Find Those Elusive ADC Sparkle Codes and Metastable States

by Walt Kester

INTRODUCTION

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

SPARKLE CODES, ERROR CODES, RABBITS, OR FLYERS

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

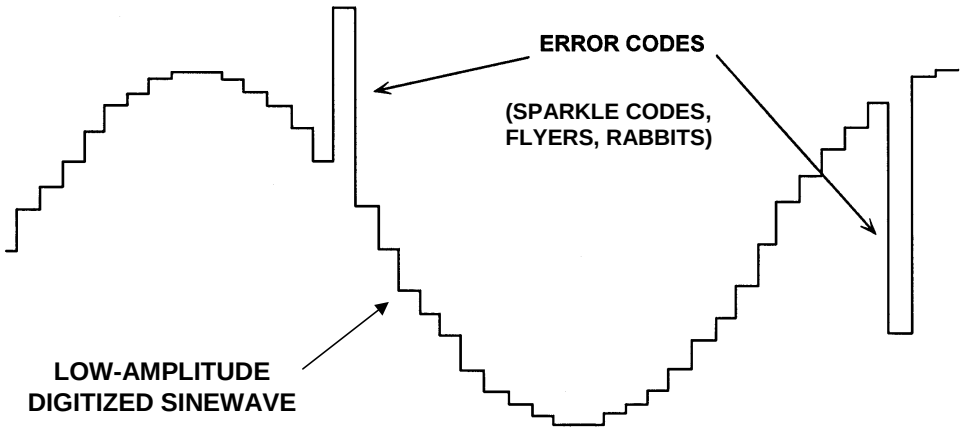

Figure 1: Exaggerated Output of ADC Showing Error Codes

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

SPARKLE CODES IN FLASH CONVERTERS

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

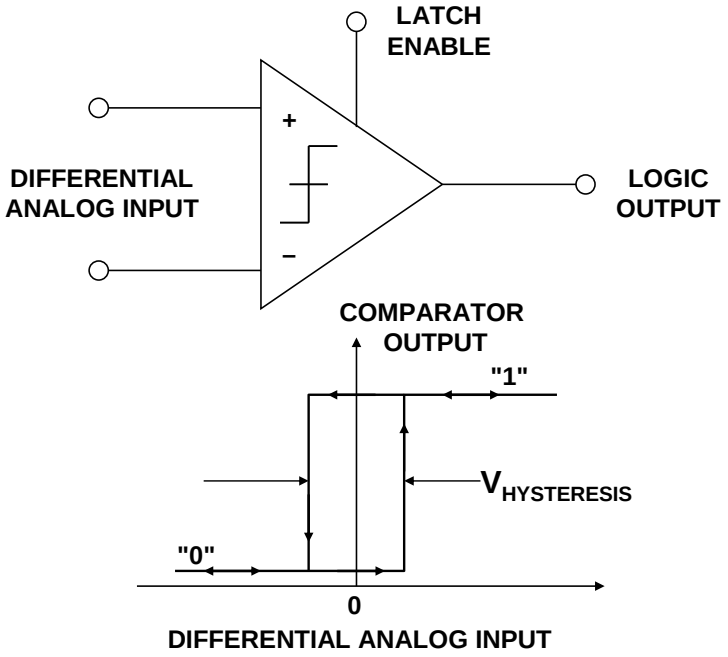

Figure 2: Latched Comparator

Early comparators were designed with vacuum tubes and were often used in radio receivers—where they were called *discriminators*, not comparators. Most modern comparators used in ADCs include a built-in latch which makes them sampling devices suitable for data converters. A typical structure is shown in Figure 3 for the AM685 ECL (emitter-coupled-logic) latched comparator introduced in 1972 by Advanced Micro Devices, Inc. (see Reference 1).

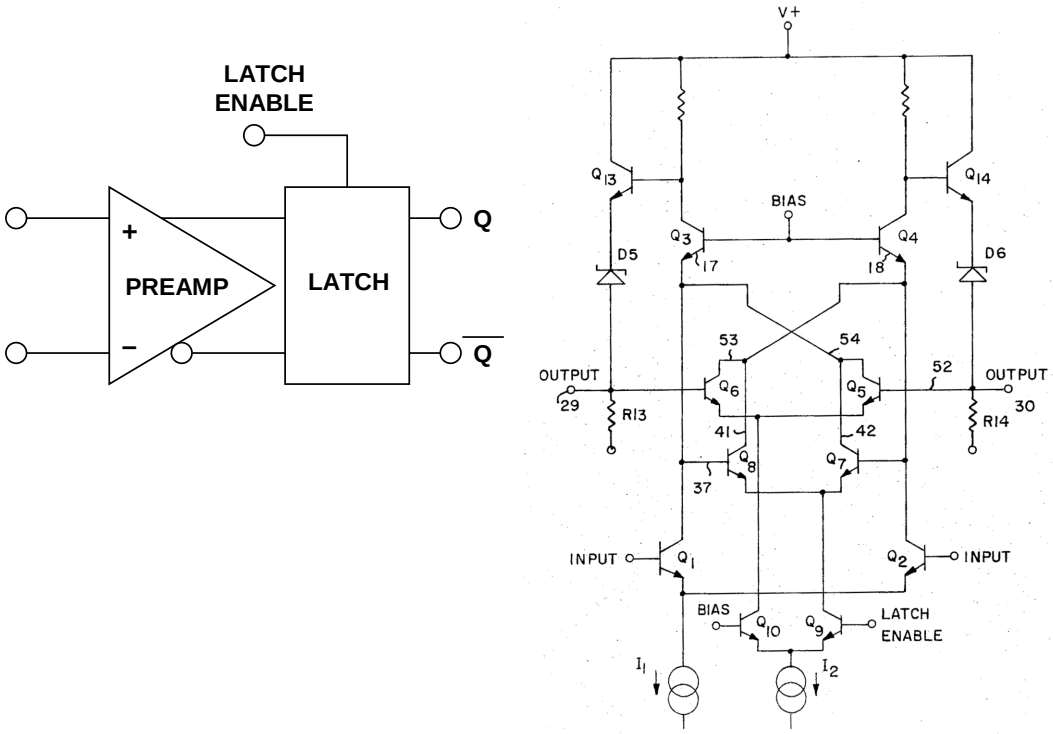

Figure 3: The AM685 ECL Comparator (1972)

The input stage preamplifier drives a cross-coupled latch. The latch locks the output in the logic state it was in at the instant when the latch was enabled. The latch thus performs a track-and-hold functon, allowing short input signals to be detected and held for further processing. Because the latch operates directly on the input stage, the signal suffers no additional delays—signals only a few nanoseconds wide can be acquired and held. The latched comparator is also less sensitive to instability caused by local feedback than an unlatched one.

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

The metastable state problem is illustrated in Figure 4. Three conditions of differential input voltage are illustrated: (1) large differential input voltage, (2) small differential input voltage, and (3) zero differential input voltage.

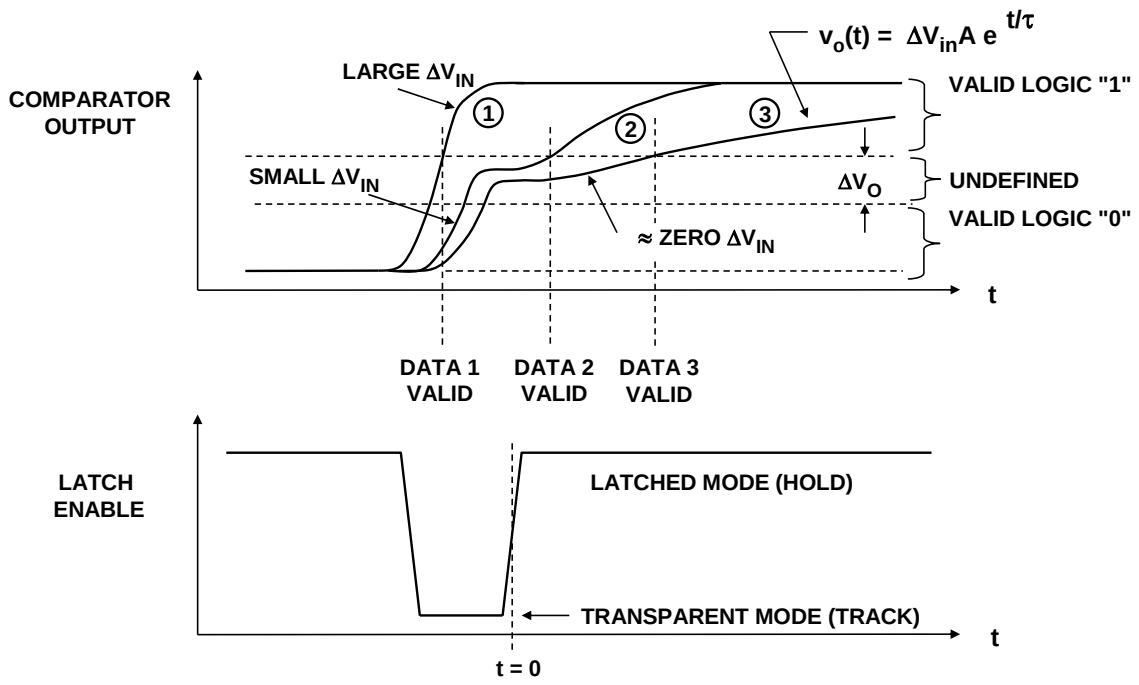

Figure 4: Comparator Metastable State Errors

The approximate equation which describes the output voltage, $V_O(t)$ is given by:

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

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

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

From a design standpoint, comparator metastability can be minimized by making the gain (A) high, minimizing the regeneration time constant (τ) by increasing the gain-bandwidth of the latch, and allowing sufficient time (t), for the output of the comparator to settle to a valid logic level. It is not the purpose of this discussion to analyze the complex tradeoffs between speed,

power, and circuit complexity when optimizing comparator designs, but an excellent treatment of the subject can be found in References 2 and 3.

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

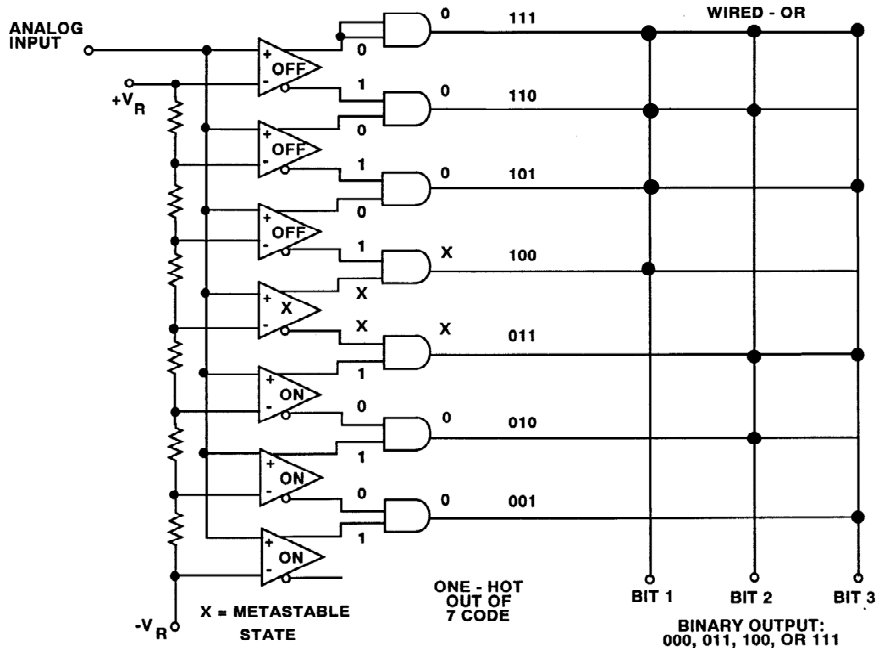

Figure 5: Metastable Comparator Output States May Cause Error Codes in Data Converters

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

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

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

The test system shown in Figure 6 may be used to test for BER in an ADC. The analog input to the ADC is provided by a high stability low noise sinewave generator. The analog input level is set slightly greater than full-scale, and the frequency such that there is always slightly less than 1-LSB change between samples as shown in Figure 7.

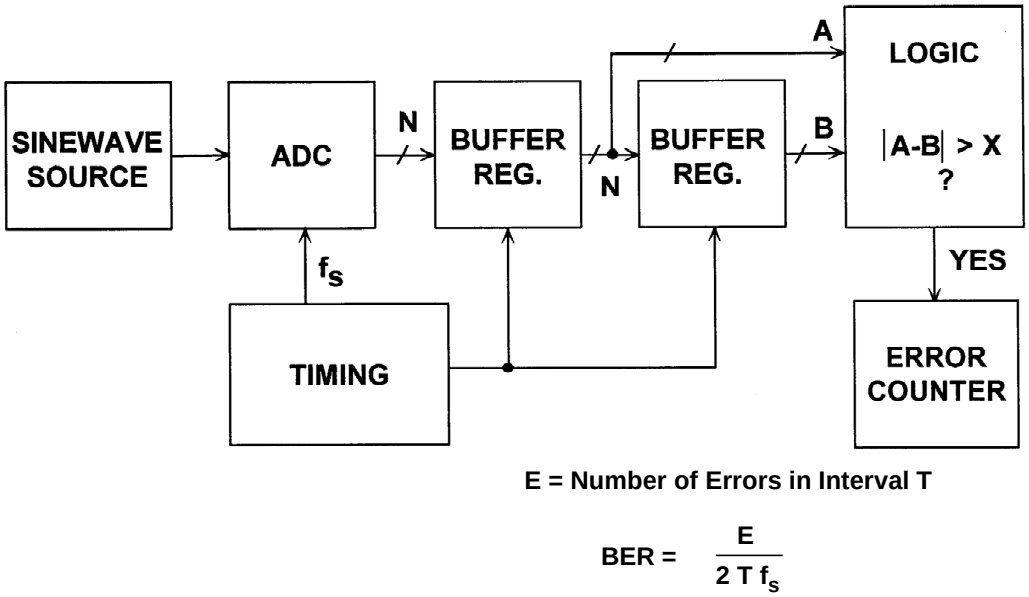

Figure 6: ADC Bit Error Rate Test Setup

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

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

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

The maximum rate of change of this signal is

$$\left.\frac{dv}{dt}\right]_{max} \leq 2^N \pi f \, . \qquad \text{Eq. 3}$$

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

$$f_{in} \leq \frac{f_s}{2^N \pi} . \qquad \text{Eq. 4}$$

Choosing an input frequency less than this value will ensure that there is at least one sample per code as shown in Figure 7.

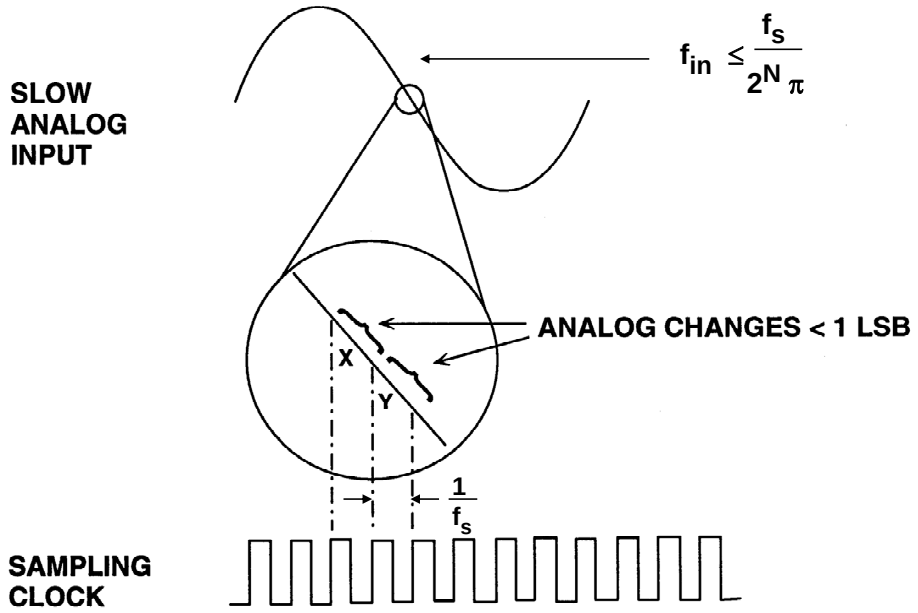

Figure 7: ADC Analog Input Signal for Low Frequency BER Test

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

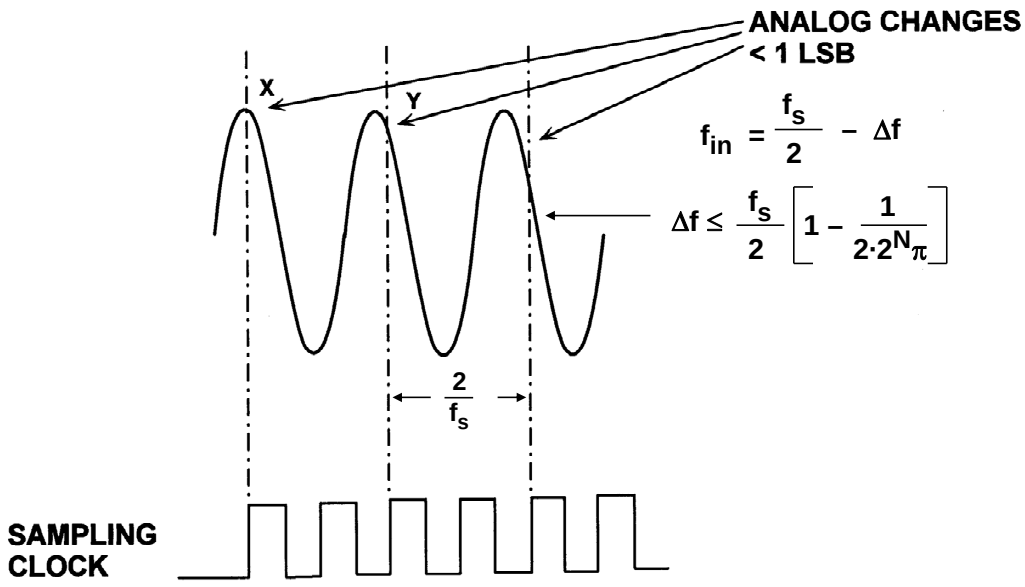

Figure 8: ADC Analog Input for High Frequency BER Test

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.

$$v(t) = \frac{2^N}{2} \sin\left[2\pi\left(\frac{f_s}{2} - \Delta f\right)t\right]. \qquad \text{Eq. 5}$$

The maximum rate of change of this signal is

$$\left.\frac{dv}{dt}\right]_{max} \leq 2^N \pi\left(\frac{f_s}{2} - \Delta f\right). \qquad \text{Eq. 6}$$

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

$$\Delta f \leq \frac{f_s}{2}\left(1 - \frac{1}{2 \cdot 2^N \pi}\right). \qquad \text{Eq. 7}$$

Establishing the BER of a well-behaved ADC is a difficult, time-consuming task—a single converter can sometimes be tested for days without an error. For example, tests on the AD9002

8-bit 150-MSPS flash converter operating at a sampling rate of 75 MSPS yielded a BER of approximately 3.7×10^{-12} (1 error per hour), with an error limit of 4 LSBs. Meaningful tests for long periods of time require special attention to EMI/RFI effects (requiring a shielded screen room), isolated power supplies, isolation from soldering irons with mechanical thermostats, isolation from other bench equipment, etc.

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

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

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

SUMMARY

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

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

REFERENCES

1. James N. Giles, "High Speed Transistor Difference Amplifier," *U.S. Patent 3,843,934*, filed January 31 1973, issued October 22, 1974. *(describes one of the first high-speed ECL comparators, the AM685).*

2. Christopher W. Mangelsdorf, "A 400-MHz Input Flash Converter with Error Correction", *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 1, February 1990, pp. 184-191. *(a discussion of the AD770, an 8-bit 200 MSPS flash ADC. The paper describes the comparator metastable state problem and how to optimize the ADC design to minimize its effects).*

3. Charles E. Woodward, "A Monolithic Voltage-Comparator Array for A/D Converters," *IEEE Journal of Solid State Circuits*, Vol. SC-10, No. 6, December 1975, pp. 392-399. *(an early paper on a 3-bit flash converter optimized to minimize metastable state errors).*

4. J. Peterson, "A Monolithic video A/D Converter," *IEEE Journal of Solid-State Circuits*, Vol. SC-14, No. 6, December 1979, pp. 932-937.

5. Yukio Akazawa et. al., "A 400MSPS 8 Bit Flash A/D Converter," *1987 ISSCC Digest of Technical Papers*, pp. 98-99. *(describes a monolithic flash converter using Gray decoding).*

6. Matsuzawa et al., "An 8b 600MHz Flash A/D Converter with Multi-stage Duplex-gray Coding," *Symposium VLSI Circuits, Digest of Technical Papers*, May 1991, pp. 113-114. *(describes a monolithic flash converter using Gray decoding).*