ANALOG-DIGITAL CONVERSION

ANALOG-DIGITAL CONVERSION

CHAPTER 2
FUNDAMENTALS OF SAMPLED DATA SYSTEMS

SECTION 2.1: CODING AND QUANTIZING
Walt Kester, Dan Sheingold, James Bryant

Analog-to-digital converters (ADCs) translate analog quantities, which are characteristic of most phenomena in the "real world," to digital language, used in information processing, computing, data transmission, and control systems. Digital-to-analog converters (DACs) are used in transforming transmitted or stored data, or the results of digital processing, back to "real-world" variables for control, information display, or further analog processing. The relationships between inputs and outputs of DACs and ADCs are shown in Figure 2.1.

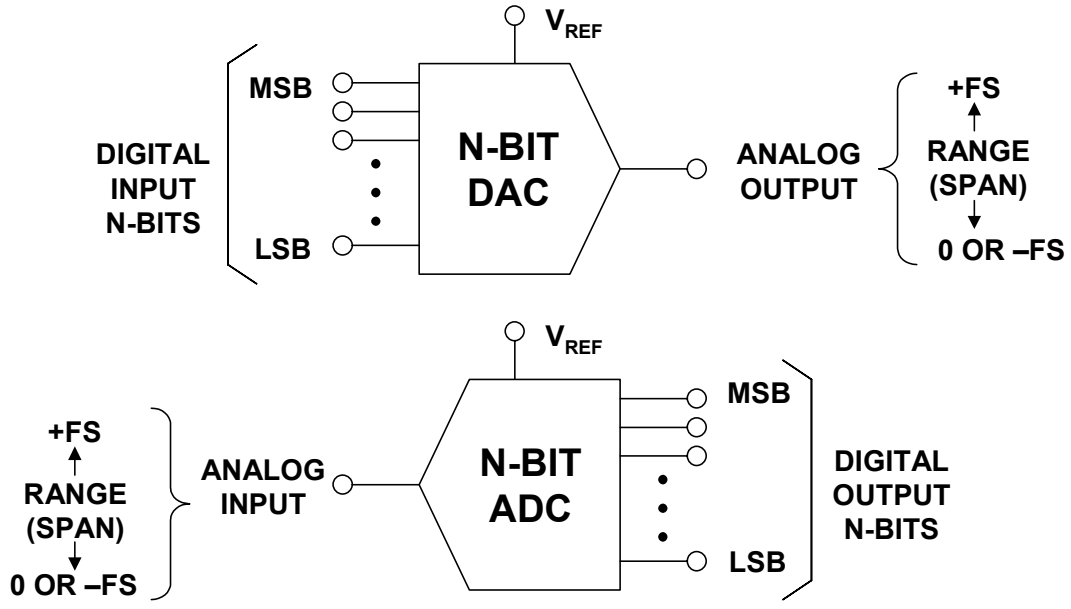


Figure 2.1: *Digital-to-Analog Converter (DAC) and Analog-to-Digital Converter (ADC) Input and Output Definitions*

Analog input variables, whatever their origin, are most frequently converted by transducers into voltages or currents. These electrical quantities may appear (1) as fast or slow "dc" continuous direct measurements of a phenomenon in the time domain, (2) as modulated ac waveforms (using a wide variety of modulation techniques), (3) or in some combination, with a spatial configuration of related variables to represent shaft angles.

Examples of the first are outputs of thermocouples, potentiometers on dc references, and analog computing circuitry; of the second, "chopped" optical measurements, ac strain gage or bridge outputs, and digital signals buried in noise; and of the third, synchros and resolvers.

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

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

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

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

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

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

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

It is composed of eight bits. The "1" at the extreme left is called the "most significant bit" (MSB, or Bit 1), and the one at the right is called the "least significant bit" (LSB, or

bit *N:* 8 in this case). The meaning of the code, as either a number, a character, or a representation of an analog variable, is unknown until the *code* and the *conversion relationship* have been defined. It is important not to confuse the designation of a particular bit (i.e., Bit 1, Bit 2, etc.) with the subscripts associated with the "a" array. The subscripts correspond to the power of 2 associated with the weight of a particular bit in the sequence.

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

In converter technology, full-scale (abbreviated *FS*) is independent of the number of bits of resolution, N. A more useful coding is *fractional* binary which is always normalized to full-scale. Integer binary can be interpreted as fractional binary if all integer values are divided by 2^N. For example, the MSB has a weight of ½ (i.e., $2^{(N-1)}/2^N = 2^{-1}$), the next bit has a weight of ¼ (i.e., 2^{-2}), and so forth down to the LSB, which has a weight of $1/2^N$ (i.e., 2^{-N}). When the weighted bits are added up, they form a number with any of 2^N values, from 0 to $(1 - 2^{-N})$ of full-scale. Additional bits simply provide more fine structure without affecting full-scale range. The relationship between base-10 numbers and binary numbers (base 2) are shown in Figure 2.2 along with examples of each.

WHOLE NUMBERS:

$$\text{Number}_{10} = a_{N-1}2^{N-1} + a_{N-2}2^{N-2} + \ldots + a_1 2^1 + a_0 2^0$$

MSB LSB

Example: $1011_2 = (1\times2^3) + (0\times2^2) + (1\times2^1) + (1\times2^0)$
$= 8 + 0 + 2 + 1 = 11_{10}$

FRACTIONAL NUMBERS:

$$\text{Number}_{10} = a_{N-1}2^{-1} + a_{N-2}2^{-2} + \ldots + a_1 2^{-(N-1)} + a_0 2^{-N}$$

MSB LSB

Example: $0.1011_2 = (1\times0.5) + (0\times0.25) + (1\times0.125) + (1\times0.0625)$
$= 0.5 + 0 + 0.125 + 0.0625 = 0.6875_{10}$

Figure 2.2: *Representing a Base-10 Number with a Binary Number (Base 2)*

Unipolar Codes

In data conversion systems, the coding method must be related to the analog input range (or span) of an ADC or the analog output range (or span) of a DAC. The simplest case is

when the input to the ADC or the output of the DAC is always a unipolar positive voltage (current outputs are very popular for DAC outputs, much less for ADC inputs). The most popular code for this type of signal is *straight binary* and is shown in Figure 2.3 for a 4-bit converter. Notice that there are 16 distinct possible levels, ranging from the all-zeros code 0000, to the all-ones code 1111. It is important to note that the analog value represented by the all-ones code is not full-scale (abbreviated FS), but FS – 1 LSB. This is a common convention in data conversion notation and applies to both ADCs and DACs. Figure 2.3 gives the base-10 equivalent number, the value of the base-2 binary code relative to full-scale (FS), and also the corresponding voltage level for each code (assuming a +10 V full-scale converter. The Gray code equivalent is also shown, and will be discussed shortly.

BASE 10 NUMBER	SCALE	+10V FS	BINARY	GRAY
+15	+FS – 1LSB = +15/16 FS	9.375	1 1 1 1	1 0 0 0
+14	+7/8 FS	8.750	1 1 1 0	1 0 0 1
+13	+13/16 FS	8.125	1 1 0 1	1 0 1 1
+12	+3/4 FS	7.500	1 1 0 0	1 0 1 0
+11	+11/16 FS	6.875	1 0 1 1	1 1 1 0
+10	+5/8 FS	6.250	1 0 1 0	1 1 1 1
+9	+9/16 FS	5.625	1 0 0 1	1 1 0 1
+8	+1/2 FS	5.000	1 0 0 0	1 1 0 0
+7	+7/16 FS	4.375	0 1 1 1	0 1 0 0
+6	+3/8 FS	3.750	0 1 1 0	0 1 0 1
+5	+5/16 FS	3.125	0 1 0 1	0 1 1 1
+4	+1/4 FS	2.500	0 1 0 0	0 1 1 0
+3	+3/16 FS	1.875	0 0 1 1	0 0 1 0
+2	+1/8 FS	1.250	0 0 1 0	0 0 1 1
+1	1LSB = +1/16 FS	0.625	0 0 0 1	0 0 0 1
0	0	0.000	0 0 0 0	0 0 0 0

Figure 2.3: *Unipolar Binary Codes, 4-bit Converter*

Figure 2.4 shows the transfer function for an ideal 3-bit DAC with straight binary input coding. Notice that the analog output is zero for the all-zeros input code. As the digital input code increases, the analog output increases 1 LSB (1/8 scale in this example) per code. The most positive output voltage is 7/8 FS, corresponding to a value equal to FS – 1 LSB. The mid-scale output of 1/2 FS is generated when the digital input code is 100.

The transfer function of an ideal 3-bit ADC is shown in Figure 2.5. There is a range of analog input voltage over which the ADC will produce a given output code; this range is the *quantization uncertainty* and is equal to 1 LSB. Note that the width of the transition regions between adjacent codes is zero for an ideal ADC. In practice, however, there is always transition noise associated with these levels, and therefore the width is non-zero. It is customary to define the analog input corresponding to a given code by the *code center* which lies halfway between two adjacent transition regions (illustrated by the black dots in the diagram). This requires that the first transition region occur at ½ LSB. The full-scale analog input voltage is defined by 7/8 FS, (FS – 1 LSB).

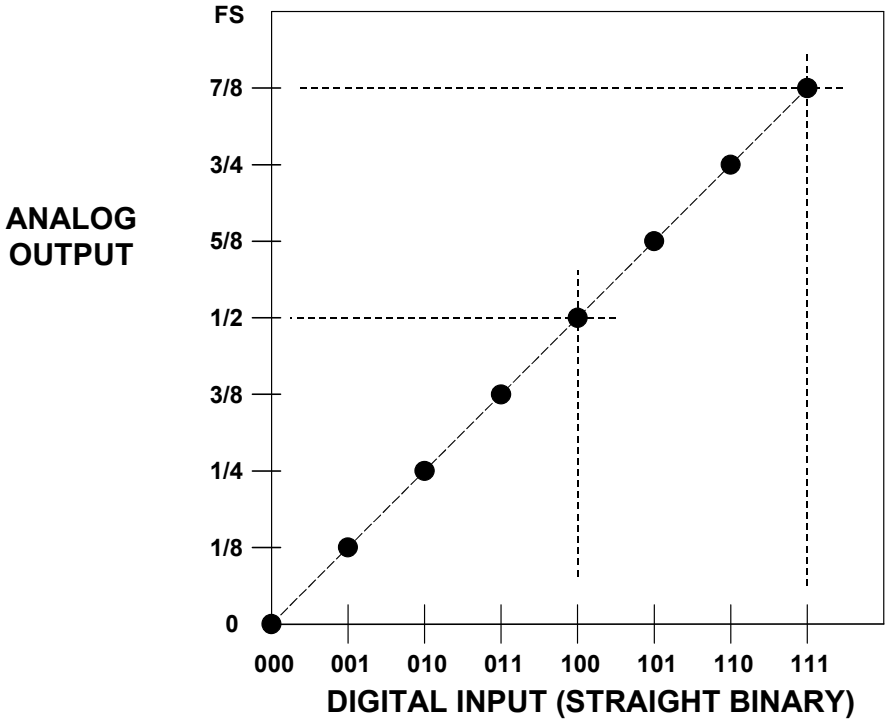


Figure 2.4: *Transfer Function for Ideal Unipolar 3-bit DAC*

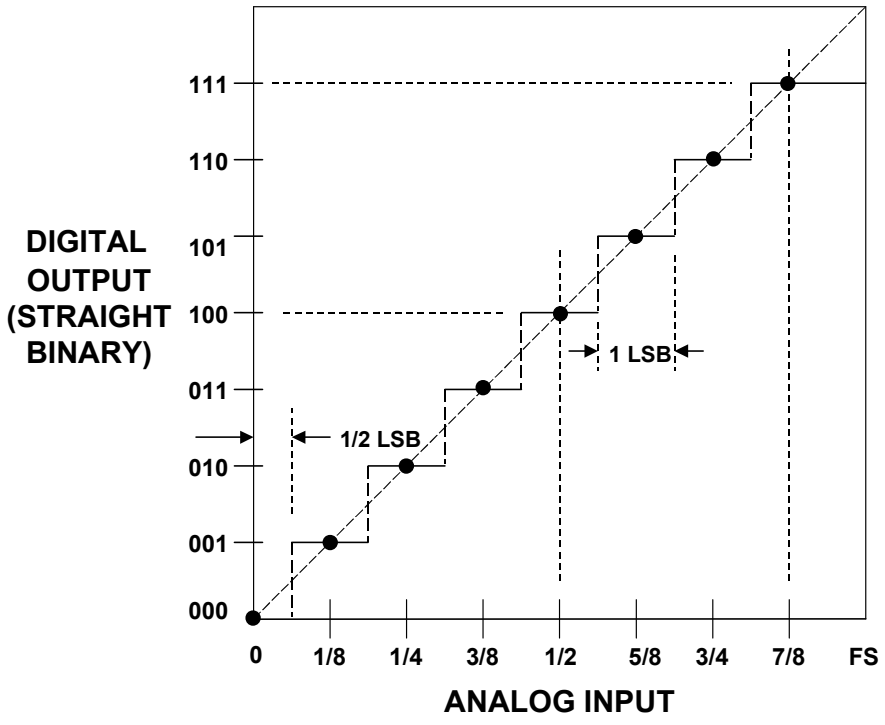


Figure 2.5: *Transfer Function for Ideal Unipolar 3-bit ADC*

Gray Code

Another code worth mentioning at this point is the *Gray* code (or *reflective-binary*) which was invented by Elisha Gray in 1878 (Reference 1) and later re-invented by Frank Gray in 1949 (see Reference 2). The Gray code equivalent of the 4-bit straight binary code is also shown in Figure 2.3. Although it is rarely used in computer arithmetic, it has some useful properties which make it attractive to A/D conversion. Notice that in Gray code, as the number value changes, the transitions from one code to the next involve only one bit at a time. Contrast this to the binary code where all the bits change when making the transition between 0111 and 1000. Some ADCs make use of it internally and then convert the Gray code to a binary code for external use.

One of the earliest practical ADCs to use the Gray code was a 7-bit, 100 kSPS electron beam encoder developed by Bell Labs and described in a 1948 reference (Reference 3). The basic electron beam coder concepts are shown in Figure 2.6 for a 4-bit device. The early tubes operated in the serial mode (A). The analog signal is first passed through a sample-and-hold, and during the "hold" interval, the beam is swept horizontally across the tube. The Y-deflection for a single sweep therefore corresponds to the value of the analog signal from the sample-and-hold. The shadow mask is coded to produce the proper binary code, depending on the vertical deflection. The code is registered by the collector, and the bits are generated in serial format. Later tubes used a fan-shaped beam (shown in Figure 2.6B), creating a "flash" converter delivering a parallel output word.

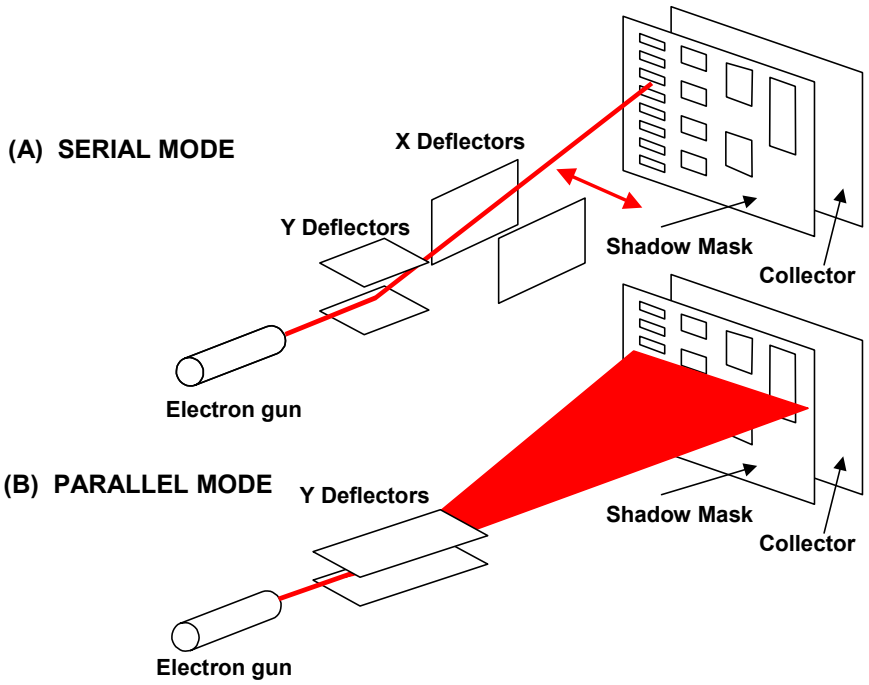


Figure 2.6: *The Electron Beam Coder: (A) Serial Mode and (B) Parallel or "Flash" Mode*

Early electron tube coders used a binary-coded shadow mask, and large errors can occur if the beam straddles two adjacent codes and illuminates both of them. The way these errors occur is illustrated in Figure 2.7A, where the horizontal line represents the beam sweep at the mid-scale transition point (transition between code 0111 and code 1000). For example, an error in the most significant bit (MSB) produces an error of ½ scale. These errors were minimized by placing fine horizontal sensing wires across the boundaries of each of the quantization levels. If the beam initially fell on one of the wires, a small voltage was added to the vertical deflection voltage which moved the beam away from the transition region.

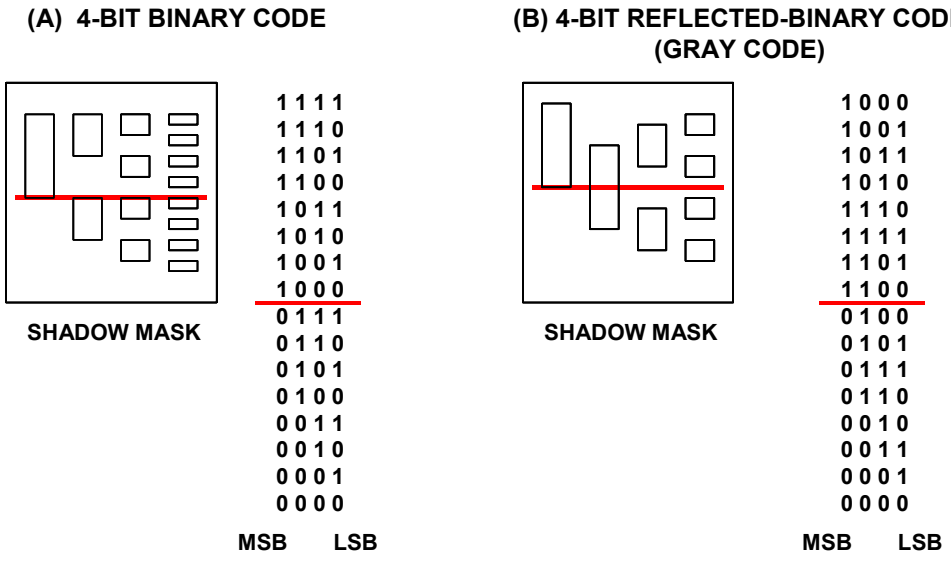


Figure 2.7: *Electron Beam Coder Shadow Masks for Binary Code (A) and Gray Code (B)*

The errors associated with binary shadow masks were eliminated by using a Gray code shadow mask as shown in Figure 2.7B. As mentioned above, the Gray code has the property that adjacent levels differ by only one digit in the corresponding Gray-coded word. Therefore, if there is an error in a bit decision for a particular level, the corresponding error after conversion to binary code is only one least significant bit (LSB). In the case of mid-scale, note that only the MSB changes. It is interesting to note that this same phenomenon can occur in modern comparator-based flash converters due to comparator metastability. With small overdrive, there is a finite probability that the output of a comparator will generate the wrong decision in its latched output, producing the same effect if straight binary decoding techniques are used. In many cases, Gray code, or "pseudo-Gray" codes are used to decode the comparator bank. The Gray code output is then latched, converted to binary, and latched again at the final output.

As a historical note, in spite of the many mechanical and electrical problems relating to beam alignment, electron tube coding technology reached its peak in the mid-l960s with an experimental 9-bit coder capable of 12-MSPS sampling rates (Reference 4). Shortly thereafter, however, advances in all solid-state ADC techniques made the electron tube technology obsolete.

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

ADCs which use the Gray code internally almost always convert the Gray code output to binary for external use. The conversion from Gray-to-binary and binary-to-Gray is easily accomplished with the exclusive-or logic function as shown in Figure 2.8.

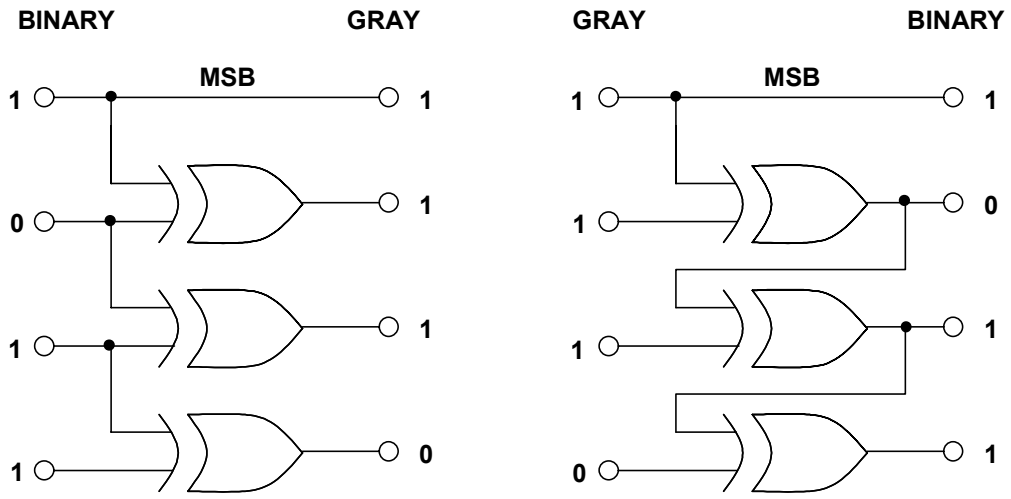


Figure 2.8: *Binary-to-Gray and Gray-to-Binary Conversion Using the Exclusive-Or Logic Function*

Bipolar Codes

In many systems, it is desirable to represent both positive and negative analog quantities with binary codes. Either *offset binary*, *twos complement*, *ones complement*, or *sign magnitude* codes will accomplish this, but offset binary and twos complement are by far the most popular. The relationships between these codes for a 4-bit systems is shown in Figure 2.9. Note that the values are scaled for a ±5-V full-scale input/output voltage range.

For *offset binary*, the zero signal value is assigned the code 1000. The sequence of codes is identical to that of straight binary. The only difference between a straight and offset binary system is the half-scale offset associated with analog signal. The most negative value (–FS + 1 LSB) is assigned the code 0001, and the most positive value (+FS – 1 LSB) is assigned the code 1111. Note that in order to maintain perfect symmetry about mid-scale, the all-zeros code (0000) representing negative full-scale (–FS) is not normally used in computation. It can be used to represent a negative off-range condition or simply assigned the value of the 0001 (–FS + 1 LSB).

BASE 10 NUMBER	SCALE	±5V FS	OFFSET BINARY	TWOS COMP.	ONES COMP.	SIGN MAG.
+7	+FS – 1LSB = +7/8 FS	+4.375	1 1 1 1	0 1 1 1	0 1 1 1	0 1 1 1
+6	+3/4 FS	+3.750	1 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0
+5	+5/8 FS	+3.125	1 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1
+4	+1/2 FS	+2.500	1 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0
+3	+3/8 FS	+1.875	1 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
+2	+1/4 FS	+1.250	1 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0
+1	+1/8 FS	+0.625	1 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
0	0	0.000	1 0 0 0	0 0 0 0	*0 0 0 0	*1 0 0 0
–1	– 1/8 FS	–0.625	0 1 1 1	1 1 1 1	1 1 1 0	1 0 0 1
–2	– 1/4 FS	–1.250	0 1 1 0	1 1 1 0	1 1 0 1	1 0 1 0
–3	– 3/8 FS	–1.875	0 1 0 1	1 1 0 1	1 1 0 0	1 0 1 1
–4	–1/2 FS	–2.500	0 1 0 0	1 1 0 0	1 0 1 1	1 1 0 0
–5	–5/8 FS	–3.125	0 0 1 1	1 0 1 1	1 0 1 0	1 1 0 1
–6	–3/4 FS	–3.750	0 0 1 0	1 0 1 0	1 0 0 1	1 1 1 0
–7	– FS + 1LSB = –7/8 FS	–4.375	0 0 0 1	1 0 0 1	1 0 0 0	1 1 1 1
–8	– FS	–5.000	0 0 0 0	1 0 0 0		

NOT NORMALLY USED
IN COMPUTATIONS (SEE TEXT)

*	ONES COMP.	SIGN MAG.
0+	0 0 0 0	0 0 0 0
0–	1 1 1 1	1 0 0 0

Figure 2.9: *Bipolar Codes, 4-bit Converter*

The relationship between the offset binary code and the analog output range of a bipolar 3-bit DAC is shown in Figure 2.10. The analog output of the DAC is zero for the zero-value input code 100. The most negative output voltage is generally defined by the 001 code (–FS + 1 LSB), and the most positive by 111 (+FS – 1 LSB). The output voltage for the 000 input code is available for use if desired, but makes the output non-symmetrical about zero and complicates the mathematics.

The offset binary output code for a bipolar 3-bit ADC as a function of its analog input is shown in Figure 2.11. Note that zero analog input defines the center of the mid-scale code 100. As in the case of bipolar DACs, the most negative input voltage is generally defined by the 001 code (–FS + 1 LSB), and the most positive by 111 (+FS – 1 LSB). As discussed above, the 000 output code is available for use if desired, but makes the output non-symmetrical about zero and complicates the mathematics.

Twos complement is identical to offset binary with the most-significant-bit (MSB) complemented (inverted). This is obviously very easy to accomplish in a data converter, using a simple inverter or taking the complementary output of a "D" flip-flop. The popularity of twos complement coding lies in the ease with which mathematical operations can be performed in computers and DSPs. Twos complement, for conversion purposes, consists of a binary code for positive magnitudes (0 sign bit), and the twos complement of each positive number to represent its negative. The twos complement is formed arithmetically by complementing the number and adding 1 LSB. For example, –3/8 FS is obtained by taking the twos complement of +3/8 FS. This is done by first complementing +3/8 FS, 0011 obtaining 1100. Adding 1 LSB, we obtain 1101.

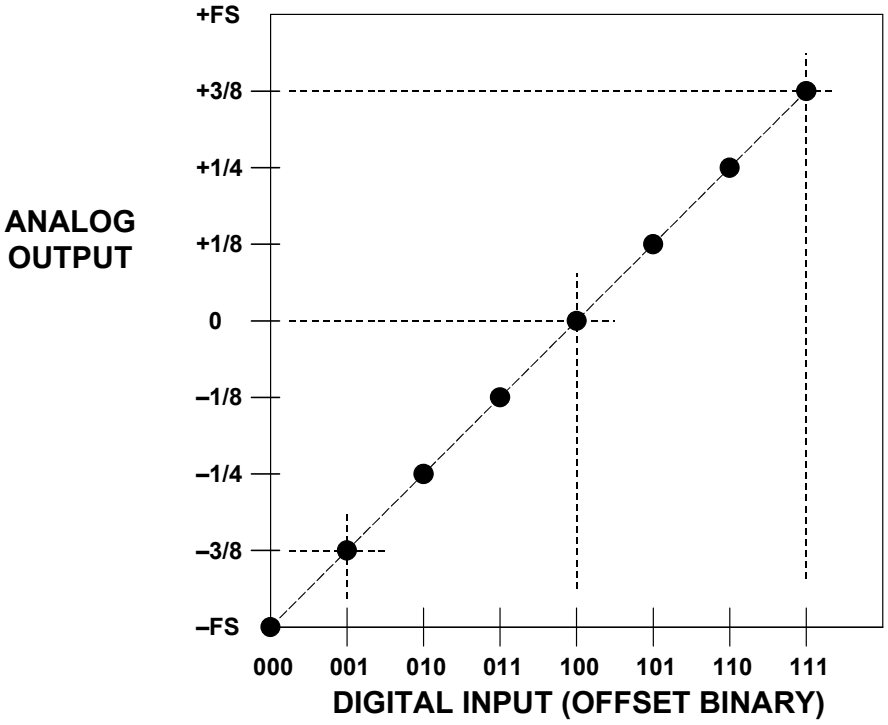


Figure 2.10: *Transfer Function for Ideal Bipolar 3-bit DAC*

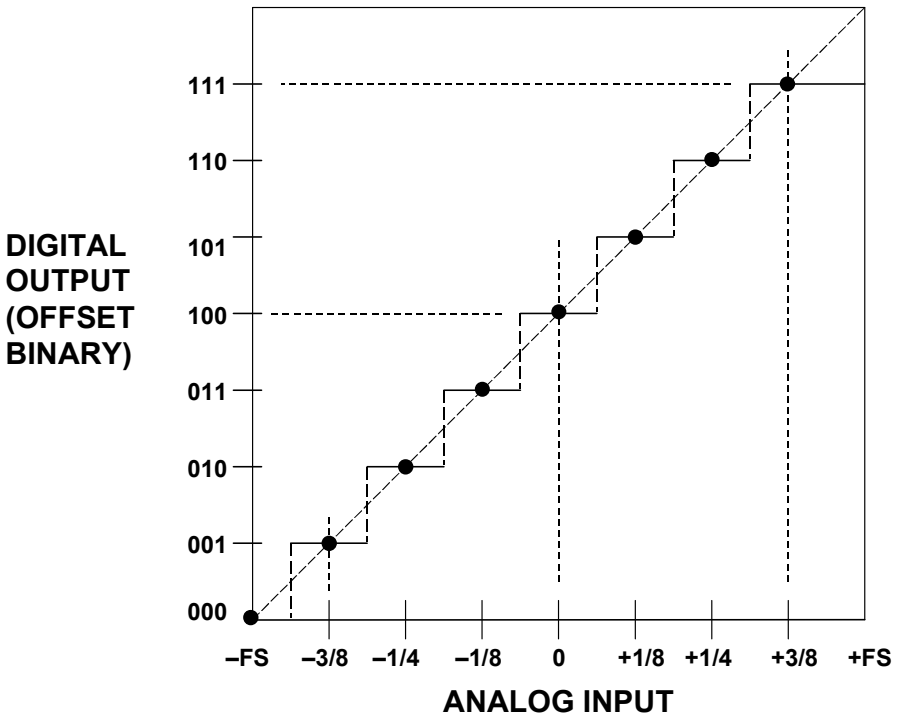


Figure 2.11: *Transfer Function for Ideal Bipolar 3-bit ADC*

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

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

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

Figure 2.12 summarizes the relationships between the various bipolar codes: offset binary, twos complement, ones complement, and sign-magnitude and shows how to convert between them.

To Convert From → / To ↓	Sign Magnitude	2's Complement	Offset Binary	1's Complement
Sign Magnitude	No Change	If MSB = 1, complement other bits, add 00…01	Complement MSB If new MSB = 1, complement other bits, add 00…01	If MSB = 1, complement other bits
2's Complement	If MSB = 1, complement other bits, add 00…01	No Change	Complement MSB	If MSB = 1, add 00…01
Offset binary	Complement MSB If new MSB = 0 complement other bits, add 00…01	Complement MSB	No Change	Complement MSB If new MSB = 0, add 00…01
1's Complement	If MSB = 1, complement other bits	If MSB = 1, add 11…11	Complement MSB If new MSB = 1, add 11…11	No Change

Figure 2.12: *Relationships Among Bipolar Codes*

The last code to be considered in this section is *binary-coded-decimal (BCD)*, where each base-10 digit (0 to 9) in a decimal number is represented as the corresponding 4-bit straight binary word as shown in Figure 2.13. The minimum digit 0 is represented as

0000, and the digit 9 by 1001. This code is relatively inefficient, since only 10 of the 16 code states for each decade are used. It is, however, a very useful code for interfacing to decimal displays such as in digital voltmeters.

BASE 10 NUMBER	SCALE	+10V FS	DECADE 1	DECADE 2	DECADE 3	DECADE 4
+15	+FS – 1LSB = +15/16 FS	9.375	1 0 0 1	0 0 1 1	0 1 1 1	0 1 0 1
+14	+7/8 FS	8.750	1 0 0 0	0 1 1 1	0 1 0 1	0 0 0 0
+13	+13/16 FS	8.125	1 0 0 0	0 0 0 1	0 0 1 0	0 1 0 1
+12	+3/4 FS	7.500	0 1 1 1	0 1 0 1	0 0 0 0	0 0 0 0
+11	+11/16 FS	6.875	0 1 1 0	1 0 0 0	0 1 1 1	0 1 0 1
+10	+5/8 FS	6.250	0 1 1 0	0 0 1 0	0 1 0 1	0 0 0 0
+9	+9/16 FS	5.625	0 1 0 1	0 1 1 0	0 0 1 0	0 1 0 1
+8	+1/2 FS	5.000	0 1 0 1	0 0 0 0	0 0 0 0	0 0 0 0
+7	+7/16 FS	4.375	0 1 0 0	0 0 1 1	0 1 1 1	0 1 0 1
+6	+3/8 FS	3.750	0 0 1 1	0 1 1 1	0 1 0 1	0 0 0 0
+5	+5/16 FS	3.125	0 0 1 1	0 0 0 1	0 0 1 0	0 1 0 1
+4	+1/4 FS	2.500	0 0 1 0	0 1 0 1	0 0 0 0	0 0 0 0
+3	+3/16 FS	1.875	0 0 0 1	1 0 0 0	0 1 1 1	0 1 0 1
+2	+1/8 FS	1.250	0 0 0 1	0 0 1 0	0 1 0 1	0 0 0 0
+1	1LSB = +1/16 FS	0.625	0 0 0 0	0 1 1 0	0 0 1 0	0 1 0 1
0	0	0.000	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Figure 2.13: *Binary Coded Decimal (BCD) Code*

Complementary Codes

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

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

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

DAC and ADC Static Transfer Functions and DC Errors

The most important thing to remember about both DACs and ADCs is that either the input or output is digital, and therefore the signal is quantized. That is, an N-bit word represents one of 2^N possible states, and therefore an N-bit DAC (with a fixed reference)

can have only 2^N possible analog outputs, and an N-bit ADC can have only 2^N possible digital outputs. As previously discussed, the analog signals will generally be voltages or currents.

The resolution of data converters may be expressed in several different ways: the weight of the Least Significant Bit (LSB), parts per million of full-scale (ppm FS), millivolts (mV), etc. Different devices (even from the same manufacturer) will be specified differently, so converter users must learn to translate between the different types of specifications if they are to compare devices successfully. The size of the least significant bit for various resolutions is shown in Figure 2.14.

RESOLUTION N	2^N	VOLTAGE (10V FS)	ppm FS	% FS	dB FS
2-bit	4	2.5 V	250,000	25	– 12
4-bit	16	625 mV	62,500	6.25	– 24
6-bit	64	156 mV	15,625	1.56	– 36
8-bit	256	39.1 mV	3,906	0.39	– 48
10-bit	1,024	9.77 mV (10 mV)	977	0.098	– 60
12-bit	4,096	2.44 mV	244	0.024	– 72
14-bit	16,384	610 µV	61	0.0061	– 84
16-bit	65,536	153 µV	15	0.0015	– 96
18-bit	262,144	38 µV	4	0.0004	– 108
20-bit	1,048,576	9.54 µV (10 µV)	1	0.0001	– 120
22-bit	4,194,304	2.38 µV	0.24	0.000024	– 132
24-bit	16,777,216	596 nV*	0.06	0.000006	– 144

***600nV is the Johnson Noise in a 10kHz BW of a 2.2kΩ Resistor @ 25°C**

**Remember: 10-bits and 10V FS yields an LSB of 10mV, 1000ppm, or 0.1%.
All other values may be calculated by powers of 2.**

Figure 2.14: *Quantization: The Size of a Least Significant Bit (LSB)*

Before we can consider the various architectures used in data converters, it is necessary to consider the performance to be expected, and the specifications which are important. The following sections will consider the definition of errors and specifications used for data converters. This is important in understanding the strengths and weaknesses of different ADC/DAC architectures.

The first applications of data converters were in measurement and control where the exact timing of the conversion was usually unimportant, and the data rate was slow. In such applications, the dc specifications of converters are important, but timing and ac specifications are not. Today many, if not most, converters are used in *sampling* and *reconstruction* systems where ac specifications are critical (and dc ones may not be)—these will be considered in Section 2.3 of this chapter.

Figure 2.15 shows the ideal transfer characteristics for a 3-bit unipolar DAC and a 3-bit unipolar ADC. In a DAC, both the input and the output are quantized, and the graph consists of eight points—while it is reasonable to discuss the line through these points, it

is very important to remember that the actual transfer characteristic is *not* a line, but a number of discrete points.

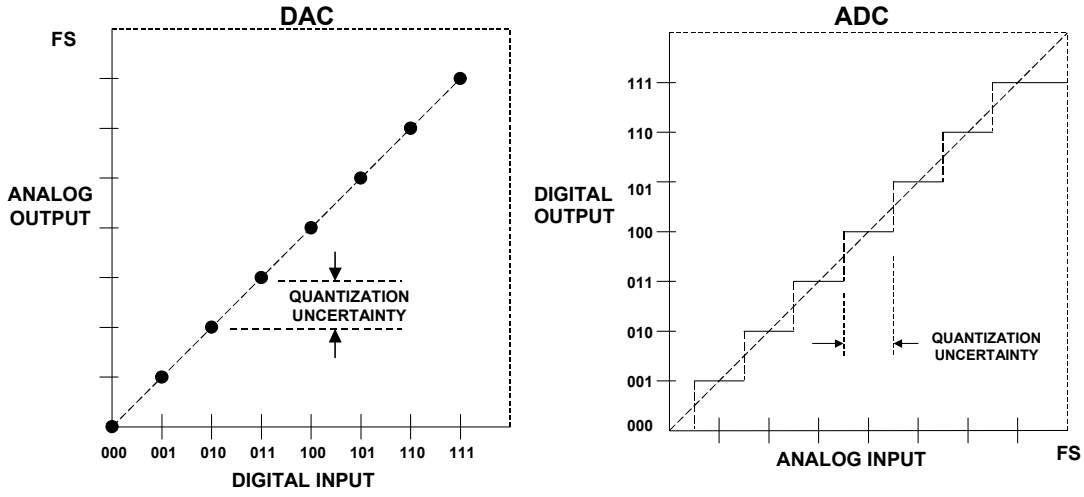


Figure 2.15: *Transfer Functions for Ideal 3-Bit DAC and ADC*

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

For both DACs and ADCs, digital full-scale (all "1"s) corresponds to 1 LSB below the analog full-scale (FS). The (ideal) ADC transitions take place at ½ LSB above zero, and thereafter every LSB, until 1½ LSB below analog full-scale. Since the analog input to an ADC can take any value, but the digital output is quantized, there may be a difference of up to ½ LSB between the actual analog input and the exact value of the digital output. This is known as the *quantization error* or *quantization uncertainty* as shown in Figure 2.15. In ac (sampling) applications this quantization error gives rise to *quantization noise* which will be discussed in Section 2.3 of this chapter.

As previously discussed, there are many possible digital coding schemes for data converters: *straight binary*, *offset binary*, *1's complement*, *2's complement*, sign *magnitude*, *gray code*, *BCD* and others. This section, being devoted mainly to the *analog* issues surrounding data converters, will use simple *binary* and *offset binary* in its examples and will not consider the merits and disadvantages of these, or any other forms of digital code.

The examples in Figure 2.15 use *unipolar* converters, whose analog port has only a single polarity. These are the simplest type, but *bipolar* converters are generally more useful in real-world applications. There are two types of bipolar converters: the simpler is merely a unipolar converter with an accurate 1 MSB of negative offset (and many converters are arranged so that this offset may be switched in and out so that they can be used as either unipolar or bipolar converters at will), but the other, known as a *sign-magnitude* converter is more complex, and has N bits of magnitude information and an additional bit which corresponds to the sign of the analog signal. Sign-magnitude DACs are quite rare,

and sign-magnitude ADCs are found mostly in digital voltmeters (DVMs). The unipolar, offset binary, and sign-magnitude representations are shown in Figure 2.16.

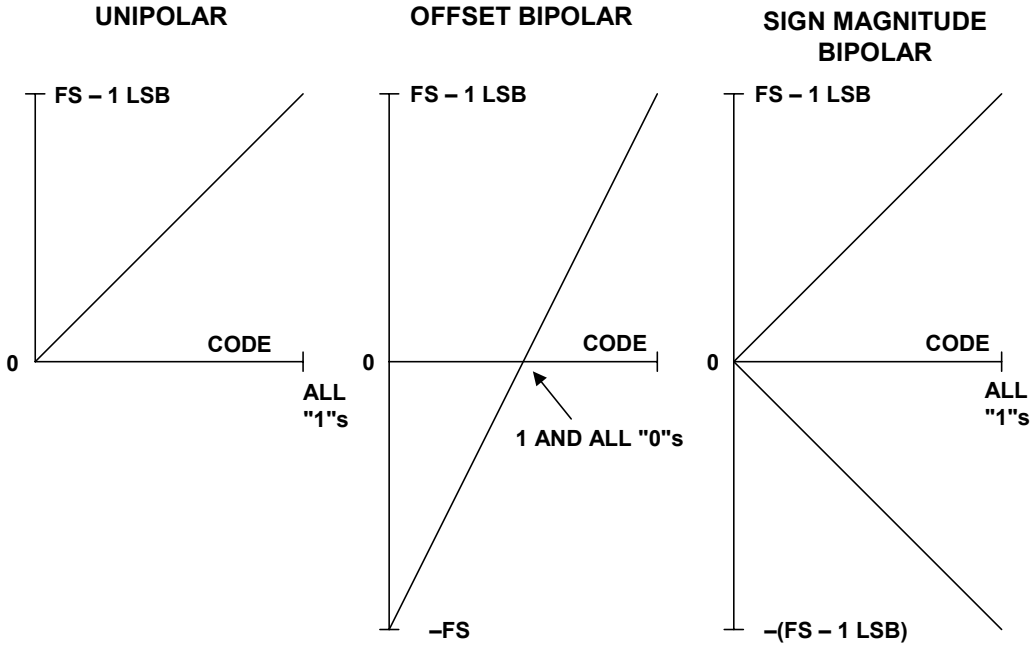


Figure 2.16: *Unipolar and Bipolar Converters*

The four dc errors in a data converter are *offset error*, *gain error*, and two types of *linearity error (differential and integral)*. Offset and gain errors are analogous to offset and gain errors in amplifiers as shown in Figure 2.17 for a bipolar input range. (Though offset error and zero error, which are identical in amplifiers and unipolar data converters, are not identical in bipolar converters and should be carefully distinguished.)

The transfer characteristics of both DACs and ADCs may be expressed as a straight line given by D = K + GA, where D is the digital code, A is the analog signal, and K and G are constants. In a unipolar converter, the ideal value of K is zero; in an offset bipolar converter it is –1 MSB. The offset error is the amount by which the actual value of K differs from its ideal value.

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

The integral linearity error of a converter is also analogous to the linearity error of an amplifier, and is defined as the maximum deviation of the actual transfer characteristic of the converter from a straight line, and is generally expressed as a percentage of full-scale (but may be given in LSBs). For an ADC, the most popular convention is to draw the straight line through the mid-points of the codes, or the code centers. There are two

common ways of choosing the straight line: *end point* and *best straight line* as shown in Figure 2.18.

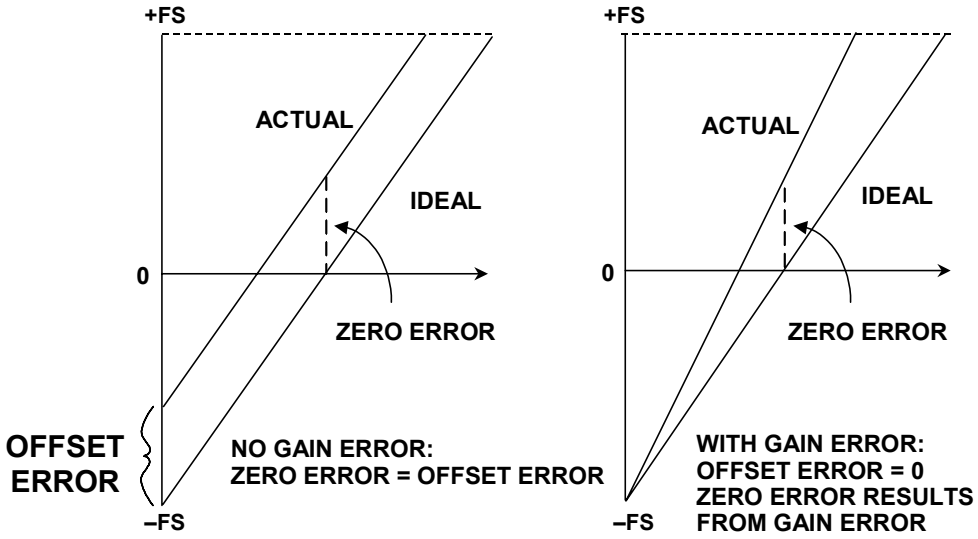


Figure 2.17: *Bipolar Data Converter Offset and Gain Error*

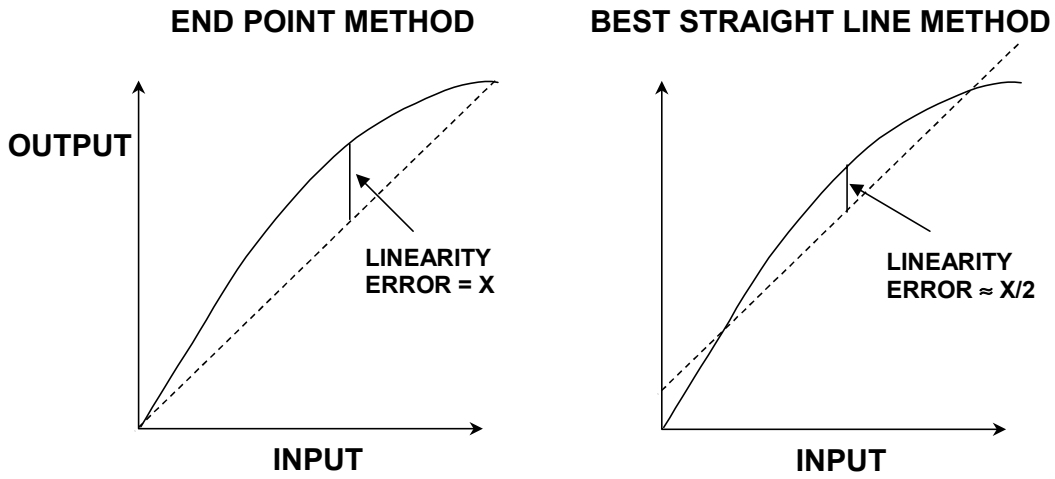


Figure 2.18: *Method of Measuring Integral Linearity Errors (Same Converter on Both Graphs)*

In the *end point* system, the deviation is measured from the straight line through the origin and the full-scale point (after gain adjustment). This is the most useful integral linearity measurement for measurement and control applications of data converters (since error budgets depend on deviation from the ideal transfer characteristic, not from some arbitrary "best fit"), and is the one normally adopted by Analog Devices, Inc.

The *best straight line*, however, does give a better prediction of distortion in ac applications, and also gives a lower value of "linearity error" on a data sheet. The best fit straight line is drawn through the transfer characteristic of the device using standard

curve fitting techniques, and the maximum deviation is measured from this line. In general, the integral linearity error measured in this way is only 50% of the value measured by end point methods. This makes the method good for producing impressive data sheets, but it is less useful for error budget analysis. For ac applications it is better to specify distortion than dc linearity, so it is rarely necessary to use the best straight line method to define converter linearity.

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

Where the change in analog signal corresponding to 1 LSB digital change is more or less than 1 LSB, there is said to be a DNL error. The DNL error of a converter is normally defined as the maximum value of DNL to be found at any transition across the range of the converter. Figure 2.19 shows the non-ideal transfer functions for a DAC and an ADC and shows the effects of the DNL error.

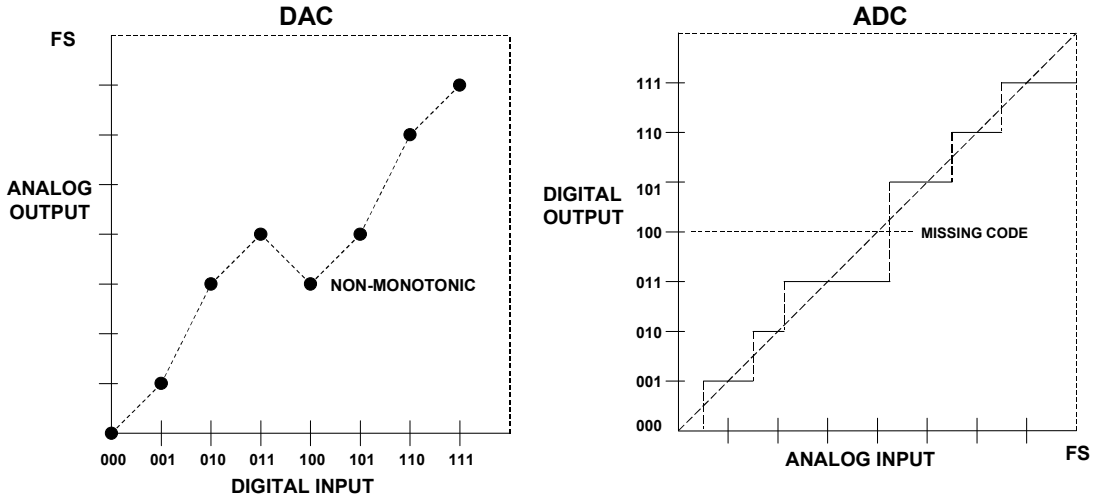


Figure 2.19: *Transfer Functions for Non-Ideal 3-Bit DAC and ADC*

The DNL of a DAC is examined more closely in Figure 2.20. If the DNL of a DAC is less than –1 LSB at any transition, the DAC is *non-monotonic* i.e., its transfer characteristic contains one or more localized maxima or minima. A DNL greater than +1 LSB does not cause non-monotonicity, but is still undesirable. In many DAC applications (especially closed-loop systems where non-monotonicity can change negative feedback to positive feedback), it is critically important that DACs are monotonic. DAC monotonicity is often explicitly specified on data sheets, although if the DNL is guaranteed to be less than 1 LSB (i.e., $|DNL| \leq 1$ LSB) then the device must be monotonic, even without an explicit guarantee.

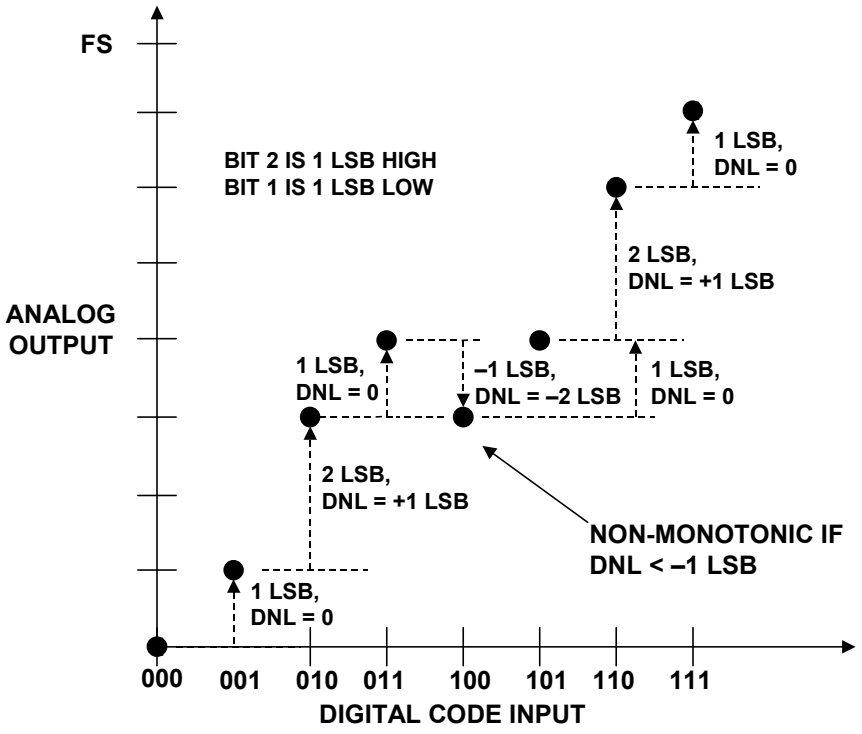


Figure 2.20: *Details of DAC Differential Nonlinearity*

In Figure 2.21, the DNL of an ADC is examined more closely on an expanded scale. ADCs can be non-monotonic, but a more common result of excess DNL in ADCs is *missing codes*. Missing codes in an ADC are as objectionable as non-monotonicity in a DAC. Again, they result from DNL < –1 LSB.

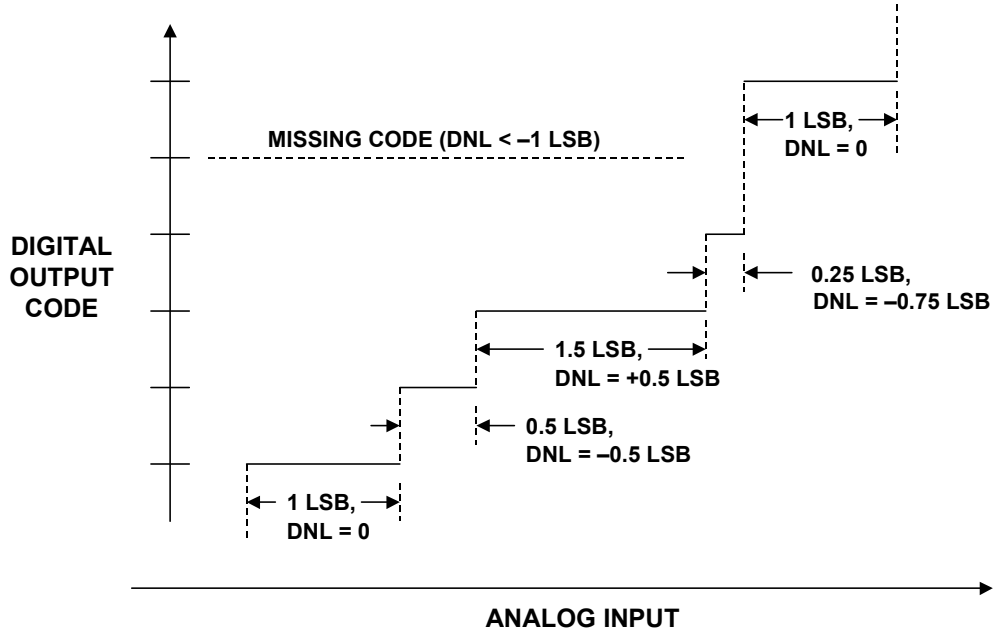


Figure 2.21: *Details of ADC Differential Nonlinearity*

Not only can ADCs have missing codes, they can also be non-monotonic as shown in Figure 2.22. As in the case of DACs, this can present major problems—especially in servo applications.

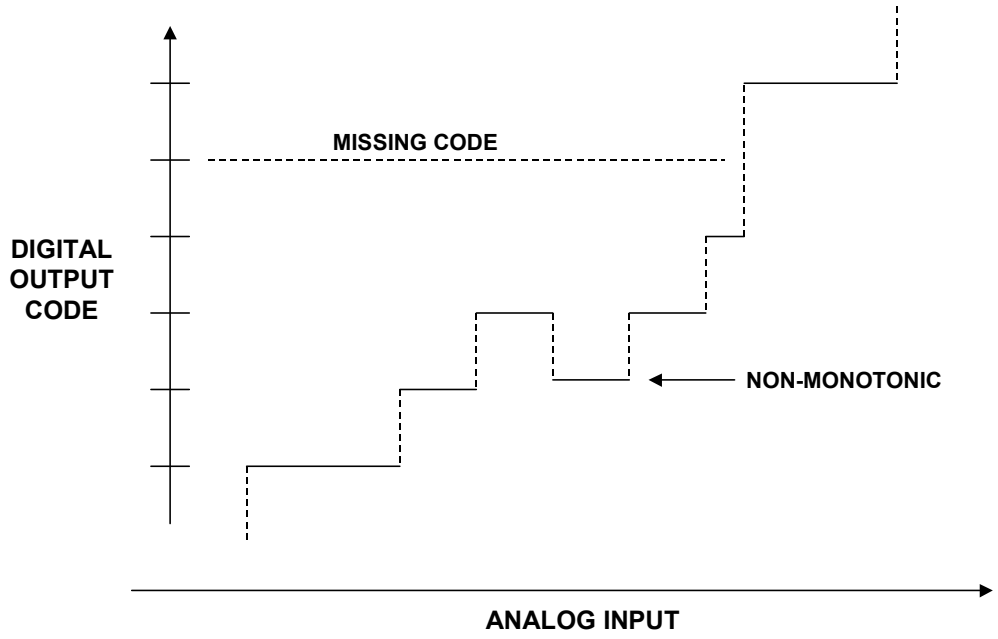


Figure 2.22: *Non-Monotonic ADC with Missing Code*

In a DAC, there can be no missing codes—each digital input word will produce a corresponding analog output. However, DACs can be non-monotonic as previously discussed. In a straight binary DAC, the most likely place a non-monotonic condition can develop is at mid-scale between the two codes: 011…11 and 100…00. If a non-monotonic conditions occurs here, it is generally because the DAC is not properly calibrated or trimmed. A successive approximation ADC with an internal non-monotonic DAC will generally produce missing codes but remain monotonic. However it is possible for an ADC to be non-monotonic—again depending on the particular conversion architecture. Figure 2.22 shows the transfer function of an ADC which is non-monotonic and has a missing code.

ADCs which use the *subranging* architecture divide the input range into a number of coarse segments, and each coarse segment is further divided into smaller segments—and ultimately the final code is derived. This process is described in more detail in Chapter 4 of this book. An improperly trimmed subranging ADC may exhibit non-monotonicity, wide codes, or missing codes at the subranging points as shown in Figure 2.23 A, B, and C, respectively. This type of ADC should be trimmed so that drift due to aging or temperature produces wide codes at the sensitive points rather than non-monotonic or missing codes.

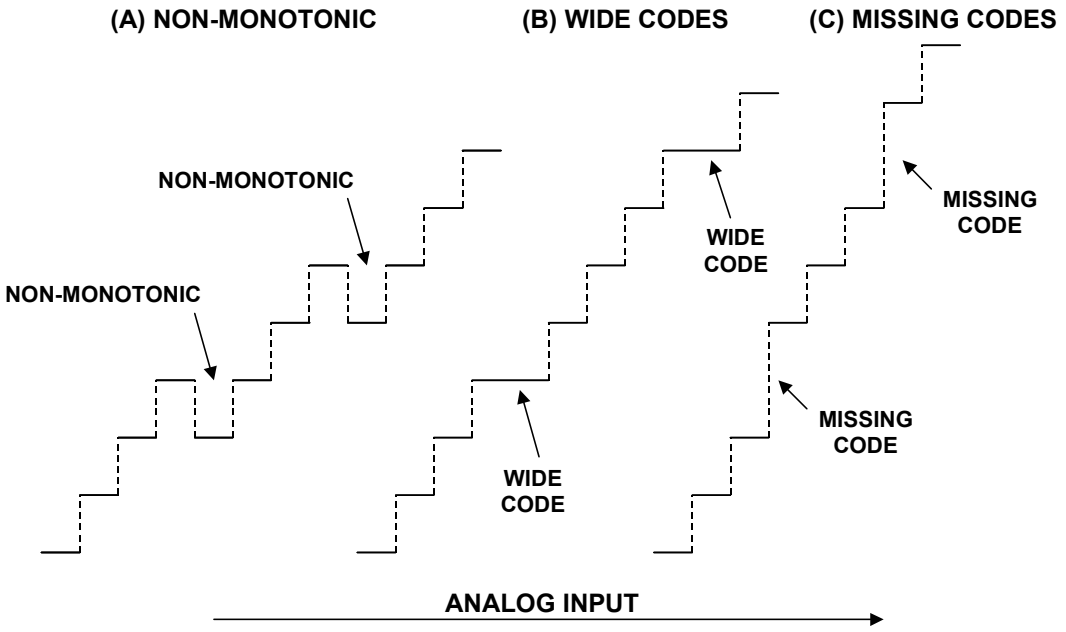


Figure 2.23: *Errors Associated with Improperly Trimmed Subranging ADC*

Defining missing codes is more difficult than defining non-monotonicity. All ADCs suffer from some inherent transition noise as shown in Figure 2.24 (think of it as the flicker between adjacent values of the last digit of a DVM). As resolutions and bandwidths become higher, the range of input over which transition noise occurs may approach, or even exceed, 1 LSB. High resolution wideband ADCs generally have internal noise sources which can be reflected to the input as effective input noise summed with the signal. The effect of this noise, especially if combined with a negative DNL error, may be that there are some (or even all) codes where transition noise is present for the whole range of inputs. There are therefore some codes for which there is *no* input which will *guarantee* that code as an output, although there may be a range of inputs which will *sometimes* produce that code.

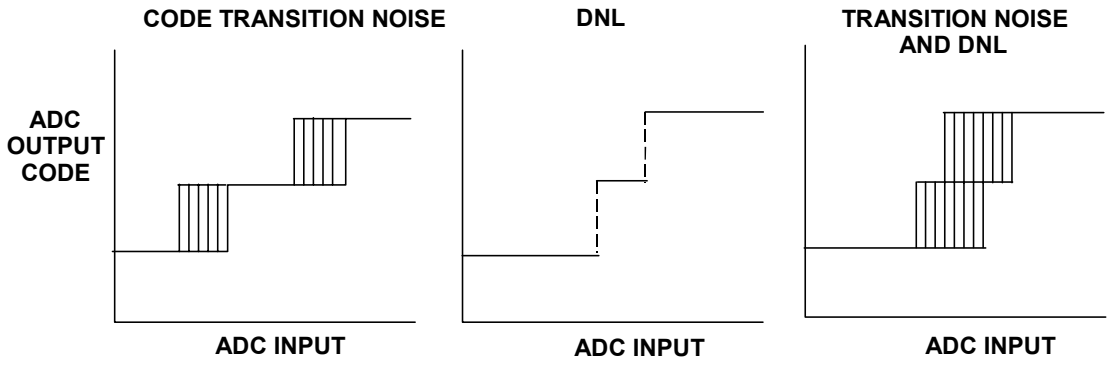


Figure 2.24: *Combined Effects of Code Transition Noise and DNL*

For low resolution ADCs, it may be reasonable to define *no missing codes* as a combination of transition noise and DNL which guarantees some level (perhaps 0.2 LSB) of noise-free code for all codes. However, this is impossible to achieve at the very high resolutions achieved by modern sigma-delta ADCs, or even at lower resolutions in wide bandwidth sampling ADCs. In these cases, the manufacturer must define noise levels and resolution in some other way. Which method is used is less important, but the data sheet should contain a clear definition of the method used and the performance to be expected. A complete discussion of effective input noise follows in Section 2.3 of this chapter.

The discussion thus far has only dealt with the most important dc specifications associated with data converters. Other less important specifications require only a definition. For specifications not covered in this section, the reader is referred to Section 2.5 of this chapter for a complete alphabetical listing of data converter specifications along with their definitions.

REFERENCES:
2.1 CODING AND QUANTIZATION

1. K. W. Cattermole, **Principles of Pulse Code Modulation**, American Elsevier Publishing Company, Inc., 1969, New York NY, ISBN 444-19747-8. *(An excellent tutorial and historical discussion of data conversion theory and practice, oriented towards PCM, but covers practically all aspects. This one is a must for anyone serious about data conversion!*

2. Frank Gray, "Pulse Code Communication," **U.S. Patent 2,632,058**, filed November 13, 1947, issued March 17, 1953. *(detailed patent on the Gray code and its application to electron beam coders)*.

3. R. W. Sears, "Electron Beam Deflection Tube for Pulse Code Modulation," **Bell System Technical Journal**, Vol. 27, pp. 44-57, Jan. 1948. *(describes an electon-beam deflection tube 7-bit,100kSPS flash converter for early experimental PCM work).*

4. J. O. Edson and H. H. Henning, "Broadband Codecs for an Experimental 224Mb/s PCM Terminal," **Bell System Technical Journal**, Vol. 44, pp. 1887-1940, Nov. 1965. *(summarizes experiments on ADCs based on the electron tube coder as well as a bit-per-stage Gray code 9-bit solid state ADC. The electron beam coder was 9-bits at 12MSPS, and represented the fastest of its type).*

5. Dan Sheingold, **Analog-Digital Conversion Handbook, 3rd Edition**, Analog Devices and Prentice-Hall, 1986, ISBN-0-13-032848-0. *(the defining and classic book on data conversion).*

SECTION 2.2: SAMPLING THEORY

Walt Kester

This section discusses the basics of sampling theory. A block diagram of a typical real-time sampled data system is shown in Figure 2.25. Prior to the actual analog-to-digital conversion, the analog signal usually passes through some sort of signal conditioning circuitry which performs such functions as amplification, attenuation, and filtering. The lowpass/bandpass filter is required to remove unwanted signals outside the bandwidth of interest and prevent aliasing.

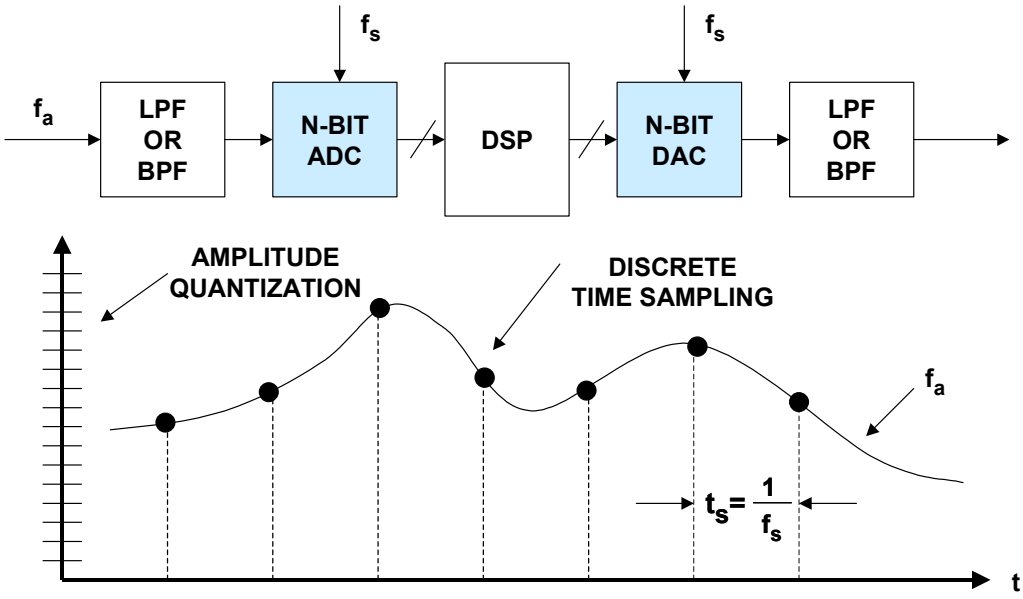


Figure 2.25: *Sampled Data System*

The system shown in Figure 2.25 is a real-time system, i.e., the signal to the ADC is continuously sampled at a rate equal to f_s, and the ADC presents a new sample to the DSP at this rate. In order to maintain real-time operation, the DSP must perform all its required computation within the sampling interval, $1/f_s$, and present an output sample to the DAC before arrival of the next sample from the ADC. An example of a typical DSP function would be a digital filter.

In the case of FFT analysis, a block of data is first transferred to the DSP memory. The FFT is calculated at the same time a new block of data is transferred into the memory, in order to maintain real-time operation. The DSP must calculate the FFT during the data transfer interval so it will be ready to process the next block of data.

Note that the DAC is required only if the DSP data must be converted back into an analog signal (as would be the case in a voiceband or audio application, for example). There are many applications where the signal remains entirely in digital format after the initial A/D conversion. Similarly, there are applications where the DSP is solely responsible for generating the signal to the DAC. If a DAC is used, it must be followed by an analog anti-imaging filter to remove the image frequencies. Finally, there are

slower speed industrial process control systems where sampling rates are much lower—regardless of the system, the fundamentals of sampling theory still apply.

There are two key concepts involved in the actual analog-to-digital and digital-to-analog conversion process: *discrete time sampling* and *finite amplitude resolution due to quantization.* An understanding of these concepts is vital to data converter applications.

The Need for a Sample-and-Hold Amplifier (SHA) Function

The generalized block diagram of a sampled data system shown in Figure 2.25 assumes some type of ac signal at the input. It should be noted that this does not necessarily have to be so, as in the case of modern digital voltmeters (DVMs) or ADCs optimized for dc measurements, but for this discussion assume that the input signal has some upper frequency limit f_a.

Most ADCs today have a built-in sample-and-hold function, thereby allowing them to process ac signals. This type of ADC is referred to as a *sampling ADC*. However many early ADCs, such as Analog Devices' industry-standard AD574, were not of the sampling type, but simply *encoders* as shown in Figure 2.26. If the input signal to a SAR ADC (assuming no SHA function) changes by more than 1 LSB during the conversion time (8µs in the example), the output data can have large errors, depending on the location of the code. Most ADC architectures are subject to this type of error—some more, some less—with the possible exception of flash converters having well-matched comparators.

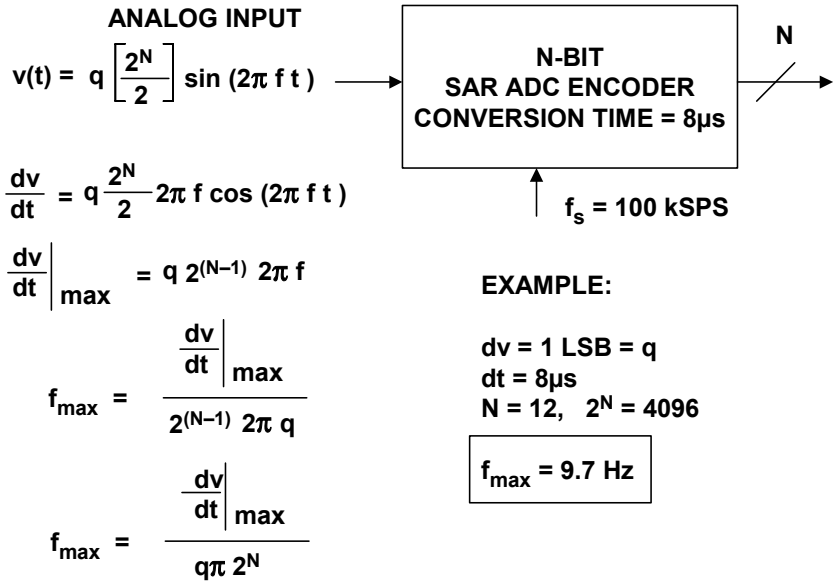


Figure 2.26: *Input Frequency Limitations of Non-Sampling ADC (Encoder)*

Assume that the input signal to the encoder is a sinewave with a full-scale amplitude ($q2^N/2$), where q is the weight of 1 LSB.

$$v(t) = q\,(2^N/2)\sin(2\pi f t). \qquad \text{Eq. 2.1}$$

Taking the derivative:

$$dv/dt = q\ 2\pi f\ (2^N/2)\cos(2\pi f t). \quad \text{Eq. 2.2}$$

The maximum rate of change is therefore:

$$dv/dt\ |_{max} = q\ 2\pi f\ (2^N/2). \quad \text{Eq. 2.3}$$

Solving for f:

$$f = (dv/dt\ |_{max})/(q\ \pi\ 2^N). \quad \text{Eq. 2.4}$$

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

$$f_{max} = 9.7\ \text{Hz}.$$

This implies any input frequency greater than 9.7 Hz is subject to conversion errors, even though a sampling frequency of 100 kSPS is possible with the 8-µs ADC (this allows an extra 2µs interval for an external SHA to re-acquire the signal after coming out of the hold mode).

To process ac signals, a sample-and-hold function is added as shown in Figure 2.27. The ideal SHA is simply a switch driving a hold capacitor followed by a high input impedance buffer. The input impedance of the buffer must be high enough so that the capacitor is discharged by less than 1 LSB during the hold time. The SHA samples the signal in the *sample* mode, and holds the signal constant during the *hold* mode. The timing is adjusted so that the encoder performs the conversion during the hold time. A sampling ADC can therefore process fast signals—the upper frequency limitation is determined by the SHA aperture jitter, bandwidth, distortion, etc., not the encoder. In the example shown, a good sample-and-hold could acquire the signal in 2 µs, allowing a sampling frequency of 100 kSPS, and the capability of processing input frequencies up to 50 kHz. A complete discussion of the SHA function including these specifications follows later in this chapter.

It is important to understand a subtle difference between a true *sample-and-hold* amplifier (SHA) and a *track-and-hold* amplifier (T/H, or THA). Strictly speaking, the output of a sample-and-hold is not defined during the sample mode, however the output of a track-and-hold tracks the signal during the sample or *track* mode. In practice, the function is generally implemented as a track-and-hold, and the terms *track-and-hold* and *sample-and-hold* are often used interchangeably. The waveforms shown in Figure 2.27 are those associated with a track-and-hold.

In order to better understand the types of ac errors an ADC can make without a sample-and-hold function, consider Figure 2.28. The photos show the reconstructed output of an 8-bit ADC (flash converter) with and without the sample-and-hold function. In an ideal flash converter the comparators are perfectly matched, and no sample-and-hold is required. In practice, however, there are timing mismatches between the comparators

which cause high-frequency inputs to exhibit nonlinearities and missing codes as shown in the right-hand photos. The data was taken by driving a DAC with the ADC output. The DAC output is a low frequency aliased sinewave corresponding to the difference between the sampling frequency (20 MSPS) and the ADC input frequency (19.98 MHz). In this case, the alias frequency is 20 kHz. (Aliasing is explained in detail in the next section).

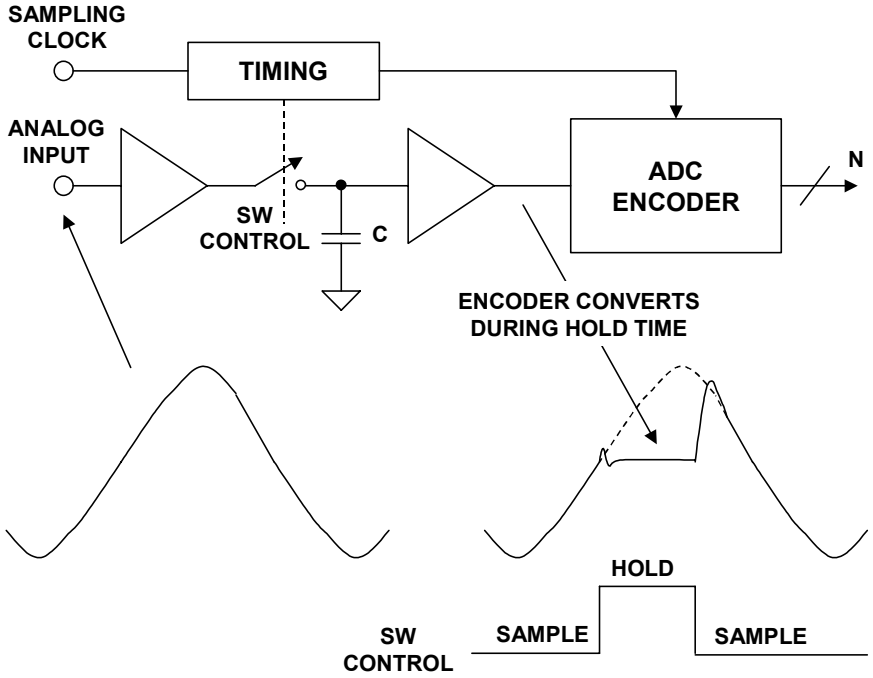


Figure 2.27: *Sample-and-Hold Function Required for Digitizing AC Signals*

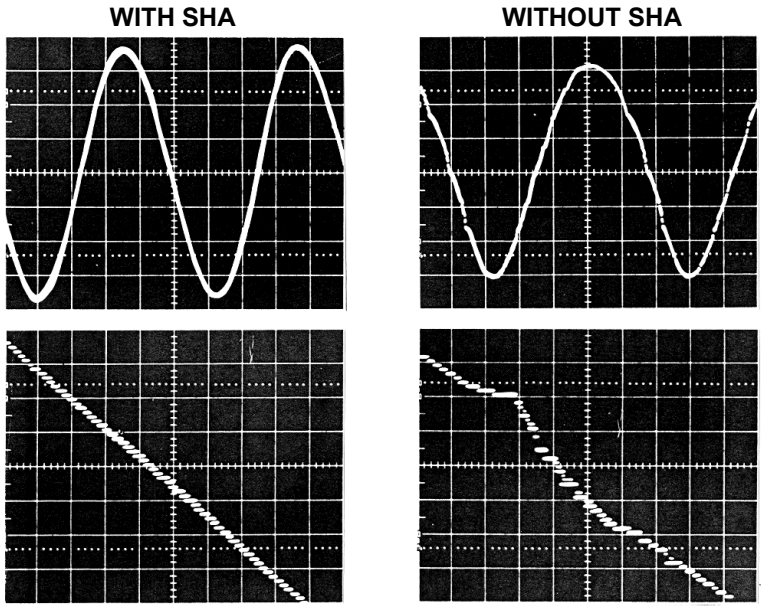


Figure 2.28: *8-bit, 20-MSPS Flash ADC With and Without Sample-and-Hold*

The Nyquist Criteria

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

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

- **A signal with a *maximum frequency* f_a must be sampled at a rate $f_s > 2f_a$ or information about the signal will be lost because of aliasing.**
- **Aliasing occurs whenever $f_s < 2f_a$**
- **The concept of aliasing is widely used in communications applications such as direct IF-to-digital conversion.**
- **A signal which has frequency components between f_a and f_b must be sampled at a rate $f_s > 2 (f_b – f_a)$ in order to prevent alias components from overlapping the signal frequencies.**

Figure 2.29: *Nyquist's Criteria*

In order to understand the implications of *aliasing* in both the time and frequency domain, first consider the case of a time domain representation of a single tone sinewave sampled as shown in Figure 2.30. In this example, the sampling frequency f_s is not at least $2f_a$, but only slightly more than the analog input frequency f_a—the Nyquist criteria is violated. Notice that the pattern of the actual samples produces an *aliased* sinewave at a lower frequency equal to $f_s – f_a$.

The corresponding frequency domain representation of this scenario is shown in Figure 2.31B. Now consider the case of a single frequency sinewave of frequency f_a sampled at a frequency f_s by an ideal impulse sampler (see Figure 2.31A). Also assume that $f_s > 2f_a$ as shown. The frequency-domain output of the sampler shows *aliases* or *images* of the original signal around every multiple of f_s, i.e. at frequencies equal to $|\pm Kf_s \pm f_a|$, K = 1, 2, 3, 4, ...

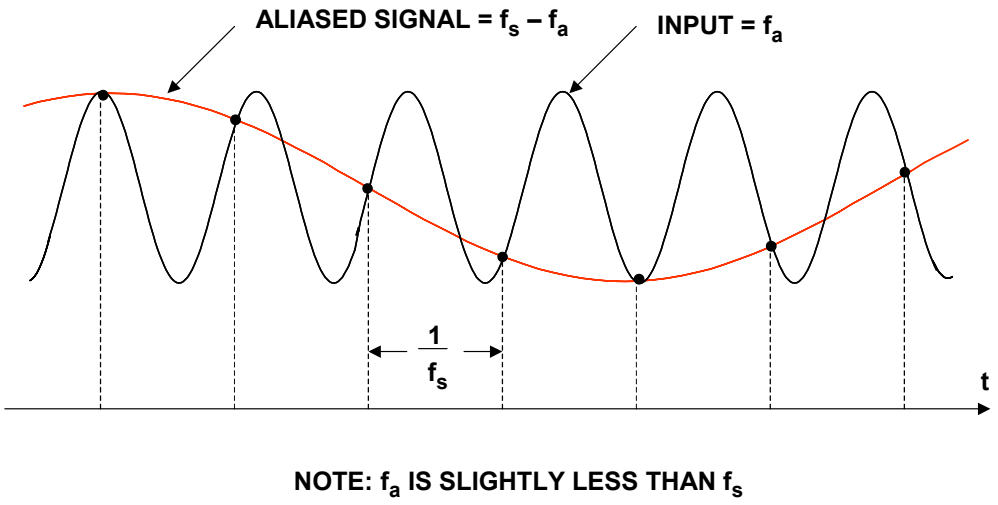


Figure 2.30: *Aliasing in the Time Domain*

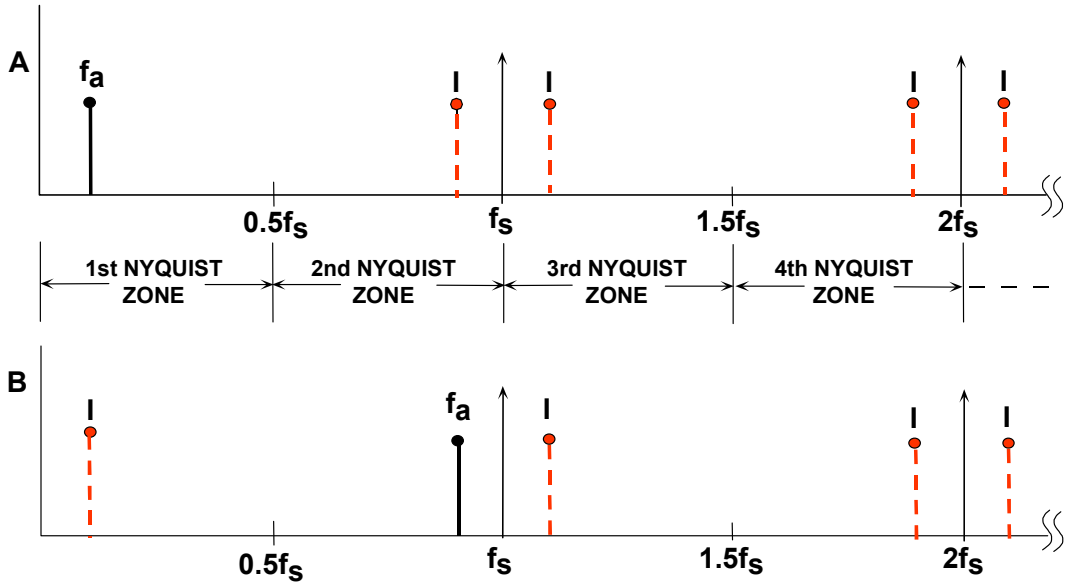


Figure 2.31: *Analog Signal f_a Sampled @ f_s Using Ideal Sampler Has Images (Aliases) at $|\pm Kf_s \pm f_a|$, K = 1, 2, 3, . . .*

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

Now consider the case of a signal which is outside the first Nyquist zone (Figure 2.31B). The signal frequency is only slightly less than the sampling frequency, corresponding to the condition shown in the time domain representation in Figure 2.30. Notice that even though the signal is outside the first Nyquist zone, its image (or *alias*), $f_s - f_a$, falls inside. Returning to Figure 2.31A, it is clear that if an unwanted signal appears at any of the

image frequencies of f_a, it will also occur at f_a, thereby producing a spurious frequency component in the first Nyquist zone.

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

Baseband Antialiasing Filters

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

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

Assume that the corner frequency of the filter is chosen to be equal to f_a. The effect of the finite transition from minimum to maximum attenuation on system dynamic range is illustrated in Figure 2.32A.

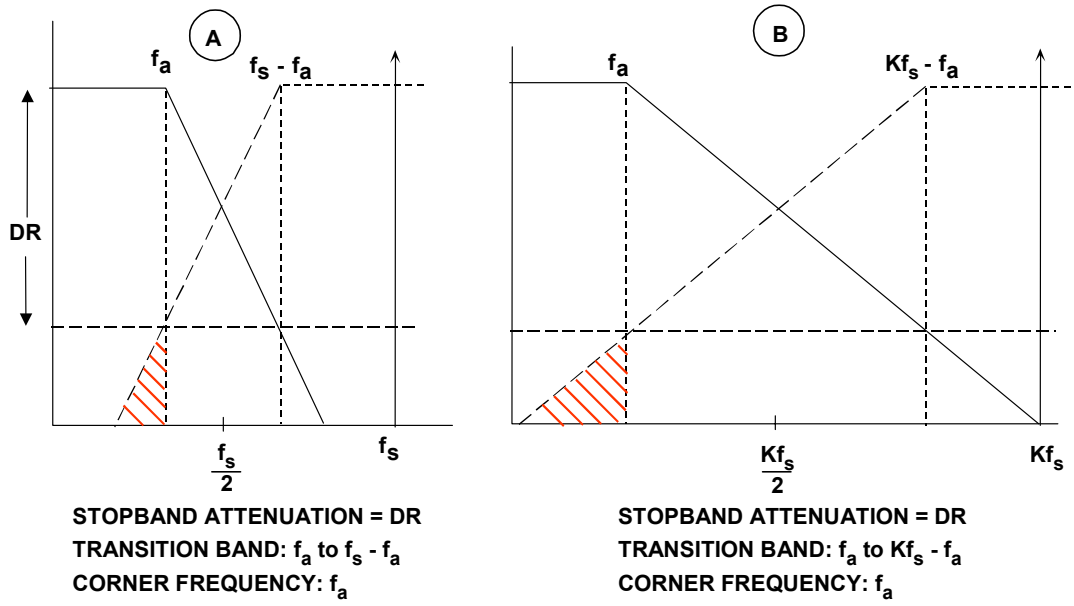


Figure 2.32: *Oversampling Relaxes Requirements on Baseband Antialiasing Filter*

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

Some texts recommend specifying the antialiasing filter with respect to the Nyquist frequency, $f_s/2$, but this assumes that the signal bandwidth of interest extends from dc to $f_s/2$ which is rarely the case. In the example shown in Figure 2.32A, the aliased components between f_a and $f_s/2$ are not of interest and do not limit the dynamic range.

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

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

Therefore, other filter types are generally more suited to applications where the requirement is for a sharp transition band and in-band flatness coupled with linear phase response. Elliptic filters meet these criteria and are a popular choice. There are a number of companies which specialize in supplying custom analog filters. TTE is an example of such a company (Reference 5).

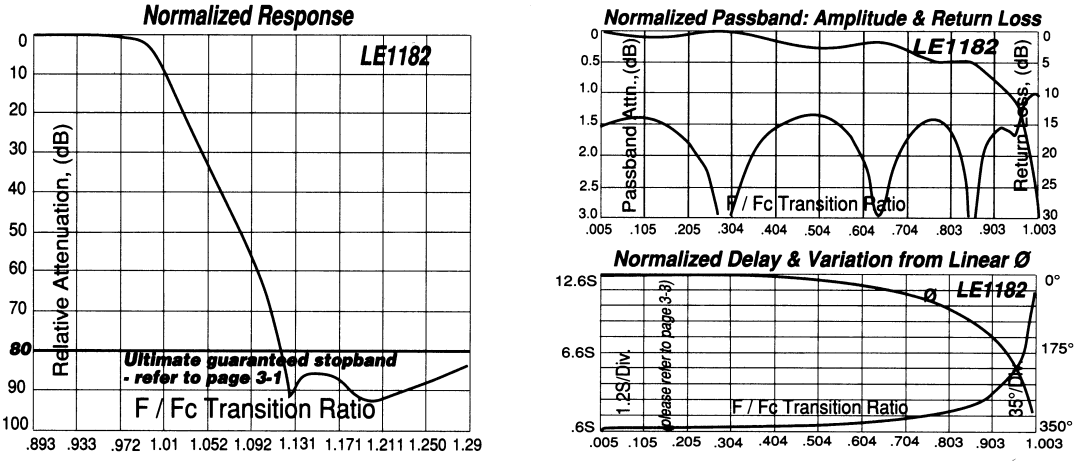
From this discussion, we can see how the sharpness of the antialiasing transition band can be traded off against the ADC sampling frequency. Choosing a higher sampling rate (oversampling) reduces the requirement on transition band sharpness (hence, the filter complexity) at the expense of using a faster ADC and processing data at a faster rate. This is illustrated in Figure 2.32B which shows the effects of increasing the sampling frequency by a factor of K, while maintaining the same analog corner frequency, f_a, and the same dynamic range, DR, requirement. The wider transition band (f_a to $Kf_s - f_a$) makes this filter easier to design than for the case of Figure 2.32A.

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

The antialiasing filter requirements can also be relaxed somewhat if it is certain that there will never be a full-scale signal at the stopband frequency $f_s - f_a$. In many applications, it is improbable that full-scale signals will occur at this frequency. If the maximum signal at the frequency $f_s - f_a$ will never exceed X dB below full-scale, then the filter stopband attenuation requirement can be reduced by that same amount. The new requirement for

stopband attenuation at $f_s - f_a$ based on this knowledge of the signal is now only DR – X dB. When making this type of assumption, be careful to treat any noise signals which may occur above the maximum signal frequency f_a as unwanted signals which will also alias back into the signal bandwidth.

There are a number of companies which specialize in supplying custom analog filters. TTE is an example of such a company (Reference 5). As an example, the normalized response of the TTE, Inc., LE1182 11-pole elliptic antialiasing filter is shown in Figure 2.33. Notice that this filter is specified to achieve at least 80 dB attenuation between f_c and $1.2f_c$. The corresponding passband ripple, return loss, delay, and phase response are also shown in Figure 2.33. This custom filter is available in corner frequencies up to 100 MHz and in a choice of PC board, BNC, or SMA with compatible packages.



Reprinted with Permission of TTE, Inc.,
11652 Olympic Blvd., Los Angeles CA 90064
http://www.tte.com

Figure 2.33: *Characteristics of 11-Pole Elliptical Filter (TTE, Inc., LE1182-Series)*

Undersampling (Harmonic Sampling, Bandpass Sampling, IF Sampling, Direct IF-to-Digital Conversion)

Thus far we have considered the case of baseband sampling, i.e., all the signals of interest lie within the first Nyquist zone. Figure 2.34A shows such a case, where the band of sampled signals is limited to the first Nyquist zone, and images of the original band of frequencies appear in each of the other Nyquist zones.

Consider the case shown in Figure 2.34B, where the sampled signal band lies entirely within the second Nyquist zone. The process of sampling a signal outside the first Nyquist zone is often referred to as *undersampling*, or *harmonic sampling*. Note that the image which falls in the first Nyquist zone contains all the information in the original signal, with the exception of its original location (the order of the frequency components within the spectrum is reversed, but this is easily corrected by re-ordering the output of the FFT).

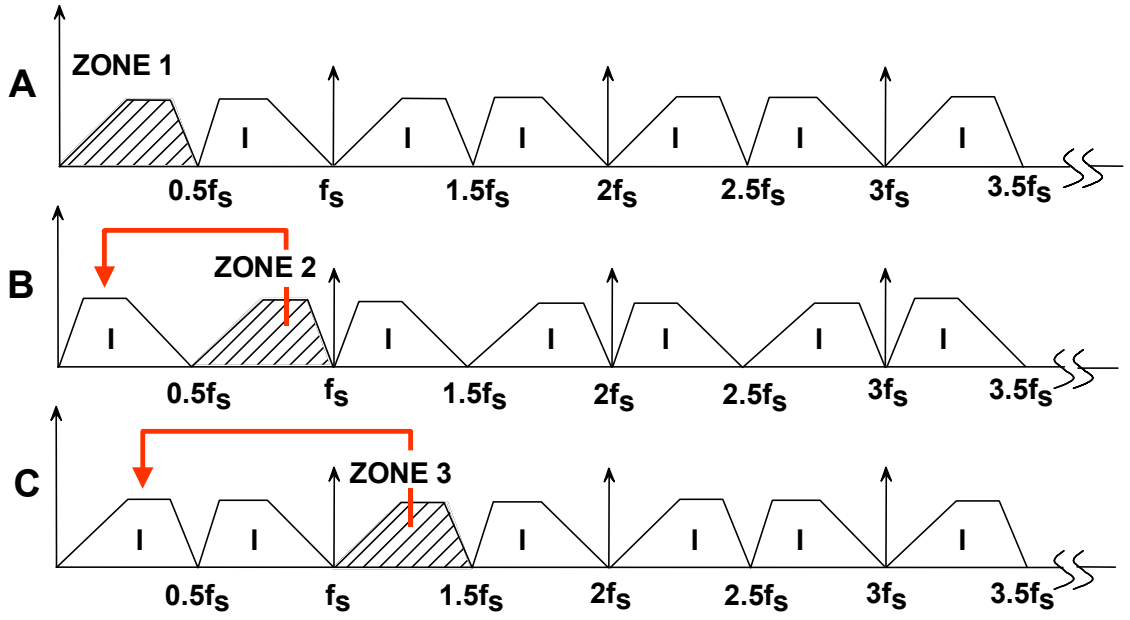


Figure 2.34: *Undersampling and Frequency Translation Between Nyquist Zones*

Figure 2.34C shows the sampled signal restricted to the third Nyquist zone. Note that the image that falls into the first Nyquist zone has no frequency reversal. In fact, the sampled signal frequencies may lie in *any* unique Nyquist zone, and the image falling into the first Nyquist zone is still an accurate representation (with the exception of the frequency reversal which occurs when the signals are located in even Nyquist zones). At this point we can clearly restate the Nyquist criteria:

A signal must be sampled at a rate equal to or greater than twice its ***bandwidth*** *in order to preserve all the signal information.*

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

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

Antialiasing Filters in Undersampling Applications

Figure 2.35 shows a signal in the second Nyquist zone centered around a carrier frequency, f_c, whose lower and upper frequencies are f_1 and f_2. The antialiasing filter is a bandpass filter. The desired dynamic range is DR, which defines the filter stopband attenuation. The upper transition band is f_2 to $2f_s - f_2$, and the lower is f_1 to $f_s - f_1$. As in the case of baseband sampling, the antialiasing filter requirements can be relaxed by proportionally increasing the sampling frequency, but f_c must also be increased so that it is always centered in the second Nyquist zone.

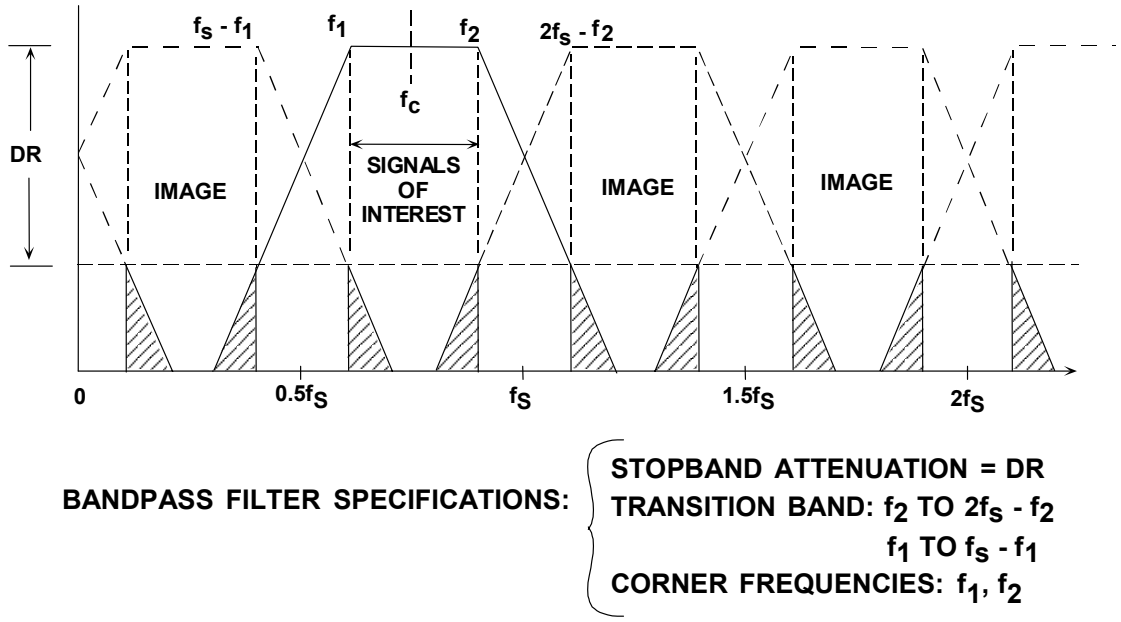


Figure 2.35: *Antialiasing Filter for Undersampling*

Two key equations can be used to select the sampling frequency, f_s, given the carrier frequency, f_c, and the bandwidth of its signal, Δf. The first is the Nyquist criteria:

$$f_s > 2\Delta f\,. \qquad \text{Eq. 2.5}$$

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

$$f_S = \frac{4f_c}{2NZ - 1}, \qquad \text{Eq. 2.6}$$

where NZ = 1, 2, 3, 4, ... and NZ corresponds to the Nyquist zone in which the carrier and its signal fall (see Figure 2.36).

NZ is normally chosen to be as large as possible while still maintaining $f_s > 2\Delta f$. This results in the minimum required sampling rate. If NZ is chosen to be odd, then f_c and its signal will fall in an odd Nyquist zone, and the image frequencies in the first Nyquist zone will not be reversed. Tradeoffs can be made between the sampling frequency and

the complexity of the antialiasing filter by choosing smaller values of NZ (hence a higher sampling frequency).

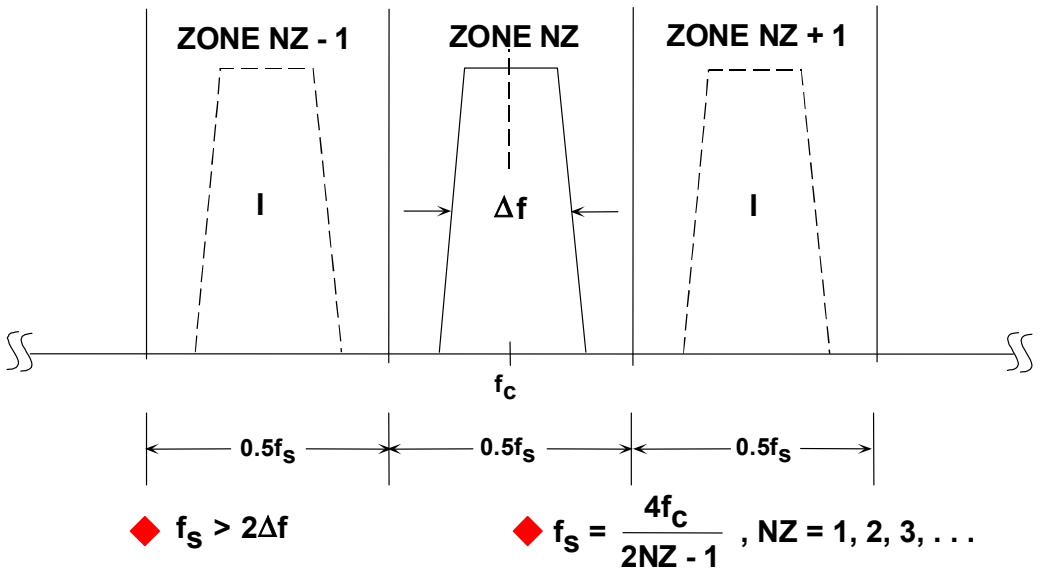


Figure 2.36: *Centering an Undersampled Signal within a Nyquist Zone*

As an example, consider a 4-MHz wide signal centered around a carrier frequency of 71 MHz. The minimum required sampling frequency is therefore 8 MSPS. Solving Eq. 2.6 for NZ using f_c = 71 MHz and f_s = 8 MSPS yields NZ = 18.25. However, NZ must be an integer, so we round 18.25 to the next lowest integer, 18. Solving Eq. 2.6 again for f_s yields f_s = 8.1143 MSPS. The final values are therefore f_s = 8.1143 MSPS, f_c = 71 MHz, and NZ = 18.

Now assume that we desire more margin for the antialiasing filter, and we select f_s to be 10 MSPS. Solving Eq. 2.6 for NZ, using f_c = 71 MHz and f_s = 10 MSPS yields NZ = 14.7. We round 14.7 to the next lowest integer, giving NZ = 14. Solving Eq. 2.6 again for f_s yields f_s = 10.519 MSPS. The final values are therefore f_s = 10.519 MSPS, f_c = 71 MHz, and NZ = 14.

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

REFERENCES:
2.2 SAMPLING THEORY

1. H. Nyquist, "Certain Factors Affecting Telegraph Speed," **Bell System Technical Journal**, Vol. 3, April 1924, pp. 324-346.

2. H.. Nyquist, Certain Topics in Telegraph Transmission Theory, **A.I.E.E. Transactions**, Vol. 47, April 1928, pp. 617-644.

3. R.V.L. Hartley, "Transmission of Information," **Bell System Technical Journal**, Vol. 7, July 1928, pp. 535-563.

4. C. E. Shannon, "A Mathematical Theory of Communication," **Bell System Technical Journal**, Vol. 27, July 1948, pp. 379-423 and October 1948, pp. 623-656.

5. TTE, Inc., 11652 Olympic Blvd., Los Angeles, CA 90064, http://www.tte.com.

NOTES:

SECTION 2.3: DATA CONVERTER AC ERRORS

Walt Kester, James Bryant

This section examines the ac errors associated with data converters. Many of the errors and specifications apply equally to ADCs and DACs, while some are more specific to one or the other. All possible specifications are not discussed here, only the most common ones. Section 2.4 of this Chapter contains a comprehensive listing of converter specifications as well as their definitions, including some not discussed in this section.

Theoretical Quantization Noise of an Ideal N-Bit Converter

The only errors (dc or ac) associated with an ideal N-bit data converter are those related to the sampling and quantization processes. The maximum error an ideal converter makes when digitizing a signal is ±½ LSB. The transfer function of an ideal N-bit ADC is shown in Figure 2.37. The quantization error for any ac signal which spans more than a few LSBs can be approximated by an uncorrelated sawtooth waveform having a peak-to-peak amplitude of q, the weight of an LSB. Although this analysis is not precise, it is accurate enough for most applications. W. R. Bennett of Bell Laboratories analyzed the actual spectrum of quantization noise in his classic 1948 paper (Reference 1). With certain simplifying assumptions, his detailed mathematical analysis simplifies to that of Figure 2.37. Other significant papers on converter noise (References 2-5) followed Bennett's classic publication.

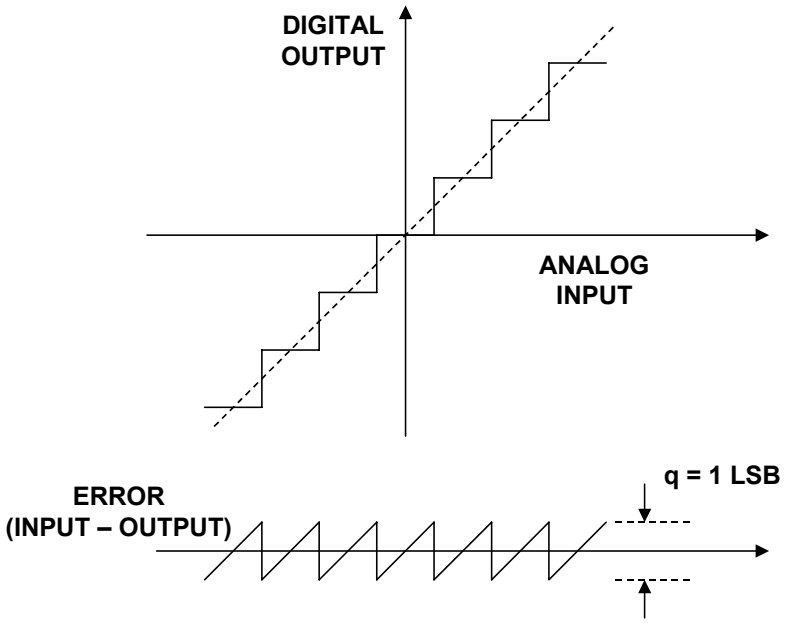


Figure 2.37: *Ideal N-bit ADC Quantization Noise*

The quantization error as a function of time is shown in Figure 2.38. Again, a simple sawtooth waveform provides a sufficiently accurate model for analysis. The equation of the sawtooth error is given by

$$e(t) = st, \; -q/2s < t < +q/2s. \qquad \text{Eq. 2.7}$$

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

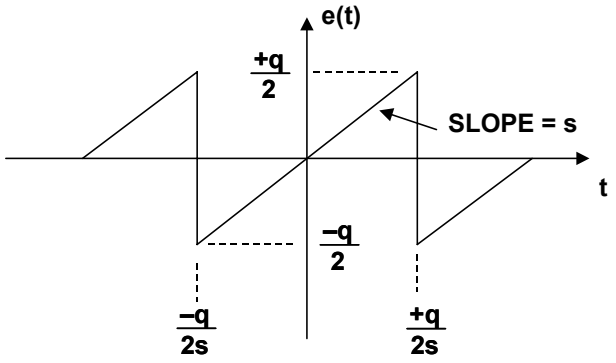
$$\overline{e^2(t)} = \frac{s}{q}\int_{-q/2s}^{+q/2s}(st)^2\,dt\,. \qquad \text{Eq. 2.8}$$

Performing the simple integration and simplifying,

$$\overline{e^2(t)} = \frac{q^2}{12}\,. \qquad \text{Eq. 2.9}$$

The root-mean-square quantization error is therefore

$$\text{rms quantization noise} = \sqrt{\overline{e^2(t)}} = \frac{q}{\sqrt{12}}\,. \qquad \text{Eq. 2.10}$$



- ERROR = $e(t) = st, \quad \frac{-q}{2s} < t < \frac{+q}{2s}$
- MEAN-SQUARE ERROR = $\overline{e^2(t)} = \frac{s}{q}\int_{-q/2s}^{+q/2s}(st)^2\,dt = \frac{q^2}{12}$
- ROOT-MEAN-SQUARE ERROR = $\sqrt{\overline{e^2(t)}} = \frac{q}{\sqrt{12}}$

Figure 2.38: *Quantization Noise as a Function of Time*

As Bennett points out (Reference 1), this noise is approximately Gaussian and spread more or less uniformly over the Nyquist bandwidth dc to $f_s/2$. The underlying assumption here is that the quantization noise is uncorrelated to the input signal. Under certain conditions where the sampling clock and the signal are harmonically related, the quantization noise becomes correlated and the energy is concentrated at the harmonics of the signal—the rms value remains approximately $q/\sqrt{12}$.

The theoretical signal-to-noise ratio can now be calculated assuming a full-scale input sinewave:

$$\text{Input FS Sinewave} = v(t) = \frac{q2^N}{2}\sin(2\pi ft). \qquad \text{Eq. 2.11}$$

The rms value of the input signal is therefore

$$\text{rms value of FS input} = \frac{q2^N}{2\sqrt{2}}. \qquad \text{Eq. 2.12}$$

The rms signal-to-noise ratio for an ideal N-bit converter is therefore

$$\text{SNR} = 20\log_{10}\frac{\text{rms value of FS input}}{\text{rms value of quantization noise}} \qquad \text{Eq. 2.13}$$

$$\text{SNR} = 20\log_{10}\left[\frac{q2^N/2\sqrt{2}}{q/\sqrt{12}}\right] = 6.02N + 1.76\ \text{dB}, \text{ over dc to } f_s/2 \text{ bandwidth.} \qquad \text{Eq. 2.14}$$

These relationships are summarized in Figure 2.39.

- **FS INPUT =** $v(t) = \left[\frac{q\,2^N}{2}\right] \sin(2\pi f t)$
- **RMS Value of FS Sinewave =** $\frac{q\,2^N}{2\sqrt{2}}$
- **RMS Value of Quantization Noise =** $\frac{q}{\sqrt{12}}$
- **SNR = 20 log$_{10}$** $\left[\frac{\text{RMS Value of FS Sinewave}}{\text{RMS Value of Quantization Noise}}\right] = 20\log_{10}2^N + 20\log_{10}\sqrt{\frac{3}{2}}$

SNR = 6.02N + 1.76dB

(Measured over the Nyquist Bandwidth : DC to $f_S/2$)

Figure 2.39: *Theoretical Signal-to-Quantization Noise Ratio of an Ideal N-Bit Converter*

Bennett's paper shows that although the actual spectrum of the quantization noise is quite complex to analyze—the simplified analysis which leads to Eq. 2.14 is accurate enough for most purposes. However, it is important to emphasize again that the rms quantization noise is measured over the full Nyquist bandwidth, dc to $f_s/2$. In many applications, the actual signal of interest occupies a smaller bandwidth, BW. If digital filtering is used to

filter out noise components outside the bandwidth BW, then a correction factor (called *process gain*) must be included in the equation to account for the resulting increase in SNR. The process of sampling a signal at a rate which is greater than twice its bandwidth is often referred to as *oversampling*. In fact, oversampling in conjunction with quantization noise shaping and digital filtering is a key concept in sigma-delta converters.

$$\text{SNR} = 6.02\text{N} + 1.76\ \text{dB} + 10\log_{10}\frac{f_s}{2\cdot \text{BW}},\ \text{over bandwidth BW.} \qquad \text{Eq. 2.15}$$

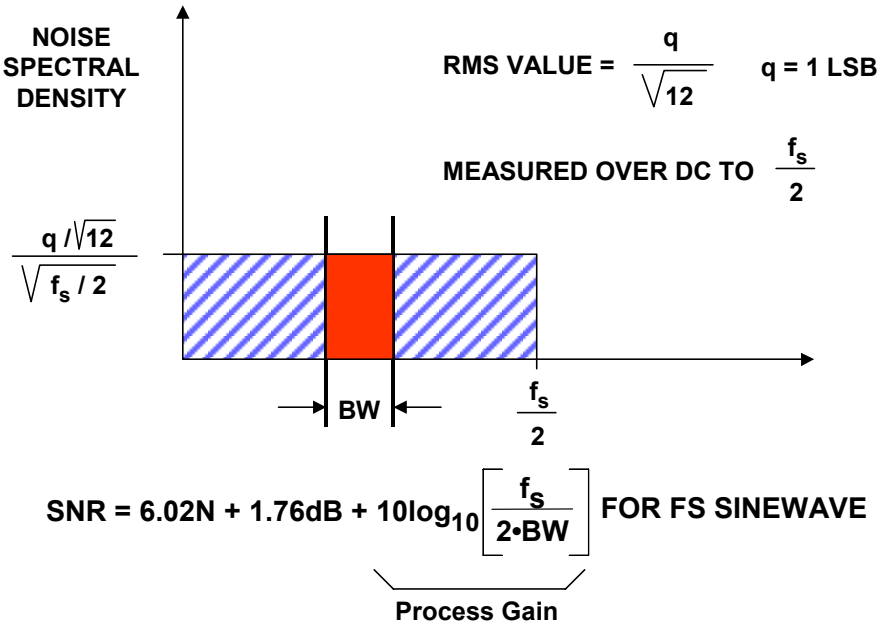


Figure 2.40: *Quantization Noise Spectrum*

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

$$\text{Process Gain} = 10\log_{10}\frac{f_s}{2\cdot \text{BW}} = 10\log_{10}\frac{65\times 10^6}{2\times 30\times 10^3} = 30.3\ \text{dB}. \qquad \text{Eq. 2.16}$$

The process gain is added to the ADC SNR specification to yield the actual SNR in the 30-kHz bandwidth. In the above example, if the ADC SNR specification is 65 dB (dc to $f_s/2$), then it is increased to 95.3 dB in the 30-kHz channel bandwidth (after appropriate digital filtering).

Figure 2.41 shows an application which combines oversampling and undersampling. The signal of interest has a bandwidth BW and is centered around a carrier frequency f_c. The sampling frequency can be much less than f_c and is chosen such that the signal of interest

is centered in its Nyquist zone. Analog and digital filtering removes the noise outside the signal bandwidth of interest, and therefore results in process gain per Eq. 2.16.

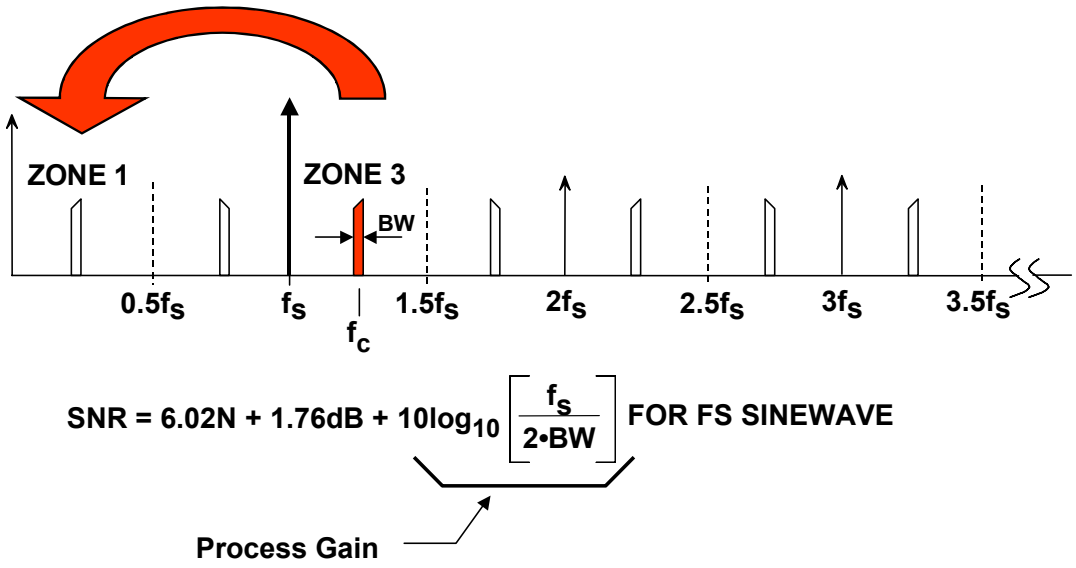


Figure 2.41: *Undersampling and Oversampling Combined Results in Process Gain*

Although the rms value of the noise is accurately approximated by $q/\sqrt{12}$, its frequency domain content may be highly correlated to the ac-input signal. For instance, there is greater correlation for low amplitude periodic signals than for large amplitude random signals. Quite often, the assumption is made that the theoretical quantization noise appears as white noise, spread uniformly over the Nyquist bandwidth dc to $f_s/2$. Unfortunately, this is not true in all cases. In the case of strong correlation, the quantization noise appears concentrated at the various harmonics of the input signal, just where you don't want them. Bennett (Reference 1) has an extensive analysis of the frequency content contained in the quantization noise spectrum in his classic 1948 paper.

In most practical applications, the input to the ADC is a band of frequencies (always summed with some unavoidable system noise), so the quantization noise tends to be random. In spectral analysis applications (or in performing FFTs on ADCs using spectrally pure sinewaves—see Figure 2.42), however, the correlation between the quantization noise and the signal depends upon the ratio of the sampling frequency to the input signal. This is demonstrated in Figure 2.43, where the output of an ideal 12-bit ADC is analyzed using a 4096-point FFT. In the left-hand FFT plot, the ratio of the sampling frequency to the input frequency was chosen to be exactly 32, and the worst harmonic is about 76 dB below the fundamental. The right hand diagram shows the effects of slightly offsetting the ratio to 4096/127 = 32.25196850394, showing a relatively random noise spectrum, where the SFDR is now about 92 dBc. In both cases, the rms value of all the noise components is approximately $q/\sqrt{12}$, but in the first case, the noise is concentrated at harmonics of the fundamental.

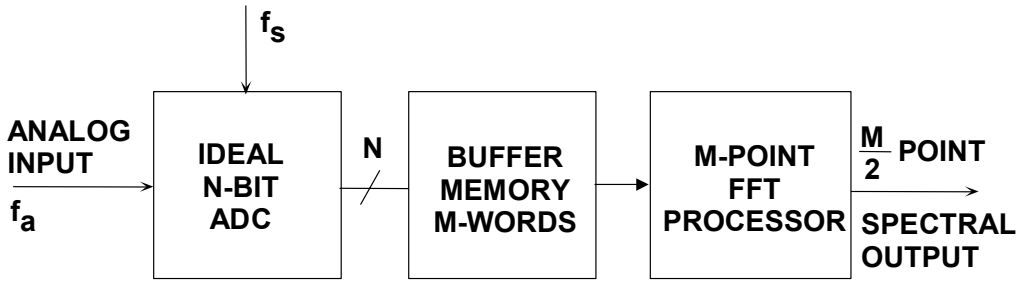


Figure 2.42: *Dynamic Performance Analysis of an Ideal N-bit ADC*

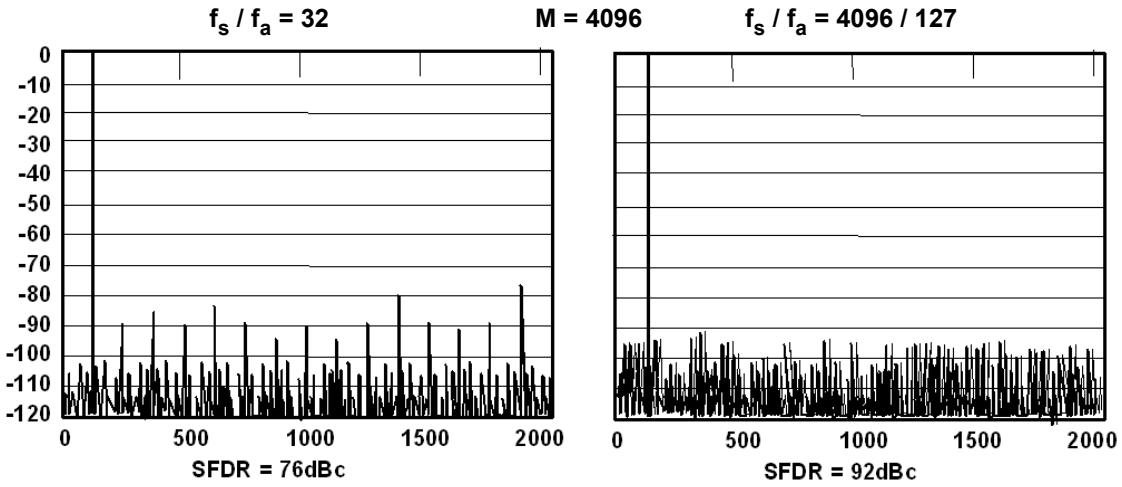


Figure 2.43: *Effect of Ratio of Sampling Clock to Input Frequency on SFDR for Ideal 12-bit ADC*

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

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

Figure 2.44 shows the FFT output for an ideal 12-bit ADC. Note that the average value of the noise floor of the FFT is approximately 100 dB below full-scale, but the theoretical SNR of a 12-bit ADC is 74 dB. The FFT noise floor is *not* the SNR of the ADC, because the FFT acts like an analog spectrum analyzer with a bandwidth of f_s/M, where M is the number of points in the FFT. The theoretical FFT noise floor is therefore $10\log_{10}(M/2)$ dB below the quantization noise floor due to the *processing gain* of the FFT. In the case

of an ideal 12-bit ADC with an SNR of 74 dB, a 4096-point FFT would result in a processing gain of $10\log_{10}(4096/2) = 33$ dB, thereby resulting in an overall FFT noise floor of 74 + 33 = 107 dBc. In fact, the FFT noise floor can be reduced even further by going to larger and larger FFTs; just as an analog spectrum analyzer's noise floor can be reduced by narrowing the bandwidth. When testing ADCs using FFTs, it is important to ensure that the FFT size is large enough so that the distortion products can be distinguished from the FFT noise floor itself.

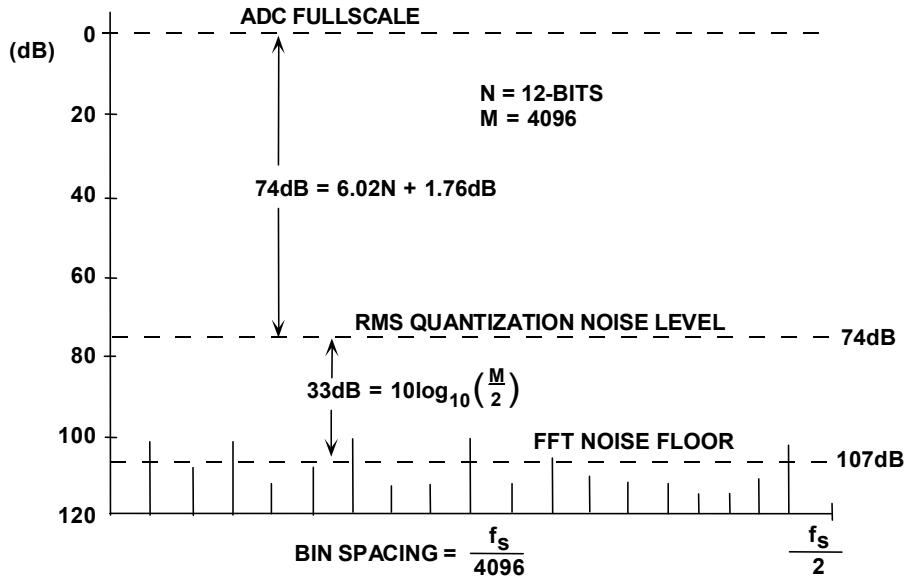


Figure 2.44: *Noise Floor for an Ideal 12-bit ADC Using 4096-point FFT*

Noise in Practical ADCs

A practical sampling ADC (one that has an integral sample-and-hold), regardless of architecture, has a number of noise and distortion sources as shown in Figure 2.45. The wideband analog front-end buffer has wideband noise, nonlinearity, and also finite bandwidth. The SHA introduces further nonlinearity, bandlimiting, and aperture jitter. The actual quantizer portion of the ADC introduces quantization noise, and both integral and differential nonlinearity. In this discussion, assume that sequential outputs of the ADC are loaded into a buffer memory of length M and that the FFT processor provides the spectral output. Also assume that the FFT arithmetic operations themselves introduce no significant errors relative to the ADC. However, when examining the output noise floor, the FFT processing gain (dependent on M) must be considered.

Equivalent Input Referred Noise

Wideband ADC internal circuits produce a certain amount of rms noise due to resistor noise and "kT/C" noise. This noise is present even for dc-input signals, and accounts for the fact that the output of most wideband (or high resolution) ADCs is a distribution of codes, centered around the nominal value of a dc input (see Figure 2.46). To measure its value, the input of the ADC is either grounded or connected to a heavily decoupled voltage source, and a large number of output samples are collected and plotted as a

histogram (sometimes referred to as a *grounded-input* histogram). Since the noise is approximately Gaussian, the standard deviation of the histogram is easily calculated (see Reference 6), corresponding to the effective input rms noise. It is common practice to express this rms noise in terms of LSBs rms, although it can be expressed as an rms voltage referenced to the ADC full-scale input range.

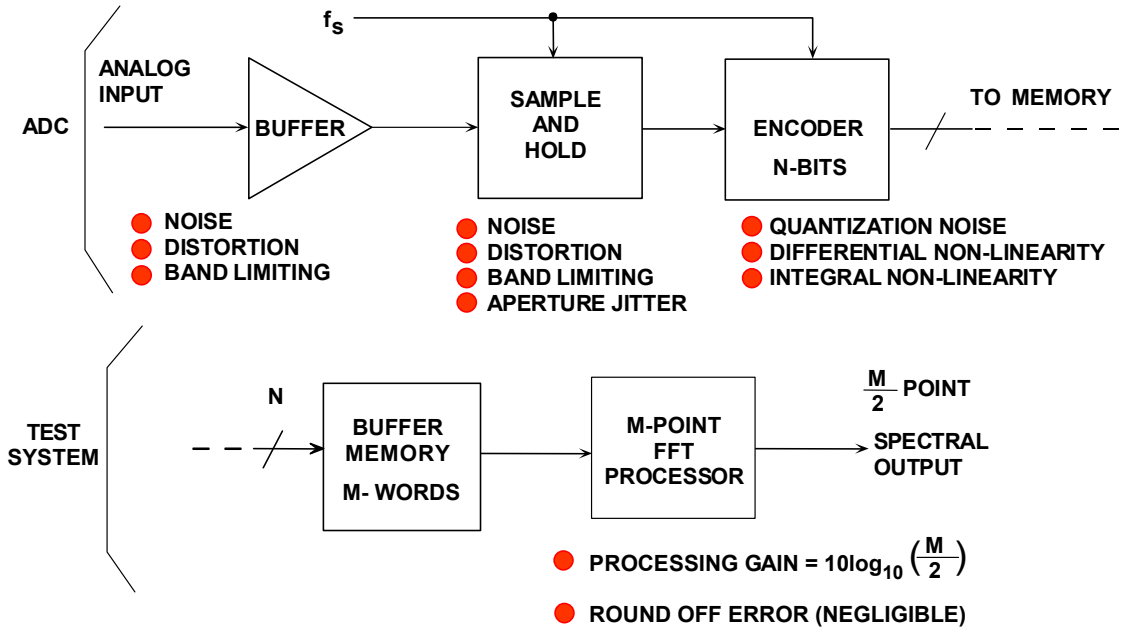


Figure 2.45: *ADC Model Showing Noise and Distortion Sources*

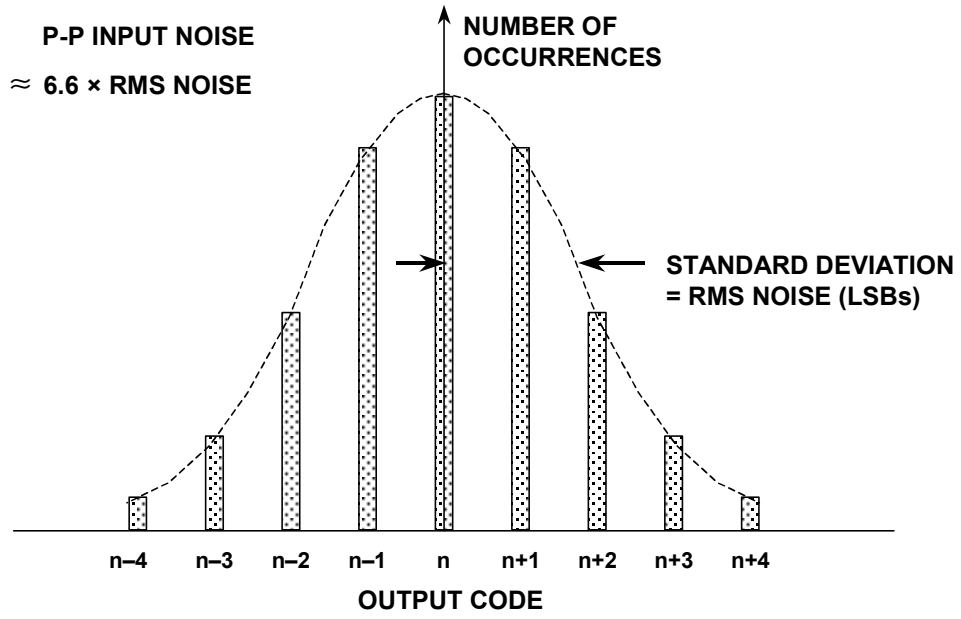


Figure 2.46: *Effect of Input-Referred Noise on ADC "Grounded Input" Histogram*

Noise-Free (Flicker-Free) Code Resolution

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

$$\text{Noise-Free Code Resolution} = \log_2 (2^N/\text{Peak-to-Peak Noise}). \qquad \text{Eq. 2.17}$$

The specification is generally associated with high-resolution sigma-delta measurement ADCs, but is applicable to all ADCs.

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

$$\text{Effective Resolution} = \log_2 (2^N/\text{RMS Input Noise}) \qquad \text{Eq. 2.18}$$

$$\text{Effective Resolution} = \text{Noise-Free Code Resolution} + 2.7 \text{ bits}. \qquad \text{Eq. 2.19}$$

The calculations are summarized in Figure 2.47.

- **Effective Input Noise = $e_{n\ rms}$**
- **Peak-to-Peak Input Noise = 6.6 $e_{n\ rms}$**
- **Noise-Free Code Resolution = $\log_2 \left[\dfrac{\text{Peak-to-Peak Input Range}}{\text{Peak-to-Peak Input Noise}} \right]$**

$$= \log_2 \left[\frac{2^N}{\text{Peak-to-Peak Input Noise (LSBs)}} \right]$$

- **"Effective Resolution" = $\log_2 \left[\dfrac{\text{Peak-to-Peak Input Range}}{\text{RMS Input Noise}} \right]$**

$$= \log_2 \left[\frac{2^N}{\text{RMS Input Noise (LSBs)}} \right]$$

= Noise-Free Code Resolution + 2.7 bits

Figure 2.47: *Calculating Noise-Free (Flicker-Free) Code Resolution from Input-Referred Noise*

Dynamic Performance of Data Converters

There are various ways to characterize the ac performance of ADCs. Before the 1970s, there was little standardization with respect to ac specifications, and measurement equipment and techniques were not well understood or available. Over nearly a 30 year period, manufacturers and customers have learned more about measuring the dynamic performance of converters, and the specifications shown in Figure 2.48 represent the most popular ones used today. Practically all the specifications represent the converter's performance in the frequency domain. The FFT is the heart of practically all these measurements and is discussed in more detail in Chapter 6 of this book.

Integral and Differential Nonlinearity Distortion Effects

One of the first things to realize when examining the nonlinearities of data converters is that the transfer function of a data converter has artifacts which do not occur in conventional linear devices such as op amps or gain blocks. The overall integral nonlinearity of an ADC is due to the integral nonlinearity of the front-end and SHA as well as the overall integral nonlinearity in the ADC transfer function. However, *differential nonlinearity is due exclusively to the encoding process* and may vary considerably dependent on the ADC encoding architecture. Overall integral nonlinearity produces distortion products whose amplitude varies as a function of the input signal amplitude. For instance, second-order intermodulation products increase 2 dB for every 1-dB increase in signal level, and third-order products increase 3 dB for every 1-dB increase in signal level.

- **Harmonic Distortion**
- **Worst Harmonic**
- **Total Harmonic Distortion (THD)**
- **Total Harmonic Distortion Plus Noise (THD + N)**
- **Signal-to-Noise-and-Distortion Ratio (SINAD, or S/N +D)**
- **Effective Number of Bits (ENOB)**
- **Signal-to-Noise Ratio (SNR)**
- **Analog Bandwidth (Full-Power, Small-Signal)**
- **Spurious Free Dynamic Range (SFDR)**
- **Two-Tone Intermodulation Distortion**
- **Multi-tone Intermodulation Distortion**
- **Noise Power Ratio (NPR)**
- **Adjacent Channel Leakage Ratio (ACLR)**
- **Noise Figure**
- **Settling Time, Overvoltage Recovery Time**

Figure 2.48: *Quantifying Data Converter Dynamic Performance*

The differential nonlinearity in the ADC transfer function produces distortion products which not only depend on the amplitude of the signal but the positioning of the differential nonlinearity errors along the ADC transfer function. Figure 2.49 shows two ADC transfer functions having differential nonlinearity. The left-hand diagram shows an error which occurs at mid-scale. Therefore, for both large and small signals, the signal crosses through this point producing a distortion product which is relatively independent of the signal amplitude. The right-hand diagram shows another ADC transfer function which has differential nonlinearity errors at 1/4 and 3/4 full-scale. Signals which are above 1/2 scale peak-to-peak will exercise these codes and produce distortion, while those less than 1/2 scale peak-to-peak will not.

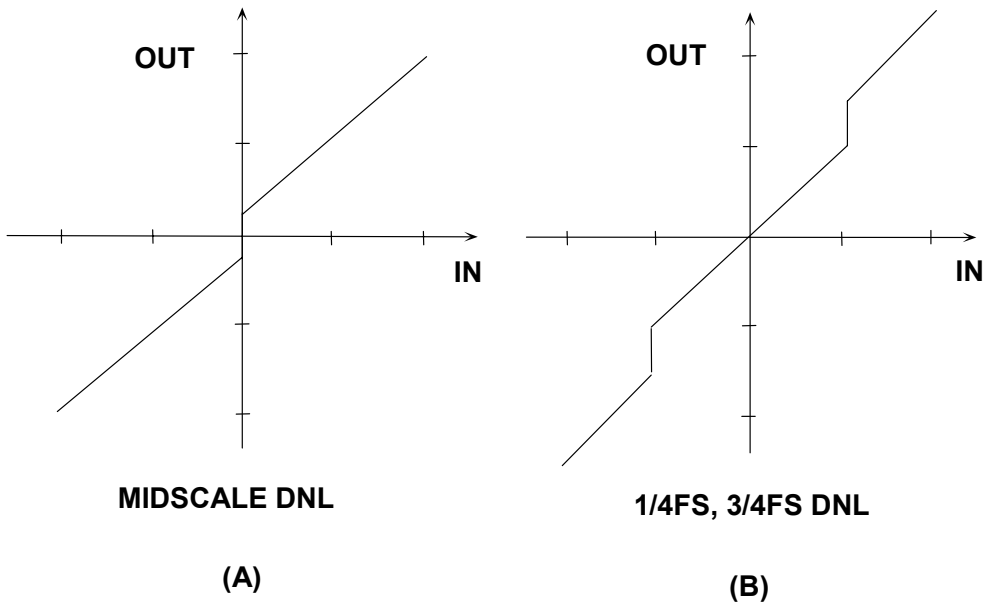


Figure 2.49: *Typical ADC/ DAC DNL Errors (Exaggerated)*

Most high-speed ADCs are designed so that differential nonlinearity is spread across the entire ADC range. Therefore, for signals which are within a few dB of full-scale, the overall integral nonlinearity of the transfer function determines the distortion products. For lower level signals, however, the harmonic content becomes dominated by the differential nonlinearities and does not generally decrease proportionally with decreases in signal amplitude.

Harmonic Distortion, Worst Harmonic, Total Harmonic Distortion (THD), Total Harmonic Distortion Plus Noise (THD + N)

There are a number of ways to quantify the distortion of an ADC. An FFT analysis can be used to measure the amplitude of the various harmonics of a signal. The harmonics of the input signal can be distinguished from other distortion products by their location in the frequency spectrum. Figure 2.50 shows a 7-MHz input signal sampled at 20 MSPS and the location of the first 9 harmonics. Aliased harmonics of f_a fall at frequencies equal to $|\pm Kf_s \pm nf_a|$, where n is the order of the harmonic, and K = 0, 1, 2, 3,.... The second and third harmonics are generally the only ones specified on a data sheet because they tend to be the largest, although some data sheets may specify the value of the *worst* harmonic.

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

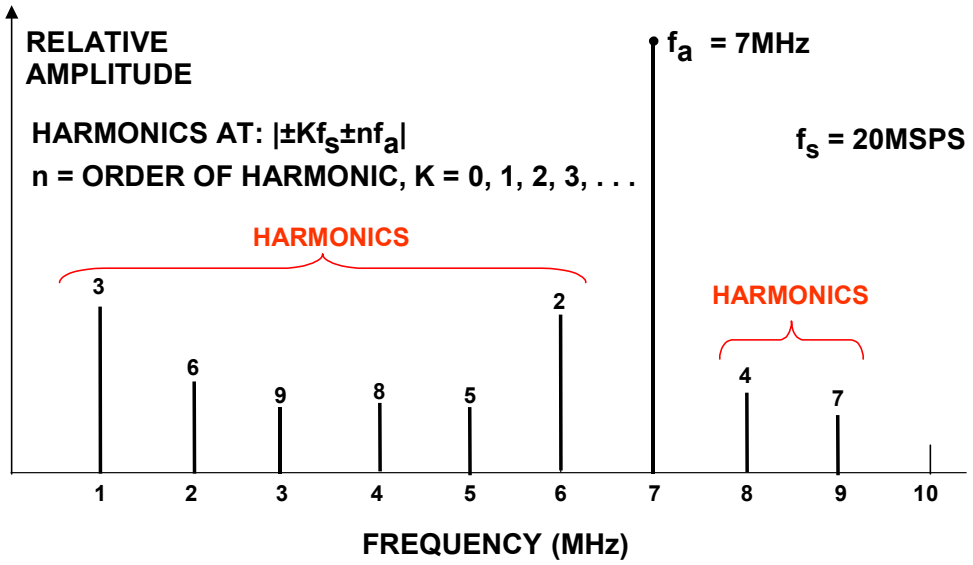


Figure 2.50: *Location of Distortion Products: Input Signal = 7 MHz, Sampling Rate = 20 MSPS*

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

Total harmonic distortion plus noise (THD + N) is the ratio of the rms value of the fundamental signal to the mean value of the root-sum-square of its harmonics plus all noise components (excluding dc). The bandwidth over which the noise is measured must be specified. In the case of an FFT, the bandwidth is dc to $f_s/2$. (If the bandwidth of the measurement is dc to $f_s/2$, THD + N is equal to SINAD—see below).

Signal-to-Noise-and-Distortion Ratio (SINAD), Signal-to-Noise Ratio (SNR), and Effective Number of Bits (ENOB)

SINAD and SNR deserve careful attention, because there is still some variation between ADC manufacturers as to their precise meaning. Signal-to-Noise-and Distortion (SINAD, or S/(N + D) is the ratio of the rms signal amplitude to the mean value of the root-sum-square (rss) of all other spectral components, *including harmonics*, but excluding dc (see Figure 2.50). SINAD is a good indication of the overall dynamic performance of an ADC as a function of input frequency because it includes all components which make up noise (including thermal noise) and distortion. It is often plotted for various input amplitudes. SINAD is equal to THD + N if the bandwidth for the noise measurement is the same. A typical plot for the AD9226 12-bit, 65-MSPS ADC is shown in Figure 2.52.

- **SINAD (Signal-to-Noise-and-Distortion Ratio):**
  - **The ratio of the rms signal amplitude to the mean value of the root-sum-squares (RSS) of all other spectral components, including harmonics, but excluding DC.**

- **ENOB (Effective Number of Bits):**

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

- **SNR (Signal-to-Noise Ratio, or Signal-to-Noise Ratio Without Harmonics:**
  - **The ratio of the rms signal amplitude to the mean value of the root-sum-squares (RSS) of all other spectral components, excluding the first 5 harmonics and DC**

Figure 2.51: *SINAD, ENOB, and SNR*

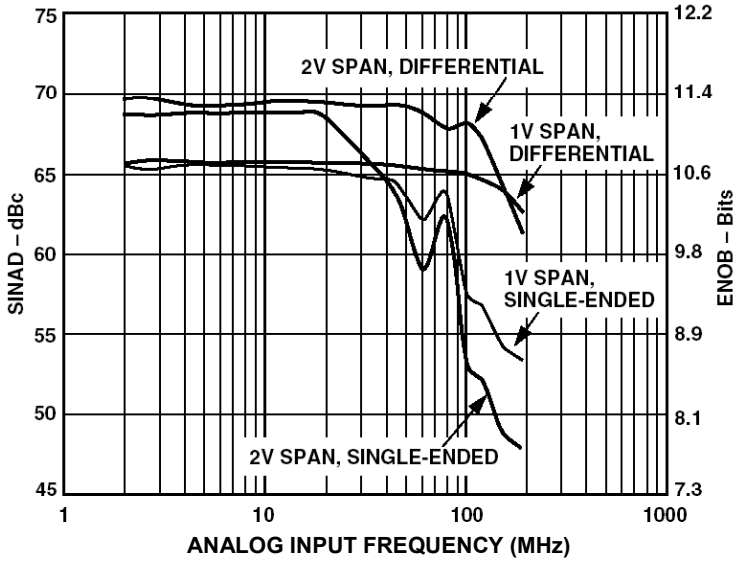


Figure 2.52: *AD9226 12-bit, 65-MSPS ADC SINAD and ENOB for Various Input Full-Scale Spans (Range)*

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

$$\text{ENOB} = \frac{\text{SINAD} - 1.76\text{ dB}}{6.02}. \qquad \text{Eq. 2.20}$$

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

Many current ADC data sheets somewhat loosely refer to SINAD as SNR, so the engineer must be careful when interpreting these specifications.

Analog Bandwidth

The analog bandwidth of an ADC is that frequency at which the spectral output of the *fundamental* swept frequency (as determined by the FFT analysis) is reduced by 3 dB. It may be specified for either a small signal (SSBW—*small signal bandwidth*), or a full-scale signal (FPBW—*full power bandwidth*), so there can be a wide variation in specifications between manufacturers.

Like an amplifier, the analog bandwidth specification of a converter does not imply that the ADC maintains good distortion performance up to its bandwidth frequency. In fact, the SINAD (or ENOB) of most ADCs will begin to degrade considerably before the input frequency approaches the actual 3-dB bandwidth frequency. Figure 2.53 shows ENOB and full-scale frequency response of an ADC with a FPBW of 1 MHz, however, the ENOB begins to drop rapidly above 100 kHz.

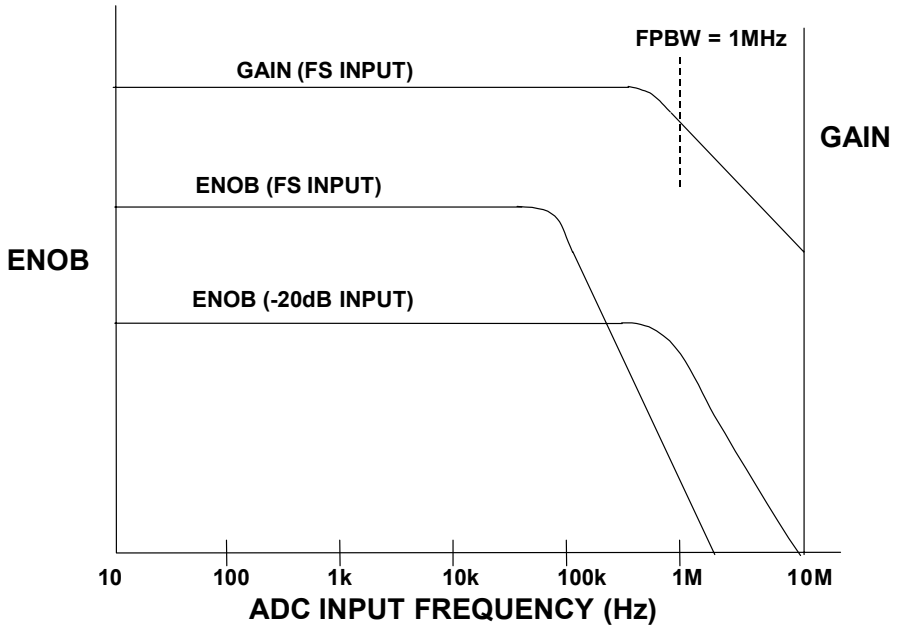


Figure 2.53: *ADC Gain (Bandwidth) and ENOB Versus Frequency Shows Importance of ENOB Specification*

Spurious Free Dynamic Range (SFDR)

Probably the most significant specification for an ADC used in a communications application is its *spurious free dynamic range* (SFDR). SFDR of an ADC is defined as the ratio of the rms signal amplitude to the rms value of the *peak spurious spectral content* measured over the bandwidth of interest. Unless otherwise stated, the bandwidth is assumed to be the Nyquist bandwidth dc to $f_s/2$.

Occasionally the frequency spectrum is divided into an *in-band* region (containing the signals of interest) and an *out-of-band* region (signals here are filtered out digitally). In this case there may be an *in-band SFDR* specification and an *out-of-band SFDR* specification, respectively.

SFDR is generally plotted as a function of signal amplitude and may be expressed relative to the signal amplitude (dBc) or the ADC full-scale (dBFS) as shown in Figure 2.54.

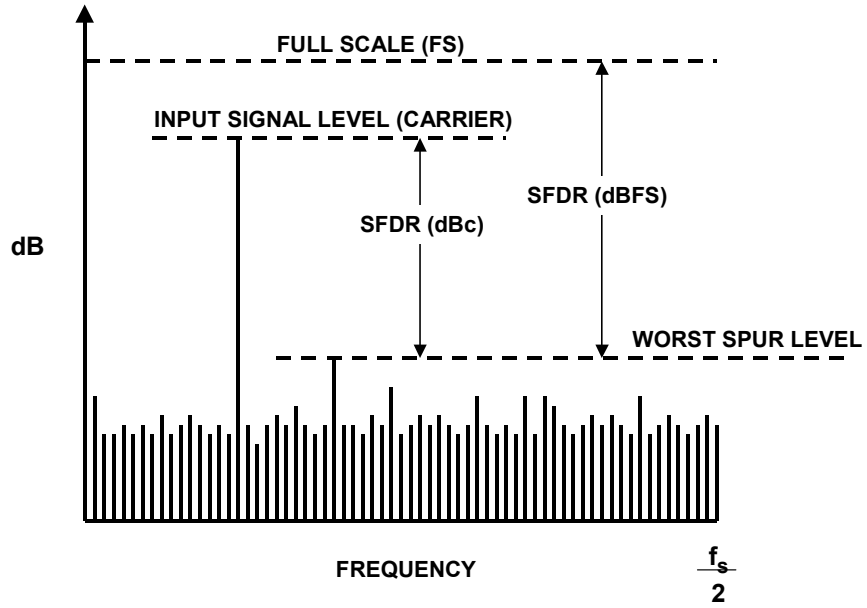


Figure 2.54: *Spurious Free Dynamic Range (SFDR)*

For a signal near full-scale, the peak spectral spur is generally determined by one of the first few harmonics of the fundamental. However, as the signal falls several dB below full-scale, other spurs generally occur which are not direct harmonics of the input signal. This is because of the differential nonlinearity of the ADC transfer function as discussed earlier. Therefore, SFDR considers *all* sources of distortion, regardless of their origin.

The AD6645 is a 14-bit, 80-MSPS wideband ADC designed for communications applications where high SFDR is important. The single-tone SFDR for a 69.1-MHz input and a sampling frequency of 80 MSPS is shown in Figure 2.55. Note that a minimum of 89-dBc SFDR is obtained over the entire first Nyquist zone (dc to 40 MHz).

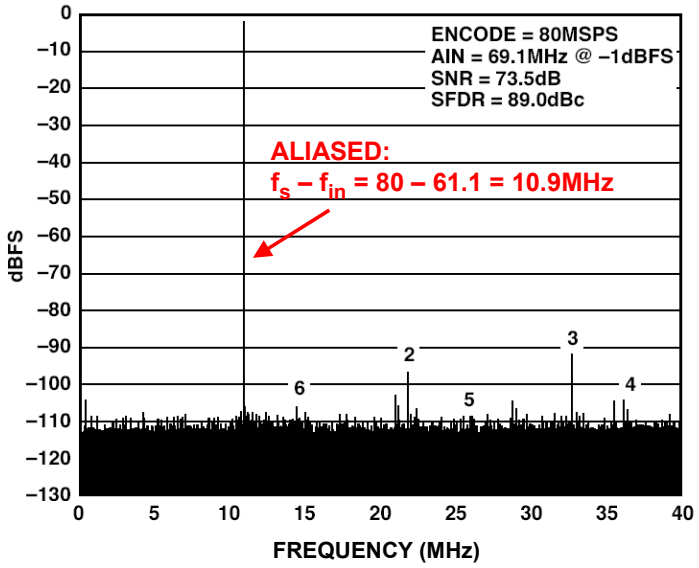


Figure 2.55: *AD6645 14-bit, 80-/105-MSPS ADC SFDR for 69.1-MHz Input*

SFDR as a function of signal amplitude is shown in Figure 2.56 for the AD6645. Notice that over the entire range of signal amplitudes, the SFDR is greater than 90 dBFS. The abrupt changes in the SFDR plot are due to the differential nonlinearities in the ADC transfer function. The nonlinearities correspond to those shown in Figure 2.49B, and are offset from mid-scale such that input signals less than about 65 dBFS do not exercise any of the points of increased DNL. It should be noted that the SFDR can be improved by injecting a small out-of-band dither signal—at the expense of a slight degradation in SNR.

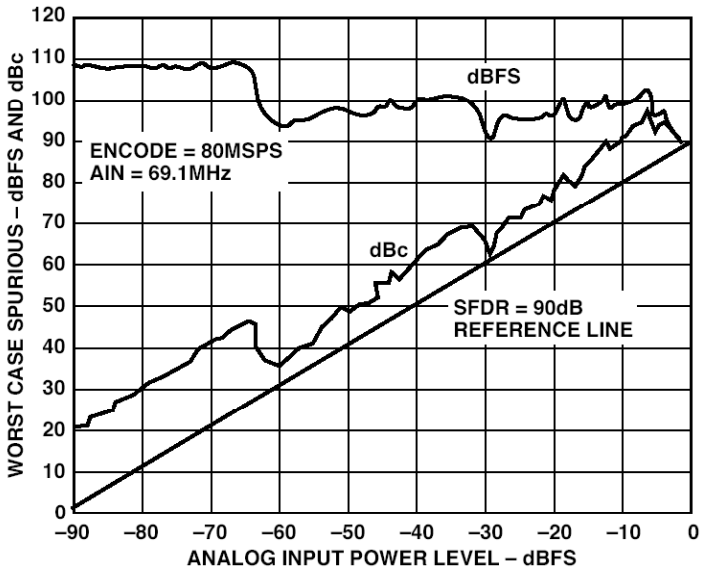


Figure 2.56: *AD6645 14-bit, 80-/105-MSPS ADC SFDR vs. Input Power Level for 69.1-MHz Input*

SFDR is generally much greater than the ADCs theoretical N-bit SNR (6.02N + 1.76 dB). For example, the AD6645 is a 14-bit ADC with an SFDR of 90 dBc and a typical SNR of 73.5 dB (the theoretical SNR for 14-bits is 86 dB). This is because there is a fundamental distinction between noise and distortion measurements. The process gain of the FFT (33 dB for a 4096-point FFT) allows frequency spurs well below the noise floor to be observed. Adding extra resolution to an ADC may serve to increase its SNR but may or may not increase its SFDR.

Two Tone Intermodulation Distortion (IMD)

Two tone IMD is measured by applying two spectrally pure sinewaves to the ADC at frequencies f_1 and f_2, usually relatively close together. The amplitude of each tone is set slightly more than 6 dB below full-scale so that the ADC does not clip when the two tones add in-phase. The location of the second and third-order products are shown in Figure 2.57. Notice that the second-order products fall at frequencies which can be removed by digital filters. However, the third-order products $2f_2 – f_1$ and $2f_1 – f_2$ are close to the original signals and are more difficult to filter. Unless otherwise specified, two-tone IMD refers to these third-order products. The value of the IMD product is expressed in dBc relative to the value of *either* of the two original tones, and not to their sum.

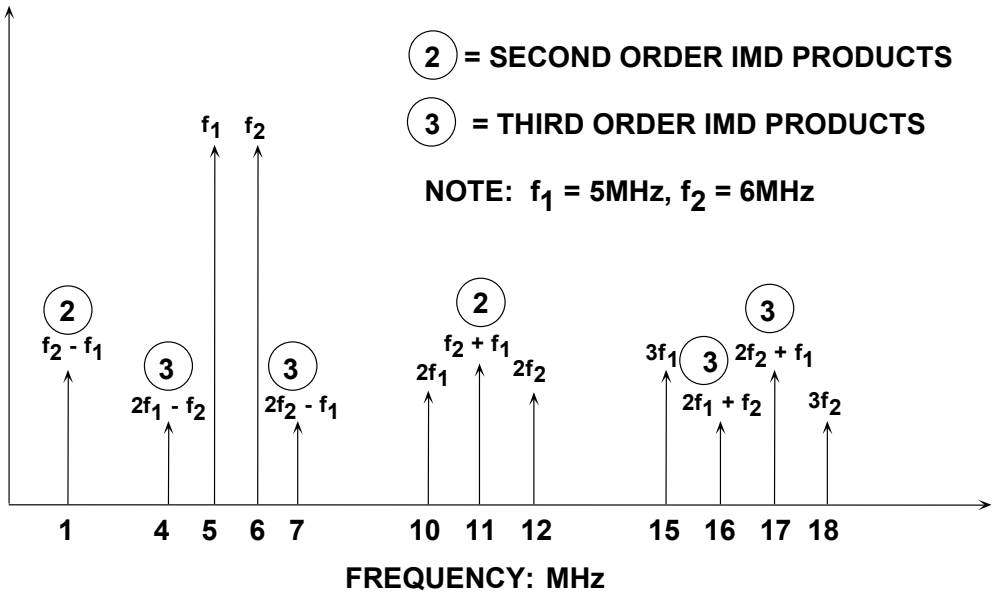


Figure 2.57: *Second and Third-Order Intermodulation Products for f_1 = 5 MHz, f_2 = 6 MHz*

Note, however, that if the two tones are close to $f_s/4$, then the aliased third harmonics of the fundamentals can make the identification of the actual $2f_2 – f_1$ and $2f_1 – f_2$ products difficult. This is because the third harmonic of $f_s/4$ is $3f_s/4$, and the alias occurs at $f_s – 3f_s/4 = f_s/4$. Similarly, if the two tones are close to $f_s/3$, the aliased second harmonics may interfere with the measurement. The same reasoning applies here; the second harmonic of $f_s/3$ is $2f_s/3$, and its alias occurs at $f_s – 2f_s/3 = f_s/3$.

Second and Third-Order Intercept Points, 1 dB Compression Point

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

In amplifiers, it is common practice to specify the third-order IMD products in terms of the *third order intercept* point, as is shown by Figure 2.58. Two spectrally pure tones are applied to the system. The output signal power in a single tone (in dBm) as well as the relative amplitude of the third-order products (referenced to a single tone) are plotted as a function of input signal power. The fundamental is shown by the *slope = 1* curve in the diagram. If the system nonlinearity is approximated by a power series expansion, it can be shown that second-order IMD amplitudes increase 2 dB for every 1 dB of signal increase, as represented by *slope = 2* curve in the diagram.

Similarly, the third-order IMD amplitudes increase 3 dB for every 1-dB of signal increase, as indicated by the *slope = 3* plotted line. With a low level two-tone input signal, and two data points, one can draw the second and third order IMD lines as they are shown in Figure 2.58 (using the principle that a point and a slope define a straight line).

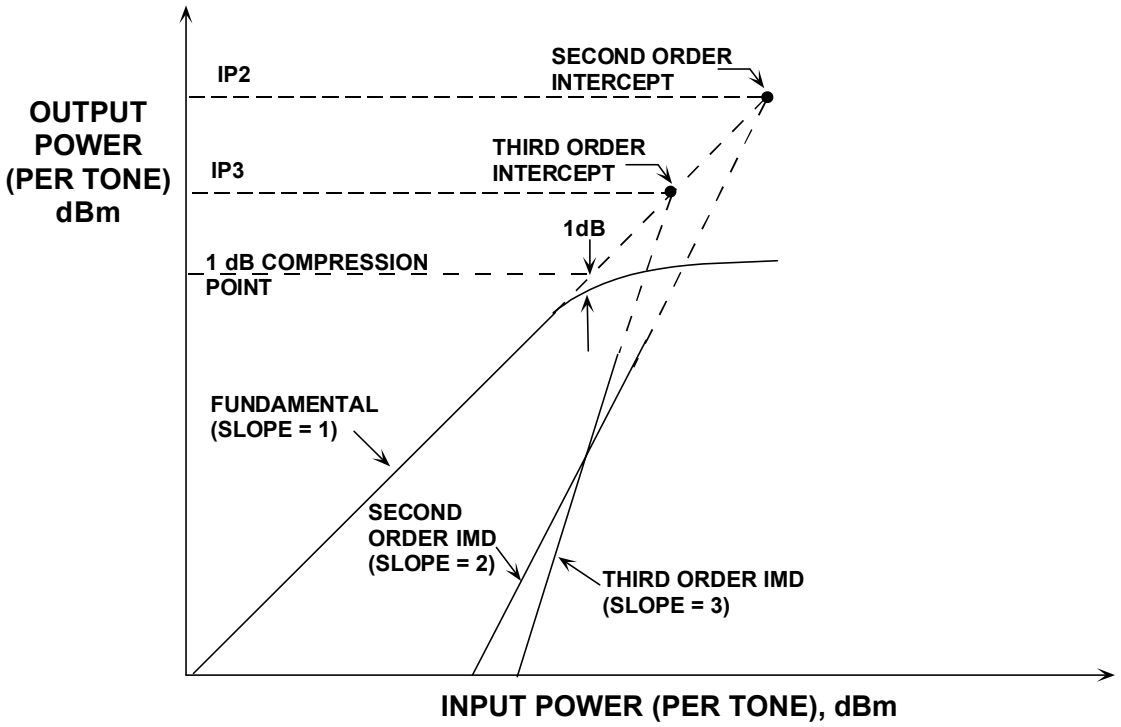


Figure 2.58: *Definition of Intercept Points and 1-dB Compression Points for Amplifiers*

Once the input reaches a certain level however, the output signal begins to soft-limit, or compress. A parameter of interest here is the *1-dB compression point*. This is the point where the output signal is compressed 1 dB from an ideal input/output transfer function. This is shown in Figure 2.58 within the region where the ideal slope = 1 line becomes dotted, and the actual response exhibits compression (solid).

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

It should be noted that IP2, IP3, and the 1-dB compression point are all a function of frequency, and as one would expect, the distortion is worse at higher frequencies. For a given frequency, knowing the third order intercept point allows calculation of the approximate level of the third-order IMD products as a function of output signal level.

The concept of *second- and third-order intercept points* is not valid for an ADC, because the distortion products do not vary in a predictable manner (as a function of signal amplitude). The ADC does not gradually begin to compress signals approaching full-scale (there is no 1-dB compression point); it acts as a *hard limiter* as soon as the signal exceeds the ADC input range, thereby suddenly producing extreme amounts of distortion because of clipping. On the other hand, for signals much below full-scale, the distortion floor remains relatively constant and is independent of signal level. This is shown graphically in Figure 2.59.

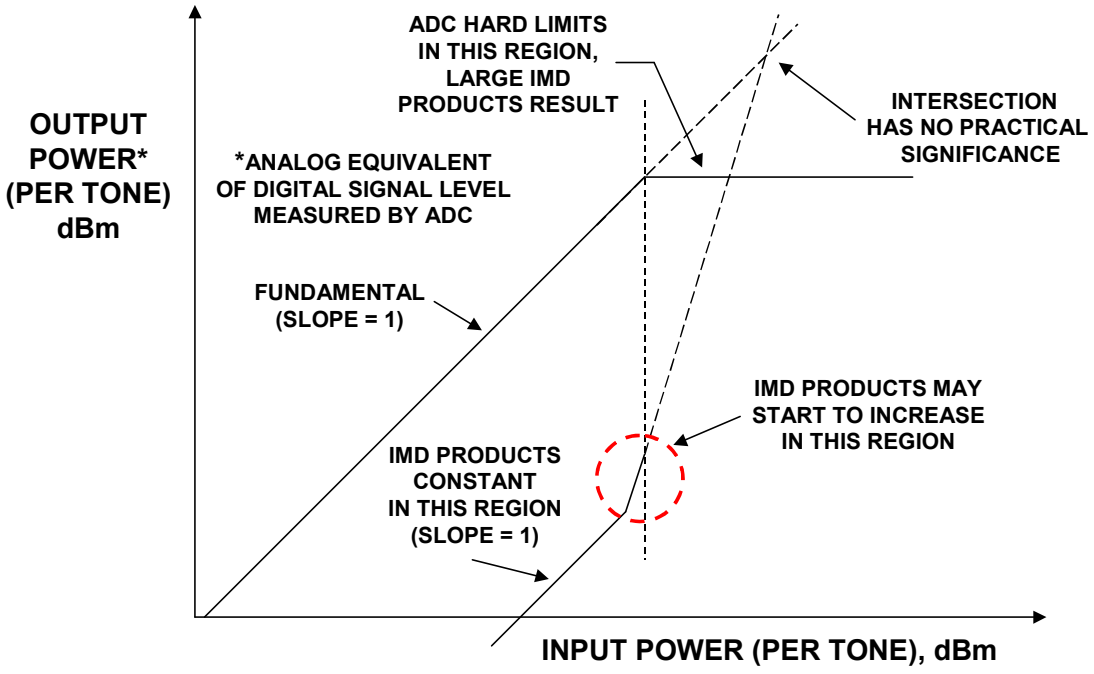


Figure 2.59: *Intercept Points for Data Converters Have No Practical Significance*

The IMD curve in Figure 2.59 is divided into three regions. For low level input signals, the IMD products remain relatively constant regardless of signal level. This implies that as the input signal increases 1 dB, the ratio of the signal to the IMD level will increase 1 dB also. When the input signal is within a few dB of the ADC full-scale range, the IMD may start to increase (but it might not in a very well-designed ADC). The exact level at which this occurs is dependent on the particular ADC under consideration—some ADCs may not exhibit significant increases in the IMD products over their full input range,

however most will. As the input signal continues to increase beyond full-scale, the ADC should function act as an ideal limiter, and the IMD products become very large.

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

Multi-Tone Spurious Free Dynamic Range

Two-tone and multi-tone SFDR is often measured in communications applications. The larger number of tones more closely simulates the wideband frequency spectrum of cellular telephone systems such as AMPS or GSM. Figure 2.60 shows the 2-tone intermodulation performance of the AD6645 14-bit, 80-/105-MSPS ADC. The input tones are at 55.25 MHz and 56.25 MHz and are located in the second Nyquist Zone.

The aliased tones therefore occur at 23.75 MHz and 24.75 MHz in the first Nyquist Zone. High SFDR increases the receiver's ability to capture small signals in the presence of large ones, and prevents the small signals from being masked by the intermodulation products of the larger ones. Figure 2.61 shows the AD6645 two-tone SFDR as a function of input signal amplitude for the same input frequencies.

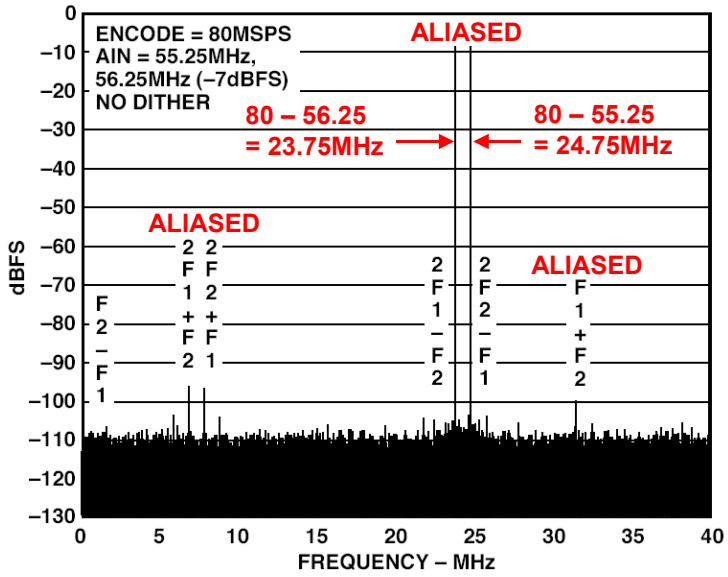


Figure 2.60: *Two-Tone SFDR for AD6645 14-bit, 80-/105-MSPS ADC, Input Tones: 55.25 MHz and 56.25 MHz*

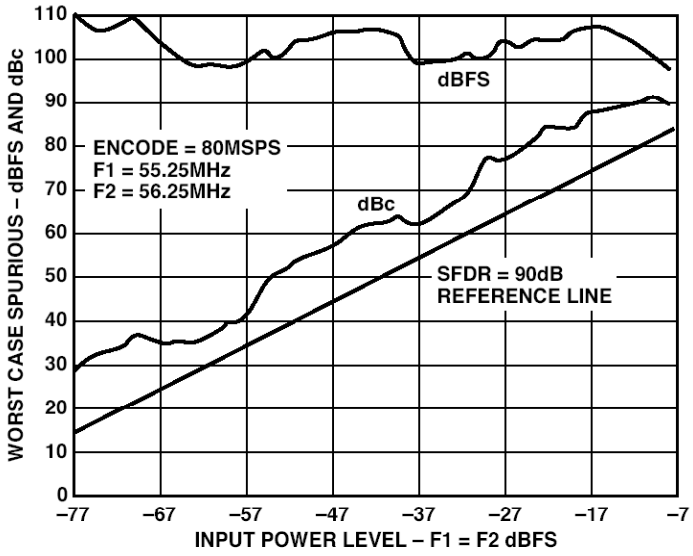


Figure 2.61: *Two-Tone SFDR vs. Input Amplitude for AD6645 14-bit, 80-/105-MSPS ADC*

Wideband CDMA (WCDMA) Adjacent Channel Power Ratio (ACPR) and Adjacent Channel Leakage Ratio (ADLR)

A wideband CDMA channel has a bandwidth of approximately 3.84 MHz, and channel spacing is 5 MHz. The ratio in dBc between the measured power within a channel relative to its adjacent channel is defined as the *adjacent channel power ratio* (ACPR).

The ratio in dBc between the measured power within the channel bandwidth relative to the noise level in an adjacent empty carrier channel is defined as *adjacent channel leakage ratio* (ACLR).

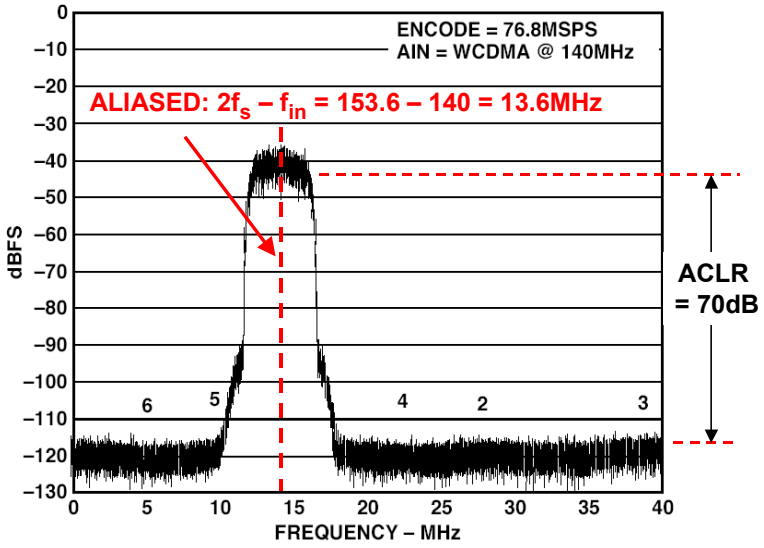


Figure 2.62: *Wideband CDMA (WCDMA) Adjacent Channel Leakage Ratio (ACLR)*

Figure 2.62 shows a single wideband CDMA channel centered at 140 MHz sampled at a frequency of 76.8 MSPS using the AD6645. This is a good example of undersampling (direct IF-to-digital conversion).The signal lies within the fourth Nyquist zone: $3f_s/2$ to $2f_s$ (115.2 MHz to 153.6 MHz). The aliased signal within the first Nyquist zone is therefore centered at $2f_s - f_a = 153.6 - 140 = 13.6$ MHz. The diagram also shows the location of the aliased harmonics. For example, the second harmonic of the input signal occurs at $2 \times 140 = 280$ MHz, and the aliased component occurs at $4f_s - 2f_a = 4 \times 76.8 - 280 = 307.2 - 280 = 27.2$ MHz.

Noise Power Ratio (NPR)

Noise power ratio has been used extensively to measure the transmission characteristics of Frequency Division Multiple Access (FDMA) communications links (see Reference 7). In a typical FDMA system, 4-kHz wide voice channels are "stacked" in frequency bins for transmission over coaxial, microwave, or satellite equipment. At the receiving end, the FDMA data is demultiplexed and returned to 4-kHz individual baseband channels. In an FDMA system having more than approximately 100 channels, the FDMA signal can be approximated by Gaussian noise with the appropriate bandwidth. An individual 4-kHz channel can be measured for "quietness" using a narrow-band notch (bandstop) filter and a specially tuned receiver which measures the noise power inside the 4-kHz notch (see Figure 2.63).

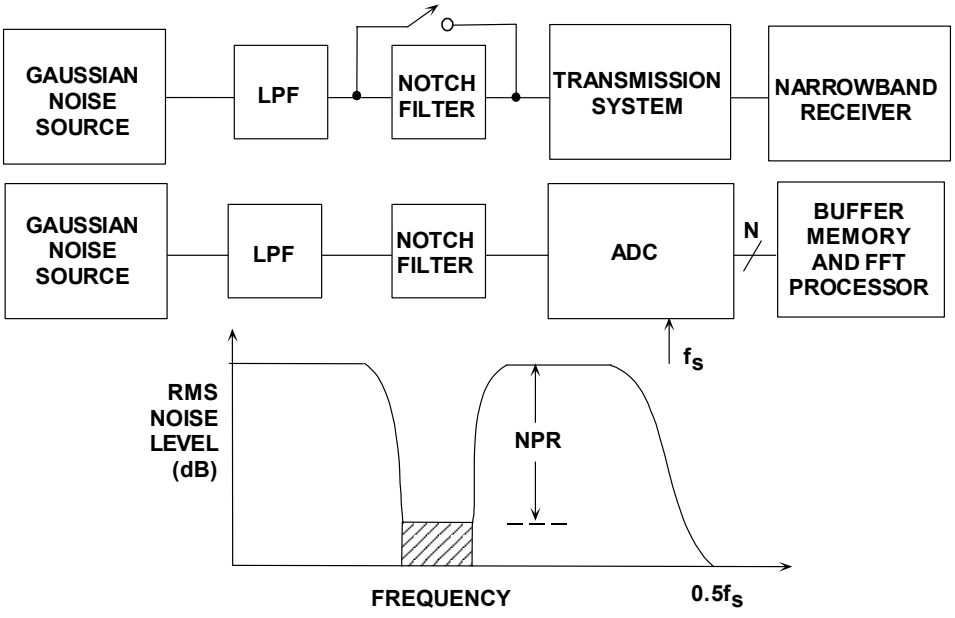


Figure 2.63: *Noise Power Ratio (NPR) Measurements*

Noise Power Ratio (NPR) measurements are straightforward. With the notch filter out, the rms noise power of the signal inside the notch is measured by the narrowband receiver. The notch filter is then switched in, and the residual noise inside the slot is

measured. The ratio of these two readings expressed in dB is the NPR. Several slot frequencies across the noise bandwidth (low, midband, and high) are tested to characterize the system adequately. NPR measurements on ADCs are made in a similar manner except the analog receiver is replaced by a buffer memory and an FFT processor.

The NPR is plotted as a function of rms noise level referred to the peak range of the system. For very low noise loading level, the undesired noise (in non-digital systems) is primarily thermal noise and is independent of the input noise level. Over this region of the curve, a 1-dB increase in noise loading level causes a 1-dB increase in NPR. As the noise loading level is increased, the amplifiers in the system begin to overload, creating intermodulation products which cause the noise floor of the system to increase. As the input noise increases further, the effects of "overload" noise predominate, and the NPR is reduced dramatically. FDMA systems are usually operated at a noise loading level a few dB below the point of maximum NPR.

In a digital system containing an ADC, the noise within the slot is primarily quantization noise when low levels of noise input are applied. The NPR curve is linear in this region. As the noise level increases, there is a one-for-one correspondence between the noise level and the NPR. At some level, however, "clipping" noise caused by the hard-limiting action of the ADC begins to dominate. A theoretical curve for 10, 11, and 12-bit ADCs is shown in Figure 2.64 (see References 8 and 21).

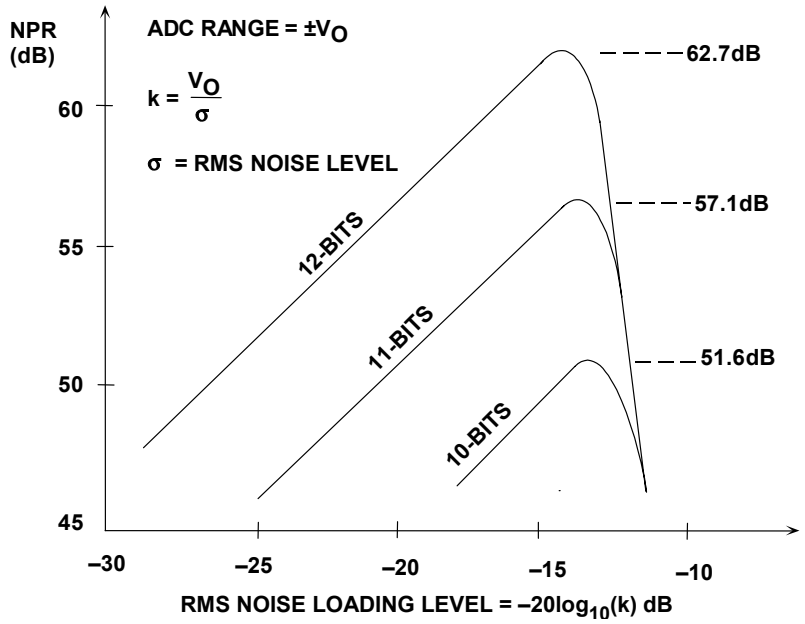


Figure 2.64: *Theoretical NPR for 10, 11, 12-bit ADCs*

Figure 2.65 shows the maximum theoretical NPR and the noise loading level at which the maximum value occurs for 8 to 16-bit ADCs. The ADC input range is $2V_O$ peak-to-peak. The rms noise level is σ, and the noise-loading factor k (crest factor) is defined as V_O/σ , the peak-to-rms ratio (k is expressed either as numerical ratio or in dB).

BITS	k OPTIMUM	k(dB)	MAX NPR (dB)
8	3.92	11.87	40.60
9	4.22	12.50	46.05
10	4.50	13.06	51.56
11	4.76	13.55	57.12
12	5.01	14.00	62.71
13	5.26	14.41	68.35
14	5.49	14.79	74.01
15	5.72	15.15	79.70
16	5.94	15.47	85.40

ADC Range = $\pm V_O$
$k = V_O / \sigma$
σ = RMS Noise Level

Figure 2.65: *Theoretical Maximum NPR for 8 to 16-bit ADCs*

In multi-channel high frequency communication systems, where there is little or no phase correlation between channels, NPR can also be used to simulate the distortion caused by a large number of individual channels, similar to an FDMA system. A notch filter is placed between the noise source and the ADC, and an FFT output is used in place of the analog receiver. The width of the notch filter is set for several MHz as shown in Figure 2.66 for the AD9430 12-bit 170-/210-MSPS ADC. The notch is centered at 19 MHz, and the NPR is the "depth" of the notch. An ideal ADC will only generate quantization noise inside the notch, however a practical one has additional noise components due to additional noise and intermodulation distortion caused by ADC imperfections. Notice that the NPR is about 57 dB compared to 62.7-dB theoretical.

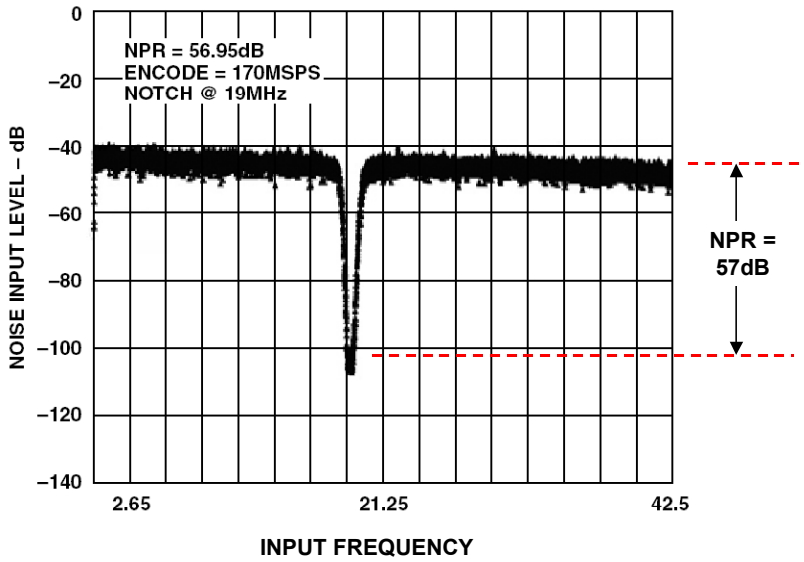


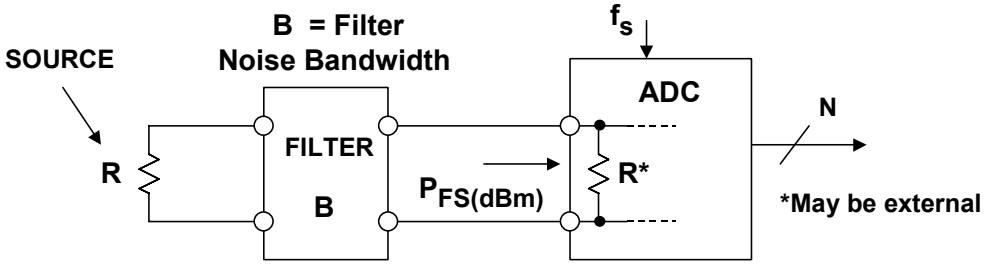
Figure 2.66: *AD9430 12-bit,170-/210-MSPS ADC NPR Measures 57 dB (62.7 dB Theoretical)*

Noise Factor (F) and Noise Figure (NF)

Noise figure (NF) is a popular specification among RF system designers. It is used to characterize RF amplifiers, mixers, etc., and widely used as a tool in radio receiver design. Many excellent textbooks on communications and receiver design treat noise figure extensively (see Reference 9, for example)—it is not the purpose of this discussion to discuss the topic in much detail, but only how it applies to data converters.

Since many wideband operational amplifiers and ADCs are now being used in RF applications, the inevitable day has come where the noise figure of these devices becomes important. As discussed in Reference 10, in order to determine the noise figure of an op amp correctly, one must not only know op amp voltage and current noise, but the exact circuit conditions—closed-loop gain, gain-setting resistor values, source resistance, bandwidth, etc. Calculating the noise figure for an ADC is even more of a challenge as will be seen.

Figure 2.67 shows the basic model for defining the noise figure of an ADC. The *noise factor*, F, is simply defined as the ratio of the total effective input noise power of the ADC to the amount of that noise power caused by the source resistance alone. Because the impedance is matched, the square of the voltage noise can be used instead of noise power. The *noise figure*, NF, is simply the noise factor expressed in dB, $NF = 10\log_{10}F$.



$$\text{NOISE FACTOR (F)} = \frac{(\text{TOTAL EFFECTIVE INPUT NOISE})^2}{(\text{TOTAL INPUT NOISE DUE TO SOURCE R})^2}$$

$$\text{NOISE FIGURE (NF)} = 10\log_{10}\left[\frac{(\text{TOTAL EFFECTIVE INPUT NOISE})^2}{(\text{TOTAL INPUT NOISE DUE TO SOURCE R})^2}\right]$$

Note: Noise Must be Measured Over the Filter Noise Bandwidth, B

Figure 2.67: *Noise Figure for ADCs: Use with Caution!*

This model assumes the input to the ADC comes from a source having a resistance, R, and that the input is band-limited to $f_S/2$ with a filter having a noise bandwidth equal to $f_S/2$. It is also possible to further band-limit the input signal resulting in oversampling and process gain, and this condition will be discussed shortly.

It is also assumed that the input impedance to the ADC is equal to the source resistance. Many ADCs have a high input impedance, so this termination resistance may be external to the ADC or used in parallel with the internal resistance to produce an equivalent termination resistance equal to R. The full-scale input power is the power of a sinewave whose peak-to-peak amplitude fills the entire ADC input range. The full-scale input sinewave given by the following equation has a peak-to-peak amplitude of $2V_O$ corresponding to the peak-to-peak input range of the ADC:

$$v(t) = V_O \sin 2\pi ft \qquad \text{Eq. 2.21}$$

The full-scale power in this sinewave is given by:

$$P_{FS} = \frac{(V_O / \sqrt{2})^2}{R} = \frac{V_O{}^2}{2R} \qquad \text{Eq. 2.22}$$

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

$$P_{FS(dBm)} = 10 \log_{10} \left[\frac{P_{FS}}{1\,mW} \right]. \qquad \text{Eq. 2.23}$$

The *noise bandwidth* of a non-ideal brick wall filter is defined as the bandwidth of an ideal brick wall filter which will pass the same noise power as the non-ideal filter. Therefore, the noise bandwidth of a filter is always greater than the 3-dB bandwidth of the filter by a factor which depends upon the sharpness of the cutoff region of the filter. Figure 2.68 shows the relationship between the noise bandwidth and the 3-dB bandwidth for Butterworth filters up to 5 poles. Note that for two poles, the noise bandwidth and 3-dB bandwidth are within 11% of each other, and beyond that the two quantities are essentially equal.

NUMBER OF POLES	NOISE BW / 3dB BW
1	1.57
2	1.11
3	1.05
4	1.03
5	1.02

Figure 2.68: *Relationship Between Noise Bandwidth and 3-dB Bandwidth for Butterworth Filter*

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

$$\mathrm{SNR} = 20\log_{10}\left[\frac{V_{\mathrm{FS\ RMS}}}{V_{\mathrm{NOISE\ RMS}}}\right] \qquad \text{Eq. 2.24}$$

Solving for $V_{NOISE\ RMS}$:

$$V_{\mathrm{NOISE\ RMS}} = V_{\mathrm{FS\ RMS}} \cdot 10^{-\mathrm{SNR}/20} \qquad \text{Eq. 2.25}$$

This is the total effective input rms noise voltage at the carrier frequency measured over the Nyquist bandwidth, dc to $f_s/2$. Note that this noise includes the source resistance noise. These results are summarized in Figure 2.69.

- **Start with the SNR of the ADC measured at the carrier frequency (Note: this SNR value does not include the harmonics of the fundamental and is measured over the Nyquist bandwidth, dc to $f_S/2$)**

$$\mathbf{SNR = 20\ log_{10}\ \frac{V_{FS\text{-}RMS}}{V_{NOISE\text{-}RMS}}}$$

$$\mathbf{V_{NOISE\text{-}RMS} = V_{FS\text{-}RMS}\ 10^{-SNR\ /\ 20}}$$

- **This is the total ADC effective input noise at the carrier frequency measured over the Nyquist bandwidth, dc to $f_S/2$**

Figure 2.69: *Calculating ADC Total Effective Input Noise from SNR*

The next step is to actually calculate the noise figure. In Figure 2.70 notice that the amount of the input voltage noise due to the source resistance is the voltage noise of the source resistance √(4kTBR) divided by two, or √(kTBR) because of the 2:1 attenuator formed by the ADC input termination resistor.

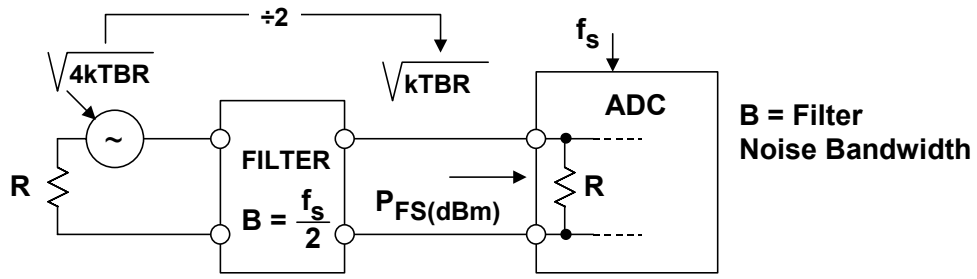
The expression for the noise factor F can be written:

$$F = \frac{V_{\mathrm{NOISE\ RMS}}^2}{kTRB} = \left[\frac{V_{\mathrm{FS\ RMS}}^2}{R}\right]\left[\frac{1}{kT}\right]\left[10^{-\mathrm{SNR}/10}\right]\left[\frac{1}{B}\right] \qquad \text{Eq. 2.26}$$

The noise figure is obtained by converting F into dB and simplifying:

$$NF = 10_{10}\log F = P_{FS(dBm)} + 174\ dBm - SNR - 10_{10}\log B, \qquad \text{Eq. 2.27}$$

Where SNR is in dB, B in Hz, T = 300 K, k = 1.38×10^{-23} J/K.



$$V_{NOISE\text{-}RMS} = V_{FS\text{-}RMS}\ 10^{-SNR/20}$$

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

$$NF = 10\log_{10}F = P_{FS(dBm)} + 174dBm - SNR - 10\log_{10}B,$$

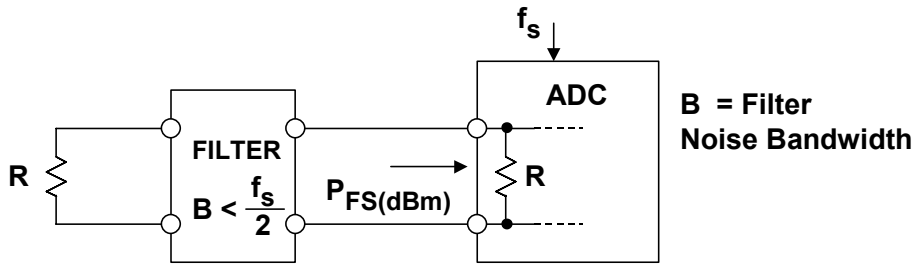
where SNR is in dB, B in Hz, T = 300K, $k = 1.38 \times 10^{-23}$ J/K

Figure 2.70: *ADC Noise Figure in Terms of SNR, Sampling Rate, and Input Power*

Oversampling and filtering can be used to decrease the noise figure as a result of the process gain as has been previously discussed. In this case, the signal bandwidth B is less than $f_s/2$. Figure 2.71 shows the correction factor which results in the following equation:

$$NF = 10_{10}\log F = P_{FS(dBm)} + 174\ dBm - SNR - 10\log_{10}[f_s/2B] - 10\log_{10}B. \qquad \text{Eq. 2.28}$$

Figure 2.72 shows an example NF calculation for the AD6645 14-bit, 80-MSPS ADC. A 52.3-Ω resistor is added in parallel with the AD6645 input impedance of 1 kΩ to make the net input impedance 50 Ω. The ADC is operating under Nyquist conditions, and the SNR of 74 dB is the starting point for the calculations using Eq. 2.28 above. A noise figure of 34.8 dB is obtained.

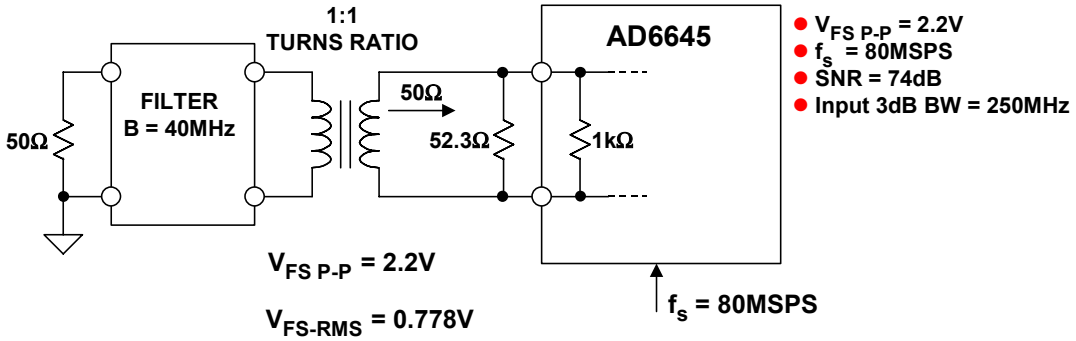


$$NF = P_{FS(dBm)} + 174dBm - SNR - 10\log_{10}\left[\frac{f_S/2}{B}\right] - 10\log_{10}B,$$

Measured DC to $f_S/2$ | Process Gain

where SNR is in dB, B in Hz, $T = 300K$, $k = 1.38 \times 10^{-23}$ J/K

Figure 2.71: *Effect of Oversampling and Process Gain on ADC Noise Figure*



$V_{FS\text{-}RMS} = 0.778V$

$$P_{FS} = \frac{(0.778)^2}{50} = 12.1mW$$

$P_{FS(dBm)} = +10.8dBm$

$$NF = P_{FS(dBm)} + 174dBm - SNR - 10\log_{10}B$$

$$= +10.8dBm + 174dBm - 74dB - 10\log_{10}(40 \times 10^6)$$

$$= 34.8dB$$

Figure 2.72: *Example Calculation of Noise Figure Under Nyquist Conditions for AD6645*

Figure 2.73 shows how using an RF transformer with voltage gain can improve the noise figure. Figure 2.73A shows a 1:1 turns ratio, and the noise figure (from Figure 2.72) is 34.8. Figure 2.73B shows a transformer with a 1:2 turns ratio. The 249-Ω resistor in parallel with the AD6645 internal resistance results in a net input impedance of 200 Ω.

The noise figure is improved by 6 dB because of the "noise-free" voltage gain of the transformer. Figure 2.73C shows a transformer with a 1:4 turns ratio. The AD6645 input is paralleled with a 4.02-kΩ resistor to make the net input impedance 800 Ω. The noise figure is improved by another 6 dB. Transformers with higher turns ratios are not generally practical because of bandwidth and distortion limitations.

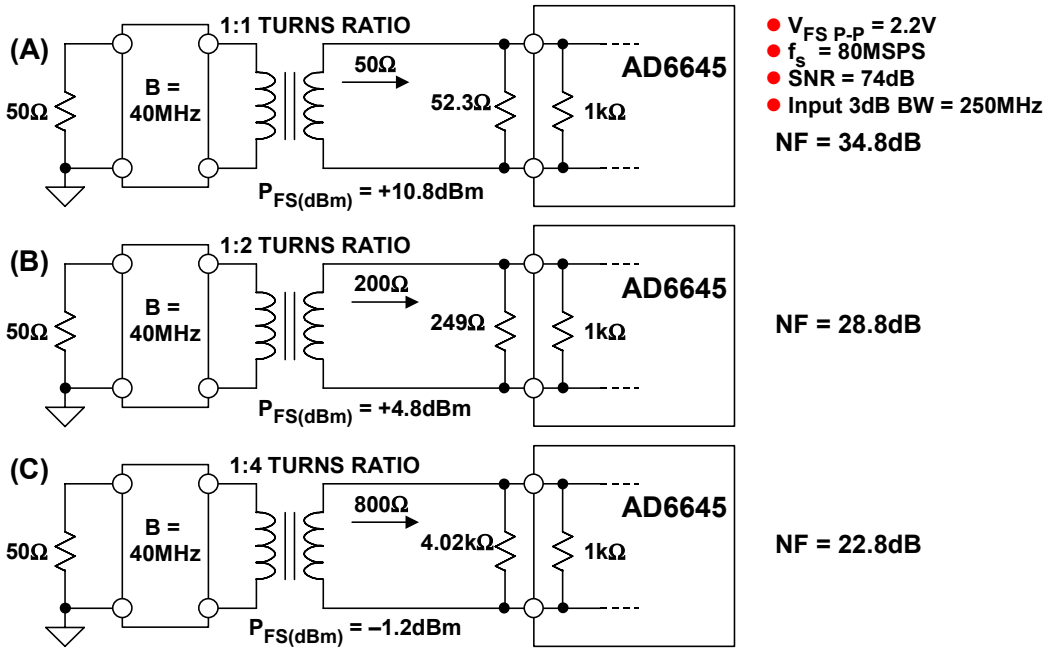
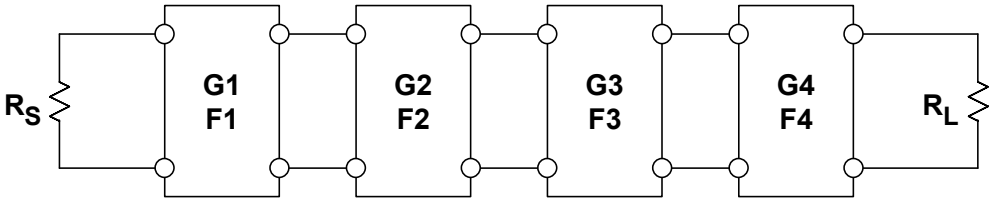


Figure 2.73: *Using RF Transformers to Improve Overall ADC Noise Figure*

Even with the 1:4 turns ratio transformer, the overall noise figure for the AD6645 was still 22.8 dB, still relatively high by RF standards. The solution is to provide low-noise high-gain stages ahead of the ADC. Figure 2.74 shows how the Friis equation is used to calculate the noise factor for cascaded gain stages. Notice that high gain in the first stage reduces the contribution of the noise factor of the second stage—the noise factor of the first stage dominates the overall noise factor.

Figure 2.75 shows the effects of a high-gain (25 dB) low-noise (NF = 4 dB) stage placed in front of a relatively high NF stage (30 dB)—the noise figure of the second stage is typical of high performance ADCs. The overall noise figure is 7.53 dB, only 3.53 dB higher than the first stage noise figure of 4 dB.



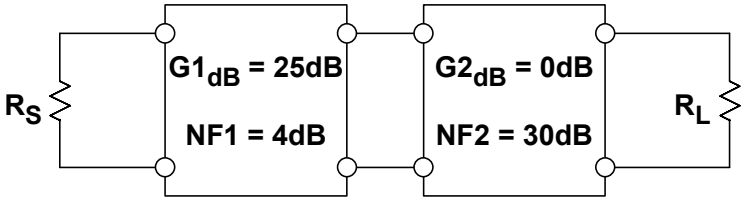
$$F_T = F1 + \frac{F2-1}{G1} + \frac{F3-1}{G1 \cdot G2} + \frac{F4-1}{G1 \cdot G2 \cdot G3} + \ldots$$

High gain in the first stage reduces the contribution of the NF of the second stage

NF of the first stage dominates the total NF

$$NF_T = 10 \log_{10} F_T$$

Figure 2.74: *Cascaded Noise Figure Using the Friis Equation*



$$G1 = 10^{25/10} = 10^{2.5} = 316, \quad F1 = 10^{4/10} = 10^{0.4} = 2.51$$

$$G2 = 1, \quad F2 = 10^{30/10} = 10^{3} = 1000$$

$$F_T = F1 + \frac{F2-1}{G1} = 2.51 + \frac{1000-1}{316} = 2.51 + 3.16 = 5.67$$

$$NF_T = 10 \log_{10} 5.67 = 7.53\text{dB}$$

- The first stage dominates the overall NF
- It should have the highest gain possible with the lowest NF possible

Figure 2.75: *Example of Two-Stage Cascaded Network*

In summary, applying the noise figure concept to characterize wideband ADCs must be done with extreme caution to prevent misleading results. Simply trying to minimize the noise figure using the equations can actually increase circuit noise.

For instance, NF decreases with increasing source resistance according to the calculations, but increased source resistance increases circuit noise. Also, NF decreases with increasing ADC input bandwidth if there is no input filtering. This is also contradictory, because widening the bandwidth increases noise. In both these cases, the circuit noise increases, and the NF decreases. The reason NF decreases is that the source

noise makes up a larger component of the total noise (which remains relatively constant because the ADC noise is much greater than the source noise); therefore according to the calculation, NF decreases, but actual circuit noise increases.

It is true that on a stand-alone basis ADCs have relatively high noise figures compared to other RF parts such as LNAs or mixers. In the system the ADC should be preceded with low-noise gain blocks as shown in the example of Figure 2.75. Noise figure considerations for ADCs are summarized in Figure 2.76.

- **NF decreases with increasing source resistance.**
- **NF decreases with increasing ADC input bandwidth if there is no input filtering.**
- **In both cases, the circuit noise increases, and the NF decreases.**
- **The reason NF decreases is that the source noise makes up a larger component of the total noise (which remains relatively constant because the ADC noise is much greater than the source noise).**
- **In practice, input filtering is used to limit the input noise bandwidth and reduce overall system noise.**
- **ADCs have relatively high NF compared to other RF parts. In the system the ADC should be preceded with low-noise gain blocks.**
- **Exercise caution when using NF!**

Figure 2.76: *Noise Figure Considerations for ADCs: Summary and Caution*

Aperture Time, Aperture Delay Time, and Aperture Jitter

Perhaps the most misunderstood and misused ADC and sample-and-hold (or track-and-hold) specifications are those that include the word *aperture*. The most essential dynamic property of a SHA is its ability to disconnect quickly the hold capacitor from the input buffer amplifier as shown in Figure 2.77. The short (but non-zero) interval required for this action is called *aperture time (or sampling aperture)*, t_a. The actual value of the voltage that is held at the end of this interval is a function of both the input signal slew rate and the errors introduced by the switching operation itself. Figure 2.77 shows what happens when the hold command is applied with an input signal of two arbitrary slopes labeled as 1 and 2. For clarity, the sample-to-hold pedestal and switching transients are ignored. The value that is finally held is a delayed version of the input signal, averaged over the aperture time of the switch as shown in Figure 2.77. The first-order model assumes that the final value of the voltage on the hold capacitor is approximately equal to the average value of the signal applied to the switch over the interval during which the switch changes from a low to high impedance (t_a).

The model shows that the finite time required for the switch to open (t_a) is equivalent to introducing a small delay t_e in the sampling clock driving the SHA. This delay is constant and may either be positive or negative. The diagram shows that the same value of t_e works for the two signals, even though the slopes are different. This delay is called

effective aperture delay time, *aperture delay time*, or simply *aperture delay*, t_e. In an ADC, the aperture delay time is referenced to the input of the converter, and the effects of the analog propagation delay through the input buffer, t_{da} and the digital delay through the switch driver, t_{dd}, must be considered. Referenced to the ADC inputs, aperture time, t_e', is defined as the time difference between the analog propagation delay of the front-end buffer, t_{da}, and the switch driver digital delay, t_{dd}, plus one-half the aperture time, $t_a/2$.

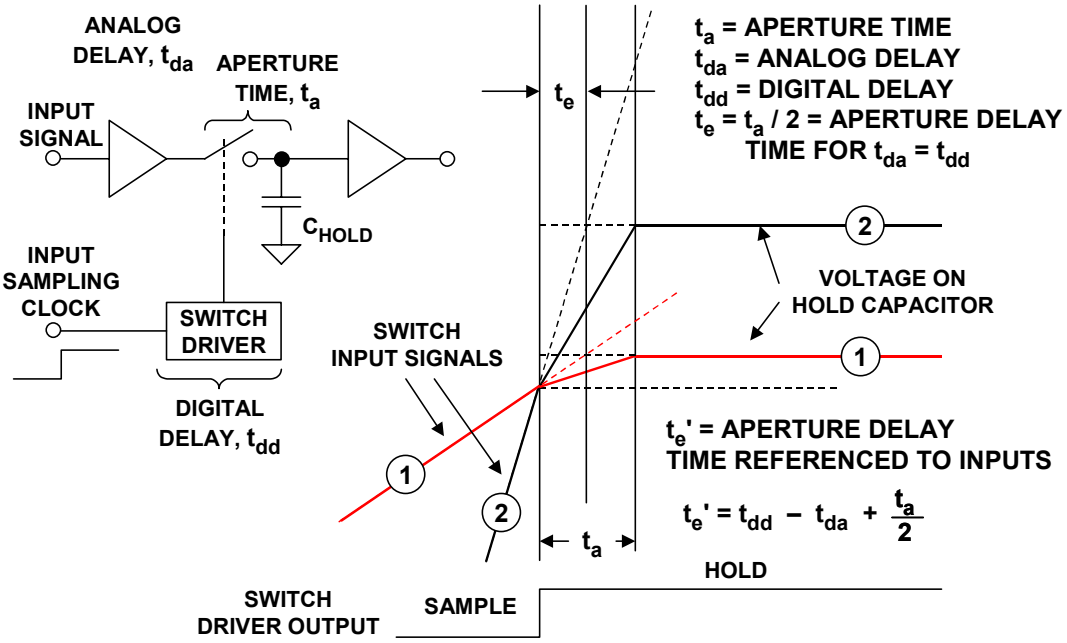


Figure 2.77: *Sample-and-Hold Waveforms and Definitions*

The effective aperture delay time is usually positive, but may be negative if the sum of one-half the aperture time, $t_a/2$, and the switch driver digital delay, t_{dd}, is less than the propagation delay through the input buffer, t_{da}. The aperture delay specification thus establishes when the input signal is actually sampled with respect to the sampling clock edge.

Aperture delay time can be measured by applying a bipolar sinewave signal to the ADC and adjusting the synchronous sampling clock delay such that the output of the ADC is mid-scale (corresponding to the zero-crossing of the sinewave). The relative delay between the input sampling clock edge and the actual zero-crossing of the input sinewave is the aperture delay time (see Figure 2.78).

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

If, however, there is *sample-to-sample* variation in aperture delay (*aperture jitter*), then a corresponding voltage error is produced as shown in Figure 2.79. This sample-to-sample variation in the instant the switch opens is called *aperture uncertainty*, or *aperture jitter* and is usually measured in rms picoseconds. The amplitude of the associated output error is related to the rate-of-change of the analog input. For any given value of aperture jitter, the aperture jitter error increases as the input dv/dt increases. The effects of phase jitter on the external sampling clock (or the analog input for that matter) produce exactly the same type of error.

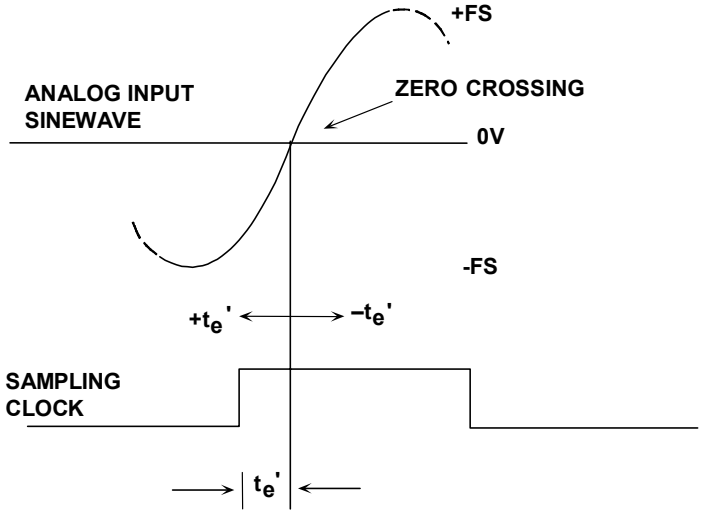


Figure 2.78: *Effective Aperture Delay Time Measured with Respect to ADC Input*

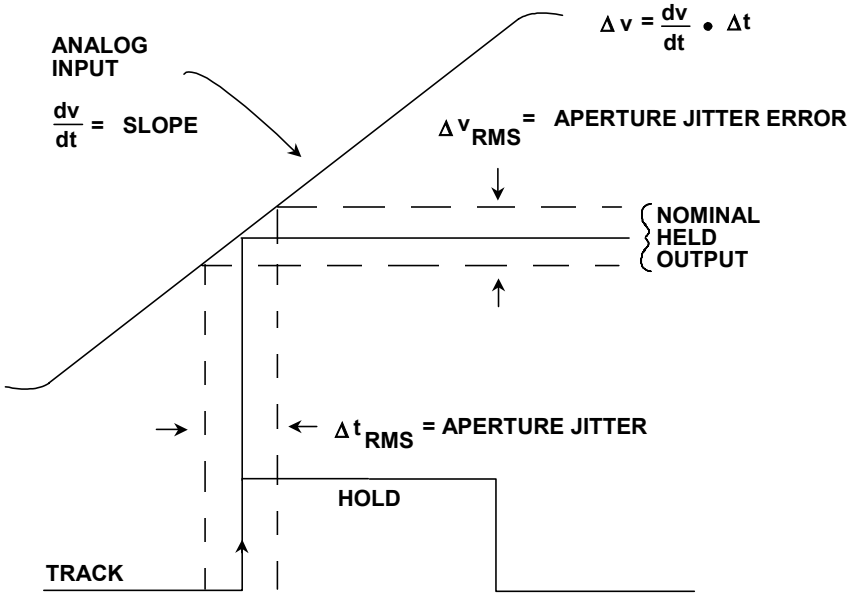


Figure 2.79: *Effects of Aperture Jitter and Sampling Clock Jitter*

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

$$v(t) = V_O \sin 2\pi ft \qquad \text{Eq. 2.29}$$

The rate of change of this signal is given by:

$$dv/dt = 2\pi fV_O \cos 2\pi ft. \qquad \text{Eq. 2.30}$$

The rms value of dv/dt can be obtained by dividing the amplitude, $2\pi fV_O$, by $\sqrt{2}$:

$$dv/dt|_{rms} = 2\pi fV_O / \sqrt{2}. \qquad \text{Eq. 2.31}$$

Now let Δv_{rms} = the rms voltage error and Δt = the rms aperture jitter t_j, and substitute:

$$\Delta v_{rms} / t_j = 2\pi fV_O / \sqrt{2}. \qquad \text{Eq. 2.32}$$

Solving for Δv_{rms} :

$$\Delta v_{rms} = 2\pi fV_O t_j / \sqrt{2}. \qquad \text{Eq. 2.33}$$

The rms value of the full-scale input sinewave is $V_O/\sqrt{2}$, therefore the rms signal to rms noise ratio is given by

$$SNR = 20\log_{10}\left[\frac{V_O/\sqrt{2}}{\Delta v_{rms}}\right] = 20\log_{10}\left[\frac{V_O/\sqrt{2}}{2\pi fV_O t_j/\sqrt{2}}\right] = 20\log_{10}\left[\frac{1}{2\pi f\, t_j}\right]. \qquad \text{Eq. 2.34}$$

This equation assumes an infinite resolution ADC where aperture jitter is the only factor in determining the SNR. This equation is plotted in Figure 2.80 and shows the serious effects of aperture and sampling clock jitter on SNR, especially at higher input/output frequencies. Therefore, extreme care must be taken to minimize phase noise in the sampling/reconstruction clock of any sampled data system.

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

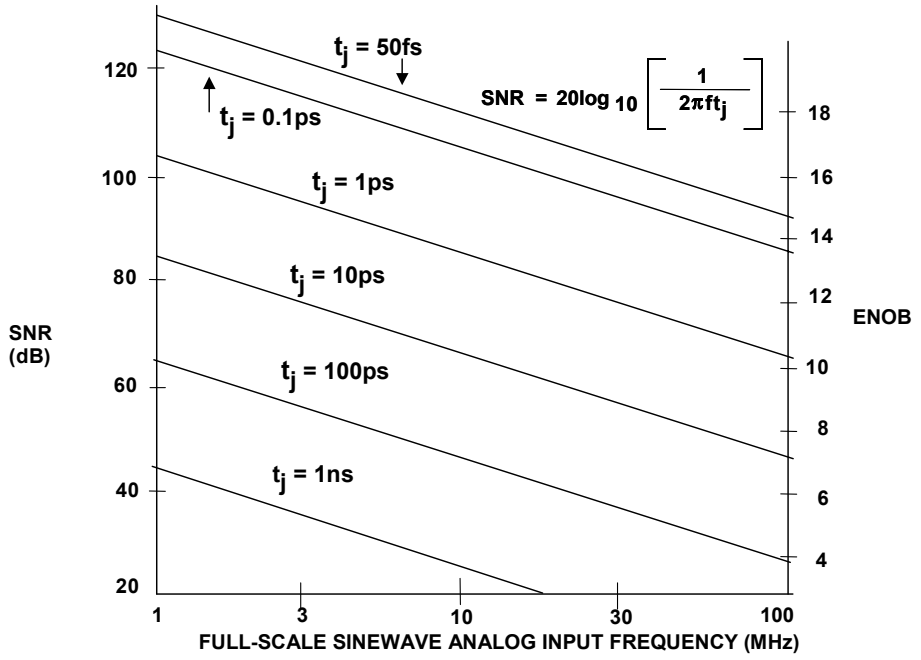


Figure 2.80: *Theoretical SNR and ENOB Due to Jitter vs. Fullscale Sinewave Input Frequency*

A Simple Equation for the Total SNR of an ADC

A relatively simple equation for the ADC SNR in terms of sampling clock and aperture jitter, DNL, effective input noise, and the number of bits of resolution is shown in Figure 2.81. The equation combines the various error terms on an rss basis. The average DNL error, ε, is computed from histogram data. This equation is used in Figure 2.82 to predict the SNR performance of the AD6645 14-bit, 80-MSPS ADC as a function of sampling clock and aperture jitter.

SAMPLING CLOCK JITTER — QUANTIZATION NOISE, DNL — EFFECTIVE INPUT NOISE

$$SNR = -20\log_{10}\left[(2\pi \times f_a \times t_{j\,rms})^2 + \frac{2}{3}\left(\frac{1+\varepsilon}{2^N}\right)^2 + \left(\frac{2\times\sqrt{2}\times V_{NOISErms}}{2^N}\right)^2\right]^{\frac{1}{2}}$$

f_a = **Analog input frequency of fullscale input sinewave**
$t_{j\,rms}$ = **Combined rms jitter of internal ADC and external clock**
ε = **Average DNL of the ADC (typically 0.41 LSB for AD6645)**
N = **Number of bits in the ADC**
$V_{NOISErms}$ = **Effective input noise of ADC (typically 0.9LSB rms for AD6645)**

If t_j = 0, ε = 0, and $V_{NOISErms}$ = 0, the above equation reduces to the familiar:

$$SNR = 6.02\,N + 1.76dB$$

Figure 2.81: *Relationship Between SNR, Sampling Clock Jitter, Quantization Noise, DNL, and Input Noise*

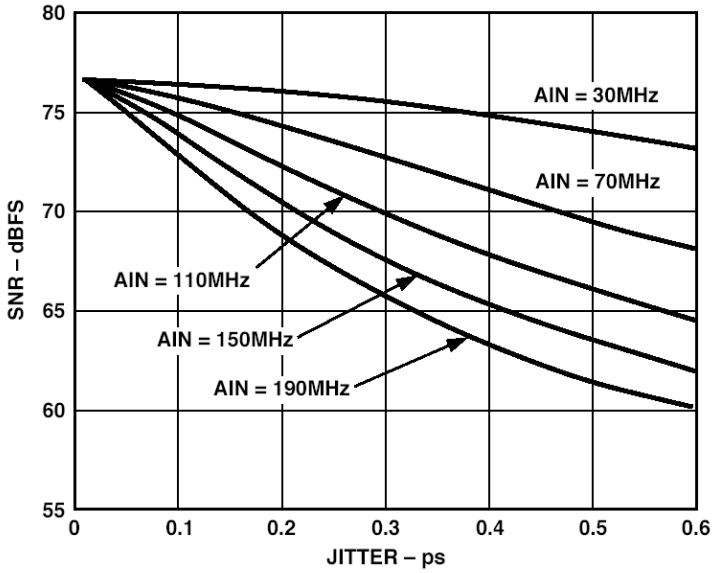


Figure 2.82: *AD6645 SNR Versus Jitter*

Before the 1980s, most sampling ADCs were generally built up from a separate SHA and ADC. Interface design was difficult, and a key parameter was aperture jitter in the SHA. Today, almost all sampled data systems use *sampling* ADCs which contain an integral SHA. The aperture jitter of the SHA may not be specified as such, but this is not a cause of concern if the SNR or ENOB is clearly specified, since a guarantee of a specific SNR is an implicit guarantee of an adequate aperture jitter specification. However, the use of an additional high-performance SHA will sometimes improve the high-frequency ENOB of even the best sampling ADC by presenting "dc" to the ADC, and may be more cost-effective than replacing the ADC with a more expensive one.

ADC Transient Response and Overvoltage Recovery

Most high speed ADCs designed for communications applications are specified primarily in the frequency domain. However, in general purpose data acquisition applications the transient response (or settling time) of the ADC is important. The *transient response* of an ADC is the time required for the ADC to settle to rated accuracy (usually 1 LSB) after the application of a full-scale step input. The typical response of a general-purpose 12 bit, 10-MSPS ADC is shown in Figure 2.83, showing a 1 LSB settling time of less than 40 ns. The settling time specification is critical in the typical data acquisition system application where the ADC is being driven by an analog multiplexer as shown in Figure 2.84. The multiplexer output can deliver a full-scale sample-to-sample change to the ADC input. If both the multiplexer and the ADC have not both settled to the required accuracy, channel-to-channel crosstalk will result, even though only dc or low frequency signals are present on the multiplexer inputs.

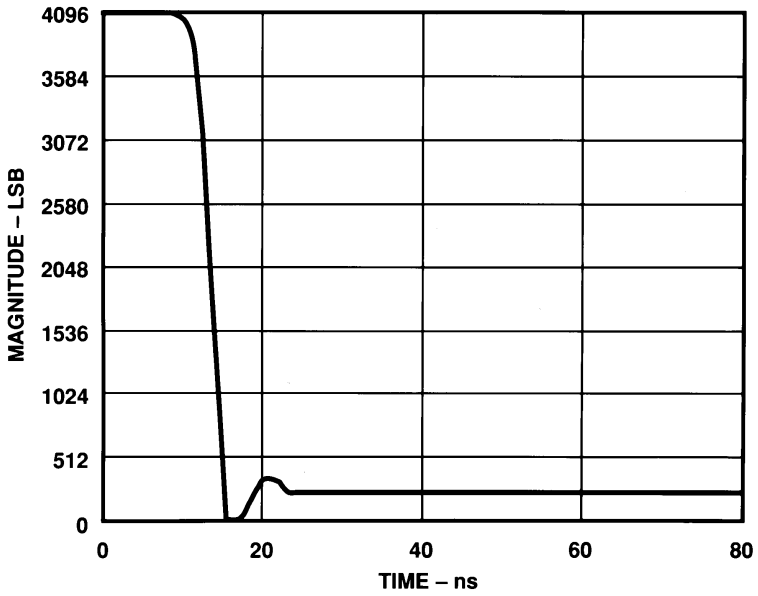


Figure 2.83: *ADC Transient Response (Settling Time)*

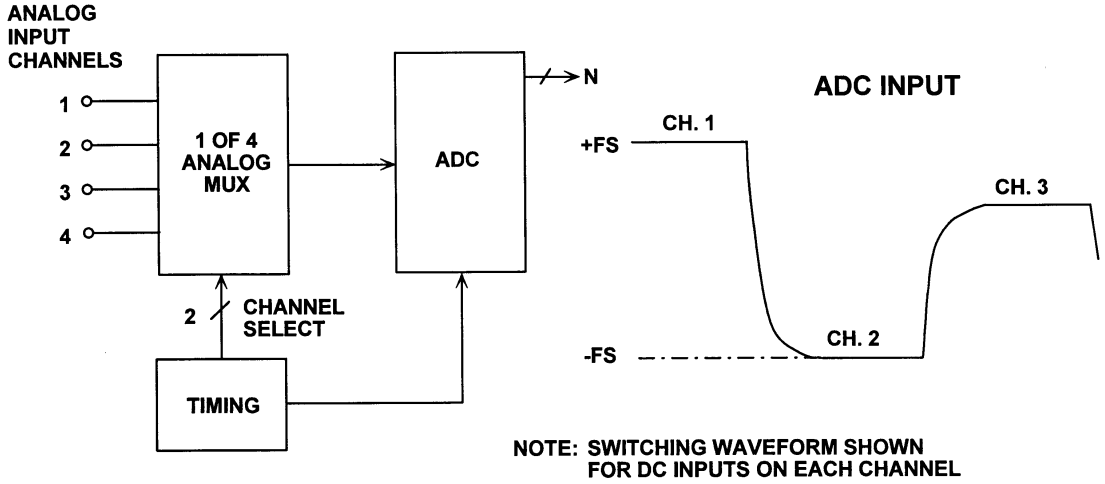


Figure 2.84: *Settling Time is Critical in Multiplexed Applications*

Most ADCs have settling times which are less than $1/f_{s\,max}$, even if not specified. However sigma-delta ADCs have a built in digital filter which can take several output sample intervals to settle. This should be kept in mind when using sigma-delta ADCs in multiplexed applications.

The importance of settling time in multiplexed systems can be seen in Figure 2.85, where the ADC input is modeled as a single-pole filter having a corresponding time constant, $\tau = RC$. The required number of time constants to settle to a given accuracy (1 LSB) is shown. A simple example will illustrate the point.

Assume a multiplexed 16-bit data acquisition system uses an ADC with a sampling frequency f_s = 100 kSPS. The ADC must settle to 16-bit accuracy for a full-scale step function input in less than $1/f_s$ = 10 µs. The chart shows that 11.09 time constants are required to settle to 16-bit accuracy. The input filter time constant must therefore be less

than $\tau = 10\ \mu s/11.09 = 900$ ns. The corresponding risetime $t_r = 2.2\tau = 1.98\ \mu s$. The required ADC full power input bandwidth can now be calculated from $BW = 0.35/t_r = 177$ kHz. This neglects the settling time of the multiplexer and second-order settling time effects in the ADC.

RESOLUTION, # OF BITS	LSB (%FS)	# OF TIME CONSTANTS
6	1.563	4.16
8	0.391	5.55
10	0.0977	6.93
12	0.0244	8.32
14	0.0061	9.70
16	0.00153	11.09
18	0.00038	12.48
20	0.000095	13.86
22	0.000024	15.25

Figure 2.85: *Settling Time as a Function of Time Constant for Various Resolutions*

Overvoltage recovery time is defined as that amount of time required for an ADC to achieve a specified accuracy, measured from the time the overvoltage signal re-enters the converter's range, as shown in Figure 2.86. This specification is usually given for a signal which is some stated percentage outside the ADC's input range. Needless to say, the ADC should act as an ideal limiter for out-of-range signals and should produce either the positive full-scale code or the negative full-scale code during the overvoltage condition. Some converters provide over- and under-range flags to allow gain-adjustment circuits to be activated. Care should always be taken to avoid overvoltage signals which will damage an ADC input.

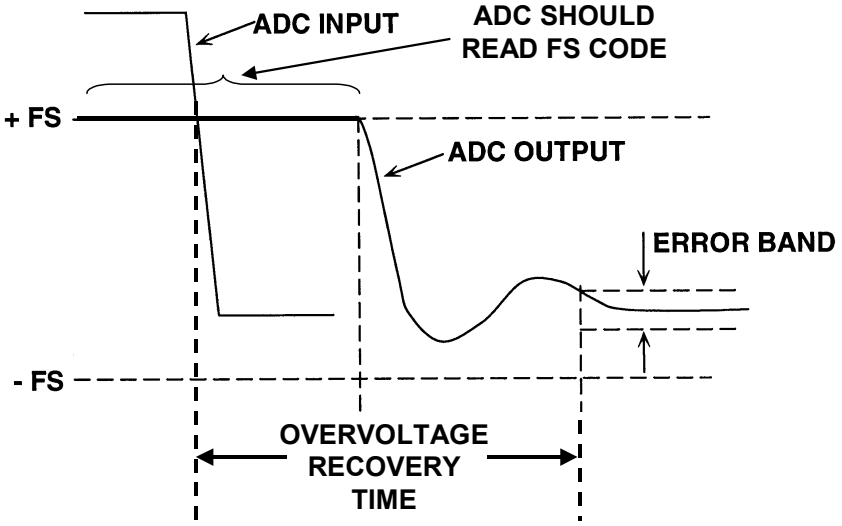


Figure 2.86: *Overvoltage Recovery Time*

ADC Sparkle Codes, Metastable States, and Bit Error Rate (BER)

A primary concern in the design of many digital communications systems using ADCs is the bit error rate (BER). Unfortunately, ADCs contribute to the BER in ways that are not predictable by simple analysis. This section describes the mechanisms within the ADCs that can contribute to the error rate, ways to minimize the problem, and methods for measuring the BER.

Random noise, regardless of the source, creates a finite probability of errors (deviations from the expected output). Before describing the error code sources, however, it is important to define what constitutes an ADC error code. Noise generated prior to, or inside the ADC can be analyzed in the traditional manner. Therefore, an ADC error code is any deviation from the expected output that is not attributable to the equivalent input noise of the ADC. Figure 2.87 illustrates an exaggerated output of a low-amplitude sinewave applied to an ADC that has error codes. Note that the noise of the ADC creates some uncertainty in the output. These anomalies are not considered error codes, but are simply the result of ordinary noise and quantization. The large errors are more significant and are not expected. These errors are random and so infrequent that an SNR test of the ADC will rarely detect them. These types of errors plagued a few of the early ADCs for video applications, and were given the name *sparkle codes* because of their appearance on a TV screen as small white dots or "sparkles" under certain test conditions. These errors have also been called *rabbits* or *flyers*. In digital communications applications, this type of error increases the overall system bit error rate (BER).

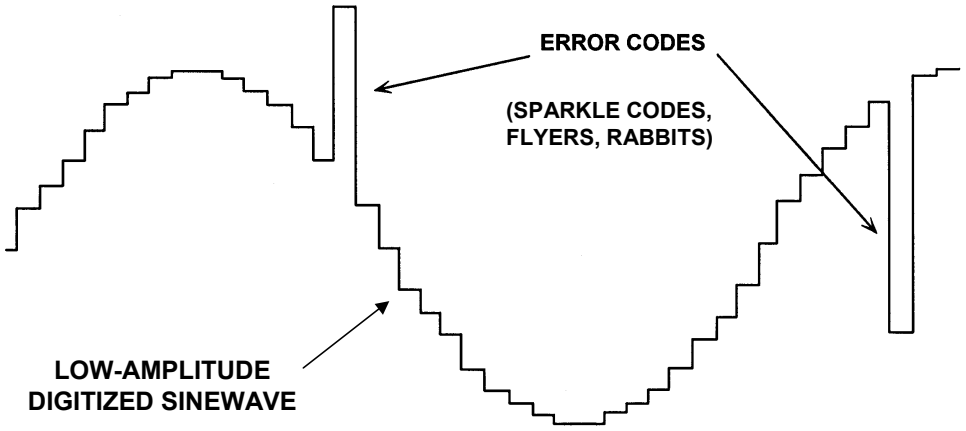


Figure 2.87: *Exaggerated Output of ADC Showing Error Codes*

In order to understand the causes of the error codes, we will first consider the case of a simple flash converter. The comparators in a flash converter are latched comparators usually arranged in a master-slave configuration. If the input signal is in the center of the threshold of a particular comparator, that comparator will balance, and its output will take a longer period of time to reach a valid logic level after the application of the latch strobe than the outputs of its neighboring comparators which are being overdriven. This phenomenon is known as *metastability* and occurs when a balanced comparator cannot reach a valid logic level in the time allowed for decoding. If simple binary decoding logic is used to decode the thermometer code, a metastable comparator output may result in a

large output code error. Consider the case of a simple 3 bit flash converter shown in Figure 2.88. Assume that the input signal is exactly at the threshold of Comparator 4 and random noise is causing the comparator to toggle between a "1" and a "0" output each time a latch strobe is applied. The corresponding binary output should be interpreted as either 011 or 100. If, however, the comparator output is in a metastable state, the simple binary decoding logic shown may produce binary codes 000, 011, 100, or 111. The codes 000 and 111 represent a one-half scale departure from the expected codes.

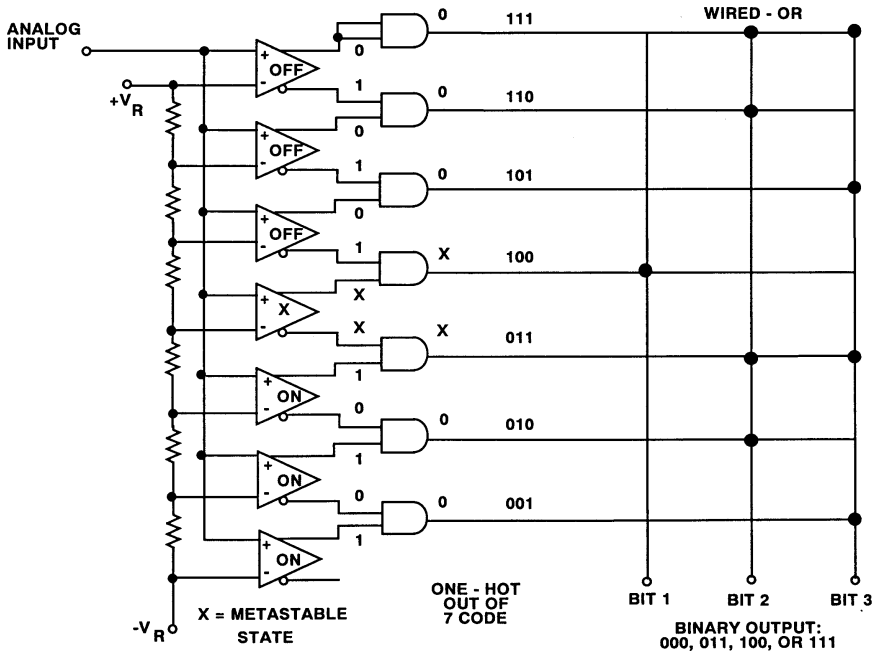


Figure 2.88: *Metastable Comparator Output States May Cause Error Codes in Data Converters*

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

Various measures have been taken in flash converter designs to minimize the metastable state problem. Decoding schemes described in References 12 to 15 minimize the magnitude of these errors. Optimizing comparator designs for regenerative gain and small time constants is another way to reduce these problems.

Metastable state errors may also appear in successive approximation and subranging ADCs which make use of comparators as building blocks. The same concepts apply, although the magnitudes and locations of the errors may be different.

The test system shown in Figure 2.89 may be used to test for BER in an ADC. The analog input to the ADC is provided by a high stability low noise sinewave generator. The analog input level is set slightly greater than full-scale, and the frequency such that there is always slightly less than 1 LSB change between samples as shown in Figure 2.90.

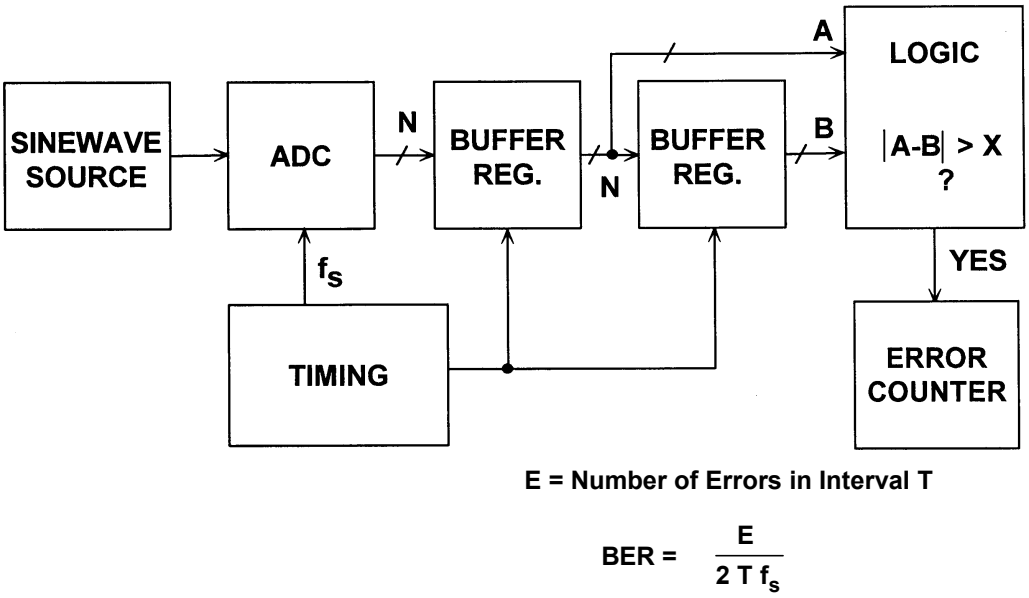


Figure 2.89: *ADC Bit Error Rate Test Setup*

The test set uses series latches to acquire successive codes A and B. A logic circuit determines the absolute difference between A and B. This difference is then compared to the error limit, chosen to allow for expected random noise spikes and ADC quantization errors. Errors which cause the difference to be larger than the limit will increment the counters. The number of errors, E, are counted over a period of time, T. The error rate is then calculated as BER = $E/2Tf_s$. The factor of 2 in the denominator is required because the hardware records a second error when the output returns to the correct code after making the initial error. The error counter therefore is incremented twice for each error. It should be noted that the same function can be accomplished in software if the ADC outputs are stored in a memory and analyzed by a computer program.

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

$$v(t) = \frac{2^N}{2} \sin 2\pi ft \qquad \text{Eq. 2.35}$$

The maximum rate of change of this signal is

$$\left.\frac{dv}{dt}\right]_{max} \leq 2^N \pi f \,. \qquad \text{Eq. 2.36}$$

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

$$f_{in} \leq \frac{f_s}{2^N \pi} \,. \qquad \text{Eq. 2.37}$$

Choosing an input frequency less than this value will ensure that there is at least one sample per code.

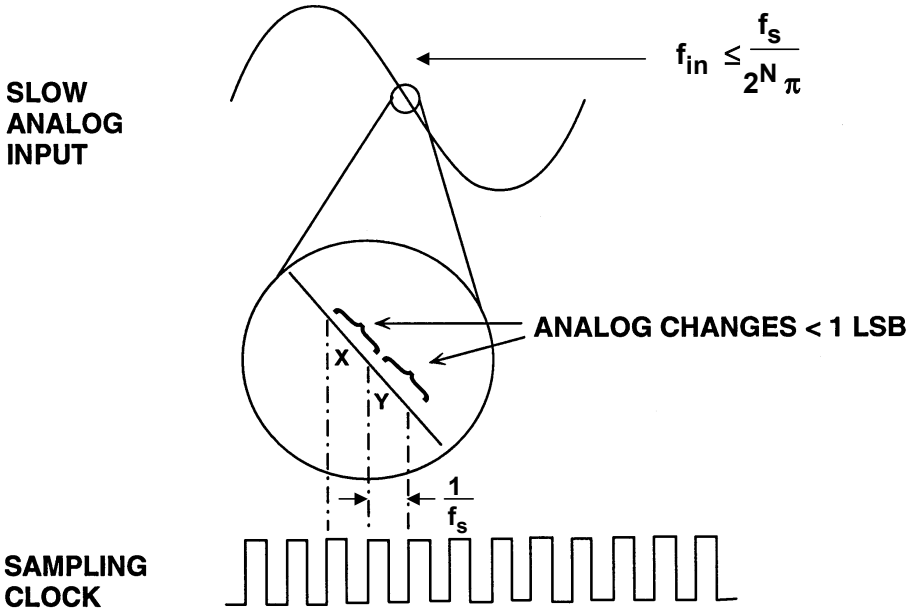


Figure 2.90: *ADC Analog Signal for Low Frequency BER Test*

The same test can be conducted at high frequencies by applying an input frequency slightly offset from $f_s/2$ as shown in Figure 2.91. This causes the ADC to slew full-scale between conversions. Every other conversion is compared, and the "beat" frequency is chosen such that there is slightly less than 1 LSB change between alternate samples. The equation for calculating the proper frequency for the high frequency BER test is derived as follows.

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

$$v(t) = \frac{2^N}{2} \sin\left[2\pi \left(\frac{f_s}{2} - \Delta f \right) t \right]. \qquad \text{Eq. 2.38}$$

The maximum rate of change of this signal is

$$\left. \frac{dv}{dt} \right]_{max} \leq 2^N \pi \left(\frac{f_s}{2} - \Delta f \right). \qquad \text{Eq. 2.39}$$

Letting dv = 1 LSB and $dt = 2/f_s$, and solving for the input frequency Δf:

$$\Delta f \leq \frac{f_s}{2} \left(1 - \frac{1}{2 \cdot 2^N \pi} \right). \qquad \text{Eq. 2.40}$$

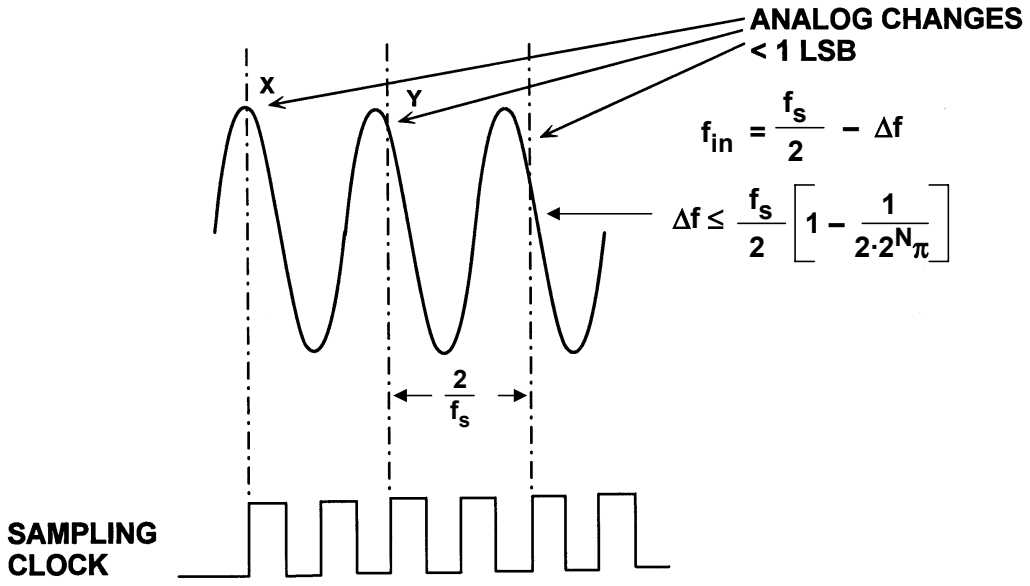


Figure 2.91: *ADC Analog Input for High Frequency BER Test*

Establishing the BER of a well-behaved ADC is a difficult, time-consuming task; a single unit can sometimes be tested for days without an error. For example, tests on a typical 8-bit flash converter operating at a sampling rate of 75 MSPS yield a BER of approximately 3.7×10^{-12} (1 error per hour) with an error limit of 4 LSBs. Meaningful tests for longer periods of time require special attention to EMI/RFI effects (possibly requiring a shielded screen room), isolated power supplies, isolation from soldering irons with mechanical thermostats, isolation from other bench equipment, etc. Figure 2.92 shows the average time between errors as a function of BER for a sampling frequency of 75 MSPS. This illustrates the difficulty in measuring low BER because the long measurement times increase the probability of power supply transients, noise, etc. causing an error.

Bit Error Rate (BER)	Average Time Between Errors
1×10^{-8}	1.3 seconds
1×10^{-9}	13.3 seconds
1×10^{-10}	2.2 minutes
1×10^{-11}	22 minutes
1×10^{-12}	3.7 hours
1×10^{-13}	1.5 days
1×10^{-14}	15 days

Figure 2.92: *Average Time Between Errors Versus BER when Sampling at 75 MSPS*

DAC Dynamic Performance

The ac specifications which are most likely to be important with DACs are *settling time*, *glitch impulse area*, *distortion,* and *Spurious Free Dynamic Range (SFDR)*.

DAC Settling Time

The input to output settling time of a DAC is the time from a change of digital code (t = 0) to when the output comes within *and remains within* some error band as shown in Figure 2.93. With amplifiers, it is hard to make comparisons of settling time, since their specified error bands may differ from amplifier to amplifier, but with DACs the error band will almost invariably be specified as ±1 or ±½ LSB. Note that in some cases, the *output* settling time may be of more interest, in which case it is referenced to the time the output first leaves the error band.

The input to output settling time of a DAC is made up of four different periods: the *switching time* or *dead time* (during which the digital switching, but not the output, is changing), the *slewing time* (during which the rate of change of output is limited by the slew rate of the DAC output), the *recovery time* (when the DAC is recovering from its fast slew and may overshoot), and the *linear settling time* (when the DAC output approaches its final value in an exponential or near-exponential manner). If the slew time is short compared to the other three (as is usually the case with current output DACs), then the settling time will be largely independent of the output step size. On the other hand, if the slew time is a significant part of the total, then the larger the step, the longer the settling time.

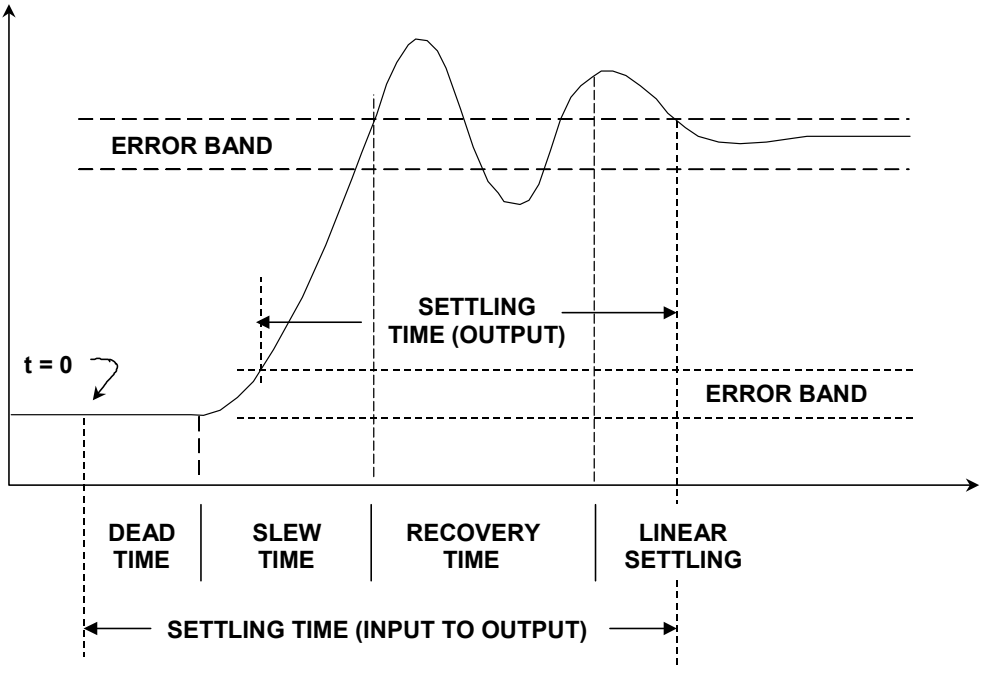


Figure 2.93: *DAC Settling Time*

Settling time is especially important in video display applications. For example a standard 1024×768 display updated at a 60-Hz refresh rate must have a pixel rate of

1024×768×60 Hz = 47.2 MHz with no overhead. Allowing 35% overhead time increases the pixel frequency to 64 MHz corresponding to a pixel duration of $1/(64\times10^6) = 15.6$ ns. In order to accurately reproduce a single fully-white pixel located between two black pixels, the DAC settling time should be less than the pixel duration time of 15.6 ns.

Higher resolution displays require even faster pixel rates. For example, a 2048×2048 display requires a pixel rate of approximately 330 MHz at a 60-Hz refresh rate.

Glitch Impulse Area

Ideally, when a DAC output changes it should move from one value to its new one monotonically. In practice, the output is likely to overshoot, undershoot, or both (see Figure 2.94). This uncontrolled movement of the DAC output during a transition is known as a *glitch*. It can arise from two mechanisms: capacitive coupling of digital transitions to the analog output, and the effects of some switches in the DAC operating more quickly than others and producing temporary spurious outputs.

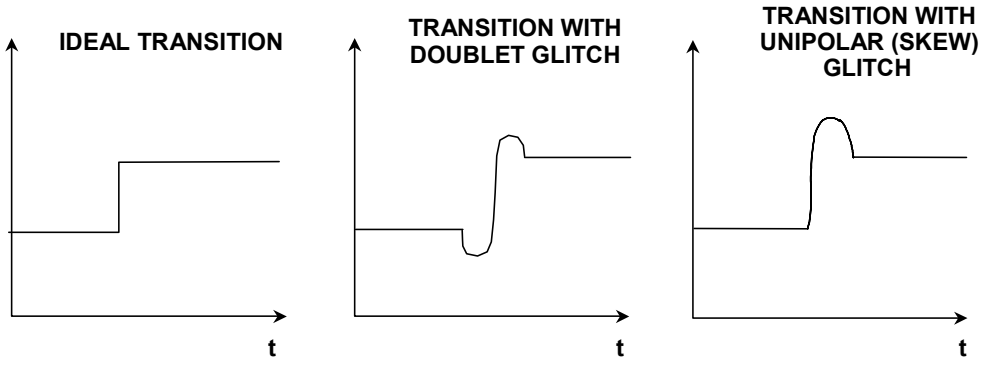


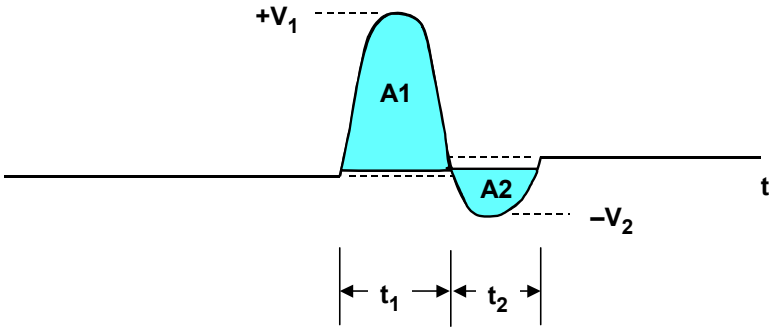
Figure 2.94: *DAC Transitions (Showing Glitch)*

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

Glitches can be characterized by measuring the *glitch impulse area*, sometimes inaccurately called glitch energy. The term *glitch energy* is a misnomer, since the unit for glitch impulse area is volt-seconds (or more probably μV-sec or pV-sec. The *peak glitch area* is the area of the largest of the positive or negative glitch areas. The glitch impulse area is the net area under the voltage-versus-time curve and can be estimated by approximating the waveforms by triangles, computing the areas, and subtracting the negative area from the positive area as shown in Figure 2.95.

The mid-scale glitch produced by the transition between the codes 0111...111 and 1000...000 is usually the worst glitch because all switches are changing states. Glitches at other code transition points (such as 1/4 and 3/4 full-scale) are generally less. Figure 2.96 shows the mid-scale glitch for a fast low-glitch DAC. The peak and net glitch areas are estimated using triangles as described above. Settling time is measured from the time the

waveform leaves the initial 1 LSB error band until it enters and remains within the final 1-LSB error band. The step size between the transition regions is also 1 LSB.



- PEAK GLITCH IMPULSE AREA = A1 ≈ $\frac{V_1 \cdot t_1}{2}$
- NET GLITCH IMPULSE AREA = A1 – A2 ≈ $\frac{V_1 \cdot t_1}{2} - \frac{V_2 \cdot t_2}{2}$

Figure 2.95: *Calculating Net Glitch Impulse Area*

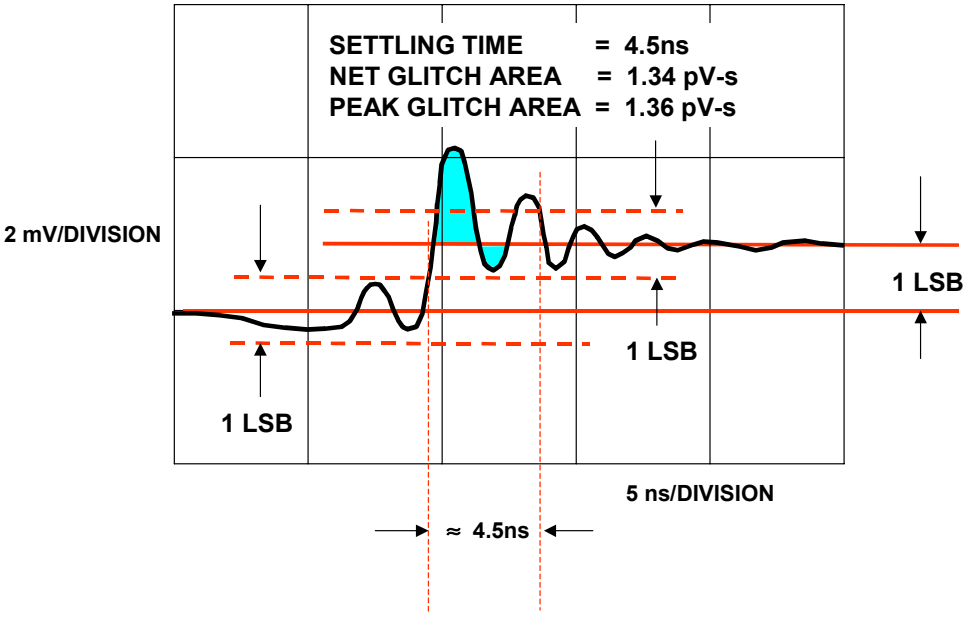


Figure 2.96: *DAC Mid-scale Glitch Shows 1.34pV-s Net Impulse Area and Settling Time of 4.5ns*

DAC SFDR and SNR

DAC settling time is important in applications such as RGB raster scan video display drivers, but frequency-domain specifications such as SFDR are generally more important in communications.

If we consider the spectrum of a waveform reconstructed by a DAC from digital data, we find that in addition to the expected spectrum (which will contain one or more frequencies, depending on the nature of the reconstructed waveform), there will also be noise and distortion products. Distortion may be specified in terms of harmonic distortion, Spurious Free Dynamic Range (SFDR), intermodulation distortion, or all of the above. Harmonic distortion is defined as the ratio of harmonics to fundamental when a (theoretically) pure sine wave is reconstructed, and is the most common specification. Spurious free dynamic range is the ratio of the worst spur (usually, but not necessarily always a harmonic of the fundamental) to the fundamental.

Code-dependent glitches will produce both out-of-band and in-band harmonics when the DAC is reconstructing a digitally generated sinewave as in a Direct Digital Synthesis (DDS) system. The mid-scale glitch occurs twice during a single cycle of a reconstructed sinewave (at each mid-scale crossing), and will therefore produce a second harmonic of the sinewave, as shown in Figure 2.97. Note that the higher order harmonics of the sinewave, which alias back into the Nyquist bandwidth (dc to $f_s/2$), cannot be filtered.

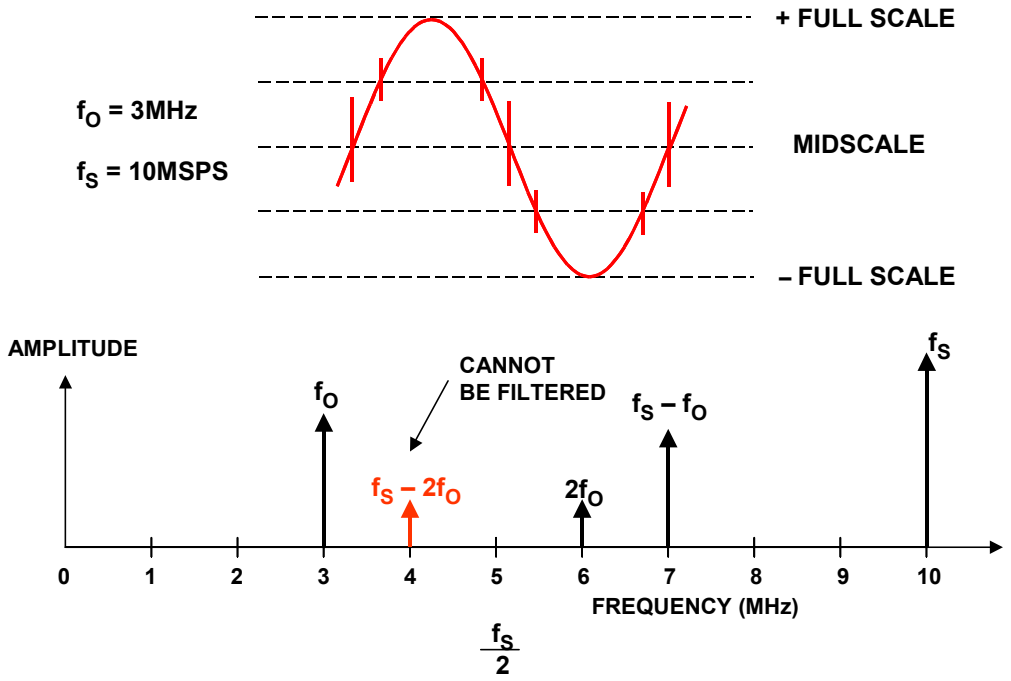


Figure 2.97: *Effect of Code-Dependent Glitches on Spectral Output*

It is difficult to predict the harmonic distortion or SFDR from the glitch area specification alone. Other factors, such as the overall linearity of the DAC, also contribute to distortion as shown in Figure 2.98. In addition, certain ratios between the DAC output frequency and the sampling clock cause the quantization noise to concentrate at harmonics of the fundamental thereby increasing the distortion at these points.

It is therefore customary to test reconstruction DACs in the frequency domain (using a spectrum analyzer) at various clock rates and output frequencies as shown in Figure 2.99. Typical SFDR for the 16-bit AD9777 Transmit TxDAC™ is shown in Figure 2.100. The clock rate is 160 MSPS, and the output frequency is swept to 50 MHz. As in the case of

ADCs, quantization noise will appear as increased harmonic distortion if the ratio between the clock frequency and the DAC output frequency is an integer number. These ratios should be avoided when making the SFDR measurements.

- **Resolution**
- **Integral Non-Linearity**
- **Differential Non-Linearity**
- **Code-Dependent Glitches**
- **Ratio of Clock Frequency to Output Frequency (Even in an Ideal DAC)**
- **Mathematical Analysis is Difficult !**

Figure 2.98: *Contributors to DDS DAC Distortion*

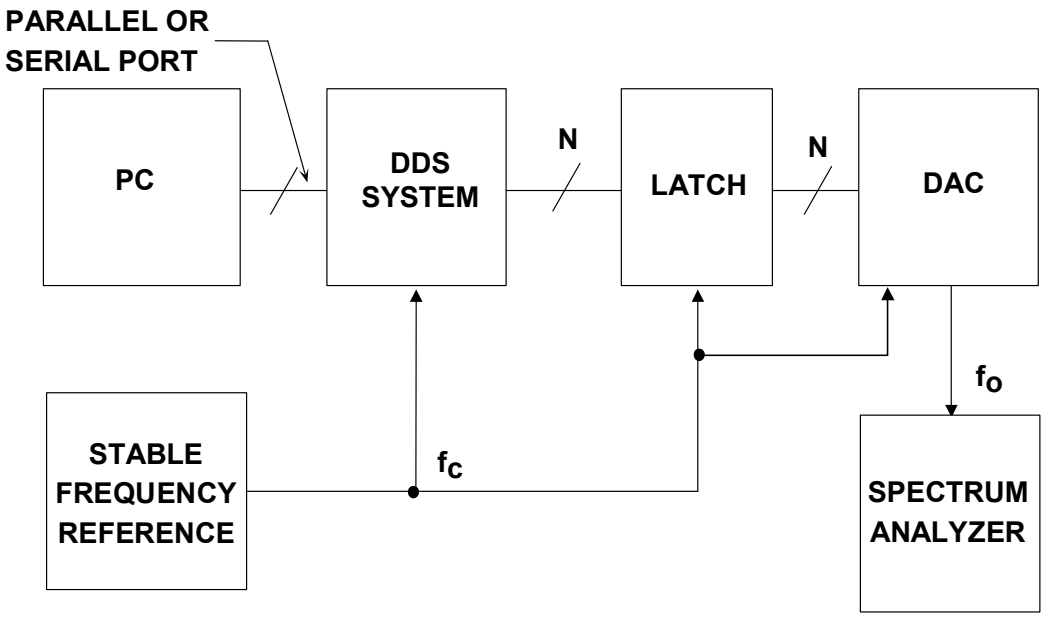


Figure 2.99: *Test Setup for Measuring DAC SFDR*

There are nearly an infinite combination of possible clock and output frequencies for a low distortion DAC, and SFDR is generally specified for a limited number of selected combinations. For this reason, Analog Devices offers fast turnaround on customer-specified test vectors for the Transmit TxDAC™ family. A test vector is a combination of amplitudes, output frequencies, and update rates specified directly by the customer for SFDR data on a particular DAC.

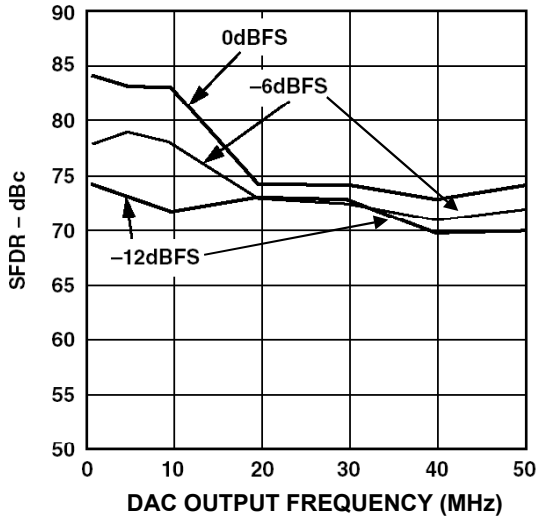


Figure 2.100: *AD9777 16-bit TxDAC® SFDR, Data Update Rate = 160 MSPS*

Measuring DAC SNR with an Analog Spectrum Analyzer

Analog spectrum analyzers are used to measure the distortion and SFDR of high performance DACs. Care must be taken such that the front end of the analyzer is not overdriven by the fundamental signal. If overdrive is a problem, a bandstop filter can be used to filter out the fundamental signal such that the spurious components can be observed.

Spectrum analyzers can also be used to measure the SNR of a DAC provided attention is given to bandwidth considerations. SNR of an ADC is normally defined as the signal-to-noise ratio measured over the Nyquist bandwidth dc to $f_s/2$. However, spectrum analyzers have a resolution bandwidth which is less than $f_s/2$—this therefore lowers the analyzer noise floor by the process gain equal to $10 \log_{10}[f_s/(2 \cdot BW)]$, where BW is the resolution noise bandwidth of the analyzer (see Figure 2.101).

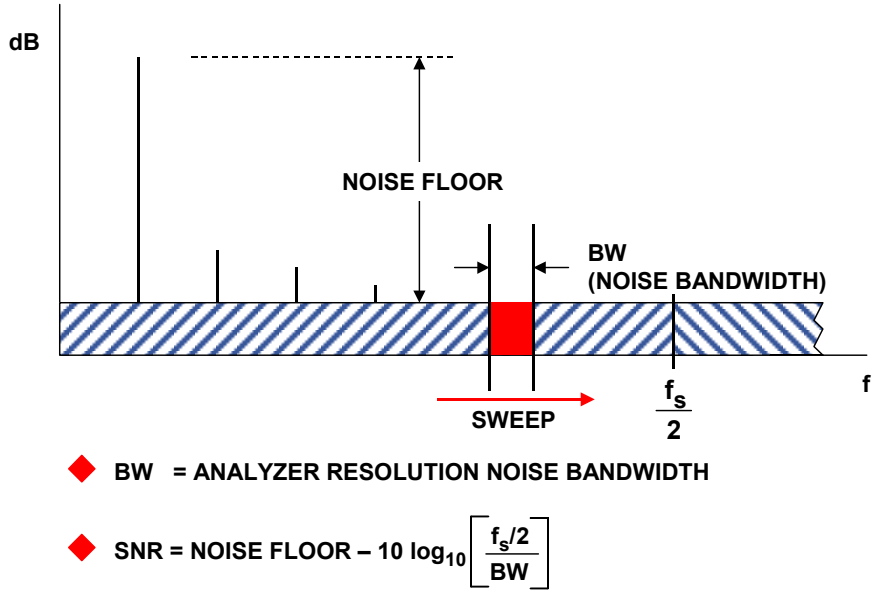


Figure 2.101: *Measuring DAC SNR with an Analog Spectrum Analyzer*

It is important that the noise bandwidth (not the 3-dB bandwidth) be used in the calculation, however from Figure 2.68 the error is small assuming that the analyzer narrowband filter is at least two poles. The ratio of the noise bandwidth to the 3-dB bandwidth of a one-pole Butterworth filter is 1.57 (causing an error of 1.96 dB in the process gain calculation). For a two-pole Butterworth filter, the ratio is 1.11 (causing an error of 0.45 dB in the process gain calculation).

DAC Output Spectrum and sin (x)/x Frequency Rolloff

The output of a reconstruction DAC can be represented as a series of rectangular pulses whose width is equal to the reciprocal of the clock rate as shown in Figure 2.102. Note that the reconstructed signal amplitude is down 3.92 dB at the Nyquist frequency, $f_c/2$. An inverse sin(x)/x filter can be used to compensate for this effect in most cases. The images of the fundamental signal occur as a result of the sampling function and are also attenuated by the sin(x)/x function.

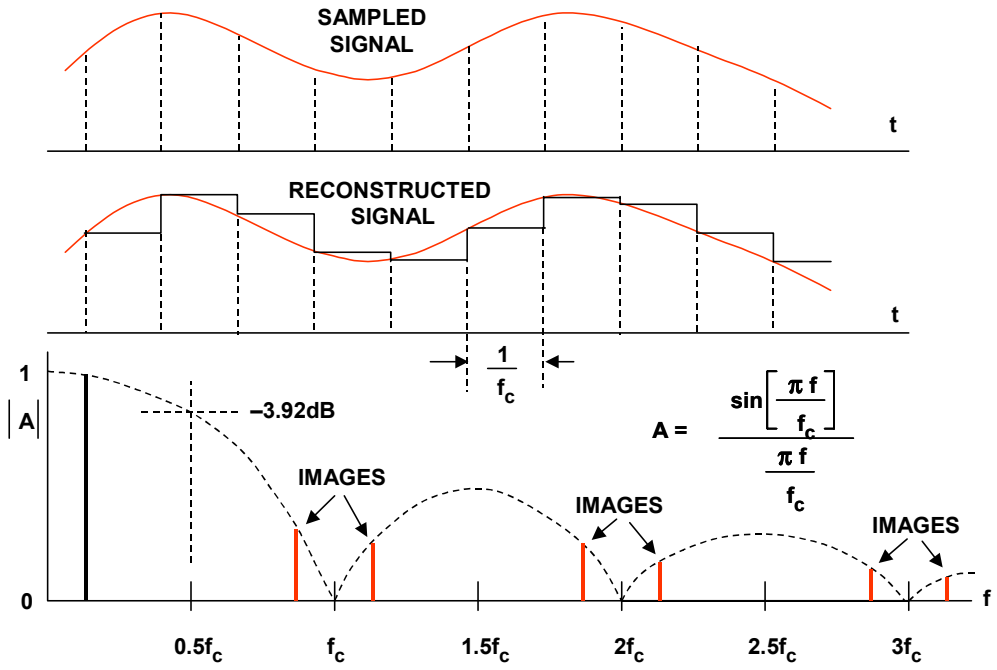


Figure 2.102: *DAC sin x/x Roll Off (Amplitude Normalized)*

Oversampling Interpolating DACs

In ADC-based systems, oversampling can ease the requirements on the antialiasing filter. In a DAC-based system (such as DDS), the concept of interpolation can be used in a similar manner. This concept is common in digital audio CD players, where the basic update rate of the data from the CD is 44.1 kSPS. Early CD players used traditional binary DACs and inserted "Zeros" into the parallel data, thereby increasing the effective update rate to 4-times, 8-times, or 16-times the fundamental throughput rate . The 4×, 8×, or 16× data stream is passed through a digital interpolation filter which generates the extra data points. The high oversampling rate moves the image frequencies higher, thereby allowing a less complex filter with a wider transition band. The sigma-delta 1-bit

DAC architecture uses a much higher oversampling rate and represents the ultimate extension of this concept and has become popular in modern CD players.

The same concept of oversampling can be applied high speed DACs used in communications applications, relaxing the requirements on the output filter as well as increasing the SNR due to process gain.

Assume a traditional DAC is driven at an input word rate of 30 MSPS (see Figure 2.103A). Assume the DAC output frequency is 10 MHz. The image frequency component at 30 – 10 = 20 MHz must be attenuated by the analog antialiasing filter, and the transition band of the filter is 10 to 20 MHz. Assume that the image frequency must be attenuated by 60 dB. The filter must therefore go from a passband of 10 MHz to 60 dB stopband attenuation over the transition band lying between 10 and 20 MHz (one octave). A filter gives 6-dB attenuation per octave for each pole. Therefore, a minimum of 10 poles is required to provide the desired attenuation. Filters become even more complex as the transition band becomes narrower.

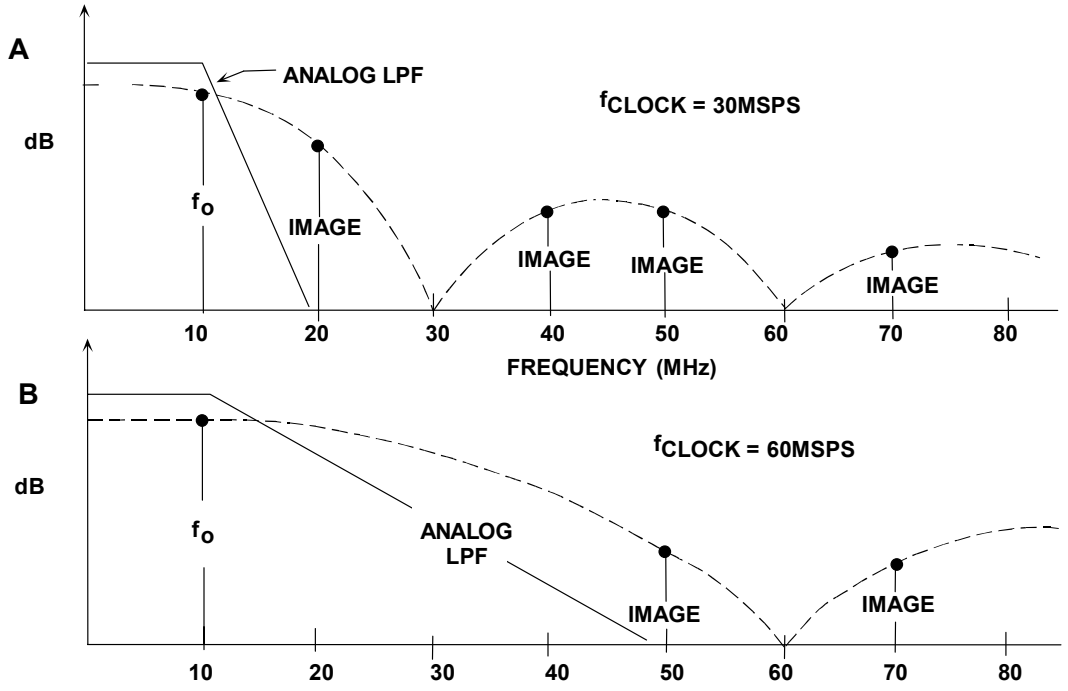


Figure 2.103: *Analog Filter Requirements for f_o = 10 MHz: (A) f_c = 30 MSPS, and (B) f_c = 60 MSPS*

Assume that we increase the DAC update rate to 60 MSPS and insert a "zero" between each original data sample. The parallel data stream is now 60 MSPS, but we must now determine the value of the zero-value data points. This is done by passing the 60-MSPS data stream with the added zeros through a digital interpolation filter which computes the additional data points. The response of the digital filter relative to the 2-times oversampling frequency is shown in Figure 2.103B. The analog antialiasing filter transition zone is now 10 to 50 MHz (the first image occurs at $2f_c - f_o = 60 - 10 =$ 50 MHz). This transition zone is a little greater than 2 octaves, implying that a 5- or 6-pole filter is sufficient.

The AD9773/AD9775/AD9777 (12/14/16-bit) series of Transmit DACs (TxDAC®) are selectable 2×, 4×, or 8× oversampling interpolating dual DACs, and a simplified block diagram is shown in Figure 2.103. These devices are designed to handle 12/14/16-bit input word rates up to 160 MSPS. The output word rate is 400 MSPS maximum. For an output frequency of 50 MHz, an input update rate of 160 MHz, and an oversampling ratio of 2×, the image frequency occurs at 320 MHz – 50 MHz = 270 MHz. The transition band for the analog filter is therefore 50 MHz to 270 MHz. Without 2× oversampling, the image frequency occurs at 160 MHz – 50 MHz = 110 MHz, and the filter transition band is 50 MHz to 110 MHz.

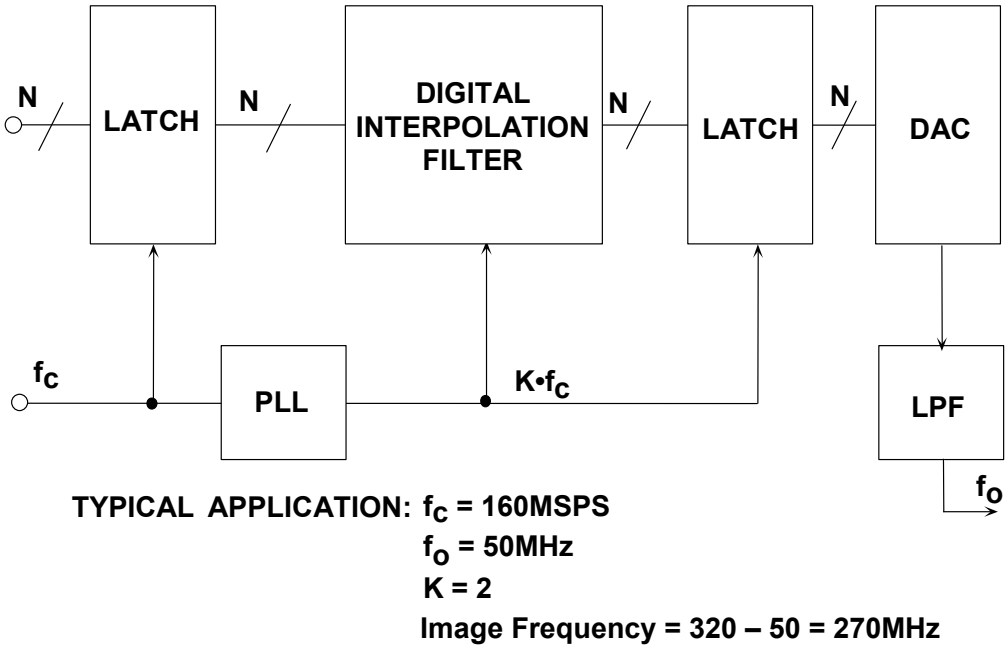


Figure 2.104: *Oversampling Interpolating TxDAC® Simplified Block Diagram*

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

REFERENCES:
2.3 DATA CONVERTER AC ERRORS

1. W. R. Bennett, "Spectra of Quantized Signals," **Bell System Technical Journal**, Vol. 27, July 1948, pp. 446-471.

2. B. M. Oliver, J. R. Pierce, and C. E. Shannon, "The Philosophy of PCM," **Proceedings IRE**, Vol. 36, November 1948, pp. 1324-1331.

3. W. R. Bennett, "Noise in PCM Systems," **Bell Labs Record**, Vol. 26, December 1948, pp. 495-499.

4. H. S. Black and J. O. Edson, "Pulse Code Modulation," **AIEE Transactions**, Vol. 66, 1947, pp. 895-899.

5. H. S. Black, "Pulse Code Modulation," **Bell Labs Record**, Vol. 25, July 1947, pp. 265-269.

6. Steve Ruscak and Larry Singer, *Using Histogram Techniques to Measure A/D Converter Noise*, **Analog Dialogue**, Vol. 29-2, 1995.

7. M.J. Tant, **The White Noise Book**, Marconi Instruments, July 1974.

8. G.A. Gray and G.W. Zeoli, *Quantization and Saturation Noise due to A/D Conversion*, **IEEE Trans. Aerospace and Electronic Systems**, Jan. 1971, pp. 222-223.

9. Kevin McClaning and Tom Vito, **Radio Receiver Design**, Noble Publishing, 2000, ISBN 1-88-4932-07-X.

10. Walter G. Jung, editor, **Op Amp Applications**, Analog Devices, Inc., 2002, ISBN 0-916550-26-5, pp. 6.144-6.152.

11. Brad Brannon, *Aperture Uncertainty and ADC System Performance*, **Application Note AN-501**, Analog Devices, Inc., January 1998. (available for download at http://www.analog.com)

12. Christopher W. Mangelsdorf, *A 400-MHz Input Flash Converter with Error Correction*, **IEEE Journal of Solid-State Circuits**, Vol. 25, No. 1, February 1990, pp. 184-191.

13. Charles E. Woodward, *A Monolithic Voltage-Comparator Array for A/D Converters*, **IEEE Journal of Solid State Circuits**, Vol. SC-10, No. 6, December 1975, pp. 392-399.

14. Yukio Akazawa et. al., *A 400MSPS 8 Bit Flash A/D Converter*, **1987 ISSCC Digest of Technical Papers**, pp. 98-99.

15. A.. Matsuzawa *et al*., *An 8b 600MHz Flash A/D Converter with Multi-stage Duplex-gray Coding*, **Symposium VLSI Circuits, Digest of Technical Papers**, May 1991, pp. 113-114.

16. Ron Waltman and David Duff, *Reducing Error Rates in Systems Using ADCs,* **Electronics Engineer**, April 1993, pp. 98-104.

17. K. W. Cattermole, **Principles of Pulse Code Modulation**, American Elsevier Publishing Company, Inc., 1969, New York NY, ISBN 444-19747-8. *(An excellent tutorial and historical discussion of data conversion theory and practice, oriented towards PCM, but covers practically all aspects. This one is a must for anyone serious about data conversion!*

18. Robert A. Witte, *Distortion Measurements Using a Spectrum Analyzer*, **RF Design**, September, 1992, pp. 75-84.

19. Walt Kester, *Confused About Amplifier Distortion Specs*?, **Analog Dialogue**, 27-1, 1993, pp. 27-29.

20. Dan Sheingold, Editor, **Analog-to-Digital Conversion Handbook**, **Third Edition**, Prentice-Hall, 1986.

21. Fred H. Irons, "The Noise Power Ratio—Theory and ADC Testing," **IEEE Transactions on Instrumentation and Measurement**, Vol. 49, No. 3, June 2000, pp. 659-665.

NOTES:

SECTION 2.4: GENERAL DATA CONVERTER SPECIFICATIONS

James Bryant

Overall Considerations

Data converters, as we have observed, have a digital port and an analog port, and like all integrated circuits they require power supplies and will draw current from those supplies. Data converter specifications will therefore include the usual specifications common to any integrated circuit, including supply voltage and supply current, logic interfaces, power on and standby timing, package and thermal issues and ESD. We shall not consider these at any length, but there are some issues which may require a little consideration.

An over-riding piece of advice here is *read the data sheet*. There is no excuse for being unaware of the specifications of a device for which one owns a data sheet—and it is often possible to deduce extra information which is not printed on it by understanding the issues and conventions involved in preparing it.

Traditional precision analog integrated circuits (which include amplifiers, converters and other devices) were designed for operation from supplies of ±15 V, and many (but not all—it is important to check with the data sheet) would operate within specification over quite a wide range of supply voltages. Today the processes used for many, but by no means all, modern converters have low breakdown voltages and absolute maximum ratings of only a few volts. Converters built with these processes may only work to specification over a narrow range of supply voltages.

It is therefore important when selecting a data converter to check both the absolute maximum supply voltage(s) and the range of voltages where correct operation can be expected. Some low-voltage devices work equally well with both 5-V and 3.3-V supplies, others are sold in 5-V and 3.3-V versions with different suffixes on their part numbers—with these it is important to use the correct one.

Absolute maximum ratings are ratings which can never be exceeded without grave risk of damage to the device concerned—they are not safe operating limits. But they are conservative—integrated circuit manufacturers try to set absolute maximum ratings so that every device that they manufacture will survive brief exposures to absolute maximum conditions. As a result many devices will, in fact, appear to operate safely and continuously outside the permitted limits. Good engineers do not take advantage of this for three reasons: (1) components are not tested outside their absolute maximum limits so, although they may be operating, they may not be operating at their specified accuracy. Also the damage done by incorrect operation may not be immediately fatal, but may cause low levels of disruption which, in turn, may (2) shorten the device's life, or (3) may affect its subsequent accuracy even when it is operated within specification again. None of these effects is at all desirable and absolute maximum ratings should always be respected.

The supply current in a data converter specification is usually the no-load current – i.e. the current consumption when the data converter output is driving a high impedance or open-circuit load. CMOS logic, and to a lesser extent some other types, have current consumption which is proportional to clock speed so a CMOS data converter current may be defined at a specific clock frequency and will be higher if the clock runs faster. Current consumption will also be higher when the output (or the reference output if there is one, or both) is loaded. There may be another figure for "standby" current—the current which flows when the data converter is connected to a power supply but is internally shut down into a non-operational low power state to conserve current.

When power is first applied to some data converters they may take several tens, hundreds or even thousands of microseconds for their reference and amplifier circuitry to stabilize and, although this is less common, some may even take a long time to "wake up" from a power saving standby mode. It is therefore important to ensure that data converters which have such delays are not used in applications where full functionality is required within a short time of power-up or wake-up.

All integrated circuits are vulnerable to electrostatic discharge (ESD), but precision analog circuits are, on the whole, more vulnerable than some other types. This is because the technologies available for minimizing such damage also tend to degrade the performance of precision circuitry, and there is a necessary compromise between robustness and performance. It is always a good idea to ensure that when you handle amplifiers, converters and other vulnerable circuits you take the necessary steps to avoid ESD.

Specifications of packages, operating temperature ranges and similar issues, although important, do not need further discussion here.

Logic Interface Issues

As it is important to read and understand power supply specifications so it is equally important to read and understand logic specifications. In the past most integrated circuit logic circuitry (with the exception of emitter-coupled logic or ECL) operated from 5-V supplies and had compatible logic levels—with a few exceptions 5-V logic would interface with other 5-V logic. Today, with the advent of low voltage logic operating with supplies of 3.3 V, 2.7 V, or even less, it is important to ensure that logic interfaces are compatible. There are several issues which must be considered—absolute maximum ratings, worst case logic levels, and timing. The logic inputs of integrated circuits generally have absolute maximum ratings, as do most other inputs, of 300 mV outside the power supply. Note that these are instantaneous ratings. If an IC has such a rating and is currently operating from a +5-V supply then the logic inputs may be between –0.3 V and +5.3 V—but if the supply is not present then that input must be between +0.3 V and –0.3 V, not the –0.3 V to +5.3 V which are the limits once the power is applied—ICs cannot predict the future.

The reason for the rating of 0.3 V is to ensure that no parasitic diode on the IC is ever turned on by a voltage outside the IC’s absolute maximum rating. It is quite common to protect an input from such over-voltage with a Schottky diode clamp. At low temperatures the clamp voltage of a Schottky diode may be a little more than 0.3 V, and

so the IC may see voltages just outside its absolute maximum rating. Although, strictly speaking, this subjects the IC to stresses outside its absolute maximum ratings and so is forbidden, this is an acceptable exception to the general rule provided the Schottky diode is at a similar temperature to the IC that it is protecting (say within ±10°C).

Some low voltage devices, however, have inputs with absolute maximum ratings which are substantially greater than their supply voltage. This allows such circuits to be driven by higher voltage logic without additional interface or clamp circuitry. But it is important to read the data sheets and ensure that both logic levels and absolute maximum voltages are compatible for all combinations of high and low supplies.

This is the general rule when interfacing different low-voltage logic circuitry—it is always necessary to check that at the lowest value of its power supply (a) the logic 1 output from the driving circuit applied to its worst-case load is greater than the specified minimum logic 1 input for the receiving circuit, and (b) with its output sinking maximum allowed current, the logic 0 output is less than the specified logic 0 input of the receiver. If the logic specifications of your chosen devices do not meet these criteria it will be necessary to select different devices, use different power supplies, or use additional interface circuitry to ensure that the required levels are available. Note that additional interface circuitry introduces extra delays in timing.

It is not sufficient to build an experimental set-up and test it. In general logic thresholds are generously specified and usually logic circuits will work correctly well outside their specified limits—but it is not possible to rely on this in a production design. At some point a batch of devices near the limit on low output swing will be required to drive some devices needing slightly more drive than usual—and will be unable to do so.

Data Converter Logic: Timing and other Issues

It is not the purpose of this brief section to discuss logic architectures, so we shall not define the many different data converter logic interface operations and their timing specifications except to note that data converter logic interfaces may be more complex than you expect—*read the data sheet*—and do not expect that because there is a pin with the same name on memory and interface chips it will behave in exactly the same way in a data converter. Also some data converters reset to a known state on power-up but many more do not.

But it is very necessary to consider general timing issues. The new low voltage processes which are used for many modern data converters have a number of desirable features. One which is often overlooked by users (but not by converter designers!) is their higher logic speed. DACs built on older processes frequently had logic that was orders of magnitude slower than the microprocessors that they interfaced with, and it was sometimes necessary to use separate buffers, or multiple WAIT instructions, to make the two compatible. Today it is much more common for the write times of DACs to be compatible with those of the fast logic with which they interface.

Nevertheless not all DACs are speed compatible with all logic interfaces, and it is still important to ensure that minimum data set-up times and write pulse widths are observed. Again, experiments will often show that devices work with faster signals than their

specification requires—but at the limits of temperature or supply voltage some may not, and interfaces should be designed on the basis of specified rather than measured timing.

SECTION 2.5: DEFINING THE SPECIFICATIONS

Dan Sheingold, Walt Kester

The following list, in alphabetical order, should prove helpful regarding specifications and their definitions. Some of the most popular ones are discussed in other places in the text as well as here. Many of the application-specific specifications are defined where they are mentioned in the text and are not repeated here. The original source for these definitions was provided by Dan Sheingold from Chapter 11 in his classic book—*Analog-to-Digital Conversion Handbook, Third Edition*, Prentice-Hall, 1986.

Accuracy, Absolute. Absolute accuracy error of a *DAC* is the difference between actual analog output and the output that is expected when a given digital code is applied to the converter. Error is usually commensurate with resolution, i.e., less 1/2 LSB of full-scale, for example. However, accuracy may be much better than resolution in some applications; for example, a 4-bit DAC having only 16 discrete digitally chosen levels would have a resolution of 1/16, but it might have an accuracy to within 0.01 % of each ideal value.

Absolute accuracy error of an *ADC* at a given output code is the difference between the actual and the theoretical analog input voltages required to produce that code. Since the code can be produced by any analog voltage in a finite band (see *Quantizing Uncertainty*), the "input required to produce that code" is usually defined as the midpoint of the band of inputs that will produce that code. For example, if 5 volts, ±1.2 mV, will theoretically produce a 12-bit half-scale code of 1000 0000 0000, then a converter for which any voltage from 4.997 V to 4.999 V will produce that code will have absolute error of (1/2)(4.997 + 4.999) – 5 volts = +2 mV.

Sources of error include gain (calibration) error, zero error, linearity errors, and noise. Absolute accuracy measurements should be made under a set of standard conditions with sources and meters traceable to an internationally accepted standard.

Accuracy, Logarithmic DACs. The difference (measured in dB) between the actual transfer function and the ideal transfer function, as measured after calibration of gain error at 0 dB.

Accuracy, Relative. Relative accuracy error, expressed in %, ppm, or fractions of 1 LSB, is the deviation of the analog value at any code (relative to the full analog range of the device transfer characteristic) from its theoretical value (relative to the same range), after the full-scale range (FSR) has been calibrated (see *Full-Scale Range*).

Since the discrete analog values that correspond to the digital values ideally lie on a straight line, the specified worst case relative accuracy error of a linear ADC or DAC can be interpreted as a measure of end-point nonlinearity (see *Linearity).*

The "discrete points" of a DAC transfer characteristic are measured by the actual analog outputs. The "discrete points" of an ADC transfer characteristic are the midpoints of the quantization bands at each code (see *Accuracy, Absolute).*

Acquisition Time. The acquisition time of a track-and-hold circuit for a step change is the time required by the output to reach its final value, within a specified error band, after the track command has been given. Included are switch delay time, the slewing interval, and settling time for a specified output voltage change.

Adjacent Channel Power Ratio (ACPR). The ratio in dBc between the measured power within a channel relative to its adjacent channel. See *Adjacent Channel Leakage Ratio* (ACLR).

Adjacent Channel Leakage Ratio (ACLR). The ratio in dBc between the measured power within the carrier bandwidth relative to the noise level in an adjacent empty carrier channel. Both ACPR and ACLR are Wideband-CDMA (WCDMA) specifications. The channel bandwidth for WCDMA is approximately 3.84 MHz with 5-MHz spacing between channels.

Aliasing. A signal within a *bandwidth* f_a must be sampled at a rate $f_s > 2f_a$ in order to avoid the loss of information. If $f_s < 2f_a$, a phenomenon called aliasing, inherent in the spectrum of the sampled signal, will cause a frequency equal to $f_s - f_a$, called an alias, to appear in the Nyquist bandwidth, dc to $f_s/2$. For example, if $f_s = 4$ kSPS and $f_a = 3$ kHz, a 1-kHz alias will appear. Note also that for $f_a = 1$ kHz (within the dc to $f_s/2$ bandwidth), an alias will occur at 3 kHz (outside the dc to $f_s/2$ bandwidth). Since noise is also aliased, it is essential to provide low pass (or band pass) filtering prior to the sampling stage to prevent out-of-band noise on the input signal from being aliased into the signal range and thereby degrading the SNR.

Analog Bandwidth. For an ADC, the analog input frequency at which the spectral power of the fundamental frequency (as determined by the FFT analysis) is reduced by 3 dB. This can be specified as full-power bandwidth, or small signal bandwidth. See also (*Bandwidth, Full Linear* and *Bandwidth, Full Power*).

Analog Bandwidth, 0.1dB. For an ADC, the analog input frequency at which the spectral power of the fundamental frequency (as determined by the FFT analysis) is reduced by 0.1 dB. This is a popular video specification. See also (*Bandwidth, Full Linear* and *Bandwidth, Full Power*).

Aperture Time. (classic definition) Aperture time in a sample-and-hold is defined as the time required for the internal switch to switch from the closed position (zero resistance) to the fully open position (infinite resistance). A first order analysis which neglects non-linear effects assumes that the input signal is averaged over this time interval to produce the final output signal. The analysis shows that this does not introduce an error as long as the switch opens in a repeatable fashion, and as long as the aperture time is reasonably short with respect to the hold time. There exists an effective sampling point in time which will cause an ideal sample-and-hold to produce the same held voltage. The difference between this effective sampling point and the actual sampling point is defined as effective aperture delay time.

Aperture Delay Time, or *Effective Aperture Delay Time.* In a sample-and-hold or track-and-hold, there exists an effective sampling point in time which will cause an ideal

sample-and-hold to produce the same held voltage. The difference between this effective sampling point and the actual sampling point is defined as the aperture delay time or effective aperture delay time. In a sampling ADC, aperture delay time can be measured by sampling the zero-crossing of a sinewave with a sampling clock locked to the sinewave. The phase of the sampling clock sampling clock is adjusted until the output of the ADC is 100…00. The time difference between the leading edge of the sampling clock and the zero-crossing of the sinewave—referenced to the analog input—is the effective aperture delay time. A dual trace oscilloscope can be used to make the measurement.

*Aperture Uncertainty (*or *Aperture Jitter)* is the sample-to-sample variation in the sampling point because of jitter. Aperture jitter is expressed as an rms quantity and produces a corresponding rms voltage error in the sample-and-hold output. In an ADC it is caused by internal noise and jitter in the sampling clock path from the sampling clock input pin to the internal switch. Jitter in the external sampling clock produces the same type of error.

Automatic Zero. To achieve zero stability in many integrating-type converters, a time interval is provided during each conversion cycle to allow the circuitry to compensate for drift errors. The drift error in such converters is substantially zero. A similar function exists in many high-resolution sigma-delta ADCs.

Bandwidth, Full-Linear. The full-linear bandwidth of an ADC is the input frequency at which the slew-rate limit of the sample-and-hold amplifier is reached. Up to this point, the amplitude of the reconstructed fundamental signal will have been attenuated by less than 0.1 dB. Beyond this frequency, distortion of the sampled input signal increases significantly.

Bandwidth, Full-Power (FPBW). The full-power bandwidth is that input frequency at which the amplitude of the reconstructed fundamental signal (measured using FFTs) is reduced by 3 dB for a full-scale input. In order to be meaningful, the FPBW must be examined in conjunction with the signal-to-noise ratio (SNR), signal-to-noise-plus-distortion ratio (SINAD), effective number of bits (ENOB), and harmonic distortion in order to ascertain the true dynamic performance of the ADC at the FPBW frequency.

Bandwidth, Analog Input Small-Signal. Analog input bandwidth is measured similarly to FPBW at a reduced analog input amplitude. This specification is similar to the small-signal bandwidth of an op amp. The amplitude of the input signal at which the small-signal bandwidth is measured should be specified on the data sheet.

Bandwidth, Effective Resolution (ERB). Some ADC manufacturers define the frequency at which SINAD drops 3 dB as the *effective resolution bandwidth (ERB)*. This is the same frequency at which the ENOB drops ½ bit. This specification is a misnomer, however, since bandwidth normally is associated with signal amplitude.

Bias Current is the zero-signal dc current required from the signal source by the inputs of many semiconductor circuits. The voltage developed across the source resistance by bias current constitutes an (often negligible) offset error. When an instrumentation amplifier performs measurements of a source that is remote from the amplifier's power-supply, there *must* be a return path for bias currents. If it does not already exist and is not

provided, those currents will charge stray capacitances, causing the output to drift uncontrollably or to saturate. Therefore, when amplifying outputs of "floating" sources, such as transformers, insulated thermocouples, and ac-coupled circuits, there must be a high-impedance dc leakage path from each input to common, or to the driven-guard terminal (if present). If a dc return path is impracticable, an *isolator* must be used.

*Bipolar Mode. (*See *Offset).*

Bipolar Offset. (See *Offset).*

Bus. A bus is a parallel path of binary information signals—usually 4, 8, 16, 32, or 64-bits wide. Three common types of information usually found on buses are data, addresses, and control signals. three-state output switches (inactive, high, and low) permit many sources—such as ADCs—to be connected to a bus, while only one is active at any time.

Byte. A byte is a binary digital word, usually 8-bits wide. A byte is often part of a longer word that must be placed on an 8-bit bus in two stages. The byte containing the MSB is called the *high byte*; that containing the LSB is called the *low byte*. A 4-bit byte is called a *nibble* on an 8-bit or greater bus.

Channel-to-Channel Isolation. In multiple DACs, the proportion of analog input signal from one DAC's reference input that appears at the output of the other DAC, expressed logarithmically in dB. See also *crosstalk.*

*Charge Transfer, Charge Injection (*or *Offset Step),* the principal component of *sample-to-hold offset* (or *pedestal),* is the small charge transferred to the storage capacitor via interelectrode capacitance of the switch and stray capacitance when switching to the *hold* mode. The offset step is directly proportional to this charge:

$$\text{Offset error} = \text{Incremental Charge/Capacitance} = \Delta Q/C.$$

It can be reduced somewhat by lightly coupling an appropriate polarity version of the *hold* signal to the capacitor for first-order cancellation. The error can also be reduced by increasing the capacitance, but this increases *acquisition time.*

Code Width. This is a fundamental quantity for ADC specifications. In an ADC where the code transition noise is a fraction of an LSB, it is defined as the range of analog input values for which a given digital output code will occur. The nominal value of a code width (for all but the first and last codes) is the voltage equivalent of 1 least significant bit (LSB) of the full-scale range, or 2.44 mV out of 10 volts for a 12-bit ADC. Because the full-scale range is fixed, the presence of excessively wide codes implies the existence of narrow and perhaps even missing codes. Code transition noise can make the measurement of code width difficult or impossible. In wide bandwidth and high resolution ADCs additional noise modulates the effective code width and appears as input-referred noise. Many ADCs have input-referred noise which spans several code widths, and histogram techniques must be used to accurately measure differential linearity.

Common-Mode Range. Common-mode rejection usually varies with the magnitude of the range through which the input signal can swing, determined by the sum of the common-mode and the differential voltage. *Common-mode range* is that range of *total* input voltage over which specified common-mode rejection is maintained. For example, if the common-mode signal is ±5 V and the differential signal is ±5 V, the common-mode range is ±10 V.

Common-Mode Rejection (CMR) is a measure of the change in output voltage when both inputs are changed by equal amounts of ac and/or dc voltage. Common-mode rejection is usually expressed either as a ratio (e.g., CMRR = 1,000,000:1) or in decibels: CMR = $20\log_{10}$CMRR; if CMRR = 10^6, CMR = 120 dB. A CMRR of 10^6 means that 1 volt of common mode is processed by the device as though it were a differential signal of 1-µV at the input.

CMR is usually specified for a full-range common-mode voltage change (CMV), at a given frequency, and a specified imbalance of source impedance (e.g., 1 kΩ source unbalance, at 60 Hz). In amplifiers, the common-mode rejection ratio is defined as the ratio of the signal gain, G, to the common-mode gain (the ratio of common-mode signal appearing at the output to the CMV at the input.

Common-Mode Voltage (CMV). A voltage that appears in common at both input terminals of a differential-input device, with respect to its output reference (usually "ground"). For inputs, V_1 and V_2, with respect to ground, CMV = ½(V_1 + V_2). An ideal differential-input device would ignore CMV. *Common-mode error (CME)* is any error at the output due to the common-mode input voltage. The errors due to supply-voltage variation, an internal common-mode effect, are specified separately.

Compliance-Voltage Range. For a current-output DAC, the maximum range of (output) terminal voltage for which the device will maintain the specified current-output characteristics.

Conversion Complete. An ADC digital output signal which indicates the end of conversion. When this signal is in the opposite state, the ADC is considered to be "busy." Also called *end-of-conversion (EOC)*, *data ready*, or *status* in some converters.

Conversion Time and *Conversion Rate.* For an ADC without a sample-and-hold, the time required for a complete measurement is called *conversion time.* For most converters (assuming no significant additional systemic delays), conversion time is essentially identical with the inverse of *conversion rate.* For simple sampling ADCs, however, the conversion rate is the inverse of the conversion time plus the sample-and-hold's acquisition time. However, in many high-speed converters, because of pipelining, new conversions are initiated before the results of prior conversions have been determined; thus, there can one, two, three, or more clock cycles of conversion delay (plus a fixed delay in some cases). Once a train of conversions has been initiated, as in signal-processing applications, the conversion rate can therefore be much faster than the conversion time would imply.

Crosstalk. Leakage of signals, usually via capacitance between circuits or channels of a multi-channel system or device, such as a multiplexer, multiple input ADC, or multiple

DAC. Crosstalk is usually determined by the impedance parameters of the physical circuit, and actual values are frequency-dependent. See also *channel-to-channel isolation.* Multiple DACs have a *digital crosstalk* specification: the spike (sometimes called a glitch) impulse appearing at the output of one converter due to a change in the digital input code of another of the converters. It is specified in nanovolt- or picovolt-seconds and measured at $V_{REF} = 0$ V.

Data Ready. *(*See *Conversion Complete).*

*Deglitcher (*See *Glitch).* A device that removes or reduces the effects of time-skew in D/A conversion. A deglitcher normally employs a track-and-hold circuit, often specifically designed as part of the DAC. When the DAC is updated, the deglitcher holds the output of the DAC's output amplifier constant at the previous value until the switches reach equilibrium, then acquires and tracks the new value.

DAC Glitch. A glitch is a switching transient appearing in the output during a code transition. The worst-case DAC glitch generally occurs when the DAC is switched between the 011…111 and 100…000 codes. The net area under the glitch is referred to as *glitch impulse area* and is measured in millivolt-nanoseconds, nanovolt-seconds, or picovolt-seconds. Sometimes the term *glitch energy* is used to describe the net area under the glitch—this terminology is incorrect because the unit of measurement is not energy.

Differential Analog Input Resistance, Differential Analog Input Capacitance, and *Differential Analog Input Impedance.* The real and complex impedances measured at each analog input port of an ADC. The resistance is measured statically and the capacitance and differential input impedances are measured with a network analyzer.

Differential Analog Input Voltage Range. The peak-to-peak differential voltage that must be applied to the converter to generate a full-scale response. Peak differential voltage is computed by observing the voltage on a single pin and subtracting the voltage from the other pin, which is 180 degrees out of phase. Peak-to-peak differential is computed by rotating the inputs phase 180 degrees and taking the peak measurement again. Then the difference is computed between both peak measurements.

Differential Gain (ΔG). A video specification which measures the variation in the amplitude (in percent) of a small amplitude color subcarrier signal as it is swept across the video range from black to white.

Differential Phase (Δφ). A video specification which measures the phase variation (in degrees) of a small amplitude color subcarrier signal as it is swept across the video range from black to white.

Droop Rate. When a sample-and-hold circuit using a capacitor for storage is in *hold,* it will not hold the information forever. Droop rate is the rate at which the output voltage changes (by increasing or decreasing), and hence gives up information. The change of output occurs as a result of leakage or bias currents flowing through the storage capacitor. The polarity of change depends on the sources of leakage within a given device. In integrated circuits with external capacitors, it is usually specified as a *(droop* or *drift)*

current, in ICs having internal capacitors, a rate of change. Note: dv/dt (volts/second) = I/C (picoamperes/picofarads).

Dual-Slope Converter. An integrating ADC in which the unknown signal is converted to a proportional time interval, which is then measured digitally. This is done by integrating the unknown for a predetermined length of time. Then a reference input is switched to the integrator, which integrates "down" from the level determined by the unknown until the starting level is reached. The time for the second integration process, as determined by the counter, is proportional to the average of the unknown signal level over the predetermined integrating period. The counter provides the digital readout.

Effective Input Noise. *(*See *Input-Referred Noise).*

Effective Number of Bits (ENOB). With a sinewave input, Signal-to-Noise-and-Distortion (SINAD) can be expressed in terms of the number of bits. Rewriting the theoretical SNR formula for an ideal N-bit ADC and solving for N:

$$N = (SNR - 1.76\ dB)/6.02$$

The actual ADC SINAD is measured using FFT techniques, and ENOB is calculated from:

$$ENOB = (SINAD - 1.76\ dB)/6.02$$

Effective Resolution. *(*See *Noise-Free Code Resolution)*

*Encode Command. (*See *Encode, Sampling Clock).*

Encode (Sampling Clock) Pulsewidth/Duty Cycle. Pulsewidth high is the minimum amount of time that the ENCODE pulse should be left in Logic 1 state to achieve rated performance; pulsewidth low is the minimum time ENCODE or pulse should be left in low state. See timing implications of changing the width in the text of high speed ADC data sheets. At a given clock rate, these specs define an acceptable ENCODE duty cycle.

Feedthrough. Undesirable signal-coupling around switches or other devices that are supposed to be turned off or provide isolation, *e.g., feedthrough error* in a sample-and-hold, multiplexer, or multiplying DAC. Feedthrough is variously specified in percent, dB, parts per million, fractions of 1 LSB, or fractions of 1 volt, with a given set of inputs, at a specified frequency.

In a multiplying *DAC, feedthrough* error is caused by capacitive coupling from an ac V_{REF} to the output, with all switches off. In a *sample-and-hold, feedthrough* is the fraction of the input signal variation or ac input waveform that appears at the output in *hold.* It is caused by stray capacitive coupling from the input to the storage capacitor, principally across the open switch.

Flash Converter. A converter in which all the bit choices are made at the same time. It requires $2^N - 1$ voltage-divider taps and comparators and a comparable amount of priority encoding logic. A scheme that gives extremely fast conversion, it requires large numbers

of nearly identical components, hence it is well suited to integrated-circuit form for resolutions up to 8 bits. Several flash converters are often used in multistage *sub-ranging converters,* to provide high resolution at somewhat slower speed than pure flash conversion.

Four-Quadrant. In a multiplying DAC, "four quadrant" refers to the fact that both the reference signal and the number represented by the digital input may be of either positive or negative polarity. Such a DAC can be thought of as a gain control for ac signals ("reference" input) with a range of positive and negative digitally controlled gains. A four-quadrant multiplier is expected to obey the rules of multiplication for algebraic sign.

Frequency-to-Voltage Conversion (FVC). The input of a FVC device is an ac waveform—usually a train of pulses (in the context of conversion); the output is an analog voltage, proportional to the number of pulses occurring in a given time. FVC is usually performed by a voltage-to-frequency converter in a feedback loop. Important specifications, in addition to the accuracy specs typical of VFCs (see *Voltage-to-Frequency conversion),* include *output ripple* (for specified input frequencies), *threshold* (for recognition that another cycle has been initiated, and for versatility in interfacing several types of sensors directly), *hysteresis,* to provide a degree of insensitivity to noise superimposed on a slowly varying input waveform, and *dynamic response* (important in motor control).

Full-Scale Input Power (ADC). Expressed in dBm (power level referenced to 1 mW). Computed using the following equation, where $V_{\text{Full Scale rms}}$ is in volts, and Z_{input} is in Ω.

$$\text{Power}_{\text{Full Scale}} = 10\log_{10}\left[\frac{\dfrac{V^2_{\text{Full Scale rms}}}{|Z|_{\text{Input}}}}{0.001}\right].$$

Full-Scale Range (FSR). For binary ADCs and DACs, that magnitude of voltage, current, or—in a multiplying DAC—gain, of which the MSB is specified to be exactly one-half or for which any bit or combination of bits is tested against its (their) prescribed ideal ratio(s). FSR is independent of resolution; the value of the LSB (voltage, current, or gain) is 2^{-N} FSR. There are several other terms, with differing meanings, that are often used in the context of discussions or operations involving full-scale range. They are:

Full-scale—similar to full-scale range, but pertaining to a single polarity. Thus, full-scale for a unipolar device is twice the prescribed value of the MSB and has the same polarity. For a bipolar device, *positive or negative full-scale* is that positive or negative value, of which the next bit after the polarity bit is tested to be one-half.

Span—the scalar voltage or current range corresponding to FSR.

All-1's—All bits on, the condition used, in conjunction with *all-zeros,* for gain adjustment of an ADC or DAC, in accordance with the manufacturer's instructions. Its magnitude, for a binary device, is $(1 - 2^{-N})$ FSR. *All-1's* is a *positive-true* definition of a

specific magnitude relationship; for complementary coding the "all-l's" code will actually be all zeros. To avoid confusion, all-l's should never be called "full-scale;" FSR and FS are independent of the number of bits, all-l's isn't.

All-0's—All bits off, the condition used in offset (and gain) adjustment of a DAC or ADC, according to the manufacturer's instructions. All-0's corresponds to zero output in a unipolar DAC and negative full-scale in an offset bipolar DAC with positive output reference. In a sign-magnitude device, all-0's refers to all bits after the sign bit. Analogous to "all-l's," "all-0's" is a *positive-true* definition of the *all-bits-off* condition; in a complementary-coded device, it is expressed by all ones. To avoid confusion, all-0's should not be called "zero" unless it accurately corresponds to true analog zero output from a DAC.

Gain. The "gain" of a converter is that analog scale factor setting that establishes the nominal conversion relationship, e.g., 10 volts full-scale. In a multiplying DAC or ratiometric ADC, it is indeed a gain. In a device with fixed internal reference, it is expressed as the full-scale magnitude of the output parameter (e.g., 10 V or 2 mA). In a fixed-reference converter, where the use of the internal reference is optional, the converter gain and the reference may be specified separately. Gain and zero adjustment are discussed under *zero.*

Glitch. Transients associated with code changes generally stem from several sources. Some are spikes, known as digital-to-analog feedthrough, or charge transfer, coupled from the digital signal (clock or data) to the analog output, defined with zero reference. These spikes are generally fast, fairly uniform, code-independent, and hence filterable. However, there is a more insidious form of transient, code-dependent, and difficult to filter, known as
the "glitch."

If the output of a counter is applied to the input of a DAC to develop a "staircase" voltage, the number of bits involved in a code change between two adjacent codes establish "major" and "minor" transitions. The most major transition is at ½-scale, when the DAC switches all bits, i.e., from 011. . .111 to 100. . .000. If, for digital inputs having no skew, the switches are faster to switch *off* than *on,* this means that, for a short time, the DAC will seek zero output, and then return to the required 1 LSB above the previous reading. This large transient spike is commonly known as a "glitch." The better matched the input transitions and the switching times, the faster the switches, the smaller will be the area of the glitch. Because the size of the glitch is not proportional to the signal change, linear filtering may be unsuccessful and may, in fact, make matters worse. *(*See also *Deglitcher.)*

The severity of a glitch is specified by *glitch impulse area,* the product of its duration and its average magnitude, i.e., the net area under the curve. This product will be recognized as the physical quantity, *impulse* (electromotive *force* × Δ*time);* however, it has also been incorrectly termed "glitch energy" and "glitch charge." Glitch impulse area is usually expressed, for fast converters, in units of pV-s or mV-ns.

The glitch can be minimized through the use of fast, non-saturating logic, such as ECL, LVDS, matched latches, and non-saturating CMOS switches.

*Glitch Charge, Glitch Energy, Glitch Impulse, Glitch Impulse Area. (*See *Glitch).*

*Harmonic Distortion, 2nd.*The ratio of the rms signal amplitude to the rms value of the second harmonic component, reported in dBc.

Harmonic Distortion, 3rd. The ratio of the rms signal amplitude to the rms value of the third harmonic component, reported in dBc.

Harmonic Distortion, Total (THD). The ratio of the rms signal amplitude to the rms sum of all harmonics (neglecting noise components). In most cases, only the first five harmonics are included in the measurement because the rest have negligible contribution to the result. The THD can be derived from the FFT of the ADC's output spectrum. For harmonics that are above the Nyquist frequency, the aliased component is used.

Harmonic Distortion, Total, Plus Noise (THD + N). Total harmonic distortion plus noise (THD + N) is the ratio of the rms signal amplitude to the rms sum of all harmonics and noise components. THD + N can be derived from the FFT of the ADC's output spectrum and is a popular specification for audio applications.

Impedance, Input. The dynamic load of an ADC presented to its input source. In unbuffered CMOS switched-capacitor ADCs, the presence of current transients at the converter's clock frequency mandates that the converter be driven from a low impedance (at the frequencies contained in the transients) in order to accurately convert. For buffered-input ADCs, the input impedance is generally represented by a resistive and capacitive component.

Input-Referred Noise (Effective Input Noise). Input-referred noise can be viewed as the net effect of all internal ADC noise sources referred to the input. It is generally expressed in *LSBs rms*, but can also be expressed as a voltage. It can be converted to a peak-to-peak value by multiplying by the factor 6.6. The peak-to-peak input-referred noise can then be used to calculate the *noise-free code resolution*. (See *noise-free code resolution*).

Intermodulation Distortion (IMD). With inputs consisting of sinewaves at two frequencies, f_1 and f_2, any device with nonlinearities will create distortion products of order (m + n), at sum and difference frequencies of $mf_1 \pm nf_2$, where m, n = 0, 1, 2, 3, …. Intermodulation terms are those for which m or n is not equal to zero. For example, the second-order terms are $(f_1 + f_2)$ and $(f_2 – f_1)$, and the third-order terms are $(2f_1 + f_2)$, $(2f_1 – f_2)$, $(f_1 + 2f_2)$, and $(f_1 – 2f_2)$. The IMD products are expressed as the dB ratio of the rms sum of the distortion terms to the rms sum of the measured input signals.

*Latency. (*See *pipelining).*

Leakage Current, Output. Current which appears at the output terminal of a DAC with all bits "off." For a converter with two complementary outputs (for example, many fast CMOS DACs), output leakage current is the current measured at OUT 1, with all digital inputs *low—and* the current measured at OUT 2, with all digital inputs *high.*

Least-Significant Bit (LSB). In a system in which a numerical magnitude is represented by a series of binary (i.e. two-valued) digits, the *least-significant bit* is that digit (or "bit") that carries the smallest value, or weight. For example, in the natural binary number 1101 (decimal 13, or $(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$), the rightmost digit is the LSB. Its analog weight, in relation to full-scale (see *Full-Scale Range),* is 2^{-N}, where N is the number of binary digits. It represents the smallest analog change that can be resolved by an n-bit converter.

In data converter nomenclature, the LSB is bit N; in bus nomenclature (integer binary), it is Data Bit 0.

Left-Justified Data. When a 12-bit word is placed on an 8-bit bus in two bytes, the high byte contains the 4- or 8- most-significant bits. If 8, the word is said to be left justified; if 4 (plus filled-in leading sign bits), the word is said to be right justified.

*Linearity. (*See also *Nonlinearity.)* Linearity error of a converter *(*also, *integral nonlinearity—*see *Linearity, Differential),* expressed in % or parts per million of full-scale range, or (sub)multiples of 1 LSB, is a deviation of the analog values, in a plot of the measured conversion relationship, from a straight line. The straight line can be either a "best straight line," determined empirically by manipulation of the gain and/or offset to equalize maximum positive and negative deviations of the actual transfer characteristic from this straight line; or, it can be a straight line passing through the end points of the transfer characteristic after they have been calibrated, sometimes referred to as "end-point" linearity. "End-point" nonlinearity is similar to relative accuracy error *(*see *Accuracy, Relative).* It provides an easier method for users to calibrate a device, and it is a more conservative way to specify linearity.

For multiplying DACs, the *analog* linearity error, at a specified analog gain (digital code), is defined in the same way as for analog multipliers, i.e., by deviation from a "best straight line" through the plot of the analog output-input response.

Linearity, Differential. In a DAC, any two adjacent digital codes should result in measured output values that are exactly 1 LSB apart (2^{-N} of full-scale for an N-bit converter). Any positive or negative deviation of the measured "step" from the ideal difference is called *differential nonlinearity,* expressed in (sub)multiples of 1 LSB. It is an important specification, because a differential linearity error more negative than –1 LSB can lead to non-monotonic response in a DAC and missed codes in an ADC using that DAC.

Similarly, in an ADC, midpoints between code transitions should be 1 LSB apart. Differential nonlinearity is the deviation between the actual difference between midpoints and 1 LSB, for adjacent codes. If this deviation is equal to or more negative than –1 LSB, a code will be missed (see *missing codes*).

Often, instead of a maximum differential nonlinearity specification, there will be a simple specification of "monotonicity" or "no missing codes", which implies that the differential nonlinearity cannot be more negative than –1 for any adjacent pair of codes. However, the differential linearity error may still be more positive than +1 LSB.

*Linearity, Integral. (*See *Linearity).* While *differential linearity* deals with errors in step size, *integral linearity* has to do with deviations of the overall shape of the conversion response. Even converters that are not subject to differential linearity errors.(e.g., integrating types) have integral linearity (sometimes just "linearity") errors.

Maximum Conversion Rate. The maximum sampling (encode) rate at which parametric testing is performed.

Minimum Conversion (Sampling) Rate. The encode rate at which the SNR of the lowest analog signal frequency drops by no more than 3 dB below the guaranteed limit.

Missing Codes. An ADC is said to have missing codes when a transition from one quantum of the analog range to the adjacent one does not result in the adjacent digital code, but in a code removed by one or more counts. Missing codes can be caused by large negative differential linearity errors, noise, or changing inputs during conversion. A converter's proclivity towards missing codes is also a function of the architecture and temperature.

Monotonicity. An DAC is said to be *monotonic* if its output either increases or remains constant as the digital input increases, with the result that the output will always be a single-valued function of the input. The condition "monotonic" requires that the derivative of the transfer function never change sign. Monotonic behavior requires that the differential nonlinearity be more positive than –1 LSB. The same basic definition applies to an ADC—the digital output code either increases or remains constant as the digital input increases. In practice, however, noise will cause the ADC output code to oscillate between two code transitions over a small range of analog input. Input-referred noise can make this effect worse, so histogram techniques are often used to measure ADC monotonicity in these situations.

Most Significant Bit (MSB). In a system in which a numerical magnitude is represented by a series of binary (i.e., two-valued) digits, the *most-significant bit* is that digit (or "bit") that carries the greatest value or weight. For example, in the natural binary number 1101 (decimal 13, or $(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$), the leftmost "1" is the MSB, with a weight of ½ nominal peak-to-peak full-scale (full-scale range). In bipolar devices, the sign bit is the MSB.

In converter nomenclature, the MSB is bit 1; in bus nomenclature, it is Data Bit (N – 1).

Multiplying DAC. A multiplying DAC differs from the conventional fixed-reference DAC in being designed to operate with varying (or ac) reference signals. The output signal of such a DAC is proportional to the product of the "reference" (i.e., analog input) voltage and the fractional equivalent of the digital input number. See also *Four-Quadrant.*

Multitone Spurious Free Dynamic Range (SFDR). The ratio of the rms value of an input tone to the rms value of the peak spurious component. The peak spurious component may or may not be an intermodulation distortion (IMD) product. May be reported in dBc (dB relative to the carrieror in dBFS (dB relative to full-scale). The amplitudes of the

individual tones are equal and chosen such that the ADC is not overdriven when they add in-phase.

Noise-Free (Flicker-Free) Code Resolution. The noise-free code resolution of an ADC is the number of bits beyond which it is impossible to distinctly resolve individual codes. The cause is the effective input noise (or input-referred noise) associated with all ADCs. This noise can be expressed as an rms quantity, usually having the units of *LSBs rms*. Multiplying by a factor of 6.6 converts the rms noise into peak-to-peak noise (expressed in *LSBs peak-to-peak*). The total range of an N-bit ADC is 2^N. The noise-free (or flicker-free) resolution can be calculated using the equation:

$$\textit{Noise-Free Code Resolution} = \log_2(2^N/\text{Peak-to-Peak Noise}).$$

The specification is generally associated with high-resolution sigma-delta measurement ADCs, but is applicable to all ADCs.

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

$$\textit{Effective Resolution} = \log_2(2^N/\text{RMS Input Noise}).$$

Noise, Peak and RMS. Internally generated random noise is not a major factor in DACs, except at extreme resolutions and dynamic ranges. Random noise is characterized by rms specifications for a given bandwidth, or as a spectral density (current or voltage per root hertz); if the distribution is Gaussian, the probability of peak-to-peak values exceeding $6.6 \times$ the rms value is less than 0.l%.

Of much greater importance in DACs is interference, in the form of high-amplitude, low-energy (hence low-rms) spikes appearing at a DAC's output, caused by coupling of digital signals in a surprising variety of ways; they include coupling via stray capacitance, via power supplies, via inadequate ground systems, via feedthrough, and by glitch-generation (see *Glitch).* Their presence underscores the necessity for maximum application of the designer's art, including layout, shielding, guarding, grounding, bypassing, and deglitching.

Noise in ADCs in effect narrows the region between transitions. Sources of noise include the input sample-and-hold, resistor noise, "KT/C" noise, the reference, the analog signal itself, and pickup in infinite variety.

Noise Power Ratio (NPR). In this measurement, wideband Gaussian noise (bandwidth $< f_s/2$) is applied to an ADC through a narrowband notch filter. The notch filter removes all noise within its bandwidth. The output of the ADC is examined with a large FFT. The ratio of the rms noise level to the rms noise level inside the notch (due to quantization noise, thermal noise, and intermodulation distortion) is defined as the *noise power ratio (NPR).* The rms noise level at the input to the ADC is generally adjusted to give the best NPR value.

*No Missing Codes Resolution. (*See *Resolution, No Missing Codes).*

Nonlinearity--or "gain nonlinearity"—is defined as the deviation from a straight line on the plot of output vs. input. The magnitude of linearity error is the maximum deviation from a "best straight line," with the output swinging through its full-scale range. Nonlinearity is usually specified in percent of full-scale output range.

Normal Mode. For an amplifier used in instrumentation, the *normal-mode* signal is the actual difference signal being measured. This signal often has noise associated with it. Signal conditioning systems and digital panel instruments usually contain input filtering to remove high frequency and line frequency noise components. *Normal-mode rejection* (NMR), is a logarithmic measure of the attenuation of normal-mode noise components at specified frequencies in dB.

Offset, Bipolar. For the great majority of bipolar converters (e.g., ±10-V output), negative currents are not actually generated to correspond to negative numbers; instead, a unipolar DAC is used, and the output is offset by half full-scale (1 MSB). For best results, this offset voltage or current is derived from the same reference supply that determines the gain of the converter.

Because of nonlinearity, a device with perfectly calibrated end points may have offset error at analog zero.

*Offset Step. (*See *Pedestal).*

Output Propagation Delay. For an ADC having a single-ended sampling (or ENCODE) clock input, the delay between the 50% point of the sampling clock and the time when all output data bits are within valid logic levels. For an ADC having differential sampling clock inputs, the delay is measured with respect to the zero-crossing of the differential sampling clock signal.

Output Voltage Tolerance. For a reference, the maximum deviation from the normal output voltage at 25°C and specified input voltage, as measured by a device traceable to a recognized fundamental voltage standard.

Overload. An input voltage exceeding the ADC's full-scale input range producing an overload condition.

Overvoltage Recovery Time. Overvoltage recovery time is defined as the amount of time required for an ADC to achieve a specified accuracy after an overvoltage (usually 50% greater than full-scale range), measured from the time the overvoltage signal reenters the converter's range. The ADC should act as an ideal limiter for out-of-range signals, producing a positive or negative full-scale code during the overvoltage condition. Some ADCs provide over- and under-range flags to allow gain-adjustment circuits to be activated.

Overrange, Overvoltage. An input signal that exceeds the full-scale input range of an ADC, but is less than an overload.

Pedestal, or *Sample-to-Hold Offset Step.* In sample/track-and-hold amplifiers, a shift in level between the last value in *sample* and the value settled-to in *hold;* in devices having fixed internal capacitors, it includes *charge transfer,* or *offset step.* However, for devices that may use external capacitors, it is often defined as the residual step error after the *charge transfer* is accounted for and/or cancelled. Since it is unpredictable in magnitude and may be a function of the signal, it is also known as *offset nonlinearity.*

Pipelining. A pipelined converter is a multistage converter which is capable of accepting a new signal before it has completed the conversion of one or more previous ones. A new signal arrives while others are still "in the pipeline." This is a technique used where a fast conversion rate is desired and the latency of individual conversions is relatively unimportant.

Power-Supply Rejection Ratio (PSRR). The ratio of a change in dc power supply voltage to the resulting change in the specified device error, expressed in percentage, parts per million, or fractions of 1 LSB. It may also be expressed logarithmically, in dB, PSR = 20 $\log_{10}$ (PSRR).

Quad-Slope Converter. This is an integrating analog-to-digital converter that goes through two cycles of *dual-slope* conversion, once with zero input and once with the analog input being measured. The errors determined during the first cycle are subtracted digitally from the result in the second cycle. The scheme can result in high-accuracy conversion.

*Quantizing Uncertainty (*or*" Quantization Error").* The analog continuum is partitioned into 2^N discrete ranges for N-bit conversion and processing. All analog values within a given quantum are represented by the same digital code, usually assigned to the nominal mid-range value. There is, therefore, an inherent quantization uncertainty of ±½ LSB, in addition to the actual conversion errors. In integrating ADCs, this "error" is often expressed as " ±1 count." Depending on the system context, it may be interpreted as a truncation (round-off) error or as noise.

Ratiometric. The output of an ADC is a digital number proportional to the *ratio* of (some measure of) the input to a reference voltage. Most requirements for conversions call for an absolute measurement, i.e., against a fixed reference; but this presumes that the signal applied to the converter is either reference-independent or in some way derived from another fixed reference. However, real references are not truly fixed; the references for both the converter and the signal source vary with time, temperature, loading, etc. Therefore, if the converter is used with signal sources that also rely on references (for example, strain-gage bridges, RTDs, thermistors), it makes sense to replace this multiplicity of references by a single system reference. In this case, reference-caused errors will tend to cancel out. This can be done by using the converter's internal reference (if it has one) as the system reference. Another way is to use a separate external system reference, which also becomes the reference for a *ratiometric* converter. For instance, if a bridge is excited with the same voltage used for the ADC reference, ratiometric operation is achieved, and the ADC output code is not a function of the reference. This is because the bridge output signal is proportional to the same voltage which defines the ADC input range.

Resolution. An N-bit binary converter has N digital data inputs (DAC) or N digital data outputs (ADC). A converter that satisfies this criterion is said to have a *resolution* of N bits.

Resolution, No Missing Codes. The *no missing code resolution* of an ADC is the maximum number of bits of resolution beyond which the ADC will have missing codes. For instance, if an 18-bit ADC has a no missing code resolution of 16 bits, then there will be no missing codes if only the 16 MSBs are utilized. Codes may be missed at the 17- and 18-bit level.

The smallest output change that can be resolved by a linear DAC is 2^{-N} of the full-scale span. Thus, for example, the resolution of an 8-bit DAC would be 2^{-8}, or 1/256. On the other hand, a nonlinear device, such as the AD7111 LOGDAC™, can ideally achieve a dynamic range of 89.625 dB, or 30,000:1, in 0.375-dB steps, using only 8 bits of digital resolution.

Right-Justified Data. When a 12-bit word is placed on an 8-bit bus in two stages, the high byte contains the 4- or 8- most-significant bits. If 8, the word is said to be left justified; if 4 (plus filled-in leading sign bits), the word is said to be right justified.

*Sample-to-Hold Offset.(*See *Pedestal).*

Sampling ADC. A sampling ADC includes a sample-and-hold function which acquires the input value at a given instant and holds it throughout the conversion time (or until the converter is ready for the next sample point). Flash ADCs and sigma-delta ADCs are inherently sampling devices.

*Sampling Clock. (*See *Encode Command).*

Sampling Frequency. The rate at which an ADC converts an analog input signal into digital outputs, not to be confused with *conversion time*.

Serial Output. A bit-serial output consists of a series of bits clocked out on a single line. There must be some means of identifying the beginning and ends of words; this can be accomplished via an additional clock line, by using synchronized clocks, and/or by providing a consistent identifying signature for the beginning of a word. Byte-serial consists of a series of bytes transmitted in sequence on a bus (see *Byte*).

Settling Time—ADC. The time required, following an analog input step change (usually full-scale), for the digital output of the ADC to reach and remain within a given fraction (usually ± ½LSB).

Settling Time—DAC. The time required, following a prescribed data change, for the output of a DAC to reach and remain within an error band (usually ±½ LSB) of the final value. Typical prescribed changes are full-scale, 1 MSB, and 1 LSB at a major carry. Settling time of current-output DACs is quite fast. The major share of settling time of a voltage-output DAC is usually contributed by the settling time of the output op-amp. DAC settling time can also be defined with respect to the output. Output settling time is the time measured from the point the output signal leaves an error band referenced to the

initial output value until the time the signal enters and remains within the error band referenced to the final output value.

Signal-to-Noise-and-Distortion Ratio (SINAD). The ratio of the rms signal amplitude (set 1 dB below full-scale to prevent overdrive) to the rms value of the sum of all other spectral components, including harmonics but excluding dc.

*Signal-to-Noise Ratio (without Harmonics).*The ratio of the rms signal amplitude (set at 1 dB below full-scale to prevent overdrive) to the rms value of the sum of all other spectral components, excluding the first five harmonics and dc. Technically, all harmonics should be excluded, but in practice, only the first five are generally significant.

Single-Slope Conversion. In the single-slope converter, a reference voltage is integrated until the output of the integrator is equal to the input voltage. The time period required for the integrator to go from zero to the level of the input is proportional to the magnitude of the input voltage and is measured by an internal clock. Measurement accuracy is sensitive to clock speed and integrating capacitance, as well as the reference accuracy.

Slew(ing) Rate. A limitation in the rate of change of output voltage, usually imposed by some basic circuit consideration, such as limited current to charge a capacitor. The output slewing speed of a voltage-output DAC is usually limited by the slew rate of the amplifier used at its output.

Spurious-Free Dynamic Range (SFDR). The ratio of the rms signal amplitude to the rms value of the peak spurious spectral component. The peak spurious component may or may not be a harmonic. May be reported in dBc (i.e., degrades as signal level is lowered) or dBFS (related back to converter full-scale).

Stability. In a well-designed, intelligently applied converter, *dynamic stability* is not an important question. The term stability usually applies to the insensitivity of the converter's characteristics to time, temperature, etc. All measurements of stability are difficult and time consuming, but stability vs. temperature is sufficiently critical in most applications to warrant universal inclusion in tables of specifications (see *Temperature Coefficient).*

Staircase. A voltage or current, increasing in equal increments as a function of time and having the appearance of a staircase (in a time plot); it is generated by applying a pulse train to a counter, and the output of the counter to the input of a DAC.

Subranging ADCs. In this type of converter, a fast converter produces the most-significant portion of the output word. This portion is stored in a holding register and also converted back to analog with a fast, high-accuracy DAC. The analog result is subtracted from the input, and the resulting residue is amplified, converted to digital at high speed, and combined with the results of the earlier conversion to form the output word. In *digitally corrected subranging* (DCS) ADCs, the two conversions are combined in a manner that corrects for the error of the LSB of the most significant bits. For example, using 8-bit and 5-bit conversion, plus this technique and a great deal of video-speed converter expertise, a full-accuracy high-speed 12-bit ADC can be built. Many pipelined subranging ADCs use more than two stages with error correction between each stage.

Successive Approximation. Successive approximation is a method of conversion by comparing an unknown against a group of weighted references. The operation of a successive-approximation ADC is generally similar to the orderly weighing of an unknown quantity on a precision balance, using a set of weights, such as 1 gram ½ gram, ¼ gram, etc. The weights are tried in order, starting with the largest. Any weight that tips the scale is removed. At the end of the process, the sum of the weights remaining on the scale will be within 1 LSB of the actual weight (±½ LSB, if the scale is properly biased—see *Zero*). The successive approximation ADC is often called a SAR ADC, because the logic block which controls the conversion process is known as a successive approximation register (SAR).

Switching Time. In a DAC, the switching time is the time taken for an analog switch to change to a new state from the previous one. It includes propagation delay time, and rise time from 10% to 90%, but does not include settling time.

Temperature Coefficient. In general, temperature instabilities are expressed as %/°C, ppm/°C, fractions of 1 LSB per degree C, or as a change in a parameter over a specified temperature range. Measurements are usually made at room temperature (25°C) and at the extremes of the specified range, and the temperature coefficient (tempco, TC) is defined as the change in the parameter, divided by the corresponding temperature change. Parameters of interest include gain, linearity, offset (bipolar), and zero.

a. *Gain Tempco:* Two factors principally affect converter gain stability with temperature. In fixed-reference converters, the reference voltage will vary with temperature. The reference circuitry and switches (and comparator in ADCs) will also contribute to the overall gain TC.

b. *Linearity Tempco:* Sensitivity of linearity (integral and/or differential linearity) to temperature, in % FSR/°C or ppm FSR/°C, over the specified range. Monotonic behavior in DACs is achieved if the differential nonlinearity is less than 1 LSB at any temperature in the range of interest. The *differential nonlinearity temperature coefficient* may be expressed as a ratio, as a maximum change over a temperature range, and/or implied by a statement that the device is monotonic over the specified temperature range. To avoid missing codes in noiseless ADCs, it is sufficient that the differential nonlinearity error be less than –1 LSB at any temperature in the range of interest. The differential nonlinearity temperature coefficient is often implied by the statement that there are no missed codes when operating within a specified temperature range. In DACs, the differential nonlinearity TC is often implied by the statement that the DAC is monotonic over a specified temperature range.

c. *Zero TC (unipolar converters):* The temperature stability of a unipolar fixed-reference DAC, measured in % FSR/°C or ppm FSR/°C, is principally affected by current leakage (current-output DAC), and offset voltage and bias current of the output op amp (voltage-output DAC). The zero stability of an ADC is dependent on the zero stability of the DAC or integrator and/or the input buffer and the comparator. It is typically expressed in μV/°C or in percent or ppm of full-scale range (FSR) per degree C.

d. *Offset Tempco:* The temperature coefficient of the all-DAC-switches-off (minus full-scale) point of a bipolar converter (in % FSR/°C or ppm FSR/°C) depends on three major factors—the tempco of the reference source, the voltage zero-stability of the output amplifier, and the tracking capability of the bipolar-offset resistors and the gain resistors. In an ADC, the corresponding tempco of the negative full-scale point depends on similar quantities—the tempco of the reference source, the voltage stability of the input buffer and the sample-and-hold, and the tracking capabilities of the bipolar offset resistors and the gain-setting resistors.

Thermal Tail. The slow drift of an amplifier having a thermally induced offset due to self-heating as it settles to a final electrical equilibrium value corresponding to internal thermal equilibrium.

Total Unadjusted Error. A comprehensive specification on some devices which includes full-scale error, relative-accuracy and zero-code errors, under a specified set of conditions.

Transient Response. *(*See *settling time).*

Two Tone SFDR. The ratio of the rms value of either input tone to the rms value of the peak spurious component. The peak spurious component may or may not be an intermodulation distortion (IMD) product. May be reported in dBc (i.e., degrades as signal level is lowered) or in dBFS (always related back to converter full-scale).

Worst Other Spur. The ratio of the rms signal amplitude to the rms value of the worst spurious component (excluding the second and third harmonic) reported in dBc.

General References on Data Conversion and Related Topics

1. Alfred K. Susskind, **Notes on Analog-Digital Conversion Techniques**, John Wiley, 1957.

2. David F. Hoeschele, Jr., **Analog-to-Digital/Digital-to-Analog Conversion Techniques**, John Wiley and Sons, 1968.

3. K. W. Cattermole, **Principles of Pulse Code Modulation**, American Elsevier Publishing Company, Inc., 1969, New York NY, ISBN 444-19747-8.

4. Hermann Schmid, **Electronic Analog/Digital Conversions**, Van Nostrand Reinhold Co., 1970.

5. Dan Sheingold, **Analog-Digital Conversion Handbook**, First Edition, Analog Devices, 1972.

6. Donald B. Bruck, **Data Conversion Handbook**, Hybrid Systems Corporation, 1974.

7. Eugene R. Hnatek, **A User's Handbook of D/A and A/D Converters**, John Wiley, New York, 1976, ISBN 0-471-40109-9.

8. Nuggehally S. Jayant, **Waveform Quantizing and Coding**, John Wiley-IEEE Press, 1976, ISBN 0-87942-074-X.

9. Dan Sheingold, **Analog-Digital Conversion Notes**, Analog Devices, 1977.

10. C. F. Kurth, editor, **IEEE Transactions on Circuits and Systems Special Issue on Analog/Digital Conversion**, CAS-25, No. 7, July 1978.

11. Daniel J. Dooley, **Data Conversion Integrated Circuits**, John Wiley-IEEE Press, 1980, ISBN 0-471-08155-8.

12. Bernard M. Gordon, **The Analogic Data-Conversion Systems Digest**, Fourth Edition, Analogic Corporation, 1981.

13. Eugene L. Zuch, **Data Acquisition and Conversion Handbook**, Datel-Intersil, 1982.

14. Frank F. E. Owen, **PCM and Digital Transmission Systems**, McGraw-Hill, 1982, ISBN 0-07-047954-2.

15. Dan Sheingold, **Analog-Digital Conversion Handbook**, Analog Devices/Prentice-Hall, 1986, ISBN 0-13-032848-0.

16. Matthew Mahoney, **DSP-Based Testing of Analog and Mixed-Signal Circuits**, IEEE Computer Society Press, 1987, ISBN 0-8186-0785-8.

17. Michael J. Demler, **High-Speed Analog-to-Digital Conversion**, Academic Press, Inc., 1991, ISBN 0-12-209048-9.

18. J. C. Candy and Gabor C. Temes, **Oversampling Delta-Sigma Data Converters**, IEEE Press, ISBN 0-87942-258-8, 1992.

19. David F. Hoeschele, Jr., **Analog-to-Digital and Digital-to-Analog Conversion Techniques**, Second Edition, John Wiley and Sons, 1994, ISBN-0-471-57147-4.

20. Rudy van de Plassche, **Integrated Analog-to-Digital and Digital-to-Analog Converters**, Kluwer Academic Publishers, 1994, ISBN 0-7923-9436-4.

21. David A. Johns and Ken Martin, **Analog Integrated Circuit Design**, John Wiley, 1997, ISBN 0-471-14448-7.

22. Mikael Gustavsson, J. Jacob Wikner, and Nianxiong Nick Tan, **CMOS Data Converters for Communications**, Kluwer Academic Publishers, 2000, ISBN 0-7923-7780-X.

23. R. Jacob Baker, **CMOS Circuit Design Volumes I and II**, John Wiley-IEEE Computer Society, 2002, ISBN 0-4712-7256-6.

24. Rudy van de Plassche, **CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters, Second Edition**, Kluwer Academic Publishers, 2003, ISBN 1-4020-7500-6.

Analog Devices' Seminar Series:

25. Walt Kester, **Practical Analog Design Techniques**, Analog Devices, 1995, ISBN 0-916550-16-8, available for download at http://www.analog.com.

26. Walt Kester, **High Speed Design Techniques**, Analog Devices, 1996, ISBN 0-916550-17-6, available for download at http://www.analog.com.

27. Walt Kester, **Practical Design Techniques for Power and Thermal Management**, Analog Devices, 1998, ISBN 0-916550-19-2, available for download at http://www.analog.com.

28. Walt Kester, **Practical Design Techniques for Sensor Signal Conditioning**, Analog Devices, 1999, ISBN 0-916550-20-6, available for download at http://www.analog.com.

29. Walt Kester, **Mixed-Signal and DSP Design Techniques**, Analog Devices, 2000, ISBN 0-916550-22-2, available for download at http://www.analog.com.

30. Walt Kester, **Mixed-Signal and DSP Design Techniques**, Analog Devices and Newnes (An Imprint of Elsevier Science), ISBN 0-75067-611-6, 2003.

31. Walter G. Jung, **Op Amp Applications**, Analog Devices, 2002, ISBN 0-916550-26-5.

NOTES: