PCI Express™ Architecture

PCI Express™ Jitter and BER
Revision 1.0

February 11, 2005

REVISION	REVISION HISTORY	DATE
0.5	First release – A. Martwick	1/4/2005
0.7	Revisions from first JWG review – A. Martwick	1/14/2005
0.99	Editorial Comments- A. Martwick	2/4/2005
1.0	Released – A. Martwick	2/11/2005

PCI-SIG disclaims all warranties and liability for the use of this document and the information contained herein and assumes no responsibility for any errors that may appear in this document, nor does PCI-SIG make a commitment to update the information contained herein.
Contact the PCI-SIG office to obtain the latest revision of this document.
Questions regarding this document or membership in PCI-SIG may be forwarded to:

Membership Services
www.pcisig.com
E-mail: administration@pcisig.com
Phone: 503-291-2569
Fax: 503-297-1090

Technical Support
techsupp@pcisig.com

DISCLAIMER

Contents

Figures

1 Introduction

1.1 About This Document

The goals of this document are to increase the knowledge base of the PCI SIG members regarding the budgeting of jitter and the budget relationship to the bit error rate (BER) for the PCI Express link. We present the models and definitions used for predicting the effect of different types of jitter in the PCI Express architecture and relate this to the BER.

Questions and comments regarding this document may be sent to the chairs of the PCI Express jitter working group:

jitter-group-chair@pcisig.com

1.2 Acknowledgements

This document has benefited from the many exciting discussions and presentations by the members of the PCI Express Jitter Work group.

1.3 Abstract

Our previous paper [1] showed that the PCI Express link has a phase relationship between the transmitter and receiver that is partially coherent. This work led to a re-budgeting of the PCI Express timings to include the contribution of the reference clock to the eye closure at the receiver. This new budget is now adopted in the PCI Express 1.1 specification [2] and in the CEM specification [3]. Future mechanical specifications must also include a budget for the reference clock effects.

In this paper, we extend the model to calculate the BER of the link based on the jitter content at the receiver of a PCI Express link. We explain the calculations used to perform the re-budgeting of PCI Express version 1.1 and discuss the issues of measuring clock jitter in the presence of noise. We then summarize the changes made to the base specification and the CEM specifications from the 1.0a to the 1.1 versions.

1.4 Limitations

This paper does not consider non-stationary sources of jitter. It is assumed that the mechanisms in the link that cause a particular jitter distribution do not change with time. This means that a system BER measurement is valid only for a particular instance of the system, and that the measurement is sufficiently long to capture all the events of interest.

This paper does not account for the data transition density effects of arbitrary data patterns. The data channel is modeled assuming a repeating 10101 pattern, equal to a clock. This simplification provides an unambiguous phase definition for both clocks and data.

We make no assumptions about the statistical nature of the data. In the bit error rate calculations we do not account for the fact that a misplaced sample might not create an error based on the statistical nature of the data itself.

1.5 Notation Convention

The upper case letters X, Y, N and H are used to denote the input signal, output signal, noise signal, and transfer function in the s domain. W is used as an intermediate term connecting X and Y. For our purposes the s domain is equivalent to the frequency domain of jw since the real part of s is always 0. In this notation multiplication operations are the equivalent of time domain convolution.

The lower case letters x, y, n and h are the time domain equivalent of the input signal, output signal, noise signal and impulse response. They are calculated from the frequency domain by the inverse Fourier transform.

The notation x_n, y_n, n_n and h_n are the discrete sampled time domain equivalent of the input signal, output signal, noise signal and impulse response. The sampling interval is not relevant to this paper and is taken to be 1.

2 PCI Express Jitter Architecture

2.1 Equivalent Jitter Model

The goal of the PCI Express jitter budget is to transfer a bit from the transmitter to the receiver with a high statistical certainty of success. The jitter budget is intended to ensure interoperability between different designs and that are connected through a wide variety of media. This requires that the data bit sent by the transmitter is sampled by the receiver near the center of the data bit. This sampling is illustrated in the figure1, where the sampling times are represented by an up arrow. These intervals are separated by some period, T. In PCI Express the ideal period T is 400 to 402 ps to an accuracy of +/- 0.03%.

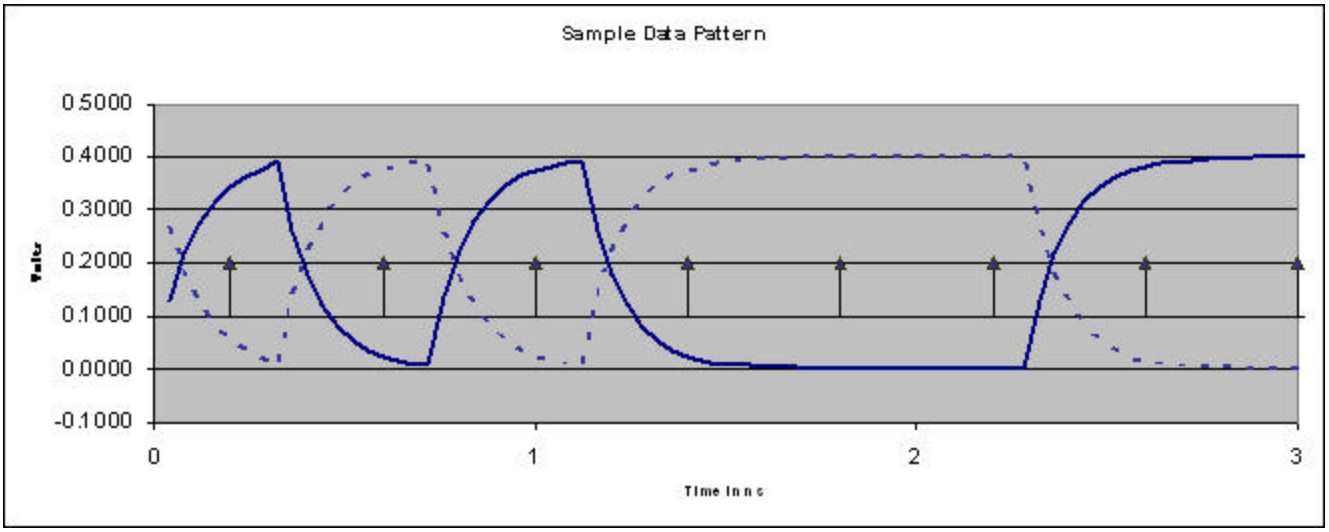

Figure 1 : Regular Sampling of Data

The PCI Express physical architecture that accomplishes the transmission and reception is shown in figure 2.

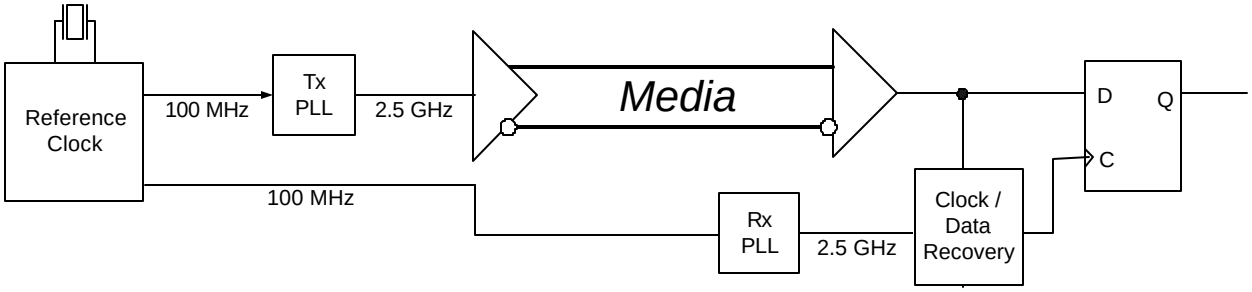

Figure 2: Physical Link Jitter Architecture

The Clock / Data Recovery block (CDR) generates a clock that samples the data into a latch at regular intervals perturbed by timing fluctuations called jitter. The particular CDR shown is a phase interpolator (PI) type which looks at the difference between the phase of the 100 MHz reference clock and the phase of the incoming data. It uses this information to adjust the location of the sampling clock with respect to the center of the bit.

Another type of CDR is the well known phase locked loop (PLL) type, not shown here. These CDRs are previously described in [1] where it is shown that the PLL type of CDR generally has better jitter tracking than the PI type of CDR. Often this performance advantage is at the expense of additional power and product cost.

PCI Express is a low cost link specified to work with the lowest performance CDR. The PCI Express specification assumes that the PI type CDR is used, and that a PLL type CDR will have no trouble meeting the same performance requirements.

The calculation of the displacement between the center of the data eye and the sampling clock is illustrated with a mathematical equivalent model of the link. This is shown in figure 3.

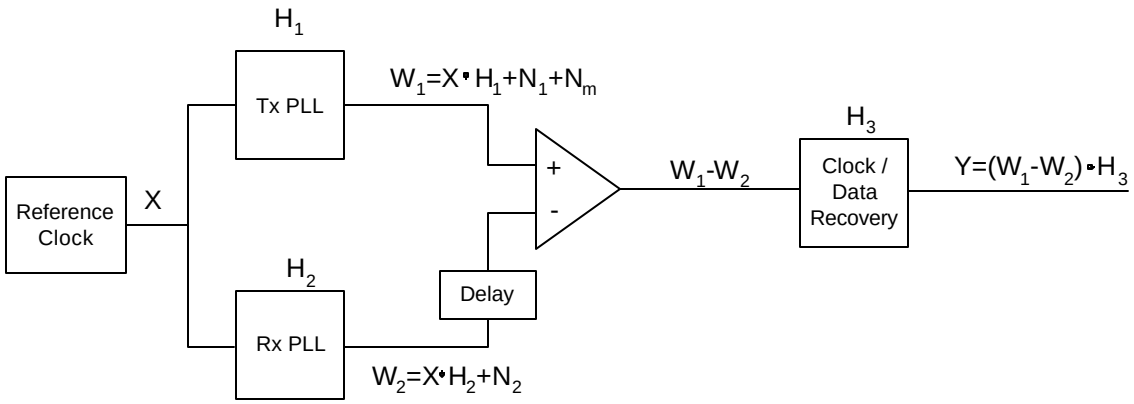

Figure 3: Equivalent Mathematical Model

In figure 3, the reference clock (100 MHz) phase jitter, X, is sent to both the transmitter and the receiver. Each device has it's own phase transfer function and injects some amount of phase jitter that is independent of the reference clock jitter. This is shown in figure 3 as N_1 and N_2. The media (traces on the PCB, cables, and connectors) adds some additional phase jitter due to the effects of dispersion, ISI, crosstalk and reflections. Finally, there is a delay between the two paths of the Tx and the reference clock that causes X to add to Y, due to the fixed phase offset. Since N2 is already uncorrelated this delay acts only on H2 and does not amplify N2. Moving forward, we will assume that the delay has been absorbed into the phase transfer function of H2.

As shown in figure 3 the phase mismatch at the receiver, Y, between the center of the data eye and the sampling clock is given by equation (1). In the time domain, y is the time difference between the edge of the data and the edge of the sampling clock and is the eye closure.

(1) $$Y = (W_1 - W_2) \cdot H_3$$

where

(2) $$W_1 = X \cdot H_1 + N_1 + N_m$$

and

(3) $$W_2 = X \cdot H_2 + N_2$$

Substituting equations (2) and (3) into (1) gives the complete system jitter equation (4).

(4) $$Y = X\,(H_1 - H_2)\,H_3 + N_1\,H3 + N_m\,H3 + N_2\,H_3$$

This can be written as

(5) $$Y = Y_{Clock} + Y_{Tx} + Y_{Media} + Y_{Rx},$$

using the definitions

(6) $$Y_{Clock} = X \cdot (H_1 - H_2) \cdot H_3$$

(7) $$Y_{Tx} = N_1 \cdot H_3$$

(8) $$Y_{Media} = N_m \cdot H_3$$

(9) $$Y_{Rx} = N_2 \cdot H_3$$

Equations 6-9 describe the jitter budgets on a PCI Express link.

The noise sources in equations 6-9 are caused by many different physical phenomenon such as thermal noise at the receiver amplifier, thermal noise and voltage noise in the VCO of the PLLs, voltage modulation of a device's power supply, cross talk, and inter symbol interference, to name just a few. This noise combines with the signal to create phase jitter at Y. All of the sources that contribute to the phase jitter are encompassed in the link budget.

2.2 PCI Express Bit Error Rate

PCI Express requires a model for the calculation of the bit error rate (BER) of the link that is applicable to either a PLL or PI based CDR. We define a bit error as the accumulation of phase jitter, y, such that the total phase difference between the data and the sampling clock exceeds ½ the unit interval (UI). This definition is independent of the data content, meaning we provide no statistical relief of the BER based on the random nature of the data.

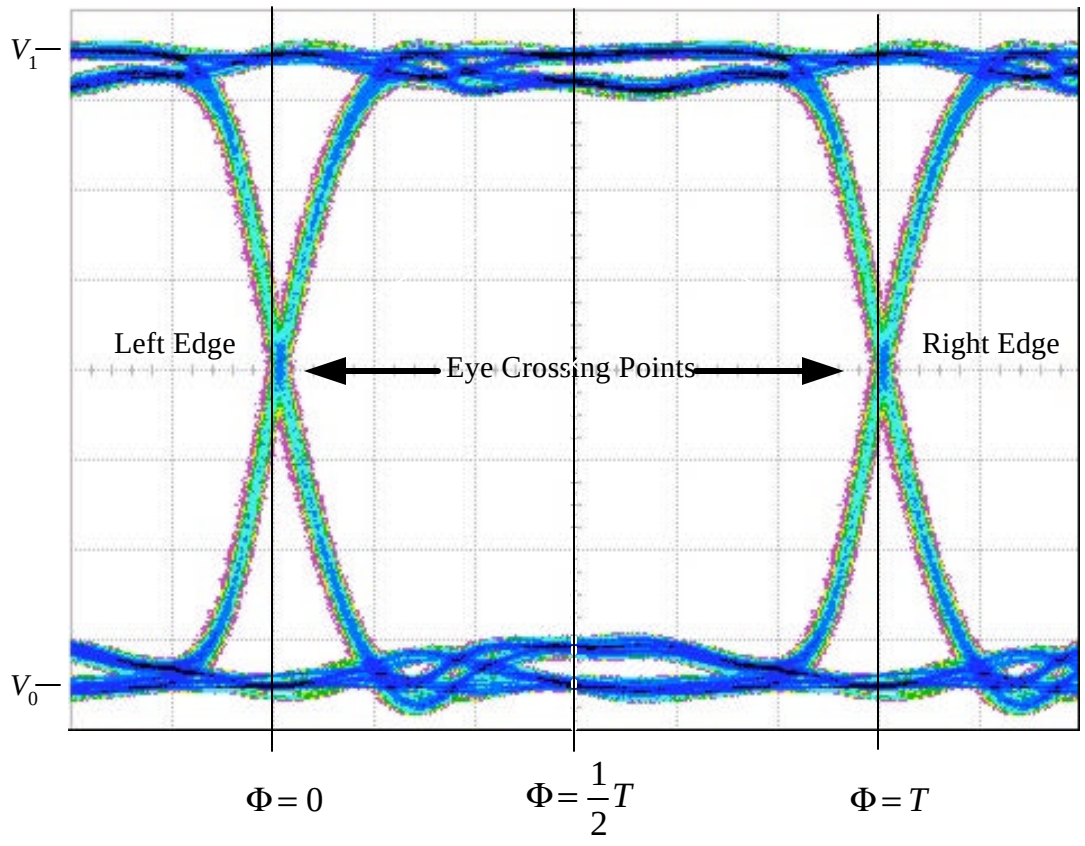

Figure 4: Ideal Sample Location is at the Bit Center

As the sample location deviates from the ideal sample location, the bit error rate increases. This can be seen qualitatively in the Eye Diagram of Figure 4. The edge crossing points are described by a statistical distribution about the points $\Phi = 0$ and $\Phi = T$. The ideal sampling point is at $\Phi = \frac{1}{2}T$. The sampling point should be placed at a region where the statistical distribution has a probability smaller than 10^{-12} as defined in the *PCI Express Base Specification* [2].

2.3 H_3 and Clock Data Recovery

The transfer function of the CDR is given as H_3 in equation (4). This frequency response is crucial to the operation and interoperability of the PCI Express link. In this model, if the magnitude of the incoming jitter, (W1-W2), is within the stop-band of H3, it will be proportionally tracked by H3 (and not passed into the system). Conversely, if the frequency of the incoming jitter (W1-W2) is in the pass-band of H3 then it does not get tracked as well and transfers into eye closure.

H_3 describes a general frequency response of many different CDR implementations. It should be noted that some CDR implementations, such as a PI type, may not posses a linear system response at all. However, such systems can be approximated as having a linear jitter tracking response for reasonable values of X.

In order to ensure interoperability, a minimum H3 response must be chosen. In the version 1.0a specification this response was given by the 250/3500 UI measurement algorithm [4]. The original 250/3500 UI mechanism has a frequency response magnitude that was investigated in [1] and is repeated here in closed form, equation (10).

$$H(f) = Max\left[2 \cdot \sin\left(\frac{n \cdot p \cdot f}{1.25e9}\right) - \frac{2 \cdot n}{3500} \cdot \sin\left(\frac{1750 \cdot p \cdot f}{1.25e9}\right)\right], \quad n = \{1,2,3,\ldots,125\} \tag{10}$$

The plot of equation (10) is shown in figure (5).

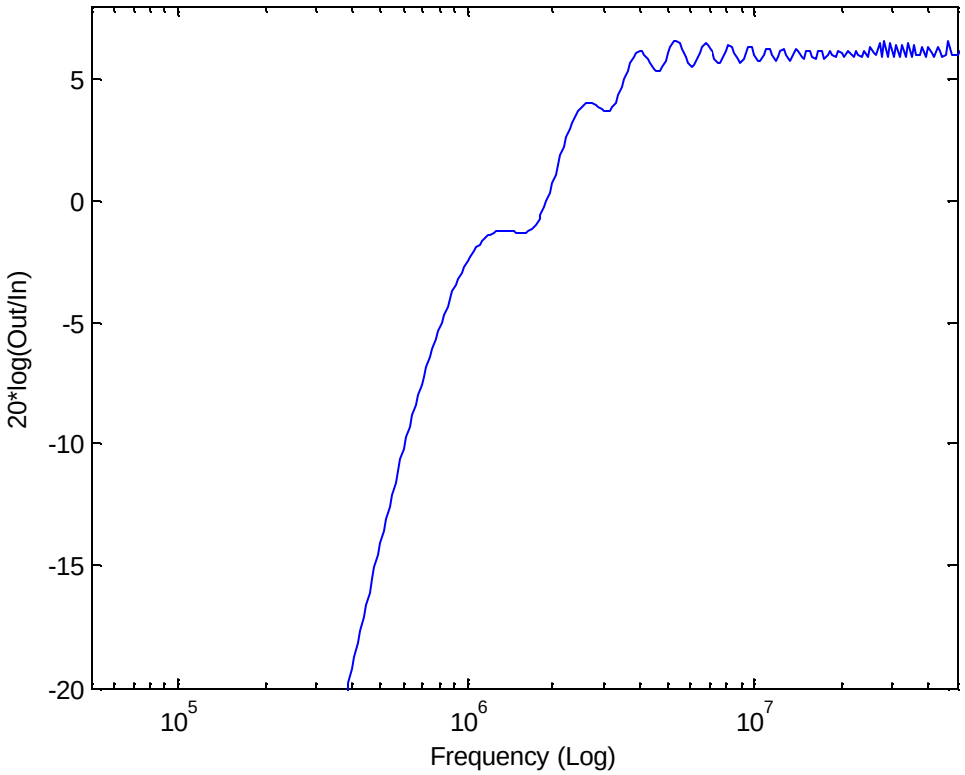

Figure 5: Frequency Respo nse of the 250/3500 Window

The figure (6) shows this frequency response overlaid with the H3 chosen in [1] as a representative CDR function that encompasses both PI and PLL type:

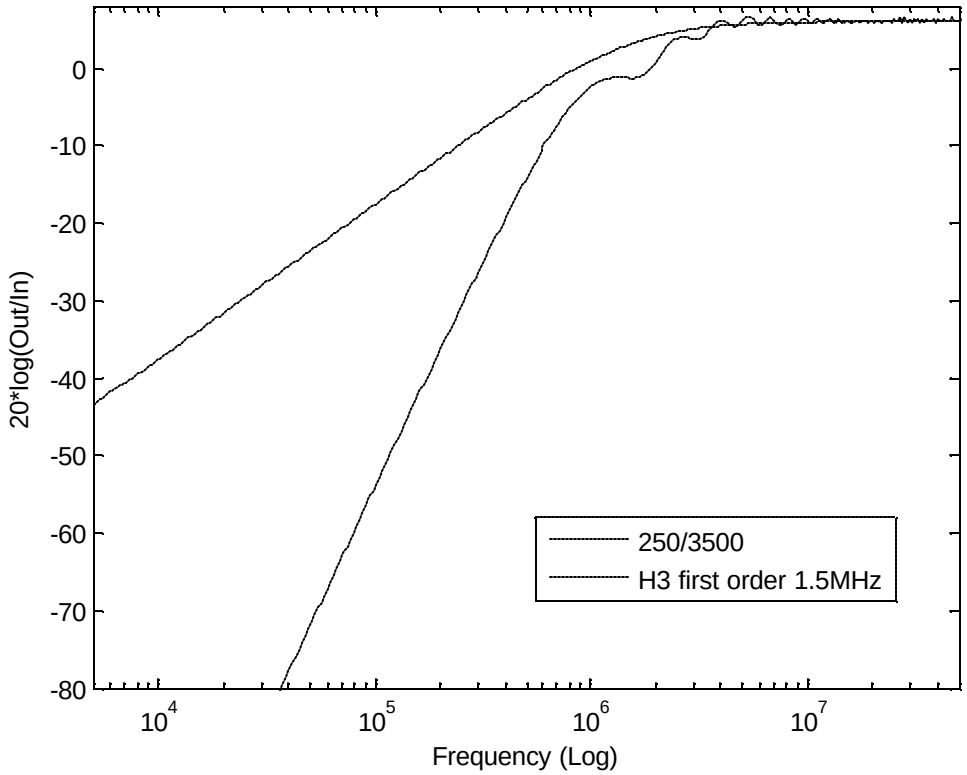

Figure 6: 250/3500 compared to 2X 1st order high pass of 1.5 MHz

As seen in figure 6, this high pass function provides an attenuation of -60dB / decade. This type of attenuation is necessary when measuring a spread spectrum clock (SSC), otherwise the SSC component (at approximately 30 KHz) contributes to the eye closure. This algorithm is described in detail in [4].

The change to the base specification accounts for the reference clock phase jitter independent of transmitter phase jitter. This is necessary to ensure interoperability of components and reference clocks. Measuring the transmitter phase jitter independently means the measurement is done with a clean clock, and there is no spread spectrum. In this case, the measurement establishes the minimum required response of H3. This provides a solid design requirement for CDR designers.

The H3 chosen in [2] is 2x a first order high pass function with a -3dB frequency at 1.5 MHz. This is the clock recovery function used to calculate the Tx jitter contribution to y. It is also used to as part of the specification for the reference clock jitter contribution in the PCI CEM Specification version 1.1.

It should be noted that a 525 / infinity algorithm produces a response that is very similar to the 1.5 MHz first order function and is shown in figure 7. The 525 / infinity window will always give a response that is equal or greater than the 1.5 MHz high pass.

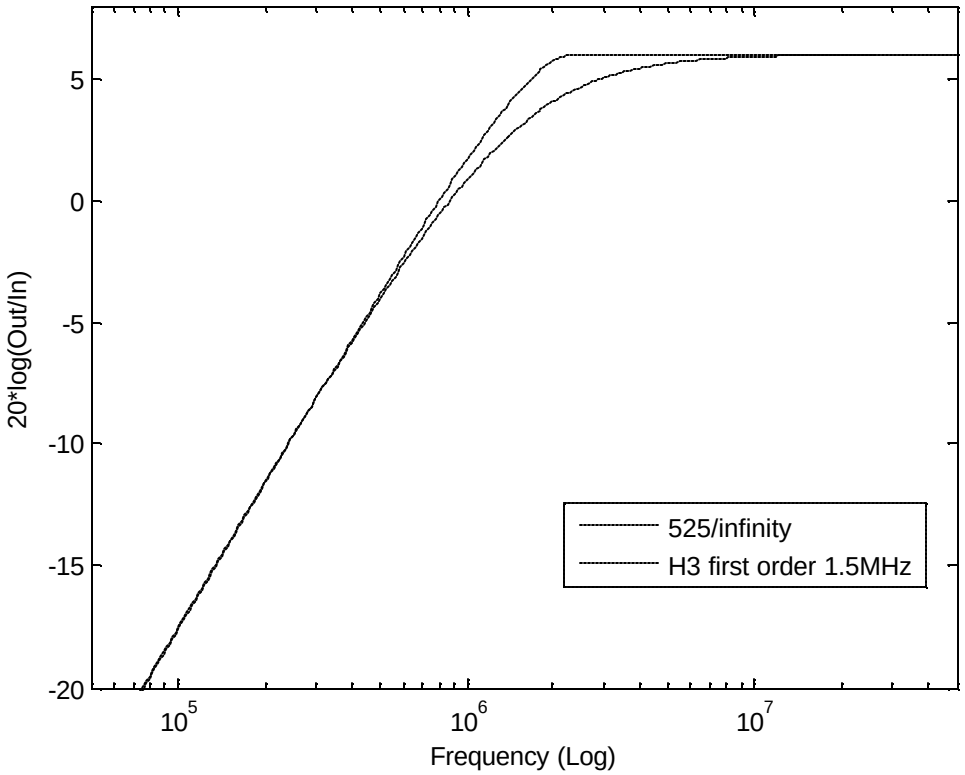

Figure 7: 525 / Infinity compared to 2x first order high pass at 1.5 MHz

2.4 Phase Jitter, Probability Density Functions, BER

The phase jitter is defined in [1] and repeated here as the difference between the measured time and the ideal bit period T. Phase Jitter is an accumulation of the time error from the ideal time of n*T.

(11) $$\Phi_n = t_n - nT, \quad n = 1, 2, \ldots, \infty$$

The ideal period, T, is calculated or recovered from the data itself. It can be recovered from the data using a variety of mathematical methods in either the time domain or frequency domain.

To define the Probability Density Function (PDF) for the phase jitter, it is useful to note that the phase jitter term, Φ_n, defines the time relative to the most recent ideal transition time. The offset of Φ_n is static, so the origin of the phase is arbitrary. This is seen in figure 4, where we choose $\Phi_n = 0$ as the ideal transition time and the PDF is given by the distribution of relative transition times, Φ_n. The fixed offset of ½ T into the center of the bit is handled internally by the receiver.

The distribution of the transition times gives the PDF as a function of the phase offset from the ideal.

Given a sufficient sample size, the PDF can be approximated by the normalized jitter distribution (i.e., the normalized time histogram of data transition times) of Φ_n.

In order to calculate the total system BER, we replace the sources of jitter in the right hand side of equation (4) with the probability density functions. Upon doing so, the addition operator is replaced with convolution and the result provides the full system jitter PDF. We can then evaluate this PDF based on our bit error rate definition, where a bit error is the probability that the sample location exceeds +/- ½ the UI.

Numerically it is simple to convolve arbitrary PDFs. It is more difficult analytically except for the special case of a Gaussian PDF distribution. In this case, the PDFs convolve as the root sum square (RSS) of the 1 sigma values. The mean value of the system PDF is always 0 due to the high pass nature of H3 and H1-H2. Since these are a high pass functions, no DC value remains in the distribution and the mean is 0 for all PDFs.

Thus is advantageous to segment the distribution into a Gaussian portion and a non-Gaussian portion. We will call the Gaussian portion that can be convolved as RSS Rj, and we will call the non-Gaussian portion Dj.

By definition, Dj is bounded and its PDF can take any form, depending on the sources that cause the D_j. A simple mathematical function we choose to adopt to quantify D_j is the so-called double delta function as shown in the following:

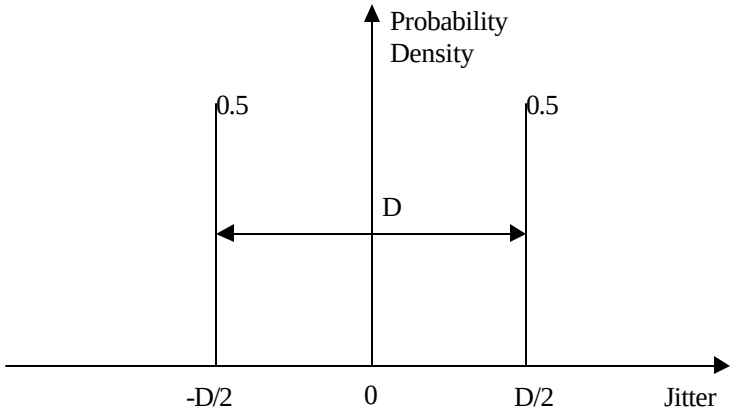

Figure 8: Illustration of DJ double delta function model

Mathematically, the D_j PDF function as shown in Figure (9) is given by:

(12) $$f_{DJ}(t) = \frac{1}{2}[d(t - \frac{D}{2}) + d(t + \frac{D}{2})]$$

Delta function makes the convolution easy to carry out since any function convolves with a delta function equals that function shifted to the location of the delta function. Mathematically, it means that the Dj peak-peak value adds linearly.

For our purposes of budgeting we will represent the Dj by the double delta model. It should also be noted that if the jitter sources are not independent they do not convolve as the RSS. For our purposes, independence of the Rj sources is assumed.

2.5 Jitter PDF Models

Figure 10 shows an example PDF consisting of Rj = 2.8 ps (at 1 sigma) numerically convolved with a Dj of 60.6 ps (+/- 30.3 ps). This is representative of the Tx requirement, as we will discuss later. The sample size that generates the Rj distribution is a normal distribution of 10^6 samples binned into a histogram of 400 bins (1 ps resolution). The Dj portion is modeled as a double delta function. The double delta distribution is shown in the first plot. The Rj PDF is in the middle plot, and the resulting convolution is shown in the third plot. The horizontal axis is in the units of ps and the vertical axis is the amplitude of the probability density function. All PDFs are normalized to have an area of 1.

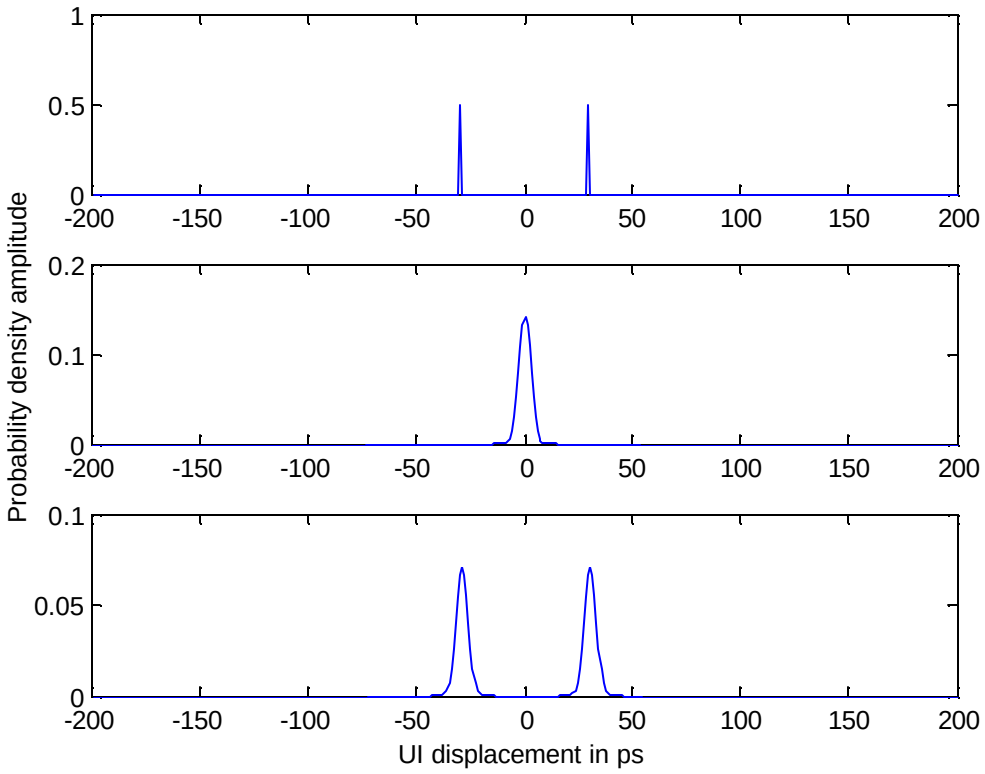

Figure 9: Tx PDF as a convolution of Dj and Rj

Figure 11 shows the budget for the reference clock. It has a Dj of 42 ps and an Rj of 4.7 ps.

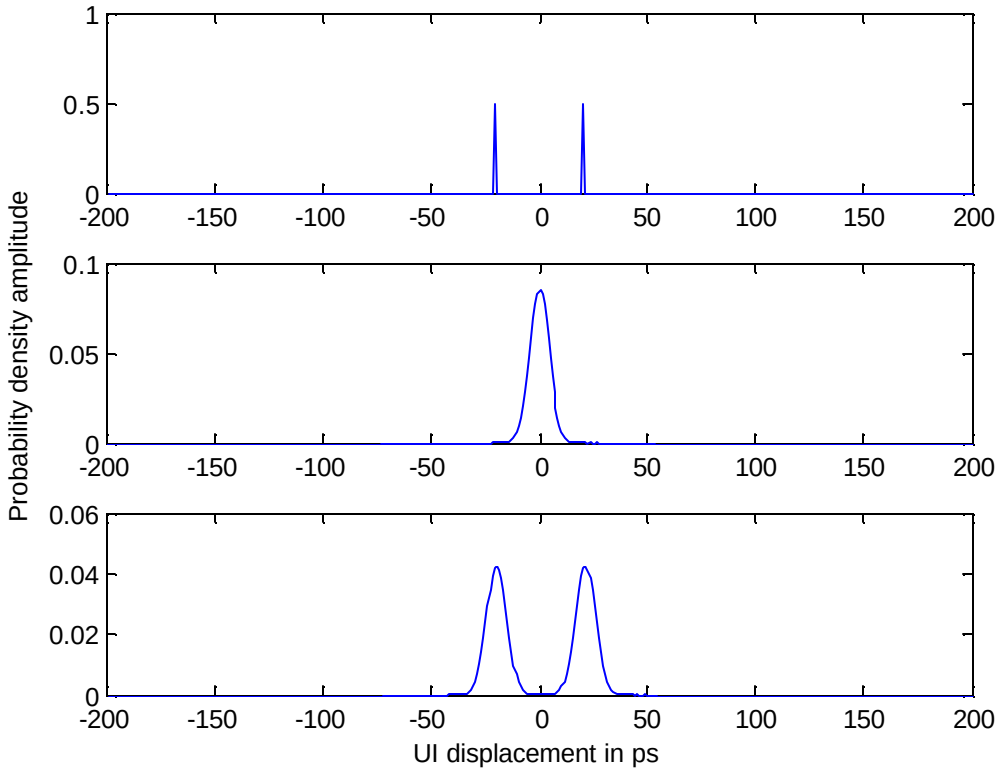

Figure 10: Reference Clock PDF as a convolution of Dj and Rj

Figure 12 shows the budget for the Tx, media and reference clock. The Dj is 192.5 ps and the Rj (one sigma) is 5.47 ps.

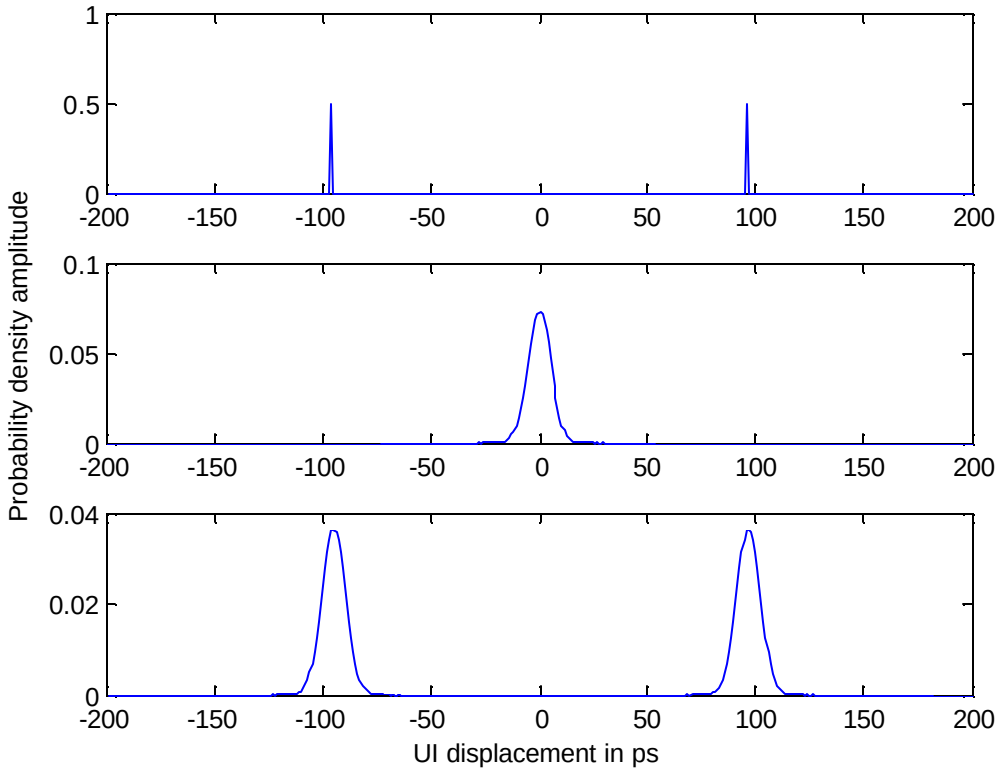

Figure 11: Everything but the Rx as a convolution of Dj and Rj

Figure 13 is a Dj of 313.1 ps and an Rj of 6.14 ps. This will be shown to be the full system budget. In this case with 10^6 samples, the area beyond the edges of the graph is 0 meaning none of these samples exceeded the +/- 200 ps limit. If it were computationally possible, using a jitter distribution of 10^{12} samples would produce an area beyond the edges of the graph less than 10^{-12}.

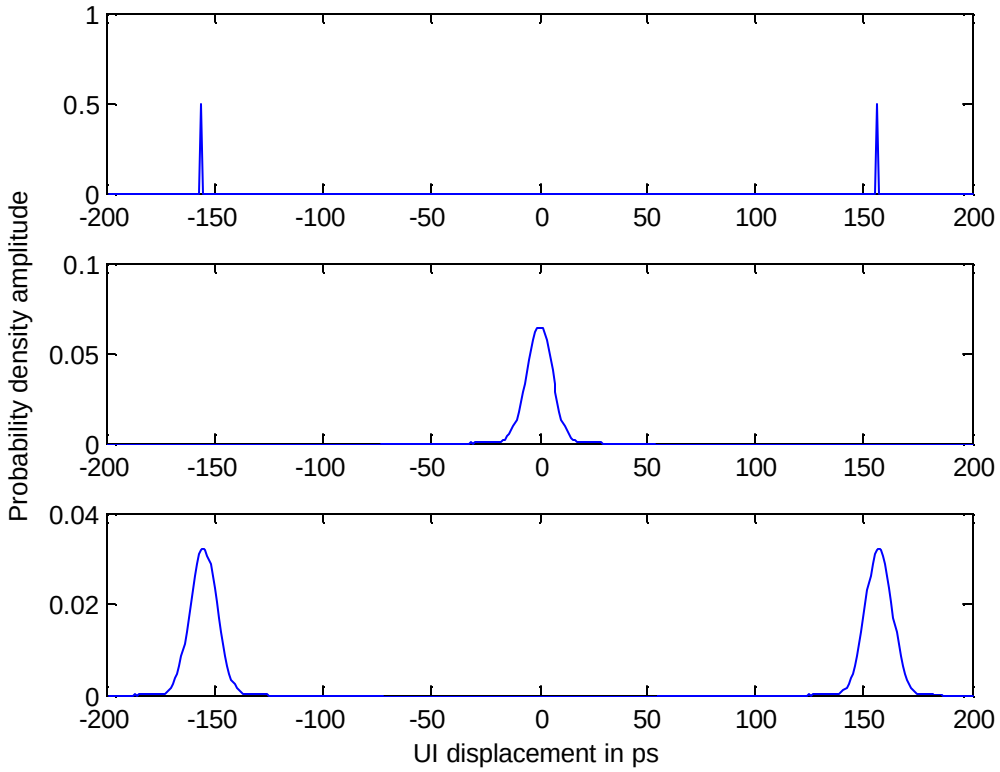

Figure 12: Total System PDF as a convolution of Dj and Rj

3 Jitter Measurement

The four sources of jitter are specified in equation (4) and are measured independently. This provides for interoperability of the link in terms of the jitter budget.

The first term, Y_{Clock}, is the 100 MHz reference clock phase jitter and should be measured at the connector where the reference clock is provided. The measurement consists of taking a sufficiently long record of the phase, X, of the 100 MHz clock and applying the transfer function $(H_1 - H_2) \cdot H_3$ to it. This can be done in the time domain as digital signal processing (DSP) filter function or in the frequency domain as provided in the MATLAB® code given in [1].

The second term is Y_{Tx}, or the transmitter phase jitter. This is measured at the connector where the transmitted data leaves the device. A sufficiently long record of the phase jitter output is taken and the transfer function H3 is applied. This measurement is intended to capture only N1 and no portion of X. This can be done by either using a high quality reference clock generator, effectively making X=0, or by adjusting for the presence of X in a two port measurement as described later.

The third term, Y_{Media}, is the budget allowed for the media. This measurement would consist of measuring the pulse response of the media channel, from which the worst case ISI can be calculated. The PDF of the media is assumed to be completely Dj.

The final term, Y_{Rx}, is the phase jitter inherent in the receiver. Since this is internal to the receiver, it must be measured through internal receiver mode test hooks.

3.1 Jitter Budgeting

A budget is assigned to each of equations (6-9) as shown in table 1.

The PDF of each component is split into a Dj and Rj portion. For the purposes of budgeting, the Rj portion are considered Gaussian and are convolved using the RSS. The Dj portions are considered to have a double delta distribution and are added linearly.

In order to derive the budget a minimum Rj term is assigned to Y_{Tx}, Y_{Rx} and Y_{Clock}. This minimum Rj term is based on measurements.

The one sigma Rj values are then extrapolated to the 10^{-12} BER. The remaining budget is Dj and is assumed to be quasi-stationary at all BERs. This value is then extrapolated down to the 10^{-6} BER. These values are used in the compliance measurement using population size of 10^6 edge crossings.

Jitter Contribution	Equation	Min Rj (ps) one sigma	Max Dj (ps) P-P	Tj at BER 10^{-12} (ps)	Tj at BER 10^{-6} (ps)
Tx	Y_{Tx}	2.8	60.6	100	87
Ref Clock	Y_{Clock}	4.7	41.9	108	86
Media	Y_{Media}	0	90	90	90
Rx	Y_{Rx}	2.8	120.6	160	147
Linear Total Tj:				458	410
Root Sum Square (RSS) Total Tj:				399.13	371.52

Table 1: Jitter Budget Calculations

Notes:

1. RSS equation for BER 10^{-12} Tj = $\sum Dj_n + 14.069 * \sqrt{\sum Rj_n^2}$
2. RSS equation for BER 10^{-6} Tj = $\sum Dj_n + 9.507 * \sqrt{\sum Rj_n^2}$

3.2 Receiver Tolerance Testing

PCI Express specifications do not provide a receiver tolerance test. This is left to the device designer and system manufacturer to ensure that their designs meet the 10^{-12} BER when used with a transmitter and media that meets the respective specifications.

As previously shown, the jitter budget assumes that the receiver has an internal jitter component in the sampling operation with a randomness equal to 2.8 ps at the first sigma, and valid out to the 7^{th} sigma. In order to test a particular receiver for correct operation the device and system designer would create a data pattern with a known jitter PDF. The BER of the link would then be monitored through internal test hooks to determine when an error occurs. The BER is then simply given by the number of errors divided by the number of bits, the sample size being sufficiently large enough to ensure confidence in the measurement. This confidence interval is typically achieved having a sample size of 5-10 times the reciprocal of the BER desired. Numerous papers and texts on statistics describe the math associated with this type of measurement.

The jitter PDF shown previously in figure 12 is not intended as a receiver tolerance mask. It is only used as a mathematical tool to bound the worst case jitter on the link. The problem with using this as a tolerance mask lies in the double delta function. If a system truly had this type of bimodal jitter components, under the appropriate data pattern it would likely fail other parts of the specification such as D+ to D- common mode requirement. In addition, a PI type CDR will not work properly with such a distribution since it cannot find the center of the distribution.

The characterization of the CDR requires jitter patterns that fall within the CDR range. The CDR should be tested to ensure it meets the minimum H3 requirement.

3.3 Measurement in the Presence of Reference Clock Jitter

It is not always possible to measure the System Board Transmitter Path Eye Diagrams with an ideal reference clock. In this case a two port measurement can adjust the measurement for the non-ideal reference clock. The notation in this section assumes the discrete time domain form of x_n. Equivalent notation sometimes used in literature is x(n) and x[n]. The convolution operator is denoted as $\otimes$.

Referring to figure 14,

1. x_n is the sampled phase jitter on the reference clock at the connector. A band-pass function is used to measure the reference clock as described in [3].

2. h_n is the impulse response of the transmitter PLL on the system board.

3. n_n is the intrinsic transmitter jitter. Using an ideal reference clock with 0 phase jitter, this would be the measured transmitter jitter. The peak-peak value of the total jitter must meet the eye requirements as specified in [2].

4. $y_n = [x_n \otimes h_n] + n_n$ is the total jitter of the reference clock, transmitter and system board interconnect at the connector, where $\otimes$ is the discrete convolution operator.

In the case of non-zero reference clock phase jitter, the intrinsic jitter of the transmitter can be calculated in the discrete time domain by the following:

(13) $$\mathbf{n_n = y_n - [x_n \otimes h_n]}$$

where x_n is convolved with h_n. The peak-peak value of n_n is then calculated.

Equation (13) has two unknowns, h_n and n_n. If h_n is known, it can be used directly. Otherwise, h_n must be assumed to be the lowest limit of the allowed PLL bandwidth with no peaking (see [2] for the specification limits).

A summary of this procedure is:

1. Take the simultaneous measurements of y_n and x_n for the system board at the connector.
2. Find $x_n \otimes h_n$.
3. Calculate $n_n = y_n - [x_n \otimes h_n]$
4. Calculate the peak-peak value of n_n.

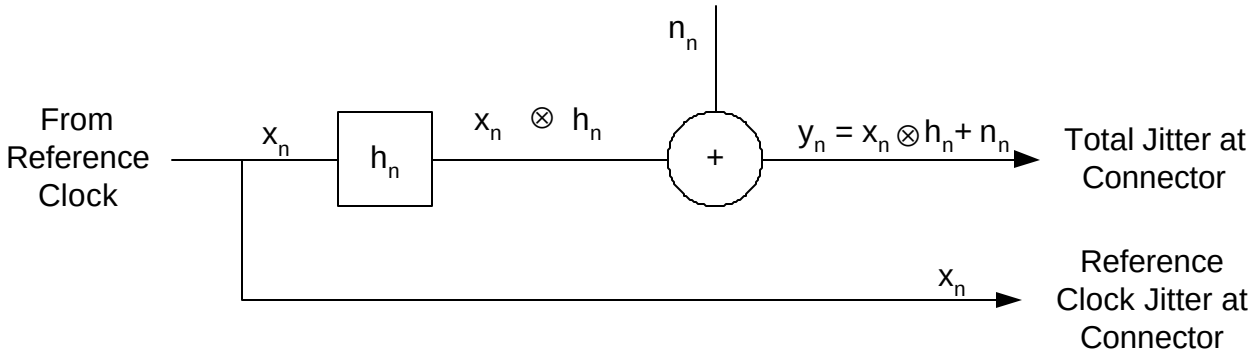

Figure 13: Two Port system measurement

3.4 PCI Express Base Specification Version 1.1 Summary

In summary, the changes to the base specification from version 1.0a to version 1.1 are as follows:

1. The maximum allowable ranges of the transfer functions H1 and H2 are specified.
2. The transmitter jitter specification, N1*H3 in the system equation, is reduced by 20 ps to 100 ps. N1 does not include the effects of X.
3. The Tx clock recovery function is given by H3 as a 1.5 MHz high pass

3.5 CEM Specification Version 1.1 Summary

In summary, the following has been added to the CEM specification [3].

1. The reference clock jitter is bounded after the system transfer function is applied
2. The Tx and media jitter numbers have been adjusted to include the reference clock
3. The delay between X, H1 and X, H2 is bounded to less than 10 ns
4. The CEM numbers are specified at a 10^{-6} BER
5. The “self jitter” terms, (N1+Nm+N2) * H3, are measured independently from X.

References

1. PCI Express Jitter Modeling, PCI SIG (www.pcisig.com), July 14th, 2004

2. PCI-SIG, *PCI Express Base Specification, Rev. 1.1*, (www.pcisig.com).

3.PCI-SIG, *PCI Express Card Electromechanical Specification, Rev. 1.0a*, (www.pcisig.com).

4. Per (Pelle) Fornberg, Dan Froelich, Andy Volk, *Transmitter Clock Recovery and Jitter Analysis Methodology For PCI Express Signaling*, Revision 0.8, Intel Corporation

Appendix A

A.1 Matlab code the PDF convolution

```
%**************************************************************************
%*
%*      Copyright (c) 2005 Intel Corp.
%*      Andy Martwick
%*
%*      This program has been developed by Intel Corporation.
%*
%*      Intel specifically disclaims all warranties, express or
%*      implied, and all liability, including consequential and other
%*      indirect damages, for the use of this code, including liability
%*      for infringement of any proprietary rights, and including the
%*      warranties of merchantability and fitness for a particular
%*      purpose.  Intel does not assume any responsibility for any
%*      errors which may appear in this code nor any responsibility to
%*      update it.
%**************************************************************************

close all;

Rj=4.7
Dj=42;

N=1e6;              % number of data points
R=Rj*randn(N,1);    % use matlab to create a Gaussian distribution
X=-200:1:200;
Rhist=hist(R,X)/N;  % get the histogram
D = zeros(400,1);
D(200-Dj/2,1) = 0.5;    % create the Dj PDF
D(200+Dj/2,1) = 0.5;    % the Dj amplitude is 1.
subplot(3,1,1);

plot((-199:200),D);
subplot(3,1,2);
plot((-200:200),Rhist);
ylabel('Probability density amplitude');

Tj=conv(D,Rhist);   % convolve the DJ and Rj
subplot(3,1,3);
plot((-199:200), Tj(200:599))
xlabel('UI displacement in ps');
%ylabel('Probability density amplitude');
```

A.2 Matlab for 250/3500 UI window

```
%*************************************************************************
%*
%*      Copyright (c) 2005 Intel Corp.
%*      Andy Martwick
%*
%*      This program has been developed by Intel Corporation.
%*
%*      Intel specifically disclaims all warranties, express or
%*      implied, and all liability, including consequential and other
%*      indirect damages, for the use of this code, including liability
%*      for infringement of any proprietary rights, and including the
%*      warranties of merchantability and fitness for a particular
%*      purpose.  Intel does not assume any responsibility for any
%*      errors which may appear in this code nor any responsibility to
%*      update it.
%*************************************************************************

close all;
step = .1:0.01:8;
f=10.^step;
filter = 3500;
window = 250;
k=zeros(length(step), window/2);

for n=1:window/2
   v=2*(sin(n*pi*f/1.25e9)-2*n/filter*sin(filter/2*pi* f/1.25e9));
   k(:,n)=v;
end

semilogx(f,20*log10(max(k')))
H3 = 2*(2*pi*i.*f) ./ ((2*pi*i.*f) + 2*pi*1.5e6);
hold on;
semilogx(f,20*log10(abs(H3)),'k');
axis([5e4 50e6 -20 8])
xlabel('Frequency (Log)');
ylabel('20*log(Out/In)')
```