Verification of Bit-Error Rate in Bang-Bang Clock and Data Recovery Circuits

Ken Kundert

Version 1c, 4 May 2010

High speed serial data links are expected to transmit data at very high rates with very high fidelity. Today speeds approaching 10 Gb/s are becoming common with 40 Gb/s on the horizon. Typically, a maximum bit-error rate (BER) of 10^{-12}-10^{-15} is required. Verifying such a small BER with direct simulation is quite impractical. Instead, a procedure is presented that separates the deterministic and random components of the jitter, the primary cause of errors, and verifies them individually. By separating these two components, the BER can be verified in a time that is independent of the value that must be achieved. In this way it is practical to verify the extremely small BERs required of today's designs. This methodology, though generic in nature, will be demonstrated by applying it to the key component in a high-speed link, the clock and data recovery circuit (CDR). And in particular, it will be applied to a type of CDR that is especially difficult to characterize, a bang-bang clock and data recovery circuit (BB-CDR).

Search Terms

Bang-bang clock and data recovery, bang-bang CDR, bit-error rate, BER simulation, jitter generation, jitter tolerance, jitter transfer, SpectreRF.

This paper was written in February 2007 and was last updated on June 11, 2022. You can find the most recent version at www.designers-guide.org. Contact the author via e-mail at ken@designers-guide.com.

1 Introduction

The quest for higher performance in computing systems is causing explosive grown in the use of high-speed serial communication interfaces. They are being used to address the communication bottlenecks both inside and between high performance computer systems [6]. As such, there is tremendous interest in finding ways to efficiently design and verify them. The clock and data recovery (CDR) circuit is the portion of the link that is both the most difficult to design and the most difficult to verify [13]. And, the most troublesome of the performance limiting aspects of the CDR is jitter. Recently progress was made by Lee et al, who reported on a procedure for estimating the jitter that is useful during the design phase [8]. However, this procedure cannot predict the bit-error rate (BER) of the CDR, and so is not capable of verifying that the achieved performance meets the high-level objectives. This paper addresses the verification problem by presenting a procedure that can be used to verify that a design meets a particular BER. This procedure is general and can be applied anywhere along the link.

2 The Challenge

Typically high-speed serial interfaces employ little to no error detection or correction. This is particularly true for inter-chip interfaces, where power constraints limit the amount of circuitry that can operate at the very high speeds of the channel. Instead, the link itself is expected to faithfully pass the data with very high reliability. Typically error rates must be below 1 part per trillion (10^{-12}) and sometimes as low as 1 part per quadrillion (10^{-15}). These very small allowed bit-error rates make verification by direct simulation impractical. Consider that to gather a statistically significant set of errors, typically a hundred are needed. That implies that verifying a 1 PPT BER requires simulating 10^{14} cycles. When simulating with very abstract compiled models, it is conceivable that one might achieve a simulation rate of 10 Mcycles/sec. Even at this rate, approximately 2000 hours would still be needed to accumulate the required number of errors. Thus, even for very simple models, this approach is impractical. Another approach is needed.

3 The Nature of Errors

Errors in serial communication systems can come from many sources and it is useful to categorize these sources as being either deterministic or random in nature [12]. Deterministic error sources are those that produce errors that can be perfectly predicted if you know enough about your system. Examples include inter-symbol interference, signal attenuation, interferers, and even the basic operation of your circuit. Errors from these sources must be avoided completely, because if they occur at all then they occur with high frequency. For the purposes of this discussion, systems that suffer errors from deterministic sources are considered to be non-functional. So the task of verifying that a system is functional is the task of verifying that the system operates without error in the presence of deterministic impairments.

Deterministic error sources are bounded and so are not the source of the infrequent errors that concern us when trying to verify a very small BER of 1 PPT or less. The rea-

son why is that if a deterministic error occurs, it occurs much more frequently that once every trillion bits. Consider inter-symbol interference (ISI), an impairment that occurs because time constants in the circuit cause the memory of previously received bits to affect the perceived value of the bit currently being received. Consider the case where your circuit receives a long pattern of bits that excites a resonance in the channel that in turn causes an error. For this to occur at a rate of 1 PPT, it must occur for one particular pattern, and only one, out of a possible one trillion patterns, or for one particular pattern in a sequence of 40 consecutive bits ($2^{40} \approx 10^{12}$). The fact is that if your system produces an error for that particular pattern, then undoubtedly it produces errors for many other closely related 40 bit patterns; probably millions or billions of others because ISI is simply nowhere near that discriminating. And so your BER will be well above 1 PPT, and by the definition given above, your circuit is non-functional.

Infrequent errors are instead due to random error sources. Random errors are due to the noise sources that are intrinsic to the devices that make up the circuit. They include the thermal, shot, and flicker noise inherently produced by transistors, resistors, etc. These error sources generally have Gaussian or Gaussian-like distributions. These distributions are unbounded, and so are capable of producing very infrequent large errors.

To understand error in serial communication systems, consider the eye diagram shown in Figure 1 [12]. The eye diagram shows the input waveform relative to t_d, the decision point, which is shown in the center of the diagram[1]. At t_d, the waveform must either be clearly high or clearly low to ensure a correct decision. The hexagon in the middle of the diagram represents an exclusion zone. The goal of the designer is to keep the waveforms from passing through any part of this zone as a way of providing a safety margin so as to assure a correct decision. In doing so the designer is said to be keeping the 'eye' open. Any transitions in the input signal (the point where the waveform crosses v_{thresh} in either direction) are expected to occur near t_c, the desired crossing point, which is one half unit interval before t_d, where a unit interval is the expected duration of time allotted for the transmission of a single symbol. In the case of a 1 Gb/s raw bit rate, the unit interval is 1 ns.

Another way of seeing this is to use a bathtub curve, as shown in Figure 2 [3,12]. When building a bathtub curve the decision point t_d is allowed to be anywhere within the unit interval and the bit error rate is plotted as a function of its placement. The bathtub curve consists of two regions. When t_d is close the either end, the BER is dominated by deterministic jitter (DJ), but as t_d moves towards the center of the UI its contribution drops precipitously and random jitter (RJ) dominates. Because of the Gaussian nature of RJ, its contribution drops much more slowly.

As seen from Figure 1, the uncertainty in the value of the waveform is primarily due to a phase (horizontal) variation and not as a result of an amplitude (vertical) variation. Thus the majority of the risk of a bit error stems from jitter in the signal, and this risk only increases as data rates continue to increase. The jitter causes the apparent horizontal variation in the waveform. One measures the jitter by plotting the histogram of threshold crossings for the waveform. The histogram is shown in the lower left corner of the eye diagram. An error would occur if the histogram spread to the point where it overlapped the decision point, t_d, as shown in Figure 3.

1. It should be understood that both t_c and t_d appear to vary on a cycle-by-cycle basis to an outside observer, but here we fix t_d and show all of the variation in t_c.

FIGURE 1 *Eye diagram (top) and jitter histogram (bottom) of a typical serial communication system.*

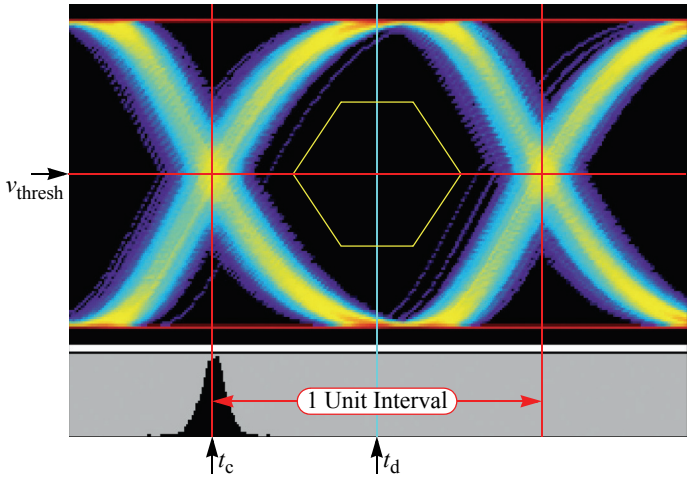


FIGURE 2 *A bathtub plot, which shows BER versus t_d over one UI, showing the regions dominated by RJ and DJ.*

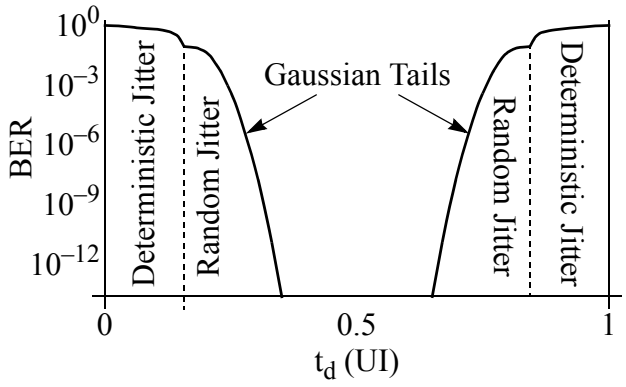
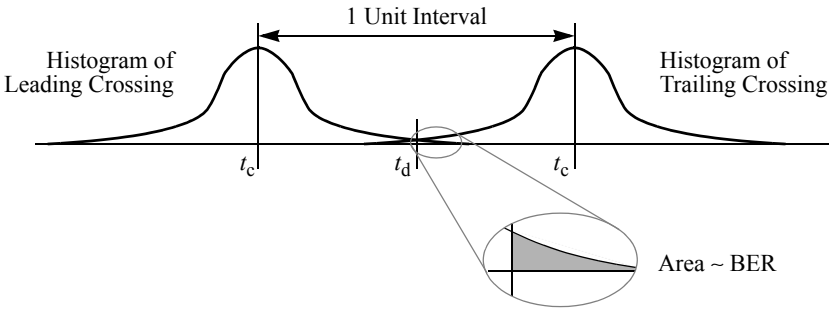


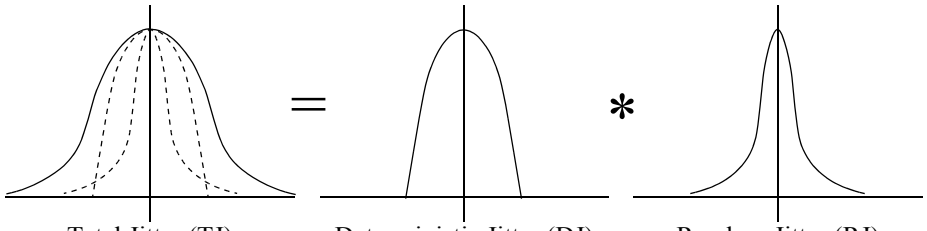
FIGURE 3 *Error occurs when the half width of histogram exceeds ½ unit interval.*



Conceptually decomposing the jitter into deterministic jitter and random jitter components allows us consider the effect of each separately. In Figure 4 the total jitter of Figure 1 is decomposed into its deterministic and random components. It is important to

recognize that there is generally more jitter in the deterministic component (its standard deviation is greater) than in the random jitter component[2], but peak jitter from the random component is generally greater than that of the deterministic component because of the long tails of the Gaussian distribution. As such, the random component must be considered when predicting the BER.

FIGURE 4 *The jitter histogram from the eye diagram decomposed into deterministic and random components. The combination of the probability density functions for the deterministic and random components equals the density function for the total jitter.*



The deterministic jitter is always bounded, but the random jitter is not because it stems from noise sources with Gaussian or Gaussian-like distributions. Thus when characterizing random jitter, the best one can do is to identify how often the jitter exceeds a particular threshold. This is shown in Table 1.

TABLE 1 *The ratio (ρ) of the peak deviation of a Gaussian process to its standard deviation where the peak deviation is defined as the value that is not be exceeded more often than a given rate (the BER).*

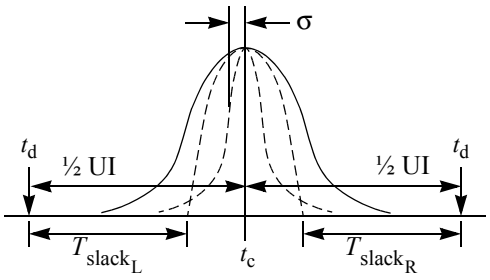
BER	ρ	BER	ρ
10^{-4}	3.9	10^{-10}	6.4
10^{-5}	4.4	10^{-11}	6.8
10^{-6}	4.9	10^{-12}	7.1
10^{-7}	5.3	10^{-13}	7.4
10^{-8}	5.7	10^{-14}	7.7
10^{-9}	6.1	10^{-15}	8.0

As can be seen by inspecting Figure 1, an error occurs if the total jitter in a cycle exceeds ½ unit interval (UI). Thus, if the random jitter has a Gaussian distribution, the BER can be found from the peak deterministic jitter and the standard deviation of the random jitter. If σ is the standard deviation of the random jitter and T_{slack} is smaller of the two values defined in Figure 5, then the BER is found by computing T_{slack}/σ and using Table 1 to convert this value into a BER. For example, if the ratio of T_{slack} to σ is 6.1 then the BER will be 10^{-9} or 1 PPB.

In the above it was implicitly assumed that t_c can be precisely placed at ½ UI. If this is not the case, then T_{slack} should be reduced to account for any uncertainty in its placement, perhaps due to systematic timing errors. So T_{slack} should be equal to $T_{slack} = ½\ UI - DJ_p - \Delta t_c$. An example of such would be if t_c could be adjusted to compensate for asymmetries in the eye diagram, but could only be adjusted over a finite number of

2. This assertion, while believed to be true, has not been verified.

FIGURE 5 *T_{slack} is the difference in time between the decision point t_d, assumed to be ½ UI from the expected crossing point t_c, and the peak deviation of the deterministic jitter. If the density function for the deterministic jitter is asymmetric, then there will be two values for T_{slack}. The smallest is used when determining the BER.*



equally sized steps, as if t_c were being produced by selecting a particular phase of a many-stage ring oscillator. In this case, if there are N stages, then t_c can only be resolved to within $\Delta t_c = UI/2N$.

To verify that our system meets a particular BER, we can simply calculate the maximum value of σ that can be tolerated. For example, if the BER must be below 10^{-12}, then

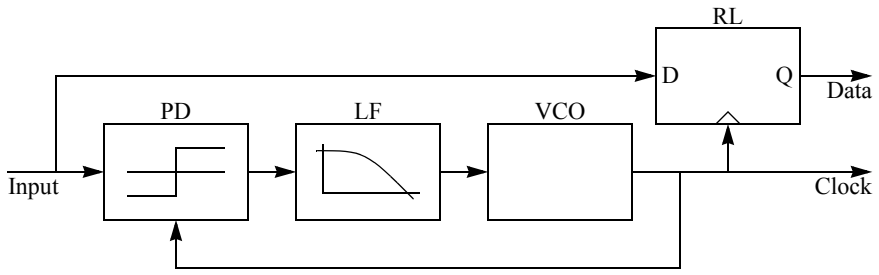
$$\sigma < \frac{T_{slack}}{\rho} \text{ where } \rho = 7.1 \text{ for BER} = 10^{-12}. \quad (1)$$

The approach used will be to first predict deterministic jitter using direct simulation. This is practical because we only must verify functionality rather than an extremely small bit-error rate. T_{slack} is computed from these results. Then, using this information the random jitter and σ are predicted analytically. The BER is then verified using Table 1 and (1). This procedure will be demonstrated by applying it to a bang-bang clock and data recovery (CDR) circuit in Section 8.

4 Bang-Bang Clock and Data Recovery

Bang-bang CDRs are generally the largest source of errors in a high-speed serial link, and are interesting because they exhibit large amounts of both deterministic and random jitter. The block diagram of a bang-bang CDR is shown in Figure 6. It consists of a binary or Alexander phase detector (PD) [4], a loop filter (LF), a voltage controlled oscillator (VCO) and a retiming latch (RL). It is unique in that the phase detector only has two output levels. Thus, this type of phase detector can communicate only two messages to the VCO: either you are late, speed up; or you are early, slow down. There is no in between, so the VCO in a bang-bang CDR will on average produce the correct frequency, but it will constantly be accelerating and decelerating, which produces a substantial amount of deterministic jitter. More recently phase detectors used in bang-bang CDRs have added a third output state, a high impedance state. This state is used whenever no edge is received. In this case the phase detector cannot determine whether the edge on the generated clock is early or late, and so indicates to the VCO that it should neither speed up nor slow down, but rather it should just coast until the next edge is received.

FIGURE 6 *Block diagram of a bang-bang clock-and-data recovery circuit.*



5 Modeling the CDR

The basic idea of this paper is to use modeling and simulation to predict the bit-error rate of a CDR. But what type of model should be used? Two possible choices are available: voltage-domain models or phase-domain models. A voltage-domain model is formulated in terms of instantaneous quantities, such as voltages and currents, whereas phase-domain models are formulated in terms of the phase of the signals. The phase of a signal cannot be determined by looking at the value of that signal at a single point in time. Rather, it can only be determined using a synchronous detection process, a process that involves averaging. Thus, phase-domain models are innately time-average models. Voltage domain models are used when the detailed system behavior (the behavior at every point in time) is needed whereas phase-domain models are used when the details of each cycle are not needed and so can be traded off to achieve much faster simulations.

To predict BER both types of models are needed. A full transistor level voltage-domain model is used to predict the response to stimuli that cause substantial changes to behavior on a cycle-by-cycle basis. This includes

1. The jitter caused by the highly-nonlinear nature of the phase detector
2. Jitter due to phase-detector metastability
3. Data dependent jitter
4. Inter-symbol interference
5. Duty-cycle dependent jitter

Conversely, the phase-domain model is used to predict the response to stimuli that are either small or slow, and so have only a small effect on the behavior from cycle to cycle. They include

1. Random jitter
2. Sinusoidal jitter on the data stream.

The abruptly discontinuous nature of the phase detector in a bang-bang CDR would normally preclude the development of a phase-domain model as the behavior of the PD would potentially change dramatically with small changes in the stimulus. However, in this case this issue is overcome by using the fact that the jitter that is always present acts to smooth the effective behavior of the PD. To account for this effect, the average behavior of the voltage-domain model is observed over a very large number of cycles to

build the phase-domain model. In other words, the phase-domain model is extracted from the composite results of the simulation of the voltage-domain model over these many cycles.

The overall procedure is as follows. A transistor-level model of the CDR is simulated while applying all large sources of jitter (enumerated in the first list given above), which all happen to be deterministic in nature. Thus, a representative input stream is applied to the circuit and a transient analysis is performed for long enough to produce an eye diagram and accurately capture the jitter histogram of Figure 1. The input data stream would normally consist of a pseudorandom sequence of valid symbols that have been corrupted with inter-symbol interference and the like as appropriate. Typically the input sequence is encoded before being applied to the CDR, often using an *N*b/*M*b code where *N* and *M* are integers. A typical example is 8b/10b where every 8 bits of input data are converted to a 10 bit symbol. The symbols are chosen to provide both limited run-length and DC balance. This means that there are not too many "1"s or "0"s in a row and that there are just as many "1"s as "0"s in a symbol. Both of these make the process of clock recovery much easier.

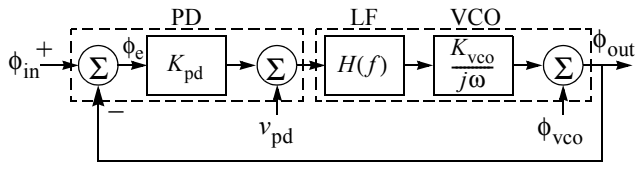
Generally, simulating thousands of cycles at a minimum is required in order to accurately resolve the deterministic jitter histogram; however this is within the capabilities of traditional circuit simulators even when the circuit is simulated completely at the transistor level with parasitics. Enough cycles should be simulated to resolve the actual shape of the histogram, not just its width. This shape is used later to build an effective phase-domain model of the phase detector.

The CDR is said to function properly in the presence of deterministic jitter if the histogram is completely contained within ±½ unit interval.

6 Phase-Domain Model of a Bang-Bang CDR

A phase-domain model of a CDR is a model that is formulated in terms of the phase of signals rather than their voltages or currents. Formulating models in this way suppresses the details of each transition and so considerably simplifies the model. In this case, it also allows us to create models that are more linear and allows for the use of small-signal analysis to better understand the behavior and performance of our circuit. A phase-domain model for the clock-recovery portion of a CDR is shown in Figure 7. To use this model, it is first necessary to develop phase-domain models for the phase detector and the VCO, and then to find the output voltage noise of the phase detector (v_{pd}) and the output phase noise for loop filter/VCO (ϕ_{vco}). For convenience, all phase variables (ϕ) will have units of UI or unit intervals.

FIGURE 7 *Phase-domain model of the clock recovery portion of a bang-bang CDR.*



6.1 Phase-Domain Model of the VCO

The voltage-controlled oscillator, or VCO, converts its input voltage to an output frequency, and the relationship between input voltage (v_c) and output frequency (f_{out}) can be represented as

$$f_{out} = F(v_c) \tag{2}$$

The mapping from voltage to frequency is designed to be linear, so a first-order model is often sufficient,

$$f_{out} = K_{vco} v_c, \tag{3}$$

where K_{vco} has units of Hz/V. It is the output phase that is needed in a phase-domain model and phase is the integral of frequency,

$$\phi_{out}(t) = \int K_{vco} v_c(t) dt \tag{4}$$

or in the frequency domain,

$$\phi_{out}(\omega) = \frac{K_{vco}}{j\omega} v_c(\omega) . \tag{5}$$

Even if the VCO transfer characteristics are nonlinear, one can fix up the model by making K_{vco} a function of v_c.

The phase noise of the VCO can be extracted as $\mathcal{L}$ using SpectreRF and then converted to S_ϕ [7], the power spectral density of ϕ, using

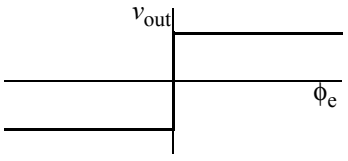
$$S_\phi(f) = \frac{1}{\pi} \mathcal{L}(\Delta f) , \tag{6}$$

where S_ϕ has units of UI/Hz^2.

6.2 Phase-Domain Model of the Phase Detector and Loop Filter

The simplest phase-domain model for a bang-bang phase detector is the one shown in Figure 8. However, this model is missing two important effects: jitter and metastability.

FIGURE 8 *The ideal transfer function of an ideal bang-bang phase detector.*



On any particular cycle the ideal phase detector acts in an abruptly nonlinear manner as shown in Figure 8. However, the jitter present during normal operation acts to shift the transfer curve on each cycle. With a phase-domain model the details of each cycle are not available, and so the transfer curve of Figure 8 cannot be used directly. Instead, a model that maps the input phase to the expected value of the output of the phase detector is used. This map is referred to as the *expected transfer function* of the PD. It is the ensemble average over jitter and can be found by convolving the ideal transfer charac-

teristics with the probability density function (PDF) of the jitter, as shown in Figure 9 [8]. To see this, allow j to be a random variable that represents the jitter on a particular transition and let $\phi = j/T$. Further, allow F to represent the transfer curve of the phase detector. Then

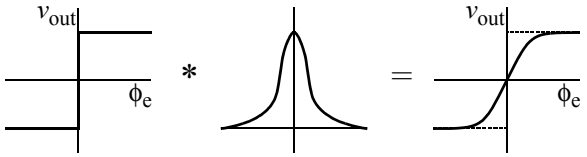
$$v_{out}(\phi_e) = F(\phi_e - \phi)\,. \tag{7}$$

In this case, v_{out} is also a random variable whose expected value is

$$E(v_{out}(\phi_e)) = \int_{-\infty}^{\infty} F(\phi_e - \phi)p(\phi)d\phi \tag{8}$$

where $p(\phi)$ is the probability density function of ϕ. This shows that the expected value of the output of the phase detector is the convolution of static phase transfer curve (F) and the probability density function of the jitter (p).

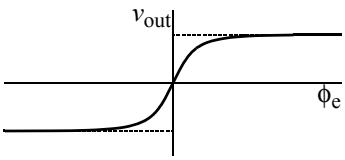
FIGURE 9 *The expected transfer function of an ideal bang-bang phase detector in the presence of jitter.*



For simplicity, only the deterministic jitter will be considered when building the phase-domain model for the phase detector. In this case, the probability density function of the jitter is given by the histogram shown in Figure 1 and found using the procedure described in Section 5. Ignoring the random jitter in this situation is a potential source of error. It is convenient because the random jitter is not yet known and will not be known until we complete the phase domain model. It is usually justified because the average behavior of the phase detector is determined largely by the relatively large excursions of deterministic jitter, which occur frequently. Though random jitter can produce excursions larger than the deterministic jitter, they occur very infrequently and so have little effect on the average behavior of the phase detector.

The other effect to consider is metastability. When a phase detector experiences metastability its output does not reach its full value when the phase difference between the two input signals is small [8]. The loop filter is sensitive to the average value of the phase detector output signal, and so the metastability can be characterized with the average output voltage over a cycle as a function of the input phase difference, as shown in Figure 10.

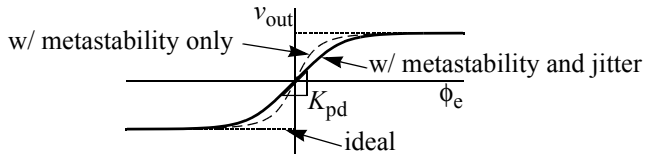
FIGURE 10 *The expected transfer function of a bang-bang phase detector that exhibits metastability.*



The metastability of the phase detector is characterized by applying a square wave to each input of the phase detector and then slowly sweeping the phase between the two and observing the output of the filter that follows the phase detector.

The composite of these two effects (jitter and metastability) is determined by convolving the jitter PDF with the phase transfer curve that includes metastability rather than the ideal curve of Figure 8. A simple linear model of the phase detector is created by setting the gain of the block, K_{pd}, equal to the slope where $\phi_e = 0$. Otherwise, the full nonlinear transfer curve can be used for the model. The full nonlinear model accurately predicts the slewing behavior that results from large rapid changes when simulated using transient analysis.

FIGURE 11 *The phase transfer characteristics of a bang-bang phase detector that exhibits metastability.*



The noise can be found by driving the PD with representative periodic input signals that cause the output to switch. For bang-bang phase detector the frequency of the input signals need to be at half the data rate and the edges of one of the two signals would alternate between being early and late relative to the edges of the other. Then the PNoise analysis of SpectreRF is used to determine the noise of the combination.

6.3 Using the Phase Domain Model to Find the Random Jitter

To determine the BER of the CDR, we must first know both the histogram of the deterministic jitter and the variance of the various sources of random jitter. The histogram of the deterministic jitter was found in Section 5 by simulating the circuit without noise and recording the times of the threshold crossings. The random jitter can be found by applying noise to the phase-domain model shown in Figure 7. If we assume that all of the blocks shown in this figure either exhibit linear behavior or have been linearized then the phase noise at the input of the phase detector is

$$\phi_e(f) = \frac{\phi_{in}(f) - \phi_{vco}(f) + K_{vco}\phi_{pd}(f)/(2\pi\hat{j}f)}{1 + K_{pd}H(f)K_{vco}/(2\pi\hat{j}f)}. \tag{9}$$

The standard deviation of ϕ_e is computed by first computing its power spectral density

$$S_{\phi_e}(f) = \frac{S_{\phi_{in}}(f) + S_{\phi_{vco}}(f) + K_{vco}^2 S_{v_{pd}}(f)/(2\pi f)^2}{1 + T(f)^2} \tag{10}$$

where

$$T(f) = \frac{K_{pd}H(f)K_{vco}}{2\pi\hat{j}f} \tag{11}$$

is the loop gain. Then integrate over frequency to compute the total noise power in terms of phase

$$\mathrm{var}(\phi_e) = \int_0^\infty \frac{S_{\phi_{in}}(f) + S_{\phi_{vco}}(f) + K_{vco}^2 S_{v_{pd}}(f)/(2\pi f)^2}{1 + T(f)^2} df \,. \tag{12}$$

The standard deviation of the RJ at the input of the phase detector is then given by

$$\sigma = \sqrt{\mathrm{var}(\phi_e)} \,. \tag{13}$$

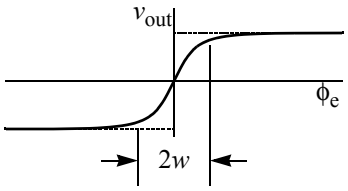
Now, as described in Section 3, the BER is then determined using Table 1 and (1). For example, assume a CDR with the width of the DJ histogram being 0.15 UI and the variance of the RJ being $\sigma = 0.05$ UI, then T_{slack} is 0.35 UI, $\rho = T_{slack}/\sigma = 7$, and the BER is just over 10^{-12}.

6.4 Large Signal Behavior

The previous section assumed the signals being applied to the phase-domain model were small, meaning that they were not large enough to excite the model to behave nonlinearly. Using a small-signal model is suitable as long as $|\phi_e| \ll w$, where w is the RMS width of the transition in the PD transfer curve, as shown in Figure 12. Once $|\phi_e|$ becomes larger than w the loop begins to slew. When this occurs the loop gain begins to drop, which increases ϕ_e, which causes the gain to drop further. This positive feedback loop causes ϕ_e to increase dramatically and create errors whenever the loop is slewing and $\phi_{in} \gg$ ½ UI (when slewing this will always be true except possibly at very high frequencies). Thus, to avoid errors it is important to keep $|\phi_e| < w$. This condition places a frequency dependent bound on the input signal that can be computed from (9) by substituting w for ϕ_e and assuming ϕ_{pd} and ϕ_{vco} are zero,

$$\phi_{in}(f) < \max((w(1 + T(f)), 0.5)) \,. \tag{14}$$

FIGURE 12 *The width of the phase detector transfer function.*



7 Performance Metrics

Generally the performance of the components in a serial link is not measured in terms of BER, where a lower BER would be considered better. Rather, the performance of receivers is characterized in terms of what level of impairments can be tolerated while still maintaining the required BER and the performance of transmitters is characterized by the level of impairments in the signal they produce. However, with the models that were constructed to explore BER, we now have all we need to predict all of these performance metrics.

The performance of a CDR is commonly characterized using several metrics: jitter generation, jitter transfer, and jitter tolerance [2,11]. Normally jitter tolerance and perhaps jitter generation are sufficient. However, for those applications where the CDR might be used in a repeater (a combination of both a receiver and a transmitter), as might be the case in SONET (Synchronous Optical Network) systems, all three are needed. These three metrics answer two basic questions of interest:

1. How much jitter can there be at the input of the system and still have it function properly? This is jitter tolerance.
2. How much jitter will there be at the output? This can be further broken down into two parts:
   a. How much jitter does the system create on its own? This is jitter generation.
   b. How much jitter passes through the system from its input to its output. This is jitter transfer.

One should note that the performance requirements of SONET CDRs differ from the performance requirements for the high speed link CDRs used in chip-to-chip communication. In particular, high speed link CDR circuits typically do not need to meet a jitter transfer specification, and are instead focused on simply achieving adequately low bit error rates. In addition, the jitter generation performance of high speed link CDR circuits is often much more relaxed than required of SONET CDR circuits.

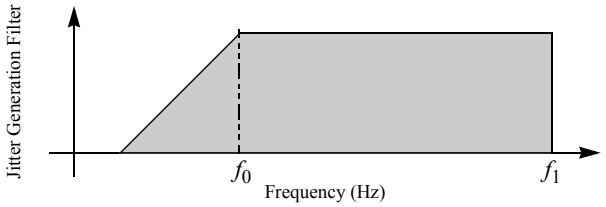
7.1 Jitter Generation

Jitter generation is the measure of the intrinsic jitter produced by the CDR and is measured at its output. Jitter generation is measured by applying a representative input signal with no jitter to the input of the CDR, and measuring its output jitter. Jitter generation is usually specified as an RMS period jitter value. The representative input would be a sequence of randomly chosen legal symbols corrupted by inter-symbol interference and other common impairments, but not jitter (the effect of jitter is accounted for with jitter transfer).

The easiest way of predicting jitter generation is to measure the width of the probability density function (PDF) of the total jitter (TJ) in an RMS sense. This is done by combining the PDFs of the deterministic and random jitter (DJ and RJ) components of the CDR when it is driven with a jitter free input. The PDF of DJ is measured by applying a clean (jitter free) representative input signal to either a voltage-domain model of the CDR (transistor-level or behavioral) and simulating for long enough to build up an accurate measurement of the jitter histogram (as shown in Figure 1). Then a phase-domain model is constructed and used as described in Section 6 to predict RJ. The TJ PDF is found by convolving the DJ and RJ PDFs. Once the TJ PDF is known, its width can be measured.

Low frequency jitter is largely ignored in SONET/SDH equipment because a downstream CDR would easily be able to track the signal in the presence of a low frequency jitter. As such the jitter generation is specified not to exceed 10 mUI RMS when measured using a high-pass filter with a 12 kHz cut-off frequency (50 kHz for OC-192) [1, 2, 9] as shown in Figure 13. This is a more difficult measurement to make with this methodology as it involves actually extracting the DJ sequence and computing its power spectral density (PSD), and then combining it with the PSD of the RJ as computed with the phase domain model, multiplying the result by the square of the transfer function of a first-order high-pass filter and then integrating the result to get the total jitter power.

FIGURE 13 *Jitter generation versus frequency for SONET specification [15].*



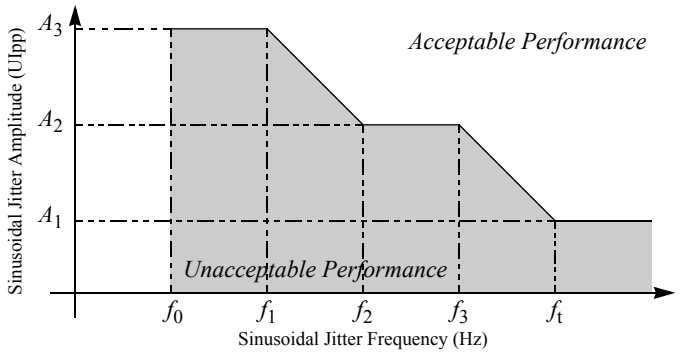
OC Level	Rate	f_0	f_1	Total Jitter
1	51.84 Mb/s	12 kHz	400 kHz	10 mUI RMS
3	155.52 Mb/s	12 kHz	1.3 MHz	10 mUI RMS
12	622.08 Mb/s	12 kHz	5 MHz	10 mUI RMS
48	2.48832 Gb/s	12 kHz	20 MHz	10 mUI RMS
192	9.95328 Gb/s	50 kHz	80 MHz	10 mUI RMS

Notice that the easier way of determining the jitter generation will always give a larger value than the SONET method as it includes the low frequency jitter. So, if the jitter generation computed by the simpler approach indicates that the design meets its requirements, there is no need to employ the more difficult approach.

7.2 Jitter Tolerance

Jitter tolerance is a measure of the ability of a CDR to operate properly (i.e., remain in lock and maintain an acceptable error rate) when jitter is applied to its input. Jitter tolerance is usually specified as minimum jitter amplitude that must be tolerated while not exceeding a specific BER as a function of frequency, as shown in Figure 14.

FIGURE 14 *Input jitter level versus frequency for SONET jitter tolerance specification [15].*



OC Level	Rate (Mb/s)	f_0 (Hz)	f_1 (Hz)	f_2 (Hz)	f_3 (Hz)	f_t (Hz)	A_1 (UI_{pp})	A_2 (UI_{pp})	A_3 (UI_{pp})
1	51.84 M/s	10	30	300	2k	20k	0.15	1.5	15
3	155.52	10	30	300	6.5k	65k	0.15	1.5	15
12	622.08	10	30	300	25k	250k	0.15	1.5	15
48	2488.32	10	600	6k	100k	1M	0.15	1.5	15
192	9953.28	10	2k	20k	400k	4M	0.15	1.5	15

In a jitter tolerance measurement, a coded pseudorandom symbol sequence is applied to the CDR. The phase of the sequence is modulated sinusoidally at a given frequency and the amplitude of the modulation is increased until a particular bit-error rate limit is exceeded, typically 10^{-12}.

The response to the sinusoidal jitter can be easily predicted with the phase domain model. Recall that an error occurs when the jitter on a particular cycle exceeds ½ UI. As shown in Figure 5 part of that ½ UI is used up by deterministic jitter that is present even when the input data stream exhibits no sinusoidal jitter. This leaves T_{slack}. Random jitter also consumes some of this interval, an amount equal to $\rho\sigma$ where σ is the standard deviation of the random jitter and is computed in (13) and ρ is determined by the required BER using Table 1. Thus, if the response to the sinusoidal jitter on the input at the phase detector input as computed by the phase-domain model is less than $T_{slack} - \rho\sigma$, then the system will satisfy the required BER. In particular, the response of interest is the difference in phase between the data input and the feedback input of the phase detector, or ϕ_e. Assume it is measured in UIs. Then, to assure the BER bound is met,

$$\phi_e < T_{slack} - \rho\sigma. \tag{15}$$

The phase domain model can be used to compute ϕ_e from ϕ_{in}, the input phase, using

$$\phi_e = \frac{\phi_{in}}{T(f) + 1}. \tag{16}$$

where T is the loop gain from (11). From this we can see that the jitter tolerance, or maximum allowable input jitter, is

$$J_{tol_{RJ}} = (T_{slack} - \rho\sigma)(T(f) + 1). \tag{17}$$

Recall from Section 6.4 that the slewing also places a bound on the size of the input. In this case

$$J_{tol_{slewing}} = w(T(f) + 1). \tag{18}$$

Both criteria must be satisfied,

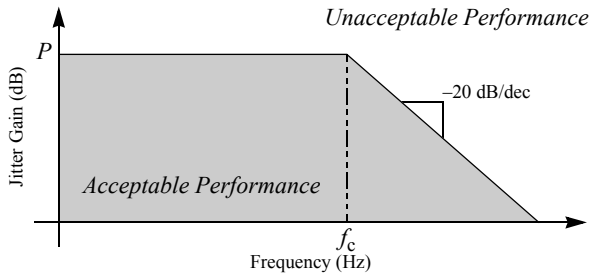
$$J_{tol} = \min(J_{tol_{RJ}}, J_{tol_{slewing}}). \tag{19}$$

7.3 Jitter Transfer

Jitter transfer (or jitter gain) relates the magnitude of jitter at the output of a CDR to the amount of jitter at its input. The required jitter transfer performance for the various SONET standards are shown in Figure 15. This specification is used to place a cap on the amount of peaking in the jitter transfer function. As such, it is a useful requirement when the CDR might be used in a repeater, because it in these situations it is important that jitter does not build as it traverses multiple repeaters.

To measure jitter transfer a coded pseudorandom sequence of symbols is applied to the CDR. The phase of the sequence is modulated sinusoidally at a given frequency and the jitter in the clock output of the CDR is measured. The ratio of the two as a function of frequency is jitter transfer,

FIGURE 15 *SONET jitter transfer specification [15].*



OC *Level*	Rate	f_c	P
1	51.84 M/s	40 kHz	0.1 dB
3	155.52 Mb/s	130 kHz	0.1 dB
12	622.08 Mb/s	500 kHz	0.1 dB
48	2.48832 Gb/s	2 MHz	0.1 dB
192	9.95328 Gb/s	120 kHz	0.1 dB

$$J_{\text{xfer}} = \frac{\phi_{\text{out}}}{\phi_{\text{in}}}. \tag{20}$$

Using the phase domain model of Figure 7, the jitter transfer is easily found to be

$$J_{\text{xfer}} = \frac{T(f)}{T(f)+1}. \tag{21}$$

This is a small-signal result. If ϕ_{in} is large enough to cause slewing then J_{xfer} will drop.[3]

8 Example

Consider a CDR for an OC-192 application and make the following assumptions about its blocks:

VCO. Assume the VCO has a nominal output frequency of 10 GHz, a gain of K_{vco} = 5 GHz/V, and $\mathcal{L}(f_{\text{m}})$ = –60 dBc where f_{m} = 100 kHz is the phase noise measurement frequency. Then from (6)

$$S_{\phi_{\text{vco}}} = \frac{\mathcal{L}(f)}{\pi} \tag{22}$$

where

$$\mathcal{L}(f) = \frac{f_{\text{m}}^2\,\mathcal{L}(f_{\text{m}})}{f^2} \tag{23}$$

and

3. On May 3, 2010, KangSub on the *Designer's Guide Forum* pointed out that deterministic jitter is missing from my treatment of jitter transfer. His post and resulting discussion can be found at *www.designers-guide.org/Forum/YaBB.pl?num=1272944729*

$$S_{\phi_{\text{vco}}} = \frac{f_{\text{m}}^2 \mathcal{L}(f_{\text{m}})}{\pi f^2} = \left(\frac{100\ \text{kHz}}{f}\right)^2 \left(\frac{1}{\pi}\frac{\mu\text{V}^2}{\text{V}^2\text{Hz}}\right). \tag{24}$$

PD. Assume now that the PD produces an output voltage of ±500 mV and when driven to produce a stream of pulses on its output generates an output noise of 51.6 μV^2 in the 1Mhz to 55 Ghz band, and so if $S_{v_{\text{pd}}}$ is assumed flat with frequency,

$$S_{v_{\text{pd}}} = \frac{n_{\text{pd}}^2}{\Delta f} = \frac{51.6\,\mu\text{V}^2}{55\ \text{GHz} - 1\ \text{MHz}}. \tag{25}$$

LF. Assume that the LF has a pole at 1 kHz and zero at 1 MHz and produces negligible noise.

Input. Further assume that the input is driven with clean 16-bit pseudo-random bit stream generator, and so

$$S_{\phi_{\text{in}}} = 0. \tag{26}$$

Finally assume that the circuit was simulated for a long time to produce the jitter histogram, shown in Figure 16. From this the maximum excursion and expected transfer function of the PD can be extracted. The maximum excursion is ±9 mUI. The expected transfer function of the PD is found by first normalizing the jitter histogram to have unit area, and then convolving it with the static transfer function of the PD, which in this case is taken to be a unit step symmetric about $\phi = 0$. The result is also shown in Figure 16. The slope of the expected transfer function at the origin is the small signal gain of the PD, which is K_{pd} = 275 V/UI.

In this case, because of the smooth bell-like shape of the histogram, it is possible to approximate the PD transfer curve with

$$V_{\text{out}}(\phi) = \tanh\left(\frac{\phi}{w}\right) \tag{27}$$

where w is the RMS width of the jitter histogram. For this circuit, w = 2 mUI. Using this model of the curve it would be possible to parameterize the desired results in terms of w, which is useful when trying to understand how the amount of DJ affects the loop gain and the various performance metrics. However, the model is not needed for a simple verification and it limits the generality of the methodology and so will not be used further.

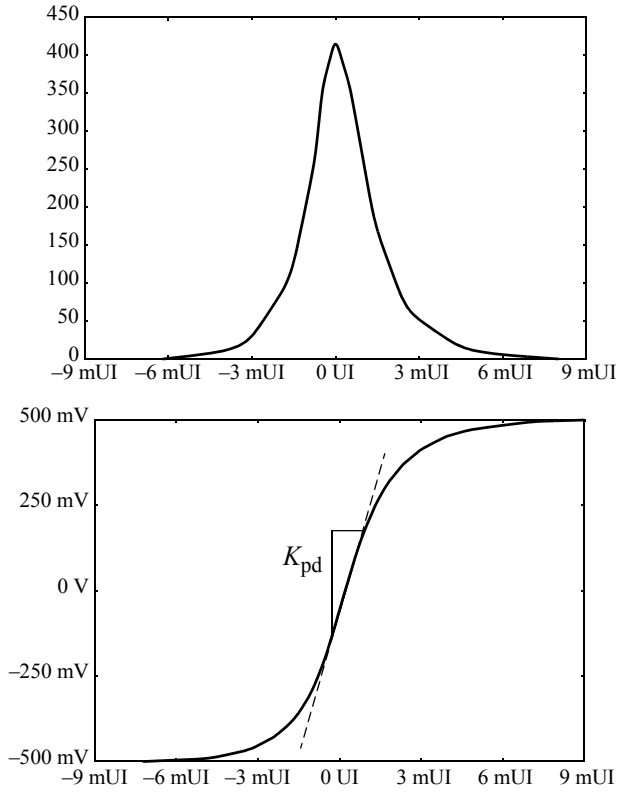
From (11) the loop gain is

$$T(f) = \frac{K_{\text{pd}} H(f) K_{\text{vco}}}{2\pi \hat{j} f} = \frac{275\ \text{V/UI}\left(1 + \frac{\hat{j}f}{1\ \text{kHz}} \middle/ \left(1 + \frac{\hat{j}f}{1\ \text{MHz}}\right)\right) 5\ \text{GHz/V}}{2\pi \hat{j} f}. \tag{28}$$

Then from (12) and (13) the standard deviation of the RJ is

$$\sigma = \sqrt{\int_0^\infty S_{\phi_{\text{e}}}(f)\,df} = \sqrt{114\,\mu\text{UI}^2} = 11\ \text{mUI}. \tag{29}$$

FIGURE 16 *The jitter histogram for the example (top) and the expected transfer function of the PD (bottom) for the example BB-CDR of Section 8.*



From (1) the system will meet its BER requirement as long as $\sigma < T_{slack}/\rho$. Here T_{slack} = 0.5 UI – 9 mUI. Thus, ρ can be as large as 46, and from Table 1 $\rho = 8$ provides a BER of better than 10^{-15}, so with a clean input symbol stream this system easily meets requirements.

For OC-192 the jitter generation should not exceed 10 mUI RMS, and with DJ_{rms} = 700 μUI and RJ = 11 mUI this design requirement is not satisfied.

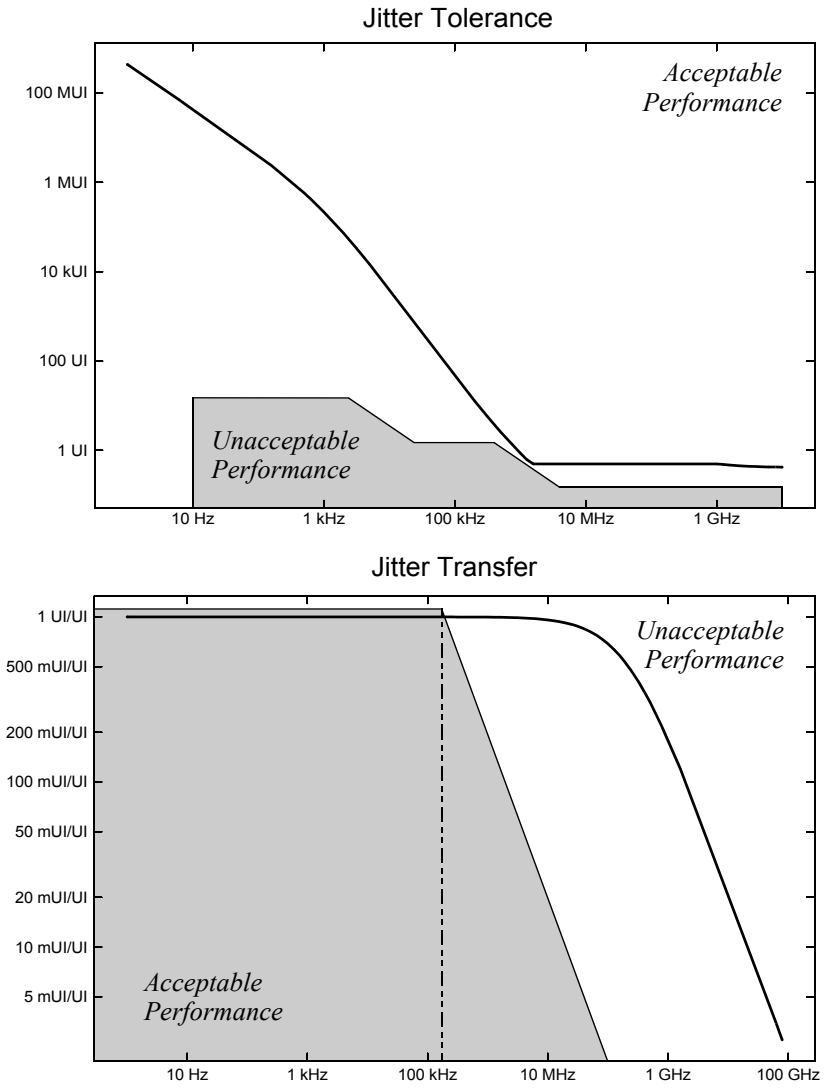
Jitter tolerance is computed using (19) and jitter transfer is computing using (21), and both are shown in Figure 17. Jitter transfer performance fails to meet the OC-192 requirements at high frequencies. There is no peaking; however at 30 MHz the bandwidth exceeds the 120 kHz limit.

The netlist for the phase domain model is given in Listing 1 and the Verilog-A code is given in Listing 2.The MDL script used to confirm the J_{tol} results is given in Listing 3.

9 Summary

An approach for predicting bit-error rate in a BB-CDR is described that provides efficient prediction of very low BER by conceptually separating the random jitter from the

FIGURE 17 *The jitter tolerance (top) and the jitter transfer (bottom) performance for the example BB-CDR of Section 8.*



LISTING 1 *The Spectre netlist for the phase-domain model.*

```
// Phase domain CDR model
simulator lang=spectre
ahdl_include “pd-cdr.va”

Jin (in 0) vsource type=sine
PD (pdout in out) pd
LF (vcoin pdout) lf
VCO (out vcoin) vco
PP (pp in out) p2p
```

LISTING 2 *The Verilog-A models for the phase-domain model (pd-cdr.va)*

```
// Phase Domain Model of CDR
`include "disciplines.vams"
`include "constants.vams"

nature Phase
        units = "UI";
        access = Ph;
        abstol = 1u;
endnature

discipline phase
        potential Phase;
enddiscipline

module pd(out, pin, nin);
        input pin, nin; phase pin, nin;
        output out; voltage out;
        parameter real Vmax = 0.5;
        parameter real Spd = 51.6u/(55G–1M);
        parameter real w = 2m;
        analog begin
                V(out) <+ Vmax*tanh(Ph(pin,nin)/w);
                V(out) <+ white_noise(Spd);
        end
endmodule

module lf(out, in);
        input in; voltage in;
        output out; voltage out;
        parameter real Hzero = 1M;
        parameter real Hpole = 1k;
        analog V(out) <+ laplace_zp(V(in), {–1M*`M_TWO_PI, 0}, {–1k*`M_TWO_PI, 0});
endmodule

module vco(out, in);
        input in; voltage in;
        output out; phase out;
        parameter real Kvco = 5G;
        parameter real Svco1 = 100k*100k*1u/`M_PI;
        analog begin
                Ph(out) <+ Kvco*idt(V(in));
                Ph(out) <+ flicker_noise(Svco1,2);
        end
endmodule

module p2p(out, pin, nin);
        input pin, nin; phase pin, nin;
        output out; phase out;
        real maximum, minimum, pp, pt, dt;
        analog begin
                @(initial_step) begin
                        maximum = Ph(pin,nin);
                        minimum = Ph(pin,nin);
                end
                if (Ph(pin,nin) > maximum)
                        maximum = Ph(pin,nin);
                if (Ph(pin,nin) < minimum)
                        minimum = Ph(pin,nin);
                @(cross(Ph(pin), +1)) begin
                        pp = maximum – minimum;
                        maximum = Ph(pin,nin);
                        minimum = Ph(pin,nin);
                        dt = $abstime – pt;
                        pt = $abstime;
                end
                Ph(out) <+ 0.5*transition(pp,0,dt/10);
        end
endmodule
```

deterministic jitter and analyzing them separately. This is an idea that is both widely useful and can be extended to handle a wide variety of impairments.

LISTING 3 *The SpectreMDL script used to confirm the J_{tol} results.*

```
alias measurement slewingError {
    int periods=5
    run tran(stop=periods/Jin:freq, errpreset='conservative)
    export real ppErr=V(pp)
}
foreach Jin:freq from swp(start=1, stop=10M, dec=10) {
    search Jin:ampl from binary(start=0.1, stop=10000, tol=1m) {
        run slewingError
    } until (slewingError->ppErr > 0.5)
    print fmt("jtol:%g\t%g\n", Jin:freq, Jin:ampl)
}
```

The method can be applied anywhere on any system where an eye diagram would be used to illustrate the system performance. Examples include any part of a serial data communication link. The CDR need not be included, but if it is, it can use any type of phase detector, specifically either linear or bang-bang. It can also be applied to clock distribution networks to determine clock skew. Finally the method naturally handles many different types of impairments.

9.1 If You Have Questions

If you have questions about what you have just read, feel free to post them on the *Forum* section of *The Designer's Guide Community* website. Do so by going to *www.designers-guide.org/Forum*. For more in depth questions, feel free to contact me in my role as a consultant at *ken@designers-guide.com*.

Acknowledgements

Jess Chen and Stephen Greenwood played a large role in developing the methodology. Mehmet Soyuer, Jean-Olivier Plouchart and Mounir Meghelli contributed by posing the challenge and providing a test case.

I would like to thank Tan Kok Siang for pointing out that the jitter transfer bandwidth for OC-192 should be 120kHz.

References

[1] Agilent Technologies. *Frequency Agile Jitter Measurement System*. Application note 1267 (5988-2749EN). *www.agilent.com*.

[2] Agilent Technologies. *Understanding Jitter and Wander Measurements and Standards*. Application note 5988-6254EN. *www.agilent.com*.

[3] Agilent Technologies. *Using Clock Jitter Analysis to Reduce BER in Serial Data Applications*. Application note 5989-5718EN. www.a*gilent.com*.

[4] J. D. H. Alexander, Clock Recovery from Random Binary Data, *Electronics Letters,* vol. 11, pp. 541-542, Oct. 1975.

[5] W. Gardner. *Introduction to Random Processes: With Applications to Signals and Systems*. McGraw-Hill, 1989.

[6] Ken Kundert. *The Role of High-Speed Serial Interfaces in MS-SoCs*. Designer's Guide Publishing, 2004. Available from *www.designers-guide.org/design*.

[7] Ken Kundert. *Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers*. Designer's Guide Publishing, 2003. Available from *www.designers-guide.org/analysis*.

[8] Jri Lee, Kenneth S. Kundert, and Behzad Razavi. Analysis and modeling of bang-bang clock and data recovery circuits. *IEEE Journal of Solid-State Circuits*, vol. 39, no 9, September 2004, pp. 1571-1580.

[9] Micrel-Synergy. *SONET OC-3 Jitter Measurement*. Application note AN-04. *www.micrel.com*.

[10] Joel Phillips and Ken Kundert. *Noise in Mixers, Oscillators, Samplers, and Logic — An Introduction to Cyclostationary Noise*. Designer's Guide Publishing, 2000. Available from *www.designers-guide.org/theory.*

[11] Behzad Razavi. *Design of Integrated Circuits for Optical Communications*. McGraw Hill, 2003.

[12] Tektronix. Understanding and Characterizing Timing Jitter. Application note 55W-16146-1. Available from *www.tektronix.com/jitter.*

[13] Richard C. Walker. Designing Bang-bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems, pp. 34-45, a chapter appearing in *Phase-Locking in High-Performance Systems – From Devices to Architectures*, edited by Behzad Razavi, IEEE Press, 2003, ISBN 0-471-44727-7. Also available from *www.omnisterra.com/walker.*

[14] T. C. Weigandt, B. Kim, and P. R. Gray. Jitter in ring oscillators. *1994 IEEE International Symposium on Circuits and Systems* (ISCAS-94), vol. 4, 1994, pp. 27-30.

[15] Telecordia Technologies. *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria*. GR-253-CORE, Issue 3, September 2000.

[16] V. Stojanovic and M. Horowitz. Modeling and analysis of high speed links. *IEEE Custom Integrated Circuits Conference Digest of Technical Papers*, 2003, pp. 589-594.