Chapter

28

Data Converter Fundamentals

Data converters (a circuit that changes analog signals to digital representations or vice-versa) play an important role in an ever-increasing digital world. As more products perform calculations in the digital or discrete time domain, more sophisticated data converters must translate the digital data to and from our inherently analog world. This chapter introduces concepts of data conversion and sampling which surround this useful circuit.

28.1 Analog Versus Discrete Time Signals

Analog-to-digital converters, also known as A/Ds or ADCs, convert analog signals to discrete time or digital signals. Digital-to-analog converters (D/As or DACs) perform the reverse operation. Figure 28.1 illustrates these two operations.To understand the functionality of these data converters, it would be wise first to compare the characteristics of analog versus digital signals.

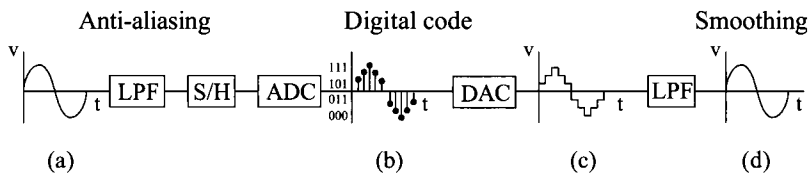

Figure 28.1 Signal characteristics caused by A/D and D/A conversion.

In Fig. 28.1 the original analog signal (a) is filtered by an anti-aliasing filter to remove any high-frequency components that may cause an effect known as aliasing (see Sec. 28.5). The signal is sampled and held and then converted into a digital signal (b). Next the DAC converts the digital signal back into an analog signal (c). Note that the output of the DAC is not as "smooth" as the original signal. A low-pass filter returns the analog signal back to its original form (plus phase shift introduced from the conversions)

after eliminating the higher order signal components caused by the conversion. This example illustrates the main differences between analog and digital signals. Whereas the analog signal in Fig. 28.1a is *continuous* and *infinite* valued, the digital signal in (b) is *discrete* with respect to time and *quantized.* The term *continuous-time signal* refers to a signal whose response with respect to time is uninterrupted. Simply stated, the signal has a continuous value for the entire segment of time for which the signal exists. By referring to the analog signal as infinite valued, we mean that the signal can possess any value between the parameters of the system. For example, in Fig. 28.1a, if the peak amplitude of the sine wave was +1V, then the analog signal can be any value between –1 and 1 V (such as 0.4758393848 V). Of course, measuring all the values between –1 and 1 V would require a piece of laboratory equipment with infinite precision.

The digital signal, on the other hand, is discrete with respect to time. This means that the signal is defined for only certain or discrete periods of time. A signal that is quantized can only have certain values (as opposed to an infinitely valued analog signal) for each discrete period. The signal illustrated in Fig. 28.1b illustrates these qualities.

28.2 Converting Analog Signals to Digital Signals

We have already established the differences between analog and digital signals. How is it possible to convert from an analog signal to a digital signal? An example will illustrate the process.

Where you live the temperature in the winter stays between 0° F and 50° F (Fig. 28.2a). Suppose you had a thermometer with only two readings, hot and cold, and you wanted to record the weather patterns and plot the results. The two quantization levels can be correlated with the actual temperature as follows:

If $0° F \leq T < 25° F$	Temperature is recorded as cold
If $25° F \leq T < 50° F$	Temperature is recorded as hot

You take a measurement every day at noon and plot the results after one week. From Fig. 28.2b, it is apparent that your discretized version of the weather is not an accurate representation of the actual weather.

Now suppose that you find another thermometer with four possible temperatures (hot, warm, cool, and cold) and you increase the number of readings to two per day. The result of this reading is seen in Fig. 28.3a. The quantization levels represent four equal bands of temperature as seen below:

If $0° F \leq T < 12.5° F$	Temperature is recorded as cold
If $12.5°F \leq T < 25° F$	Temperature is recorded as cool
If $25° F \leq T < 37.5° F$	Temperature is recorded as warm
If $37.5° F \leq T < 50° F$	Temperature is recorded as hot

Here, the digital version of the weather still looks nothing like the actual weather pattern, but the critical issues in digitizing an analog signal should be apparent. The actual weather pattern is the analog signal. It is continuous with respect to time, and its value can be between 0° F and 50° F (even 33.9638483920398439° F!). The accuracy of the

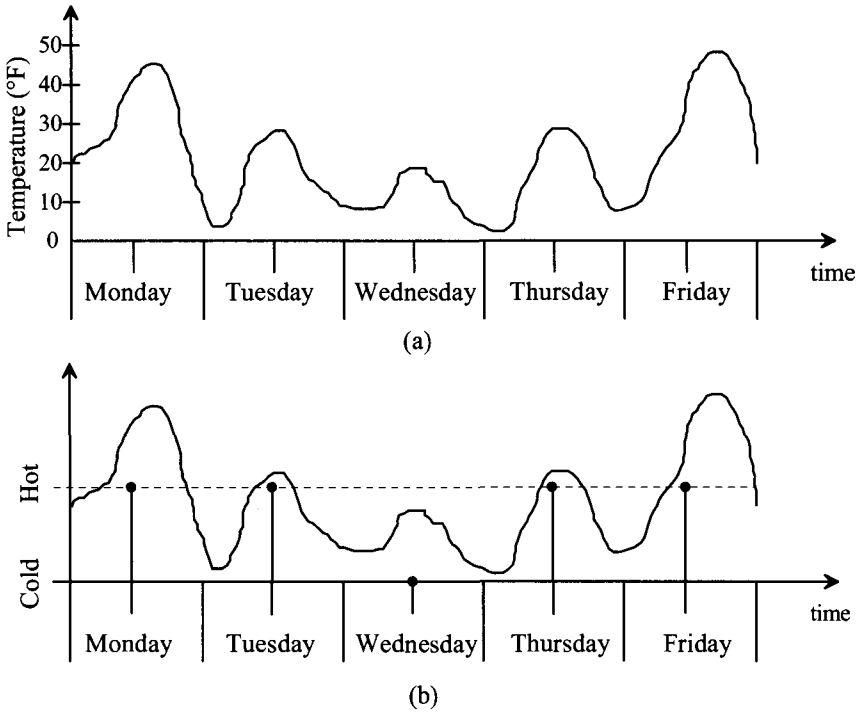

Figure 28.2 (a) An analog signal representing the temperature where you live and (b) a digital representation of the analog signal taking one sample per day with two quantization levels.

digitized signal is dependent on two things: the number of samples taken and the resolution, or number of quantization levels, of the converter. In our example, we need to increase both the number of samples and the resolution of thermometer.

Suppose that finally we obtain a thermometer with 25 temperature readings and that we take a reading eight times per day. Each of the 25 quantization levels now represents a 2° F band of temperature. From Fig. 28.3b, we can see that the digital version of the weather is approaching that of the actual analog signal. If we kept increasing both sampling time and resolution, the difference between the analog and the digital signals would become negligible. This brings up another critical issue: how many samples should one take in order to accurately represent the analog signal?

Suppose a sudden rainstorm swept through your town and caused a sharp decrease in temperature before returning to normal. If that storm had occurred between our sampling times, our experiment would not have shown the effects of the storm. Our sampling time was too slow to catch the change in the weather. If we had increased the number of samples, we would have recognized that something happened which caused the temperature to drop dramatically during that period.

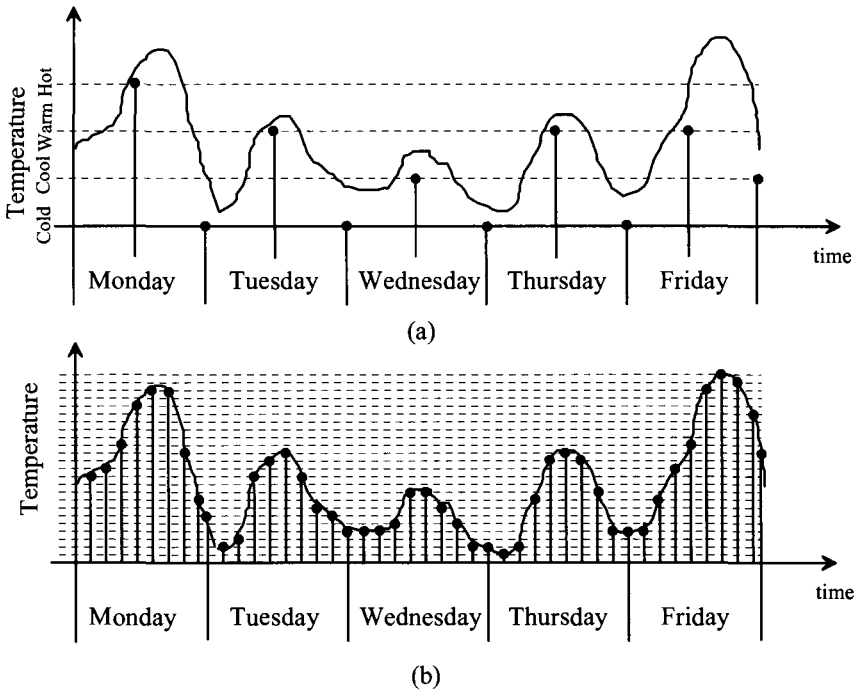

Figure 28.3 Digital representation of the temperature taking (a) two samples per day with four quantization levels and (b) nine samples per day with 25 quantization levels.

As it turns out, the *Nyquist Criterion* defines how fast the sampling rate needs to be to represent an analog signal accurately. This criterion requires that the sampling rate is at least two times the highest frequency contained in the analog signal. In our example, we need to know how quickly the weather can change and then take samples twice as fast as that value. The Nyquist Criterion can be described as

$$f_{sampling} = 2f_{MAX} \tag{28.1}$$

where $f_{sampling}$ is the sampling frequency required to accurately represent the analog signal and f_{MAX} is the highest frequency of the sampled signal.

How much resolution should we use to represent the analog signal accurately? There is no absolute criterion for this specification. Each application will have its own requirements. In our weather example, if we were only interested in following general trends, then the 25 quantization levels would more than suffice. However, if we were interested in keeping an accurate record of the temperature to within ±0.5° F, we would need to double the resolution to 50 quantization levels so that each quantization level would correspond to each degree ±0.5° F (Fig. 28.4).

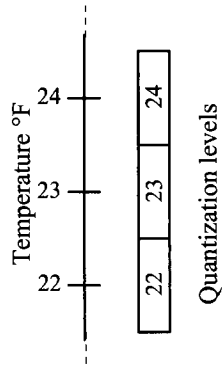

Figure 28.4 Quantization levels overlap actual temperature by ±½° F.

28.3 Sample-and-Hold (S/H) Characteristics

Sample-and-hold (S/H) circuits are critical in converting analog signals to digital signals. The behavior of the S/H is analogous to that of a camera. Its main function is to "take a picture" of the analog signal and hold its value until the ADC can process the information. It is important to characterize the S/H circuit when performing data conversion. Ignoring this component can result in serious error, for both speed and accuracy can be limited by the S/H. Ideally, the S/H circuit should have an output similar to that shown in Fig. 28.5a. Here, the analog signal is instantly captured and held until the next sampling period. However, a finite period of time is required for the sampling to occur. During the sampling period, the analog signal may continue to vary; thus, another type of circuit is called a track-and-hold, or T/H. Here, the analog signal is "tracked" during the time required to sample the signal, as seen in Fig. 28.5b. It can be seen that S/H circuits operate in both static (hold mode) and dynamic (sample mode) circumstances. Thus, characterization of the S/H will be discussed in the context of these two categories. Figure 28.6 presents a summary of the major errors associated with a S/H. A discussion of each error follows.

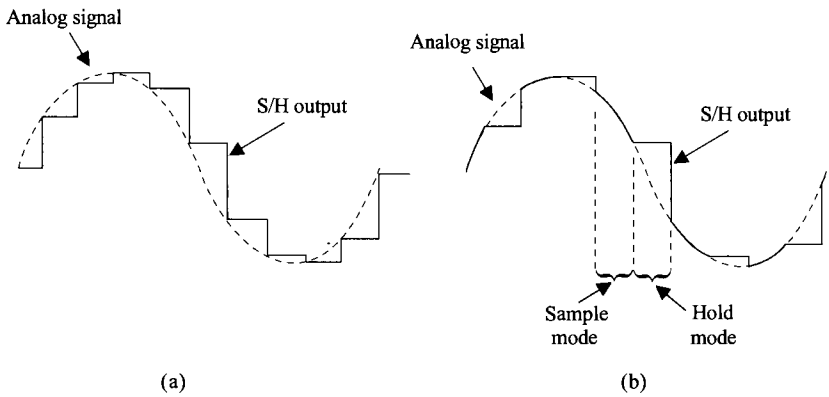

Figure 28.5 The output of (a) an ideal S/H circuit and (b) a track-and-hold (T/H).

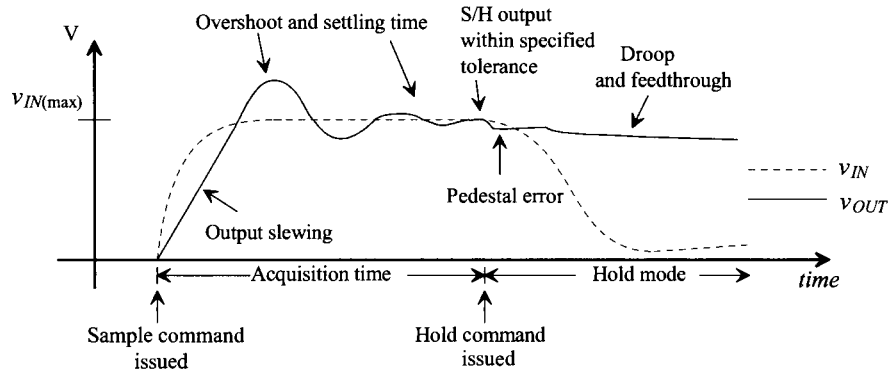

Figure 28.6 Typical errors associated wth an S/H.

Sample Mode

Once the sampling command has been issued, the time required for the S/H to track the analog signal to within a specified tolerance is known as the *acquisition time.* In the worst-case scenario, the analog signal would vary from zero volts to its maximum value, $v_{IN(max)}$. And the worst-case acquisition time would correspond to the time required for the output to transition from zero to $v_{IN(max)}$. Since most S/H circuits use amplifiers as buffers (as seen in Fig. 28.7), it should be obvious that the acquisition will be a function of the amplifier's own specifications. For example, notice that if the input changes very quickly, then the output of the T/H could be limited by the amplifier's slew rate. The amplifier's stability is also extremely critical. If the amplifier is not compensated correctly, and the phase margin is too small, then a large *overshoot* will occur. A large overshoot requires a longer *settling time* for the S/H to settle within the specified tolerance. The error tolerance at the output of the S/H also depends on the amplifier's *offset, gain error* (ideally, the S/H should have a gain of 1) and *linearity* (the gain of the S/H should not vary over the input voltage range).

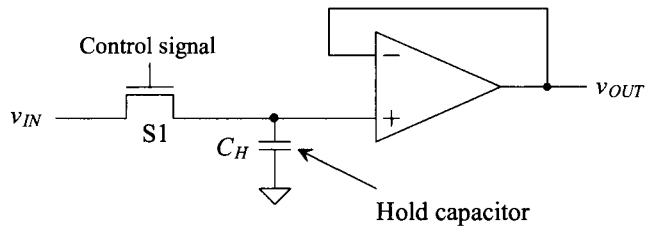

Figure 28.7 Track-and-hold circuit using an output buffer.

Hold Mode

Once the hold command is issued, the S/H faces other errors. Pedestal error occurs as a result of charge injection and clock feedthrough. Part of the charge built up in the channel of the switch is distributed onto the capacitor, thus slightly changing its voltage. Also, the clock couples onto the capacitor via overlap capacitance between the gate and the source or drain. Another error that occurs during the hold mode is called *droop*. This error is related to the leakage of current from the capacitor due to parasitic impedances and to the leakage through the reverse-biased diode formed by the drain of the switch. This diode leakage can be minimized by making the drain area as small as can be tolerated. Although the input impedance of the buffer amplifier is very large, the switch has a finite OFF impedance through which leakage can occur. Current can also leak through the substrate. The key to minimizing droop is increasing the value of the sampling capacitor. The trade-off, however, is increased time that's required to charge the capacitor to the value of the input signal.

Aperture Error

A transient effect that introduces error occurs between the sample and the hold modes. A finite amount of time, referred to as aperture time, is required to disconnect the capacitor from the analog input source. The aperture time actually varies slightly as a result of noise on the hold-control signal and the value of the input signal, since the switch will not turn off until the gate voltage becomes less than the value of the input voltage less one threshold voltage drop. This effect is called *aperture uncertainty* or *aperture jitter*. As a result, if a periodic signal were being sampled repeatedly at the same points, slight variations in the hold value would result, thus creating *sampling error*. Figure 28.8 illustrates this effect. Note that the amount of aperture error is directly related to the frequency of the signal and that the worst-case aperture error occurs at the zero crossing, where *dV/dt* is the greatest. This assumes that the S/H circuit is capable of sampling both positive and negative voltages (bipolar). The amount of error that can be tolerated is directly related to the resolution of the conversion. Aperture error will be discussed again in Sec. 28.5 as it relates to the error in an ADC.

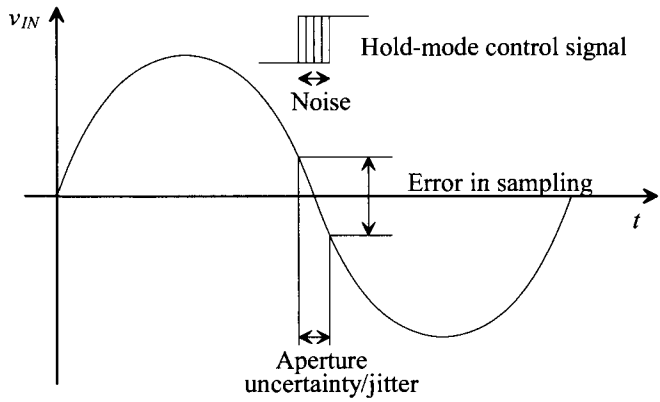

Figure 28.8 Aperture error.

Example 28.1
Find the maximum sampling error for a S/H circuit that is sampling a sinusoidal input signal that could be described as

$$v_{IN} = A \sin 2\pi ft$$

where A is 2 V and f = 100 kHz. Assume that the aperture uncertainty is equal to 0.5 ns.

The sampling error due to the aperture uncertainty can be thought of as a slew rate such that

$$\frac{dV}{dt} = \frac{d}{dt} A \sin 2\pi ft = 2\pi fA \cos 2\pi ft$$

with maximum slewing occurring when the cosine term is equal to 1. Therefore,

$$\frac{dV}{dt}(max) = 2\pi fA = (2\pi \cdot 100 \text{ kHz})(2 \text{ V})$$

and the maximum sampling error is

$$Maximum\ Sampling\ Error = dV(max) \quad \text{or}$$

$$(0.5 \times 10^{-9} \text{ s})(2\pi \cdot 100 \text{ kHz})(2 \text{ V}) = 0.628 \text{ mV} \ \blacksquare$$

28.4 Digital-to-Analog Converter (DAC) Specifications

Probably the most popular digital-to-analog converter application is converting stored digital audio and/or video signals. For example, stored digital information in MP3 format can be converted into music via a high-precision DAC. Many characteristics define a DAC's performance. Each characteristic will be discussed before we look at the basic architectures in Ch. 29. This "top-down" approach allows a smoother transition from the data converter characteristics to the actual architectures, since most data converters have similar performance limitations. A discussion of some of the basic definitions associated with DACs follows. It should be noted that DACs and ADCs can use either voltage or current as their analog signal. For purposes of describing specifications, it will be assumed that the analog signal is a voltage.

A block diagram of a DAC can be seen in Fig. 28.9. Here an N-bit digital word is mapped into a single analog voltage. Typically, the output of the DAC is a voltage that is some fraction of a reference voltage (or current), such that

$$v_{OUT} = FV_{REF} \tag{28.2}$$

where v_{OUT} is the analog voltage output, V_{REF} is the reference voltage, and F is the fraction defined by the input word, D, that is N bits wide. The number of input combinations represented by the input word D is related to the number of bits in the word by

$$\text{Number of input combinations} = 2^N \tag{28.3}$$

A 4-bit DAC has a total of 2^4 or 16 total input values. A converter with 4-bit resolution must be able to map a change in the analog output, which is equal to 1 part in 16. The maximum analog output voltage for any DAC is limited by the value of V_{REF}. If the input is an N-bit word, then the value of the fraction, F, can be determined by,

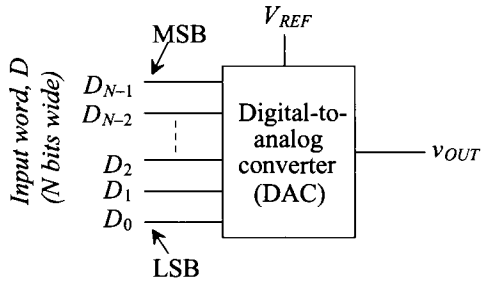

Figure 28.9 Block diagram of the digital-to-analog converter.

$$F = \frac{D}{2^N} \tag{28.4}$$

Therefore, if a 3-bit DAC is being used, the input, D, is $100 = 4_{10}$, and V_{REF} is 5 V, then the value of F is

$$F = \frac{100}{2^3} = \frac{4}{8} \tag{28.5}$$

and the analog voltage that appears at the output becomes,

$$v_{OUT} = \frac{4}{8}(5) = 2.5 \text{ V} \tag{28.6}$$

By plotting the input word, D, versus v_{OUT} as D is incremented from 000 to 111, the *transfer curve* seen in Fig. 28.10 would be generated. The y-axis has been normalized to V_{REF}; therefore, the graduated marks also represent F by Eq. (28.2). Some important characteristics need to be discussed here. First, notice that the transfer curve is not continuous. Since the input is a digital signal, which is inherently discrete, the input signal can only have eight values that must correspondingly produce eight output voltages. If a straight line connected each of the output values, the slope of the line would ideally be one increment/input code value. Also note that the maximum value of the output is 7/8. Since the case where $D = 000$ has to result in an analog voltage of 0 V, and a 3-bit DAC has eight possible analog output voltages, then the analog output will increase from 0 V to only 7/8 V_{REF}.

Again, using Eq. (28.2), this means that the maximum analog output that can be generated by the 3-bit DAC is

$$v_{OUT(\max)} = \frac{7}{8} \cdot V_{REF} \tag{28.7}$$

This maximum analog output voltage that can be generated is known as *full-scale voltage*, V_{FS}, and can be generalized to any N-bit DAC as

$$V_{FS} = \frac{2^N - 1}{2^N} \cdot V_{REF} \tag{28.8}$$

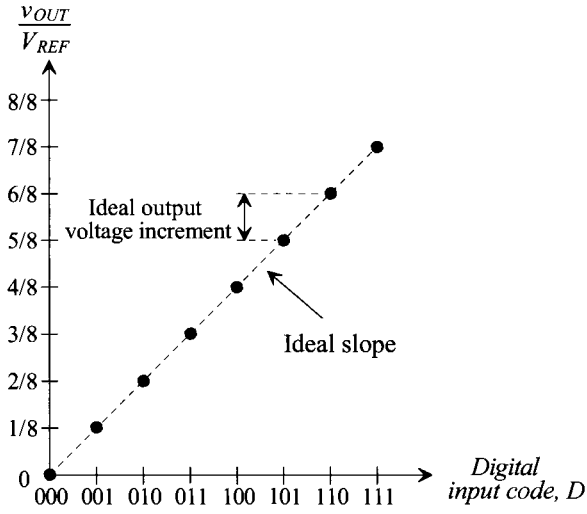

Figure 28.10 Ideal transfer curve for a 3-bit DAC.

The *least significant bit* (*LSB*) refers to the rightmost bit in the digital input word. The LSB defines the smallest possible change in the analog output voltage. The LSB will always be denoted as D_0. One LSB can be defined as

$$1\ LSB = \frac{V_{REF}}{2^N} \tag{28.9}$$

In the previous case of the 3-bit DAC, 1 LSB = 5/8 V, or 0.625 V. Generating an output in multiples of 0.625 V may not seem difficult, but as the number of bits increases, the voltage value of one LSB decreases for a fixed value of V_{REF}.

The *most significant bit* (*MSB*) refers to the leftmost bit of the digital word, *D*. In the previous example, $D = 100$ or $D_2D_1D_0$, with D_2 being the MSB. Generalizing to the *N*-bit DAC, the MSB would be denoted as D_{N-1}. (Since the LSB is denoted as bit 0, the MSB is denoted as *N*-1.) Note that when discussing DACs, the MSB causes the output to change by 1/2 V_{REF}.

When discussing data converters, the term *resolution* describes the smallest change in the analog output with respect to the value of the reference voltage, V_{REF}. This is slightly different from the definition of LSB in that resolution is typically given in terms of bits and represents the *number of unique output voltage levels*, i.e., 2^N.

Example 28.2

Find the resolution for a DAC if the output voltage is desired to change in 1 mV increments while using a reference voltage of 5 V.

The DAC must resolve

$$\frac{1\text{ mV}}{5\text{ V}} = 0.0002 \text{ or } 0.02\% \text{ adjustability}$$

Therefore, the *accuracy* required for 1 LSB change over a range of V_{REF} is

$$\frac{1\,LSB}{V_{REF}} = \frac{1}{2^N} = 0.0002 \tag{28.10}$$

and solving N for the resolution yields

$$N = \text{Log}_2(\frac{5\text{ V}}{1\text{ mV}}) = 12.29 \text{ bits}$$

which means that a 13-bit DAC will be needed to produce the accuracy capable of generating 1 mV changes in the output using a 5 V reference. ■

Example 28.3

Find the number of input combinations, values for 1 LSB, the percentage accuracy, and the full-scale voltage generated for a 3-bit, 8-bit, and 16-bit DAC, assuming that V_{REF} = 5 V.

Using Eqs. (28.3), (28.8), (28.9), and (28.10), we can generate the following information:

Resolution	Input combinations	1 LSB	% accuracy	V_{FS}
3	8	0.625 V	12.5	4.375 V
8	256	19.5 mV	0.391	4.985 V
16	65,536	76.29 μV	0.00153	4.9999 V

The value of 1 LSB for an 8-bit converter is 19.5 mV, while 1 LSB for a 16-bit converter is 76.3 μV (a factor of 256)! Increasing the resolution by 1 bit increases the accuracy by a factor of 2. The precision required to map the analog signal at high resolutions is very difficult to achieve. We will examine some of these issues as we examine the limitations of the data converter in Ch. 29.

Note that a data converter may have a resolution of 8 bits, where an LSB is 19.5 mV as above, while having a much higher accuracy. For example, we could require the 8-bit data converter above to have an accuracy of 0.1%. The higher accuracy results in a more ideal (linear) DAC. A typical specification for DAC accuracy is ±½ LSB for reasons discussed below. ■

Differential Nonlinearity

As seen in the ideal DAC in Fig. 28.10, each adjacent output increment should be exactly one-eighth. Since the y-axis is normalized, the values for the increment heights will be unitless. However, the increment heights can be easily converted to volts by multiplying the height by V_{REF}. This corresponds to the ideal increment corresponding to 0.625 V = 1 LSB (assuming V_{REF} = 5 V).

Nonideal components cause the analog increments to differ from their ideal values. The difference between the ideal and nonideal values is known as *differential nonlinearity*, or *DNL* and is defined as

DNL_n = Actual increment height of transition n – Ideal increment height (28.11)

where n is the number corresponding to the digital input transition. The DNL specification measures how well a DAC can generate uniform analog LSB multiples at its output.

Example 28.4

Determine the DNL for the 3-bit nonideal DAC whose transfer curve is shown in Fig. 28.11. Assume that $V_{REF} = 5$ V.

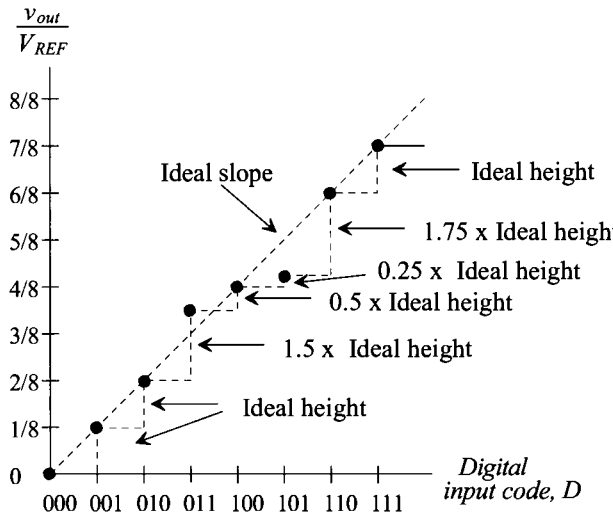

Figure 28.11 Example of differential nonlinearity for a 3-bit DAC.

The actual increment heights are labeled with respect to the ideal increment height, which is 1 LSB, or 1/8 of $\frac{v_{OUT}}{V_{REF}}$. Notice that there is no increment corresponding to 000, since it is desirable to have zero output voltage with a digital input code of 000. The increment height corresponding to 001, however, is equal to the corresponding height of the ideal case seen in Fig. 28.10; therefore, $DNL_1 = 0$. Similarly, DNL_2 is also zero since the increment associated with the transition at 010 is equal to the ideal height. Notice that the 011 increment, however, is not equal to the ideal curve but is 3/16, or 1.5 times the ideal height.

$$DNL_3 = 1.5 \text{ LSB} - 1 \text{ LSB} = 0.5 \text{ LSB}$$

Since we have already determined in Eq. (28.9) that for a 3-bit DAC, 1 LSB = 0.625 V, we can convert the DNL_3 to volts as well. Therefore, $DNL_3 = 0.5$ LSB = 0.3125 V. However, it is popular to refer to DNL in terms of LSBs. The remainder of the digital output codes can be characterized as follows:

$$DNL_4 = 0.5 \text{ LSB} - 1 \text{ LSB} = -0.5 \text{ LSB}$$

$$DNL_5 = 0.25 \text{ LSB} - 1 \text{ LSB} = -0.75 \text{ LSB}$$

$$DNL_6 = 1.75 \text{ LSB} - 1 \text{ LSB} = 0.75 \text{ LSB}$$

$$DNL_7 = 1 \text{ LSB} - 1 \text{ LSB} = 0$$

If we were to plot the value of DNL (in LSBs) versus the input digital code, Fig. 28.12 would result. The DNL for the entire converter used in this illustration is ±0.75 LSB since the overall error of the DAC is defined by its worst-case DNL. ■

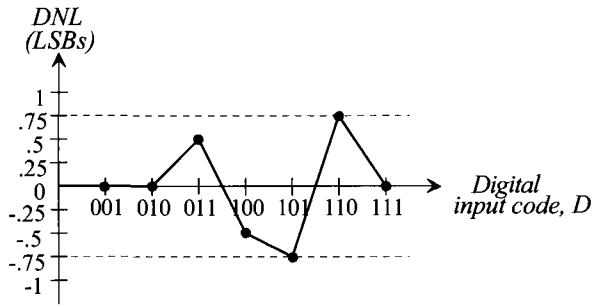

Figure 28.12 DNL curve for the nonideal 3-bit DAC.

Generally, a DAC will have less than ±½ LSB of DNL if it is to be *N*-bit accurate. A 5-bit DAC with 0.75 LSBs of DNL actually has the resolution of a 4-bit DAC. If the DNL for a DAC is less than –1 LSBs, then the DAC is said to be *nonmonotonic*, which means that the analog output voltage does not always increase as the digital input code is incremented. A DAC should always exhibit *monotonicity* if it is to function without error.

Integral Nonlinearity

Another important static characteristic of DACs is called *integral nonlinearity* (*INL*). Defined as the difference between the data converter output values and a reference straight line drawn through the first and last output values, INL defines the linearity of the overall transfer curve and can be described as

$$\text{INL}_n = \text{Output value for input code } n - \text{Output value of the reference line at that point} \tag{28.12}$$

An illustration of this measurement is presented in Fig. 28.13. It is assumed that all other errors due to offset and gain (these will be discussed shortly) are zero. An example follows shortly.

It is common practice to assume that a converter with *N*-bit resolution will have less than ±½ LSB of DNL and INL. The term, ½ *LSB*, is a common term that typically denotes the maximum error of a data converter (both DACs and ADCs). For example, a 13-bit DAC having greater than ±½ LSB of DNL or INL actually has the resolution of a 12-bit DAC. The value of ½ LSB in volts is simply

$$0.5\,LSB = \frac{V_{REF}}{2^{N+1}} \tag{28.13}$$

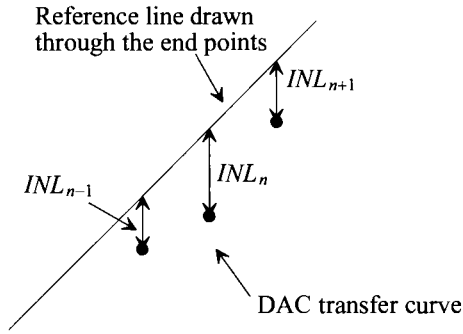

Figure 28.13 Measuring the INL for a DAC transfer curve.

Example 28.5

Determine the INL for the nonideal 3-bit DAC shown in Fig. 28.14. Assume that $V_{REF} = 5$ V.

First, a reference line is drawn through the first and last output values. The INL is zero for every code in which the output value lies on the reference line; therefore, $INL_2 = INL_4 = INL_6 = INL_7 = 0$. Only outputs corresponding to 001, 011, and 101 do not lie on the reference. Both the 001 and the 011 transitions occur ½ LSB higher than the straight-line values; therefore, $INL_1 = INL_3 = 0.5$ LSB. By the same reasoning, $INL_5 = -0.75$ LSB. Therefore, the INL for the DAC is considered to be its worst-case INL of +0.5 LSB and –0.75 LSB. The INL plot for the nonideal 3-bit DAC can be seen in Fig. 28.15. ■

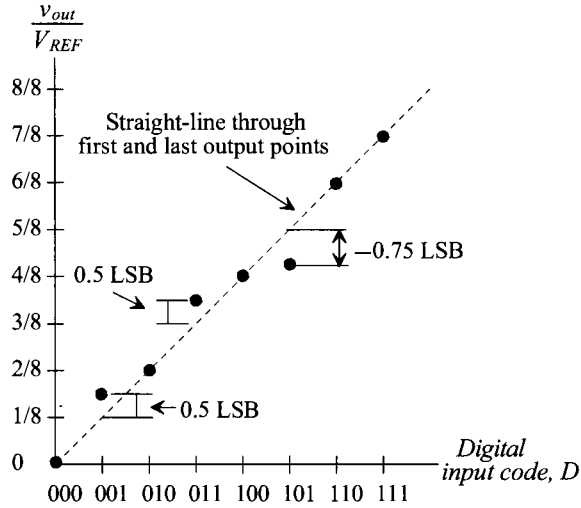

Figure 28.14 Example of integral nonlinearity for a DAC.

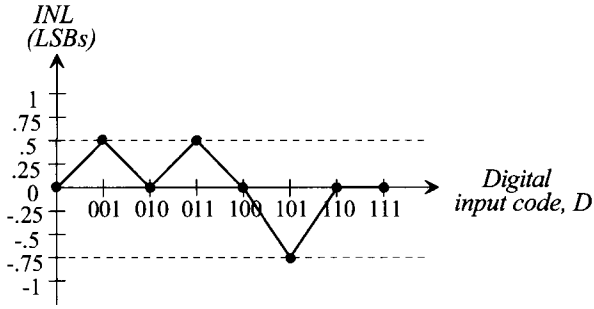

Figure 28.15 INL curve for the nonideal 3-bit DAC.

It should be noted that other methods are used to determine INL. One method compares the output values to the ideal reference line, regardless of the positions of the first and last output values. If the DAC has an offset voltage or gain error, this will be included in the INL determination. Usually, the offset and gain errors are determined as separate specifications.

Another method, described as the "best-fit" method, attempts to minimize the INL by constructing the reference line so that it passes as closely as possible to a majority of the output values. Although this method does minimize the INL error, it is a rather subjective method that is not as widely used as drawing the reference line through the first and last output values.

Offset

The analog output should be 0 V for $D = 0$. However, an offset exists if the analog output voltage is not equal to zero. This can be seen as a shift in the transfer curve as illustrated in Fig. 28.16. This specification is similar to the offset voltage for an operational amplifier except that it is not referred to the input.

Gain Error

A gain error exists if the slope of the best-fit line through the transfer curve is different from the slope of the best-fit line for the ideal case. For the DAC illustrated in Fig. 28.17, the gain error becomes

$$\text{Gain error} = \text{Ideal slope} - \text{Actual slope} \tag{28.14}$$

Latency

This specification defines the total time from the moment that the input digital word changes to the time the analog output value has settled to within a specified tolerance. Latency should not be confused with settling time, since latency includes the delay required to map the digital word to an analog value plus the settling time. It should be noted that settling time considerations are just as important for a DAC as they are for a S/H or an operational amplifier.

Signal-to-Noise Ratio (SNR)

Signal-to-noise (*SNR*) is defined as the ratio of the signal power to the noise at the analog output. In amplifier applications, this specification is typically measured using a sinewave

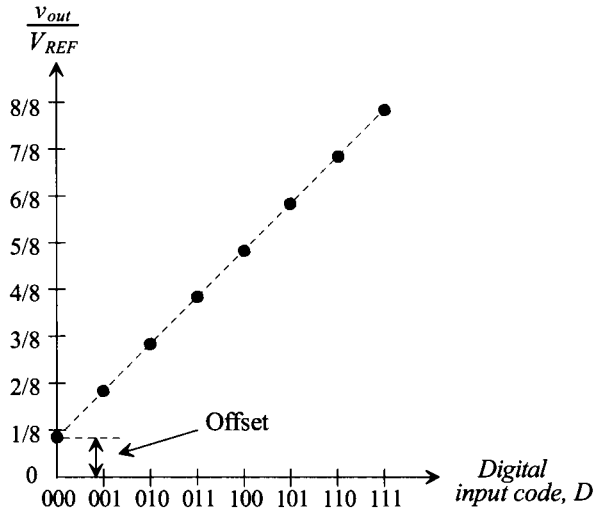

Figure 28.16 Illustration of offset error for a 3-bit DAC.

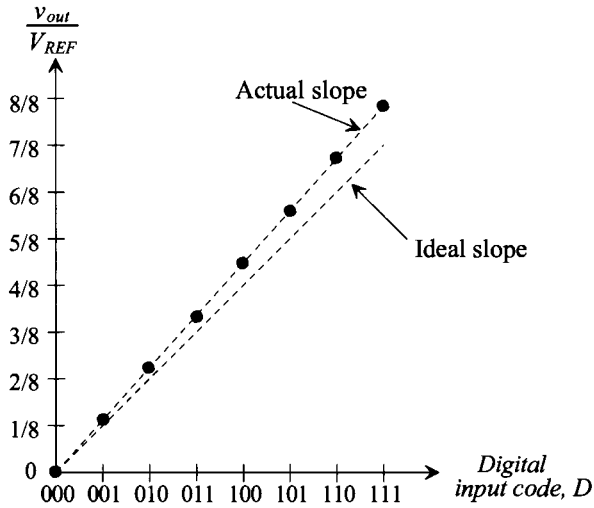

Figure 28.17 Illustration of gain error for a 3-bit DAC.

input. For the DAC, a "digital" sinewave is generated through instrumentation or through an A/D. The *SNR* can reveal the true resolution of a data converter as the effective number of bits can be quantified mathematically. A detailed derivation of the *SNR* is presented in Sec. 28.6, on the discussion of ADC specifications.

Dynamic Range

Dynamic range is defined as the ratio of the largest output signal over the smallest output signal. For both DACs and ADCs, the dynamic range is related to the resolution of the converter. For example, an N-bit DAC can produce a maximum output of $2^N - 1$ multiples of LSBs and a minimum value of 1 LSB. Therefore, the dynamic range in decibels is simply

$$DR = 20\text{Log}\left(\frac{2^N - 1}{1}\right) \approx 6.02 \cdot N \text{ dB} \tag{28.15}$$

A 16-bit data converter has a dynamic range of 96.33 dB.

28.5 Analog-to-Digital Converter (ADC) Specifications

Many of the specifications that describe the ADC are similar to those that describe the DAC. However, there are subtle differences. Since the DAC is converting a discrete signal into an analog representation that is also limited by the resolution of the converter, a fixed number of inputs and outputs are generated. However, with the ADC, the input is an analog signal with an infinite number of values, which then has to be quantized into an N-bit digital word (Fig. 28.18). This process is much more difficult than the digital-to-analog process. In fact, many ADC architectures use a DAC as a critical component.

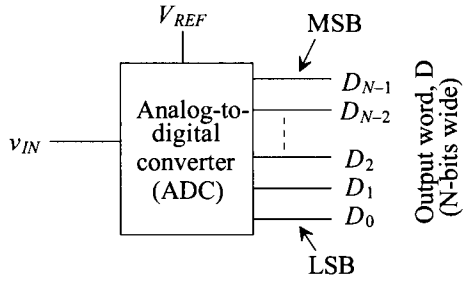

Figure 28.18 Block diagram of the analog-to-digital converter.

For example, in the previous discussion of DACs, it was determined that for a 16-bit DAC, the converter would need to generate output voltages in multiples of 76 μV. However, for the ADC, the converter needs to resolve differences in the analog signal of 76 μV. This means that the ADC must be able to detect changes in the input signal on the order of 1 part in 65,536! In contrast, the DAC had a finite number of input combinations (2^N). The ADC, however, has to "quantize" the infinite-valued analog signal into many segments so that

$$\text{Number of quantization levels} = 2^N \tag{28.16}$$

This distinction is subtle but must be recognized to understand the differences between the two types of conversion.

Examine Fig. 28.19a. The digital output, D, of an ideal, 3-bit ADC is plotted versus the analog input, v_{IN}. Note the difference in the transfer curve for the ADC versus the DAC (Fig. 28.10). The y-axis is now the digital output, and the x-axis has been normalized to V_{REF}. Since the input signal is a continuous signal and the output is discrete, the transfer curve of the ADC resembles that of a staircase. Another fact to observe is that the 2^N quantization levels correspond to the digital output codes 0 to 7. Thus, the maximum output of the ADC will be 111 ($2^N - 1$), corresponding to the value for which $\frac{v_{IN}}{V_{REF}} \geq \frac{7}{8}$. Figure 28.19b corresponds to the error caused by the quantization.

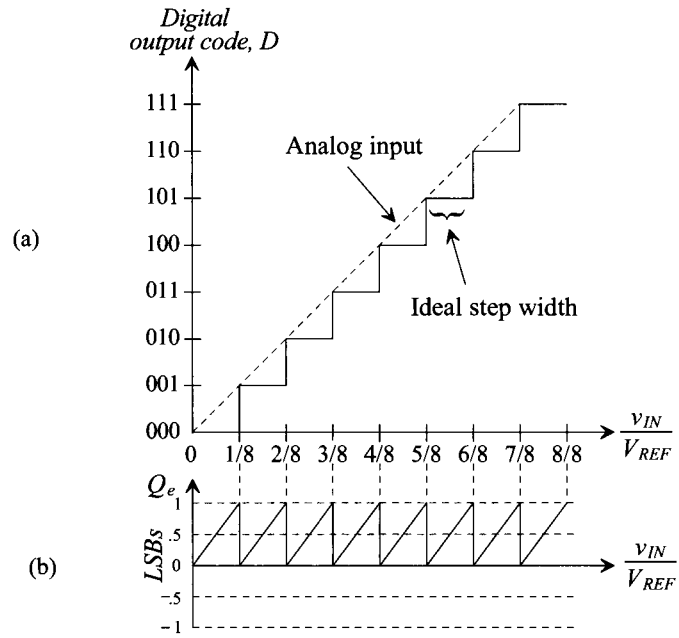

Figure 28.19 (a) Transfer curve for an ideal ADC and (b) its corresponding quantization error.

The value of 1 LSB for this ADC can be calculated using Eq. (28.9) and is the ideal step width (1/8) in Fig. 28.19 (versus the height for the DAC) multiplied by V_{REF}. Therefore, assuming that $V_{REF} = 5$ V,

$$1 \text{ LSB} = 0.625 \text{ V} \tag{28.17}$$

Quantization Error

Since the analog input is an infinite valued quantity and the output is a discrete value, an error will be produced as a result of the quantization. This error, known as *quantization error*, Q_e, is defined as the difference between the actual analog input and the value of the output (staircase) given in voltage. It is calculated as

$$Q_e = v_{IN} - V_{staircase} \tag{28.18}$$

where the value of the staircase output, $V_{staircase}$, can be calculated by

$$V_{staircase} = D \cdot \frac{V_{REF}}{2^N} = D \cdot V_{LSB} \tag{28.19}$$

where D is the value of the digital output code and V_{LSB} is the value of 1 LSB in volts, in this case 0.625 V. We can also easily convert the value of Q_e in units of LSBs. In Fig. 28.19a, Q_e can be generated by subtracting the value of the staircase from the dashed line. The result can be seen in Fig. 28.19b. A sawtooth waveform is formed centered about ½ LSBs. Ideally, the magnitude of Q_e will be no greater than one LSB and no less than 0. It would be advantageous if the quantization error were centered about zero so that the error would be at most ±½ LSBs (as opposed to +1LSB). This is easily achieved as seen in Fig. 28.20a and b. Here, the entire transfer curve is shifted to the left by ½ LSB, thus making the codes centered around the LSB increments on the x-axis. This drawing illustrates that at best, an ideal ADC will have quantization error of ±½ LSB.

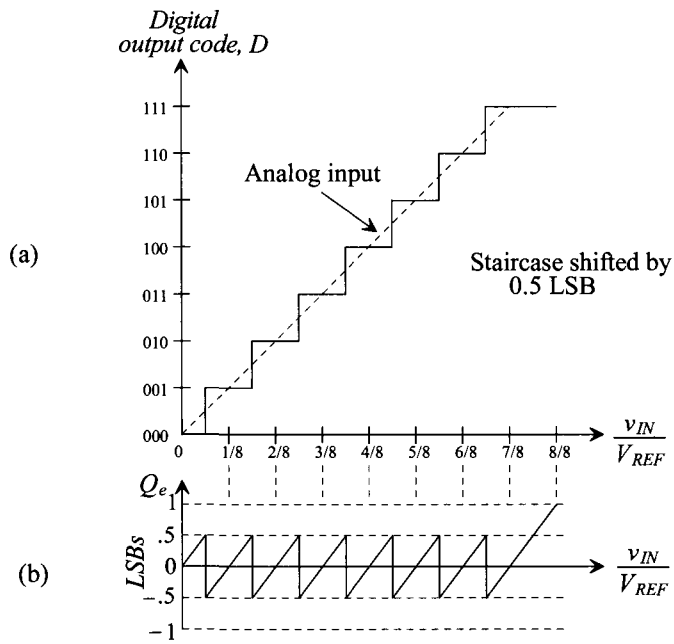

Figure 28.20 (a) Transfer curve for an ideal 3-bit ADC with (b) quantization error centered about zero.

In shifting this curve to the left, notice that the first code transition occurs when $\frac{v_{IN}}{V_{REF}} \geq \frac{1}{16}$. Therefore, the range of $\frac{v_{IN}}{V_{REF}}$ for the digital output corresponding to 000 is half as wide as the ideal step width. The last code transition occurs when $\frac{v_{IN}}{V_{REF}} \geq \frac{13}{16}$ (between 6/8 and 7/8). Note that the step width corresponding to this last code transition is 1.5 times larger than the ideal width and that the quantization error extends up to 1 LSB when $\frac{v_{IN}}{V_{REF}} = 1$. However, the converter would be considered to be out of range once $\frac{v_{IN}}{V_{REF}} \geq \frac{15}{16}$ (halfway between 7/8 and 8/8), so the problem is moot.

Differential Nonlinearity

Differential nonlinearity for an ADC is similar to that defined for a DAC. However, for the ADC, DNL is the difference between the actual code *width* of a nonideal converter and the ideal case. Figure 28.21 illustrates the transfer curve for a nonideal 3-bit ADC. The values for the DNL can be solved as follows:

$$\text{DNL} = \text{Actual step width} - \text{Ideal step width} \tag{28.20}$$

Since the step widths can be converted to either volts for LSBs, DNL can be defined using either units. The value of the ideal step is 1/8. Converting to volts, this becomes

$$V_{idealstepwidth} = \tfrac{1}{8} \cdot V_{REF} = 0.625\ V = 1\ \text{LSB} \tag{28.21}$$

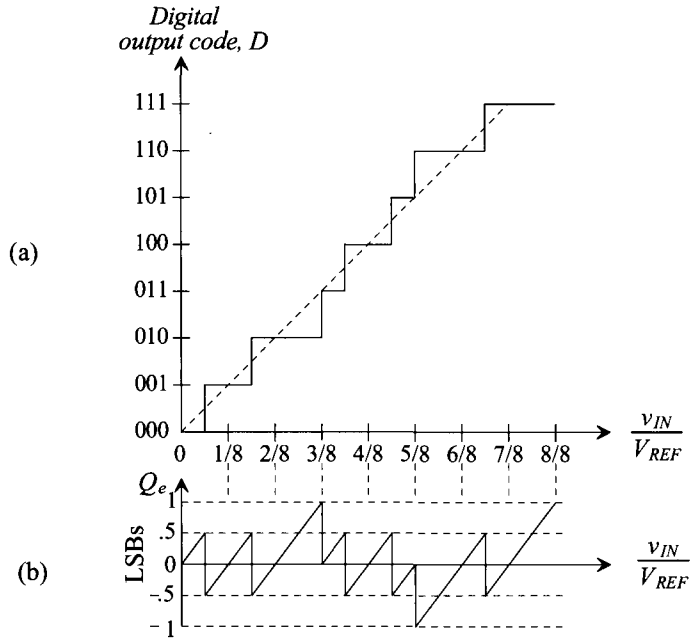

Figure 28.21 (a) Transfer curve for a nonideal 3-bit ADC used in Ex. 28.4 with (b) quantization error illustrating differential nonlinearity.

Example 28.6

Using Fig. 28.21a, calculate the differential nonlinearity of the 3-bit ADC. Assume that $V_{REF} = 5$ V. Draw the quantization error, Q_e, in units of LSBs.

The DNL of the converter can be calculated by examining the step width of each digital output code. Since the ideal step width of the 000 transition is ½ LSB, then $\text{DNL}_0 = 0$. Also note that the step widths associated with 001 and 100 are equal to 1 LSB; therefore, both DNL_1 and DNL_4 are zero. However, the remaining values code widths are not equal to the ideal value but can be calculated as

$DNL_2 = 1.5\ LSB - 1\ LSB = 0.5\ LSB$

$DNL_3 = 0.5\ LSB - 1\ LSB = -0.5\ LSB$

$DNL_5 = -0.5\ LSB$

$DNL_6 = 0.5\ LSB$

$DNL_7 = 0\ LSB$ (since the ideal step width is 1.5 LSB wide at this code transition)

The overall DNL for the converter used in this illustration is ±0.5 LSB. Note that the quantization error illustrated in Fig. 28.21b is directly related to the DNL. As DNL increases in either direction, the quantization error worsens. Each "tooth" in the quantization error waveform should ideally be the same size. ■

Missing Codes

It is of interest to note the consequences of having a DNL that is equal to –1 LSB. Figure 28.22 illustrates an ADC for which this is true. The total width of the step corresponding to 101 is completely missing; thus, the value of DNL_5 is –1 LSB. Any ADC possessing a DNL that is equal to –1 LSB is guaranteed to have a missing code. Notice that the step width corresponding to 010 is 2 LSBs and that the value for DNL_2 is +1 LSB. However, there is not a missing code corresponding to 011, since the step width of code 011 depends on the 100 transition. Therefore, an ADC having a DNL greater than +1 LSB is not guaranteed to have a missing code, though in all probability a missing code will occur.

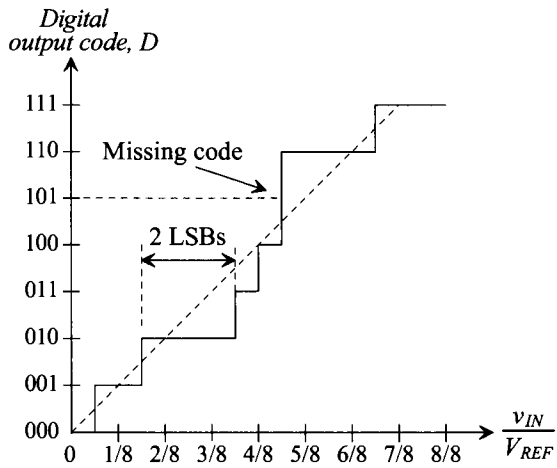

Figure 28.22 Transfer curve for a nonideal 3-bit ADC with a missing code.

Integral Nonlinearity

Integral nonlinearity (INL) is defined similarly to that for a DAC. Again, a "best-fit" straight line is drawn through the end points of the first and last code transition, with INL being defined as the difference between the data converter code transition points and the straight line with all other errors set to zero.

Example 28.7

Determine the INL for the ADC whose transfer curve is illustrated in Fig. 28.23a. Assume that V_{REF} = 5 V. Draw the quantization error, Q_e, in units of LSBs.

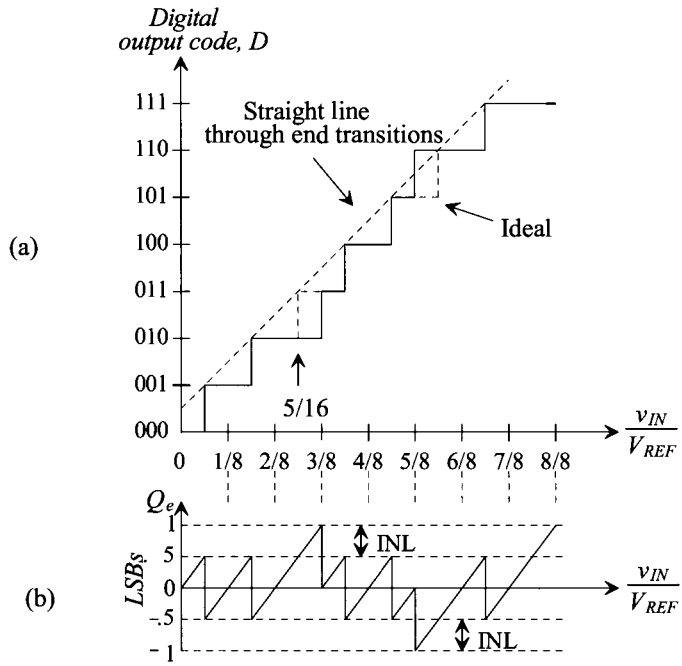

Figure 28.23 (a) Transfer curve of a nonideal 3-bit ADC and (b) its quantization error illustrating INL.

By inspection, it can be seen that all of the transition points occur on the best-fit line except for the transitions associated with code 011 and 110. Therefore,

$$INL_0 = INL_1 = INL_2 = INL_4 = INL_5 = INL_7 = 0$$

The INL corresponding to the remaining codes can be calculated as

$$INL_3 = 3/8 - 5/16 = 1/16 \text{ or } 0.5 \text{ LSB}$$

Similarly, INL_6 can be calculated in the same manner and is found to be –0.5 LSB. Thus, the overall INL for the converter is the maximum value of INL corresponding to ±0.5 LSB.

The INL can also be determined by inspecting the quantization error in Fig. 28.23b. Here, the INL will be the magnitude of the quantization error which lies outside the ±½ LSB band of Q_e. It can be seen that Q_e = 1 LSB, corresponding to the point at which INL = 0.5 LSB for digital output code 011, and that Q_e = –1 LSB at the output code corresponding to INL = –0.5 LSB for digital output code 110. ■

Offset and Gain Error

Offset and gain error are identical to the DAC case. *Offset error* occurs when there is a difference between the value of the first code transition and the ideal value of ½ LSBs. As seen in Fig. 28.24a, the offset error is a constant value. Note that the quantization error becomes ideal after the initial offset voltage is overcome. *Gain error* or *scale factor error*, seen in Fig. 28.24b, is the difference in the slope of a straight line drawn through the transfer characteristic and the slope of 1 of an ideal ADC. Causes of offset and gain error are discussed in Ch. 29, but it is important here to understand their overall effects on ADC transfer curves.

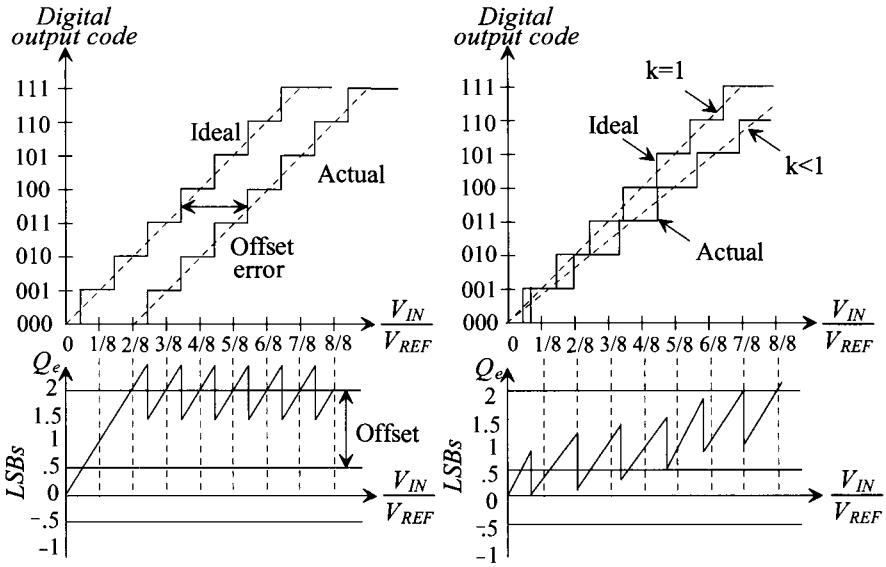

Figure 28.24 Transfer curve illustrating (a) offset error and (b) gain error.

So far, we have examined only the DC characteristics of an ADC. However, examining the dynamic aspects of the converter will lead to a whole new set of errors. Sampling is inherently a dynamic process since the accuracy of the sample is dependent on the speed of the analog signal. Many effects that occur during sampling limit the overall performance of the converter.

Aliasing

As mentioned earlier in the chapter, the Nyquist Criterion requires that a signal be sampled at least two times the highest frequency contained in the signal. What would happen if this criterion were ignored and the sampling rate was actually less than that amount? A phenomenon known as aliasing would occur.

Examine Fig. 28.25. Here, an analog signal is being sampled at a rate slower than the Nyquist Criterion requires. As a result, it appears that a totally different signal (see

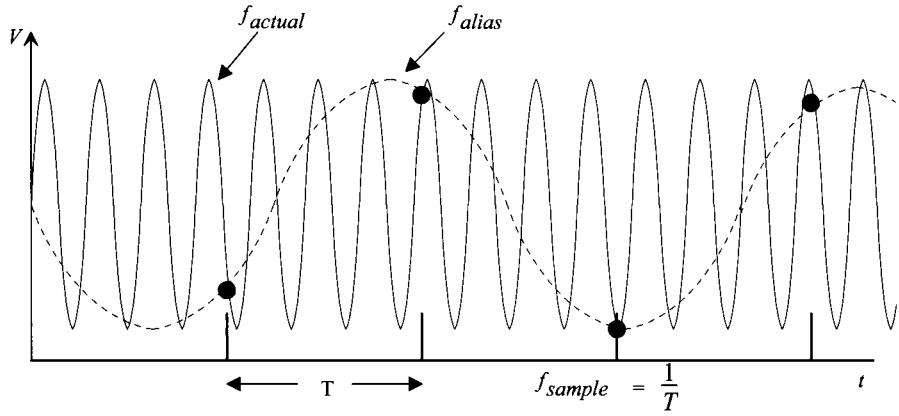

Figure 28.25 Aliasing caused by undersampling.

example dashed line) is being sampled. The different frequency signal is an "alias" of the original signal, and its frequency can be calculated using

$$f_{alias} = f_{actual} + k f_{sample} \quad (k = \ldots -2, -1, 0, 1, 2, 3\ldots) \tag{28.22}$$

where f_{actual} is the frequency of the analog signal, f_{sample} is the sampling frequency, and f_{alias} is the frequency of the alias signal.

Aliasing can be eliminated by both sampling at higher frequencies and by filtering the analog signal before sampling and removing any frequencies that are greater than one-half the sampling frequency. It is good practice to filter the analog signal before sampling to eliminate any unknown higher order frequency components or noise that could result in aliasing.

A frequency domain analysis may further illustrate the concepts of aliasing. Figure 28.26 shows the analog signal, the *sampling function* (represented by a unit impulse train) and the resulting sampled signal in both the time and frequency domains. The analog signal in Fig. 28.26a is represented as a simple band-limited signal with center frequency, f_o. This simply means that the signal is contained within the frequency range shown. In Fig. 28.26b, the sampling function is shown in both the time and frequency domain. The sampling function simply represents the action of sampling at discrete points in time. The frequency domain version of the sampling function is similar to its time domain counterpart, except that the x-axis is now represented as $f = 1/T$. Since each of the impulses has a value of 1, the resulting sampled signal shown in Fig. 28.26c is the impulse function multiplied by the amplitude of the analog signal at each discrete point in time. Remembering that multiplication in the time domain is equivalent to convolution in the frequency domain, we note that the frequency domain representation of the sampled signal reveals that the overall signal consists of multiple versions of the band-limited signal at multiples of the sampling frequency.

Note in Fig. 28.26b that as the sampling time increases, the sampling frequency decreases and the impulses in the frequency domain become more closely spaced. This results in 28.26d, which illustrates the aliasing as the multiple versions of the

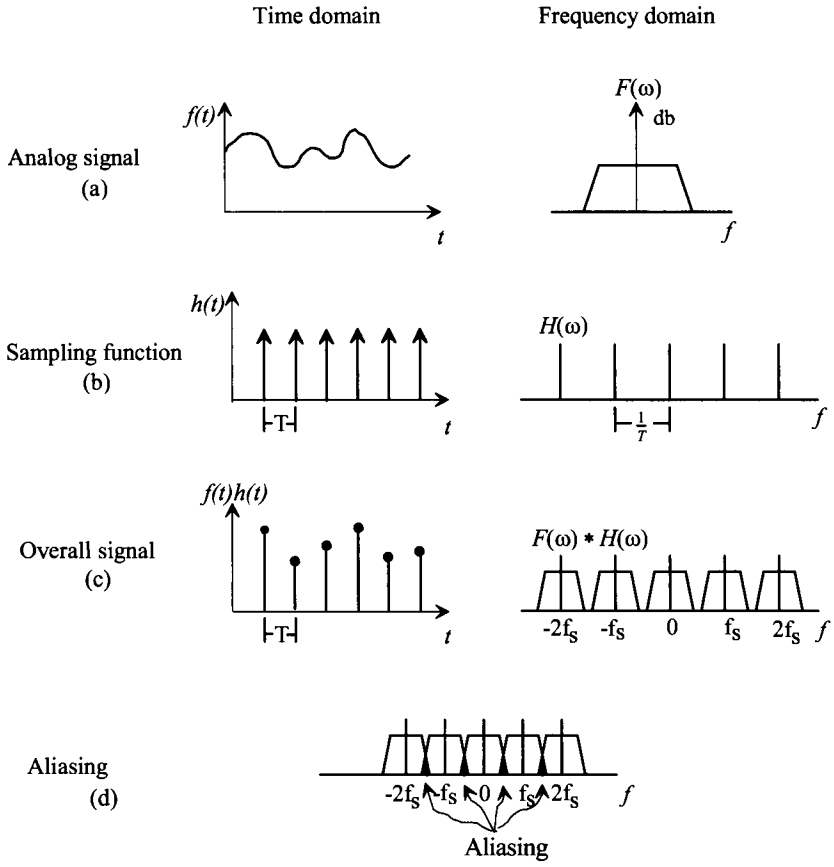

Figure 28.26 Illustration of aliasing in the time and frequency domain. (a) The analog signal; (b) the sampling function; (c) the overall signal; and (d) aliasing in the frequency domain.

band-limited signal begin to overlap. One could also filter the signal "post-sampling" and eliminate the frequencies for which overlap occurs. The point at which the spectra overlap is called the *folding frequency*.

As mentioned earlier, the solutions to aliasing are higher sampling frequency and filtering. Focusing on just one of the solutions may worsen the situation for several reasons. Some noise signals are wide band, which means that they have a large bandwidth. Attempting to increase only the sampling frequency to eliminate the aliasing effects of the noise would be a practical impossibility, not to mention a costly one. However, simply filtering the input signal and the sampled signal adds delays to the overall conversion and increases the expense of the circuit. It is best to use a combination of the two to minimize the problem most efficiently.

Signal-to-Noise Ratio

Signal-to-noise (*SNR*) ratios of ADCs represent the value of the largest RMS input signal into the converter over the RMS value of the noise. Typically given in dB, the expression for *SNR* is

$$SNR = 20\text{Log}\left(\frac{v_{in(max)}}{v_{noise}}\right) \tag{28.23}$$

If it is assumed that the input signal is a sinewave with a peak-to-peak value equal to the full-scale reference voltage of the converter, then the RMS value for $v_{in(max)}$ becomes

$$v_{in(max)} = \frac{V_{REF}}{2\sqrt{2}} = \frac{2^N(V_{LSB})}{2\sqrt{2}} \tag{28.24}$$

where V_{LSB} is the voltage value of 1 LSB. The value of the noise (if the data converter is considered to be ideal) will be equivalent to the RMS value of the error signal, Q_e (in volts), shown in Fig. 28.20b. The RMS value of Q_e can be calculated to be

$$Q_{e,RMS} = \left[\frac{1}{V_{LSB}}\int_{-0.5V_{LSB}}^{0.5V_{LSB}} (V_{LSB})^2 dV_{LSB}\right]^{0.5} = \frac{V_{LSB}}{\sqrt{12}} \tag{28.25}$$

Therefore, the *SNR* for the ideal ADC will be the ratio of these two RMS values,

$$SNR = 20\cdot\text{Log}\frac{\frac{2^N(V_{LSB})}{2\sqrt{2}}}{Q_{e,RMS}} \tag{28.26}$$

which can be written in terms of N as simply

$$SNR = 20N\text{Log}(2) + 20\text{Log}\sqrt{12} - 20\text{Log}(2\sqrt{2}) = 6.02N + 1.76 \tag{28.27}$$

Equation (28.27) is an important one relating *SNR* to the resolution of the ADC. For 16-bit data conversion, one must design a circuit that will have an *SNR* of (6.02)(16) + 1.76 = 98.08 dB! Equation (28.27) can also be used in calculating the *signal-to-noise plus distortion ratio*, also known as *SNDR*. Since the output data is digital, we cannot use a spectrum analyzer to calculate this ratio but must instead use a *Discrete Fourier Transform* (*DFT*) and examine the data in the digital domain.

Another useful application of Eq. (28.27) is the determination of effective number of bits given a system with a known *SNR* or *SNDR*. For example, if a 16-bit ADC yielded an SNDR of 88 dB, then the effective resolution of the converter would be

$$N = \frac{88 - 1.76}{6.02} = 14.32 \text{ bits} \tag{28.28}$$

and the ADC would be producing the resolution equivalent to that of a 14-bit converter.

Aperture Error

The aperture error described in Sec. 28.3 (S/H) should be related to the errors associated with the ADC. In the previous discussion, the aperture error resulted in sampling error (Fig. 28.8). However, now that ADC characteristics have been discussed, we can relate the sampling error to the ADC. Since we know that the maximum errors associated with an ADC are related to ½ LSB, we can assume that the maximum sampling error associated with the aperture uncertainty can be no larger than ½ LSB.

Example 28.8
Find the maximum resolution of an ADC which can use the S/H described in Ex. 28.1 while maintaining a sampling error less than ½ LSB.

Since it was determined that the maximum sampling error produced by the given aperture uncertainty was 0.628 mV, we can relate this value to the highest resolution of an ADC by assuming that 0.628 mV will be less than or equal ½ LSB. Therefore,

$$0.628 \text{ mV} \leq .5\ LSB = \frac{V_{REF}}{2^{N+1}} = \frac{5}{2^{N+1}}$$

or

$$2^{N+1} \leq 7961.8$$

which, solving for N (limited to an integer), yields a maximum resolution of 11 bits. ■

28.6 Mixed-Signal Layout Issues

Naturally, analog ICs are more sensitive to noise than digital ICs. For any analog design to be successful, careful attention must be paid to layout issues, particularly in a digital environment. Sensitive analog nodes must be protected and shielded from any potential noise sources. Grounding and power supply routing must also be considered when using digital and analog circuitry on the same substrate. Since a majority of ADCs use switches controlled by digital signals, separate routing channels must be provided for each type of signal.

Techniques used to increase the success of mixed-signal designs vary in complexity and priority. Strategies regarding the systemwide minimization of noise should always be considered foremost. A mixed-signal layout strategy can be modeled as seen in Fig. 28.27. The lowest issues are foundational and must be considered before each succeeding step. The successful mixed-signal design will always minimize the effect of the digital switching on the analog circuits.

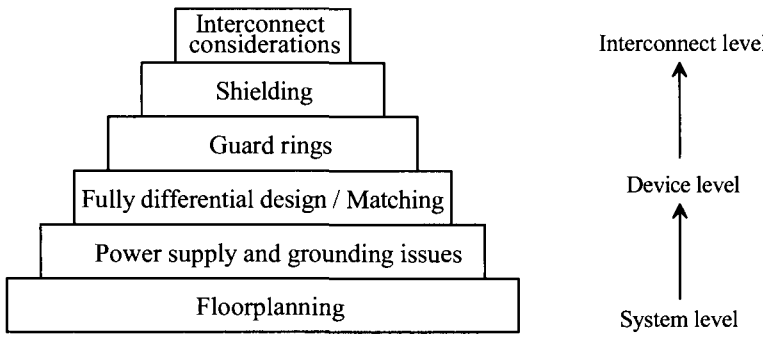

Figure 28.27 Mixed-signal layout strategy.

Floorplanning

The placement of sensitive analog components can greatly affect a circuit's performance. Many issues must be considered. In designing a mixed-signal system, strategies regarding the "floorplan" of the circuitry should be thoroughly analyzed well before the layout is to begin.

The analog circuitry should be categorized by the sensitivity of the analog signal to noise. For example, low-level signals or high-impedance nodes typically associated with input signals are considered to be sensitive nodes. These signals should be closely guarded and shielded, especially from digital output buffers. High-swing analog circuits such as comparators and output buffer amplifiers should be placed between the sensitive analog and the digital circuitry.

The digital circuitry should also be categorized by speed and function. Obviously, since digital output buffers are usually designed to drive capacitive loads at very high rates, they should be kept farthest from the sensitive analog signals. Next, the high and lower speed digital should be placed between the insensitive analog and the output buffers. An example of this type of strategy can be seen in Fig. 28.28 [1]. Notice that *the sensitive analog is as far away as possible from the digital output buffers* and that the least sensitive analog circuitry is next to the least offensive digital circuitry.

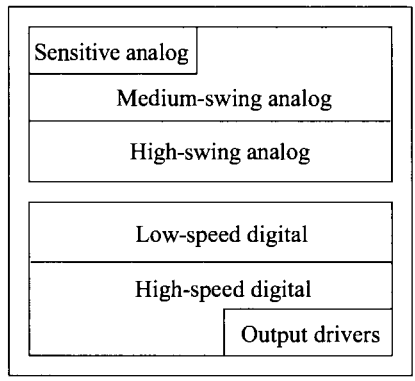

Figure 28.28 Example of a mixed-signal floorplan [1].

Power Supply and Grounding Issues

When analog and digital circuits exist together on the same die, danger exists of injecting noise from the digital system to the sensitive analog circuitry through the power supply and ground connections. Much of the intercoupling can be minimized by carefully considering how power and ground are supplied to both the analog and digital circuits.

In Fig. 28.28a analog and digital circuitry share the same routing to a single pad for power and ground. The resistors, R_{I1} and R_{I2}, represent the small, nonnegligible resistance of the interconnect to the pad. The inductors, L_{B1} and L_{B2}, represent the inductance of the bonding wire which connects the pads to the pin on the lead frame.

Since digital circuitry is typified by high amounts of transient currents due to switching, a small amount of resistance associated with the interconnect can result in significant voltage spikes. Low-level analog signals are very sensitive to such interference, thus resulting in a contaminated analog system. Another significant voltage spike can occur due to the inductance of the bonding wire. Since the voltage across the inductor is proportional to the change in current through it, voltage spikes equating to hundreds of millivolts can result! Both of these voltage effects are true for both the power and ground connection.

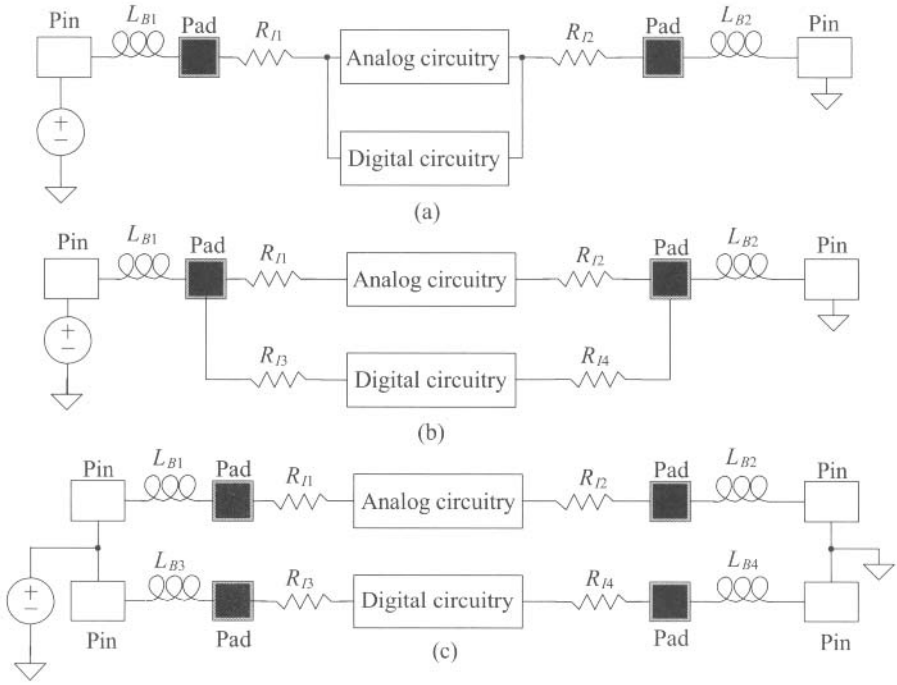

Figure 28.29 Power and ground connection examples, (a) poor noise immunity and (b) better noise immunity, and (c) using separate power and ground pins to achieve even better immunity.

One way to reduce the interference, seen in Fig. 28.29b, is to prohibit the analog and digital circuit from sharing the same interconnect. The routing for the supply and ground for both the analog and digital sections are provided separately. Although this eliminates the parasitic resistance due to the common interconnect, there is still a common inductance due to the bonding wire which causes interference.

Another method that minimizes interference even more than the previous case is seen in Fig. 28.29c. *By using separate pads and pins, the analog and the digital circuits are completely decoupled.* The current through the analog interconnect is much less abrupt than the digital; thus, the analog circuitry now has a "quiet" power and ground. However, this technique depends on whether extra pins and pads are available for this

use. The separate power supply and ground pins are then connected externally. *It is not wise to use two separate power supplies because if both types of circuits are not powered up simultaneously, latch-up could easily result.*

In cases presented in Fig. 28.29b and c, *the resistance associated with the analog connection to ground or supply can be reduced by making the power supply and ground bus as wide as feasible.* This reduces the overall resistance of the metal run, thus decreasing the voltage spikes that occur across the resistor. The inductor itself is impossible to eliminate, though it can be minimized with careful planning. Since the length of the bonding wire depends on the distance from the pad to the lead frame, *one could reduce the effect of the wire inductance by reserving pins closest to the die for sensitive connections such as analog supply and ground.* This, again, illustrates the importance of floorplanning.

Fully Differential Design

Fully differential operational amplifiers were discussed earlier, Fig. 28.30. The noise sources represent the noise from digital circuitry coupled through the parasitic stray capacitors. If equal amounts of noise are injected into the differential amplifiers, then the common-mode rejection inherent in the amplifiers will eliminate most or all of the noise. This, of course, depends on the symmetry of the amplifiers, meaning that matching the transistors in the amplifier becomes crucial. Therefore, *in a mixed-signal environment, layout techniques should be used to improve matching such as common-centroid and interdigitated techniques.*

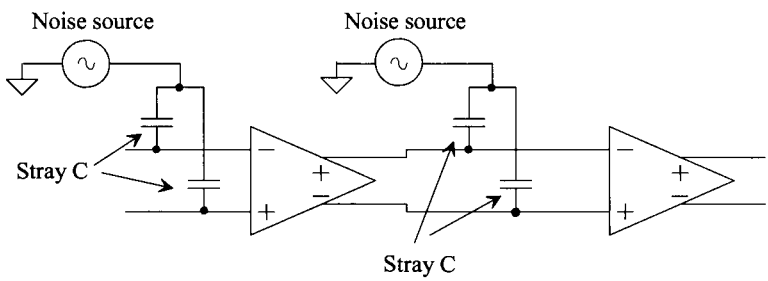

Figure 28.30 Differential output op-amps showing parasitic coupling to noise sources.

Guard Rings

Guard rings should be used wisely throughout a mixed-signal environment. Circuits that process sensitive signals should be placed in a separate well (if possible) with guard rings attached to the analog *VDD* supply. In the case of an n-well (only) process, the n-type devices outside the well should have guard rings attached to analog ground placed around them. Digital circuits should be placed in their own well with guard rings attached to digital *VDD*. Guard rings placed around the n-channel digital devices also help minimize the amount of noise transmitted from the digital devices.

Shielding

A number of techniques exist which can shield sensitive, low-level analog signals from noise resulting from digital switching. A shield can take the form of a layer tied to analog ground placed between two other layers, or it can be a barrier between two signals running in parallel.

If at all possible, one should avoid crossing sensitive analog signals, such as low-level analog input signals, with any digital signals. The parasitic capacitance coupling the two signal lines can be as much as a couple of fF, depending on the process. If it cannot be avoided, then attempt to carry the digital signal using the top layer of metal (such as metal2). If the analog signal is an input signal, then it will most likely be carried by the poly layer (or a lower level of metal). A strip of metal1 can be placed between the two layers and connected to analog ground (see Fig. 28.31).

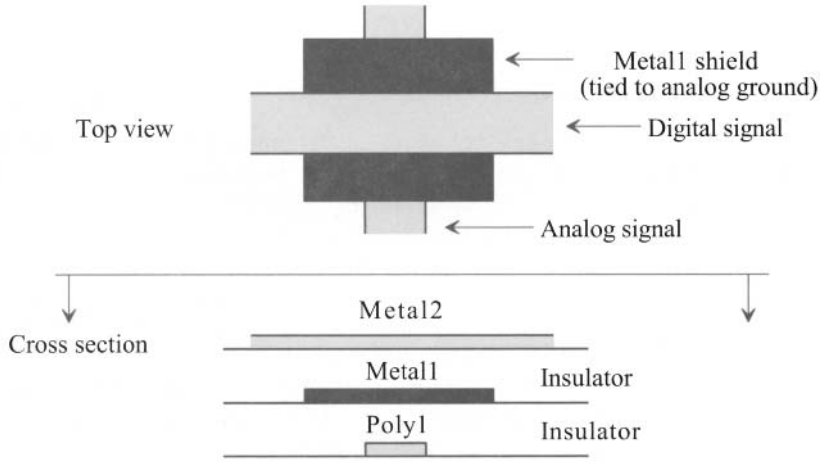

Figure 28.31 Shielding a sensitive analog signal from a digital signal crossover using a metal 1 shield layer.

Another situation that should be avoided is running an interconnect containing sensitive analog signals parallel and adjacent to any interconnect carrying digital signals. Coupling occurs due to the parasitic capacitance between the lines. If this situation cannot be avoided, then an additional line connected to analog ground should be placed between the two signals, as seen in Fig. 28.32. This method can also be used to partition the analog and digital sections of the chip.

In addition, the n-well can be used as a bottom-plate shield to protect analog signals from substrate noise. Poly resistors (or capacitors) used for sensitive analog signals can be shielded by placing an n-well beneath the components and connecting the well to analog *VDD*.

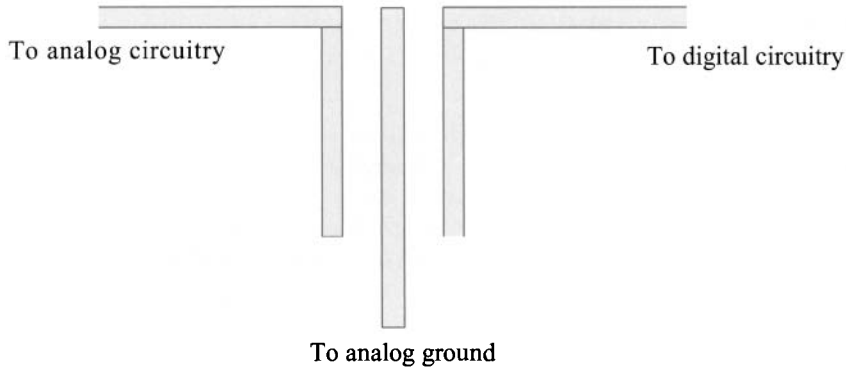

Figure 28.32 Using a dummy metal strip to provide shielding to two parallel signals.

Other Interconnect Considerations

Finally, some other layout strategies will incrementally improve the performance of the analog circuitry. However, if the previous strategies are not followed, these suggestions will be useless. *When routing the analog circuitry, minimize the lengths of current carrying paths.* This will simply reduce the amount of voltage drop across the path due to the metal1 or metal2 resistance. *Vias and contacts should also be used very liberally whenever changing layers.* Not only does this minimize resistance in the path, but it also improves fabrication reliablity. *Avoid using poly to route current carrying signal paths.* Not only is the poly higher in resistance value, but also the additional contact resistance required to change layers will not be insignificant. If the poly is made wider to lower the resistance, additional parasitic capacitance will be added to the node. *Use poly to route only high-impedance gate nodes that carry virtually no current.*

ADDITIONAL READING

[1] Y. Tsividis, *Mixed Analog-Digital VLSI Devices and Technology: An Introduction*, McGraw-Hill Publishing Co., 1996.

[2] B. Razavi, *Principles of Data Conversion System Design*, IEEE Press, 1995.

[3] M. J. Demler, *High-Speed Analog-to-Digital Conversion*, Academic Press, 1991.

[4] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Co., 1990.

[5] D. H. Sheingold, *Analog-Digital Conversion Handbook*, Prentice-Hall Publishing, 1986.

[6] S. K. Tewksbury, et al., "Terminology Related to the Performance of S/H, A/D and D/A Circuits," *IEEE Transactions on Circuits and Systems*, CAS-25, vol. CAS-25, pp. 419–426, July 1978.

PROBLEMS

28.1 Determine the number of quantization levels needed if one wanted to make a digital thermometer that was capable of measuring temperatures to within 0.1 °C accuracy over a range from –50 °C to 150 °C. What resolution of ADC would be required?

28.2 Using the same thermometer as above, what sampling rate, in samples per second, would be required if the temperature displayed a frequency of $15°\cdot\sin(0.01\cdot 2\pi t)$?

28.3 Determine the maximum droop allowed in an S/H used in a 16-bit ADC assuming that all other aspects of both the S/H and ADC are ideal. Assume $V_{ref} = 5$ V.

28.4 An S/H circuit settles to within 1 percent of its final value at 5 μs. What is the maximum resolution and speed with which an ADC can use this data assuming that the ADC is ideal?

28.5 A digitally programmable signal generator uses a 14-bit DAC with a 10-volt reference to generate a DC output voltage. What is the smallest incremental change at the output that can occur? What is the DAC's full-scale value? What is its accuracy?

28.6 Determine the maximum DNL (in LSBs) for a 3-bit DAC, which has the following characteristics. Does the DAC have 3-bit accuracy? If not, what is the resolution of the DAC having this characteristic?

Digital Input	Voltage Output
000	0 V
001	0.625 V
010	1.5625 V
011	2.0 V
100	2.5 V
101	3.125 V
110	3.4375 V
111	4.375 V

28.7 Repeat Problem 28.6 calculating the INL (in LSBs).

28.8 A DAC has a reference voltage of 1,000 V, and its maximum INL measures 2.5 mV. What is the maximum resolution of the converter assuming that all the other characteristics of the converter are ideal?

28.9 Determine the INL and DNL for a DAC that has a transfer curve shown in Fig. 28.33.

28.10 A DAC has a full-scale voltage of 4.97 V using a 5 V reference, and its minimum output voltage is limited by the value of one LSB. Determine the resolution and dynamic range of the converter.

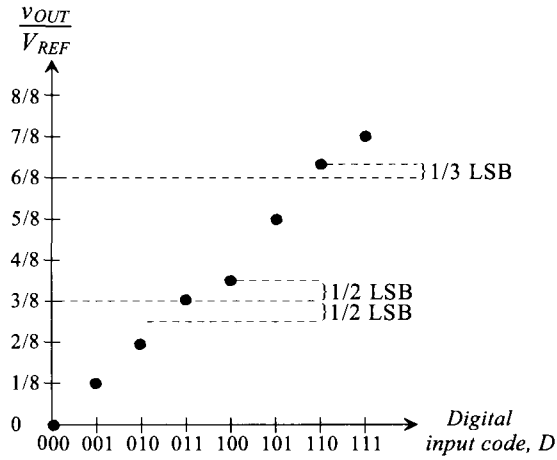

Figure 28.33 Transfer curves for Problem 28.9.

28.11 Prove that the RMS value of the quantization noise shown in Fig. 28.20b is as stated in Eq. (28.25).

28.12 An ADC has a stated *SNR* of 94 dB. Determine the effective number of bits of resolution of the converter.

28.13 Discuss the methods used to prevent aliasing and the advantages and disadvantages of each.