, Newnes, 2005. ISBN 978-0750678414

[3] M. Gustavsson, J. J. Wikner, and N. Tan, *CMOS Data Converters for Communications*, Springer, 2000. ISBN 978-0792377801

[4] S. R. Norsworthy, R. Schreier, and G. C. Temes (eds.), *Delta-Sigma Data Converters: Theory, Design, and Simulation*, Wiley-IEEE Press, 1997. ISBN 978-0780310452

[5] J. C. Candy and G. C. Temes (eds.), *Oversampling Delta-Sigma Data Converters*, Wiley-IEEE Press, 1992. ISBN 978-0879422851

[6] S. K. Tewksbury and R. W. Hallock, *Oversampled, Linear Predictive and Noise-Shaping Coders of Order N*>1, IEEE Trans. Circuits and Sys., Vol. CAS-25, pp. 436-447, July 1978.

[7] W. R. Bennett, "Spectra of Quantized Signals," *Bell System Technical Journal*, Vol. 27, pp. 446-472, July 1948.

QUESTIONS

5.1 Develop an expression for the effective number of bits in terms of the measured signal-to-noise ratio if the input sinewave has a peak amplitude of 50% of ($V_{REF+} - V_{REF-}$).

5.2 When using Eq. (5.14) what is the assumed ADC input signal? Put your answer in terms of the ADC reference voltages.

5.3 Describe, in your own words, the difference between specifying SNR and SNDR.

5.4 Using SPICE simulations with an ideal ADC and DAC, show how coherent sampling can result in an RMS value of quantization noise larger than what is specified by Eq. (5.3). Comment on the shape of the quantization noise's spectrum.

5.5 Suppose a perfectly stable clock is available (ΔT_s is zero in Eq. [5.21]). Would we still have a finite aperture window if the clock has a finite rise time? Describe why or why not?

5.6 How do the number of bits lost because of aperture jitter change with the frequency of an ADC input sinewave? If the ADC input is a DC signal, is aperture jitter a concern? Why?

5.7 Why must Bennett's criteria be valid for the averaging filter in Fig. 5.29 to reduce the quantization noise in the digital output signal? Give an example input signal where averaging will not reduce quantization noise.

5.8 Assuming Eq. (5.57) is valid, rederive Eq. (5.13) including the effects of averaging K ADC output samples. Is Eq. (5.13) or the equation derived here valid for a slow or DC input signal? Comment on why or why not.

5.9 If Bennett's criteria are valid, does averaging ADC outputs (or DAC inputs) put any restrictions on the bandwidth of the input signal? Why? Give an example.

5.10 How accurate does an 8-bit ADC have to be in order to use a digital filter to average 16 output samples for a final output resolution of 10-bits (see Eq. [5.59])? Assume the ideal LSB of the 8-bit converter is 10 mV. Your answer should be given in both mV and % of the full-scale.

5.11 Show the detailed derivation of Eq. (5.66).

5.12 Summarize, and compare, the advantages and disadvantages of predictive and noise-shaping data converters.