DesignCon 2011

Studying Clock Recovery Performance using IBIS-AMI Models

Barry Katz, Signal Integrity Software, Inc.
bkatz@sisoft.com

Dr. Michael Steinberger, Signal Integrity Software, Inc.
msteinb@sisoft.com

Abstract

Clock recovery loop behavior can seriously affect the performance of a high-speed serial channel and yet the only data that is generally available is that which is inferred through jitter tolerance testing for a small number of channels.

This paper introduces a method for extracting the jitter transfer function, jitter tolerance and pattern dependent jitter from an IBIS-AMI model under realistic channel conditions, and presents example results from a generic bang-bang clock recovery design and seven commercial SerDes designs.

Author Biographies

Barry Katz, President and CTO for SiSoft, founded SiSoft in 1995. As CTO, Barry is responsible for leading the definition and development of SiSoft's products. He has devoted much of his efforts at SiSoft to delivering a comprehensive design methodology, software tools, and expert consulting to solve the problems faced by designers of leading edge high-speed systems. He was the founding chairman of the IBIS Quality committee. Barry received an MSEE degree from Carnegie Mellon and a BSEE degree from the University of Florida.

Dr. Michael Steinberger is currently responsible for leading the development of SiSoft's serial link analysis products. He has over 30 years experience in the design and analysis of very high speed electronic circuits. Prior to joining SiSoft, Dr. Steinberger worked at Cray Inc., where he designed very high density interconnects and increased the data rate and path lengths to the state of the art. Mike holds a B.S. from the California Institute of Technology and a Ph.D. from the University of Southern California, and has been awarded 13 U.S. patents.

Problem Description

This paper addresses the performance evaluation of the clock recovery loop during the design of either the SerDes macro or an end to end high speed serial channel, and the subsequent correlation of the clock recovery loop model to measured data once hardware becomes available.

Regardless whether bit error rate of a high speed serial link is estimated using jitter analysis techniques (see, for example, [1]) or as a conditional probability integral (see, for example, [2]), the probability density function (PDF) of the clock is central to the calculation. Since the clock is provided by a clock recovery loop, the jitter transfer function, jitter injection, and pattern dependent jitter of the clock recovery loop directly affect the result.

As part of any thorough analysis of a high speed serial channel design, it is therefore important to model the clock recovery loop, evaluate the clock recovery loop performance under realistic channel conditions, and eventually correlate the clock recovery loop model to measured data. There are several obstacles to be overcome in order to achieve this, however:

1. The design of the clock recovery loop may not be available to the person responsible for the performance analysis. This will almost always be the case for system developers, and is sometimes the case for SerDes modelers as well.

2. The clock recovery loop cannot be evaluated in isolation, but rather must be evaluated in the context of realistic channel conditions, including the channel loss and the equalization solution. This increases the complexity of the modeling task.

3. Techniques suitable for analysis and time domain simulation may not be suitable for hardware measurement, and vice versa. This is especially the case for the clock recovery loop because the recovered clock and the clock to data timing may not be directly visible in hardware. Conversely, the most common hardware measurement technique is jitter tolerance testing, and a direct time domain simulation of that technique is not practical.

This paper therefore presents solutions to the following two problems:

1. Derive an estimate of the jitter transfer function, jitter injection, and pattern dependent jitter of a clock recovery loop under realistic channel conditions through time domain simulation.

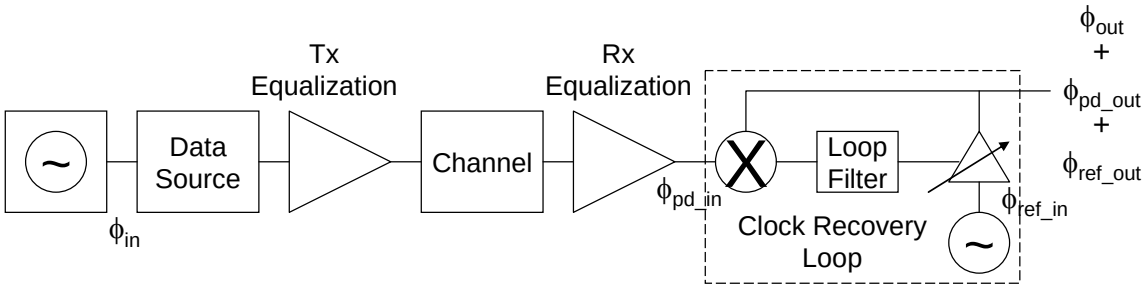
2. Validate the model of a clock recovery loop by directly comparing simulated results to measured data, again under realistic channel conditions.

The solution is to obtain the jitter transfer function and pattern dependent jitter using IBIS-AMI (I/O Buffer Information Specification – Algorithmic Modeling Interface) models [3], [4] in a different way than was originally intended, calculate the jitter

injection function from that data, and then calculate a jitter tolerance curve that can be compared directly to measured data.

Background

Figure 1 illustrates the variables that will be used to describe clock recovery loop behavior.



ϕ_{in}	Transmitter phase modulation
ϕ_{out}	Transmitter phase modulation component of recovered clock
ϕ_{pd_in}	Pattern dependent jitter at phase detector input
ϕ_{pd_out}	Pattern dependent jitter component of recovered clock
ϕ_{ref_in}	Receive reference clock phase noise
ϕ_{ref_out}	Reference clock phase noise component of recovered clock

Figure 1: Clock recovery loop phase components

In Figure 1, the transmitter is driven by a clock whose phase may be variable. This transmitter phase will result in an output phase which is intended to track the input phase.

In addition, there will be output phase variations due to variations in the data pattern, and therefore called pattern dependent jitter.

Finally, most clock recovery loops phase modulate the output of an internal or external oscillator; and regardless what form this oscillator takes, its output will have some phase noise with respect to the transmit clock. While the clock recovery loop will be able to track out some of this oscillator phase noise, some of this phase noise will be present in the recovered clock. This phase noise will be referred to as input and output reference phase noise.

Jitter Transfer Function

The underlying assumption is that the output phase of the clock recovery loop is a linear time invariant function of the input phase. From this assumption it can be proven (e.g., [5]) that the output phase can be expressed in terms of the input phase as

$$\phi_{out}(t) = \phi_{in}(t) \otimes h_{jtf}(t) \equiv \int_{-\infty}^{\infty} \phi_{in}(t-\tau) \cdot h_{jtf}(\tau) d\tau$$

Taking the Fourier transform of $h_{jtf}(t)$ produces the jitter transfer function. That is,

$$\Phi_{out}(\omega) = H_{jtf}(\omega) \cdot \Phi_{in}(\omega)$$

This means, among other things, that if the input phase is a sine wave, then the output phase will be a sine wave at the same frequency, though most likely a different amplitude. It also means that if the input phase is the sum of multiple sine waves, then the output phase will have the same frequency components, each at its own amplitude.

Since the goal of the clock recovery loop is that the output phase should match the input phase as closely as possible, it is useful to define a loop error function which indicates how well this goal has been achieved.

$$\Phi_{out}(\omega) - \Phi_{in}(\omega) = (H_{jtf}(\omega) - 1) \cdot \Phi_{in}(\omega) \equiv H_e(\omega) \cdot \Phi_{in}(\omega)$$

Pattern Dependent Jitter

The recovered clock phase is derived in some way from the transition times between individual pairs of bits. While on average these transition times reflect the phase of the transmit clock, they are also affected by intersymbol interference. This phase variation due to intersymbol interference is an additional input into the clock recovery loop that the loop is intended to filter out. However, some of this phase noise will remain at the output of the clock recovery loop.

Note that at the input to the clock recovery loop, there is no way to distinguish pattern dependent jitter from phase variation of the transmit clock. Therefore, the jitter transfer function of the clock recovery loop will be applied in the same way to both the transmitter phase and the pattern dependent jitter.

$$\Phi_{pd\_out}(\omega) = H_{jtf}(\omega) \cdot \Phi_{pd\_in}(\omega)$$

While pattern dependent jitter can be reduced somewhat by minimizing the intersymbol interference, and by making sure that all data transitions are used, the only other design parameter that can be used to control pattern dependent jitter is clock recovery loop bandwidth, or in other words the bandwidth of the jitter transfer function. Choice of clock recovery loop bandwidth is therefore a tradeoff between increasing loop bandwidth to improve tracking of the transmit clock and reducing loop bandwidth to minimize pattern dependent jitter.

Reference Clock Phase Noise

The clock recovery loop phase detector outputs the phase difference between the transmitter clock phase, as evident from the received data signal, and the selected phase of the local reference clock. Since the reference clock phase noise enters the phase detector from a different port from the transmitter clock phase and pattern dependent jitter, it is filtered by the clock recovery loop’s error function rather than its jitter transfer function.

$$\Phi_{ref\_out}(\omega) = H_e(\omega) \cdot \Phi_{ref\_in}(\omega)$$

Thus, the effects of reference clock phase noise can be estimated using the loop error function as derived by the methods reported in this paper. This paper will not pursue this application any further, however.

Jitter Tolerance Test

This section provides a description of jitter tolerance testing that will later be used to predict jitter tolerance test results from a combination of analysis and simulation. Figure 2 illustrates the jitter tolerance test setup.

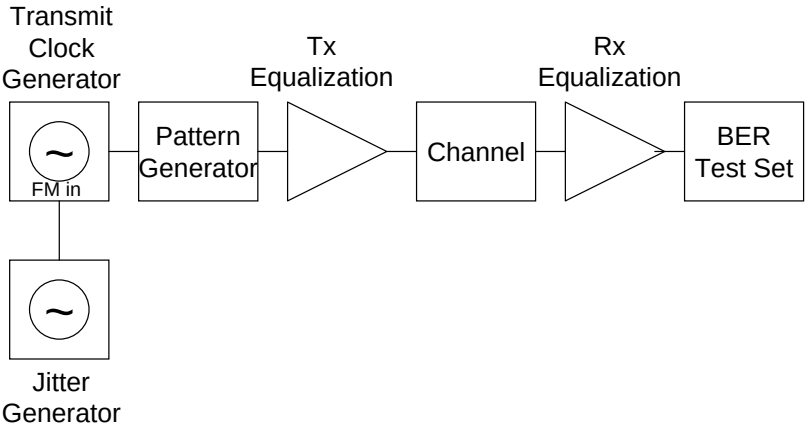


Figure 2: Jitter tolerance test setup

In this setup, a sinusoidal stimulus from the jitter generator is used to modulate the transmit clock, thus producing a sinusoidal jitter component. For each of several sinusoidal jitter frequencies, the amplitude of the jitter is increased until a target bit error rate is reported by the bit error rate test set. The amplitude of the jitter, measured in unit intervals (UI), is then recorded as the tolerated jitter for that frequency.

NOTE: Because the jitter input to the transmit clock generator is through the FM input port, the calibration of this setup must account for the fact that phase is the integral of frequency.

While jitter tolerance testing is in widespread use, it is not a quick measurement to make because the target bit error rate is low enough that it takes a few minutes to measure. Therefore a direct implementation of this method using time domain simulation is not practical.

The solution for time domain simulation is to break the jitter tolerance calculation into two parts: a timing margin and a clock timing error.

The timing margin is the absolute value of the minimum timing offset that will increase the bit error rate so that it equals the target bit error rate. This margin must be evaluated with the specified channel, equalization solution, pattern dependent jitter and reference clock phase noise, but without any additional jitter introduced at the transmitter.

The clock timing error is the transmit clock tracking error, as described in previous sections of this paper.

Given these definitions, the target bit error rate will be reached when the maximum clock timing error is greater than or equal to the timing margin for a sufficiently large fraction of the time. From a rigorous mathematical perspective, this fraction must be nonzero, and is a function of the bathtub curve for the link. From a practical perspective, however, the bit error rate increases so rapidly once the timing margin has been exceeded, that the fraction of the time that the timing margin is exceeded will typically be quite small. To keep the analysis as simple as possible, we will adopt the approximation that the target bit error rate will occur when the maximum clock timing error equals the timing margin.

This assumption will cause the jitter tolerance to be underestimated slightly. If this under-estimation is important, the timing margin used in the calculation should be the timing margin for a bit error rate which is perhaps one order of magnitude greater than the target bit error rate.

The overall analysis procedure is therefore

1. Determine the timing margin for the channel and equalization using either statistical analysis or time domain simulation. (The timing margin is independent of the jitter frequency, and is therefore a constant for the remainder of the analysis.)

2. Determine the jitter transfer function, and therefore the error function of the clock recovery loop.

3. For each jitter frequency, divide the timing margin (in UI) by the magnitude of the clock recovery loop error function to obtain the jitter tolerance for that frequency.

Given the timing margin τ_m for a given channel and equalization solution, the jitter tolerance as a function of jitter frequency is

$$\tau_{jt}(\omega) = \frac{\tau_m}{|H_e(w)|}$$

IBIS-AMI Modeling

IBIS-AMI modeling was developed in 2007 and introduced in 2008 as a standardized way to model SerDes IP. The original goals were

1. Model the equalization behavior of the SerDes so that it can be part of the performance analysis of an end to end high speed serial channel.
2. Create models that will run in multiple EDA environment from multiple vendors.
3. Enable the performance analysis of channels for which the receiver is being supplied by a different vendor than the transmitter.
4. Protect proprietary information about the SerDes design.

5. Enable a wide variety of simulations and analyses.

An IBIS-AMI model is a library of compiled functions with a specified function signature, plus some supporting files. The model is written using a general purpose programming language (typically C), giving the model developer a great deal of flexibility in the implementation of the model. Yet since the model is delivered as a compiled library, the proprietary details of the SerDes being modeled are protected. Furthermore, since the function signatures are standardized, the model can be run in any EDA tool that supports the standard.

The IBIS-AMI interface defines two different ways that the data path behavior of a transmitter or receiver can be expressed. One mode of behavior is to output the impulse response of the circuit for a given channel impulse response, and the other mode is to process a segment of time domain waveform, with the expectation that the model will be required to process many successive waveform segments in a single simulation. Every IBIS-AMI model must support at least one of these modes, and the majority of models currently available support both modes.

When used in a time domain simulation, the IBIS-AMI model of a receiver can produce an array of times at which the edges of the recovered clock occur. This array is called the clock_ticks array. The primary purpose of the clock_ticks array is to make it possible for the EDA tool to display the eye diagram and estimate the bit error rate. The original intent was that each clock tick would define the beginning of a waveform segment to be plotted in the eye diagram, in the same way that a clock signal is used to trigger an eye diagram on an oscilloscope.

If the IBIS-AMI model was written to accurately represent the architecture of the clock recovery loop, then the content of the clock_ticks array is a direct representation of the behavior of the clock recovery loop. In this case, the clock_ticks array can also be used to analyze the behavior of the clock recovery loop.

Method

The jitter transfer function of a clock recovery loop within an IBIS-AMI model can be estimated in a time domain simulation by injecting a known amount of jitter into the transmitted signal and observing how much of that jitter is present in the clock_ticks array that is output from the model. That is, both the input jitter and output jitter are known, so it’s possible to calculate their discrete Fourier transform and then calculate the jitter transfer function through the equation

$$H_{jtf}(\omega) = \frac{\Phi_{out}(\omega)}{\Phi_{in}(\omega)}$$

This method is complicated by the fact that the phase noise in the clock_ticks array is the sum of contributions from the transmit jitter, pattern dependent jitter, and reference clock phase noise. If one can assume that each of these phase noise contributors is statistically

independent, then the effects of the pattern dependent jitter and reference clock phase noise can be reduced by averaging over many sets of samples.

While there are several mathematically equivalent ways to achieve the same result, the averaging method chosen was
1. Accumulate a series of input and output phase samples.
2. Calculate the discrete Fourier transform of each set of samples.
3. Calculate the apparent jitter transfer function for the set of samples.
4. Add the Fourier transformed data to an accumulating frequency domain average for the input phase, output phase, and jitter transfer function.
NOTE: The complex values of the frequency domain data were retained, thus preserving phase as well as magnitude information.

For the data reported in this paper, the time window for a single series of samples was 65,000 bits. Simulations of 10 million bits (averaging 150 sample series) produced usable results, while 100 million bits (averaging 1500 samples series) were used to produce the more presentable results shown in this paper. These simulations took a couple of hours each to run, which isn't exactly quick, but it is practical.

Several choices of transmit phase noise stimulus wave shape were viable; however, the transmit phase noise stimulus chosen was uniform spectral density Gaussian phase noise, commonly referred to as random jitter (RJ).

The RJ level chosen was 0.05 UI. It is an unavoidable principle of measurement that the act of measuring changes that which is being measured, and this principle certainly applies to the method being reported here. In particular, the transmit RJ will necessarily change the behavior of the clock recovery loop by at least some small amount. For example, the results measured using an RJ level that increased the bit error rate to 10-3 would certainly be suspect. Conversely, if the RJ level isn't high enough, then the component of RJ at the clock recovery loop output will be difficult to detect. The RJ level of 0.05 UI was chosen as a compromise between these two extremes. It is small enough that the eye diagrams were open for most of the cases reported, but large enough to be detected at the output of the clock recovery loop. Figure 3 shows a typical eye diagram without and with 0.05 UI of transmit RJ.

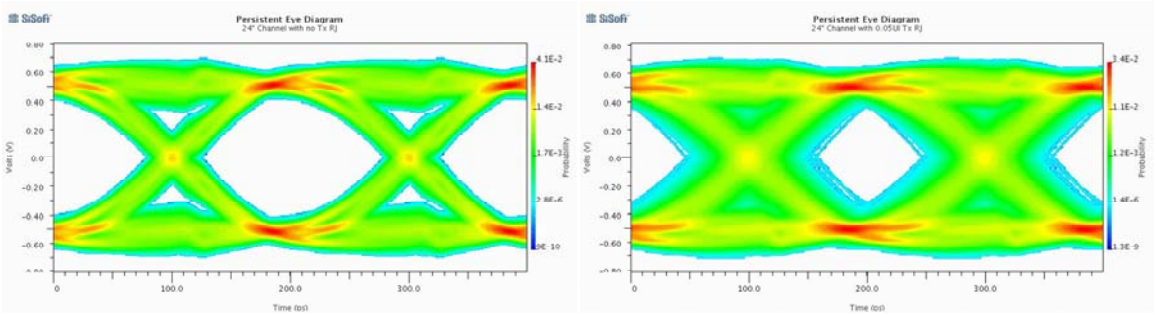


Figure 3: Typical eye diagram without and with 0.05 UI Tx RJ

Figure 4 below will demonstrate that the estimated pattern dependent jitter is approximately the same with and without the 0.05 UI of Tx RJ. This suggests, but does

not prove, that the estimated jitter transfer function with this level of Tx RJ also represents the jitter transfer function without Tx RJ.

There are also several possible ways to estimate the pattern dependent jitter. One way is simply to set the transmit RJ to zero and measure the output phase spectral density. Another way is to
1. Set the RJ as described above.
2. Calculate the averaged jitter transfer function as above.
3. Multiply the input RJ times the jitter transfer function and subtract the result from the total output phase noise.
At least for 0.05 UI Tx TJ, the two methods yield comparable results.

Each of these two methods has its advantages and disadvantages. As compared to determining the jitter transfer function and pattern dependent jitter in a single simulation, determining the pattern dependent jitter separately yields results with much lower measurement noise, at the expense of a separate simulation. Figure 4 is an example comparison of the estimated pattern dependent jitter using the single simulation and dedicated simulation approaches.

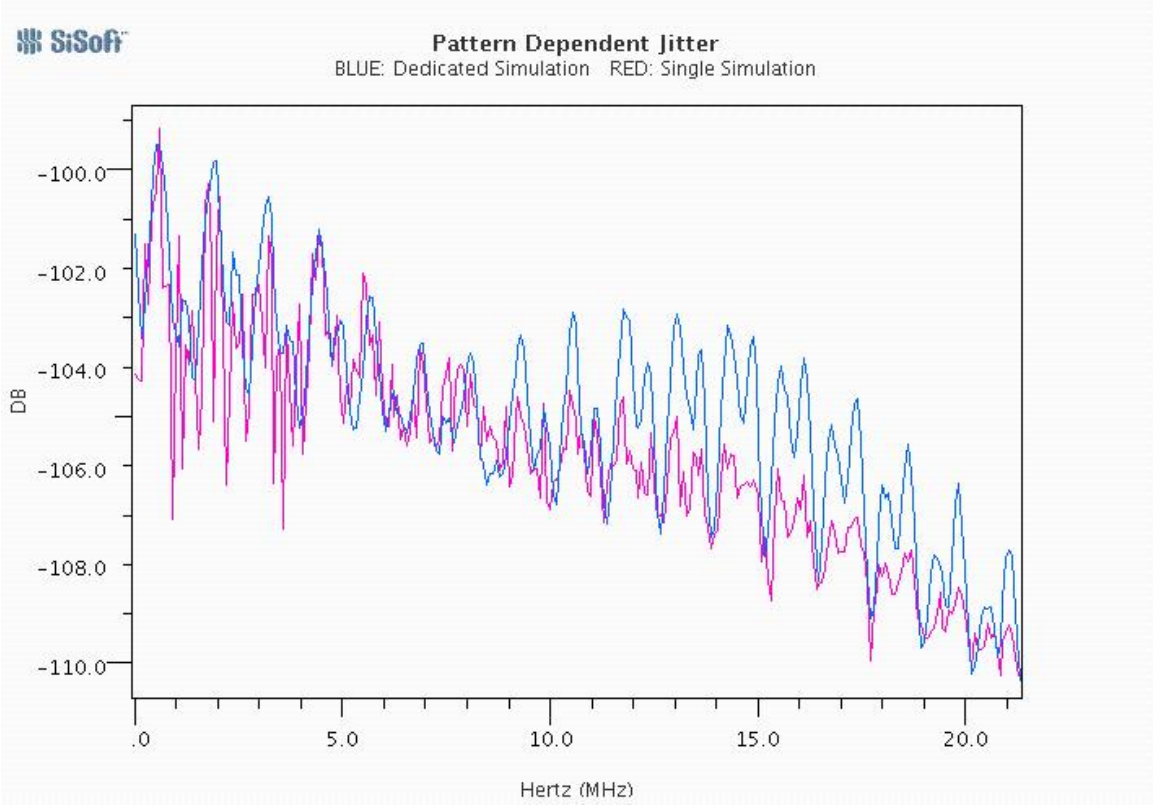


Figure 4: Pattern dependent jitter as estimated using the single simulation and dedicated simulation approaches

The results reported in this paper were obtained using the single simulation approach.

Test Conditions

As the results will demonstrate, the performance of the clock recovery loop is affected by the channel and equalization. Therefore, each IBIS-AMI model was evaluated with the same XAUI-HM-Zd transmission paths used in [2]. Transmission path lengths were 4", 16" and 24". The data rate was 5.0 Gb/s. In each case, the transmitter was the generic transmitter without equalization, and the receive equalization was adjusted to obtain an open eye diagram and a low bit error rate.

Results

Generic Bang-Bang Clock Recovery

As described in [4], SiSoft supplies a receiver model with a generic bang-bang clock recovery loop. This model was used for preliminary investigations.

Figure 5 is a graph of the magnitude of the jitter transfer function for each of the three path lengths. In this Figure, the 3dB bandwidth of the clock recovery loop decreases from 12.1 MHz for the 4" path to 6.1 MHz for the 24" path. Figure 4 also shows a rational transfer function fit to the 4" path jitter transfer function. The poles are at -14 MHz and -33 MHz.

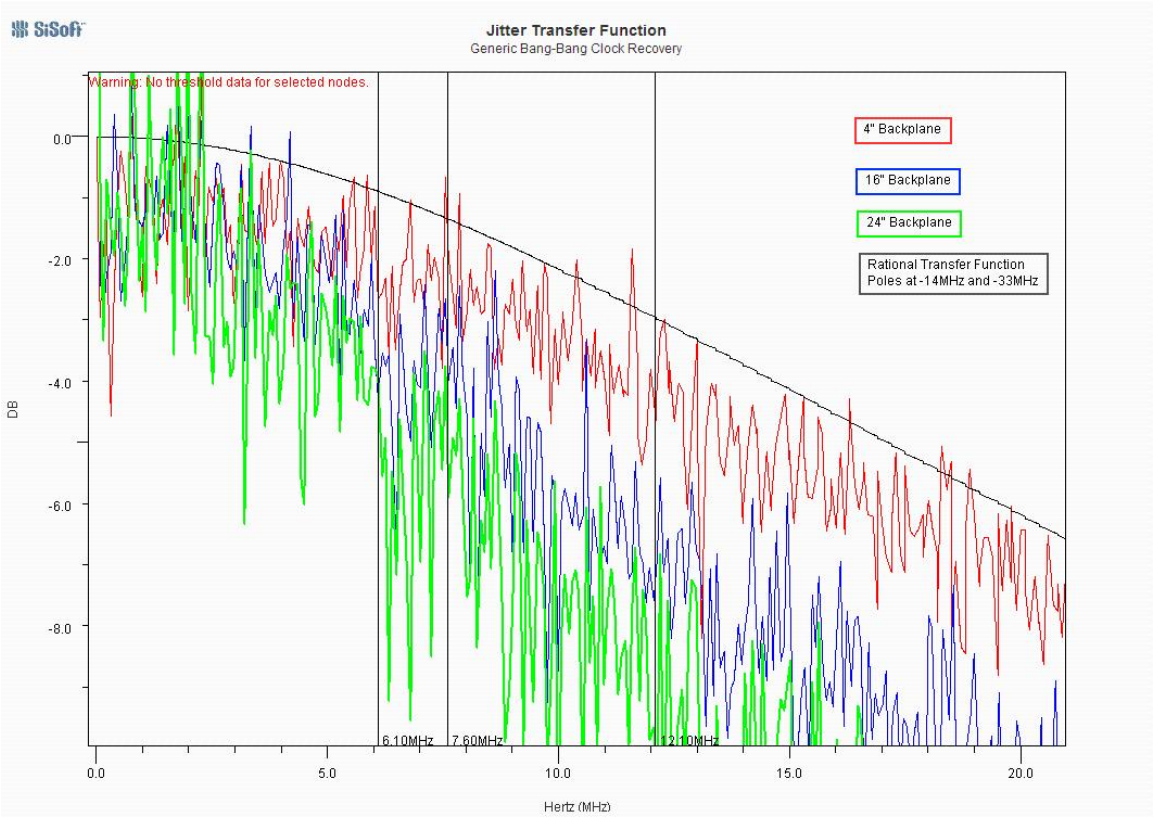


Figure 5: Jitter transfer function magnitude as a function of frequency and transmission path length

As stated earlier, the calculation of the jitter transfer function preserves the phase as well as magnitude information. It is therefore possible to create a polar plot of the jitter transfer function. Figure 6 is an example of such a plot for the generic bang-bang clock recovery loop.

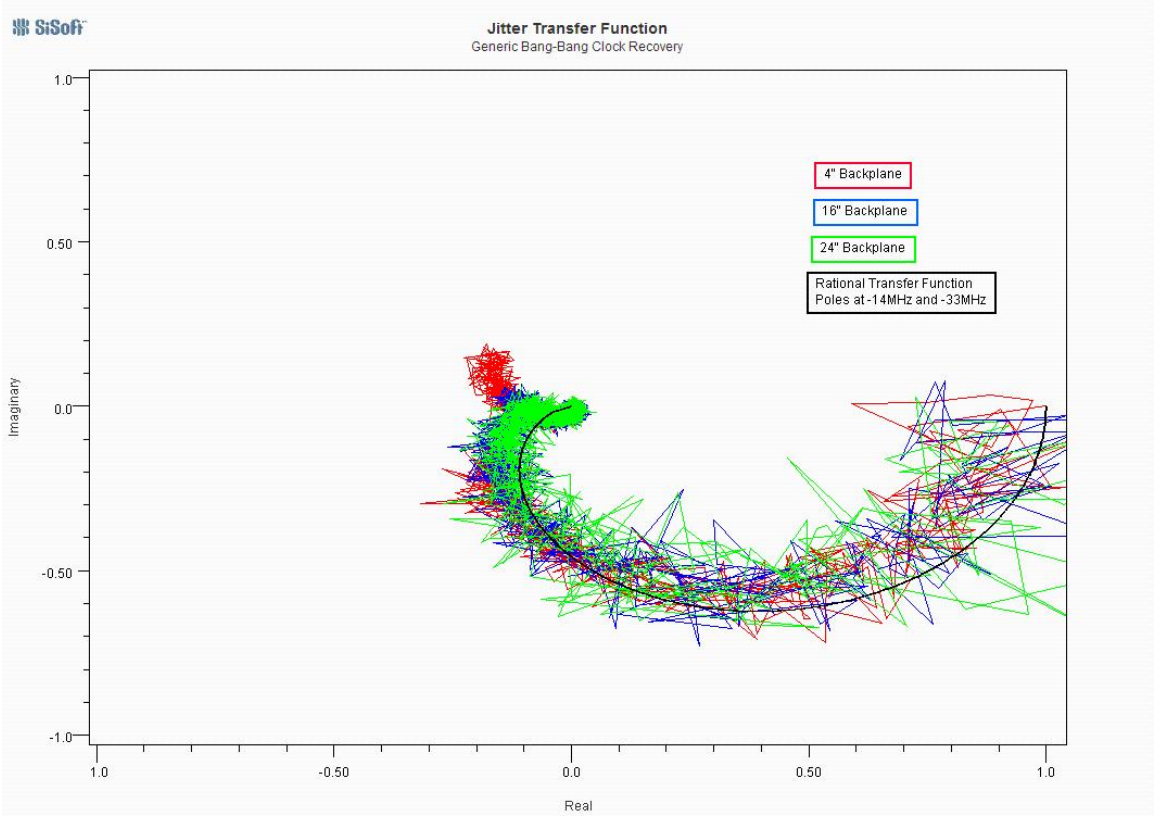


Figure 6: Polar plot of jitter transfer functions

Note that although the clock recovery loop bandwidth varies as a function of transmission path length, the polar plot of the jitter transfer function remains the same. This characteristic was observed for every SerDes design studied for this paper.

Each clock recovery loop has a phase detector that somehow compares the data transition times to the clock edges. One possible explanation for the invariance of the polar shape is that phase detector sensitivity scales the loop gain, and therefore the closed loop bandwidth, while all other factors in the open loop response remain unchanged. Intersymbol interference reduces the sensitivity of this phase detection mechanism, and the reduced phase detector sensitivity reduces the clock recovery loop bandwidth. The principle is illustrated in Figure 7.

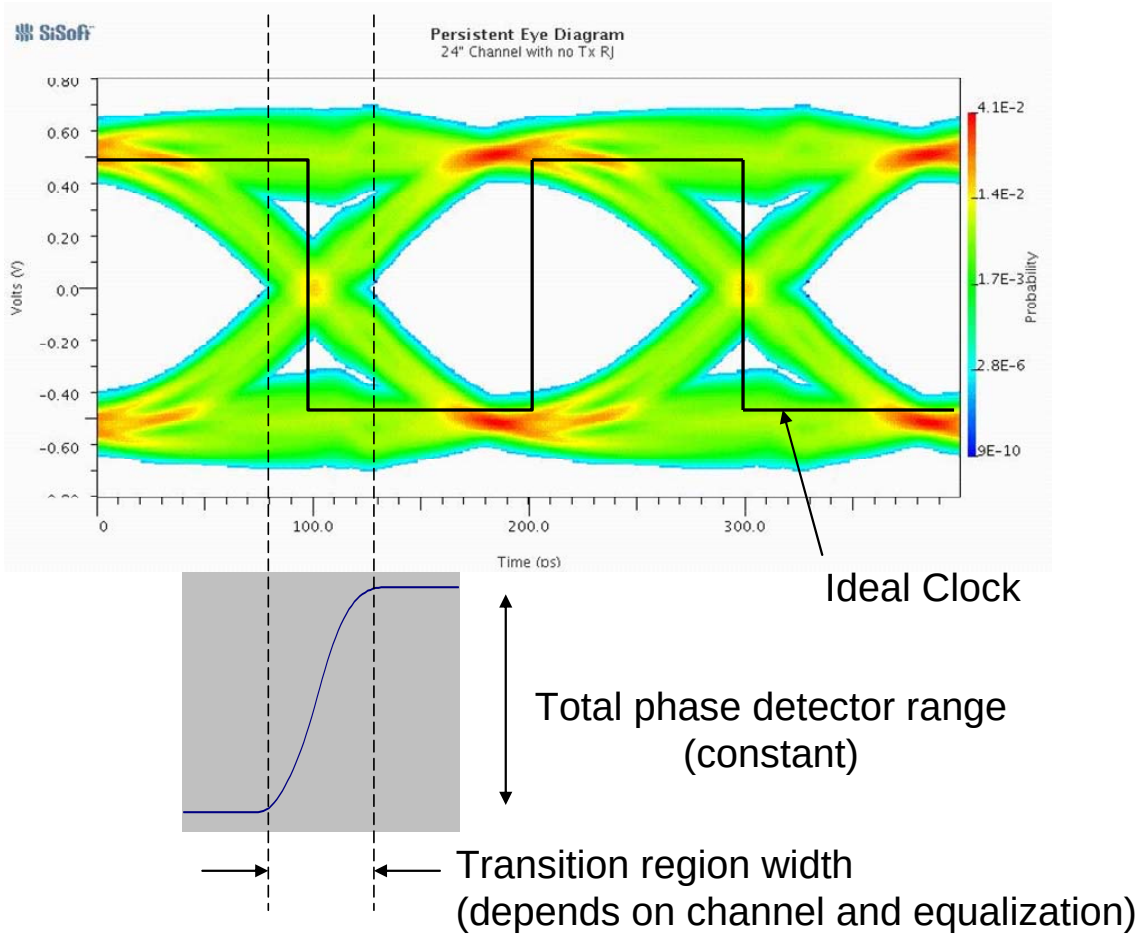


Figure 7: Derivation of phase detector sensitivity

Figure 7 shows the transition region of an eye diagram. If the clock timing minus one half bit time (e.g., the falling edge of the recovered clock) occurs before the transition region, then the phase detector will produce its most negative output, regardless how early the clock timing is. Similarly, if the falling edge of the recovered clock occurs after the transition region, then the phase detector will produce its most positive output. The clock recovery loop acts to place the phase detector output in the middle of its range, and therefore the falling edge of the recovered clock in the middle of the transition region. The sensitivity of the phase detector is the amount the phase detector output changes for a unit change in clock timing. Since the minimum and maximum phase detector output levels are fixed, the average sensitivity of the phase detector is inversely proportional to the width of the transition region.

Figure 8 is plot of the magnitude of the spectral density of the pattern dependent jitter as a function of frequency.

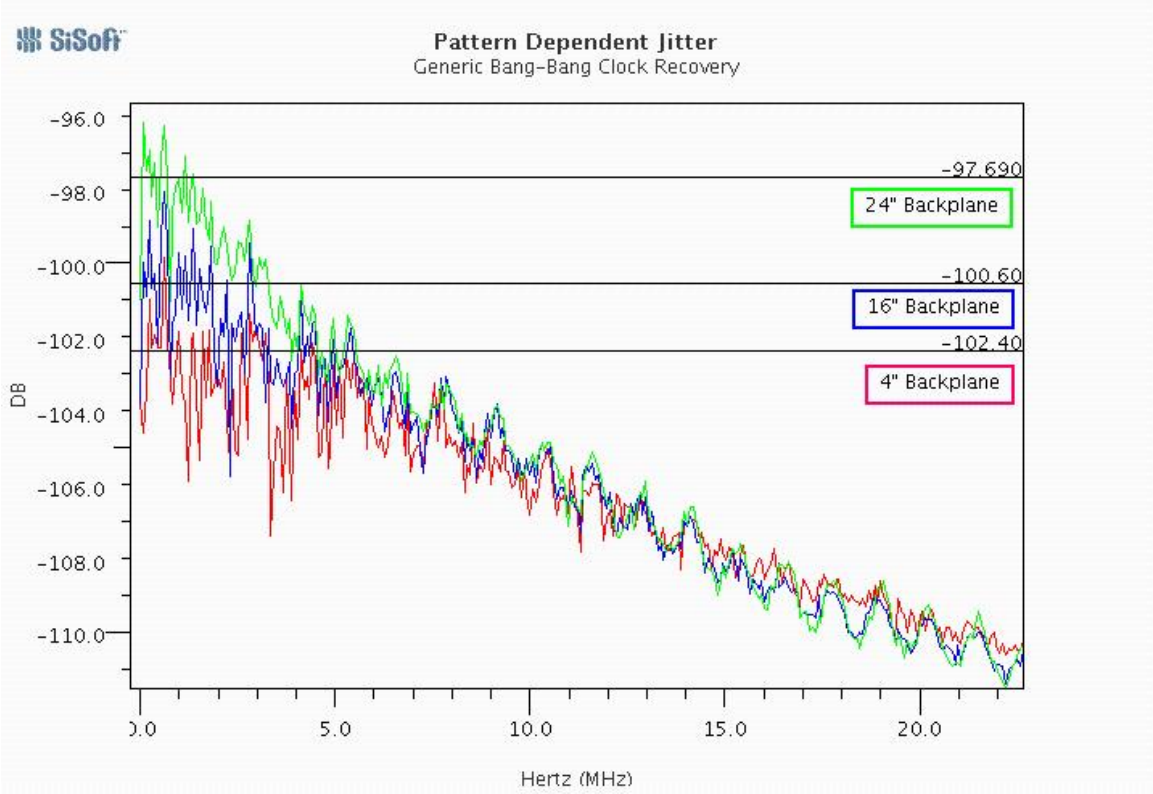


Figure 8: Pattern dependent jitter spectral density

Note that as the transmission path gets longer, the spectral density of the pattern dependent jitter at low frequencies increases as the loop bandwidth decreases. The net result is that the total pattern dependent jitter remains nearly constant. For the case studied, the pattern dependent jitter was 0.50 ps rms for the 4" path, 0.54 ps rms for the 16" path, and 0.69 ps rms for the 24" path.

We examined the correlation between deterministic jitter (DJ) and pattern dependent jitter and concluded that although there was a weak correlation, DJ does not seem to be a sensitive predictor of pattern dependent jitter.

Commercial IP

From the study of the generic bang-bang clock recovery loop, it appears that jitter transfer function bandwidth and total pattern dependent jitter are two useful metrics for a clock recovery loop. Jitter transfer function bandwidth is a measure of the clock recovery loop's ability to track the transmit clock and, all other things being equal, a large bandwidth is a good thing. Total pattern dependent jitter directly affects the serial link's performance, and should be as low as possible.

The analyses described above were applied to seven different IBIS-AMI models from six different IP vendors. Figure 9 is a plot of the clock recovery loop bandwidth vs. pattern dependent jitter for each of these designs and for all three transmission paths.

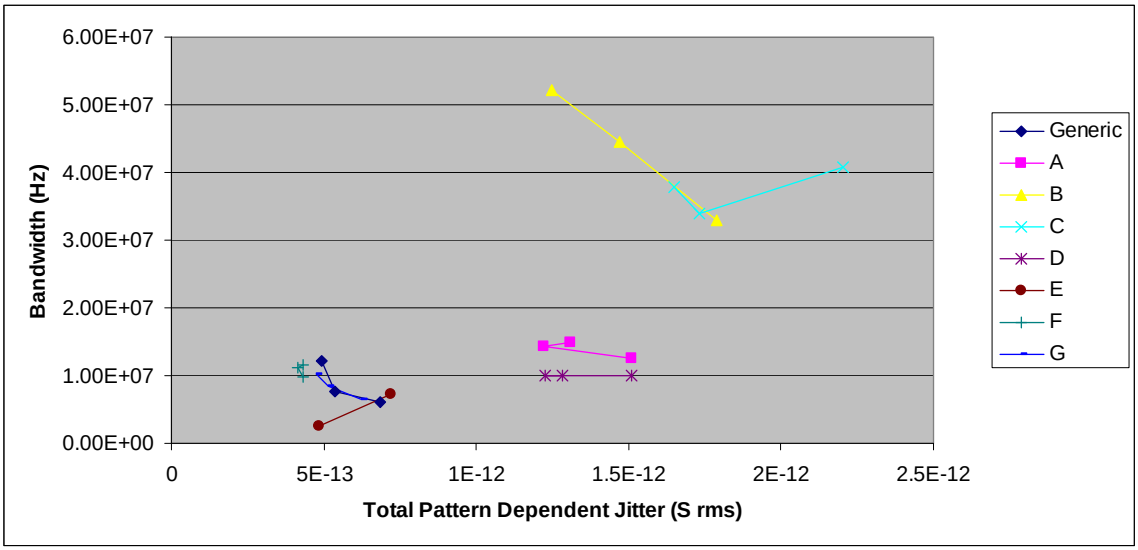


Figure 9: Jitter transfer function bandwidth vs. total pattern dependent jitter for seven different SerDes designs and three different transmission paths

Jitter Tolerance

We estimated the jitter tolerance of one of the commercial IBIS-AMI models and compared the results to measured jitter tolerance data for the same design. The results are shown in Figure 10.

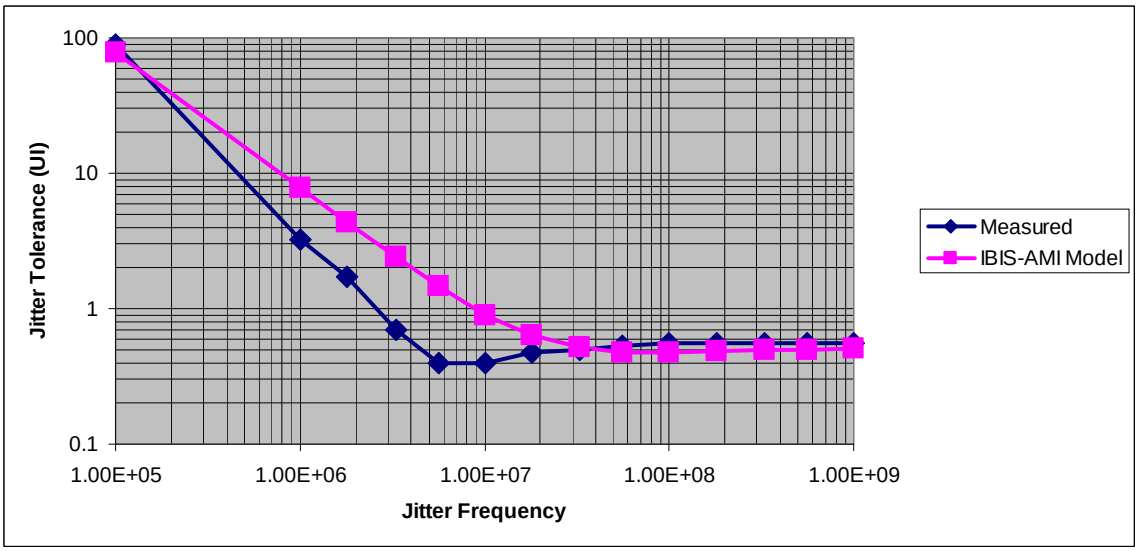


Figure 10: Comparison of estimated and measured jitter tolerance

Unfortunately, the document which reported the measured jitter tolerance did not state all of the details of the clock recovery configuration or the transmission path, so only a qualitative comparison can be made. In this case, it appears that the IBIS-AMI model over-estimates the clock recovery bandwidth by approximately a factor of four. Since the configuration used in the model was that for the highest loop bandwidth, and the model does support other clock recovery loop configurations, it's possible that the measurement was made using a different configuration than the model.

What this exercise does demonstrate is that the clock recovery loop behavior of an IBIS-AMI model can be compared directly to measured data, and that this comparison can be used to improve model correlation.

Conclusions

This paper has demonstrated the use of IBIS-AMI models to study clock recovery loop behavior and a practical way to correlate the observed behavior with measured data. An RJ stimulus was injected at the transmitter and the effect of this stimulus was measured in the clock_ticks array at the output of the receiver model. The results were used to estimate the jitter transfer function, pattern dependent jitter, and jitter tolerance.

Seven different SerDes designs from six different IP vendors were studied under realistic channel conditions.

The conclusions of this analysis are
1. Clock recovery loop performance varies significantly between SerDes designs.
2. Total pattern dependent jitter remains nearly constant as a function of transmission path.
3. The polar shape of the jitter transfer function remains constant as a function of transmission path even though the transfer function bandwidth varies significantly.

References

[1] Y. Chang, D. Oh, C. Madden, “Jitter Modeling in Statistical Link Simulation”, paper WED-AM-5-SS-1, International Symposium on Electromagnetic Modeling, Detroit, Michigan, August 18-22, 2008.
[2] B. Katz, M. L. Steinberger, T. Westerhoff, “Crosstalk Analysis of a System Based on XAUI HM-Zd Evaluation Backplane Data”, paper WED-AM-5-SS-3, International Symposium on Electromagnetic Modeling, Detroit, Michigan, August 18-22, 2008.
[3] *I/O Buffer Information Specification (IBIS)*, Version 5.0, copyright IBIS Open Forum 2008, August 29, 2008.
[4] M. Steinberger, T. Westerhoff, C. White, “Demonstration of SerDes Modeling using the Algorithmic Model Interface (AMI) Standard”, paper 7-TA3, DesignCon 2008, Santa Clara, California, February 4-7, 2008.
[5] http://en.wikipedia.org/wiki/LTI_system_theory#Exponentials_as_eigenfunctions.