Jitter in Digital Communication Systems, Part 1

Justin D. Redd, Ph.D.

1 Background

In digital communications, binary encoded information (a sequence of 1's and 0's) is sent from a transmitter to one or more receivers. The transmission medium between the transmitter and receiver (*the channel*) may be copper cable, optical fiber, free space, etc. Major design goals of most digital communication systems include maximizing bandwidth efficiency and minimizing bit error ratio (*BER*).

A basic characteristic of digital communications systems is the need for synchronization between the binary encoded data (the *bit stream)* and the various circuit elements in the transmitter and receiver. Bit synchronization information is generally conveyed by the *bit clock*, which is a square wave signal that has a frequency (in Hz) equal to the data rate (in bits per second).

The bit stream can be encoded into a voltage waveform in a variety of ways. One of the most commonly used binary encoding schemes (and the one used in this paper) is nonreturn-to-zero (NRZ). In NRZ encoding, a binary one is represented by a high voltage (or optical power) level and a binary zero is represented by a low voltage (or optical power) level. The time duration for a single bit (the bit period) is called the unit interval (UI). The UI is the same for each bit in the bit stream and it is equal to the reciprocal of the data rate. For example, one UI for a 622 Mbit/s data rate is $1/(622 \times 10^6$ bit/s) = 1.6 ns per bit. The relationship between an NRZ encoded bit stream and the bit clock is illustrated in Figure 1.

A fundamental problem is how to get the bit synchronization information from the transmitter to the receiver. Direct transmission of the bit clock would require significant additional bandwidth and is vulnerable to channel effects that may result in clock/data misalignment at the receiver. In general, digital communication systems transmit only the bit stream and then regenerate the bit clock at the receiver through use of a clock and data recovery (CDR) circuit. Imperfections in receiver bit clock regeneration result in timing errors in the recovered bit clock. These timing errors are called *jitter*.

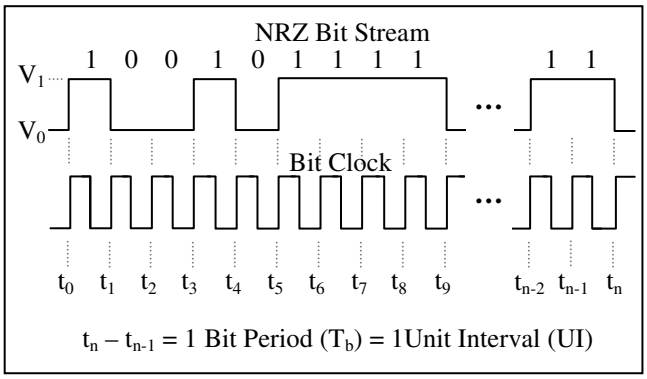

Figure 1. NRZ encoded bit stream

2 Defining Jitter

Two common definitions of jitter are:

"Jitter is defined as the short-term variations of a digital signal's significant instants from their ideal positions in time. Significant instants could be (for example) the optimum sampling instants." (SONET)[1]

and

"The deviation from the ideal timing of an event. The reference event is the differential zero crossing for electrical signals and the nominal receiver threshold power level for optical systems." (Fibre Channel)[2]

Although there are differences in these two (and many other) definitions, it is important to focus on the fundamental similarity, which is that jitter has to do with the time difference between the ideal and actual occurrence of an event. The event in question may change, depending on the application. It could be, for example, the rising or falling edge of a clock, the optimum sampling instant of an NRZ encoded waveform, the differential zero crossing, or something else; but *jitter is simply the time difference between when a pre-defined event should have occurred and when it actually did occur.*

2. 1 Instantaneous Jitter

As noted above, jitter is based on the timing of a specific event. We will call the event *E*, and the time that it actually occurs t_E. Since, for most practical applications, *E* will occur repeatedly, the n^{th} occurrence of *E* can be represented as *E*[n], and the time of the n^{th} occurrence is t_E[n]. Using this notation, the definition of jitter can be written mathematically as

$$j[n] = t_E[n]_{Ideal} - t_E[n]_{Actual} \quad (1)$$

where j[n] is the *instantaneous jitter* associated with the n^{th} occurrence of the event.

As an example, we can consider the case of a clock signal of frequency *f* Hz. Ideally, every period of the clock should be exactly 1/*f* seconds. Because of thermal and other effects, however, there may be a finite amount of random phase noise on the clock signal. If we choose the rising edge of the clock signal as the event of interest then, as soon as we measure the time location of the one rising edge (we'll call this measurement t_E[0]), we can predict the exact (or ideal) time that each succeeding rising edge should occur. In this case

$$t_E[n]_{Ideal} = t_E[0] + \frac{n}{f}. \quad (2)$$

When we take measurements on the real clock signal we find the actual time for each rising edge ($t_E[n]_{Actual}$). We can then compute the instantaneous jitter j[n] at each rising edge using equation (1).

2. 2 Units of Measure

Jitter is, by definition, a measure of time. Units of picoseconds (ps) are commonly used. In many applications it is convenient to normalize jitter to the unit interval (UI). Normalized jitter is computed by dividing jitter in units of time by the time for one UI as shown in Equation 3.

$$Jitter(\text{UI}) = \frac{Jitter(\text{sec})}{1\,\text{UI(sec)}} \quad (3)$$

2. 3 Average and Peak-to-Peak Jitter

In the same sense that average power is usually more interesting than instantaneous power, it is generally more useful to talk about jitter in terms of average and peak-to-peak values than instantaneous values.

The root-mean-square (rms) method is commonly used to compute average jitter and units are given as ps_{rms} or UI_{rms}. The following equations can be used to convert an uncorrelated sequence of instantaneous jitter measurements to their estimated rms average values:

$$\mu_j = \frac{1}{N}\sum_{n=1}^{N} j[n] \quad (4)$$

$$\sigma_j = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(j[n]-\mu_j\right)^2} \quad (5)$$

In these equations, μ_j is the mean value of the jitter, *N* is the total number of instantaneous jitter values measured, and σ_j is the rms average (also called the standard deviation or one-sigma) value.

Peak-to-peak jitter values for a set of *N* instantaneous jitter measurements are computed by subtracting the minimum instantaneous jitter measurement from the maximum instantaneous jitter measurement, as in equation 6.

$$Jitter_{p-p} = \max\{j[n]\} - \min\{j[n]\} \quad (6)$$

Units for peak-to-peak jitter measurements are given as $ps_{p\text{-}p}$ or $UI_{p\text{-}p}$.

3 Types of Jitter

Jitter can be divided into two fundamental types, called random jitter and deterministic jitter. Random jitter (RJ) is unpredictable and has a Gaussian probability density function. Deterministic jitter (DJ) is predictable (assuming prior knowledge of the bit stream characteristics) and has definite amplitude limits (i.e., it is *bounded*). RJ is caused by thermal (or other random) noise effects in the system that are induced into the phase of the clock and data signals. DJ is caused by process or component interactions in the system, such as the effect of limited bandwidth on specific patterns of 1's and 0's in the bit stream. Figures 2 and 3 are example eye diagrams showing excessive RJ (Figure 2) and excessive DJ (Figure 3).

3. 1 Deterministic Jitter

Deterministic jitter can be divided into a number of subtypes, which can generally be grouped into four categories:

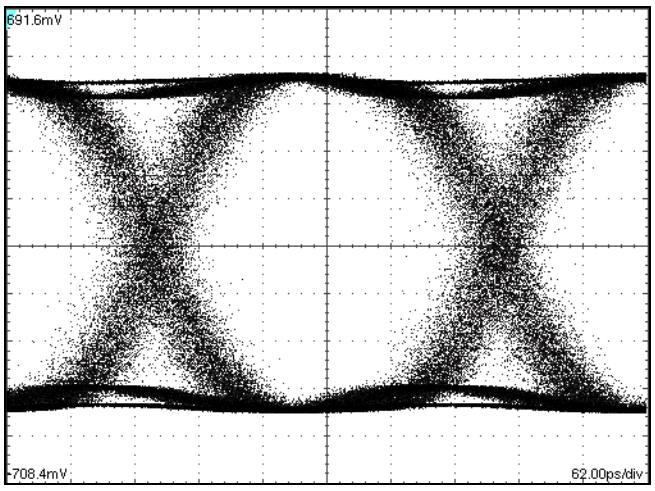

Figure 2. Eye diagram showing RJ

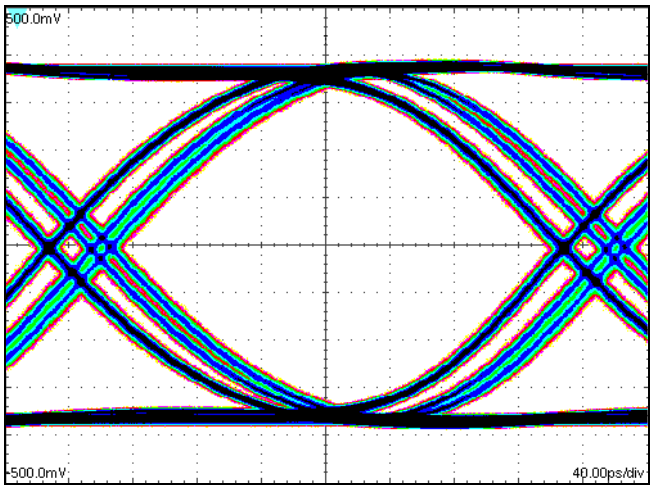

Figure 3. Eye diagram showing DJ

3.1.1 Duty Cycle Distortion (DCD) and Pulse-Width Distortion (PWD) are different names for the same thing. (We will refer to it as PWD in this document.) PWD can be defined as the difference between the pulse width of a high output (representing a "1") and the pulse width of a low output (representing a "0"). PWD causes a distortion in the eye diagram where the eye crossings are offset up or down from the vertical midpoint of the eye. Figure 4 shows an eye diagram that is distorted by PWD.

PWD can be quantified by driving the system with a clock like pattern (such as 1 0 1 0 …), measuring the width of the high and low pulses, and then using the following equation:

$$PWD = [(longer\ pulse) - (shorter\ pulse)] / 2 \quad (7)$$

The most common causes of PWD are voltage offsets between the differential inputs and differences between the rise and fall times in the system.

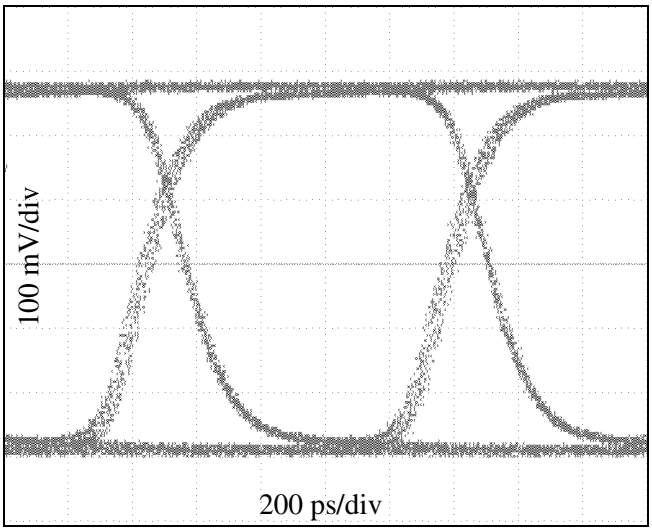

Figure 4. Eye diagram showing PWD

3.1.2 Data Dependent Jitter (DDJ) and Intersymbol Interference (ISI) are two different names for the same type of jitter, viewed from the perspective of time and frequency, respectively. Another common name for DDJ/ISI is pattern dependent jitter (PDJ). This subtype of DJ is illustrated in Figure 5.

DDJ refers to jitter from a time-domain perspective. It can be defined as the jitter that appears in a system when the pattern of bits changes from a clock-like square wave (e.g., 11 00 11 00 …) to a non clock-like pattern. In other words, the jitter is different for every unique bit pattern.

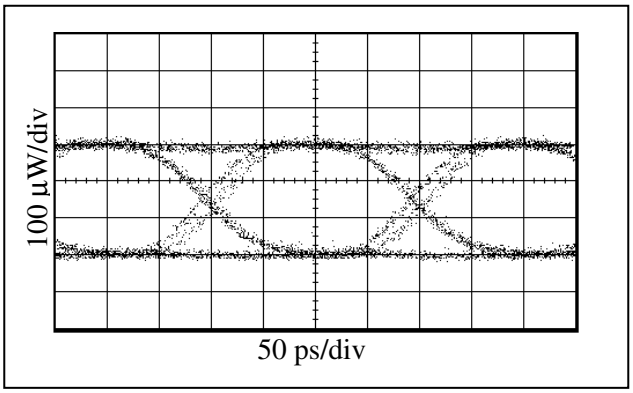

Figure 5. Eye diagram showing DDJ/ISI/PDJ

ISI refers to jitter from a frequency-domain perspective. It can be defined as pulse-spreading that is caused by bandwidth limits in the system. When the bandwidth of the system is much greater than the bandwidth required by the pulse, then pulse-spreading (ISI) is very small. When the bandwidth of the system is approximately the same, or less than the bandwidth required by the pulse, then pulses are spread into adjacent bit times, causing distortion in the adjacent pulses. Quickly changing bit patterns (e.g., 1010…) require more power at high-frequencies, whereas slowly changing bit patterns (e.g., 1111111100000000…) require more power at low frequencies. If the system has high- or low-frequency cutoffs within the bandwidth required by the bit pattern, the resulting distortion is called ISI jitter. In other words, ISI jitter is different for every unique bit pattern (the same as DDJ).

DDJ/ISI can be reduced by either reducing the bandwidth requirement of the pulse (pulse-shaping) or increasing the bandwidth of the system. It is important to note that low-frequency and high-frequency cutoffs can cause DDJ/ISI. For example, an AC-coupling capacitor that is too small for the data rate and bit pattern can block enough of the low-frequency components of the pattern to cause DDJ/ISI (see Figure 6). In the same sense, a system with a high-frequency roll-off that is too low for the data rate will block high-frequency components of the pattern, causing DDJ/ISI, as shown in Figure 7. Another possible cause of DDJ/ISI is peaking and/or ringing.

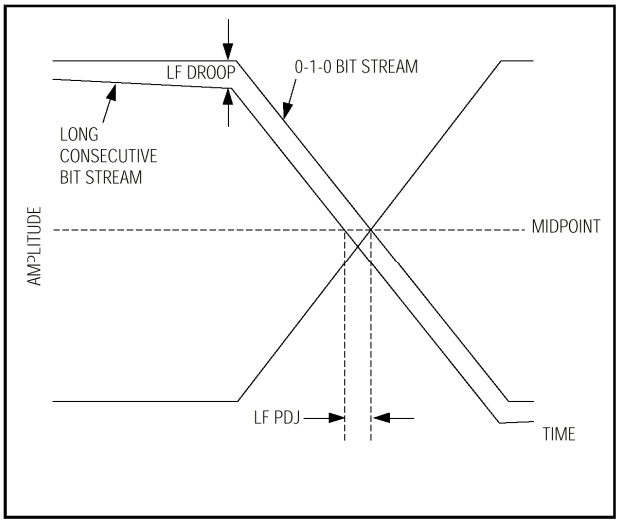

Figure 6. Pattern-dependent jitter due to low-frequency cutoff

3.1.3 Sinusoidal Jitter (SJ) displaces the timing of the rising and falling edges of the bit stream by varying amounts. The amplitude of the timing displacement from edge to edge follows a sinusoidal pattern. Equation 7 is a mathematical description of SJ.

$$j[n] = A\sin[2\pi f\frac{n}{r} + \phi] \qquad (7)$$

In equation 7, j[n] represents the instantaneous jitter at edge n, A is the jitter amplitude, f is the jitter frequency, r is the data rate, and ϕ represents a random phase offset. Sinusoidal jitter is seldom encountered in real systems, but it is widely used in jitter testing.

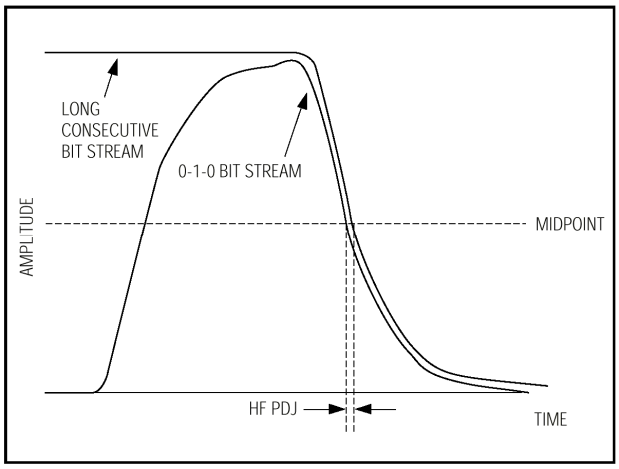

Figure 7. Pattern-dependent jitter due to high-frequency roll-off

3.1.4 Uncorrelated and Bounded Jitter is essentially any deterministic jitter that does not fit into one of the other sub-categories of deterministic jitter. It can be defined as jitter that is bounded (i.e., deterministic), but not correlated with the bit pattern. Common causes of this type of jitter are power supply noise and cross talk.

3.2 More on RMS Versus Peak-to-Peak Jitter

Random jitter has a Gaussian probability density function (PDF) which is, by definition, unbounded. The Gaussian PDF can be completely defined by its mean and rms average. Since the tails of the Gaussian PDF extend to infinity, there is a small, but finite, possibility that the instantaneous jitter at any edge could have an infinite value. For this reason, random

jitter is normally specified and measured in terms of its rms average (see equation 5). Peak-to-peak measurements of random jitter are inherently ambiguous unless some additional boundary conditions are imposed (see *Total Jitter*, below).

Deterministic jitter has, by definition, upper and lower bounds. It is therefore straightforward to measure DJ in terms of peak-to-peak values. RMS average values for DJ are ambiguous, since PDFs for DJ can take any bounded form (i.e., There is no correlation between the DJ rms average and any particular PDF.) Peak-to-peak jitter for separate system components can be directly added together to compute the worst-case overall peak-to-peak jitter (see *Total Jitter*, below).

RMS jitter measurements for separate system components cannot be directly added, but may be combined by taking the square-root of the sum of the squares, as shown in equation 8 (where σ represents the rms average).

$$\sigma_{TOTAL} = \sqrt{\sigma_1 + \sigma_2 + \sigma_3 + \ldots} \qquad (8)$$

3.3 Total Jitter (TJ)

Total jitter (TJ) is the combination of all RJ and DJ components. In order to compute total jitter, it is necessary to convert all rms average jitter (RJ) numbers to peak-to-peak values. All of the peak-to-peak jitter sub-components can then be added together to get the total jitter as a peak-to-peak quantity.

Converting rms jitter to peak-to-peak jitter can be done by defining arbitrary limits on the Gaussian PDF that is characteristic of RJ. One of the most useful ways accomplish this is based on the bit error ratio (BER) required by the system. To convert from rms jitter to peak-to-peak jitter using this method, simply multiply the rms jitter by the value α corresponding to the appropriate BER in Table 1 [2,3]. It is important to emphasize that this method is only applicable to Gaussian distributed random jitter (i.e., it is invalid to use this method to try and convert peak-to-peak DJ to rms jitter). For example, if the system BER requirement is 10^{-12} and the RJ is $4ps_{rms}$, the corresponding peak-to-peak jitter is $4ps_{rms}$ x 14.1 = $56.4ps_{peak\text{-}to\text{-}peak}$. (For more details on this method, see Maxim application note HFAN 4.0.2, Converting Between RMS and Peak-to-Peak Jitter at a Specified BER.)

Peak-to-peak jitter quantities for separate system components can be directly added together, but this results in a worst-case TJ number that usually over estimates measured TJ. This is because the peak values of DJ for any particular system sub-component occur at the specific points in the bit pattern that cause the most stress to the bandwidth limits of that sub-component. Also, the peak values of RJ occur at random but infrequent times. Since each system component has unique bandwidth characteristics, the peak DJ may occur at different points in the bit pattern for each of them, and the probability that the peak DJ occurs simultaneous with the peak RJ is low. Summation of the individual peak-to-peak jitter numbers for each sub-component in the system gives a result based on the assumption that all of the peak jitter values occur simultaneously (an unlikely event). Thus *the measured TJ will always be less than or equal to the sum of the sub-component peak-to-peak jitter measurements*. Also, if the total system RJ and DJ are measured separately, *the measured TJ will always be less than or equal to the sum of the measured peak-to-peak RJ and the measured DJ*.

Table 1. Scaling Factors (α) Corresponding to System Bit Error Ratio (BER)

BER	α	BER	α
10^{-9}	11.996	10^{-13}	14.698
10^{-10}	12.723	10^{-14}	15.301
10^{-11}	13.412	10^{-15}	15.883
10^{-12}	14.069	10^{-16}	16.444

Because of the inequality between TJ and the sum of RJ and DJ, it is difficult to specify all three quantities (TJ, RJ, and DJ) simultaneously. To overcome this difficulty, it is common to specify the required TJ quantity along with either the RJ or DJ (but not both). The unspecified quantity of RJ or DJ is indirectly specified as whatever amount is required to achieve the TJ number. For example: "$0.22UI_{rms}$ RJ plus enough DJ to achieve a TJ of 0.6UI."

4 Specifying System Jitter Performance

Jitter performance requirements for system components are specified in terms of jitter generation, jitter transfer, and jitter tolerance.

4. 1 Jitter Generation

Jitter generation can be defined as the quantity of jitter added to a signal. This parameter is primarily used with transmitting components, such as laser drivers, serializers, line drivers, regenerator clock/data recovery circuits (CDRs), and limiting amplifiers.

4. 2 Jitter Transfer

Jitter transfer is a measure of amount of jitter in the input signal that is transferred to the output signal. It is usually expressed as the ratio of output jitter to input jitter at a specific jitter frequency. Jitter transfer is used to specify the performance of regeneration components, including clock and data recovery (CDR) circuits, data retimers, etc.

4. 3 Jitter Tolerance

The ability of a receiving device to correctly detect incoming data, despite jitter, is called jitter tolerance. Jitter tolerance can be defined as the amplitude of the incoming jitter that causes the BER of the recovered data to exceed a specified limit. It is measured at a discrete jitter frequency. Jitter tolerance measurements at discrete frequencies are usually combined in order to plot jitter tolerance over a range of jitter frequencies. Jitter tolerance is used to specify the performance of receiving components, such as clock and data recovery (CDR) circuits and deserializers.

---

[1] Bell Communications Research, Inc (Bellcore), *Synchrouous Optical Network (SONET) Transport Systems: Common Generic Critteria, TR-253-CORE,* Issue 1, December 1994

[2] National Committee for Information Technology Standardization (NCITS), *T11.2/Project 1230, Rev 8 (working draft)*, http://www.t11.org, 15 March 1999.

[3] B. Sklar, *Digital Communications: Fundamentals and Applications,* Englewood Cliffs, New Jersey: Prentice Hall, pp. 733-743, 1988,