ANALOG-DIGITAL CONVERSION

ANALOG-DIGITAL CONVERSION

CHAPTER 5
TESTING DATA CONVERTERS

SECTION 5.1: TESTING DACs
Walt Kester, Dan Sheingold

Static DAC Testing

The resolution of a DAC refers to the number of unique output voltage levels that the DAC is capable of producing. For example, a DAC with a resolution of 12 bits will be capable of producing 2^{12}—or 4,096—different voltage levels at its output. Similarly, a DAC with a resolution of 16 bits can produce 2^{16}—or 65,536—levels at its output.

Inherent in the specification of resolution, especially for control applications, is the requirement for *monotonicity*. The output of a monotonic DAC always stays the same or increases for an increasing digital code. The quantitative measure of monotonicity is the specification of differential nonlinearity (step size).

The static absolute accuracy of a DAC can be described in terms of three fundamental kinds of errors: *offset errors*, *gain errors*, and *integral nonlinearity*.

Linearity errors are the most important of the three kinds, because in many applications the user can adjust out the offset and gain errors, or compensate for them without difficulty by building end-point auto-calibration into the system design, whereas linearity errors cannot be conveniently or inexpensively nulled out. But before we can understand the nature of linearity errors and how to test for them, the end-point errors must first be established.

There are many methods to measure the static errors of a DAC—the proper choice depends upon the specific objectives of the testing. For instance, an IC manufacturer generally performs production testing on DACs using specialized automatic test equipment. On the other hand, a customer evaluating various DACs for use in a system does not generally have access to sophisticated automatic test equipment and must therefore devise a suitable bench test setup. A basic DAC static test setup is shown in Figure 5.1. This flexible test setup allows the application of various digital codes to the DAC input and uses an accurate digital voltmeter for measuring the DAC output. Computer control can be used to automate the process, but it should be noted that in many cases DAC static testing can be performed by simply using mechanical switches to apply various codes to the DAC and reading the output with the voltmeter.

Today there are a large number of applications where the static performance of a DAC is rarely of direct concern to the customer—even in the evaluation phase—and ac performance is much more important. DACs used in audio and communications quite often do not even have traditional static specifications listed on the data sheet, and various noise and distortion specifications are of much more interest. However, the

traditional static specifications such as differential nonlinearity (DNL) and integral nonlinearity (INL) are most certainly reflected in the ac performance. For instance, low distortion, a key audio and communication requirement, is directly related to low INL. Large INL and DNL errors will increase both the noise and distortion level of a DAC and render it unsuitable for these demanding applications. In fact, one often finds that the static performance of these ac-specified DACs is quite good, even though it is not directly specified.

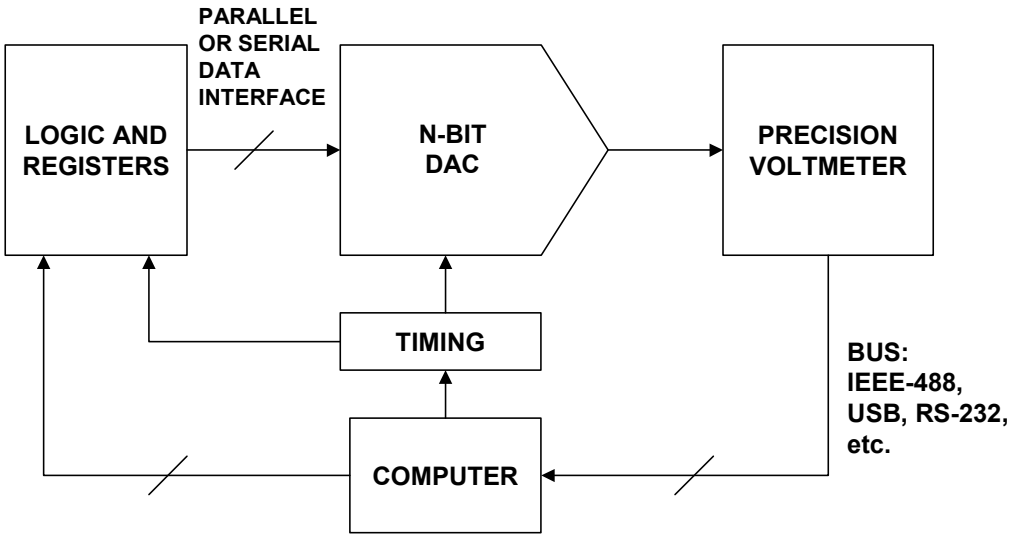

Figure 5.1: *Basic Test Setup for Measuring DAC Static Transfer Characteristics*

The following sections on static DAC testing are therefore more oriented to DACs which are used in traditional industrial control, measurement, or instrumentation applications where monotonicity, DNL, INL, gain, and offset are important.

End-Point Errors

The most commonly specified end-point errors associated with DACs are *offset error*, *gain error*, and *bipolar zero error*. Note that bipolar zero error is only associated with bipolar output DACs, whereas offset and gain error is common to both unipolar and bipolar DACs.

Figure 5.2 shows the effects of offset and gain error in a unipolar DAC. Note that in Figure 5.2A, all output points are offset from the ideal (shown as a dotted line) by the same amount. Any such error—either positive or negative—that affects all output points by the same amount is an *offset* error.

The offset error can be measured by applying the all "0"s code to the DAC and measuring the output deviation from 0 volts.

Figure 5.2B shows the effect of *gain* error only. The ideal transfer function has a slope defined by drawing a straight line through the two end points. The slope represents the gain of the transfer function. In non-ideal DACs, this slope can differ from the ideal, resulting in a *gain* error—which is usually expressed as a percent because it affects each

code by the same percentage. If there is no offset error, gain error is easily determined by applying the all "1"s code to the DAC and measuring its output, designated as V_{111} (assuming a 3-bit DAC). An ideal DAC will measure exactly V_{FS} – 1 LSB, so the gain error is computed using the equation:

$$\text{Gain Error}(\%) = 100\left[\frac{V_{111}}{V_{FS} - 1\,\text{LSB}} - 1\right]. \qquad \text{Eq. 5.1}$$

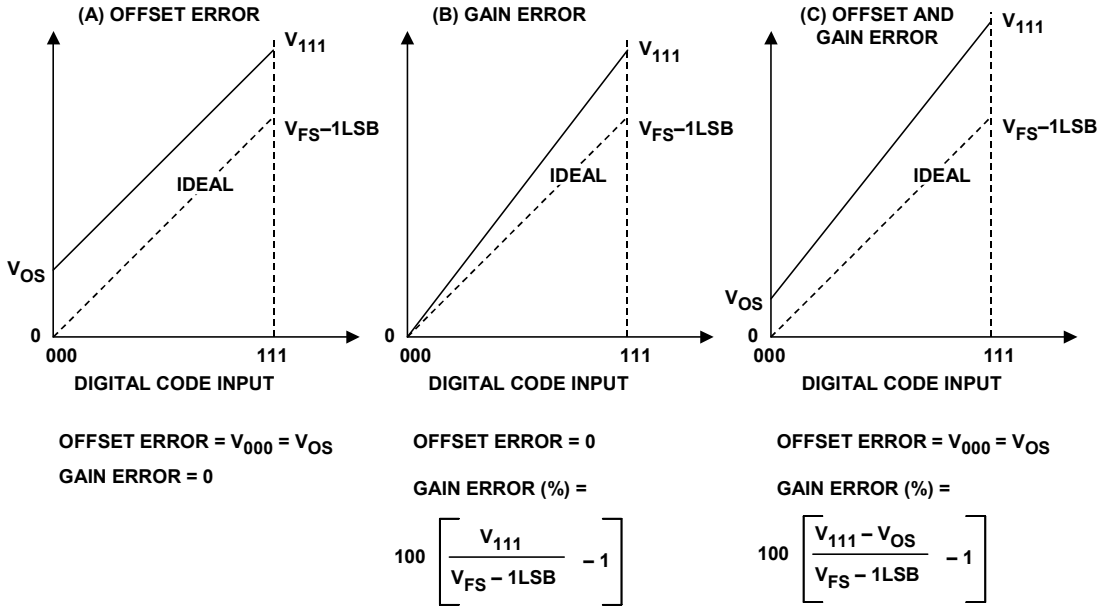

Figure 5.2: *Measuring Offset and Gain Error in a Unipolar DAC*

Figure 5.2C shows the case where there is both offset and gain error. The first step is to measure the offset error, V_{OS}, by applying the all "0"s code and measuring the output. Next, apply the all "1"s code and measure the output V_{111}. The gain error is then calculated using the equation:

$$\text{Gain Error}(\%) = 100\left[\frac{V_{111} - V_{OS}}{V_{FS} - 1\,\text{LSB}} - 1\right]. \qquad \text{Eq. 5.2}$$

Figure 5.3 shows how gain and offset errors affect the ideal transfer function of a bipolar output DAC. The offset error in Figure 5.3A, V_{OS}, is measured by applying the all "0"s code to the DAC input and measuring the output. Ideally, the DAC should have an output of –FS with all "0"s at its input. The difference between the actual output and –FS is the offset, V_{OS}. In a bipolar DAC it is also common to specify and measure the *bipolar zero error* (or *zero error*) because of its importance in many applications. It is measured by applying the mid-scale code 100 to the DAC and measuring its output. If there is no gain error, the bipolar zero error is the same as the offset error as shown in Figure 5.3A.

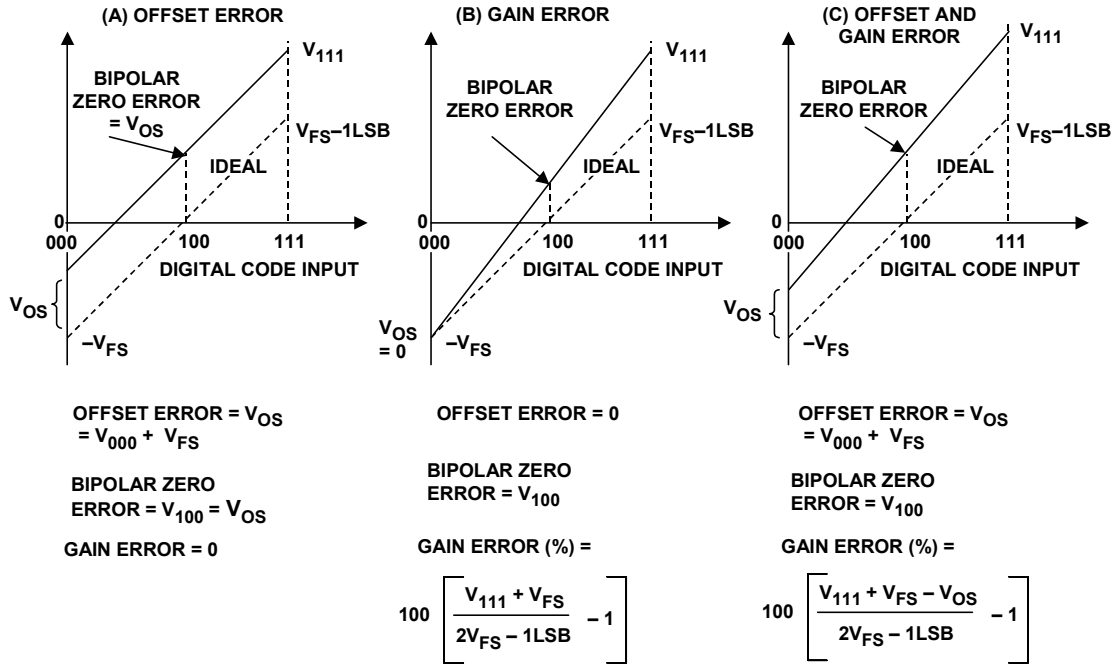

Figure 5.3: *Measuring Offset, Bipolar Zero, and Gain Error in a Bipolar DAC*

Figure 5.3B shows the case where there is gain error, but no offset error. Notice that the bipolar zero error is affected by the gain error. The DAC output V_{111} is measured by applying the all "1"s code, and the gain error is calculated from the equation:

$$\text{Gain Error}(\%) = 100\left[\frac{V_{111} + V_{FS}}{2V_{FS} - 1\,\text{LSB}} - 1\right]. \qquad \text{Eq. 5.3}$$

The bipolar zero error is determined by applying 100 to the DAC and measuring its output.

Figure 5.3C shows the case where the bipolar DAC has both gain and offset error. The offset error is determined as above by applying the all "0"s code, measuring the DAC output, and subtracting it from the ideal value, V_{FS}. The all "1"s code is applied to the DAC and the output V_{111} is measured. The gain error is calculated using the equation:

$$\text{Gain Error}(\%) = 100\left[\frac{V_{111} + V_{FS} - V_{OS}}{2V_{FS} - 1\,\text{LSB}} - 1\right]. \qquad \text{Eq. 5.4}$$

The bipolar zero error is determined by applying 100 to the DAC and measuring its output.

Bipolar zero error in DACs using offset-type coding is a derived, rather than a fundamental quantity, because it is actually the sum of the bipolar offset error, the bipolar

gain error, and the MSB linearity error. For this reason, it is important to specify whether this measurement is made before or after offset and gain have been trimmed or taken into account. Because of this error sensitivity, DACs that crucially require small errors at zero are usually unipolar types, with sign-magnitude coding and polarity-switched output amplifiers.

Linearity Errors

In a DAC, we are concerned with two measures of the linearity of its transfer function: integral nonlinearity, INL (or relative accuracy), and differential nonlinearity, DNL.

Integral nonlinearity is the maximum deviation, at any point in the transfer function, of the output voltage level from its ideal value—which is a straight line drawn through the actual zero and full-scale of the DAC.

Differential nonlinearity is the maximum deviation of an actual analog output step, between adjacent input codes, from the ideal step value of +1 LSB, calibrated based on the gain of the particular DAC. If the differential nonlinearity is more negative than –1 LSB, the DACs transfer function is non-monotonic.

Superposition and DAC Errors

Before proceeding with illustrations of DAC transfer functions showing linearity errors, it would be useful to consider the property of *superposition*, and be able to recognize its signature. Mathematically, superposition, a property of linear systems, implies that, if the influences of a number of phenomena at a particular point are measured individually, with all other influences at zero as each is asserted, the resulting total, with any number of these influences operating, will always be equal to the arithmetic sum of the individual measurements.

For example, let us assume that a simple binary-weighted DAC is ideal, except that each bit has a small linearity error associated with it. If each bit error is independent of the state of the other bits, then the linearity error at any code is simply the algebraic sum of the errors of each bit in that code (i.e., superposition holds). In addition, by using end-point linearity, we have defined the linearity error at zero and full-scale to be zero. Thus, the sum of all the bit errors must be zero, since all bits are summed to give the all "1"s value.

The bit errors can be either positive or negative; therefore, if their sum is zero, the sum of the positive errors (positive summation) must be equal to the sum of the negative errors (negative summation). These two summations constitute the worst-case integral nonlinearities of the DAC.

Intelligent use of superposition, coupled with a complete understanding of the architecture of the DAC under test, generally allows for a reduction in the number of measurements required to adequately determine DNL and INL. This is significant when one considers that a 16-bit DAC has a total of $2^{16} = 65{,}536$ possible output levels. Measuring each level individually would be a time-consuming task.

Measuring DAC DNL and INL Using Superposition

One can often determine whether or not superposition holds true by simply examining the architecture of a DAC, and this topic was covered in detail in Chapter 3 of this book (see Section 3.1). Superposition generally holds true for binary weighted DACs, so we will examine this class first. Issues relating to linearity measurements on fully-decoded DACs (string DACs), segmented DACs, and sigma-delta DACs will be treated later in this section, as they generally do not obey the rules of superposition.

Figure 5.4 shows the transfer function of a 3-bit DAC where superposition holds. Offset and gain errors have been removed from the data points so that the zero and full-scale errors are zero. This DAC has an error in the first and second bit weights. Bit 1 is 1.5 LSBs low, and bit 2 is 1.5 LSBs high. The value of the DNL is calculated for each of the eight possible output voltages. The transfer function has a non-monotonicity at the 100 code with a DNL of –3 LSBs. The INL for any output is the algebraic sum of the DNLs leading up to that particular output. For instance, the INL at the 101 code is equal to $DNL_{001} + DNL_{010} + DLN_{011} + DNL_{100} + DNL_{101} = 0 + 1.5 + 0 – 3 + 0 = –1.5$ LSBs.

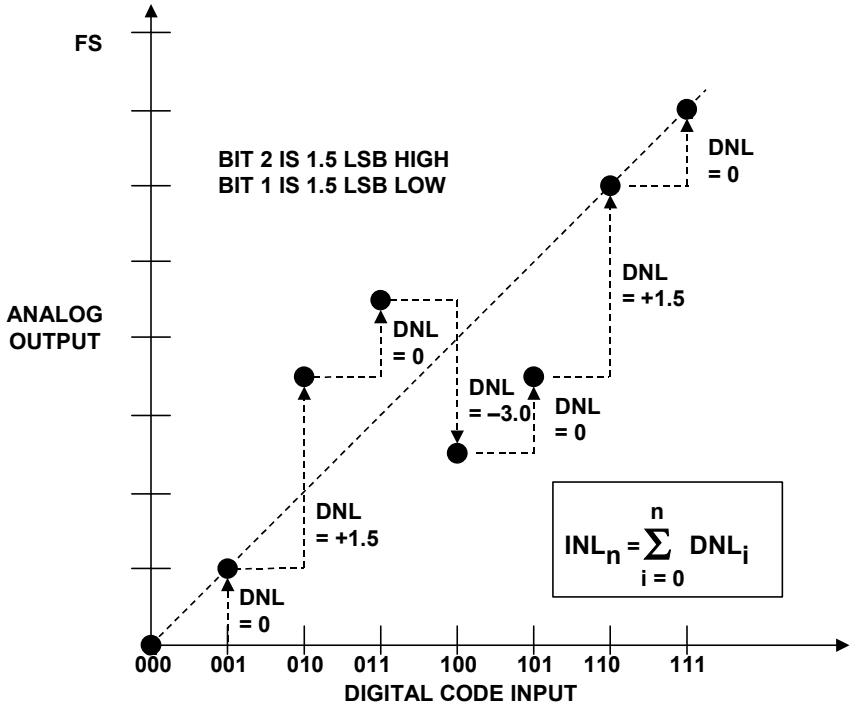

Figure 5.4: *Non-Ideal 3-Bit DAC Transfer Function Where Superposition Holds*

The DNL and INL of the 3-bit DAC is plotted in Figure 5.5. The INL has odd symmetry about the midpoint of the transfer function, i.e., the INL of any particular code is equal and opposite in sign from the INL of the complementary code. For instance, the INL at code 010 is +1.5, and the INL at the complementary code 101 is –1.5. In addition, the algebraic sum of all the INLs must equal zero, i.e., the sum of the positive INLs must equal the sum of the negative INLs.

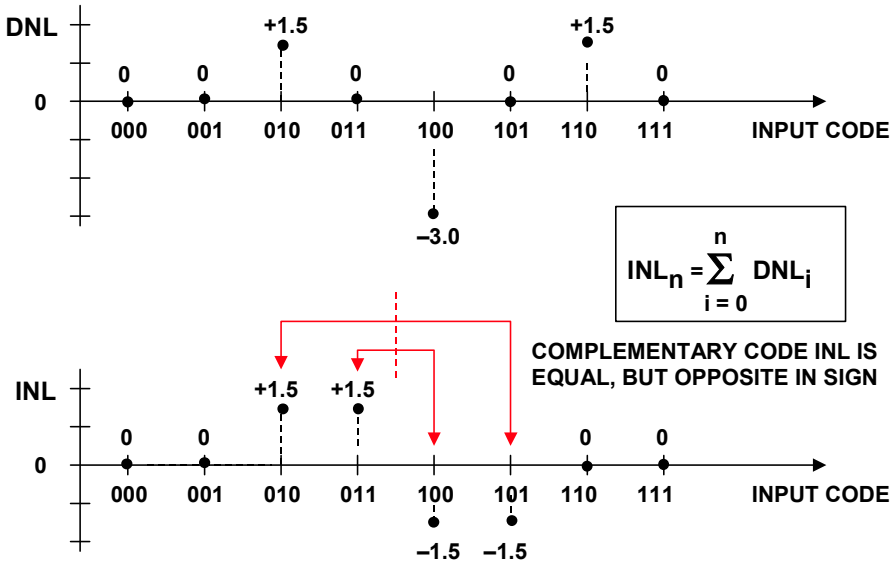

Figure 5.5: *Differential and Integral Linearity for Non-Ideal 3-Bit Binary-Decoded DAC Where Superposition Holds*

In order to further illustrate the relationship between INL and DNL, Figure 5.6 shows plots for a 4-bit DAC where superposition holds. Because of superposition, it is not necessary to measure the linearity error at all codes. The INL for each of the four bits is measured, corresponding to the codes 0001, 0010, 0100, and 1000. The codes 0001 and 0100 have positive errors, therefore the worst case positive INL occurs at the code 0101 and is equal to the sum of the two INL errors, 0.25 + 0.5 = 0.75 LSBs. The worst case negative INL occurs at the complementary code 1010 and is equal in magnitude to the worst case positive INL.

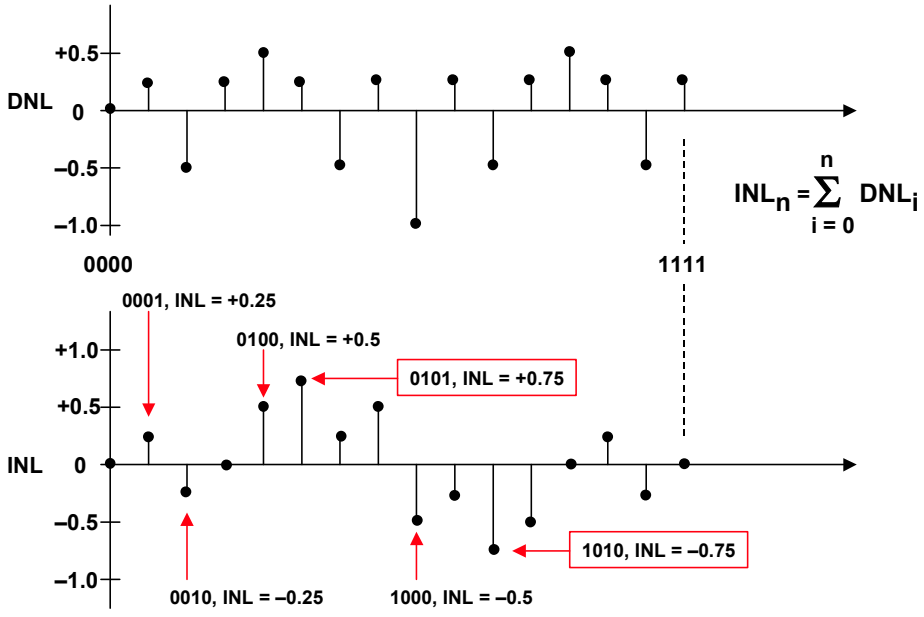

Figure 5.6: *DNL and INL for 4-Bit DAC Where Superposition Holds*

Figure 5.7 summarizes the linearity measurements required for a 4-bit binary-weighted DAC where superposition holds. First, the offset and gain error must be removed. The offset corresponds to the all "0"s code output, $V_{OS} = V_{0000}$. Next, measure the all "1"s code output, V_{1111}. The theoretical value of 1 LSB (with gain and offset errors removed) can then be calculated using the equation:

$$1\ \text{LSB} = \frac{V_{1111} - V_{0000}}{15}. \qquad \text{Eq. 5.5}$$

The theoretical value of the points being tested is then computed by multiplying the LSB weight (per Eq. 5.5) by the base-10 value of the binary code and adding the offset voltage.

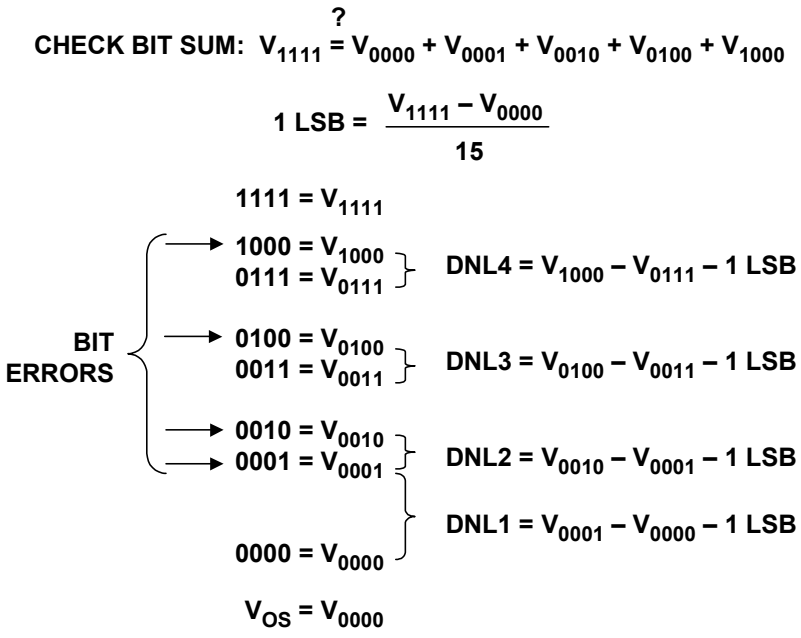

Figure 5.7: *Major-Carry Bit Tests for 4-Bit Binary-Decoded DAC Where Superposition Holds*

The voltages corresponding to codes 0000, 0001, 0010, 0100, and 1000 are then measured and the linearity error of each bit computed. The all "1"s code, 1111, previously measured should equal the sum of the individual bit voltages: 0000, 0001, 0010, 0100, and 1000. This is a good test to verify that superposition holds. A guideline is that if the algebraic sum of the actual bit voltages is not within ±0.5 LSB of the measured voltage for the all "1"s code, then there is enough interaction between the bits to justify more comprehensive all-codes testing. Next, the DNL is measured at each of the major-carry points: 0000 to 0001, 0001 to 0010, 0011 to 0100, and 0111 to 1000. In a well-behaved DAC where superposition holds, these tests should be sufficient to verify the static performance. The example shown in Figure 5.7 is for a 4-bit DAC which requires a total of 8 measurements plus the calculations of the individual errors, although the measurements constitute the major portion of the test time. In the general case of an N-bit DAC, the procedure just discussed requires 2N measurements plus the error calculations.

There is a method to further reduce the number of required measurements to N + 2, but it depends even more heavily upon the validity of superposition. In this method, only the individual bit values and the all "1"s code are measured. For the 4-bit case above, this would correspond to the DAC output for 0000, 0001, 0010, 0100, 1000, and 1111. Superposition should be verified by comparing V_{1111} with $V_{0000} + V_{0001} + V_{0010} + V_{0100} + V_{1000}$. The value of the LSB is computed per Eq. 5.5 above. The theoretical values of the individual bit contributions are then calculated by multiplying the base-10 bit weight by the value of the LSB and adding the offset voltage. The INL values for each bit are then computed. Finally, the DNL values are indirectly calculated from the bit values as follows:

$$DNL1 = V_{0001} - V_{0000} - 1\ LSB, \quad \text{Eq. 5.6}$$
$$DNL2 = V_{0010} - V_{0001} - 1\ LSB, \quad \text{Eq. 5.7}$$
$$DNL3 = V_{0100} - (V_{0010} + V_{0001} - V_{0000}) - 1\ LSB, \quad \text{Eq. 5.8}$$
$$DNL4 = V_{1000} - (V_{0100} + V_{0010} + V_{0001} - 2V_{0000}) - 1\ LSB. \quad \text{Eq. 5.9}$$

Here, the voltages corresponding to the 0011 and the 0111 code are computed using superposition rather than measured directly.

Linearity errors in DACs where superposition holds can take many forms. Figure 5.8 shows just two examples for a 3-bit binary-coded DAC. In both examples, the linearity error exhibits odd symmetry about the midpoint of the transfer function. Notice that in Figure 5.8A, the INL is ±1 LSB, and the DAC is non-monotonic because of the –2 LSB DNL at the 100 code. Figure 5.8B shows a DAC where the INL is ±0.5 LSB and the DNL is –1 LSB at the 100 code, thereby just bordering on non-monotonicity.

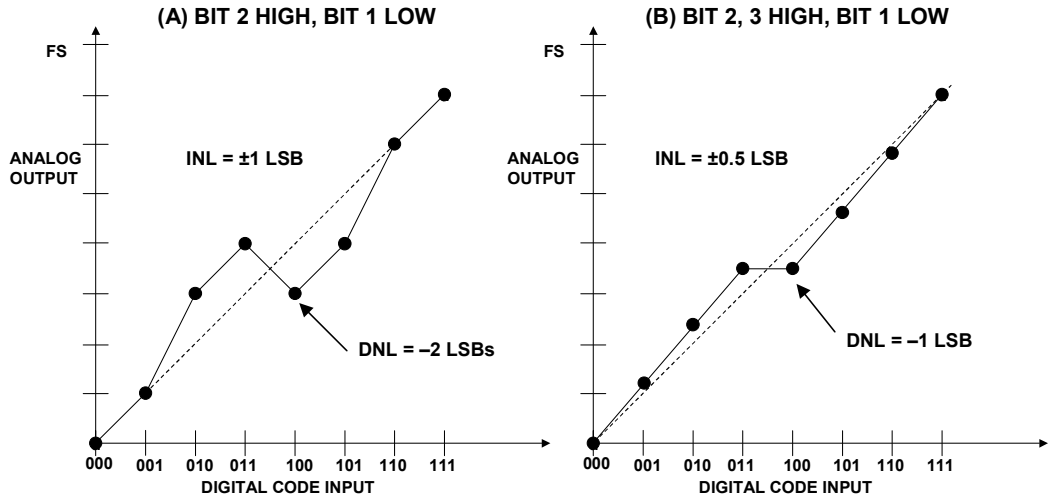

Figure 5.8: *Linearity Errors in Binary-Decoded 3-Bit DACs Where Superposition Holds*

We are now in a position to make two general statements about INL and DNL and their relationship to each other and to the monotonicity of a DAC. *First, if a DAC has an INL specification of less than ±0.5 LSB, this guarantees that its DNL is no more than ±1 LSB, and that the output is monotonic. On the other hand, just because the DNL error is less than ±1 LSB, and the DAC is monotonic, it cannot be assumed that the INL is less than ±0.5 LSB.*

The second point is illustrated in Figure 5.9 where the worst DNL error is –1 LSB, but the INL is ±0.75 LSBs.

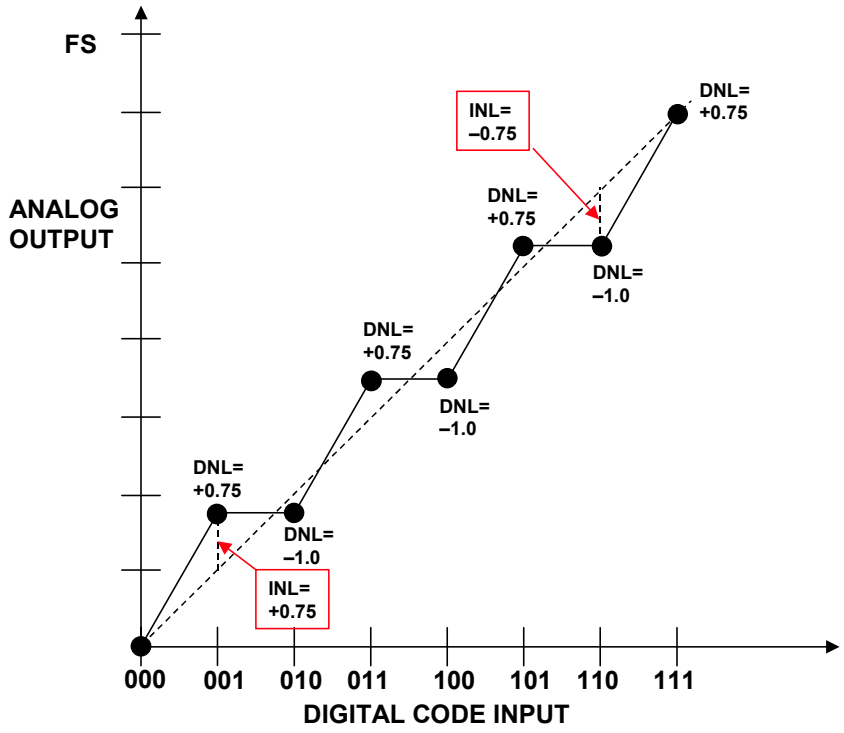

Figure 5.9: *DNL Specification of < ±1 LSB Does Not Guarantee INL < ±0.5 LSB*

Measuring DAC INL and DNL Where Superposition Does Not Hold

There are several DAC architectures where superposition does not hold, and the techniques described in the previous section may not give a clear picture of the INL and DNL performance. Again, the exact methodology of selecting data points is highly dependent on the architecture of the DAC under test, therefore it is not possible to examine them all but simply to point out some general concepts.

The simple fully-decoded (string DAC) architecture is a good example where the individual bit weights vary depending upon the values of the other bits. Superposition does not hold, and the bit weights are not independent as in the binary-weighted DACs previously discussed.

In a string DAC (Kelvin divider), the DNL is primarily determined by the matching of adjacent resistors in the "string", which is generally quite good. However, the INL may

follow either a bow- or an s-shaped curve due to gradual changes in the absolute value of the resistors when moving from one end of the string to the other. Figure 5.10 illustrates two possible transfer functions. Figure 5.10A shows a bow-shaped INL function with no trimming, while Figure 5.10B shows the effects of trimming the MSB (described in Chapter 3) where the mid-scale output is forced to the correct value.

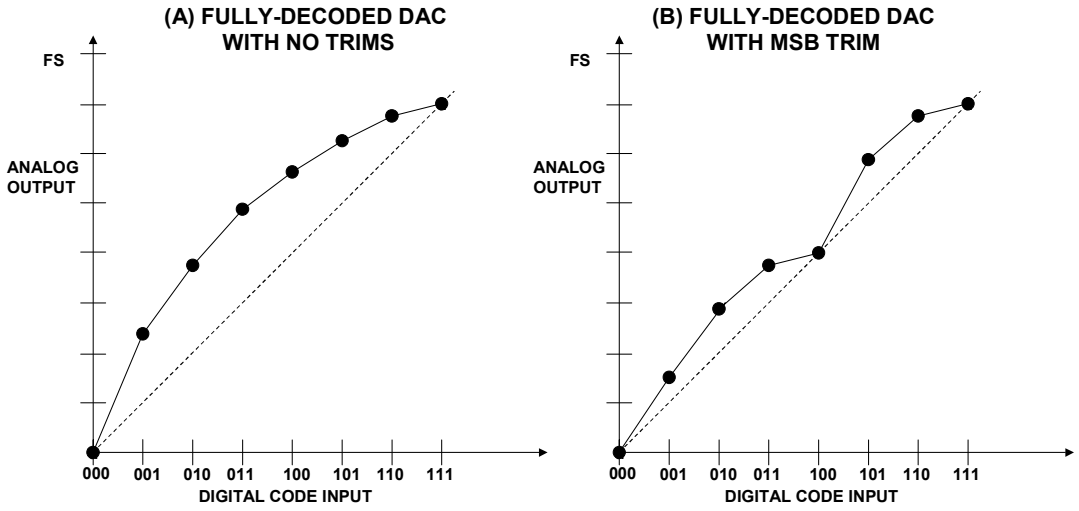

Figure 5.10: *INL of Fully-Decoded 3-Bit DACs (String DACs)*

The highest resolution stand-alone string DACs at present are 10-bit digital potentiometers (i.e., AD5231/AD5235). However, the fully-decoded DAC is a common building block in segmented low distortion high resolution DACs as discussed in Chapter 3. In practice, 4-, 5-, or 6-bit fully decoded segments are popular in modern DACs.

The only way to fully characterize a segmented DAC is to perform all-codes testing—a time consuming process, to say the least. However, one can establish a reasonable amount of confidence about segmented DACs by intelligent utilization of knowledge about the particular DAC architecture. As an example, Figure 5.11 shows a simplified segmented DAC consisting of a fully-decoded 3-bit MSB DAC and a 4-bit binary-weighted LSB DAC. The logic behind the selection of measurements on the right-hand side of the diagram is as follows.

First, the all "1"s code and all "0"s code are measured so that the weight of the LSB and the offset and can be determined. Next, each of the LSB DAC bits are tested so its DNL and INL can be determined (it is assumed that superposition holds for the LSB DAC, since it is a binary-weighted architecture). This is done with the MSB DAC bits at all "0"s for convenience. The DNL should then be measured at each of the MSB DAC transitions. In addition, the DNL is measured for code above the MSB DAC transitions where the LSB DAC code is 0001. This ensures that the lower and upper range of the LSB DAC is tested on each segment of the MSB DAC. Finally, one INL measurement is made in each of the 7 MSB segments, corresponding to the outputs where the LSB DAC code is 0000.

This technique for selecting test codes can obviously be extended to cover segmented DACs of higher resolution using the same principles in the simple example.

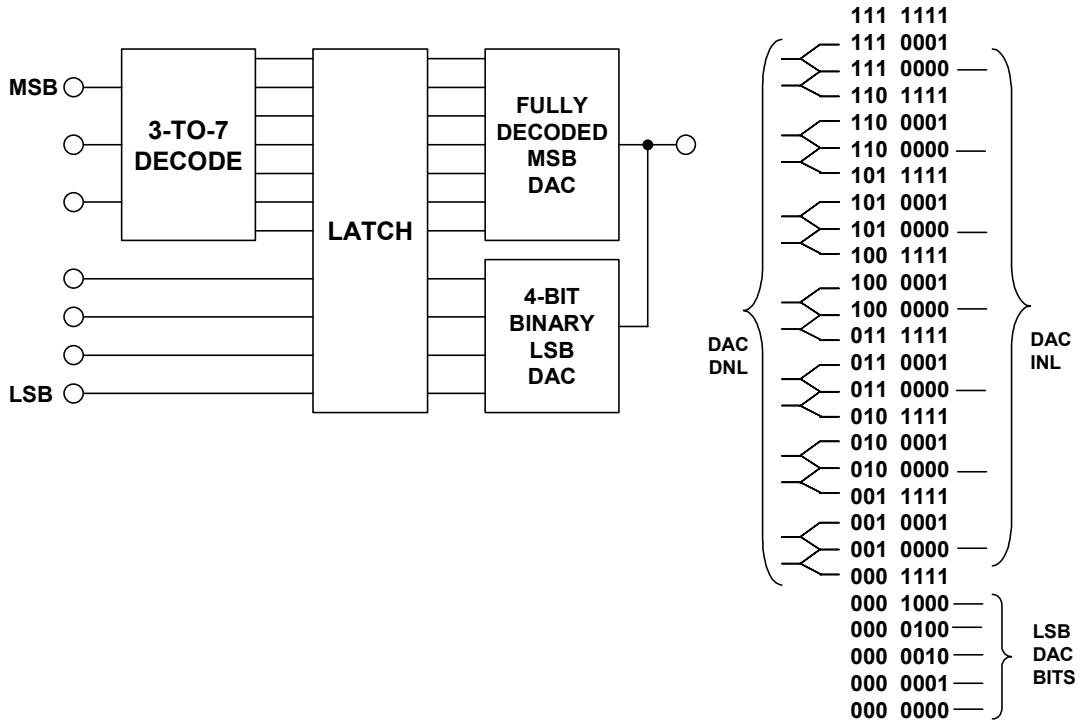

Figure 5.11: *3-Bit×4-Bit Segmented 7-Bit DAC Test Codes*

Much more could be said about testing DAC static linearity, but in light of the majority of today's applications and their emphasis on ac performance, the guidelines put forth in this discussion should be adequate to illustrate the general principles. As we have seen, foremost in importance is a detailed understanding of the particular DAC architecture so that a reasonable test/evaluation plan can be devised which minimizes the actual number of codes tested. This is highly dependent upon whether the DAC is binary-weighted (superposition generally holds), fully-decoded, or segmented. The data sheet for the DAC under test generally will provide adequate information to make this determination.

For the interested reader, References 1 and 2 give more details regarding testing DAC static linearity, including automatic and semi-automatic test methods such as using "reference" DACs as part of the test circuitry to verify the DAC under test. The entire area of ATE testing is much too broad to be treated here and of much more interest to the IC manufacturer than the end user.

Finally, there are several types of DACs which generally do not have static linearity specifications, or if they do, they do not compare very well with more traditional DACs designed for low frequency applications. The first of these are DACs designed for voiceband and audio applications. This type of DAC, although fully specified in terms of ac parameters such as THD, THD + N, etc., generally lacks dc specifications (other than perhaps gain and offset); and generally should not be used in traditional industrial control or instrumentation applications where INL and DNL are critical. However, these DACs almost always use the sigma-delta architecture (either single-bit or mult-bit with data scrambling) which inherently ensures good DNL performance.

DACs specifically designed for communications applications, such as the TxDAC™-family, have extensive frequency-domain specifications; but their static specifications make them less attractive than other more traditional DACs for precision low frequency applications. It is common to see INL and DNL specifications of several LSBs for 14- and 16-bit DACs in this family, with monotonicity guaranteed at the 12-bit level. It should by no means be inferred that these are inferior DACs—it is just that the application requires designs which optimize frequency-domain performance rather than static.

Testing DACs for Dynamic Performance

Settling Time

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

The fundamental definitions of full-scale settling time is repeated in Figure 5.12. The definition is quite similar to that of an op amp. Notice that settling time can be defined in two acceptable ways. The more traditional definition is the amount of time required for the output to settle with the specified error band measured with respect to the 50% point of either the data strobe to the DAC (if it has a parallel register driving the DAC switches) or the time when the input data to the switches changes (if there is no internal register).

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

The error band is usually defined in terms of an LSB or % full-scale. It is customary, but not mandatory, to define the error band as 1 LSB. However, measuring full-scale settling time to 1 LSB at the 12-bit level (0.025% FS) is possible with care, but measuring it to 1 LSB at the 16-bit level (0.0015% FS) presents a real instrumentation challenge. For this reason, high-speed DACs such as the TxDAC family specify 14- and 16-bit settling time to the 12-bit level, 0.025% FS (typically less than 11 ns).

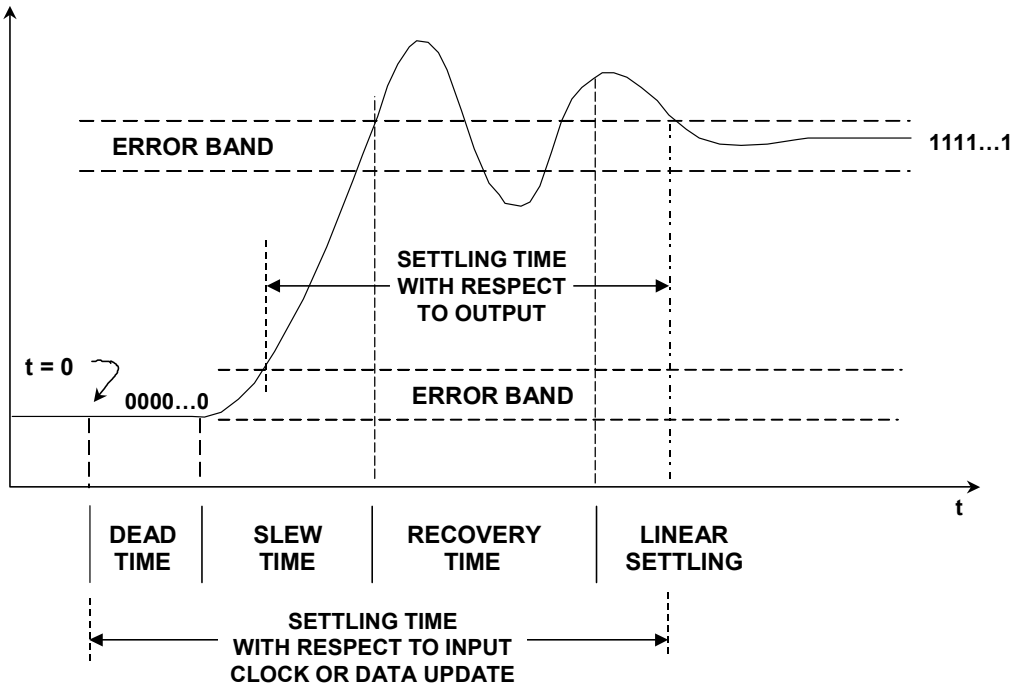

Figure 5.12: *DAC Full-Scale Settling Time Measurement*

Mid-scale settling time is of interest, because in a binary-weighted DAC, the transition between the 0111…1 code and the 1000…0 code produces the largest transient. In fact, if there is significant bit skew, the transient amplitude can approach full-scale. Figure 5.13 shows a waveform along with the two acceptable definitions of settling time. As in the case of full-scale settling time, mid-scale settling time can either be referred to the output or to the latch strobe (or the bit transitions if there is no internal latch).

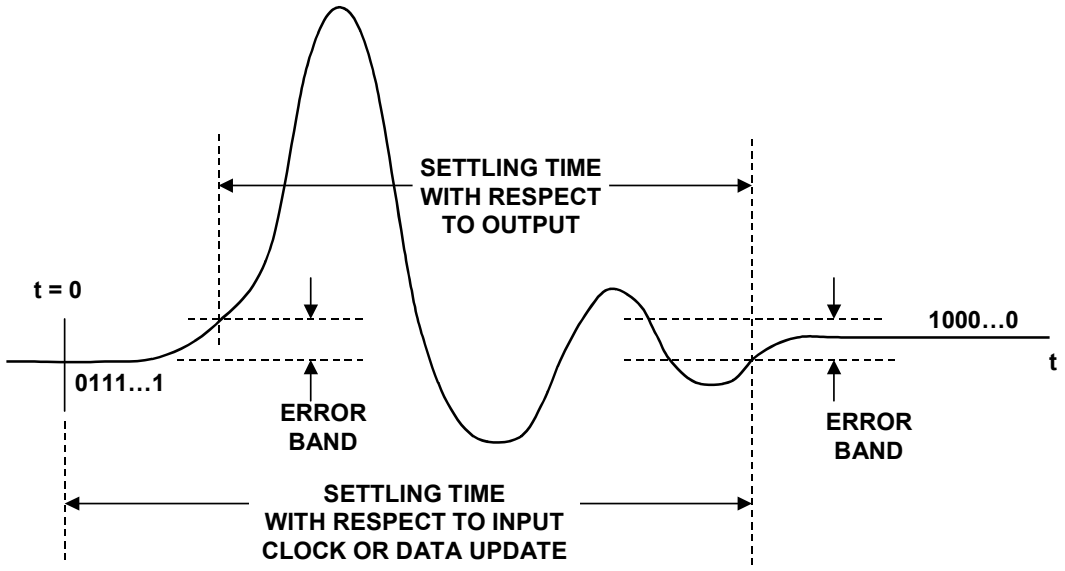

Figure 5.13: *DAC Mid-Scale Settling Time Measurement*

Glitch Impulse Area

Glitch impulse area (sometimes incorrectly called glitch energy) is easily estimated from the mid-scale settling time waveform as shown in Figure 5.13. The areas of the four triangles are used to calculate the net glitch area. Recall that the area of a triangle is one-half the base times the height. If the total positive area equals the total negative glitch area, then the net area is zero. The specification given on most data sheets is the net glitch area, although in some cases, the peak area may specified instead.

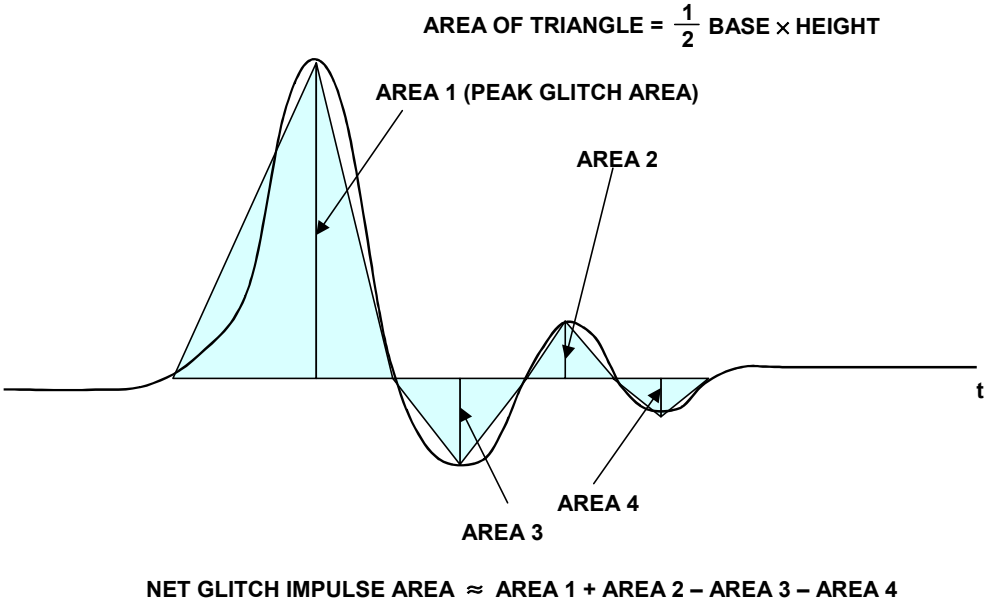

Figure 5.14: *Glitch Impulse Area Measurement*

Oscilloscope Measurement of Settling Time and Glitch Impulse Area

A wideband fast-settling oscilloscope is crucial to settling time measurements. There are several considerations in selecting the proper scope. The required bandwidth can be calculated based on the rise/falltime of the DAC output, for instance, a 1-ns output risetime and falltime corresponds to a bandwidth of $0.35/t_r = 350$ MHz. A scope of at least 500-MHz bandwidth would be required.

Modern digital storage scopes (DSOs) and digital phosphor scopes (DPOs) are popular and offer an excellent solution for performing settling time measurements as well as many other waveform analysis functions (see Reference 3). These scopes offer real-time sampling rates of several GHz and are much less sensitive to overdrive than older analog scopes or traditional sampling scopes. Overdrive is a serious consideration in measuring settling time, because the scope is generally set to maximum sensitivity when measuring a full-scale DAC output change. For instance, measuring 12-bit settling for a 1-V output (20 mA into 50 Ω) requires the resolution of a signal within a 0.25-mV error band riding on the top of a 1-V step function.

From a historical perspective, older analog oscilloscopes were sensitive to overdrive and could not be used to make accurate step function settling time without adding additional

circuitry. Quite a bit of work was done during the 1980s on circuits to cancel out portions of the step function using Schottky diodes, current sources, etc. References 4, 5, and 6 are good examples of various circuits which were used during that time to mitigate the oscilloscope overdrive problems.

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

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

Figure 5.15 shows a test setup for measuring settling time. It is generally better to make a direct connection between the DAC output and the 50-Ω scope input and avoid the use of probes. FET probes are notorious for giving misleading settling time results. If probes must be used, compensated passive ones are preferable, but they should be used with care. Skin effect associated with even short lengths of properly terminated coaxial cable can give erroneous settling time results. In making the connection between the DAC and the scope, it is mandatory that a good low impedance ground be maintained. This can be accomplished by soldering the ground of a BNC connector to the ground plane on the DAC test board and using this BNC to connect to the scope's 50-Ω input. A manufacturer's evaluation board can be of great assistance in interfacing to the DAC and should be used if available.

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

$$\text{DAC Settling Time} = \sqrt{(\text{Total Settling Time})^2 - (\text{Op Amp Settling Time})^2}\ . \quad \text{Eq. 5.10}$$

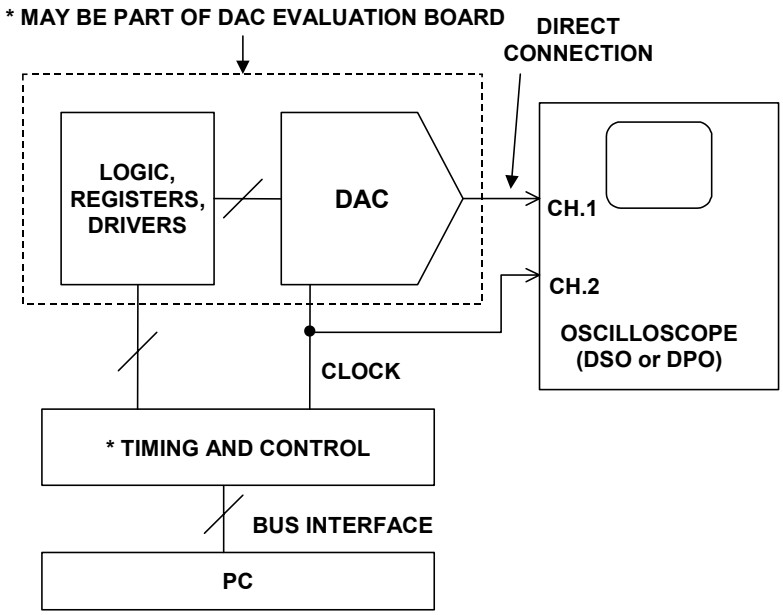

Figure 5.15: *Test Setup for Measuring Settling Time and Glitch Impulse Area*

Distortion Measurements

Because so many DAC applications are in communications and frequency analysis systems, practically all modern DACs are now specified in the frequency domain. The basic ac specifications were discussed previously in Chapter 3 of this book and include harmonic distortion, total harmonic distortion (THD), signal-to-noise ratio (SNR), total harmonic distortion plus noise (THD + N), spurious free dynamic range (SFDR), etc. In order to test a DAC for these specifications, a proper digitally-synthesized signal must be generated to drive the DAC (for example, a single or multi-tone sinewave).

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

Figure 5.16 shows a typical test setup for measuring the distortion and noise of a DAC. The first consideration, of course, is the generation of the digital signal to drive the DAC. To achieve this, modern arbitrary waveform generators (for example Tektronix AWG2021 with Option 4) or word generators (Tektronix DG2020) allow almost any waveform to be synthesized digitally in software, and are mandatory in serious frequency domain testing of DACs (see Reference 3). In most cases, these generators have standard waveforms pre-programmed, such as sinewaves and triangle waves, for example. In many communications applications, however, more complex digital waveforms are

required, such as two-tone or multi-tone sinewaves, QAM, GSM, and CDMA test signals, etc. In many cases, application-specific hardware and software exists for generating these types of signals and can greatly speed up the evaluation process.

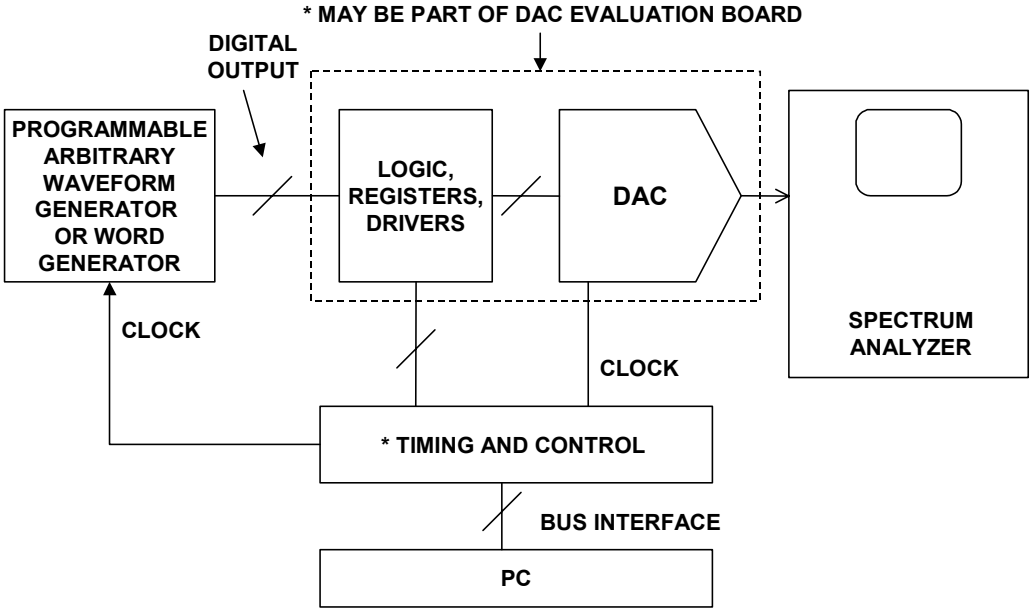

Figure 5.16: *Test Setup for Measuring DAC Distortion and Noise*

Analog Devices and other manufacturers of high performance DACs furnish evaluation boards which greatly simplify interfacing to the test equipment. Because many communications DACs (such as the TxDAC®-family) have quite a bit of on-chip control logic, their evaluation boards have interfaces to PCs via the SPI, USB, parallel, or serial ports, as well as Windows®-compatible software to facilitate setting the various DAC options and modes of operation.

Testing DACs which are part of a direct-digital-synthesis (DDS) system is somewhat easier because the DDS portion of the IC acts as the digital signal generator for the DAC. Testing these DACs often requires no more than the manufacturer's evaluation board, a PC, a stable clock source, and a spectrum analyzer. A complete discussion of DDS is included in Chapter 8 of this book and will not be repeated here.

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

amount. If the harmonics drop more than the fundamental signal drops, then the analyzer is distorting the signal.

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

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

Once the proper analyzer is selected, measuring the various distortion and noise-related specifications such as SFDR, THD, SNR, SINAD, etc., is relatively straightforward. Refer back to Chapter 2 of this book for definitions if required. The analyzer resolution bandwidth must be set low enough so that the harmonic products can be resolved above the noise floor. Figure 5.17 shows a typical spectral output where the SFDR is measured.

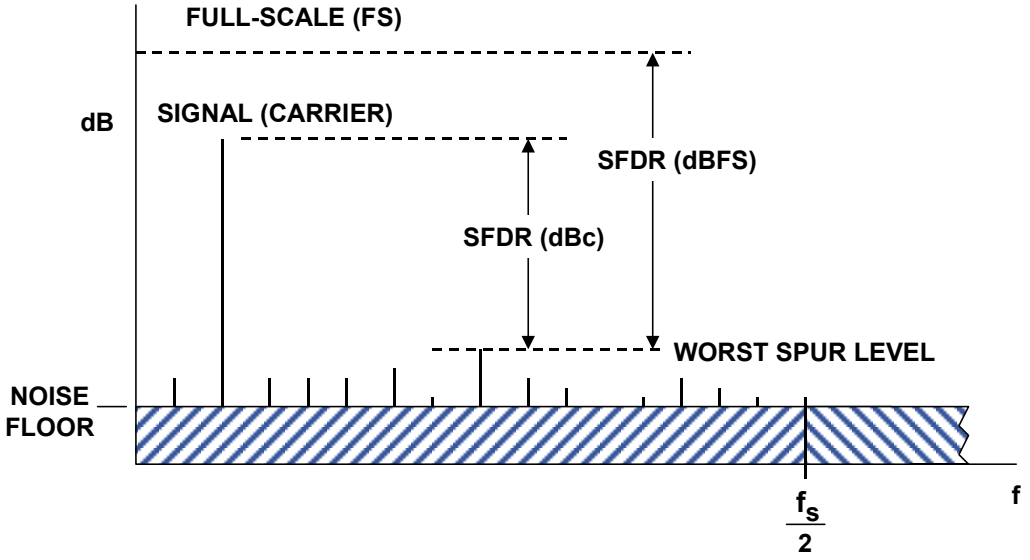

Figure 5.17: *Measuring DAC Spurious Free Dynamic Range (SFDR)*

Figure 5.18 shows how to measure the various harmonic distortion components with a spectrum analyzer. The first nine harmonics are shown. Notice that aliasing causes the 6^{th}, 7^{th}, 8^{th}, 9^{th}, and 10^{th} harmonic to fall back inside the $f_s/2$ Nyquist bandwidth.

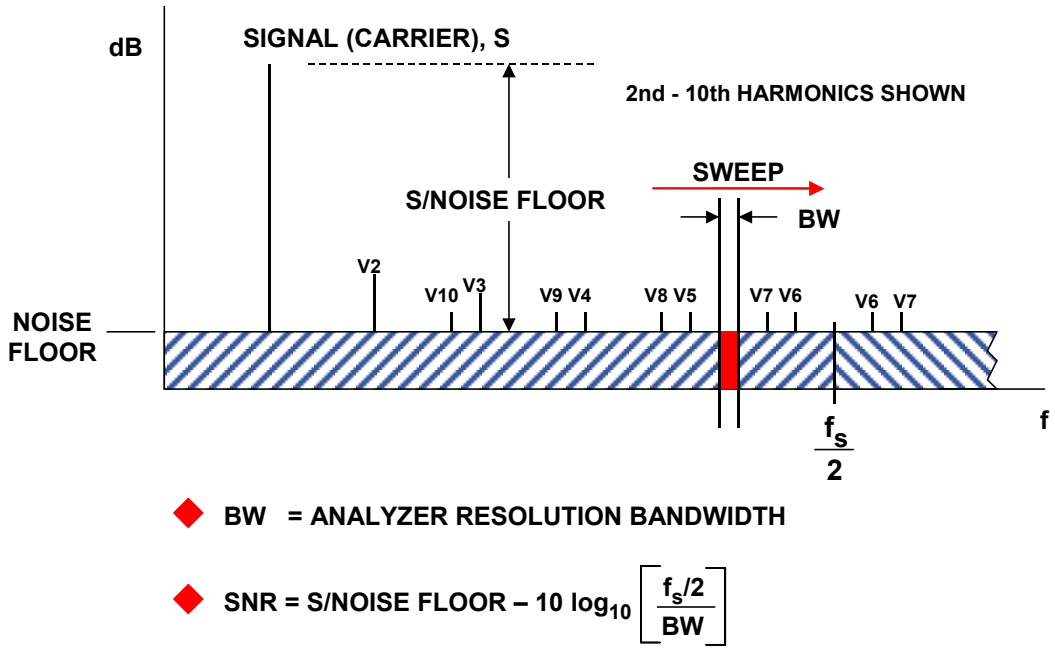

Figure 5.18: *Measuring DAC Distortion and SNR with an Analog Spectrum Analyzer*

The spectrum analyzer can also be used to measure SNR if the proper correction factors are taken into account. Figure 5.18 shows the analyzer sweep bandwidth, BW, which in most cases will be considerably less than $f_s/2$. First, measure the noise floor level with respect to the signal level at a point in the frequency spectrum which is relatively free of harmonics. This corresponds to the value "S/NOISE FLOOR" in the diagram. The actual SNR over the dc to $f_s/2$ bandwidth is obtained by subtracting the process gain, $10\log_{10}(f_s/2 \cdot BW)$.

$$\text{SNR} = \text{S/NOISE FLOOR} - 10\log_{10}(f_s/2 \cdot BW). \qquad \text{Eq. 5.11}$$

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

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

$$\text{Process Gain} = \text{S/Noise Floor} - \text{SNR}. \qquad \text{Eq.5.12}$$

The accuracy of this measurement should be verified by enabling the 9^{th} bit of the DAC and ensuring that the analyzer noise floor drops by 6 dB. If the noise floor does not drop by 6 dB, the measurement should be repeated using only the first 6 bits of the DAC. If

near theoretical SNR is not achieved at the 6-bit level, the DAC under consideration is probably not suitable for ac applications where noise and distortion are important.

The relationship between SINAD, SNR, and THD is shown in Figure 5.19. THD is defined as the ratio of the signal to the root-sum-square (rss) of a specified number harmonics of the fundamental signal. IEEE Std. 1241-2000 (Reference 9) suggests that the first 10 harmonics be included. Various manufacturers may choose to include fewer than 10 harmonics in the calculation. Analog Devices defines THD to be the root-sum-square of the first 6 harmonics (2^{nd}, 3^{rd}, 4^{th}, 5^{th}, and 6^{th}) for example. In practice, the difference in dB between THD measured with 10 versus 6 harmonics is less than a few tenths of a dB, unless there is an extreme amount of distortion. The various harmonics, V1 through V6, are measured with respect to the signal level, S, in dBc. They are then converted into a ratio, combined on an rss basis, and converted back into dB to obtain the THD.

- $\text{SNR} = \text{S/NOISE FLOOR} - 10\log_{10}\left[\frac{f_S/2}{BW}\right]$
- $\text{THD} = 20\log_{10}\sqrt{\left[10^{-V2/20}\right]^2 + \left[10^{-V3/20}\right]^2 + \cdots + \left[10^{-V6/20}\right]^2}$
- $\text{SINAD} = 20\log_{10}\sqrt{\left[10^{-SNR/20}\right]^2 + \left[10^{-THD/20}\right]^2}$

NOTE: NOISE FLOOR, SNR, THD, SINAD, V2, V3, … , V6 in units of dBc

Figure 5.19: *Calculating S/(N+D) (SINAD) from SNR and THD*

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

$$\text{SINAD} = 20\log_{10}\sqrt{\left(10^{-\text{SNR}/20}\right)^2 + \left(10^{-\text{THD}/20}\right)^2}. \qquad \text{Eq. 5.13}$$

One of the most important factors in obtaining accurate distortion measurements is to ensure that the DAC output frequency, f_o is not a sub-harmonic of the update rate, f_c. If f_c/f_o is an integer, then the quantization error is not random, but is correlated with the output frequency. This causes the quantization noise energy to be concentrated at harmonics of the fundamental output frequency, thereby producing distortion which is an artifact of the sampling process rather than nonlinearity in the DAC. It should be noted that these same artifacts occur in testing ADCs as previously described in Chapter 2 of this book.

To illustrate this point, Figure 5.20 shows simulated results for an ideal 12-bit DAC where the left-hand diagram shows the output frequency spectrum for the case of f_c/f_o = 32. Notice that the SFDR is approximately 76 dB. The right-hand spectral output shows

the case where the f_c/f_o ratio is a non-integer—the quantization noise is now random—and the SFDR is 92 dB.

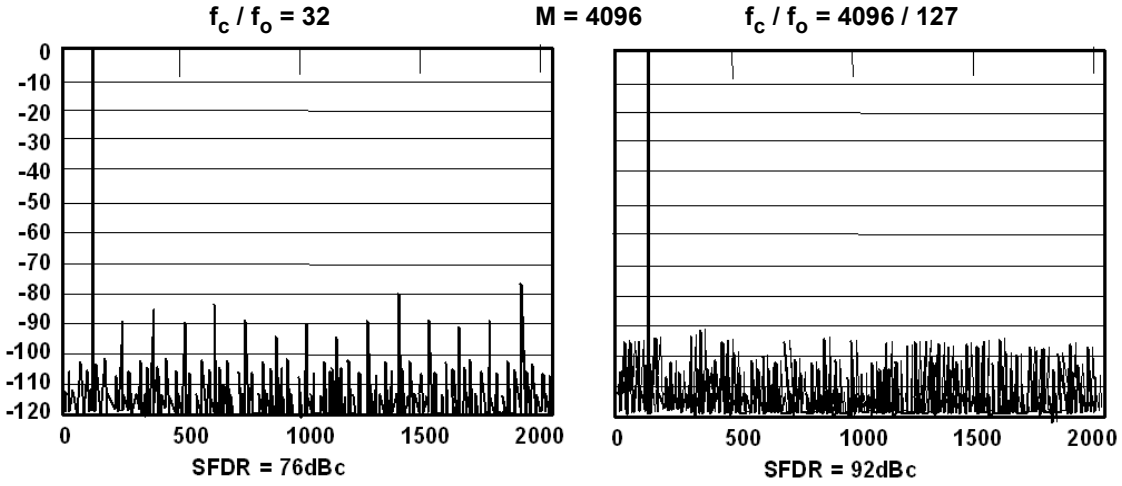

Figure 5.20: *Effect of Ratio of Sampling Clock f_c to Output Frequency f_o on SFDR for Ideal 12-bit DAC*

Accurately specifying and measuring spectral purity of DACs used in frequency synthesis applications therefore presents a significant challenge to the manufacturer because of the large number of possible combinations of clock and output frequencies as well as output signal amplitudes. Data is traditionally presented in several formats. Figure 5.21 shows two possible spectral outputs of the AD9851 10-bit DDS DAC updated at 180 MSPS. Notice that the two output frequencies (1.1 MHz and 40.1 MHz) are chosen such that they are not a sub-harmonic of the 180-MSPS clock frequency.

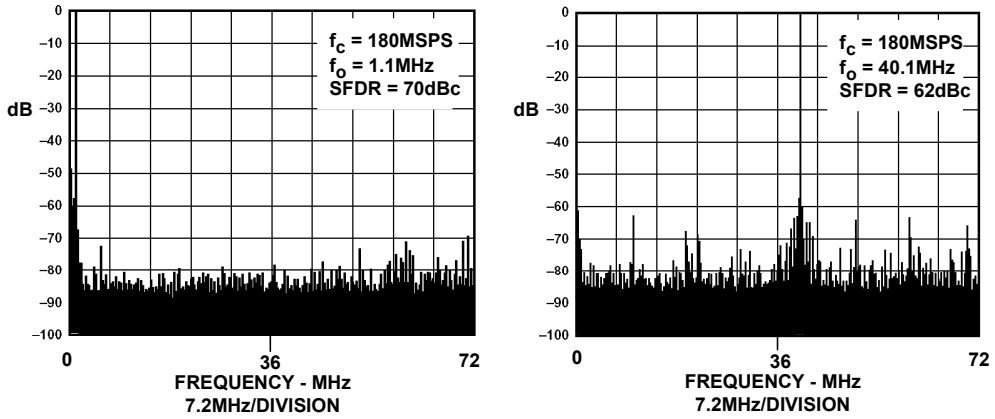

Figure 5.21: *AD9851 10-Bit, 180-MSPS DDS Spectral Output*

The TxDAC®-series of DACs have been specifically optimized for low distortion and noise as required in communications systems. The 14-bit AD9744 is an example of the family, and its single and dual-tone performance for an output frequency of approximately 15 MHz and a clock frequency of 78 MSPS is shown in Figure 5.22.

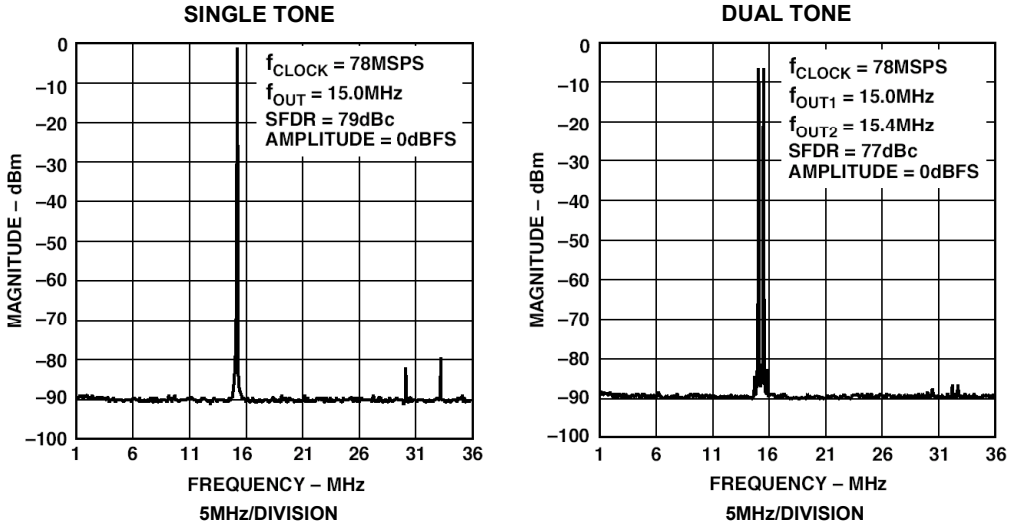

Figure 5.22: *AD9744, 14-Bit, 165-MSPS TxDAC® Spectral Output*

Because of the wide range of possible clock and output frequencies, Analog Devices offers special fast-turnaround measurements (typically 48 hours) on TxDACs for specific customer test vectors. This powerful service allows system designers to do advance frequency planning to ensure optimum distortion performance for their application.

In lieu of specific frequency measurements, there is another useful test method that gives a good overall indicator of the DAC performance at various combinations of output and clock frequencies. Specifically, this involves testing distortion for output frequencies, f_o, equal to $f_c/3$ and $f_c/4$. In practice, the output frequency is slightly offset by a small amount, Δf, where Δf is a non-integer fraction of f_c, i.e., $\Delta f = kf_c$, where $k << 1$. For an output frequency of $f_c/3 - \Delta f$, the even-order harmonics are spaced at intervals of Δf around the fundamental f_o output frequency as shown in Figure 5.23. The worst even-order harmonic is measured at various clock frequencies up to the maximum allowable while maintaining this same ratio. The same procedure should be repeated for an output frequency $f_c/4 - \Delta f$, in which case the odd-order harmonics are uniformly spaced around the output frequency as shown in Figure 5.24.

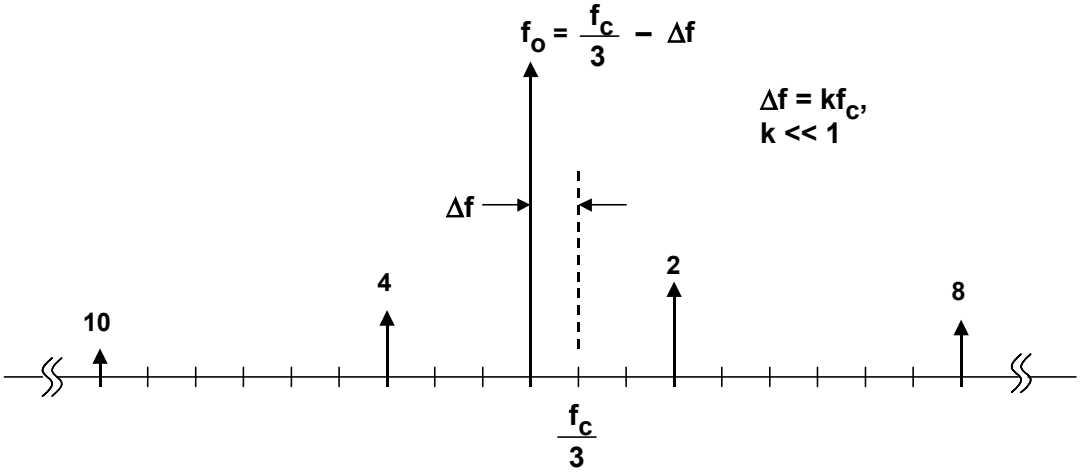

Figure 5.23: *Location of Even Harmonics for $f_o = f_c/3 - \Delta f$*

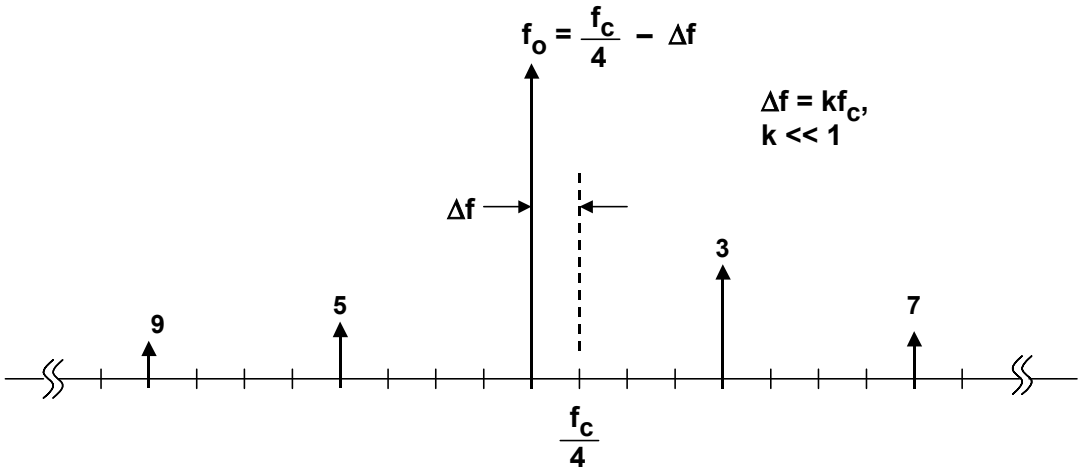

Figure 5.24: *Location of Odd Harmonics for $f_o = f_c/4 - \Delta f$*

These measurements are relatively easy to make, since once the ratio of f_o to f_c is established by the DDS or digital waveform generator, it is preserved as the clock frequency is changed. Figure 5.25 shows a typical plot of SFDR versus clock frequency for a low distortion DAC with two output frequencies $f_c/3$ and $f_c/4$. In most cases, the $f_c/3$ distortion represents a worst case condition and is good for comparing various DACs.

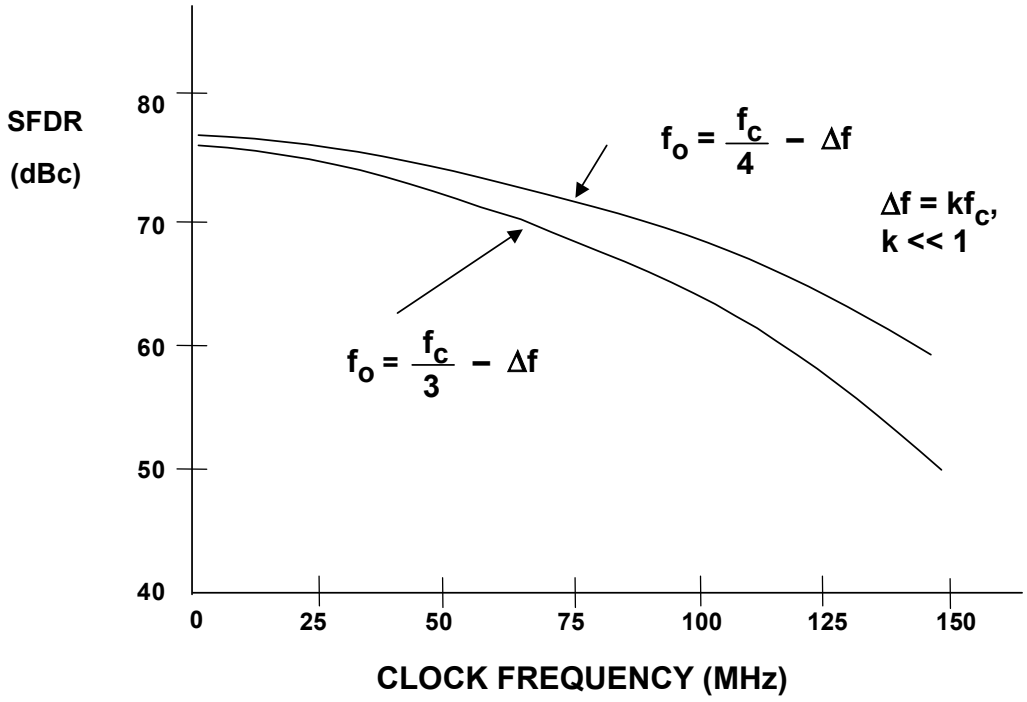

Figure 5.25: *Worst Harmonic vs. Clock Frequency for $f_o = f_c/3 - \Delta f$ and $f_o = f_c/4 - \Delta f$*

REFERENCES:
5.1 TESTING DACs

1. Jim R. Naylor, "Testing Digital/Analog and Analog/Digital Converters," **IEEE Transactions on Circuits and Systems**, Vol. CAS-25, July 1978, pp. 526-538.

2. 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).*

3. **Tektronix, Inc.,**14200 SW Karl Braun Drive, P. O. Box 500, Beaverton, OR 97077, Phone: (800) 835-9433, http://www.tek.com. *(the website contains a wealth of information on oscilloscopes, measurement techniques, probing, etc., as well as complete specifications on products).*

4. Howard K. Schoenwetter, "High Accuracy Settling Time Measurements," **IEEE Transactions on Instrumentation and Measurement**, Vol. IM-32, No. 1, March 1983, pp. 22-27.

5. James R. Andrews, Barry A. Bell, Norris S. Nahman, and Eugene E. Baldwin, "Reference Waveform Flat Pulse Generator," **IEEE Transactions on Instrumentation and Measurement**, Vol. IM-32, No. 1, March 1983, pp. 27-32.

6. Barry Harvey, "Take the Guesswork out of Settling-Time Measurements," **EDN**, September 19 1985, pp. 177-189.

7. **Rohde & Schwarz, Inc**., 8661A Robert Fulton Dr.,Columbia, MD 21046-2265, Phone: (410) 910-7800, http://www.rohde-schwarz.com. *(a premier manufacturer of spectrum analyzers, the website contains tutorials on frequency analysis as well as product specifications).*

8. **Audio Precision**, 5750 S.W. Arctic Drive, Beaverton, Oregon 97005, http://www.audioprecision.com. *(the recognized industry standard for professional audio measurement equipment).*

9. **IEEE Std. 1241-2000, IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters**, IEEE, 2001, ISBN 0-7381-2724-8.

NOTES:

SECTION 5.2: TESTING ADCs

Walt Kester

A Brief Historical Overview of Data Converter Specifications and Testing

Although the overall history of data converters has been outlined in Chapter 1 of this book, the evolution of data converter specifications and associated testing methods deserves additional comment here. It is interesting to note how DSP-based tests, in particular, have become nearly universally-accepted industry standards for today's ADCs.

The development of data converters, starting in the 1940s and continuing until today, can still be divided roughly into two paths depending on the sampling frequency: low-speed (usually associated with higher precision up to 24-bits), and high-speed (generally associated with lower precision—but recently extending into the 14- and 16-bit level, so the boundary is becoming less clear). This was certainly true in the 1940s and 1950s, when ADCs and DACs for PCM applications in the Bell System typically required sampling frequencies in the 100-kSPS range at resolutions of 5-9 bits. These converters were generally tested with the ADC and the matching DAC connected in a "back-to-back" fashion (forming a coder-decoder, or *codec*), since the performance of the combination was what determined overall system performance. Analog signals from analog test signal generators drove the ADC, and analog test equipment was used to measure the signal generated by the DAC. As will be discussed later in this section, "back-to-back" testing still has an important role up to 12-bits or so of resolution, particularly in preliminary evaluations of ADC performance.

The first high-performance general-purpose commercial data converters became available in the mid-1950s, pioneered by the 11-bit, 50-kSPS DATRAC vacuum tube converter designed by Bernard M. Gordon at Epsco in 1954. Gordon himself was a pioneer is defining the performance of data converters, especially those related to precision applications, and wrote many of the early articles on converter specifications (see References 1, 3, 4, 5, 9, 10, and 20).

The interest in ADCs and DACs increased rapidly in the 1960s, as solid-state data converters as well as mainframe computers became available. Early driving forces were data analysis, instrumentation, PCM, and radar applications.

In ac applications, there was still no way to directly measure the frequency-domain performance of so-called "sampling" ADCs because of the lack of low-cost memory and readily available digital computers. Any frequency-domain performance characteristics had to be measured using the "back-to-back" method and required that the reconstruction DAC had better static and dynamic performance than the ADC being tested.

By the mid-1970s, the mini-computer (such as the DEC PDP-series) made frequency domain testing using the fast Fourier transform (FFT) practical for ADC manufacturers (see Reference 13, for example). The IEEE-488 bus (initially the HP-IB) became a

convenient way to transfer data from a buffer memory containing the ADC samples to the computer for analysis.

Also during the 1970s, the use of ADCs and DACs in new applications such as digital video (References 17, 18) made application-specific testing a requirement.

The 1980s saw widespread growth in ac testing of ADCs (see References 19, 21, 22, 23, 24, 25, 27). Manufacturers began to standardize on ac specifications such as SNR, SINAD, ENOB, THD, etc., and these became integral parts of all sampling ADC data sheets. These specifications were vital to emerging applications in communications, where wide dynamic range was of utmost importance.

Some of the early pioneering work on ac specifications and testing in the 1980s was done by various IEEE committees involved in preparing ADC/DAC standards for digital video (Reference 28) and waveform recorders (References 22 and 36). The waveform recorder standard (IEEE Std. 1057-1994, R2001) was later expanded to include terminology and test methods for general purpose ADCs (Reference 37).

In addition to the evolution of ac tests, histogram tests for measuring the static DNL and INL performance of ADCs virtually replaced older methods (see References 21, 22, 24, 25, 27).

By the 1990s, frequency-domain testing of ADCs and DACs became the norm, and readily available FFT software, PCs, and manufacturer's evaluation boards placed it within easy grasp of most customers. Today (2004) nearly all sampling ADCs are fully characterized for ac performance, with the exception of high-resolution ADCs designed for measurement applications. Although there are still a few inconsistencies here and there in the industry, most ADC manufacturers have adopted basically the same set of specifications and related terminology as discussed in Chapter 2 of this book.

Static ADC Testing

As seen in Section 5.1, static testing of DACs basically consists of a series of measurements of the output voltage for various digital input codes. Knowledge of the specific DAC architecture and the corresponding error characteristic may allow a reduction in the actual number of individual voltage measurements, however, this only serves to speed up test time, and doesn't change the fundamental test method or concept.

In a DAC, there is one unique output voltage for each digital input code, regardless of DNL or INL errors. An ADC, on the other hand, does not have a unique voltage input corresponding to each output code—there is a small input voltage range equal to 1 LSB in width (for an ideal noiseless ADC) that will produce the same digital output code. This is called the *quantization uncertainty*, and it can be the source of confusion when specifying and measuring ADC static transfer characteristics. Figure 5.26 shows two possible methods for defining the relationship between the ADC analog input and the digital output code. Method A defines the static transfer characteristic in terms of the *code centers*, however, there is no direct way to measure these points because of the quantization uncertainty. Method B defines the static transfer characteristic in terms of the code transitions, which can be measured directly. All that is required to measure the

code transitions is an analog voltage source and DVM connected to the ADC input and a means of observing the digital outputs, such as an LED display. The analog input is varied until the LEDs "flutter" between two codes, and the input voltage is recorded. It should be noted, however, that this method only works well if the ADC (and the input voltage source) has an effective peak-to-peak input-referred noise which is less than 1 LSB. Larger amounts of input-referred noise tend to mask the transitions and make the measurement increasingly difficult.

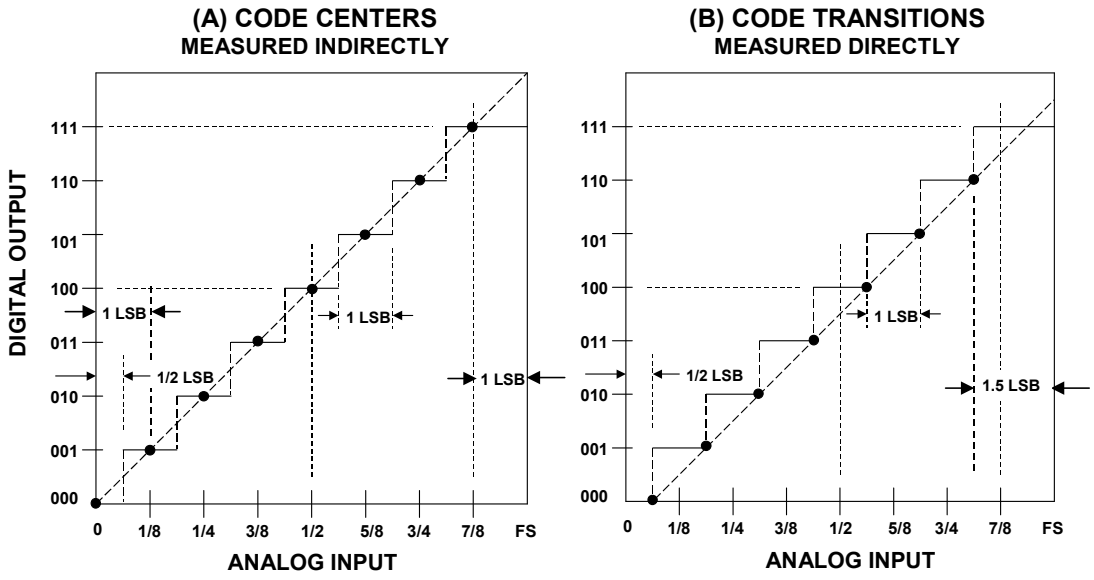

Figure 5.26: *Measuring ADC Code Transistions to Determine Code Centers*

However, assume for the moment that the ADC is relatively noise-free and that the transitions can be easily measured. Once the code transition points are known, the code center can be calculated as the voltage which is halfway between the corresponding code transitions. But in fact, the entire ADC transfer function can be defined entirely by the code transitions as shown in Figure 5.26B. Note that the code transitions points are shifted ½ LSB to the left of the code centers for the ideal ADC. The advantage of using the code transition method directly is that the DNL for a particular code is simply the difference between the corresponding code transitions.

The code-center method can lead to misleading results as shown in Figure 5.27. Notice that the ADC transfer function has alternating wide and narrow codes, but the line drawn through the endpoints of the code centers indicates perfect INL. On the other hand, the line drawn through the endpoints of the code transitions shows the true INL of ½ LSB.

Figure 5.28 shows a simple test setup for measuring the code transition points of an ADC. The analog input is driven from a precision low-noise voltage source, and the ADC digital outputs are observed using an LED display. Displaying the true and complement of each ADC bit makes determining the precise centers of the code transition points easier by simply adjusting the analog input voltage for equal brightness between the true

and complementary LED display for the bits which are "fluttering." Additional logic is obviously required if the output of the ADC is in serial format.

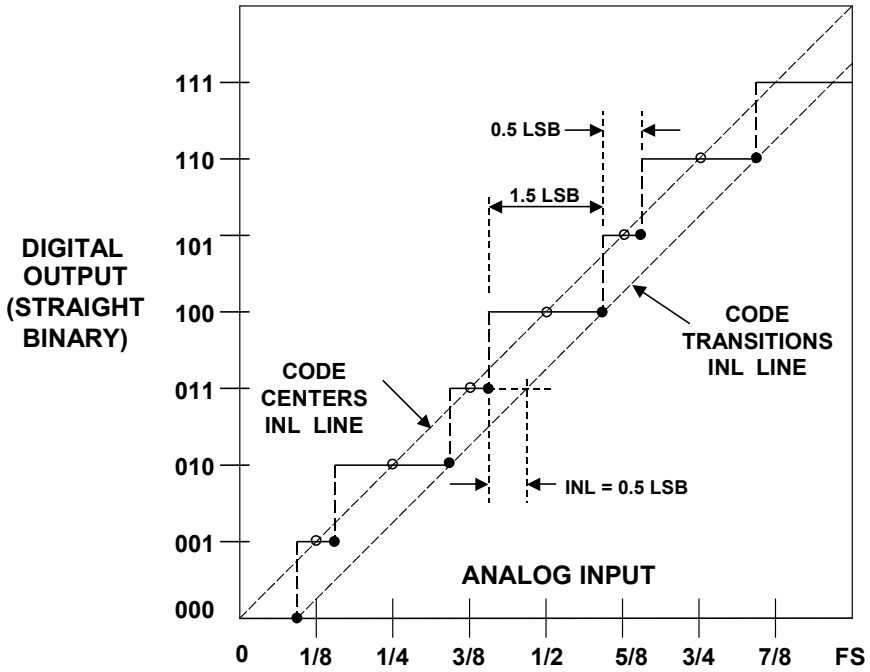

Figure 5.27: *Code Transitions Preferable to Code Centers for Measuring ADC DNL and INL*

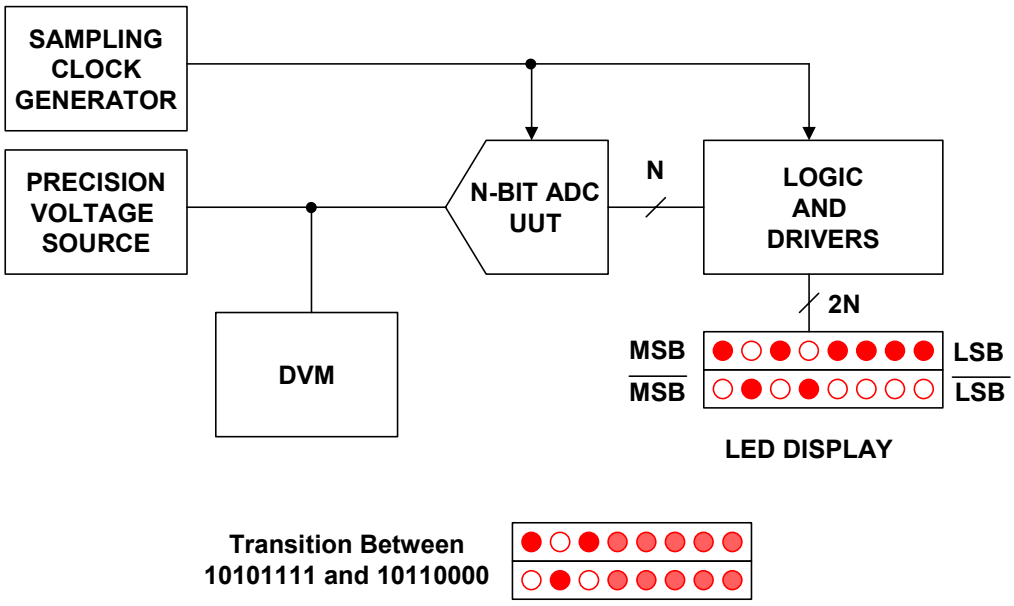

Figure 5.28: *Simple Test Setup for Measuring ADC Code Transitions*

Although quite simple, the manual test setup is very useful with low noise ADCs, especially if only a few code transitions need to be measured, such as when determining offset and gain errors. Figure 5.29 shows the details for measuring ADC gain and offset

in terms of the code transitions. The example is for a 3-bit ADC, but is applicable to any resolution.

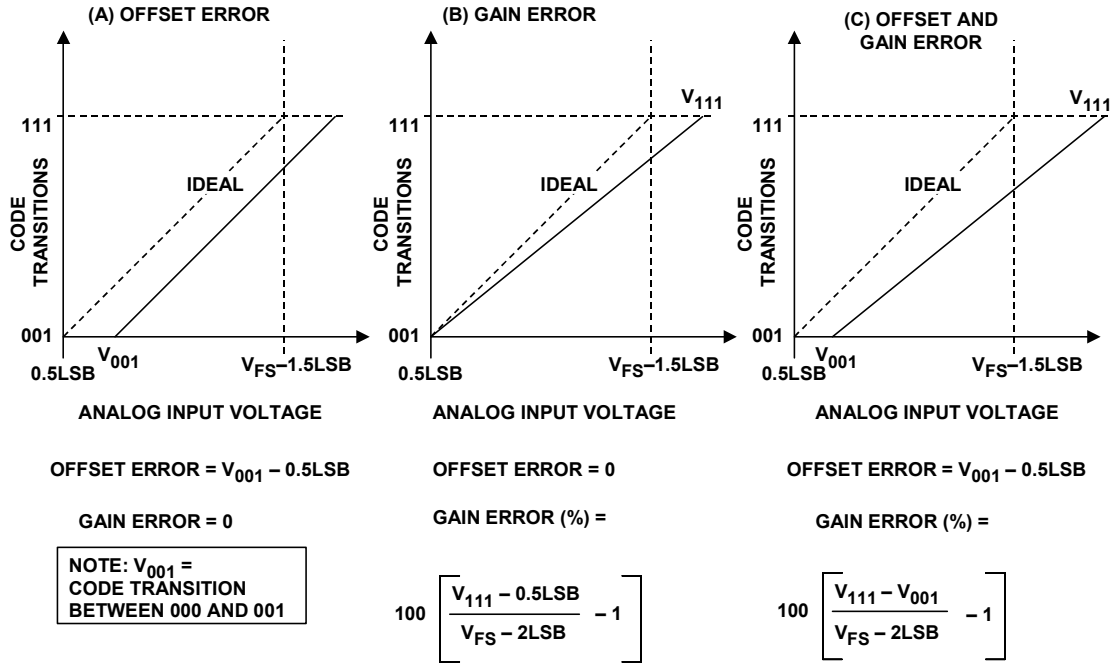

Figure 5.29: *Measuring ADC Offset and Gain Error*

Notice that for an ideal ADC, the first code transition between 000 and 001 (designated V_{001}) occurs at an analog input voltage of 0.5 LSB and the last transition between codes 110 and 111 (designated V_{111}) occurs at a voltage of FS – 1.5 LSB, following the conventions shown previously in Figure 5.26B. The definitions in Figure 5.29 apply to a unipolar ADC but can be easily modified for a bipolar ADC. In a bipolar ADC (as discussed in Chapter 2 of this book) the convention is that a zero-volt input to the ADC should fall at the code center corresponding to the code 100…0, with adjacent code transitions 0.5 LSB above and below zero-volts. As in the case of DACs, the value of the ADC bipolar zero may be of interest and is easily calculated from the corresponding code transition values.

Back-to-Back Static ADC Testing

Another useful static ADC test method that dates back to the early days of data converters involves connecting a DAC to the output of the ADC and measuring the performance of the ADC using conventional analog test methods. The success of this "back-to-back" method depends upon the ability to obtain a suitable DAC which has an accuracy significantly greater than the ADC under test. For instance, a 12-bit ADC requires at least a 14-bit accurate DAC, even though only 12 bits of the DAC are actually used.

Figure 5.30 shows a back-to-back test setup for measuring the static performance of the ADC by generating the actual error waveform for the ADC transfer function. The ADC drives an accurate DAC, and the output of the DAC is subtracted from the analog input to the ADC, amplified (if required), and filtered for observation with an oscilloscope. The

subtraction function is implemented by complementing the ADC output code before driving the DAC, so that the DAC output is an inverted quantized representation of the ADC input. A simple resistive summer using a potentiometer can then be used to combine the signals. A very low-frequency linear ramp is applied to the ADC, and the potentiometer adjusted to null out the effects of gain differences between the ADC and the DAC. A linear ramp input makes the interpretation of the error waveform easier because the horizontal axis (time) can be calibrated directly in LSBs. If desired, the DAC update clock can be a sub-multiple of the ADC sampling clock to ease the settling time requirements on the DAC and make the error waveform easier to observe. A lowpass filter at the input of the oscilloscope is useful for removing high frequency components which may obscure the actual error waveform.

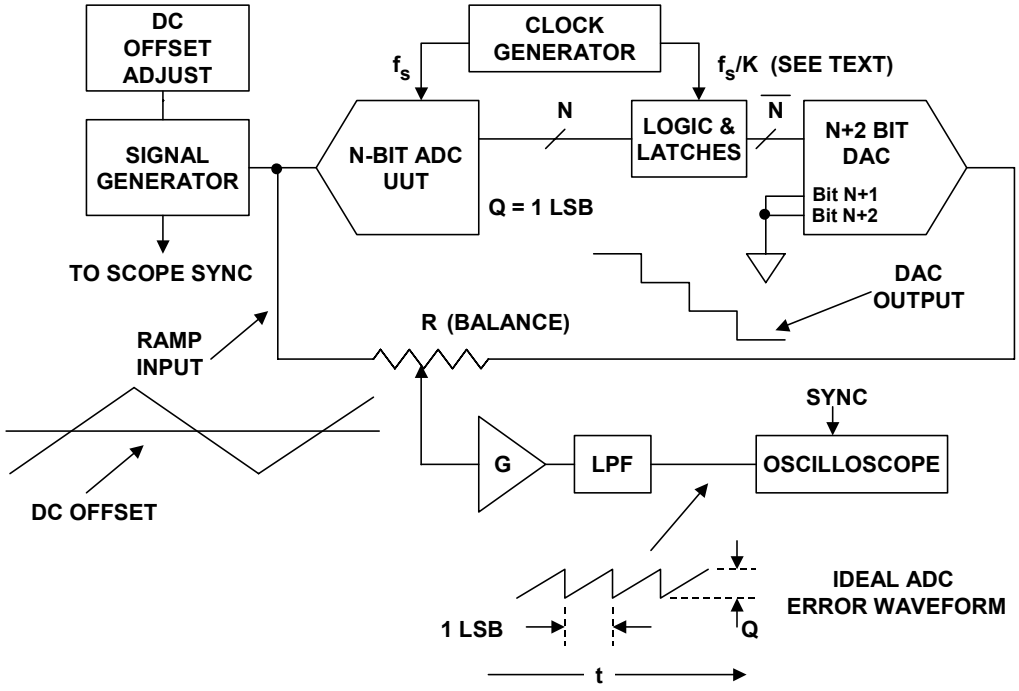

Figure 5.30: *ADC/DAC Back-to-Back Static Testing*

Figure 5.31A, B, and C show some typical error waveforms which result from various errors in the ADC. Notice that the sawtooth error waveform is composed of a number of "teeth". The width of the tooth corresponds to the code width and can be used to make DNL measurements directly as shown in Figure 5.31A. The dotted vertical lines along the horizontal axis show where the ideal code transitions should occur, and the effects of positive and negative DNL are shown as well as a missing code.

The height of each tooth is determined by the DAC, and if there is a missing code, the height will correspond to 2 LSBs rather than 1 LSB. Figure 5.31B shows a non-monotonic condition, represented by a temporary reversal in the direction of the error waveform indicated by the arrows.

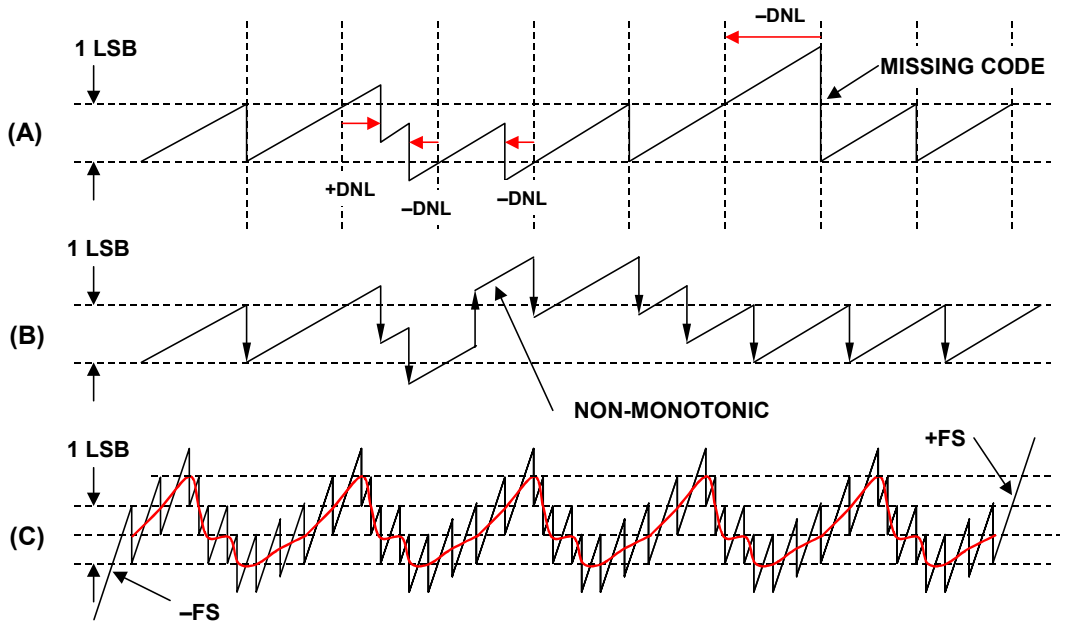

Figure 5.31: *ADC Error Waveforms Using "Back-to-Back" Test Setup*

When the "back-to-back" method is used to make static DNL measurements, a low amplitude triangle waveform with a precision dc offset adjustment works best. The dc offset can be slowly adjusted, and the DNL error waveform swept across the entire range of the ADC. If the frequency of the error waveform is too high, the error components cannot be observed. INL measurements, on the other hand, should be made with a full-scale triangle, and the maximum deviation of the error waveform from a line connecting the endpoints can be used to calculate the INL as shown in Figure 5.31C. In this case, it is not necessary to be able to observe each of the individual teeth that make up the overall error waveform. The balance potentiometer should be adjusted to force the endpoints to the same level so that the INL can be measured. The worst case endpoint INL is approximately +1 LSB in Figure 5.31C.

In summary, the "back-to-back" method is a useful tool in making static ADC linearity measurements in a simple bench test setup. As the ADC resolution is increased, the frequency of the input signal must be made lower, the amplitude of the error waveform decreases, and the effects of ADC noise and DAC errors become more pronounced.

The technique works best for ADCs of 12-bits of resolution or less, and where the code transition noise (i.e., input-referred noise) is less than a few tenths of an LSB. Measurements on 12-bit ADCs require considerable care, and extending the technique to 14-bit ADCs is quite difficult. Measurements on 16-bit ADCs require at least 18-bit accurate DACs, which are not widely available with traditional dc specifications.

Crossplot Measurements of ADC Linearity

The crossplot method for measuring ADC linearity was developed in the early days of data converters as an easy method for determining integral and differential linearity. This

test is suitable for quick evaluations where a high degree of precision is not required. The concept is nearly the same as the "back-to-back" method previously described, except that only two or three of the ADC LSBs are converted back into analog format. Figure 5.32 shows the basic test setup for the crossplot test.

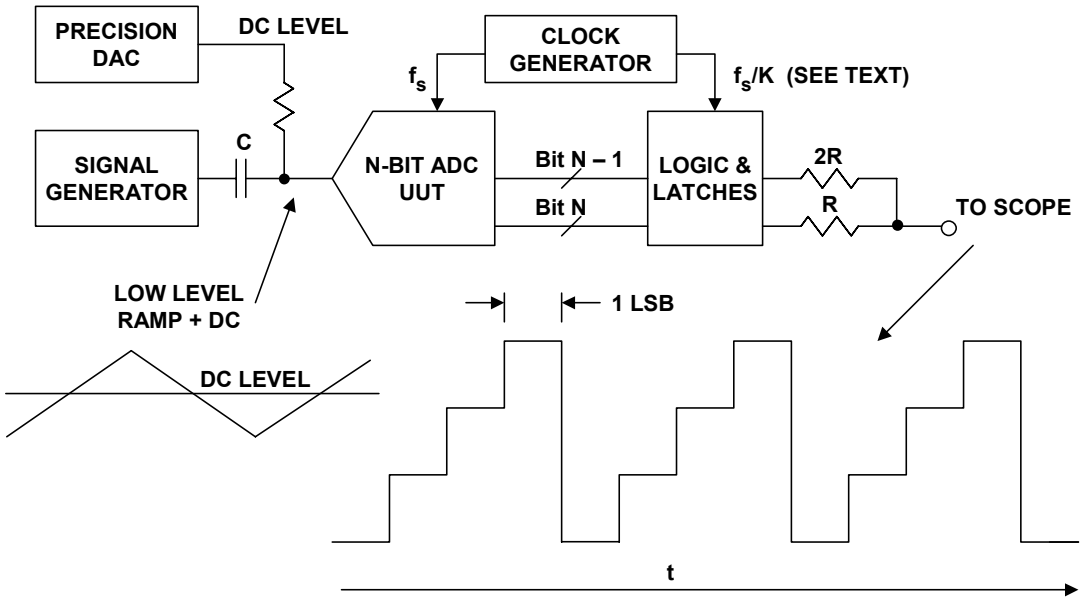

Figure 5.32: *ADC Crossplot Test*

Two resistors are used for the simple DAC to reconstruct the two LSBs of the ADC output. A precision DAC supplies the dc input to the ADC, and a low-amplitude triangle waveform serves to modulate the dc level. As in the "back-to-back" test method, the horizontal axis of the oscilloscope can be calibrated in terms of LSBs. The output of the 2-bit DAC is a four-level staircase waveform as shown. The crossplot method was initially developed to test successive approximation ADCs designed around internal binary-weighted DACs.

As previously discussed in this chapter, superposition holds in this type of DAC, and therefore in a SAR ADC based upon such a DAC. All codes need not be checked, only the major carries. In conducting the test, the precision DAC supplies the dc level corresponding to the particular major-carry code transition under test. The horizontal axis of the oscilloscope is calibrated in terms of LSBs. The three waveforms shown in Figure 5.33 illustrate DNL errors and a missing code error. Notice that this method will not accurately detect more than two missing codes above and below the major carry under test. Also, the simplicity and efficiency of the crossplot technique is lost when testing ADCs of architectures other than successive approximation (using binary-weighted internal DACs), as more codes must be tested.

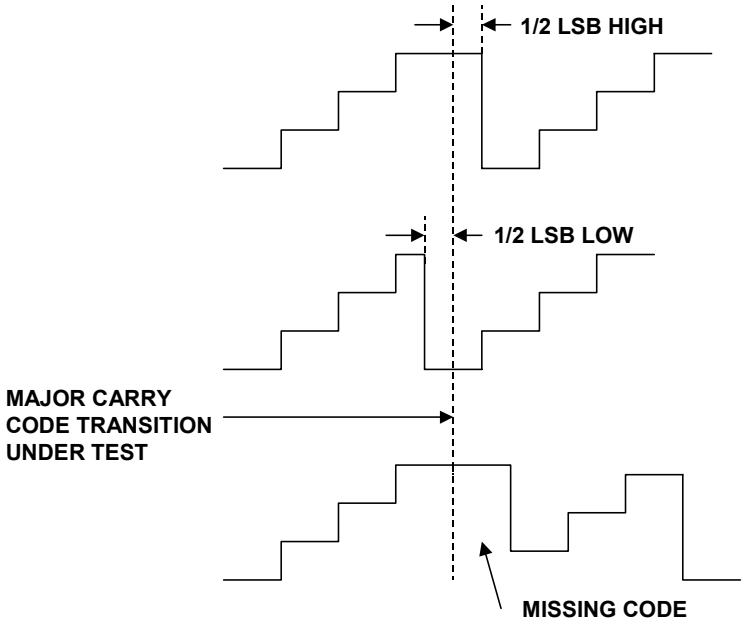

Figure 5.33: *Crossplot Waveforms Showing ADC Static Errors Around Major Code Transitions*

Servo-Loop Code Transition Test

The servo-loop code transition test setup shown in Figure 5.34 lends itself to automated measurements, either as part of ATE systems or in PC-based controllers. The loop begins with an op amp configured as an integrator. Switches (usually CMOS) at the integrator input are used to select between positive and negative voltage sources. With a constant dc voltage at the integrator input, a linear ramp will be generated which is sampled by the ADC. The output of the ADC goes to the "A" input of a digital comparator. The "B" input to the digital comparator specifies the code transition to be measured. If the ADC output is less than the selected code, the A < B comparator output connects the negative voltage source to the resistor R, and the output of the integrator ramps up. At the point when A ≥ B, the positive voltage source is connected to the resistor R, and the output of the integrator ramps down.

The feedback of the servo loop causes this oscillation to continue, producing a triangle waveform which is centered about the dc level of the code transition. The voltmeter at the integrator output provides an accurate measurement of the code transition voltage. When the loop time constant is properly adjusted, the amplitude of the triangle waveform should ideally be a fraction of an LSB.

This technique can be used to measure the endpoints of the transfer function, i.e., the first and last code transitions, V_1, V_{N-1}. These values can then be used to calculate the nominal LSB value as follows:

$$LSB_{NOM} = \frac{V_{N-1} - V_1}{2^N - 2} \qquad \text{Eq. 5.14}$$

The endpoint code transitions and the nominal LSB weight provide the required information to calculate DNL and INL for any desired code transition. Gain and offset errors can also be calculated from the endpoint code transitions.

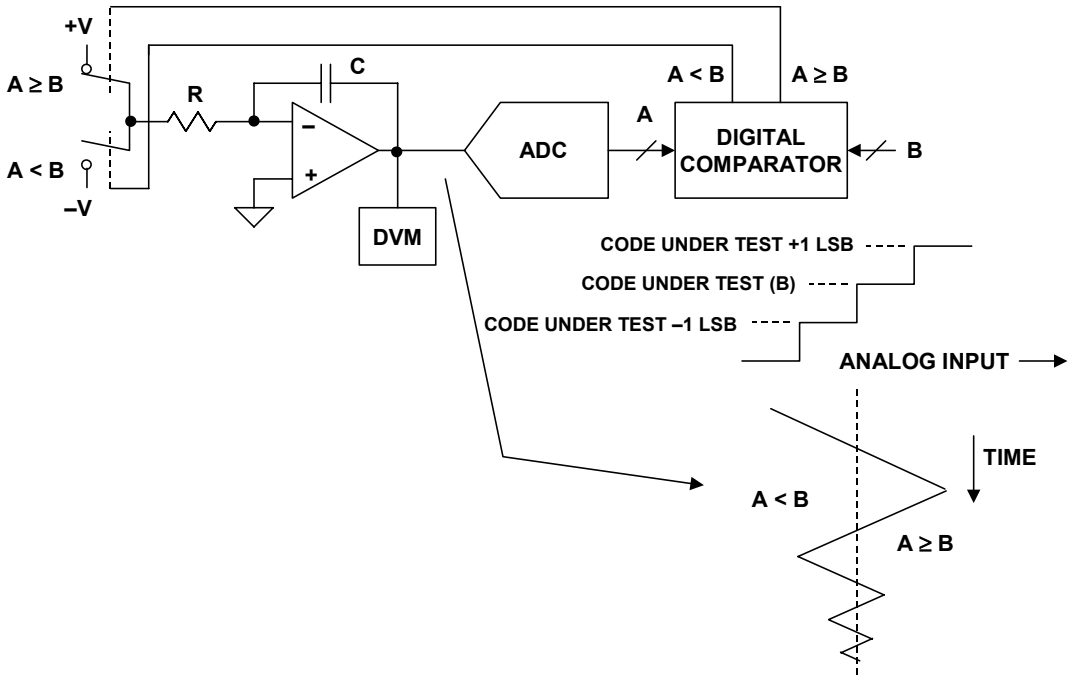

Figure 5.34: *Integrating Servo-Loop ADC Tester*

Hysteresis and/or noise in the ADC can create problems in the servo-loop test method. In some cases, it is beneficial to have control of the integrator time constant, starting out with a short time constant for fast initial response, and ending with a longer one for more accuracy.

Computer-Based Servo-Loop ADC Tester

Figure 5.35 shows a generalized computer-based ADC servo-loop tester suitable for an ATE or PC-based system. This configuration allows the ultimate flexibility by allowing complete software programmability. It also allows the use of averaging techniques to reduce the effects of ADC input-referred noise. It is suitable for more dedicated setups where high volume testing is required, but is most likely not justifiable for simply evaluating ADC performance on the bench.

All of the test methods described thus far for determining the code transitions work best when the ADC under test has input-referred noise less than a few tenths of an LSB peak-to-peak, such as shown in Figure 5.36. The computer-based servo-loop tester of Figure 5.35 is the exception, because it allows for data averaging to remove the effects of excess noise. Some test algorithms which work perfectly for low-noise ADCs simply may not converge for high levels of input noise as shown in Figure 5.37.

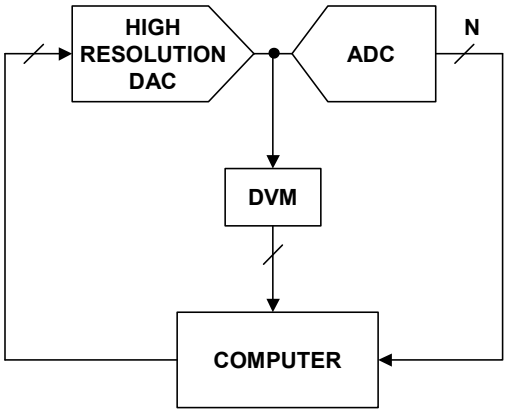

Figure 5.35: *Generalized Computer Controlled Servo-Loop ADC Tester*

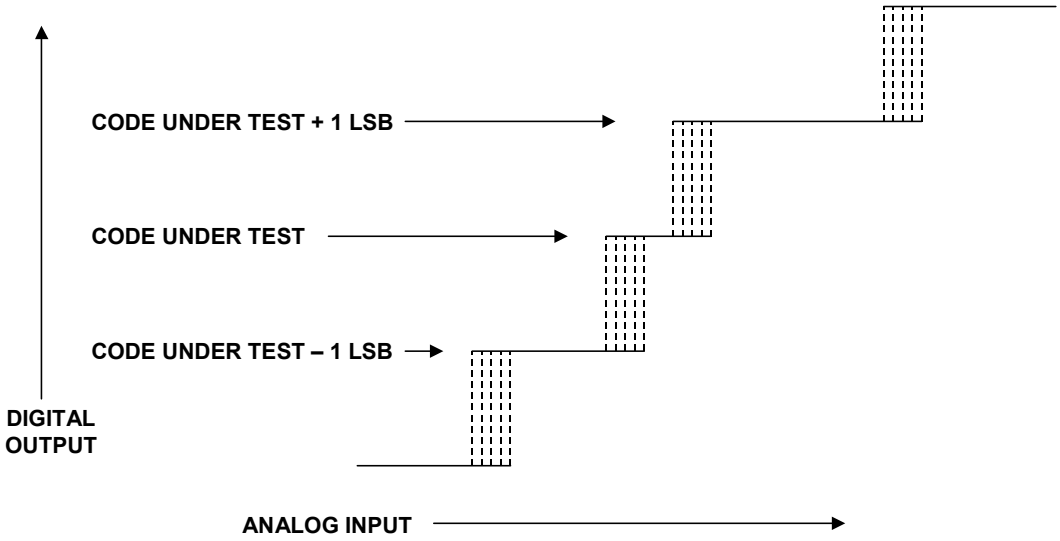

Figure 5.36: *ADC Transfer Function With Relatively Low Input-Referred Noise*

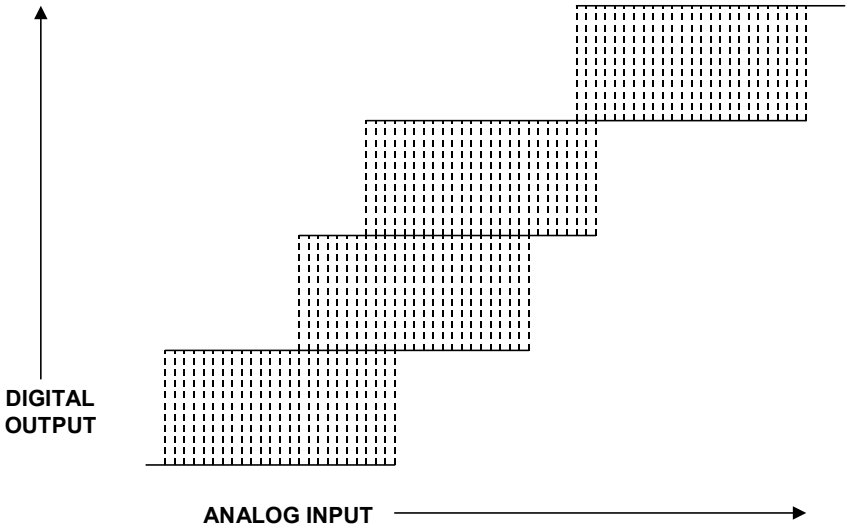

Figure 5.37: *ADC Transfer Function With Relatively High Input-Referred Noise*

As ADC technology has evolved over the years, code transition noise has rarely been a problem with 6-, 8-, or 10-bit converters. Even at the 12- and 14-bit level, the effective input bandwidth of early ADCs in the 1970s was low enough so that input-referred noise was still at a level which allowed the "back-to-back," crossplot, and integrating servo-loop test methods to work satisfactorily for most converters.

In the 1980s and 1990s, however, the requirements for increasingly higher sampling rate ADCs with associated higher input bandwidth has led to higher input-referred noise, simply because of the fundamental physical laws governing circuit designs such as resistor noise and KT/C noise. Although low input-referred noise 12-, 14-, 16-, and 18-bit ADCs are available today for low sampling rate, low bandwidth applications; many of the wide bandwidth ADCs for communications applications have peak-to-peak input-referred noise which often exceeds 1, 2, or more LSBs. These ADCs are used in applications where SFDR is actually the performance-limiting specification—process gain due to oversampling and averaging techniques are used to reduce the effects of random noise

Because of the need for a high speed automated test method suitable for measuring DNL and INL of all ADCs, regardless of noise, the histogram (code density) test method (described in the next section) is by far the most popular today.

Histogram (Code Density) Test with Linear Ramp Input

Histogram testing of an ADC involves collecting a large number of digitized samples over a period of time, for a well-defined input signal with a known probability density function. The ADC transfer function is then determined by a statistical analysis of the samples. For example, a linear ramp (in actual practice, a triangular waveform is used) which slightly exceeds both ends of the range of the ADC is a popular histogram test signal. A large number of samples are collected for the triangular waveform input, and the number of occurrences of each code are tallied. If the ADC has no INL or DNL errors, all codes have equal probability of occurrence (with the exception of the end-point all "0"s and all "1"s codes), and there should be the same number of counts in each code bin.

Figure 5.38 shows a typical histogram test setup. A linear triangular waveform which slightly overdrives the ADC is applied. The frequency of the waveform should be low enough such that the ADC does not make ac-related errors, and the frequency must not be sub-harmonically related to the sampling frequency. A total of M_T samples are collected for codes 1 to $2^N - 2$. Notice that the "overflow" counts that fall in the all "0"s bin (code 0) and the all "1"s bin (code $2^N - 1$) are not included in the M_T total but still add to the total number of samples required. Therefore, the triangular waveform should be adjusted such that the number of overflow hits is no larger than needed to ensure that the ADC is sufficiently overdriven and that the portion of the waveform within the ADC range is linear to the required accuracy (10% overdrive is reasonable for most ADCs).

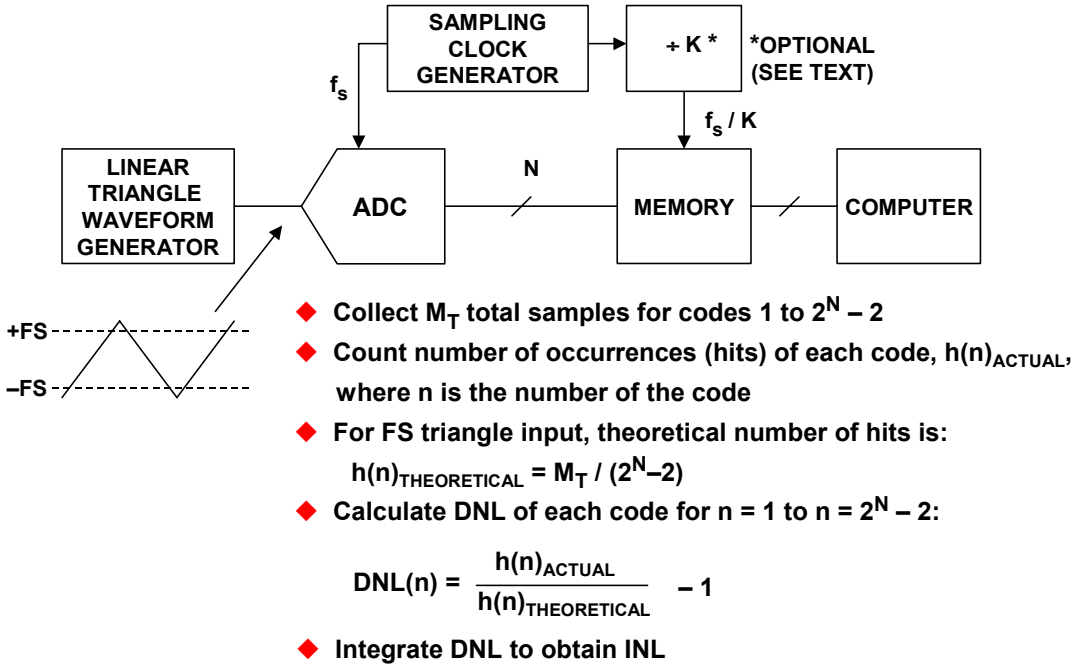

Figure 5.38: *Histogram (Code Density) Test Setup*

The number of occurrences ("hits"), h(n), in each code bin, n, are then recorded for n = 1 to n $= 2^N - 2$. The theoretical number of hits in each bin (assuming perfect INL and DNL) is simply $h(n)_{THEORETICAL} = M_T/(2^N - 2)$. If $h(n)_{ACTUAL}$ is the actual number of hits in a bin, then the DNL of that particular code is given by:

$$DNL(n) = \frac{h(n)_{ACTUAL}}{h(n)_{THEORETICAL}} - 1. \qquad \text{Eq. 5.15}$$

Figure 5.39 shows a typical display of the histogram data. Wide codes, narrow codes, and missing codes are easily spotted in the display. The actual DNL for each code is easily calculated from the histogram data using Eq. 5.15. Once the DNL is calculated, the INL is simply the integral of the DNL as shown in Figure 5.40.

It should be obvious that the histogram test eliminates the effects of input-referred noise by averaging it over all the code bins. The noise and hysteresis associated with each individual code transition is also averaged. Therefore, the histogram test is ideally suited for modern wide bandwidth high precision ADCs and is universally accepted, especially with the proliferation of standard PC-based software and ADC manufacturer's evaluation boards.

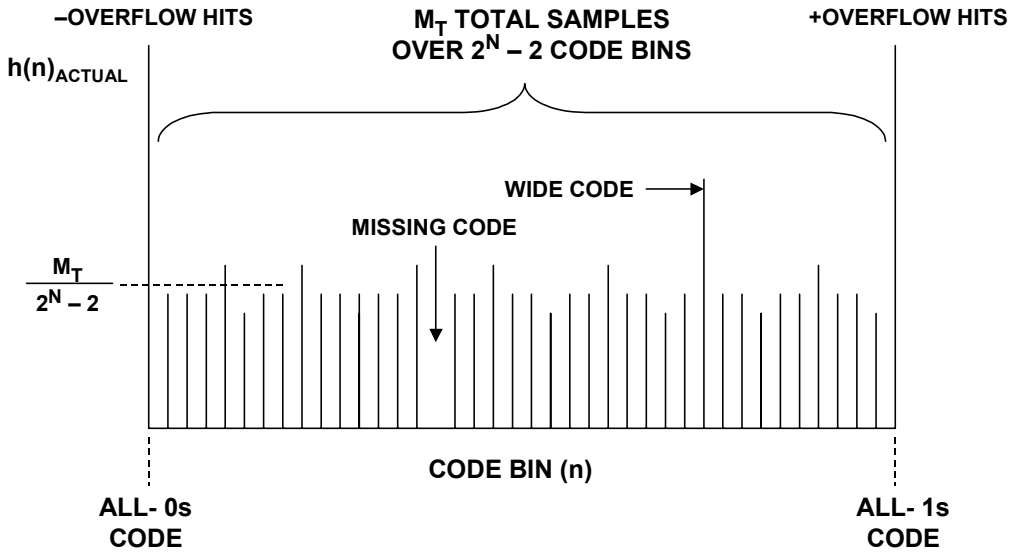

Figure 5.39: *Histogram for Linear Ramp Test*

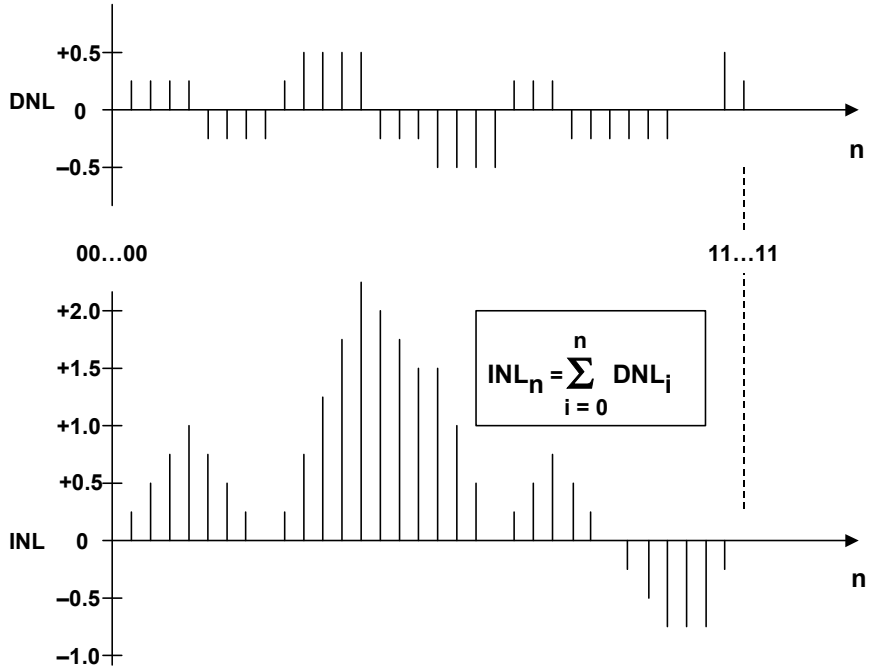

Figure 5.40: *Integration of DNL Data Yields INL*

Notice, however, that the histogram test alone does not necessarily imply monotonicity in an ADC, i.e., the order in which the codes occur with respect to the input cannot be determined directly. However, a non-monotonic ADC will also generally have a higher level of distortion, and this condition is easily detected with an FFT analysis of the output data. Since histogram and FFT tests use essentially the same hardware, both are normally part of a comprehensive ADC test plan.

There are several important factors to consider when conducting histogram tests. One of the most important is the number of samples required to accurately measure DNL and

INL. Assume a 12-bit, 1-MSPS ADC, and that it is desired to obtain an average of 20 hits in each code bin. With 20 hits per code bin, a DNL resolution of 1/20 = 0.05 LSB is possible. This implies that the input ramp should sweep through all of the 2^{12} = 4,096 levels, dwelling on each code long enough to produce 20 hits per level. The total number of samples required is therefore $M_T = 20 \times 4{,}096 = 81{,}920$. Since the sampling frequency is 1 MSPS, this implies that the ramp must make a full-scale transition in 82 ms in order to obtain 20 hits per code bin. This assumes no overhead for the overflow samples.

Now assume that the ramp is generated by an ideal 16-bit DAC. This means that the DAC divides each of the 12-bit bins into 16 levels, or 1/16 = 0.06 LSB. The total uncertainty in measuring the DNL is therefore 0.05 LSB (20 hits per bin) plus 0.06 LSB (16-bit ramp generator DAC), for a measurement uncertainty of 0.11 LSB.

Generating the linear ramp using a DAC becomes impractical for testing ADCs of greater resolution than 12 bits because of the difficulty of designing highly accurate relatively fast settling DACs with greater than 16-bit resolution. In addition, the samples that occur during the DAC settling time must be ignored, thereby further complicating the test. A much more practical method is to use a linear triangular wave function generator with an output frequency which is not a sub-harmonic of the ADC sampling clock frequency. In this case, determining the required number of samples is a relatively complex statistical problem involving confidence levels, probabilities, etc. It is beyond the scope of this discussion to get into the details of the statistics, but the reader is referred to References 27, 37, and 38 for details. Fortunately, the accuracy of the test setup can be empirically verified by simply examining the repeatability of the measurements on several records of data, and then making the appropriate adjustments.

It was mentioned that the frequency of the triangular wave input must not be a sub-harmonic of the ADC sampling clock, otherwise there will be repetitive code patterns within each input cycle which will skew the DNL data and render it meaningless. This artifact is identical to that discussed in Chapter 2 of this book regarding quantization noise. For the histogram test to give accurate results, the quantization noise must be random. If the input frequency is a sub-harmonic of the sampling frequency, the quantization noise error signal can be periodic, thereby indicating false missing codes and large DNL errors. In most cases, jitter and noise associated with the sampling clock, analog input, and ADC tend to mitigate this effect somewhat in a practical test setup, but the integrity of the histogram measurements should be verified by slightly varying either the sampling clock frequency or the input frequency and making sure the DNL data remains relatively constant.

Finally, the histogram can place constraints on the buffer memory in terms of the speed required to handle the data output from fast ADCs as well as memory size to handle the large number of samples required (possibly several hundred thousand). The speed requirement can be reduced by taking every K^{th} sample from the ADC, since it is not required that the samples be contiguous. This in itself does not reduce the total number of samples required—it simply eases the memory speed requirement at the expense of a longer test time.

The triangular histogram test will give good DNL measurements at 16-bit or higher resolutions if proper precautions are taken as described above. INL measurements, on the

other hand, can be no more accurate than the INL of the triangular input waveform. Maintaining suitable waveform INL can be a problem at 12-bit and higher resolutions. In addition, any high frequency noise which may be present on the triangular waveform cannot be removed by filtering, because that will also affect the waveform linearity.

For these reasons, a sinewave rather than a triangular waveform is often used as an input to the ADC when making histogram DNL and INL measurements. Sinewaves can be generated with extremely high linearity and low noise with appropriate filtering. However, unlike the triangular wave, the sinewave input does not yield an equal probability for all codes. It can be shown (see References 21, 24, 27, 32, 36, 37, and 38) that for an N-bit ADC with a full-scale input range equal to $\pm V_{FS}$, and an input sinewave of amplitude A, the probability of occurrence of code n is given by:

$$p(n) = \frac{1}{\pi}\left[\sin^{-1}\left\{\frac{V_{FS}\left(n - 2^{N-1}\right)}{A \cdot 2^{N}}\right\} - \sin^{-1}\left\{\frac{V_{FS}\left(n - 1 - 2^{N-1}\right)}{A \cdot 2^{N}}\right\}\right]. \qquad \text{Eq. 5.16}$$

This equation is plotted in Figure 5.41. Notice that the probability of occurrence increases at the peaks of the sinewave near $\pm V_{FS}$ because the dv/dt is less (more hits per bin) than at the zero-crossing where the dv/dt is the highest (fewer hits per bin).

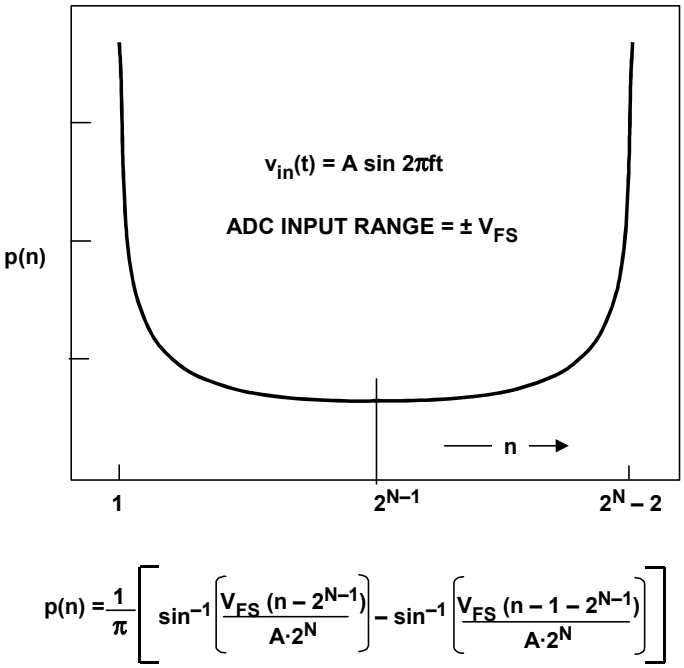

Figure 5.41: *Sinewave Probability Density Function*

With a sinewave input, the theoretical number of hits for the n^{th} code is given by:

$$h(n)_{THEORETICAL} = p(n)M_T . \quad \text{Eq. 5.17}$$

The corresponding DNL error for that code is given by:

$$DNL(n) = \frac{h(n)_{ACTUAL}}{p(n)M_T} - 1 . \quad \text{Eq. 5.18}$$

There are several precautions which must be taken in order to obtain accurate results using the sinewave histogram test. As previously discussed for the triangular test waveform, the sinewave frequency must not be a sub-harmonic of the sampling frequency. The amplitude of the sinewave input, A, should be chosen such that the ADC is slightly overdriven at both ends of its range. The effects of dc offset should then be removed by adjusting the offset of the sinewave such that there are an equal number of hits above and below the mid-scale point, i.e. the number of hits from code 0 to code $2^{N-1} - 1$ should equal the number of hits from code 2^{N-1} to code $2^N - 1$:

$$\sum_{n=0}^{2^{N-1}-1} h(n) = \sum_{n=2^{N-1}}^{2^N-1} h(n) . \quad \text{Eq. 5.19}$$

Finally, the value of A should be estimated using the actual histogram data from the following equation:

$$A_{ESTIMATE} = \frac{V_{FS}}{\sin\left[\frac{M_T}{M_T + h(0) + h(2^N - 1)} \cdot \frac{\pi}{2}\right]} . \quad \text{Eq. 5.20}$$

The estimated value of A predicted by Eq. 5.20 should then be used in Eq. 5.16 for calculating the p(n) values for each code. It is important that these steps be taken to account for the ADC gain and offset in the actual test setup so that accurate values of p(n) can be obtained for the final DNL calculation using Eq. 5.18.

Typical sinewave histogram DNL and INL measurements on the AD9236 12-bit 80-MSPS ADC are shown in Figure 5.42 as an example of how this data typically appears on an ADC data sheet.

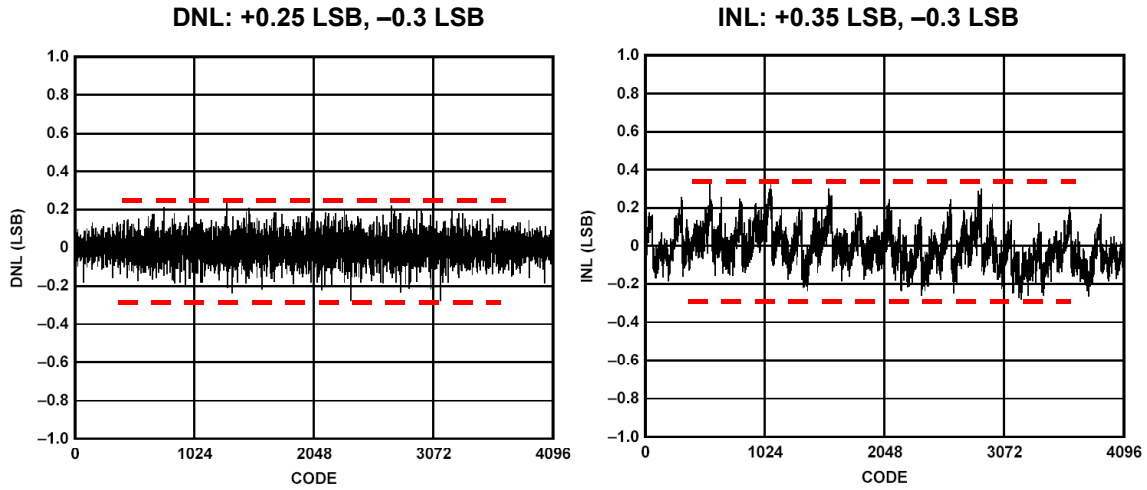

Figure 5.42: *Typical Static DNL and INL Histogram Plots for AD9236 12-Bit, 80-MSPS ADC*

Another useful application of the histogram test method is measuring the ADC input-referred noise. This is easily accomplished by simply terminating the ADC input with the appropriate resistance, applying a dc input (the actual value is not critical, but a voltage near mid-scale can be used for convenience), and recording a number of output samples. If the peak-to-peak input referred ADC noise is less than 1 LSB, the output samples should all correspond to a single code value. If the dc input happens to fall exactly on a code transition, the samples will be divided between two adjacent codes—however, the dc input can be offset by ½ LSB in order to produce a single code at the output. The effect of input-referred noise is to spread the output samples over a number of code bins as shown in Figure 5.43. Assuming the noise is Gaussian, the input-referred noise is simply the standard deviation (σ) of the distribution, and is generally expressed in LSBs rms.

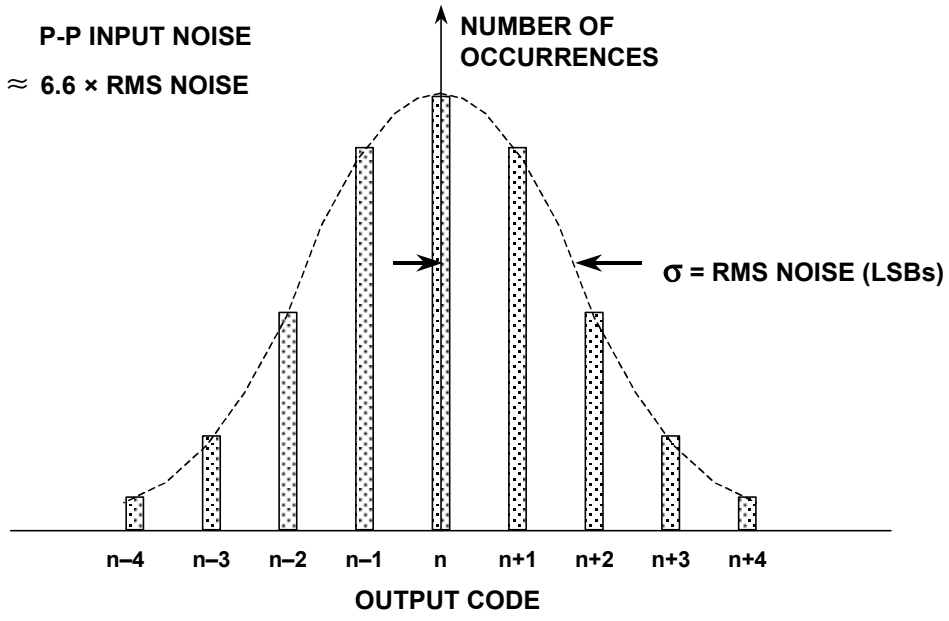

Figure 5.43: *Measuring Input-Referred Noise Using "Grounded Input" Histogram*

Dynamic ADC Testing

The vast majority of integrated circuit ADCs today are *sampling* ADCs that contain an internal sample-and-hold function of one form or another. In addition to the traditional dc specifications, sampling ADCs are generally fully specified in terms of ac performance characteristics such as SINAD, ENOB, SNR, SFDR, etc. In the early 1970s, the vast majority of ac testing was performed using the "back-to-back" test method where a high performance DAC was used to reconstruct the ADC output, thereby allowing the use of traditional analog test equipment. In the mid-1970s, DSP-based testing of ADCs began to evolve—today, practically all ADC testing is performed using some type of digital analysis of the ADC output data. While manufacturer-supplied evaluation boards, along with PC-based software packages, have placed these digital techniques within the grasp of most serious ADC users, examination of some of the older methods is still a worthwhile exercise.

This section discusses some of the traditional "back-to-back" test methods which still have application today in bench test setups, as well as the newer DSP-based methods. Issues relating to signal generation are also discussed where applicable.

Manual "Back-to-Back" Dynamic ADC Testing

A typical "back-to-back" ADC ac test setup is shown in Figure 5.44. As in the case of static testing using this method, the key to its success is finding a DAC which has at least 2-bits better dc and ac performance than the ADC under test. In the 1970s and 1980s this was generally not too much of a problem, because 12-bit fast settling DACs with relatively low distortion were generally available—sometimes an external deglitcher was required to obtain sufficiently low levels of distortion.

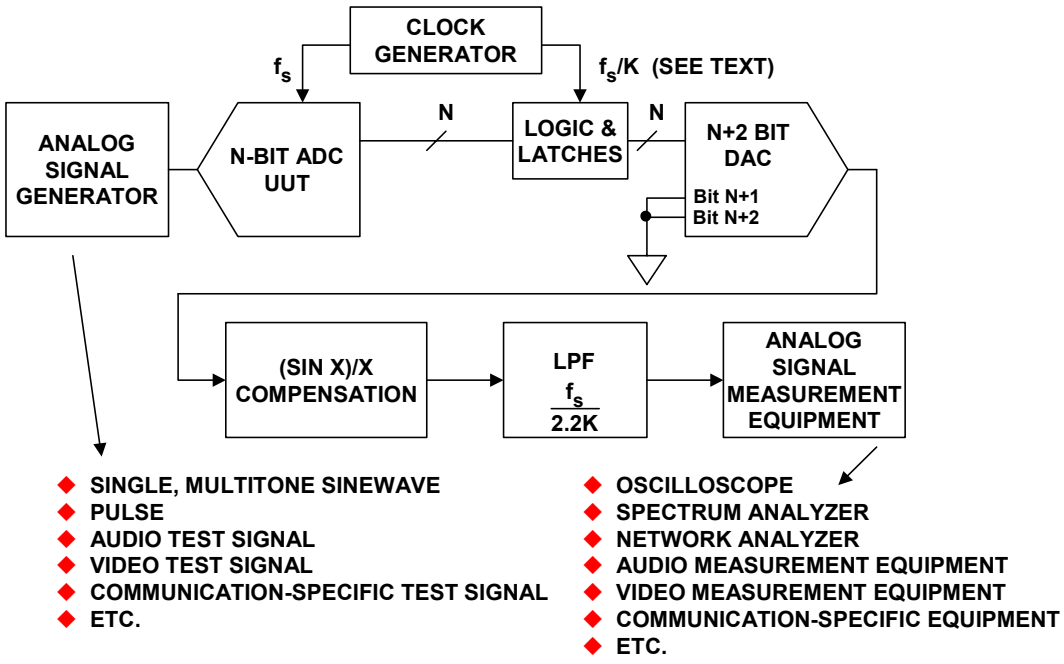

Figure 5.44: *"Back-to-Back" Setup for ADC Dynamic Testing*

The back-to-back method was widely used with 8- and 10-bit high-speed ADCs. Most manufacturer's ADC evaluation boards during the period contained a reconstruction DAC which in most cases was sufficient to perform some basic ac tests. Today, however, if a reconstruction DAC is included on manufacturer's ADC evaluation board, it is generally there to facilitate simple functionality tests—the actual ac evaluation of the ADC should be performed using DSP techniques.

Back-to-back testing is still relevant today in applications where the signal is digitized, processed, and converted back to analog form, such as in audio codecs. In these cases, the back-to-back performance of the ADC/DAC combination is what determines overall system performance, and it is not as important to know exactly how the ac errors are divided between the ADC and the DAC. In these applications, the analog test methods previously described for measuring SNR, SINAD, THD, SFDR can be utilized.

Today, there are many high-performance high-speed ADCs for which selecting ac-compatible DACs suitable for testing the ADC is extremely difficult. This is especially true at the 12-, 14-, and 16-bit resolution level in communications applications. In many cases, the closest match will be a DAC with complete ac specifications suitable for DDS applications, such as the TxDAC® series.

Returning to Figure 5.44, the output of the reconstruction DAC is a series of rectangular pulses whose widths equal the reciprocal of the sampling frequency. As discussed previously in Chapter 2 of this book, the frequency response of this signal follows a (sin x)/x function, therefore system measurements that depend on frequency response must take into account this theoretical rolloff, or include a suitable compensating filter as shown.

In many cases, it is useful to clock the reconstruction DAC at an even sub-multiple of the ADC sampling frequency (K = 2, 4, 8, etc.) to relax the settling time requirements on the DAC. The output lowpass filter is chosen to have a cutoff frequency of approximately $f_s/2.2K$ so that images are attenuated over the bandwidth of interest.

As previously mentioned, an advantage of the "back-to-back" test method is that traditional analog test equipment associated with the particular application can be used (see listings in Figure 5.44), and no additional computer hardware or DSP-based software is required.

Two of the most powerful tests for ADC ac linearity using the back-to-back test setup are the *envelope test* and the *beat frequency* test. The envelope test measures the ADC ac performance with a signal which is near $f_s/2$, and the beat frequency test uses a signal near f_s—both tests utilize the same test setup shown in Figure 5.45.

Figure 5.46 shows the sampled signal for the envelope test, where the input signal is slightly offset from $f_s/2$ by a small amount, Δf. Notice that the low frequency Δf signal appears in the two envelopes of the sampled signal. If the K factor in Figure 5.45 is set for K = 2, every other ADC sample is clocked into the output reconstruction DAC, yielding the low frequency Δf signal, and removing one of the envelopes. In practice, this

reduces the effects of DAC settling time glitches, and the low frequency Δf signal can be easily observed on an oscilloscope for nonlinearities and missing codes, or the distortion measured with a spectrum analyzer. A Δf frequency of a few hundred kilohertz generally performs satisfactorily. Both the sampling clock and the input signal should be derived from stable frequency synthesizers or crystals to prevent phase noise on the low frequency Δf signal.

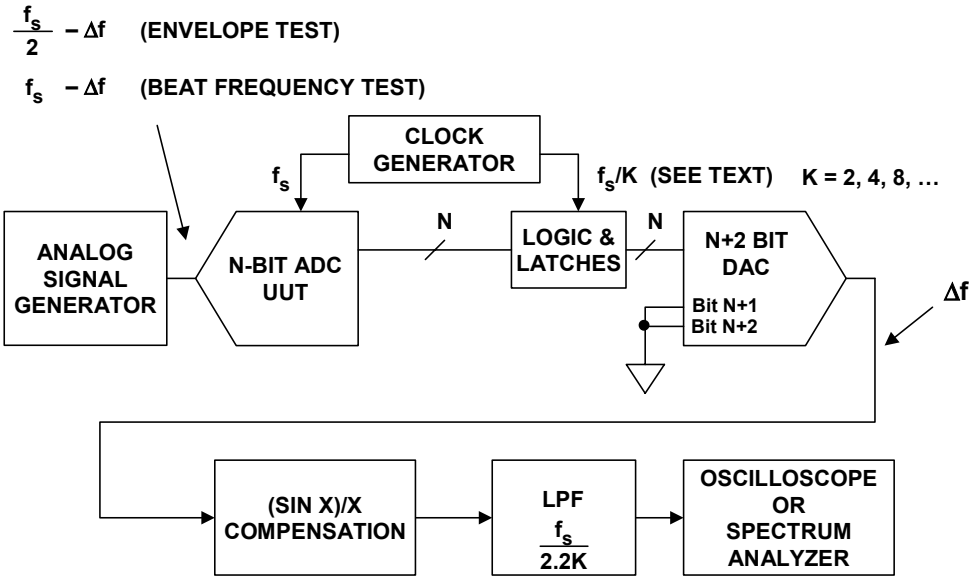

Figure 5.45: *Envelope and Beat Frequency Test Setup*

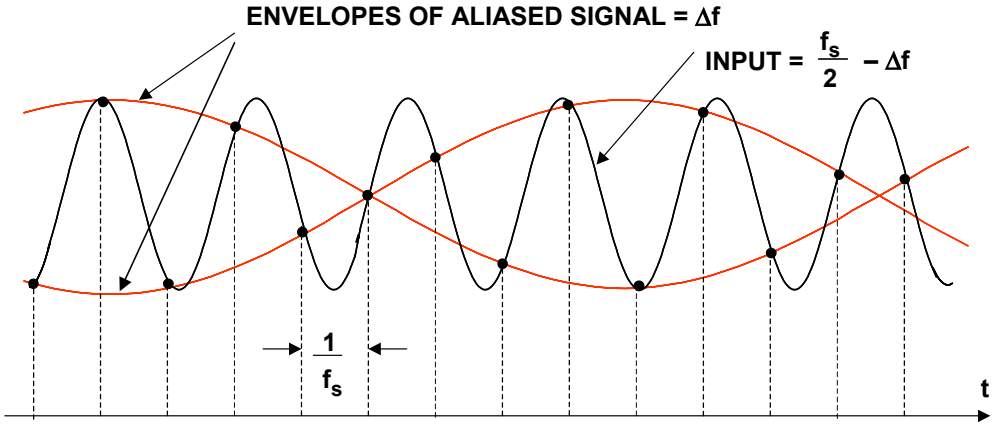

Figure 5.46: *Envelope Test with Input Frequency Near $f_s/2$*

The *beat frequency* test is essentially the same as the envelope test, except the input signal is placed near the sampling frequency f_s as shown in Figure 5.48. In this case, the low frequency "beat" is obtained directly without the need for dividing the clock to the DAC. However, dividing the clock to the DAC will decrease the sensitivity to settling time glitches as in the envelope test.

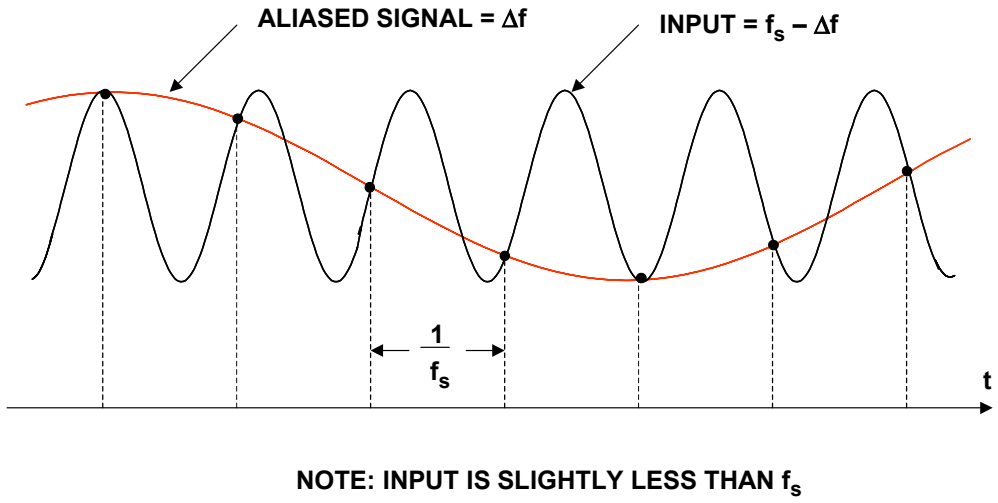

Figure 5.47: *Beat Frequency Test with Input Frequency Near f_s*

It should be noted that if the DAC clock divider is set for K = 2, the low frequency Δf signal can be observed for inputs which are near any multiple of $f_s/2$, thereby allowing the ADC to be evaluated for use in undersampling applications. The factor K should be at least 2 in order to eliminate the dual envelopes for signals near multiples of $f_s/2$. If the DAC requires more settling time in order to produce a clean display of the low frequency Δf signal, K can be chosen to be 4 or 8.

Gross ac nonlinearities and missing codes can be observed with an oscilloscope as shown in the waveforms of Figure 5.48. These tests were made on an 8-bit flash ADC sampling at 20 MSPS with an input signal slightly offset from 20 MHz. The expanded view on the right shows dramatically the effects of inadequate comparator matching in the flash ADC.

Finally, the back-to-back test setup can be used to easily measure the ADC large-signal input bandwidth. For this test, an oscilloscope is connected directly to the DAC output before the compensation and filtering and synchronized to the sampling clock. The input sinewave is set to a low frequency and adjusted until the DAC indicates the ADC is barely clipping the positive and negative peaks of the input sinewave. The input frequency can then be increased, and the measurement repeated. An increase in input signal level required to cause clipping corresponds to a decrease in the ADC gain; a decrease in input level implies an increase in gain. This test can be extended well beyond the Nyquist frequency, because the DAC response is not critical—it is only being used to detect the point of ADC clipping. Small-signal bandwidth can also be measured, where the input sinewave level is set to activate the same pre-determined number of DAC output levels at each test frequency.

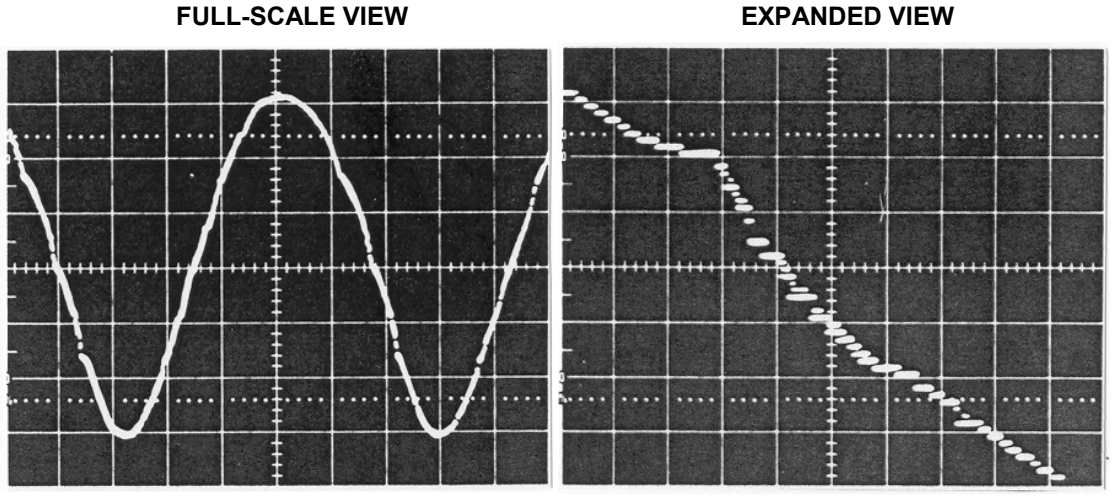

Figure 5.48: *Beat Frequency and Envelope Tests Show ADC AC Nonlinearities*

In summary, the back-to-back ADC test method can serve as a quick check of overall ADC ac performance for resolutions of up to about 10 bits. It is also useful in testing end-to-end performance in systems which use a reconstruction DAC in conjunction with an ADC. Serious evaluations of high performance ADCs with 12-bits or more of resolution requires the use of digital techniques described in the next few sections. For this type of ADC, the back-to-back test may still be useful, however, for quick checks for functionality.

Measuring Effective Number of Bits (ENOB) Using Sinewave Curve Fitting

The first DSP-based test to be discussed is the ENOB test using the sinewave curve fitting method. In order for the results to be valid, the input frequency must not be a sub-harmonic of the sampling frequency. This requirement has been previously discussed in Chapter 2 of this book, as well as the section on DAC testing in this chapter. A number of samples, M (the data record length), are first collected in a buffer memory. A good rule-of-thumb is to make M large enough to contain at least 5 complete cycles of the input sinewave (Reference 36, p. 28). The data is then read into the computer, and a best-fit sinewave computed. The rms error of the actual sample points referenced to the best-fit sinewave is then used to compute the ENOB. The requirements on the buffer memory can be relaxed using a frequency of f_s/K to clock the memory at the expense of increased test time. The test setup is shown in Figure 5.49.

The algorithm must compute the amplitude, phase, frequency, and offset of the best-fit sinewave (4-parameters). Various algorithms for doing this are available and described in the references (see References 27, 36, and 37). If the input frequency and sampling rate are accurately known, a 3-parameter algorithm (References 36 and 37) should be used (the 4-parameter algorithms may not always converge). Once the best-fit sinewave is known, the actual rms quantization error, Q_A, is computed based on the data points in the data record. This value includes errors due to INL, DNL, missing codes, aperture jitter,

noise, etc. The theoretical rms quantization error is the well-known $Q_T = q/\sqrt{12}$, where q is the weight of the LSB. The effective number of bits (ENOB) is then calculated by:

$$\text{ENOB} = \text{N} - \log_2\left[\frac{Q_A}{Q_T}\right]. \qquad \text{Eq. 5.21}$$

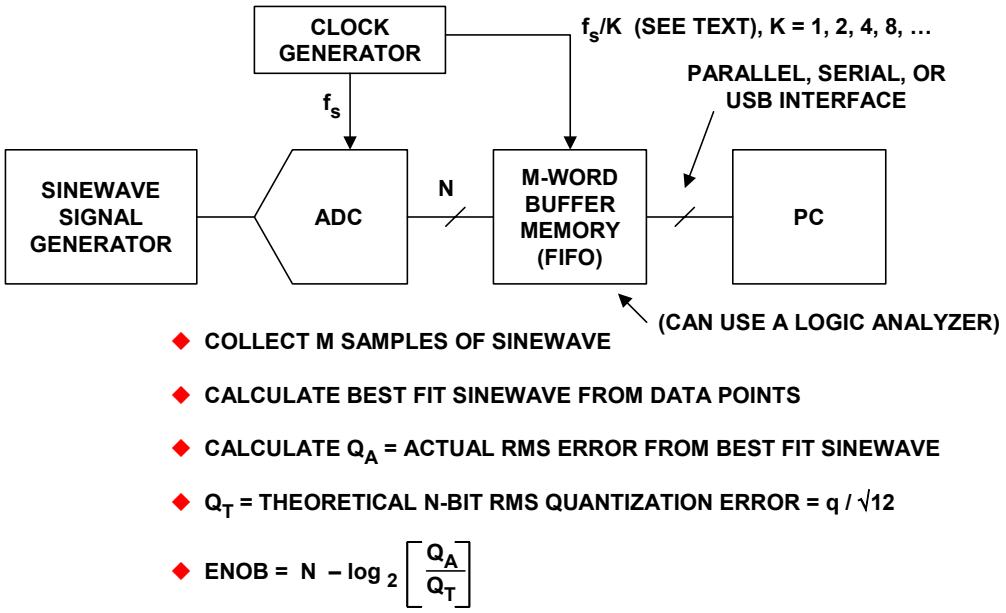

Figure 5.49: *Sinewave Curve Fit Test Setup for Measuring ADC ENOB*

Note that the sinewave curve fitting method gives no information regarding the harmonic distortion content of the error. The FFT spectral analysis technique to be described in the next section must be used if frequency-related performance measurements such as SFDR, THD, etc., are required.

It should be noted that the most popular method today for calculating ENOB makes use of the signal-to-noise and distortion ratio, SINAD, described in Chapter 2 of this book. The SINAD is easily calculated from the FFT output. ENOB is then calculated from full-scale SINAD using the equation,

$$\text{ENOB} = \frac{\text{SINAD} - 1.76\ \text{dB}}{6.02\ \text{dB}}. \qquad \text{Eq. 5.22}$$

The two calculations for ENOB for the same ADC under the same conditions should be approximately equal if the input sinewave is full-scale. If the input signal is less than full-scale, Eq. 5.22 must be corrected as follows in order to compare it to the ENOB value predicted by the sinewave curve fit in Eq. 5.21:

$$\text{ENOB} = \frac{\text{SINAD} - 1.76\ \text{dB} + \text{Level of Signal Below FS}}{6.02\ \text{dB}}. \qquad \text{Eq. 5.23}$$

FFT Basics

This section covers the basics of FFTs so that their application to ADC testing can be better understood. The emphasis is on the concepts rather than the mathematics. For the interested reader, there are a number of excellent references on the subject, and References 39 and 40 are highly recommended as a starting point. Additional references on FFTs and DSP in general are given at the end of the section (References 45-52).

Fourier analysis forms the basis for much of digital signal processing (see Figure 5.50). Simply stated, the Fourier transform (there are actually several members of this family) allows a time domain signal to be converted into its equivalent representation in the frequency domain. Conversely, if the frequency response of a signal is known, the inverse Fourier transform allows the corresponding time domain signal to be determined.

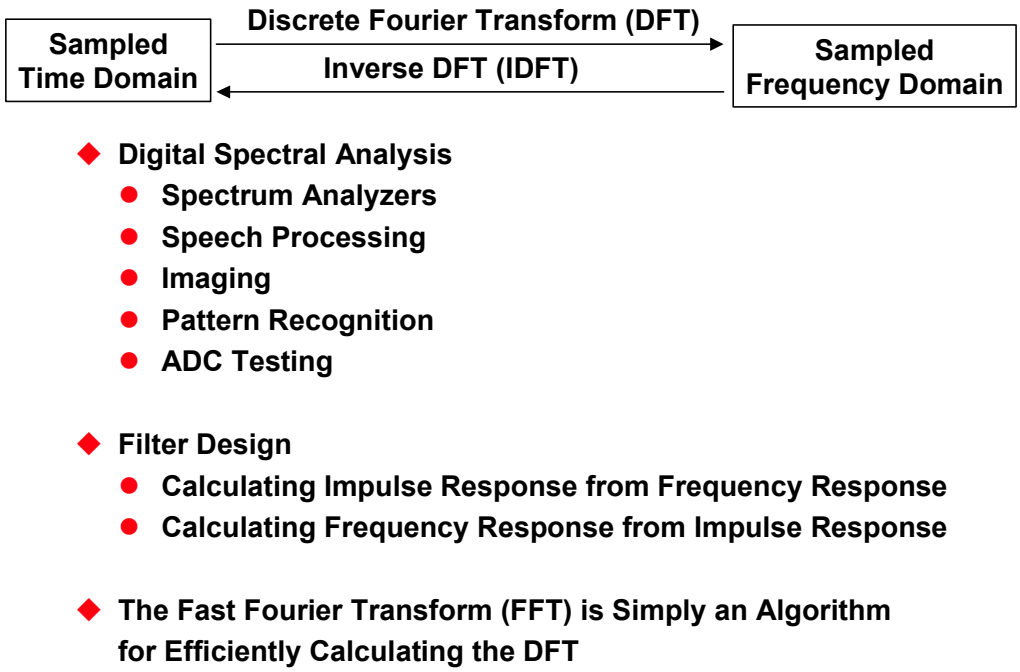

Figure 5.50: *Applications of the Discrete Fourier Transform (DFT)*

In addition to frequency analysis, these transforms are useful in filter design, since the frequency response of a filter can be obtained by taking the Fourier transform of its impulse response. Conversely, if the frequency response is specified, then the required impulse response can be obtained by taking the inverse Fourier transform of the frequency response. Digital filters can be constructed based on their impulse response, because the coefficients of an FIR filter and its impulse response are identical.

The Fourier transform family (*Fourier Transform*, *Fourier Series, Discrete Time Fourier Series*, and *Discrete Fourier Transform*) is shown in Figure 5.51. These accepted definitions have evolved (not necessarily logically) over the years and depend upon

whether the signal is *continuous–aperiodic*, *continuous–periodic*, *sampled–aperiodic, or sampled–periodic*. In this context, the term *sampled* is the same as *discrete* (i.e., a *discrete* number of time samples).

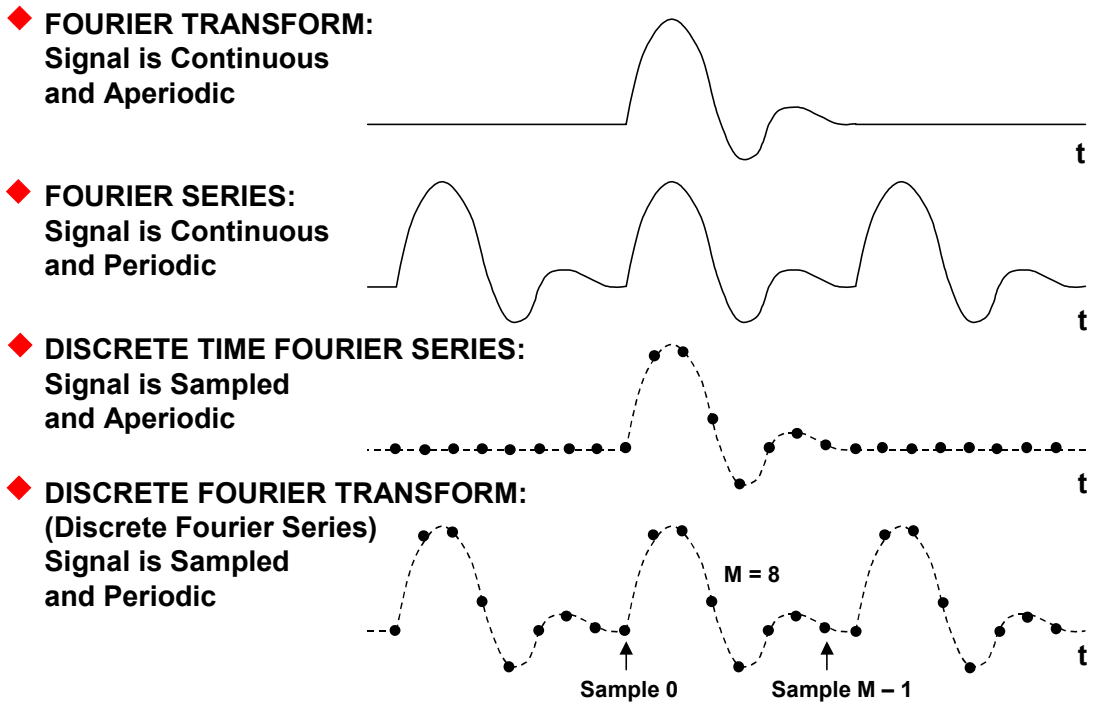

Figure 5.51: *Fourier Transform Family*

The only member of this family which is relevant to digital signal processing is the *Discrete Fourier Transform (DFT)* which operates on a *sampled* time domain signal which is *periodic*. The signal must be periodic in order to be decomposed into the summation of sinusoids. However, only a finite number of samples (M) are available for inputting into the DFT. This dilemma is overcome by placing an infinite number of groups of the same M samples "end-to-end," thereby forcing mathematical (but not real-world) periodicity as shown in Figure 5.51.

The fundamental analysis equation for obtaining the M-point DFT is as follows:

$$X(k) = \frac{1}{M}\sum_{n=0}^{M-1} x(n)e^{-j2\pi nk/M} = \frac{1}{M}\sum_{n=0}^{M-1} x(n)\left[\cos(2\pi nk/M) - j\sin(2\pi nk/M)\right]. \quad \text{Eq. 5.24}$$

At this point, some terminology clarifications are in order regarding the above equation (also see Figure 5.52). X(k) (capital letter X) represents the DFT frequency output at the k^{th} spectral point, where k ranges from 0 to M–1. The quantity M represents the number of sample points in the DFT data record, and should be a power of 2 (required by FFT routines).

The quantity x(n) (lower case letter x) represents the n^{th} time sample, where n also ranges from 0 to M – 1. In the general equation, x(n) can be real or complex, however for a

single ADC, the time samples have only a real component, and the imaginary component is set to zero.

- **A Periodic Signal Can be Decomposed into the Sum of Properly Chosen Cosine and Sine Waves (Jean Baptiste Joseph Fourier, 1807)**
- **The DFT Operates on a Finite Number (M) of Digitized Time Samples, x(n). When These Samples are Repeated and Placed "End-to-End", they Appear Periodic to the Transform.**
- **The Complex DFT Output Spectrum X(k) is the Result of Correlating the Input Samples with sine and cosine Basis Functions:**

$$X(k) = \frac{1}{M}\sum_{n=0}^{M-1} x(n)\, e^{\frac{-j2\pi nk}{M}} = \frac{1}{M}\sum_{n=0}^{M-1} x(n)\left[\cos\frac{2\pi nk}{M} - j\sin\frac{2\pi nk}{M}\right]$$

$$0 \le k \le M-1$$

Figure 5.52: *The Discrete Fourier Transform (DFT)*

Notice that the cosine and sine terms in the equation can be expressed in either polar or rectangular coordinates using Euler's equation:

$$e^{j\theta} = \cos\theta + j\sin\theta. \qquad \text{Eq. 5.25}$$

The DFT output spectrum can therefore be represented in either polar form (magnitude and phase) or rectangular form (real and imaginary) as shown in Figure 5.53. The conversion between the two forms is straightforward.

- $X(k) = ReX(k) + j\, ImX(k)$
- $MAG\,[X(k)] = \sqrt{ReX(k)^2 + ImX(k)^2}$
- $\varphi\,[X(k)] = \tan^{-1}\frac{ImX(k)}{ReX(k)}$

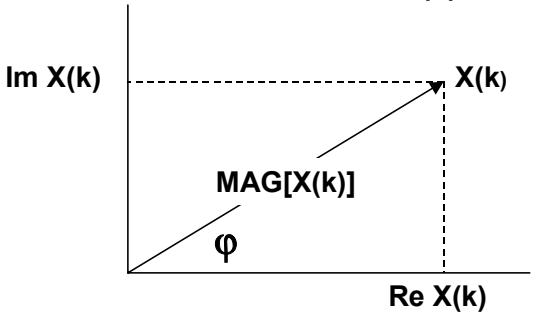

Figure 5.53: *Converting Real and Imaginary DFT Outputs into Magnitude and Phase*

In practice, a Fast Fourier Transform (FFT) is used to compute the DFT. The FFT is simply an algorithm that reduces the required number of mathematical computations. There are many FFT algorithms available, but the most popular is the Radix-2 algorithm. Computing the DFT using Equation 5.24 requires M complex multiply operations and M – 1 complex additions for each of the X(k) terms. For the entire transform, this results in M^2 complex multiplications and M(M – 1) complex additions for a total of $2M^2 – M$ complex operations. The FFT was developed to reduce the number of calculations by exploiting the symmetry properties of the DFT to eliminate redundant calculations. For example, the Radix-2 algorithm requires only $M \cdot \log_2 M$ complex operations. For a 1024-point transform, the DFT requires 2,096,128 complex operations compared to only 10,240 for the FFT. To utilize FFT algorithms, M must be an integer power of two.

Regardless of the algorithm, for each time-domain sample, a complex conjugate pair, ReX(k) + jImX(k) will be generated from the FFT. For example, if the time-domain sample size M = 16,384 (2^{14}), the resulting FFT array will contain 16,384 complex samples. In order to generate a frequency domain plot from this data, the magnitude of each complex sample must be calculated using the equation:

$$\text{MAG } X(k) = \sqrt{\text{Re } X(k)^2 + \text{Im } X(k)^2} \ . \qquad \text{Eq. 5.26}$$

If required, the corresponding phase of each point can be calculated using the equation:

$$\varphi \ X(k) = \tan^{-1} \frac{\text{Im } X(k)}{\text{Re } X(k)}. \qquad \text{Eq. 5.27}$$

Most FFTs are written to accept complex input data, in which case the FFT output will contain 16,384 magnitude (and phase) values representing frequencies between plus and minus $f_s/2$. If Equation 5.24 is used, the "positive" frequency values occur between FFT outputs k = 0 to k = M/2 (corresponding to the frequency range between dc and $f_s/2$), and the "negative" frequency values occur between FFT outputs k = M/2 and k = M – 1 (corresponding to the frequency range between $f_s/2$ and f_s). Although "complex" ADCs are not available, it is very common to use two ADCs to synchronously sample the I and Q data streams from a quadrature demodulator, in which the FFT input data is complex—however, this is a special case.

In testing a single ADC, the input data to the FFT is real, and the imaginary part of each complex input sample must be set to zero. For real input data, the FFT output samples between k = M/2 and k = M – 1 (the "negative" frequencies) represent an exact mirror image of those between k = 0 and k = M/2, and can be ignored.

The output of a typical M-point FFT is shown in Figure 5.54. The resolution of the FFT is determined by M, and the frequency *bin width* is f_s/M. The larger M, the more frequency resolution. Figure 5.54 also shows the relationship between the average noise floor of the FFT and the broadband quantization noise level (quantization noise is approximately uniformly distributed over the bandwidth dc to $f_s/2$). Each time M is doubled, the average noise in the $\Delta f = f_s/M$ frequency bin decreases by 3 dB. Note

however, that averaging the results of several individual FFTs does not change the noise floor, but only reduces the variations in the noise components.

As in the previous tests described, the requirements on the buffer memory speed can be relaxed by clocking the memory at a slower rate equal to f_s/K. However, from the perspective of the FFT performed on the actual sampled values, the sampling frequency is now f_s/K rather than the f_s. Input frequencies greater than $f_s/2K$ will appear as aliased signals in the FFT output. This must be kept in mind when interpreting the FFT output spectrum.

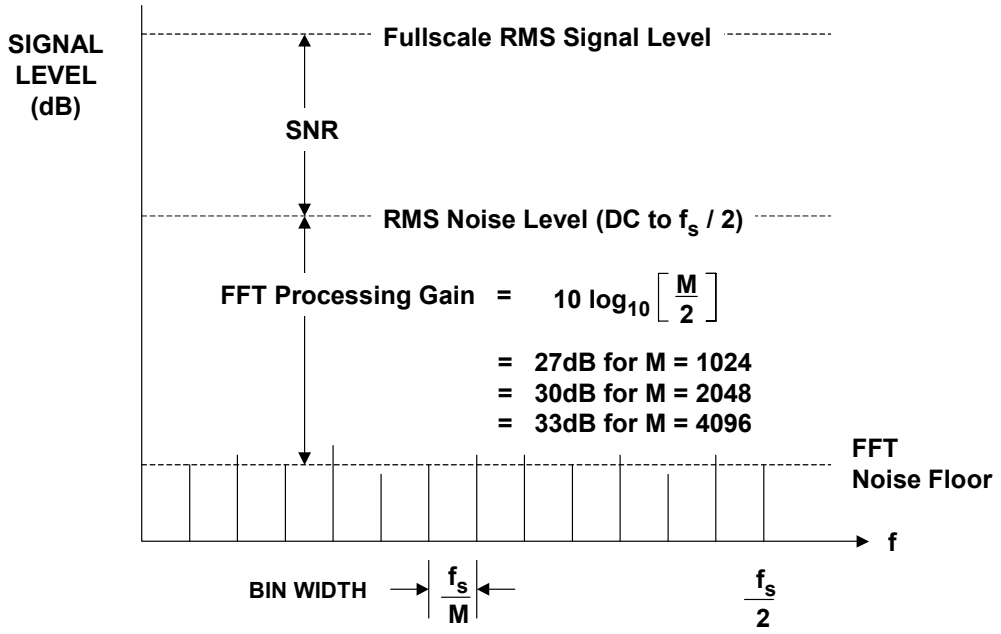

Figure 5.54: *FFT Output Shows Effects of Processing Gain*

In order to obtain spectrally pure results, the FFT data window must contain an exact integral number of sinewave cycles, otherwise spectral leakage will occur. Spectral leakage in FFT processing can best be understood by considering the case of performing an M-point FFT on a pure sinusoidal input. Two conditions will be considered. In Figure 5.55, the ratio between the sampling frequency and the input sinewave frequency is such that precisely an integral number of cycles are contained within the data window (frame, or record). Recall that the DFT assumes that an infinite number of these windows are placed end-to-end to form a periodic waveform as shown in the diagram as the periodic extensions. Under these conditions, the waveform appears continuous, and the DFT or FFT output will be a single tone located at the input signal frequency.

Figure 5.56 shows the condition where there are not an integral number of sinewave cycles within the data window. The discontinuities which occur at the endpoints of the data window result in leakage in the frequency domain, because of the sidelobes which are generated. In addition to the sidelobes, the main lobe of the sinewave is smeared over several frequency bins. This process is equivalent to multiplying the input sinewave by a rectangular window pulse which has the familiar sin(x)/x frequency response and associated smearing and sidelobes.

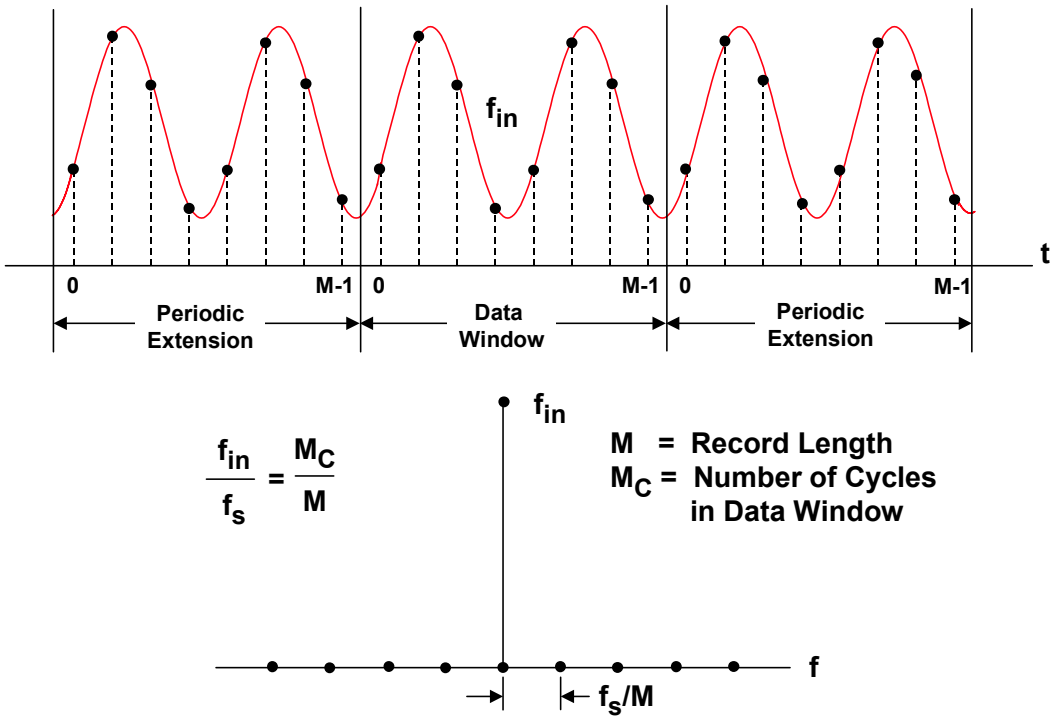

Figure 5.55: *FFT of Sinewave Having Integral Number of Cycles in Data Window*

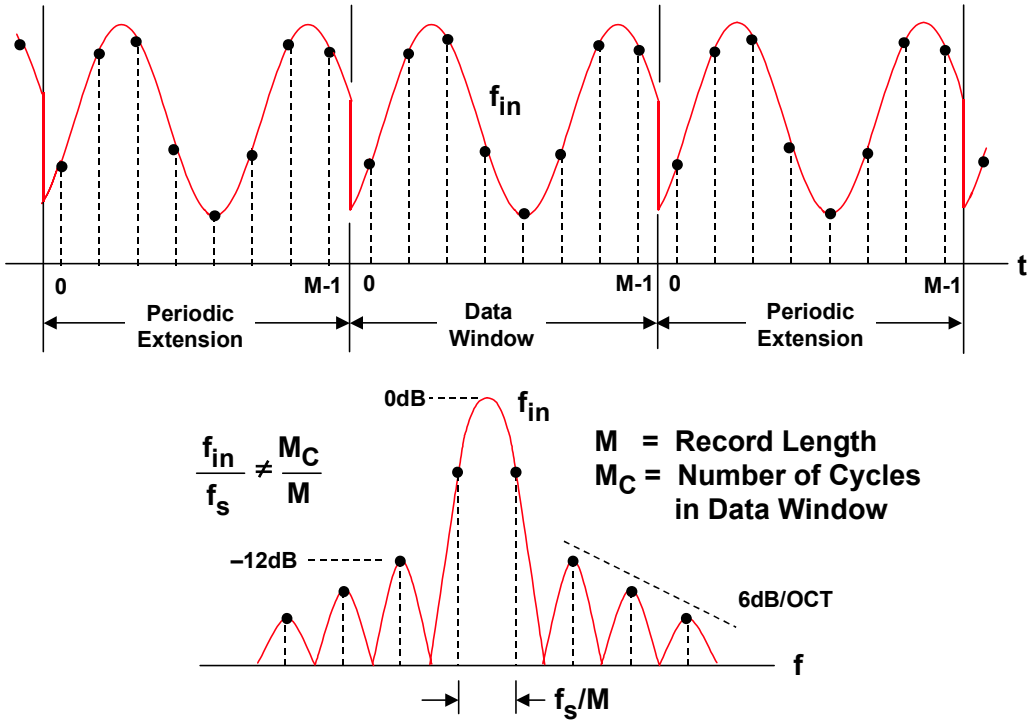

Figure 5.56: *FFT of Sinewave Having Non-Integral Number of Cycles in Data Window*

Notice that the first sidelobe is only 12 dB below the fundamental, and that the sidelobes roll off at only 6 dB/octave beyond that point. This situation would be unsuitable for most spectral analysis applications. Since in practical FFT spectral analysis applications, the exact input frequencies are unknown, something must be done to minimize these sidelobes. This is accomplished by choosing a window function other than the rectangular window. The input time samples are multiplied by an appropriate window function which brings the signal to zero at the edges of the window as shown in Figure 5.57. The selection of a window function is primarily a tradeoff between main-lobe spreading and sidelobe roll off. Reference 50 is highly recommended for an in-depth treatment of window functions.

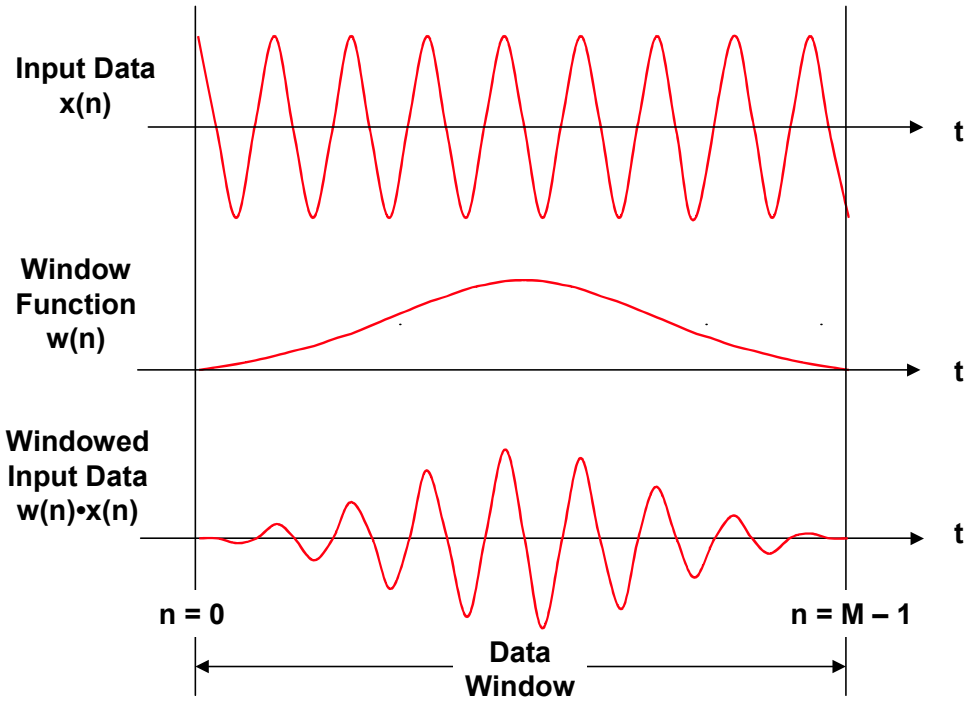

Figure 5.57: *Windowing to Reduce Spectral Leakage*

The mathematical functions which describe four popular window functions (Hamming, Blackman, Hanning, and Minimum 4-term Blackman-Harris) are shown in Figure 5.58. The computations are straightforward, and the window function data points are usually pre-calculated and stored in the DSP memory to minimize their impact on FFT processing time. The frequency response of the rectangular, Hamming, and Blackman windows are shown in Figure 5.59. The Hanning window and the Minimum 4-Term Blackman-Harris window are popular in ADC testing and are shown in Figure 5.60.

Figure 5.61 shows the tradeoff between main-lobe spreading and sidelobe amplitude and roll-off for the popular window functions.

- **Hamming:** $w(n) = 0.54 - 0.46 \cos\left[\frac{2\pi n}{M}\right]$
- **Blackman:** $w(n) = 0.42 - 0.5 \cos\left[\frac{2\pi n}{M}\right] + 0.08 \cos\left[\frac{4\pi n}{M}\right]$
- **Hanning:** $w(n) = 0.5 - 0.5 \cos\left[\frac{2\pi n}{M}\right]$
- **Minimum 4-Term Blackman Harris:** $w(n) = 0.35875 - 0.48829 \cos\left[\frac{2\pi n}{M}\right] + 0.14128 \cos\left[\frac{4\pi n}{M}\right] - 0.01168 \cos\left[\frac{6\pi n}{M}\right]$

$$0 \le n \le M - 1$$

Figure 5.58: *Some Window Functions*

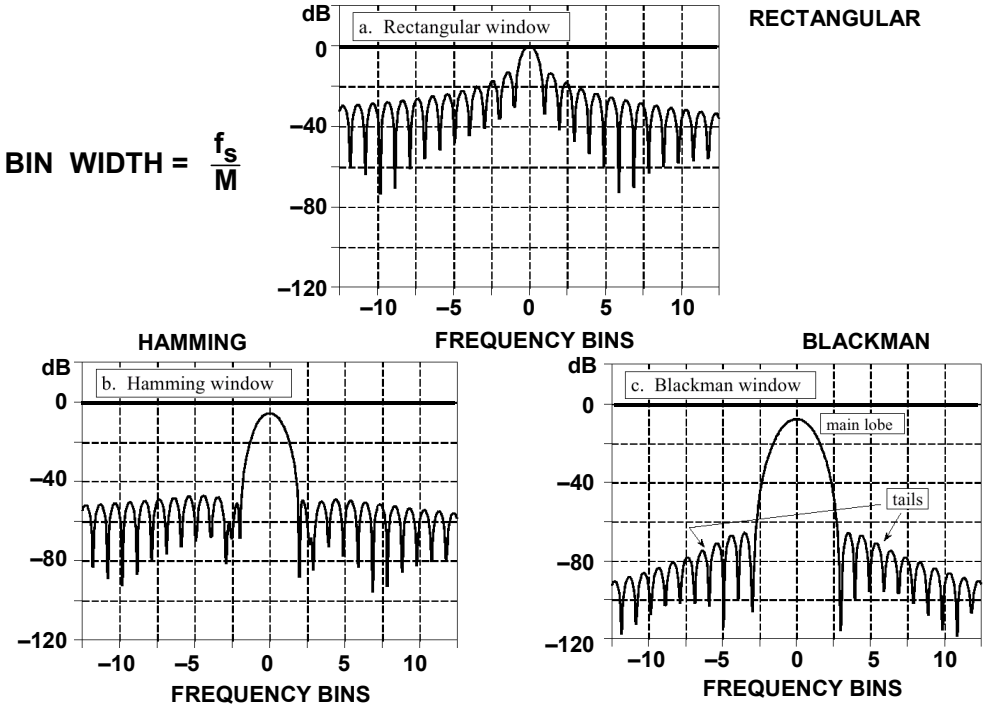

Figure 5.59: *Frequency Response of Rectangular, Hamming, and Blackman Windows for M = 256*

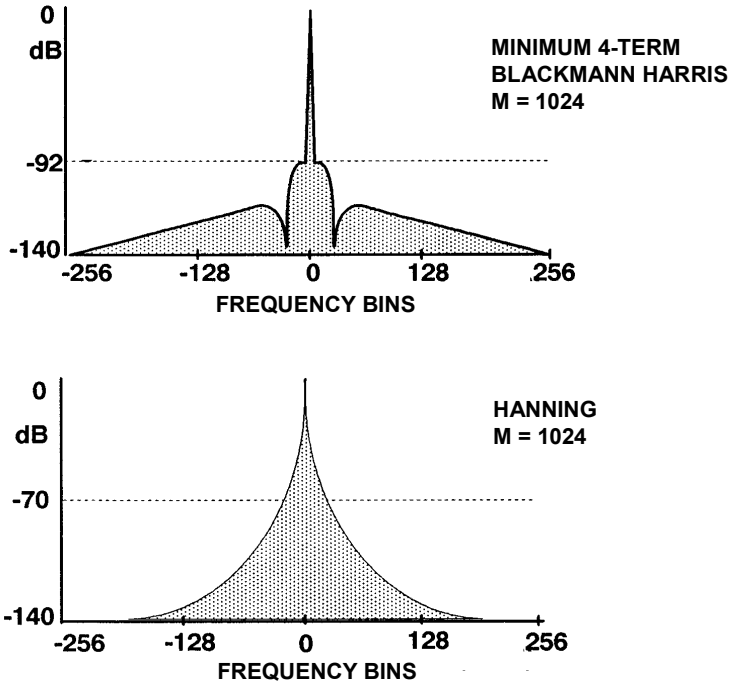

Figure 5.60: *Comparison of Two Popular Window Functions Used in ADC Testing*

WINDOW FUNCTION	3dB BW (Bins)	6dB BW (Bins)	HIGHEST SIDELOBE (dB)	SIDELOBE ROLLOFF (dB/Octave)
Rectangle	0.89	1.21	–12	6
Hamming	1.3	1.81	– 43	6
Blackman	1.68	2.35	–58	18
Hanning	1.44	2.00	–32	18
Minimum 4-Term Blackman-Harris	1.90	2.72	–92	6

Figure 5.61: *Popular Windows and Figures of Merit*

FFT Test Setup Configuration and Measurements

The typical FFT test setup shown in Figure 5.62 can be implemented in a number of ways. Maximum utilization of manufacturer's evaluation boards greatly simplifies the test and ensures proper layout of the critical components surrounding the ADC. A well-designed evaluation board should have input buffer amplifiers and/or transformers to

drive the ADC, sampling clock conditioning circuits (perhaps even a stable crystal oscillator), voltage references (if required), output data registers, and appropriate input/output connectors. The importance of a low-jitter sampling clock source cannot be overemphasized at this point. The effects of sampling clock jitter on the ADC SNR has previously been discussed in Chapter 2, and Chapter 6 of this book has further suggestions regarding low-jitter clock sources. Careful attention to grounding, layout, and decoupling is also important, as coupling of the digital outputs into either the sampling clock or the ADC input can degrade SNR, SINAD, and SFDR performance. For these reasons, the use of a manufacturer's evaluation board is highly recommended.

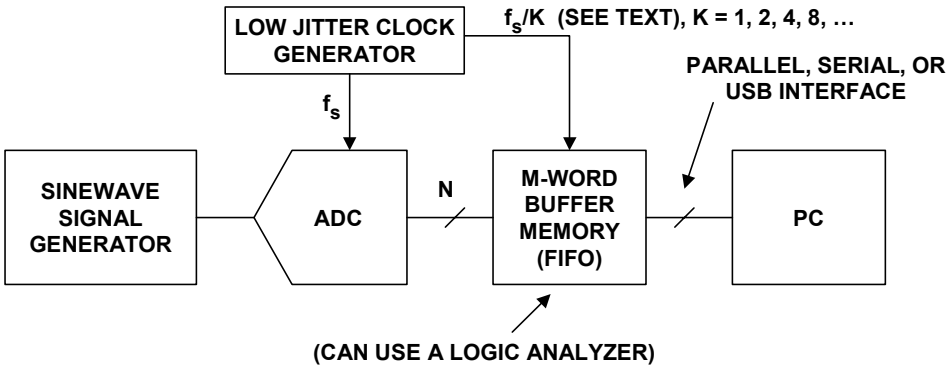

Figure 5.62: *FFT Test Setup*

It should be noted that there are a variety of popular commercial software packages available today which include suitable FFT routines (References 53, 54, and 55), so there is little need to actually write the FFT itself.

In order to further simplify the evaluation process, Analog Devices offers a High-Speed ADC FIFO Evaluation Kit that interfaces directly to a connector on the ADC evaluation board (see Figure 5.63). The FIFO evaluation kit includes a memory board to capture blocks of data from the ADC as well as Windows® compatible ('95, '98, 2000, NT) ADC Analyzer™ software. The FIFO board can be connected to the parallel port of a PC through a standard printer cable and used with the ADC Analyzer software to quickly evaluate the performance of the high speed ADC.

The FIFO board contains two 32K, 16-bit wide FIFOs, and data can be captured at clock rates up to 133 MSPS on each channel. Memory upgrades are available to increase the size of the FIFO to 64K, 132K, or 256K. Two versions of the FIFO are available—one version is used with dual ADCs or ADCs with demultiplexed digital outputs, and the other version is used with single-channel ADCs. Users can view the FFT output and analyze SNR, SINAD, SFDR, THD, and harmonic distortion information.

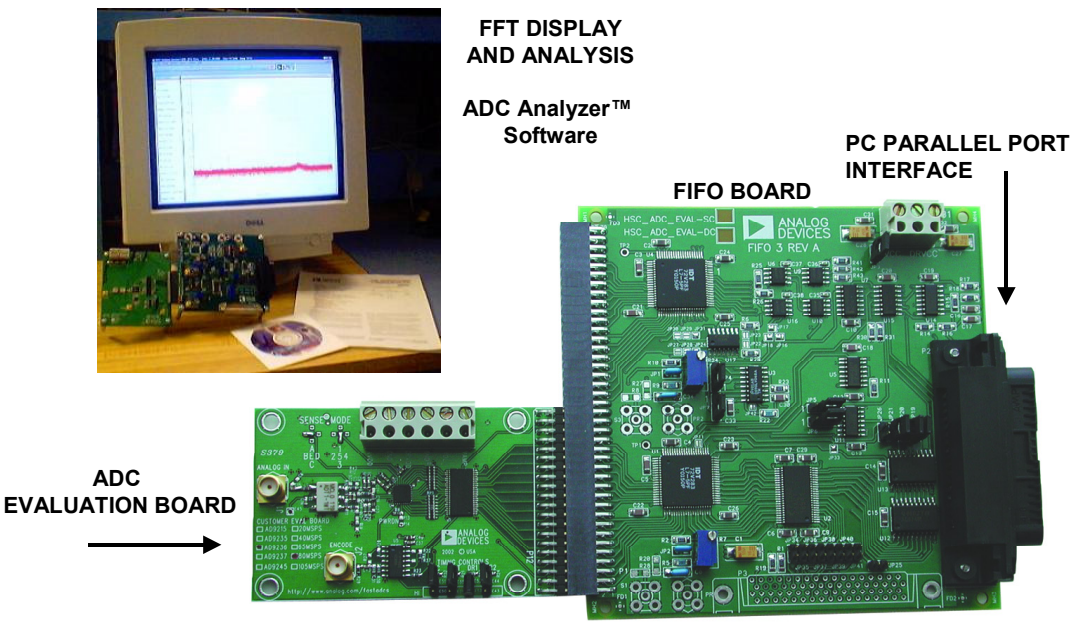

Figure 5.63: *Analog Devices' High Speed ADC FIFO Evaluation Kit*

After the ADC Analyzer software computes the FFT, there are two ways to evaluate the ADC performance: graphically and computationally. In order to plot the data in a meaningful way, the magnitude data must be converted to decibels (dB). This can be done using the formula:

$$\mathrm{dB} = 10\log_{10}\left[\frac{\mathrm{Magnitude}^2}{\mathrm{Magnitude}_{\mathrm{Max}}{}^2}\right] = 20\log_{10}\left[\frac{\mathrm{Magnitude}}{\mathrm{Magnitude}_{\mathrm{Max}}}\right], \qquad \text{Eq. 5.28}$$

where *Magnitude* is the individual array elements computed by the FFT, and $Magnitude_{Max}$ is the maximum magnitude element in the array.

A typical FFT plot using the FIFO evaluation kit and the ADC Analyzer software is shown in Figure 5.64 for the AD9430 12-bit, 170-/210-MSPS ADC. For this test, the sampling rate is 170 MSPS, the input frequency is 10.314 MHz, and the FFT size is M = 16,384.

The key inputs to the ADC Analyzer software configuration file are shown in Figure 5.65. Note that averaging the results of several FFTs does not change the FFT noise floor, it just reduces the variations in the random noise. Most inputs to the file are self-explanatory, but the issue of coherent versus non-coherent sampling deserves further discussion.

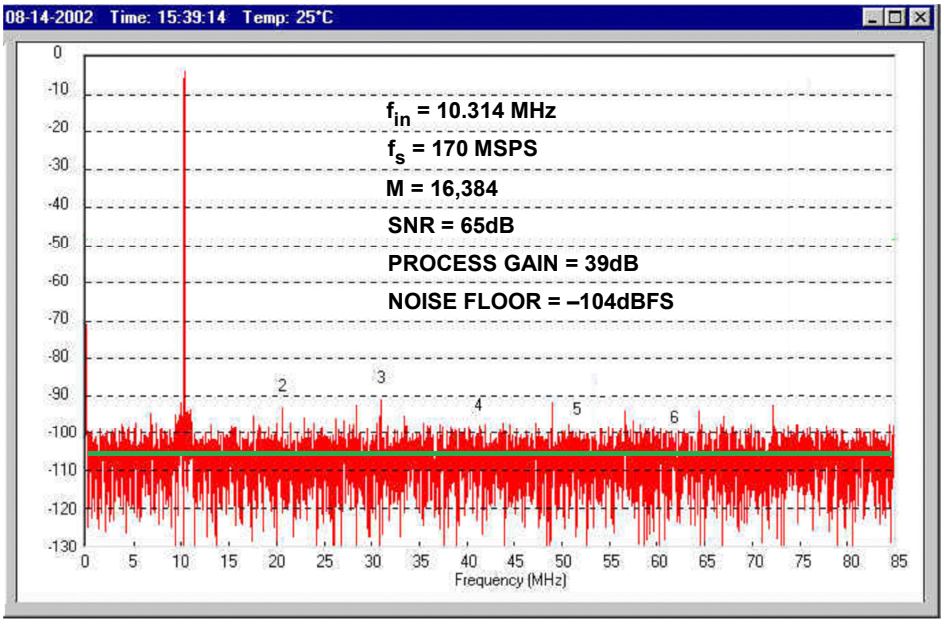

Figure 5.64: *Typical FFT Output Display for AD9430 12-Bit, 170-/210-MSPS ADC Using FIFO Kit*

- **Device, Device Number, Number of Bits, Temperature, Default Data Directory**
- **Number of FFT Samples (16,384 Default, up to 32K), up to 256K with FIFO Upgrades**
- **Two's Complement or Straight Binary Coding**
- **Number of FFT Averages (Default 5)**
- **Sampling Frequency**
- **Fundamental Leakage (±10 Bins Default for Minimum 4-term Blackman Harris, ±25 Bins for Hanning)**
- **Harmonic Leakage (±3 Bins Default)**
- **DC Leakage (6 Bins Default)**
- **Maximum Number of Harmonics (default- 2nd, 3rd, 4th, 5th, 6th)**
- **Windowing (Hanning, Min. 4-Term Blackman-Harris-Default, None)**
- **Utilize Coherent Sampling Frequency Calculator for no windowing**
- **Power Supply**

Figure 5.65: *Inputs to ADC Analyzer Configuration File*

If coherent sampling is used, the fundamental and its harmonics fall in single frequency bins as shown in Figure 5.66. A strict relationship between the input frequency, f_{in}, and the sampling frequency, f_s, must be observed:

$$\frac{f_{in}}{f_s} = \frac{M_C}{M}, \qquad \text{Eq. 5.29}$$

where M_C is the integer number of cycles of the input sinewave contained in the data record M. In performing the SINAD calculation, the rss value of all the frequency bins (excluding the dc term) is the value of the noise and distortion energy.

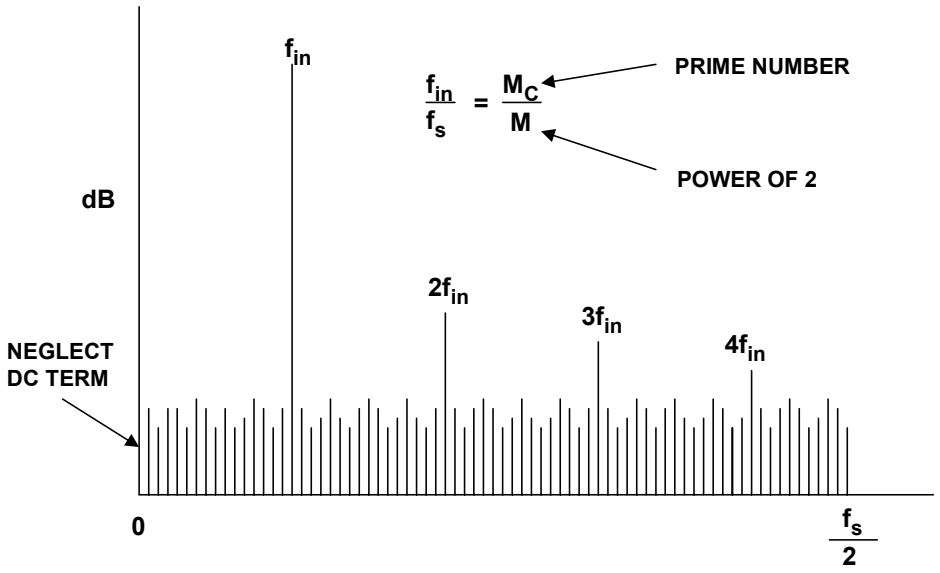

Figure 5.66: *FFT Output Signal and Harmonics for Coherent Sampling*

In order to prevent repetitive data patterns in the data record and ensure a random quantization noise spectrum, M_C should be a prime number. The input frequency and the sampling clock should be generated from locked frequency synthesizers in order to maintain the exact relationship. The ADC Analyzer software contains a Coherent Sampling Calculator to facilitate the calculation of the input frequency. To use the Coherent Sampling Calculator, the correct sampling frequency must first be entered in the configuration file. Then either the approximate analog input frequency or the number of sinewave cycles is entered. The Calculator then recommends a coherent input frequency and the number of cycles based on the input values.

If non-coherent sampling is used, the energy of the fundamental and its harmonics leaks into adjacent bins as shown in Figure 5.67. As previously discussed, the amount of leakage is dependent upon the particular windowing function used. When calculating the energy of the fundamental signal and its harmonics, the rss value of a number of adjacent bins should be used as shown in Figure 5.67. It is important not to count these bins as noise bins, because this will give an inaccurate SNR calculation.

The dc component of the FFT output also exhibits leakage with non-coherent sampling, and those bins should also be excluded from the noise calculations to prevent erroneous results. In the ADC Analyzer software, the default signal leakage bins are ±10 bins for the Minimum 4-term Blackman-Harris window and ±25 bins for the Hanning window. The default for the harmonic leakage is ±3 bins, and the dc leakage default is 6 bins. All the leakage values can be modified if required, although the default values work well in most cases.

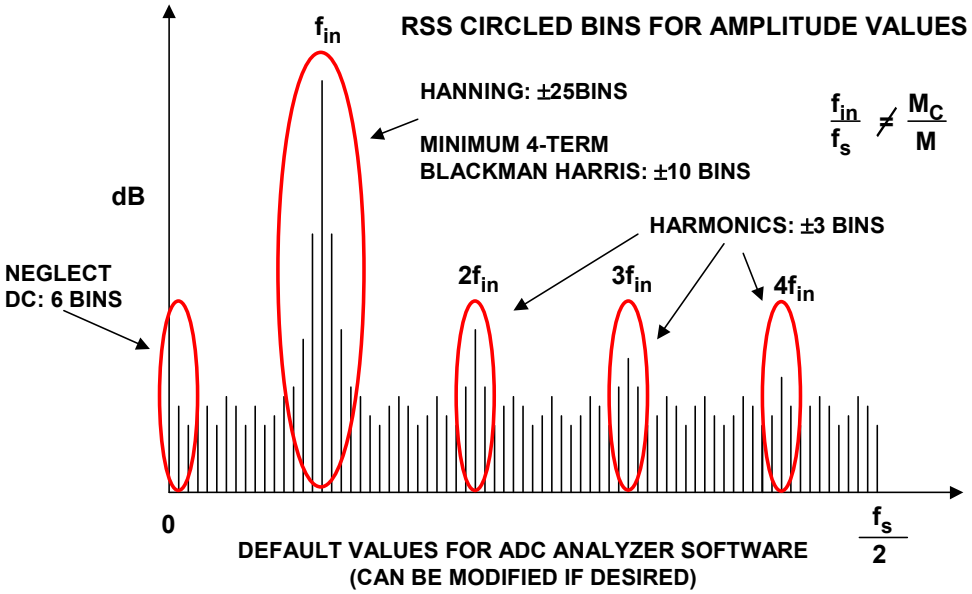

Figure 5.67: *FFT Output Signal and Harmonic Leakage for Non-Coherent Sampling*

The SNR and SINAD measurements can be drastically affected if signal leakage is not taken into account. The value of the noise for use in the SNR calculation is obtained by taking the rss value of all noise bins, excluding the leakage bins around dc, the fundamental signal, and the 2nd, 3rd, 4th, 5th, and 6th harmonics. Therefore, including signal or dc leakage bins with amplitudes above the noise floor will give a SINAD measurement which is lower than actual.

The value of noise and distortion for the SINAD calculation is obtained by taking the rss value of all noise and distortion bins excluding the leakage bins around dc and the fundamental signal. Therefore, including signal, harmonic, or dc leakage bins with amplitudes above the noise floor will give an SNR measurement which is lower than actual.

The decision to use coherent or non-coherent testing is largely a matter of preference. While coherent testing eliminates the requirement for windowing, selecting the appropriate frequencies and their ratios can become tedious, especially when multi-tone tests are required. In addition, locked frequency synthesizers are required to maintain the exact frequency ratios. One can argue that coherent testing is more suitable to a laboratory environment, while non-coherent testing is more like a real-world application, where the exact input frequencies are unknown. In practice, either method will yield approximately the same final results provided the tests are performed correctly.

The ADC Analyzer software not only gives the FFT plot but also calculates the various performance characteristics, and they are summarized in Figure 5.68 for a single-tone test signal and in Figure 5.69 for a two-tone test signal.

- **Time domain reconstruction of captured data**
- **FFT Plot**
- **Calculated and displayed values**
  - **Analog and digital power supply voltages**
  - **Sampling frequency**
  - **Analog input frequency**
  - **SNR (relative to signal)**
  - **SNRFS (relative to full-scale)**
  - **SINAD**
  - **Level of fundamental signal (dBFS)**
  - **Harmonics: 2nd, 3rd, 4th, 5th, 6th (dBc)**
  - **WoSpur: Worst non-harmonic spur (dBc)**
  - **THD (rss value of 2nd, 3rd, 4th, 5th, and 6th harmonics), dBc**
  - **SFDR (dBc)**
  - **Noise Floor (dBFS)**

Figure 5.68: *ADC Analyzer Software Outputs Single-Tone Input*

- **Time domain reconstruction of captured data**
- **FFT Plot**
- **Calculated and displayed values**
  - **Analog and digital power supply voltages**
  - **Sampling frequency**
  - **Analog 1: First analog input frequency**
  - **Analog 2: Second analog input frequency**
  - **Fundamental 1: Level of first fundamental tone (dBFS)**
  - **Fundamental 2: Level of second fundamental tone (dBFS)**
  - **F1 + F2: Sum of the fundamental tones (dBFS)**
  - **F2 – F1: Difference of the fundamental tones (dBFS)**
  - **IMD Product Levels at: 2F1 – F2, 2F1 + F2, 2F2 – F1, 2F2 + F1 (dBFS)**
  - **WoIMD: Worst IMD product (dBc)**
  - **SFDR (dBc)**
  - **Noise Floor (dBFS)**

Figure 5.69: *ADC Analyzer Software Outputs Two-Tone Input*

Verifying the FFT Accuracy

In many cases, it is desirable to perform some type of test verification on the FFT software itself, independent of the actual ADC under test. This can be done in a number of ways. A simple method is to simply disable a number of the ADC LSBs and see if the calculated values of SINAD and SNR approach the theoretical numbers for the reduced resolution. The assumption here is that a high-performance 12- or 14-bit ADC should

approach near theoretical performance at the 8-bit level.. The LSBs can be disabled in hardware or software.

Another method requires the generation of an ideal N-bit digital sinewave, running the data through the FFT, and comparing the results to theoretical. The n^{th} time sample for an ideal N-bit ADC can be written as:

$$v(n) = INT\left[2^{N-1} \sin\left(\frac{2\pi n f_{in}}{f_s} \right) \right], \qquad \text{Eq. 5.30}$$

where the "INT" function simply truncates the fractional portion of v(n).

The SINAD and SNR values obtained from the FFT can be compared with theoretical, and the overall dynamic range of the FFT routine can be found by simply increasing N until the SINAD and SNR values no longer increase by 6.02 dB per extra bit of resolution.

Generating Low-Distortion Sinewave Inputs

Generating test signals with the spectral purity required to make low distortion high frequency measurements is a challenging task. Although there are low distortion signal generators available, they can be quite expensive. For quick evaluations where such equipment is not available, proper filtering can produce a suitable test signal from lower cost generators.

A test setup for generating a low distortion single tone is shown in Figure 5.70. The sinewave oscillator should have low phase noise to prevent elevation of the ADC noise floor. The output of the oscillator is passed through a bandpass (or lowpass) filter, which removes any harmonics present in the oscillator output. The filter may not be required if the distortion of the generator is low enough. The generator distortion should be 10 dB lower than the desired accuracy of the measurement. The 6-dB attenuator isolates the DUT (ADC) from the output of the filter. The impedance at each interface should be maintained at 50 Ω for best performance (75-Ω components can be used, but 50-Ω attenuators and filters are generally more readily available). The termination resistor, R_T, is selected so that the parallel combination of R_T and the input impedance of the DUT is 50 Ω.

Before performing the actual distortion measurement, the oscillator output should be set to the correct frequency and amplitude. Measure the distortion at the output of the attenuator with the DUT replaced by a 50-Ω termination resistor (generally the 50-Ω input of a spectrum analyzer). Next, replace the 50-Ω load with R_T and the DUT. Measure the distortion at the DUT input a second time. This allows non-linear DUT loads to be identified. Non-linear DUT loads (such as flash ADCs with signal-dependent input capacitance, or switched-capacitor CMOS ADCs) can introduce distortion at the DUT input.

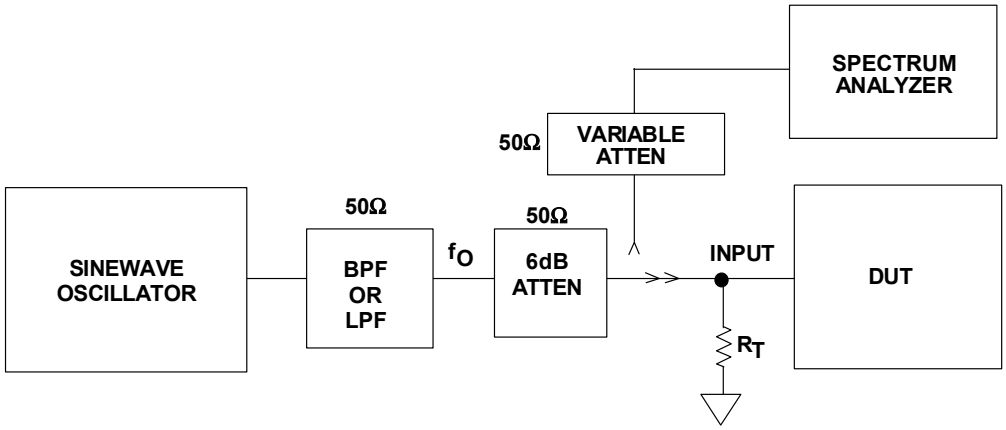

Figure 5.70: *Low Distortion Single-Tone Generator*

Generating two tones suitable for IMD measurements can be very challenging. A low-distortion two-tone generator is shown in Figure 5.71. Two bandpass (or lowpass) filters are required as shown. Harmonic suppression of each filter must be better than the desired measurement accuracy by at least 6 dB. A 6-dB attenuator at the output of each filter serves to isolate the filter outputs from each other and prevent possible cross-modulation. The outputs of the attenuators are combined in a passive 50-Ω combining network, and the combiner drives the DUT. The oscillator outputs are set to the required level, and the IMD of the final output of the combiner is measured. The measurement should be made with a single termination resistor, and again with the DUT connected to identify non-linear loads.

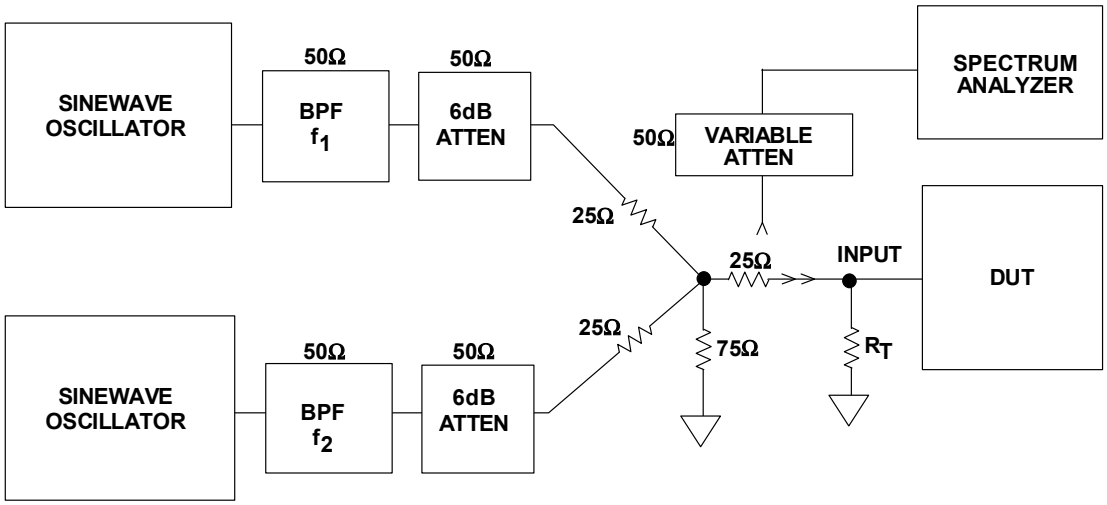

Figure 5.71: *Low Distortion Two-Tone Generator*

Analog spectrum analyzers should be checked independently for distortion when measuring these types of signals. Most have 50-Ω inputs, therefore an isolation resistor between the device under test (DUT) and the analyzer is required to simulate DUT loads greater than 50 Ω. After adjusting the spectrum analyzer for bandwidth, sweep rate, and sensitivity, check it carefully for input overdrive as shown in Figure 5.72. The simplest method is to use the variable attenuator to introduce a fixed amount of attenuation in the

analyzer input path. Both the signal and any harmonics should be attenuated by that same fixed amount (10 dB, for instance) as observed on the screen of the spectrum analyzer. If the harmonics are attenuated by more than 10 dB, then the input amplifier of the analyzer is introducing distortion, and the sensitivity should be reduced. Many analyzers have an attenuator on the front panel for introducing a known amount of attenuation when checking for overdrive. It should be noted that most high quality modern spectrum analyzers are not generally as sensitive to input overdrive as their older counterparts. However, these simple checks don't take long to perform and are well worth the additional effort to ensure accurate distortion measurements.

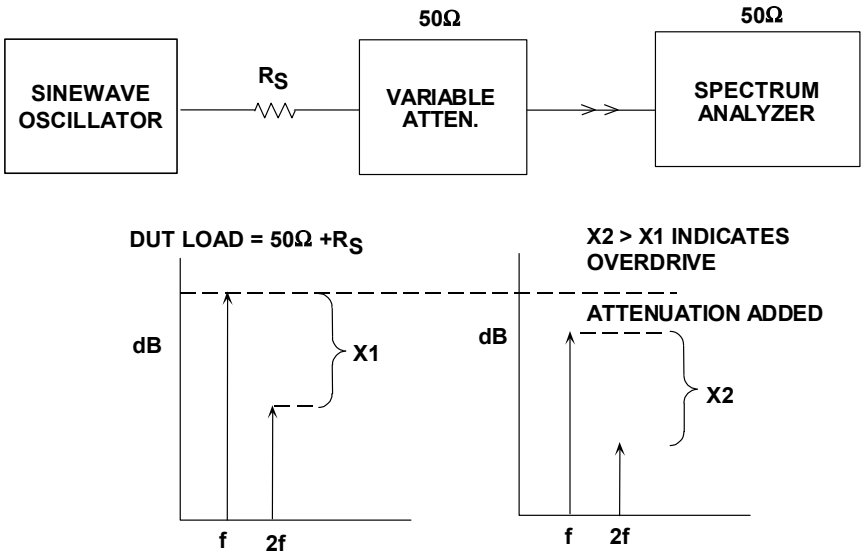

Figure 5.72: *Verifying Spectrum Analyzer Sensitivity to Input Overdrive*

If the generator is found to be sensitive to overdrive, a method to minimize the problem is shown in Figure 5.73. The amplitude of the fundamental signal is first measured with the notch filter switched out. The harmonics are measured with the notch filter switched in. The insertion loss of the notch filter, X dB, must be added to the measured level of the harmonics.

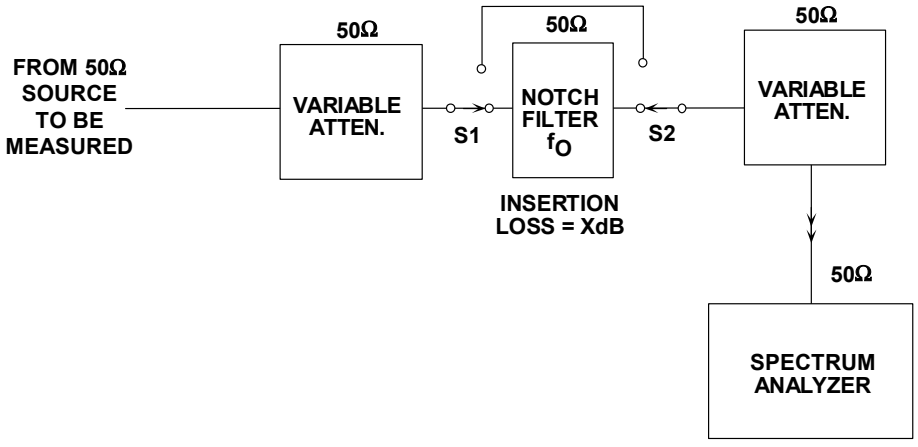

Figure 5.73: *Notch Filter Removes the Fundamental Signal to Minimize Analyzer Overdrive*

Noise Power Ratio (NPR) Testing

As described in Chapter 2 of this book, the Noise Power Ratio (NPR) measurement had its origin in the early days of frequency-division-multiplexed (FDM) telephone systems. Multiple FDM channels are simulated with Gaussian noise, and a 4-kHz wide channel is "notched" out of the input signal. The residual signal inside the notch after passing through the transmission system represents clipping noise, thermal noise, and IMD distortion products. The NPR is simply the ratio of the average noise power outside the notch to the average noise power inside the notch, i.e, it is the "depth" of the notch. Because NPR is a function of the signal amplitude, the input signal amplitude is varied until the point of maximum NPR is reached. A simplified test setup for measuring the NPR of an ADC is shown in Figure 5.74.

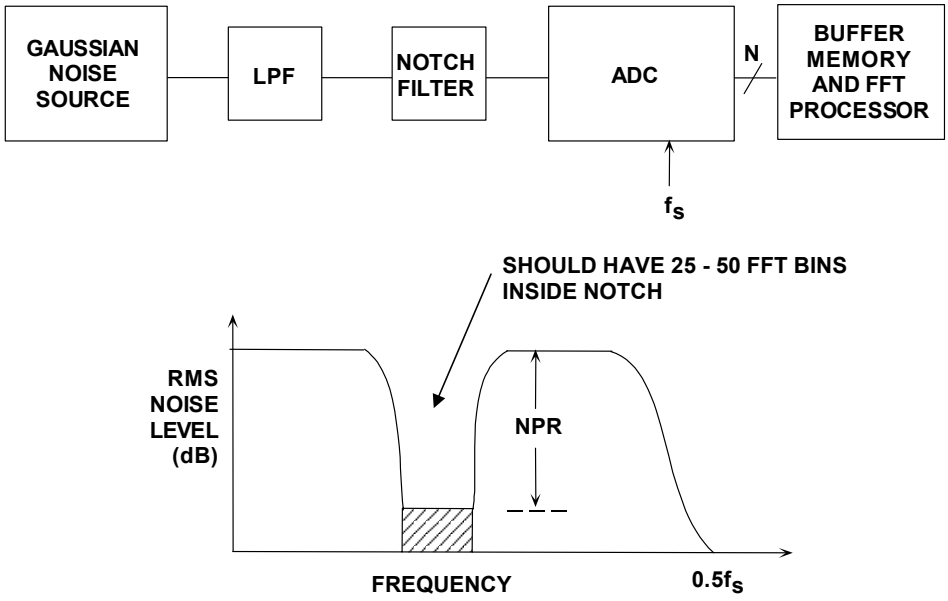

Figure 5.74: *Noise Power Ratio (NPR) Test Setup*

Conceptually, the test is easily implemented by performing an FFT on the ADC output and analyzing the data. In practice, however, an extremely large buffer memory and FFT is required in order to measure the NPR in a 4-kHz wide voice channel. For example, with a sampling rate of 170 MSPS, a 256-K FFT has a frequency resolution (bin width) of 170 MSPS/256,000 = 664 Hz. This produces only 6 samples "inside" the notch—not enough to get repeatable readings of the noise level. The solution is to use a wider notch filter, rather than use an impractically large FFT. The notch frequency filter width should be increased such that at least 25 to 50 samples fall inside the notch. Even with the wider notch, several FFT runs should be averaged in order to stabilize the NPR data. Using the wider notch does not invalidate the results by any means, and more closely simulates the wider channels used in modern communications systems. In fact, the NPR test is often a good substitute for a multi-tone test, assuming the input tones are not phase-correlated. As in the case of any FFT-based test on high performance ADCs, the sampling clock jitter must be at an acceptably low level so as not to affect the ADC noise floor.

A typical NPR display for the AD9430 12-bit, 170-MSPS ADC is shown in Figure 5.75. The FFT size is 16,384 which gives a frequency resolution of 170 MSPS/16,384 = 10.4 kHz. The notch filter width is approximately 500 kHz, yielding approximately 48 samples inside the notch. Due to the specific requirements on the center frequency, width, and bandstop rejection, custom-made notch filters are generally required in order to implement NPR tests on ADCs.

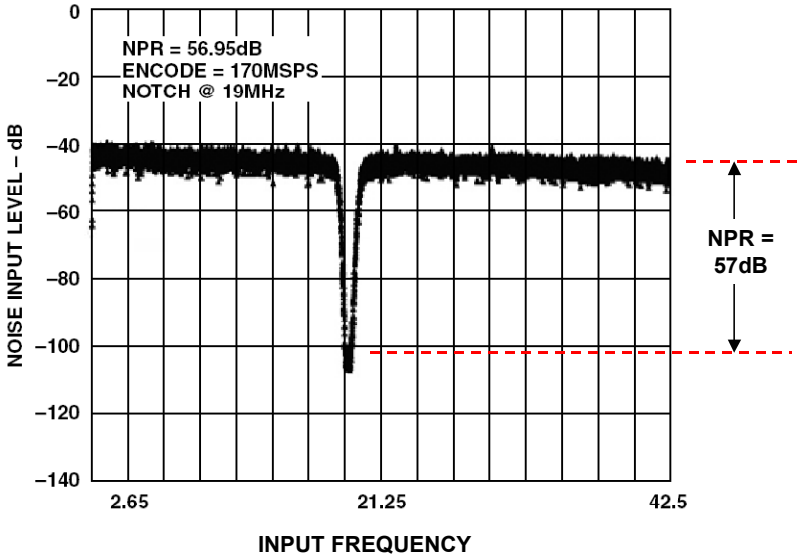

Figure 5.75: *AD9430 12-bit, 170-MSPS ADC NPR Measures 57 dB (62.7-dB Theoretical)*

Measuring ADC Aperture Jitter Using the Locked-Histogram Test Method

A test setup for measuring aperture jitter using the "locked-histogram" test is shown in Figure 5.76. The ADC input signal and the sampling clock are derived from the same low phase noise clock generator in order to minimize jitter between the two signals. Recall from the discussions in Chapter 2 of this book that the effects of sampling clock jitter are indistinguishable from internal ADC aperture jitter. Ideally, the test is run at the maximum ADC sampling frequency. However, if the input bandwidth of the ADC is not high enough, the test can be run at one-half the maximum sampling rate.

The clock generator drives the sampling clock input of the ADC directly through an attenuator. The clock generator also drives a bandpass filter (to convert the squarewave into a sinewave). The output of the bandpass filter passes through a passive variable phase shifter, an attenuator, and is then ac-coupled into the ADC input. The setup is calibrated by setting Attenuator 1 to minimum attenuation and then setting the output level of the clock generator such that the input signal to the ADC is full-scale ($2V_{FS}$ peak-to-peak). The dc offset to the ADC is then adjusted such that the input sinewave is centered around mid-scale. Attenuator 2 is then adjusted to provide the proper level for the sampling clock input to the ADC.

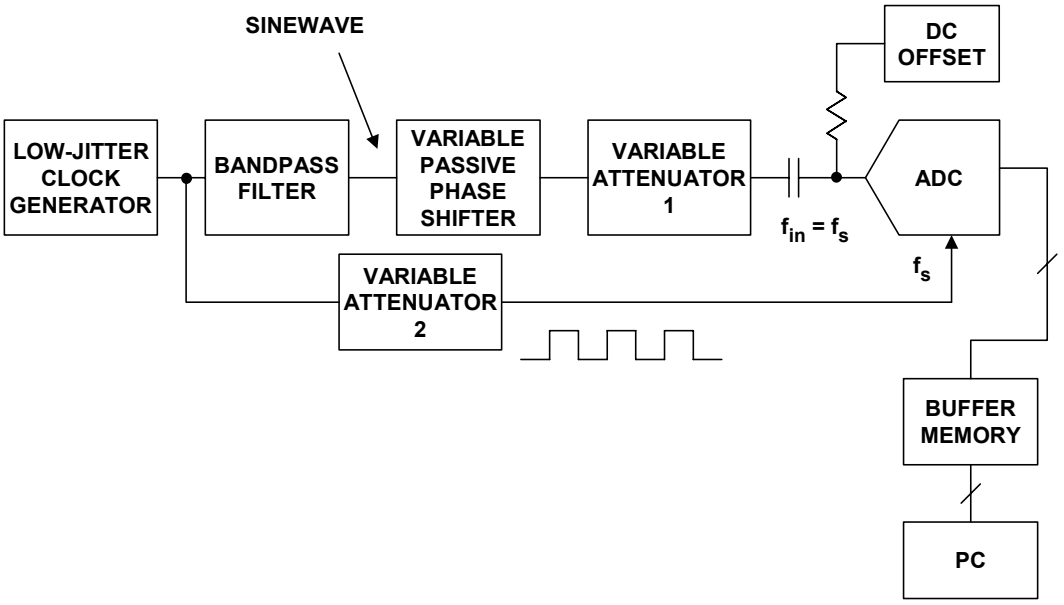

Figure 5.76: *Locked-Histogram Test Setup for Measuring ADC Aperture Jitter*

Next, the attenuation of Attenuator 1 is increased until the peak-to-peak input to the ADC is only exercising a few codes above and below mid-scale. The variable phase shifter is adjusted such that the mid-scale code occurs most of the time. Input-referred noise may cause a distribution of codes—if so, the PC software calculates the standard deviation of the distribution, σ_L, in LSBs. This corresponds to the input-referred noise expressed in LSBs. Attenuator 1 is then set for a full-scale input to the ADC. The variable phase shifter is adjusted until the mid-scale code has the highest probability of occurrence. The standard deviation of the new code distribution, σ_H, now includes the effects of input-referred noise as well as aperture jitter. The noise sources combine on an rss basis:

$$\sigma_H^2 = \sigma_L^2 + \sigma_A^2, \qquad \text{Eq. 5.31}$$

where σ_A is the rms noise (in LSBs) due to aperture jitter.

Equation 5.31 can be solved for σ_A as follows:

$$\sigma_A = \sqrt{\sigma_H^2 - \sigma_L^2}\ . \qquad \text{Eq. 5.32}$$

The full-scale input sinewave is given by $v_{in}(t) = V_{FS}\sin 2\pi f_{in} t$, where the ADC input range is $\pm V_{FS}$. The rate-of-change of the full-scale sinewave at the zero crossing is given by:

$$\left.\frac{dv}{dt}\right|_{max} = V_{FS} 2\pi f_{in}\,. \qquad \text{Eq. 5.33}$$

For a slope of $dv/dt|_{max}$, the rms aperture time, t_a, is related to the corresponding rms voltage error, Δv_{rms}, by the equation:

$$t_a = \frac{\Delta v_{rms}}{\left.\frac{dv}{dt}\right|_{max}} . \qquad \text{Eq. 5.34}$$

The rms noise due to aperture jitter in LSBs, σ_A, can be related to Δv_{rms} by:

$$\Delta v_{rms} = \sigma_A \cdot \frac{V_{FS}}{2^{N-1}} . \qquad \text{Eq. 5.35}$$

Substituting Equation 5.32, 5.35, and 5.33 into Equation 5.34:

$$t_a = \frac{\sqrt{\sigma_H{}^2 - \sigma_L{}^2}}{2\pi f_{in} \cdot 2^{N-1}} . \qquad \text{Eq. 5.36}$$

The calculations are summarized in Figure 5.77.

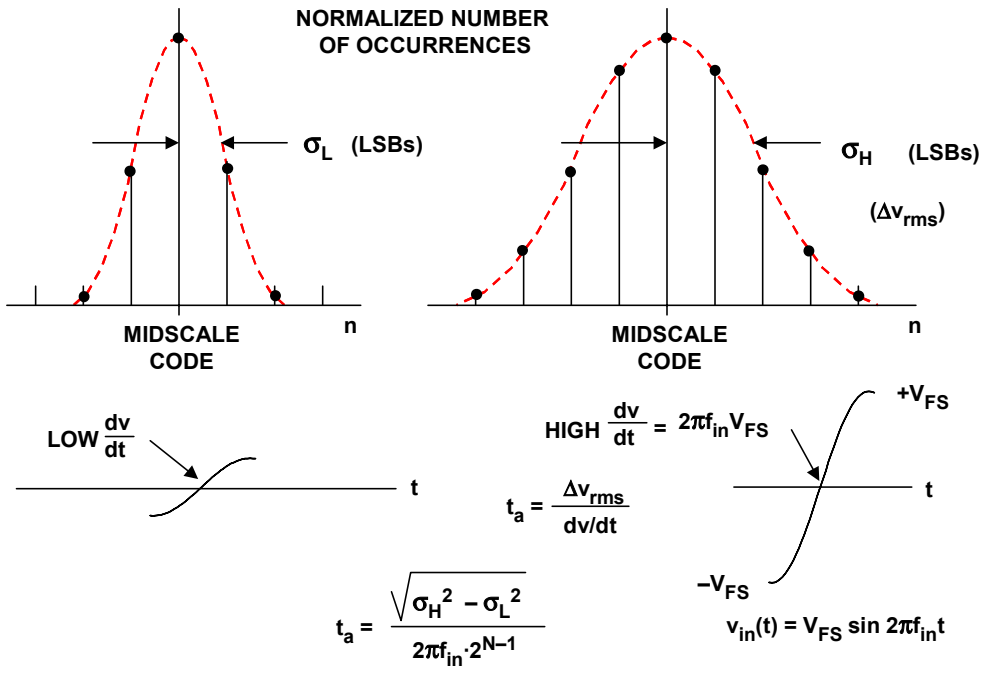

Figure 5.77: *Calculating Aperture Jitter Based on Locked-Histogram Test*

It should be noted that the test setup can be implemented using a low jitter sinewave signal generator to drive the ADC input and use the sync trigger pulse output of the same generator to drive the sampling clock input of the ADC. However, if aperture jitter measurements of less than 100-ps rms are expected, the jitter between the signal generator output and the trigger output may be of this same magnitude, thereby corrupting the overall measurement.

Measuring Aperture Delay Time

Aperture delay time can be measured with the same test setup used for the locked-histogram aperture jitter test, or the simpler test setup shown in Figure 5.78. The analog input frequency is locked to the sampling clock frequency, and a full-scale sinewave input applied to the ADC input. The delay in the sampling clock signal is adjusted until the PC histogram distribution of codes indicates that the ADC is being sampled at the zero-crossing of the sinewave, corresponding to the mid-scale code. The aperture delay is simply the difference between the 50% point of the leading edge of the sampling clock and the zero-crossing of the sinewave, measured with a dual-trace oscilloscope.

The aperture delay can be positive or negative as shown in Figure 5.78. The frequency of the sinewave input signal is not critical, but it should be high enough so that the small aperture delay time can be accurately measured. However, the frequency should not exceed the analog input bandwidth of the ADC. For convenience, a frequency of one-half the maximum ADC sampling frequency is a good starting point, and represents a reasonable upper limit.

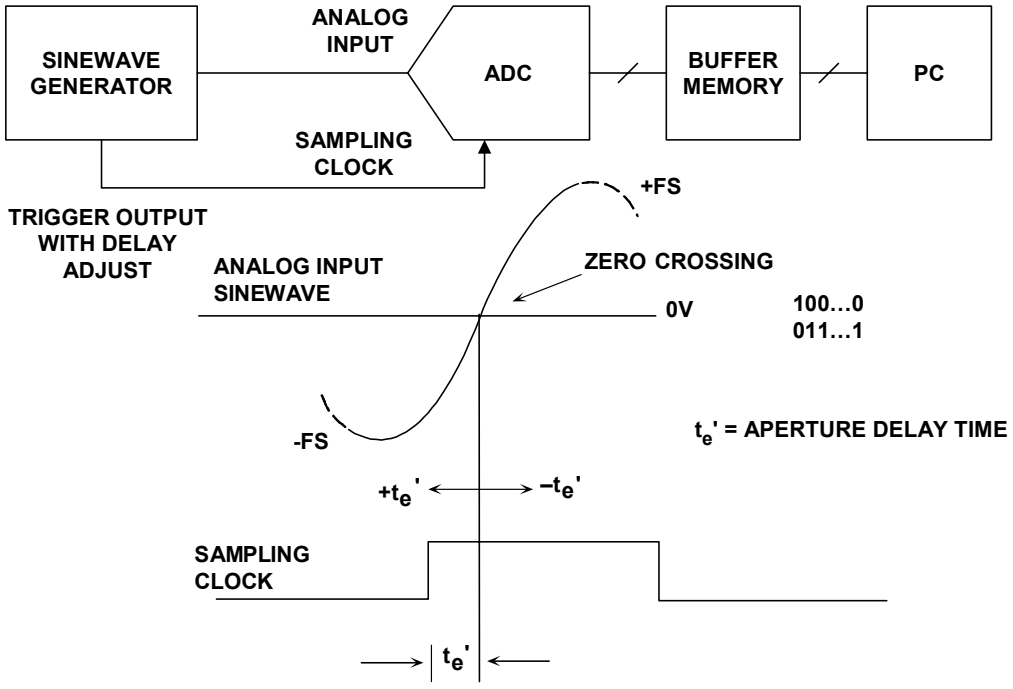

Figure 5.78: *Measuring Aperture Delay Time*

Measuring ADC Aperture Jitter Using FFTs

The FFT test routine for measuring ADC SNR is an excellent indirect method for measuring aperture jitter. The caveat in this test is that the measurement includes the jitter of the sampling clock generator as well as the ADC internal aperture jitter. Therefore, a generator should be selected with an rms jitter specification which is several times less than the specified aperture jitter of the ADC under test, since jitter combines on an rss

basis. The basic test setup for the aperture jitter test is shown in Figure 5.79 along with the key calculations.

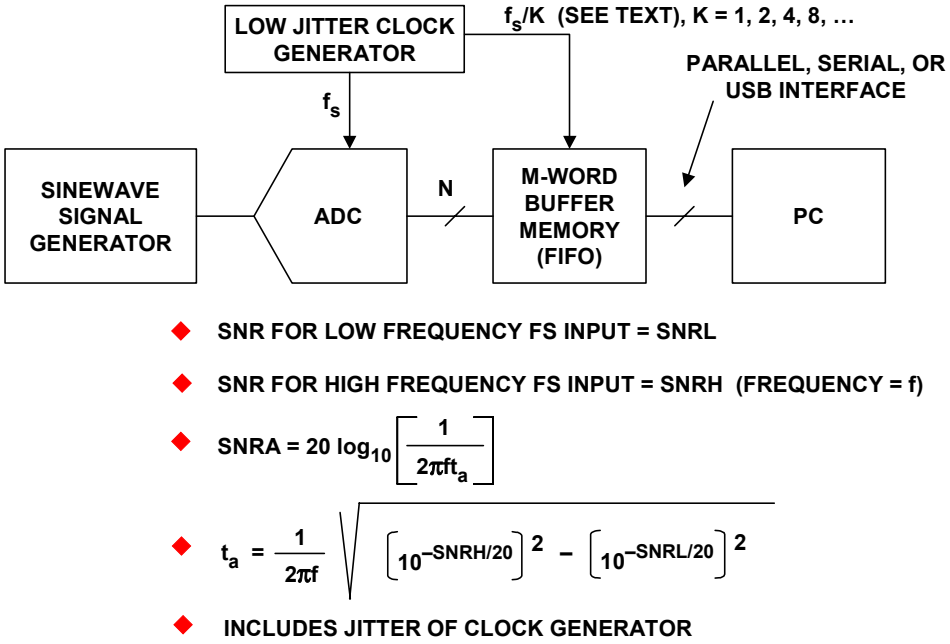

Figure 5.79: *Measuring Aperture Jitter Based on Degradation in SNR at High Frequencies*

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

$$SNRA = 20\log_{10}\left[\frac{1}{2\pi f t_a}\right], \qquad \text{Eq. 5.37}$$

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

$$t_a = \frac{1}{2\pi f}\cdot\frac{1}{10^{SNRA/20}}. \qquad \text{Eq. 5.38}$$

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

$$\left(\frac{1}{10^{SNRH/20}}\right)^2 = \left(\frac{1}{10^{SNRL/20}}\right)^2 + \left(\frac{1}{10^{SNRA/20}}\right)^2. \quad \text{Eq. 5.39}$$

Re-arranging Eq. 5.39:

$$\left(\frac{1}{10^{SNRA/20}}\right) = \sqrt{\left(\frac{1}{10^{SNRH/20}}\right)^2 - \left(\frac{1}{10^{SNRL/20}}\right)^2}. \quad \text{Eq. 5.40}$$

Substituting Equation 5.40 into Equation 5.38:

$$t_a = \frac{1}{2\pi f} \cdot \sqrt{\left(\frac{1}{10^{SNRH/20}}\right)^2 - \left(\frac{1}{10^{SNRL/20}}\right)^2}. \quad \text{Eq. 5.41}$$

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

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

Measuring ADC Analog Bandwidth Using FFTs

Analog input bandwidth can be easily measured by simply observing the FFT output display as the input frequency is swept from low to high frequency. The amplitude of the input signal must be held constant as the frequency is increased, and the relative amplitude of the FFT fundamental output signal is observed. When the amplitude of the fundamental signal in the FFT display drops 3 dB from the initial amplitude, that is defined as the 3-dB bandwidth. The measurement can be made with a full-scale input to obtain the full-power bandwidth (FPBW) or at a low amplitude to obtain the small-signal bandwidth. Notice that this definition of FPBW says nothing about the amount of distortion present in the FFT output at the FPBW frequency, which can be considerable for some ADCs.

In order to include the effects of distortion in the bandwidth measurement, a specification called *effective resolution bandwidth* (ERB) is sometimes used to define the input frequency at which the full-scale SINAD drops by 3-dB, corresponding to the loss of 0.5 ENOB. The ERB is easily measured by the same procedure described above if the calculated SINAD rather than the fundamental signal amplitude is used as the bandwidth criteria.

Settling time

Settling time can be measured with the test setup shown in Figure 5.80 based around a flat pulse generator and a synchronized adjustable delay clock for the ADC sampling clock input. The flat pulse generator is adjusted to give slightly less than a full-scale step input to the ADC, and the sampling clock delay is adjusted until the output has settled to within 1 LSB of the final value. The delay between the 50% point of the input pulse and the leading edge of the sampling clock can be measured with a dual-channel oscilloscope. If the peak-to-peak input-referred noise of the ADC is less than 1 LSB, a simple LED display can be used to observe the ADC output. If the input-referred noise is more than 1 LSB peak-to-peak, then a histogram may be required to determine the most frequently occurring output code. The sampling frequency does not necessarily need to be set to the maximum value, it can be reduced as required. Note that the aperture delay time must be subtracted from the measured settling time value.

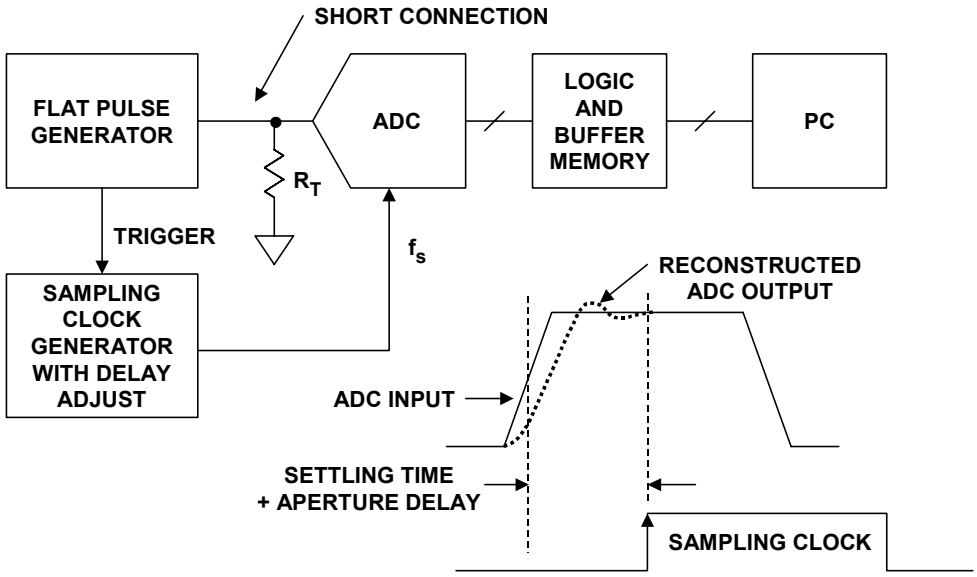

Figure 5.80: *Settling Time Test Setup*

The characteristics of the flat pulse generator must be known precisely, and the distance from its output to the ADC input should be kept to a minimum. A simple flat pulse generator is shown in Figure 5.81, based upon low-capacitance Schottky diodes. The input to the network at "A" is adjusted so that the start of the "B" waveform occurs at slightly above the negative full-scale range of the ADC. When the "A" voltage goes positive, diode D2 is reversed biased, and the voltage at "B" is flat, except for a small transient due to the reverse-bias diode capacitance and the small reverse leakage current that flows into the effective 100-Ω termination. The positive portion of the "A" voltage should be no more than required to reverse bias diode D2. Diode D1 ensures that the signal generator is terminated in a net 50-Ω impedance when the voltage at "A" is positive. Another diode can be placed in series with D2 in order to lower the effective coupling capacitance, and the negative portion of the "A" voltage adjusted as required to make up for the voltage drop of the extra diode. This network as shown delivers a ½ full-scale pulse to a bipolar ADC. Key to its success is keeping the diode-resistor network connections extremely short and close to the ADC input.

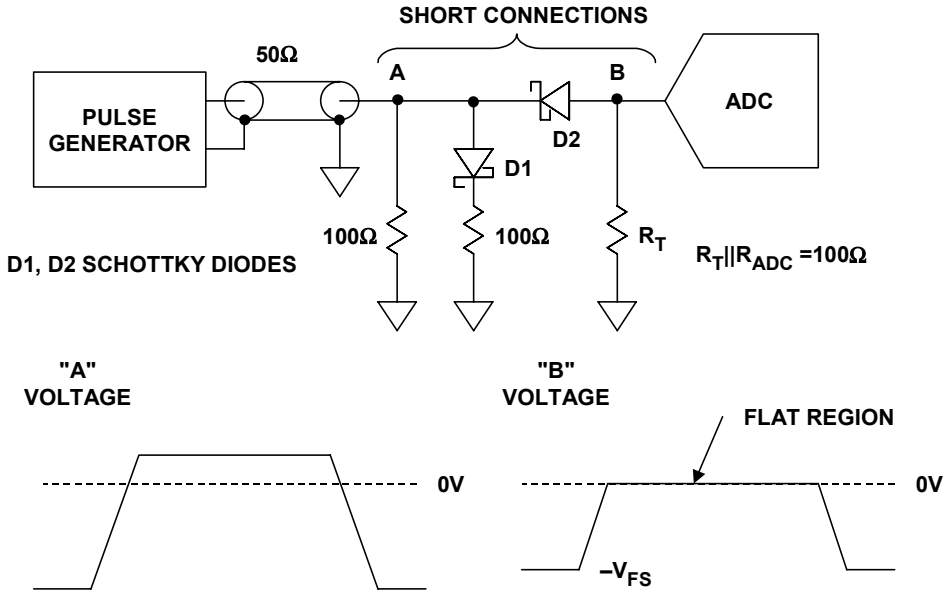

Figure 5.81: *A Simple Flat Pulse Generator*

Overvoltage Recovery Time

The test setup and method for overvoltage recovery is identical to that of the settling time setup, but the voltage at "A" is adjusted so that the start of the waveform is out of the ADC input range as shown in Figure 5.82. The amount of overvoltage is generally specified as a percentage of the ADCs range. For a converter with a 2-V peak-to-peak input range, 50% overvoltage would correspond to 1 V above or below the nominal 2-V range.

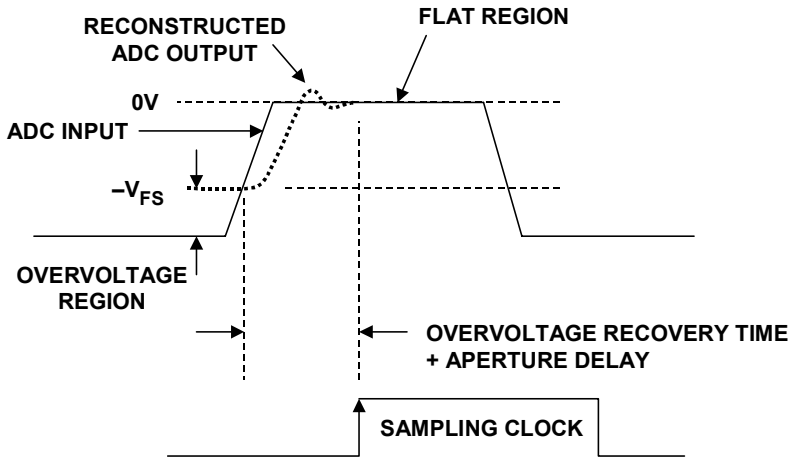

Figure 5.82: *Overvoltage Recovery Test Waveform from Flat Pulse Generator*

As in the transient response test, the aperture delay time must be considered when making the measurement.

The overvoltage recovery test can verify that the ADC behaves as an ideal limiter for signals outside its nominal range. The ADC should continue to read either all "0"s, or all "1"s (for offset binary coding) while the input signal is outside the range.

Video Testing, Differential Gain and Differential Phase

The development of board-level and modular 8-bit high-speed ADCs in the early 1970s led to widespread interest in using digital techniques in traditionally analog video equipment, such as time base correctors, frame synchronization and storage, standards conversion, on-line monitoring of television signals, digital special effects, and image enhancement (see extensive bibliography in Reference 18). Most of these digital "black boxes" contained an ADC, digital processing of some sort, followed by a reconstruction DAC. Therefore traditional video test equipment could be used to check the performance of the ADC and DAC back-to-back, acting as a *codec* (coder-decoder) as shown in Figure 5.83.

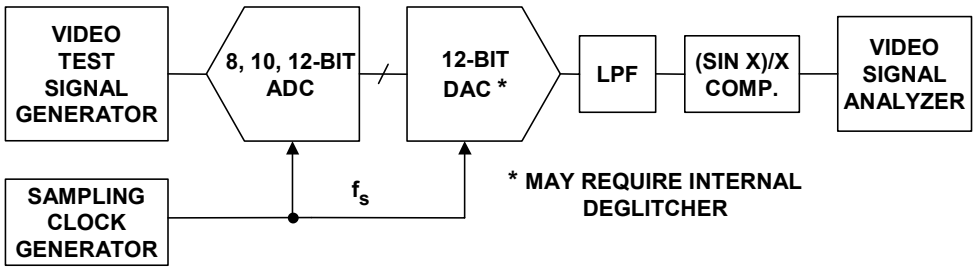

Figure 5.83: *Composite Video Test Setup Using Analog Methods*

In a composite video signal (see Figure 5.84) the color information is contained in the chrominance (color) signal and which is superimposed on the luminance (black and white) signal. In NTSC the color subcarrier frequency is approximately 3.58 MHz, in PAL it is approximately 4.43 MHz. The amplitude of the chrominance signal determines the color saturation, and the phase of the chrominance signal (with respect to the color burst) determines the actual color.

Differential gain and differential phase are two of the most important specifications in composite video applications. Differential gain is defined as the variation in amplitude (in percent) of a small amplitude subcarrier signal as it is swept across the video range from black to white. Differential phase is defined as the phase variation (in degrees) of a small amplitude subcarrier signal as it is swept across the video range from black to white. These are important because differential gain errors will distort the degree of color saturation and differential phase errors will cause incorrect hues in the picture (per the definitions above).

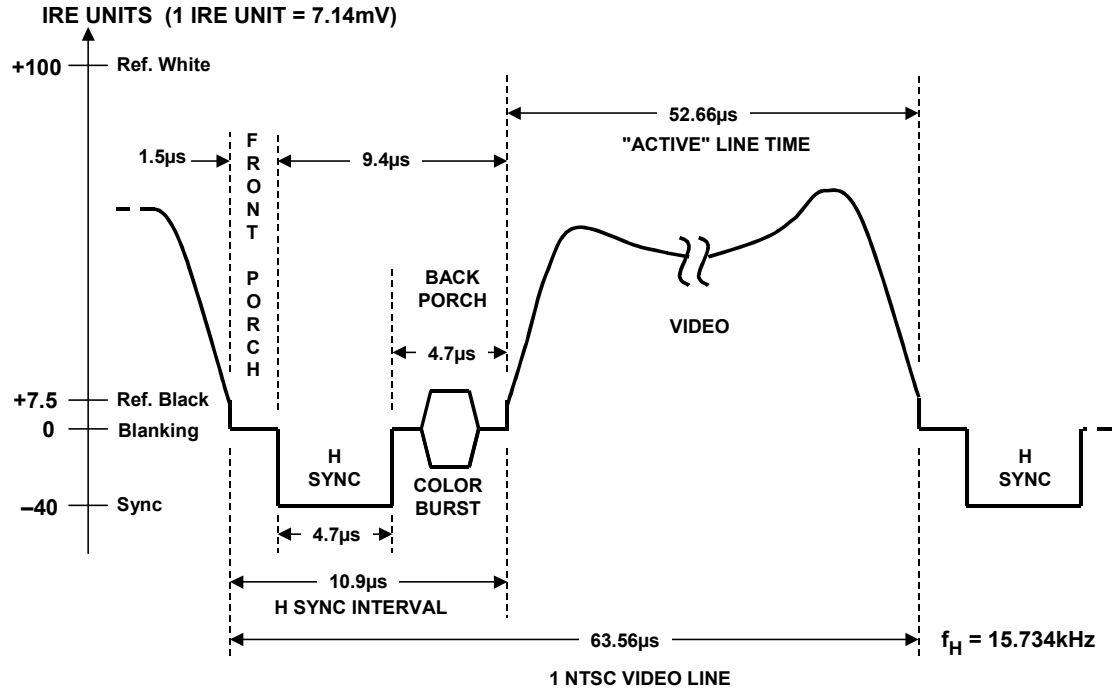

Figure 5.84: *NTSC Composite Color Video Line*

Some of the popular differential gain and phase video test signals are shown in Figure 5.85. All of them make use of either a staircase or a ramp that is modulated by the color subcarrier. Any of these signals are fine for testing purely analog video systems, but obtaining accurate test results in a system which contains a video codec presents a special challenge. This is because the nature of the test signal causes the quantization noise to give misleading results. The situation is exacerbated because in composite video applications it is very common to operate the sampling clock at exactly 4 times the color subcarrier frequency (leading to repetitive quantization error patterns as previously discussed).

For instance, consider the 10-step staircase test signal with 20-IRE modulation. This corresponds typically to 30 quantization levels p-p for an 8-bit system with headroom. If the test signal is sampled at exactly 4 times the subcarrier frequency, and a particularly unfavorable combination of sampling phase and dc level are present, a quantization error of 1 LSB in the amplitude measurement can result in any amplitude measurement on any single step of the staircase. In fact, one step can measure 1 LSB high, and the next 1 LSB low, resulting in a possible differential gain error of 2 LSBs, or 2/30 = 6.7%—even for a perfect 8-bit codec. Early differential gain and phase measurements of this type in the 1970s led to displays on vectorscopes such as those shown in Figure 5.86A for the 10-step, 20-IRE modulated staircase.

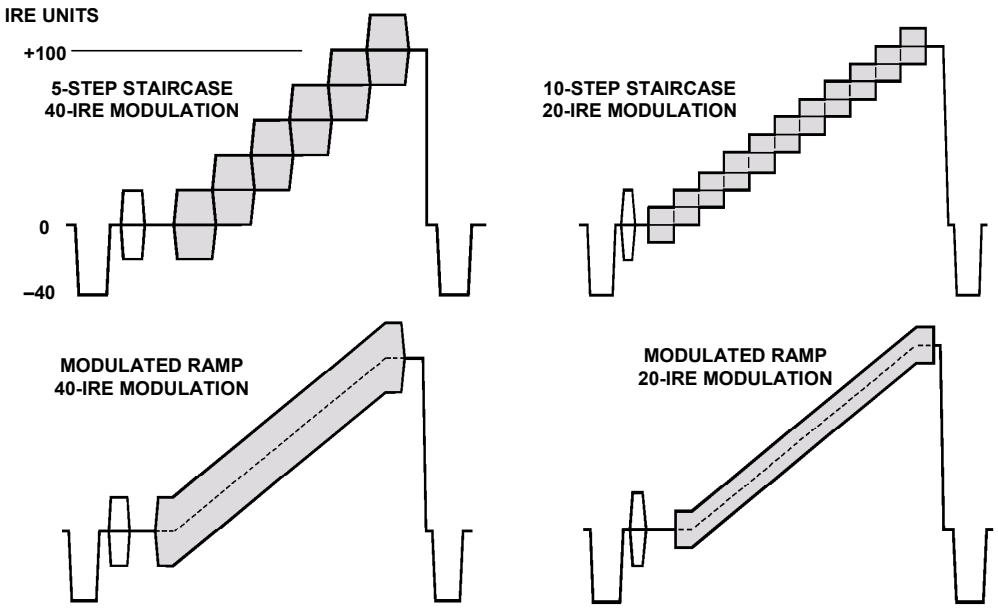

Figure 5.85: *Some Differential Gain and Phase Video Test Signals*

Notice that the differential gain and phase error is approximately 7% and 2°, respectively, and is mostly due to the quantization errors. This problem was analyzed by Felix (Reference 43) who predicted a theoretical worst case 8-bit differential gain and phase error of 8% and 5°, respectively for the 20-IRE unit modulated staircase test signal. Theoretically, a 9-bit system would yield differential gain and phase measurements of approximately 4% and 2.5°, and a 10-bit system 2% and 1.25°.

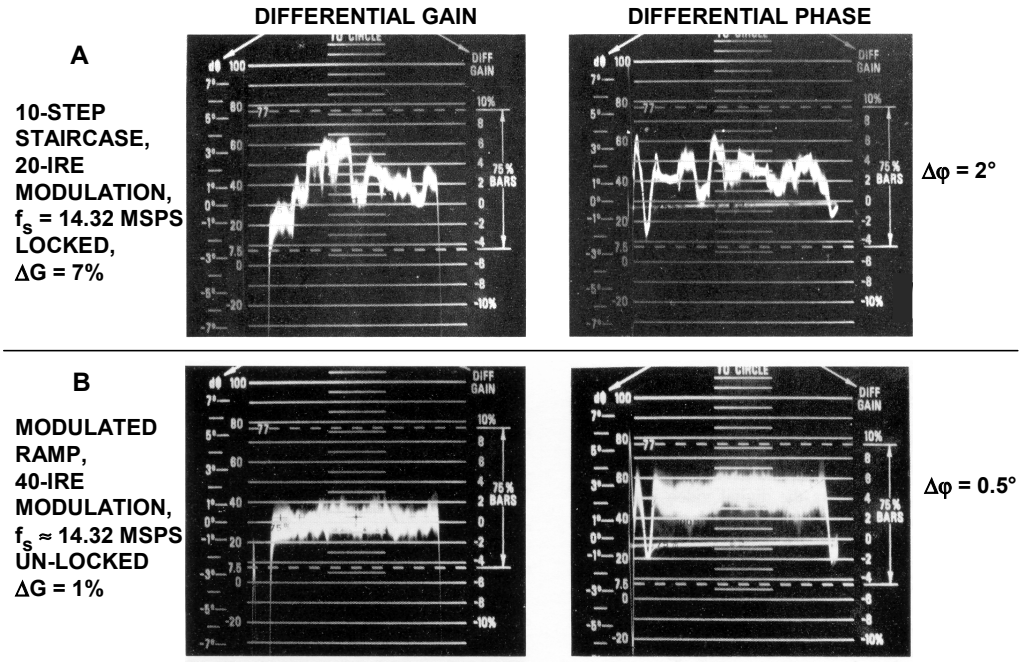

Figure 5.86: *Mid-1970s Vectorscope Differential Gain and Phase Measurements for 8-Bit ADC*

Several things can be done to help the differential gain and phase measurement problem as recommended in IEEE Standard 746-1984 (Reference 28). Selecting 40-IRE unit modulation immediately divides the theoretical worst case differential gain and phase numbers by a factor of two, since the quantization error is now a smaller percentage of the modulation. Using the 40-IRE unit modulated ramp test signal rather than the modulated staircase is preferred, because the continuously changing level of a ramp helps to integrate the discontinuities of the quantization levels. Finally, unlocking the sampling frequency from the color subcarrier helps to randomize the quantization levels. Figure 5.86B shows the results of using the 40-IRE unit modulated ramp with an unlocked sampling clock, where the "true" differential gain and phase measurements are now approximately 1% and 0.5°, respectively.

It has also been suggested in an ITU recommendation (Reference 44) that a 30-mV p-p "dither" sinewave at a frequency of 5.162 MHz for NTSC and 6.145 MHz for PAL be summed with the test signal input to the ADC. These frequencies are outside the normal cutoff frequencies of the systems (4.2 MHz for NTSC and 5.0 MHz for PAL). The dither frequencies fall at the second-null points of the respective lowpass filters. Most video signal measuring instruments incorporate various types of filters for easy and accurate measurements. These are usually efficient in removing the dither signals, but if they are not sufficient, then external filters must be added.

An all-digital method for differential gain and phase ADC testing is recommended in IEEE Standard 1241-2000 (Reference 37). The test setup is shown in Figure 5.87, and a recommended test signal is shown in Figure 5.88.

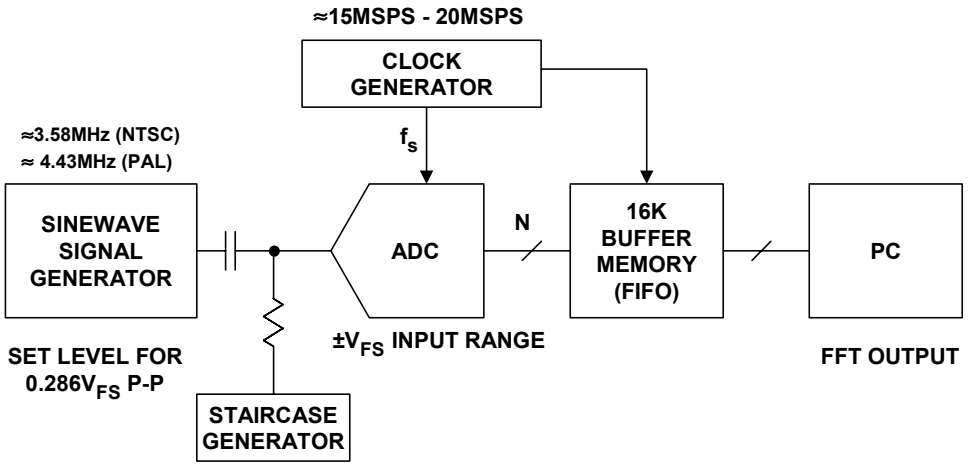

Figure 5.87: *Digital Differential Gain and Phase Measurements*

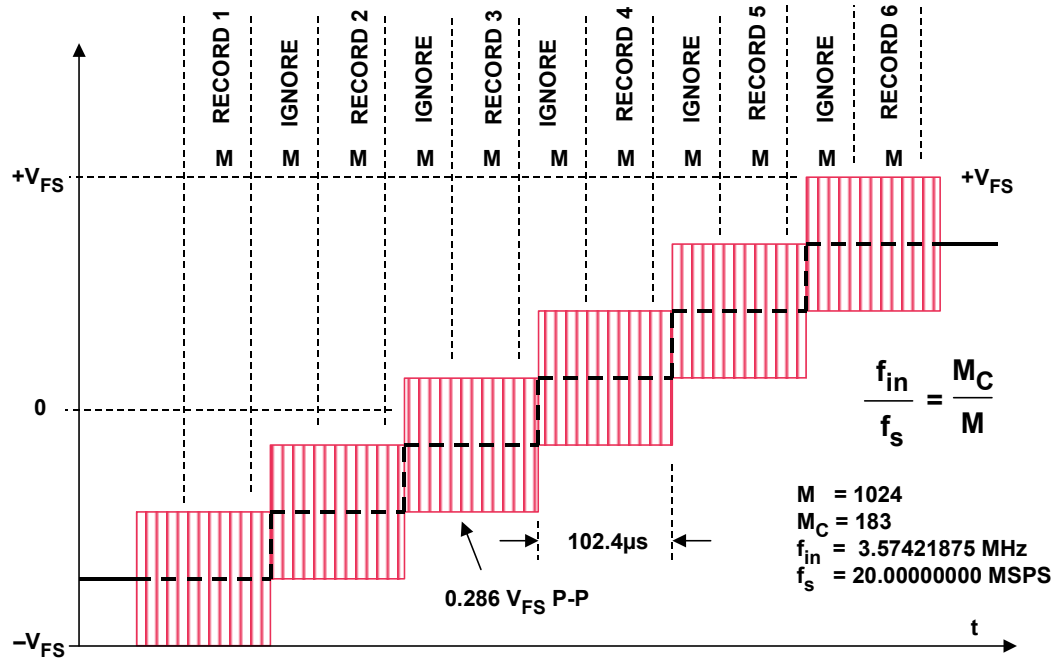

Figure 5.88: *Test Signal for Digital Differential Gain and Phase Test*

The sinewave generator output and the staircase generator output are combined to produce the composite ADC input waveform. The sampling frequency, f_s, input frequency, f_{in}, record length, M, and number of cycles within the record, M_C,are all chosen to satisfy the condition for coherent sampling:

$$\frac{f_{in}}{f_s} = \frac{M_C}{M}. \qquad \text{Eq. 5.42}$$

Notice that 11 records of M = 1024 are taken during the duration of the test waveform, for a total buffer memory requirement of 11,264. Only the records corresponding to the settled waveforms are used in computing the 6 individual FFTs—the others are ignored, since the data during these intervals is not valid because they occur during the changes of the staircase and the associated settling time. Choosing f_{in}, f_s, M, and M_C is somewhat arbitrary, but the numbers shown in Figure 5.88 approximate video conditions. The starting point is to fix the sampling frequency f_s = 20 MSPS, and the record length M = 1024 (recall M must be a power of 2). M_C should be a prime number, and letting M_C = 183, the resulting input frequency is 3.57421875 MHz, which is close to the NTSC color subcarrier frequency of 3.579545 MHz. The width of each step of the staircase should correspond to 2048 samples taken at 20 MSPS, or 102.4 µs.

The buffer memory is loaded with 11,264 samples relative to the waveform as shown. A total of 11 FFT records are taken, each record containing 1024 points, and the 5 records corresponding to the waveform transition and settling time intervals are ignored. Only 6 FFTs are actually computed, and the amplitude and phase of the signals in each FFT can be compared to calculate the differential gain and phase error. Ideally, the phase of each

of the 6 amplitudes should be equal because there is exactly 1 record of unused data between each of the records used in calculating the FFTs, i.e., the start of each record corresponds to exactly the same point on the input sinewave.

Bit Error Rate (BER)Tests

The concept of errors caused by metastability in ADCs has been discussed at length in Chapter 2. This section concentrates on the test aspects of the resulting bit error rate (BER). The test system shown in Figure 2.89 can 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.

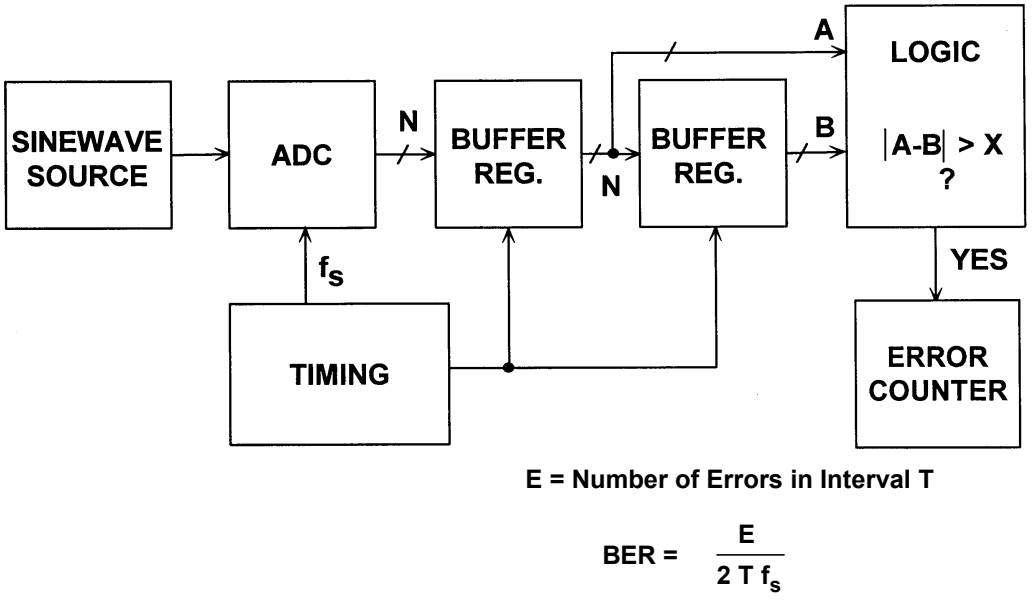

Figure 2.89: *ADC Bit Error Rate Test Setup*

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 is therefore incremented twice for each error. It should be noted that the same function can be accomplished in software if the ADC outputs are stored in a memory and analyzed by a computer program.

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

$$v(t) = \frac{2^N}{2} \sin 2\pi ft \qquad \text{Eq. 5.43}$$

The maximum rate of change of this signal is

$$\left.\frac{dv}{dt}\right]_{max} \leq 2^N \pi f \,. \qquad \text{Eq. 5.44}$$

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

$$f_{in} \leq \frac{f_s}{2^N \pi}. \qquad \text{Eq. 5.45}$$

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

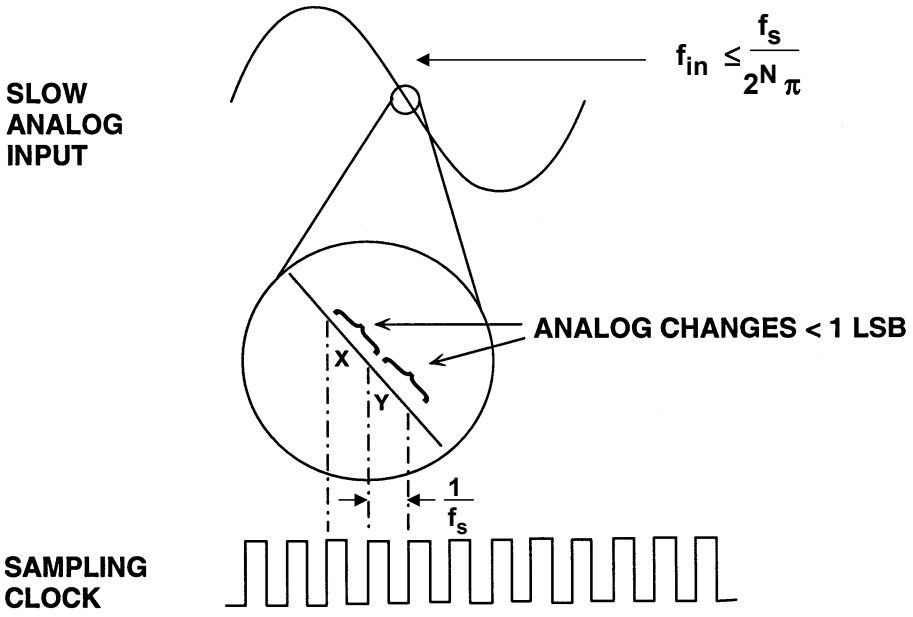

Figure 2.90: *ADC Analog Signal for Low Frequency BER Test*

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.

$$v(t) = \frac{2^N}{2} \sin\left[2\pi \left(\frac{f_s}{2} - \Delta f \right) t \right]. \qquad \text{Eq. 5.46}$$

The maximum rate of change of this signal is

$$\left. \frac{dv}{dt} \right]_{max} \leq 2^N \pi \left(\frac{f_s}{2} - \Delta f \right). \qquad \text{Eq. 5.47}$$

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

$$\Delta f \leq \frac{f_s}{2} \left(1 - \frac{1}{2 \cdot 2^N \pi} \right). \qquad \text{Eq. 5.48}$$

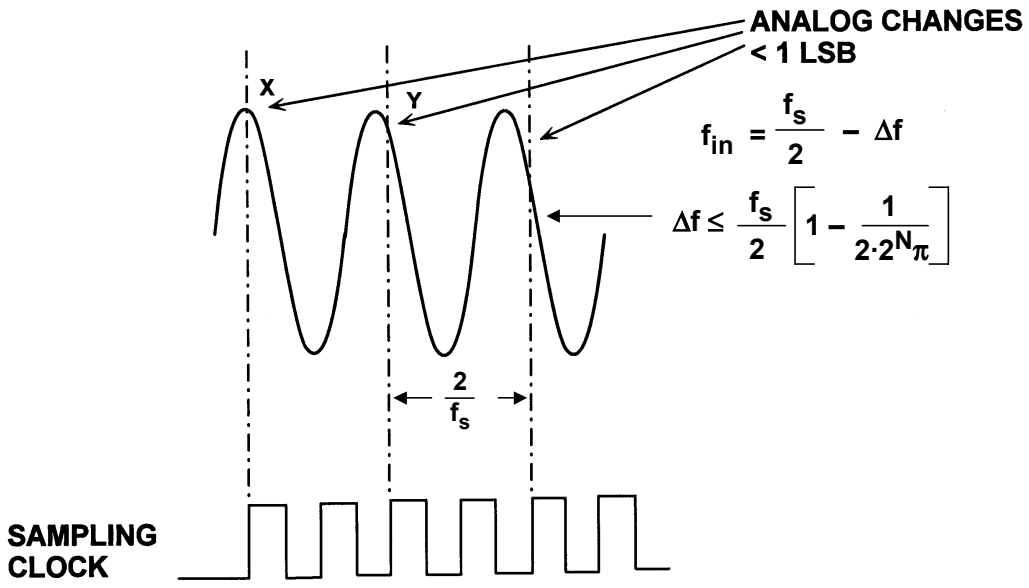

Figure 2.91: *ADC Analog Input for High Frequency BER Test*

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. Measuring low BER therefore requires long measurement times which increase the probability of power supply transients, noise, etc. causing a false error. Meaningful tests for long periods of time require special attention to EMI/RFI effects (possibly requiring a shielded screen room), isolated power supplies, etc.

Ideally, the BER test requires that each contiguous sample of the ADC output be analyzed, thereby making it difficult to implement using a buffer memory and a PC as in a typical FFT-based ADC test setup. Errors are missed which occur during the time the buffer memory output is being read by the PC. However, this method can be used provided the memory size is large (preferably 256K), and a large number of data records

are taken as shown in Figure 5.92. The "dead time" used by the PC to read the data from the memory and analyze it will add to the total test time, especially if the BER is very low.

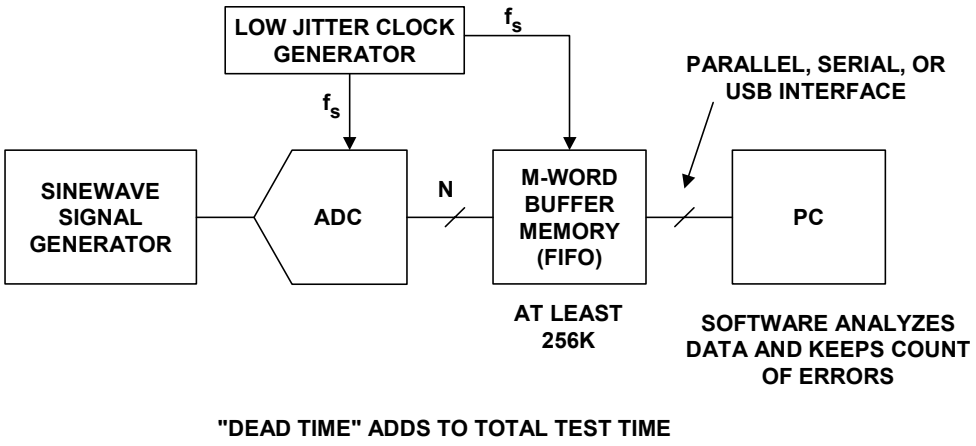

Figure 5.92: *Alternate Test Setup for PC-Based BER Test*

REFERENCES:
5.2 TESTING ADCs

The following references have been placed in their approximate chronological order so the reader can gain a good historical perspective on the evolution of specifications and testing methods for ADCs. Most of the references are referred to in the text, however some are included simply as additional background material.

1. Bernard M. Gordon, "Definition of Accuracy of Voltage to Digital Converters," **Instruments and Control Systems**, May 1959, pp.710.

2. W. M. Gaines, "Terminology for Functional Characteristics of Analog to Digital Converters," **Control Engineering**, February 1961.

3. Bernard M. Gordon, "Designing Sampled Data Systems," **Control Engineering**, April 1961, pp. 127-132.

4. Bernard M. Gordon, "How to Specify Analog-to-Digital Converters," **Electronic Design**, May 10 1961, pp. 36-39.

5. Bernard M. Gordon, "How Much Do Components Limit Converter Performance?" **Electronic Design**, June 21, 1961, pp.52-53.

6. P. Barr, "Influence of Aperture Time and Conversion Rate on the Accuracy of A/D Converters," **Data Systems Engineering**, May 1964, pp. 30-34.

7. A. Van Doren, "Solving Error Problems in Digital Conversion Systems," **Electromechanical Design**, April 1966, pp.44-46.

8. J. Freeman, "Specifying Analog to Digital Converters," **The Electronic Engineer**, June 1968, pp. 44-48.

9. Bernard M. Gordon, "Speaking Out on Analog to Digital Converters," **EEE Magazine**, December 1968.

10. Bernard M. Gordon, "Bernard Gordon of Analogic Speaks Out on What's Wrong with A/D Converter Specs," **EEE Magazine**, February 1969, pp. 54-61.

11. Dan Sheingold, **Analog-Digital Conversion Handbook**, Analog Devices, 1972.

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

13. Bill Pratt, "Test A/D Converters Digitally," **Electronic Design**, December 6, 1975.

14. Stuart K. Tewksbury, F.C. Meyer, D.C. Rollenhagen, H.K. Schownwetter, and Thomas Souders, "Terminology Related to the Performance of S/H, A/D, and D/A Circuits," **IEEE Transactions on Circuits and Systems**, Vol. CAS-25, July 1978, pp. 419-426.

15. Eugene L. Zuch, **Data Acquisition and Conversion Handbook**, Datel-Intersil, Inc., 1979, ISBN 0-9602946-0-0.

16. Jim R. Naylor, "Testing Digital/Analog and Analog/Digital Converters," **IEEE Transactions on Circuits and Systems**, Vol. CAS-25, July 1978, pp. 526-538.

17. Walter A. Kester, "Characterizing and Testing A/D and D/A Converters for Color Video Applications," **IEEE Transactions on Circuits and Systems**, Vol. CAS-25, July 1978, pp. 539-550.

18. W. A. Kester, "PCM Signal Codecs for Video Applications," **SMPTE Journal**, Number 88, November 1979, pp. 770-778.

19. Tim Wilhelm, "Test A/D Converters Quickly and Efficiently, **Electronic Design**, October 15, 1981, pp. 193-198.

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

21. Martin Neil and Art Muto, "Tests Unearth A/D Converter's Real-World Performance," **Electronics**, February 24 1982, pp. 127-132.

22. Bruce E. Peetz, Arthur S. Muto, and Martin Neil, "Measuring Waveform Recorder Performance," **Hewlett-Packard Journal**, Vol. 33, No. 11, November 1982, pp. 21-29.

23. Walter Kester, "Test Video A/D Converters Under Dynamic Conditions," **EDN**, August 18, 1982.

24. Joey Doernberg, Hae-Seung Lee, and David A. Hodges, "Full-Speed Testing of A/D Converters," **IEEE Journal of Solid State Circuits**, Vol. SC-19, No. 6, December 1984, pp. 820-827.

25. T. Michael Souders, Donald R. Flach, and Thick C. Wong, "An Automatic Test Set for the Dynamic Characterization of A/D Converters," **IEEE Transactions on Instrumentation and Measurement**, Vol. IM-32, No. 1, March 1983, pp. 180-186.

26. 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).*

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

28. **IEEE Std. 746-1984, IEEE Standard for Performance Measurements of A/D and D/A Converters for PCM Television Video Circuits**, IEEE, 1984.

29. Walt Kester, "Designer's Guide to Flash ADC Testing Part 1: Flash ADCs Provide the Basis for High-Speed Conversion," **EDN**, January 4, 1990, pp. 101-110.

30. Walt Kester, "Designer's Guide to Flash ADC Testing Part 2: DSP Test Techniques Keep Flash ADCs in Check," **EDN**, January 18, 1990, pp. 133-142.

31. Walt Kester, "Designer's Guide to Flash ADC Testing Part 3: Measure Flash ADC Performance for Trouble-Free Operation, **EDN**, February 1, 1990, pp. 103-114.

32. Michael J. Demler, **High-Speed Analog-to-Digital Conversion**, Academic Press, Inc., 1991, Chapter 6.

33. Walt Kester, "Designer's Guide to Sampling A/D Converters Part 1: Basic Characteristics Distinguish Sampling A/D Converters, **EDN**, September 3, 1992, pp. 135-144.

34. Walt Kester, "Designer's Guide to Sampling A/D Converters Part 2: Peripheral Circuits can Make or Break Sampling-ADC Systems, **EDN**, October 1, 1992, pp. 97-105.

35. Walt Kester, "Designer's Guide to Sampling A/D Converters Part 3: Layout, Grounding, and Filtering Complete Sampling ADC System, **EDN**, October 15, 1992, pp. 127-134.

36. **IEEE Std. 1057-1994 (R2001), IEEE Standard for Digitizing Waveform Recorders**, IEEE, 1994, ISBN 1-55937-488-8.

37. **IEEE Std. 1241-2000, IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters**, IEEE, 2001, ISBN 0-7381-2724-8.

38. Jerome Blair, "Histogram Measurement of ADC Nonlinearities Using Sine Waves," IEEE Transactions on Instrumentation and Measurement, Vol. 43, No. 3, June 1994, pp. 373-383.

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

40. Steven W. Smith, **Digital Signal Processing: A Practical Guide for Engineers and Scientists**, Newnes (An Imprint of Elsevier Science), 2002, ISBN 0-75067-444-X.

41. R. Jacob Baker, Harry W. Li, and David E. Boyce, **CMOS Circuit Design, Layout, and Simulation**, Wiley-IEEE Press, 1997, ISBN 0-78033-416-7, Chapter 28, 29.

42. R. Jacob Baker, **CMOS Mixed-Signal Circuit Design**, John Wiley and Sons, 2002, ISBN 0-4712-2754-4.

43. Micheal O. Felix, "Differential Phase and Gain Measurements in Digital Video Signals," **SMPTE Journal**, Vol. 85, February 1976, pp. 76-79.

44. "Measuring Methods for Digital Video Equipment with Analogue Input/Output," **ITU Recommendation ITU-R BT.1204**, 1995, http://www.itu.int/publications/bookshop.

Additional References on FFTs and DSP

45. C. Britton Rorabaugh, **DSP Primer**, McGraw-Hill, 1999.

46. Richard J. Higgins, **Digital Signal Processing in VLSI**, Prentice-Hall, 1990.

47. V. Oppenheim and R. W. Schafer, **Digital Signal Processing**, Prentice-Hall, 1975.

48. L. R. Rabiner and B. Gold, **Theory and Application of Digital Signal Processing**, Prentice-Hall, 1975.

49. John G. Proakis and Dimitris G. Manolakis, **Introduction to Digital Signal Processing**, MacMillian, 1988.

50. Fredrick J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," **Proc. IEEE**, Vol. 66, No. 1, 1978 pp. 51-83.

51. R. W. Ramirez, **The FFT: Fundamentals and Concepts**, Prentice-Hall, 1985.

52. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series," **Mathematics Computation**, Vol. 19, pp. 297-301, April 1965.

FFT and Engineering Data Analysis Software Packages

53. MATLAB®, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, Phone: 508-647-7000, http://www.mathworks.com.

54. LabVIEW™, National Instruments Corporation, 11500 N. Mopac Expressway, Austin, Texas 78759-3504, Phone: 512-683-0100, http://www.ni.com.

55. Mathcad, Mathsoft Engineering & Education, Inc., 101 Main Street, Cambridge, MA 02142-1521, Phone: 617-444-8000, http://www.mathsoft.com.

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: