Multi-Gigabit Serial Link Analysis – Piecing Together a Design and Verification Strategy

Michael Steinberger, SiSoft
msteinb@sisoft.com

Todd Westerhoff, SiSoft
twesterh@sisoft.com

SiSoft
6 Clock Tower Place, Suite 250
Maynard, MA
twesterh@sisoft.com

ABSTRACT

Designers embarking on projects with 3+ Gbps serial links are confronted with a confusing array of tools, techniques and terminology. Traditional time-domain simulation using HSPICE alone won't provide an adequate estimate of the link's Bit Error Rate (BER), which is the key metric used to assess a serial link's reliability. Designers must sort through analytical methods in the time domain, the frequency domain and statistical methods, and often assemble their own analytical solutions from combinations of existing EM tools, EDA simulation tools and mathematical modeling applications.

This paper introduces terminology associated with serial link design and different analysis methods (time domain, frequency domain, statistical). We discuss the role of 3D EM modeling and nonlinear circuit simulation tools (HSPICE) in solving the overall problem. We also present current standardization efforts for SerDes equalization and clock recovery modeling.

This paper is intended as an overview of serial link design and analysis for those designers new to the subject, who are looking to understand all the different potential pieces of the design puzzle and how they fit together.

Table of Contents

1.0 Introduction 3
2.0 Fundamental Problem Statement 3
3.0 Traditional Analytical Techniques 4
4.0 Solution Overview 5
4.1 Phase 1: System Physical Design 6
4.1.1 Electrical Path Metrics 6
4.1.2 Network Analysis Tools 7
4.2 Phase 2: IC Technology Selection 8
4.2.1 Equalizer Models 9
4.2.2 Phase 2 Bit Error Rate Estimators 9
4.3 Phase 3: Design Refinement 11
5.0 Bit Error Rate (BER) Analysis 12
6.0 Jitter-based Analysis 13
7.0 Communications Analysis 14
8.0 Equalization Techniques 15
8.1 Transmit De-emphasis 15
8.2 Receive Peaking Amplifier 17
8.3 Decision Feedback Equalization (DFE) 17
8.4 Feed Forward Equalization (FFE) 17
9.0 IBIS AMI Modeling 18
10.0 Convolution Engine 18
11.0 Peak Distortion Analysis 21
12.0 Conclusions and Recommendations 22

Table of Figures

Figure 1 – Channel impulse response compared to data pulse widths ……. 4
Figure 2 – speed serial link block diagram ……..... 5
Figure 3 – Bathtub curve for static timing analysis example …… 12
Figure 4 – Bathtub curve for high speed serial channel example …….. 13
Figure 5 – Bathtub curve for high speed serial channel with crosstalk …….. 15
Figure 6 – Transmit de-emphasis block diagram …… 16
Figure 7 – Sample transmit output waveform with de-emphasis …… 16
Figure 8 – Diagram of convolution engine algorithm ……. 19
Figure 9 – Statistical eyes for time domain simulation and convolution engine……...20
Figure 10 – Statistical eyes with and without crosstalk…… 20
Figure 11 – BER contour plots, including PDA contours …… 21

1.0 Introduction

Designers embarking on projects with 3+ Gbps serial links are confronted with a confusing array of tools, techniques and terminology. Traditional time-domain simulation using HSPICE alone won't provide an adequate estimate of the link's Bit Error Rate (BER), which is the key metric used to assess a serial link's reliability. Designers must sort through analytical methods in the time domain, the frequency domain and statistical methods, and often assemble their own analytical solutions from combinations of existing EM tools, EDA simulation tools and mathematical modeling applications.

This paper introduces terminology associated with serial link design and different analysis methods (time domain, frequency domain, statistical). Following a brief problem statement and comparison to parallel interface analysis, this paper presents a three phase design process, describing the methods and tools that are best suited to each phase of the process. Succeeding sections provide a more in-depth description for some of these methods and tools.

2.0 Fundamental Problem Statement

Higher data rate, same distance: While high speed serial links have been designed to transmit higher data rates than more traditional parallel interfaces, the real challenge is that the transmission media they are designed to use are only marginally better than those used by parallel interfaces, and the transmission distances have remained essentially unchanged.

Figure 1 illustrates this challenge by showing the impulse response for a typical high speed serial channel. The impulse response for a parallel channel will look very similar except that portion of the impulse response after the main impulse may not be as well controlled because larger discontinuities may have been allowed to remain in the path. Figure 1 also shows the width of a single data bit at 1 Gb/s and at 6.25 Gb/s. Note that whereas the impulse response will largely decay before the end of a data bit at 1 Gb/s, only the main impulse will be complete at the end of a data bit at 6.25 Gb/s.

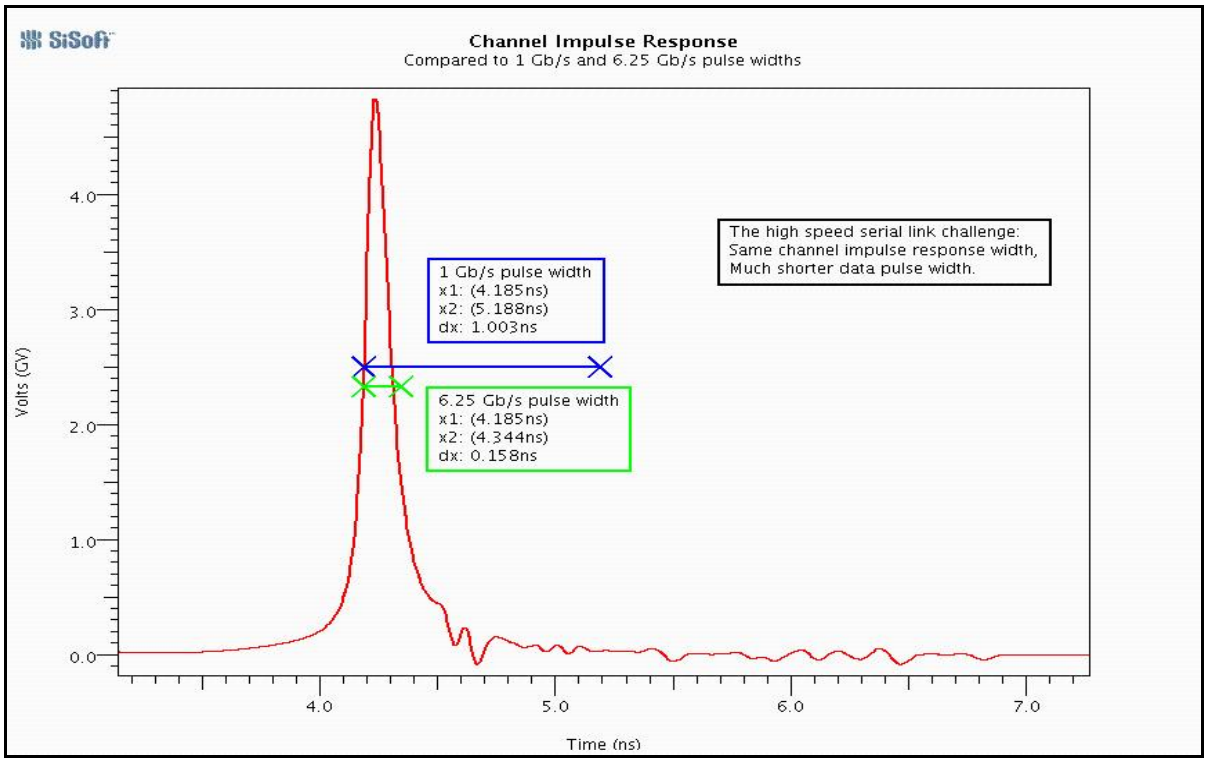

Figure 1: Channel impulse response compared to data pulse widths

There are three distinct facets to this challenge:

1. **Inter-symbol Interference:** The impulse response of the transmission path is approximately the same width as for a parallel interface, while the data bits are much shorter. Thus, many more data bits interact with each other, resulting in many more data patterns to analyze and requiring equalization to minimize the inter-symbol interference (ISI). Another way to look at the same phenomenon is that the frequency band of interest is much greater, and so the variation of loss and propagation delay across that frequency band is much greater.
2. **Clock to Data Timing:** Since the data bits are shorter, there is less timing margin available, and perturbations to timing are in general more critical.
3. **Crosstalk:** While the isolation of the transmission media can be improved through better design, crosstalk generally increases with frequency.

The net result is that bit errors are an almost unavoidable fact of life for high speed serial links, and bit error rate becomes the primary performance metric rather than timing margin.

3.0 Traditional Analytical Techniques

Parallel interfaces are normally simulated by modeling the combination of the driver, interconnect and receiver, applying a stimulus sequence, and simulating in the time domain. This methodology allows the driver and receiver to be modeled using either transistor level HSPICE models or IBIS analog behavioral models, thus capturing any non-linear or voltage dependent behavior in the active circuits. While this method permits a detailed analysis of I/O behavior, performance is limited. When simple transistor level models are used for the active devices, compute time considerations typically limit simulations to a few thousand bits. When

more complex models are used that model transmitter and receiver equalization, performance drops to where simulations of only hundreds of bits (at best) are practical.

Transistor level analysis of serial links thus represents something of a paradox. Detailed models of the transmitter and receiver offer the best possible representation of active device behavior, but the limited run lengths supported mean that transistor level analysis offers poor characterization of link-level ISI effects.

4.0 Solution Overview

The high speed serial link block diagram in Figure 2 provides a context for the overall solution. After parallel to serial conversion, the transmit data may be processed by an optional transmit equalizer before being driven onto the passive electrical interconnect. At the output of the electrical interconnect, the received signal may be amplified by a receive amplifier, and may be further equalized by a receive equalizer prior to clock recovery and data recovery. Link performance is determined by the quality of the data signal at the input to data recovery and the accuracy of the clock to data timing at this point, which is often called the *data decision point*.

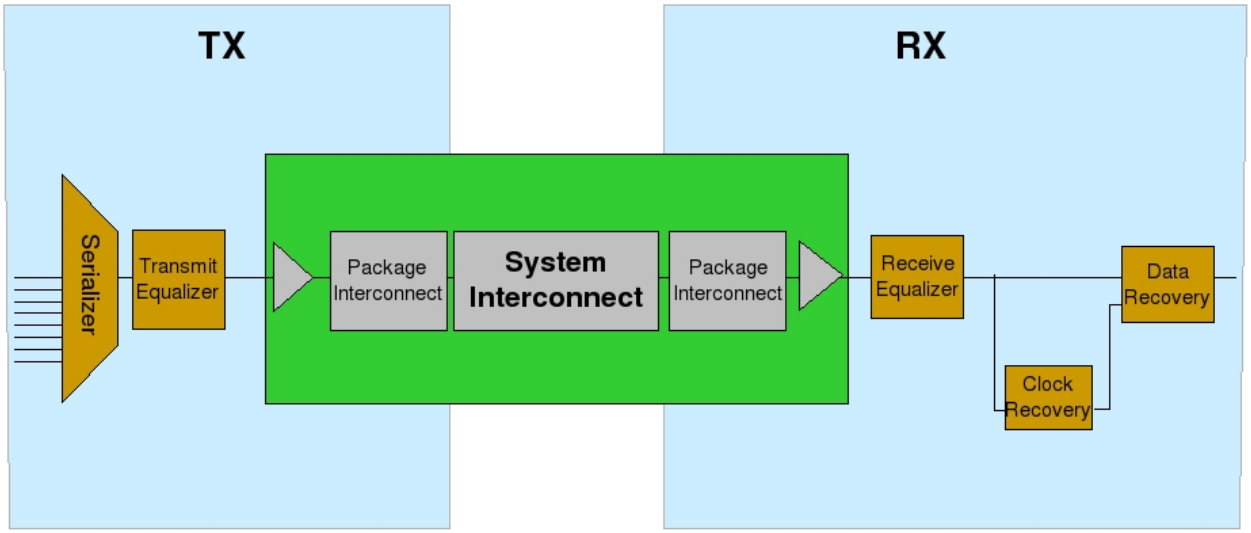

Figure 2: High speed serial link block diagram

The overall analysis solution for a high speed serial link should integrate solutions to all three facets of the problem. While such a solution must necessarily draw from a wide range of technical disciplines, and therefore will have a number of steps to it, there is a flow that can be followed. In general, this flow will have three phases.

Phase 1: System Physical Design. The overall system design is refined until it achieves acceptable cost and performance targets based on feasible data rates, interconnect lengths, routing densities, components and materials. Design decisions include

- Topology of the major system components such as racks, shelves, and PC boards
- Connector systems
- Transmission media, including cable media and PC board stackup, dielectric, trace lengths, spacing and cross section.

Phase 2: IC Technology Selection. The major ICs are chosen to support the system cost and performance targets. In the case of ASICs, the semiconductor vendor and SerDes macro designs are chosen. The system physical design may be refined based on the IC technologies chosen.

Phase 3: Implementation. Detailed design and analysis is performed to make sure that the system performs as expected.

In each phase of the design flow, all three facets of the serial channel design problem must be addressed. In each case, however, the tools and techniques should be chosen to suit the task at hand. There is no one set of tools or techniques that is a good general purpose choice.

The following subsections will recommend specific tools and techniques for each phase and facet, and describe the trade-offs that make them an appropriate choice for the task at hand. Later sections will described the less familiar techniques in more detail.

4.1 Phase 1: System Physical Design

4.1.1 Electrical Path Metrics

Especially during the preliminary stages of the system physical design, there are many options to be evaluated; so it doesn't make a great deal of sense to devote much time and effort to any one option. It is therefore appropriate to base design decisions on the results of network analysis and defer communications analysis to later phases of the design process. Several metrics have proven to be useful for preliminary evaluation of design options.

- **Loss Budget:** It turns out that most electrical transmission paths exhibit a loss characteristic in which the number of dB's of loss is approximately a linear function of frequency. This has resulted in a terminology whereby the loss of an electrical transmission path is quoted at a frequency equal to one half the target data rate. Thus, one refers to a “10 dB path” or a “25 dB path” without necessarily having to state the data rate. Furthermore, a given path could easily be a 10 dB path at 2 Gb/s and a 25 dB path at 5 Gb/s.

  One of the benefits of this terminology is that it normalizes out the data rate and is therefore a direct indication of how difficult the path will be to equalize. Up to a 15 dB path is fairly easy to equalize using any of the the known techniques. 20 dB is not too difficult using a combination of transmit and receive equalization. 25 dB is close to the limits of most reasonably available technology and 30 dB is generally considered to be the practical limit.

- **Crosstalk:** It is a good idea to keep an estimate of crosstalk from the beginning of the project. There's a simple metric if the design is expected to have a lot of crosstalk isolation and a more complex metric if the design might get close to some performance limits.

  The simple metric is that if *all the significant crosstalk will be **far end** crosstalk* and the crosstalk isolation is greater than 30 dB up to a frequency equal to the data rate, then crosstalk should not be a problem.

The more complex metric is a signal to crosstalk energy ratio at the input to the receiver.

$$R = \frac{\int |H(f)|^2 \frac{\sin^2(\pi f / f_d)}{(\pi f / f_d)^2} df}{\int |H_c(f)|^2 \frac{\sin^2(\pi f / f_a)}{(\pi f / f_a)^2} df}$$

Where f_d is the payload data rate, f_a is the crosstalk aggressor data rate, $H(f)$ is the transfer function of the desired channel, and $H_c(f)$ is the crosstalk transfer function. Any signal to crosstalk ratio less than 25 dB is cause for concern; while a signal to crosstalk ratio greater than 30 dB is safe.

When evaluating crosstalk, it is especially important to evaluate all cases of near end crosstalk because such cases often have a strong aggressor and a victim that has been weakened by transmission loss.

- **Receive Signal Level:** Even if the input to a receiver is perfectly equalized, a certain signal level is required to achieve a low bit error rate. This minimum receive level will vary widely from one SerDes design to another, and should be one of the parameters on the SerDes data sheet. For planning purposes, however, a received signal level of over 200mV rms will usually be acceptable while a received signal level of less than 100mV rms will not be sufficient for many SerDes designs.

- **Gain Ripple:** Excessive gain ripple is cause for concern, even for very low loss paths. Such gain ripple will occur whenever there are two large discontinuities in the path separated by more than a quarter of a data unit interval. The gain ripple should be limited to no more than perhaps 1dB peak to peak.

4.1.2 Network Analysis Tools

When analyzing an electrical path at microwave frequencies, it is a good strategy to first characterize individual circuit elements such as transmission lines, connectors, vias and packages in isolation and then combine those elements in an end to end path analysis. Care must be taken, however, to characterize the circuit elements using the same electromagnetic boundary conditions that they will encounter in the complete electrical path. For example, there is an interaction between the fields due to a pad and via and the fields due to the transmission line that connects to them. Depending on the relative sizes and orientations, this interaction could be significant.

There are a number of tools which can be helpful in predicting the performance of an electrical path.

- **Vector Network Analyzer:** Whenever possible, it's a good idea to measure the individual elements of the electrical path such as connectors, packages, vias, and transmission lines. A vector network analyzer (VNA) measures both the magnitude and phase of both the transmission and reflection coefficients of a network and typically outputs them in the form of S parameters. S parameters measured using a VNA can be used directly in subsequent analyses.

- **2D Electromagnetic Field Solver:** A 2D electromagnetic field solver solves for the electric and magnetic fields of a given cross section on the assumption that the fields do not vary in the third (z) dimension. A 2D field solver is a very effective way to estimate the characteristic impedance of different transmission line geometries and dimensions.

- **3D Electromagnetic Field Solver:** A 3D electromagnetic field solver solves for the electric and magnetic fields in all three dimensions. Furthermore, there are two types of 3D field solvers: static and full wave. A static field solver assumes that the time delay of a signal across the structure is negligible and therefore solves for the fields independent of frequency; while a full wave solver takes the time delay across the structure into account, and therefore generates results which vary with frequency. The rule of thumb is that if the structure is less than a tenth of a data bit long (taking dielectric constant into account), then a static solution is adequate whereas for structures more than a tenth of a data bit long, the full wave solution is recommended.

  A 3D field solver is useful for analyzing compact, complex structures such as connectors and vias. It is not suitable for very large structures because of the amount of computation required. Results are only reliable if the boundary conditions are adequate and the tool has been used correctly. Considerable skill is required to obtain meaningful results.

- **Microwave Circuit Solver:** Whereas an end to end electrical transmission path is too complex to analyze using a field solver (i.e., Maxwell's equations), there are microwave circuit solvers that use specialized solutions to Kirchoff's equations to solve an end to end path efficiently and accurately.

- **HSPICE:** HSPICE can be used to include the driver and receiver amplifiers with the electrical path. This is particularly useful for obtaining a step response or pulse response which includes any nonlinear effects due to the driver. S parameters can be included from the VNA, field solver or microwave circuit solver. Simulations of up to a few hundred bits are practical. Care must be taken to make sure that the electrical path model is causal, as failure to do so will prevent convergence.

4.2 Phase 2: IC Technology Selection

The purpose of phase 2 is to design or choose pin electronics that will meet system performance requirements. In particular, the combination of the electrical path with the equalization solution in the SerDes must achieve a satisfactorily low bit error rate.

While there won't be as many options to evaluate as there were in Phase 1, there will still be numerous choices of equalization solution; and there will probably be several iterations on the electrical design before a satisfactory combination of electrical path and equalization solution is found.

The net result is that in Phase 2, it is very desirable to use a bit error rate estimation technique that is reasonably fast while still taking into account as many data patterns as possible. There are several techniques in use, and some of these are well suited to Phase 2.

Three things are required to estimate the bit error rate: a model of the electrical path, a model of the equalization solution, and a bit error rate estimation method. The preliminary model of the electrical path was prepared in Phase 1 and can be refined as necessary using the same techniques. The next two subsections will review the options for modeling the equalization solution and estimating the bit error rate.

4.2.1 Equalizer Models

It will not usually be possible or practical to simulate the equalization solution at the transistor level. Also, SerDes IP vendors are reluctant to expose the details of their circuit designs, even for the purposes of performance analysis. Equalization is therefore usually simulated using algorithmic models that process waveforms instead of voltages and currents. There are several options.

- **Generic Equalizer Model:** Section 8 describes several well recognized equalization techniques, and generic models of these techniques can be used to evaluate in a general way what level of equalization is going to be possible. These models can either be implemented in a home-grown tool (e.g., MatLab, Python PyNum, PERL Data Language, C/C++, AMS) or provided by an EDA tool.

- **Vendor-supplied Environment:** Some SerDes IP vendors offer software tools for modeling the performance of their IP with user-defined electrical paths. While these tools have the advantages of being supplied with the IP and being backed by the IP vendor, they usually will only model links which use SerDes from that supplier.

- **IBIS AMI Algorithmic Model:** Section 9 describes an extension to the IBIS standard whereby IP vendors supply executable models of their IP which conform to a standard API. This has the advantage of allowing IP vendors to provide detailed models without exposing the details of their designs. This also allows models from different vendors to be combined in a single performance analysis environment such as an EDA tool.

4.2.2 Phase 2 Bit Error Rate Estimators

Bit error rate estimation methods will be listed in order of increasing detail. In this spectrum of options, a more or less arbitrary distinction will be made between Phase 2 methods and Phase 3 methods. The less detailed methods tend to take less computation, and are therefore better suited to Phase 2; while the more detailed methods tend to provide more information on the effect of design details, and are therefore better suited to Phase 3. It is perhaps unfortunate that many of the more familiar or intuitively appealing methods are also more detailed, and therefore come later in the list of methods.

For each method, the description includes a brief summary of how the three facets of the serial channel performance are reflected in the analysis.

- **Peak Distortion Analysis (PDA):** As described in section 11, peak distortion analysis determines the worst case data pattern with respect to a specified criterion, and then calculates the eye diagram that would result from that worst case pattern. If all the sampling clock edges will fall inside the worst case eye opening, then the link can be expected to operate error-free. This is useful for providing a mask compliance type of eye mask and for making sure that the

potential impact of the worst case condition is acceptable; however, when used to the exclusion of other methods, it will tend to produce overly conservative designs.

  - **Intersymbol Interference:** The inner edge of the worst case eye is determined by the maximum possible intersymbol interference.
  - **Clock to Data Timing:** The width of the worst case eye opening is an indication of the timing margin that is available.
  - **Crosstalk:** If the crosstalk were peak-limited, it would be possible to include its effects with the effects of intersymbol interference when computing the worst case eye; however, that would typically yield an excessively conservative result.

- **Convolution Engine:** Using the recursive convolution algorithm described in section 10, a convolution engine calculates the probability density function (PDF) of the eye diagram from the signal's step response or pulse response. This is often called a “statistical eye”. It can be used to very quickly evaluate the average effect of all possible messages of a given length. This is a very effective way to explore a wide range of design options while predicting the average bit error rate.

  - **Intersymbol Interference:** The effects of intersymbol interference are reflected directly in the PDF of the eye diagram. This is a very accurate and efficient way to calculate the intersymbol interference due to all possible combinations of bits, even for very long message lengths.
  - **Clock to Data Timing:** Using the communications analysis technique described in section 5, the PDF of the clock can be combined with the PDF of the eye diagram to produce a very rigorous BER estimate.
  - **Crosstalk:** The PDF of the crosstalk can be convolved with the PDF of the eye diagram to produce a composite PDF which accurately and completely reflects the effects of crosstalk. When combined with the communications analysis technique as mentioned above, the resulting BER estimate is both rigorous and complete.

- **Jitter-based Analysis:** Jitter-based BER estimation is intuitively appealing and a useful extension of static timing analysis. It is used widely, especially as a way to define a performance budget in some high speed serial interface standards. It is also the appropriate technique to use when multiple links are connected by saturated buffers. As discussed in section 6, however, jitter-based BER estimation is only accurate for a very limited set of conditions.

  - **Intersymbol Interference:** Intersymbol interference is generally considered to be the primary source of deterministic jitter (DJ). The tails of the jitter distribution due to intersymbol interference can tend to look like the tails of a gaussian distribution, howver, and there is some debate as to whether or not these tails should be called random jitter (RJ).
  - **Clock to Data Timing:** Especially under the conditions for which jitter-base BER estimation is accurate, the total jitter (TJ) is a direct measure of the variation of clock to data timing.
  - **Crosstalk:** Crosstalk is generally considered to be a source of RJ; however, this characterization is only accurate when the crosstalk is very small compared to the

desired signal. For large enough levels of crosstalk, jitter-based analysis breaks down altogether.

4.3 Phase 3: Design Refinement

In Phase 3, the primary activities are to refine the designs of the PC boards, integrate and test the system, resolve problems as they arise during system integration, and correlate with measured data. A fortunate few may add SerDes design and/or evaluation to this list.

Two BER estimation methods are best suited for this phase of the design process.

- **Semi-Analytical BER Estimation:** Semi-analytical BER estimation is a BER estimation method in which up to a few million bits of the desired signal are simulated in the time domain, and then the effect of the noise, crosstalk and clock phase noise is estimated using statistical techniques, as decribed in section 7. This has the advantage of allowing time variant and nonlinear effects to be studied, such as pattern dependence in the clock recovery and equalization control loops to be studied while maintaining the efficiency and dynamic range of statistical calculations. The ability to simulate specific data patterns also makes this approach useful for correlating with measured data or determining worst case performance.

    - **Intersymbol Interference:** The eye diagram accumulated over the course of the time domain simulation is either a random or deterministic sampling of the intersymbol interference. While this is sufficient if the intersymbol interference only occurs over short message lengths, it is difficult to obtain a representative sample for long message lengths.
    - **Clock to Data Timing:** The same conditional probability technique of section 5 that is used with the statistical eye from a convolution engine can be used with the eye diagram from a time domain simulation to produce a reasonably accurate BER estimate. This estimate will tend to vary by a factor of ten or more, however, unless the number of symbols in the simulation is quite large.
    - **Crosstalk:** Although crosstalk could be included explicitly in the time domain simulation, this would be used primarily to determine the effects of crosstalk on the clock recovery and equalization control loops. For BER estimation, it's much more effective to include the crosstalk in the statistical part of the analysis.

- **HSPICE:** HSPICE can be used to simulate data patterns up to perhaps a few thousand bits at the transistor level in a reasonable amount of time. This is useful primarily for preparing and validating algorithmic models, correlating with measured data, or investigating very detailed nonlinear effects. One could also use HSPICE as the time domain simulator inside a semi-analytical BER estimator; however, the accuracy of the BER estimates would be limited by the small number of symbols in the simulation.

    - **Intersymbol Interference:** A few thousand bits can provide only a limited sample of the intersymbol interference.
    - **Clock to Data Timing:** Clock phase noise could be included explicitly in the simulation; and this could provide useful information about the behavior of the clock recovery loop circuit design. However, the statistics would not be adequate for any meaningful BER estimate.

- **Crosstalk:** Similar statement as for clock to data timing.

5.0 Bit Error Rate (BER) Analysis

Bit error rate analysis can be viewed as a logical extension of static timing analysis. Figure 3 shows a plot of bit error rate vs. clock to data timing as it is used for static timing analysis. If the clock to data timing is in a certain range, the bit error rate is zero, and outside that range the bit error rate is a half. This is an idealized version of what is commonly called a “bathtub” curve, so-called because it typically looks like the cross section of a bathtub. Figure 3 also shows a specific value for clock to data timing that will result in a BER of zero; that is, static timing passes.

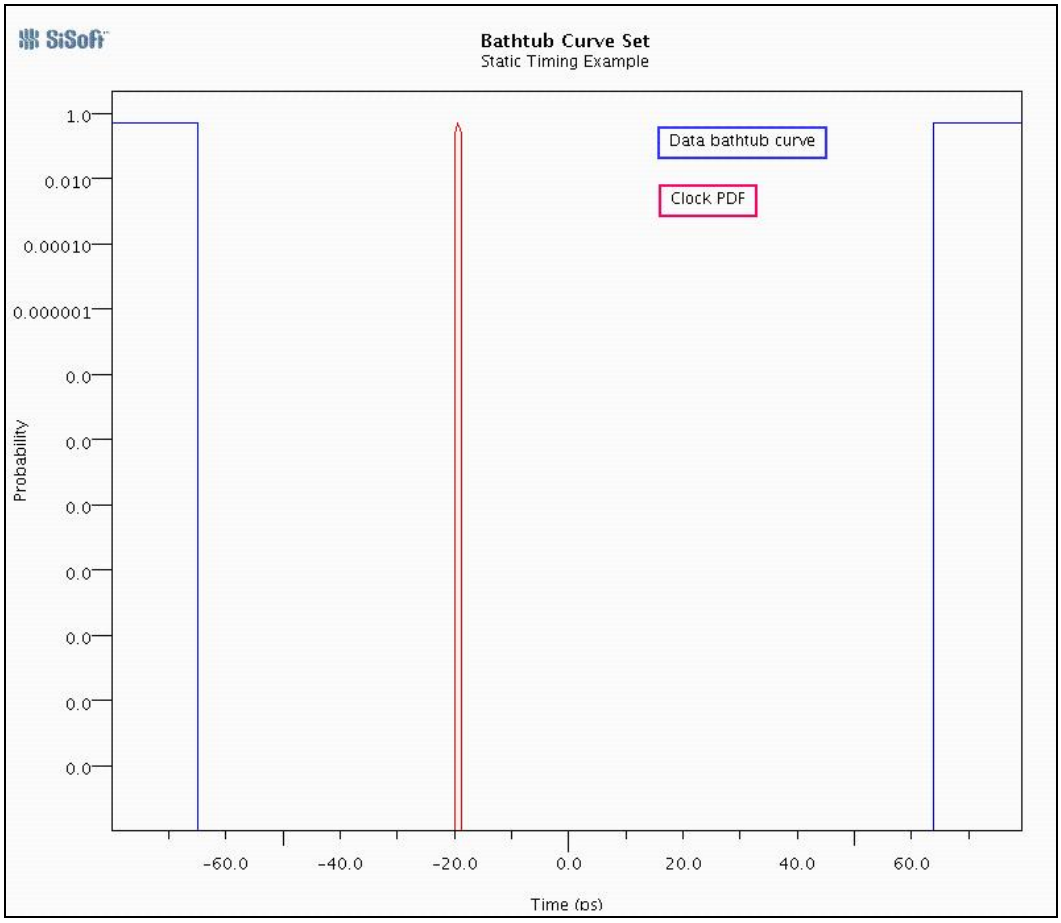

Figure 3: Bathtub curve for static timing analysis example

Figure 4 shows an extension of Figure 3 in which the probability of error at a particular clock to data timing is not necessarily zero or a half, but possibly something in between. That is, at a particular clock to data timing, a data transition may or may not have occurred. There are no longer any guarantees. Similarly, rather than having a single value for the clock to data timing, Figure 4 shows a probability density function for the clock to data timing. In other words, there is a finite probability that the clock to data timing will occur in a particular time window, and there is a finite probability that if the clock to data timing does occur in that time window, an error will occur.

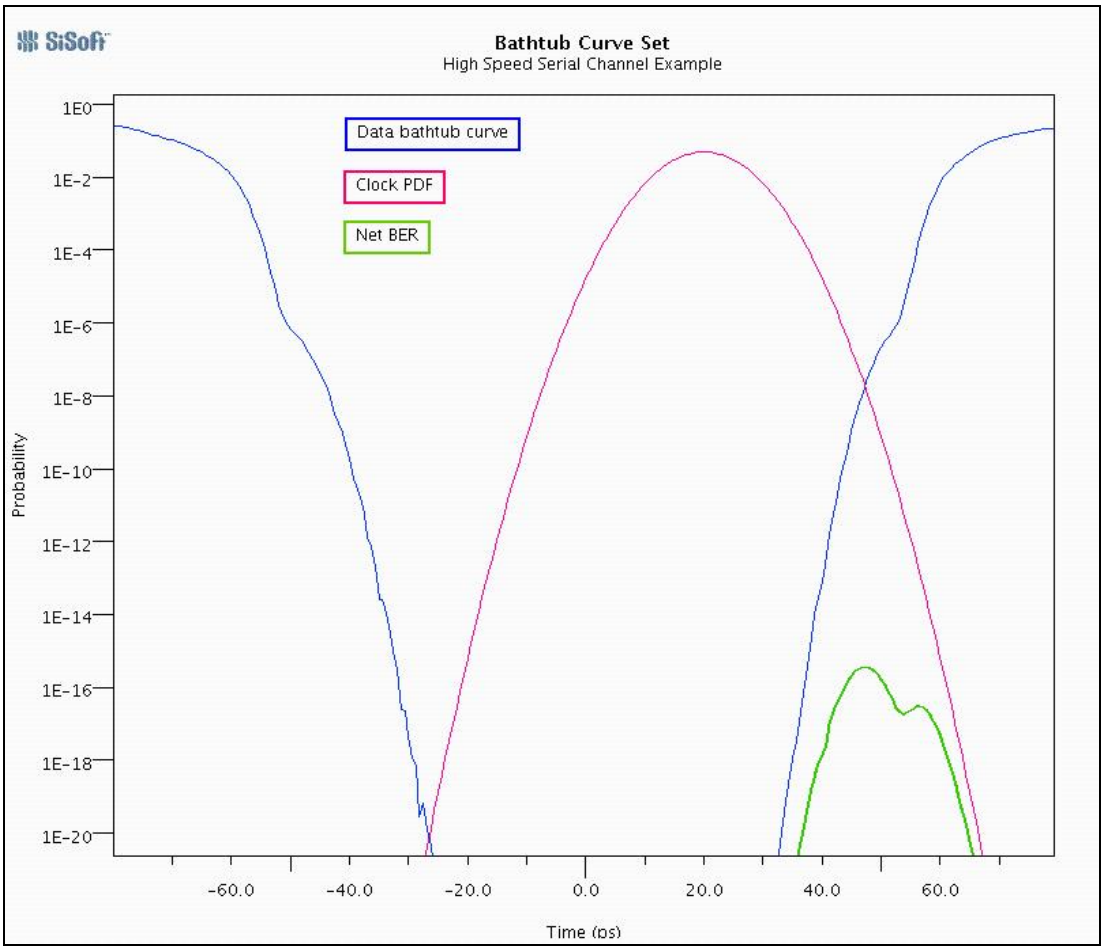

Figure 4: Bathtub curve for high speed serial channel example

If the clock timing is a random process which is statistically independent from the data transition timing, then the probability of error can be calculated using the conditional probability equation

$$P(err) = \int P(err \mid t)\, p(t) dt$$

That is, the bit error rate is integral of the bathtub curve times the PDF of the sampling clock. This also is shown in Figure 4.

6.0 Jitter-based Analysis

Jitter-based analysis reduces the BER estimation to an estimate of the probability that clock to data timing will be met. The bathtub curve is estimated as the probability that a data transition will occur before a given time (on the left half of the curve) or after a given time (on the right half of the curve). The PDF of the clock timing can either be treated as above; but more commonly it is incorporated into the bathtub curve as another source of jitter.

With jitter-based analysis, amplitude disturbances such as crosstalk are converted to timing offsets using an equation such as

$$\Delta t = \Delta v \frac{dv}{dt}$$

This is reasonably accurate so long as $\frac{dv}{dt}$ is relatively constant and unaffected by Δv; and otherwise not. In particular, jitter-based analysis will break down altogether if there is any chance that the total amplitude disturbance can be comparable to the eye height.

Jitter based analysis also assumes that jitter can be added together in a jitter budget; however, this is only accurate if the individual sources of jitter are electrically isolated from one another. While this is a good assumption when combining clock and data timing because the clock and data have separate electrical paths, this is not necessarily a good assumption when a driver and receiver are connected through a passive electrical network.

One should also be careful how one adds the jitter together. The typical approach is to add all the DJ components together as a worst case value, add the variances of the RJ components together, and then combine the two. The typical experience has been, however, that this yields an excessively conservative result. A much more precise and realistic procedure is to explicitly determine the PDF for each independent jitter component, convolve the PDFs together, and then determine the jitter excursion that corresponds with the maximum acceptable bit error rate. As noted in the previous paragraph, however, this procedure will only work if the jitter components are truly independent.

7.0 Communications Analysis

The communications analysis method described here was developed for digital radio communications, and has produced accurate results over a wide range of conditions. In communications analysis, the PDF of the amplitude is computed directly for each possible clock to data timing. For each such timing, the probability of error is the integral of that portion of the PDF which is on the other side of the decision threshold from the transmitted data bit. For example, if the transmitted data was a one and the decision threshold is zero, then the probability of error is the integral of the PDF for all amplitude values less than zero. For NRZ data centered around zero, the BER estimate is

$$P(err \mid t) = \int_{0}^{\infty} p(x,t \mid d = 0)dx + \int_{-\infty}^{0} p(x,t \mid d = 1)dx$$

where *d* is the transmitted data bit.

Amplitude impairments can be combined with the desired signal through convolution. Given two signals *a* and *b* which are statistically independent, the PDF of their sum can be computed as the convolution of their individual PDFs.

$$p_{a+b}(x,t) = p_a(x,t) * p_b(x,t)$$

The assumption of statistical independence is a good one for impairments such as noise and crosstalk, and may or may not be a good one for intersymbol interference, depending on the nature of the payload data and encoding. To achieve its efficiency and accuracy, a convolution engine depends on the assumption that the data bits are independent. Semi-analytical BER

estimation assumes that the noise and crosstalk is independent of the signal but makes no assumptions about the data bits.

Figure 5 illustrates a case in which the crosstalk level is high enough to cause a noticeable bit error rate at any clock to data timing. Communications analysis provides a rigorous treatment of such cases whereas other methods do not.

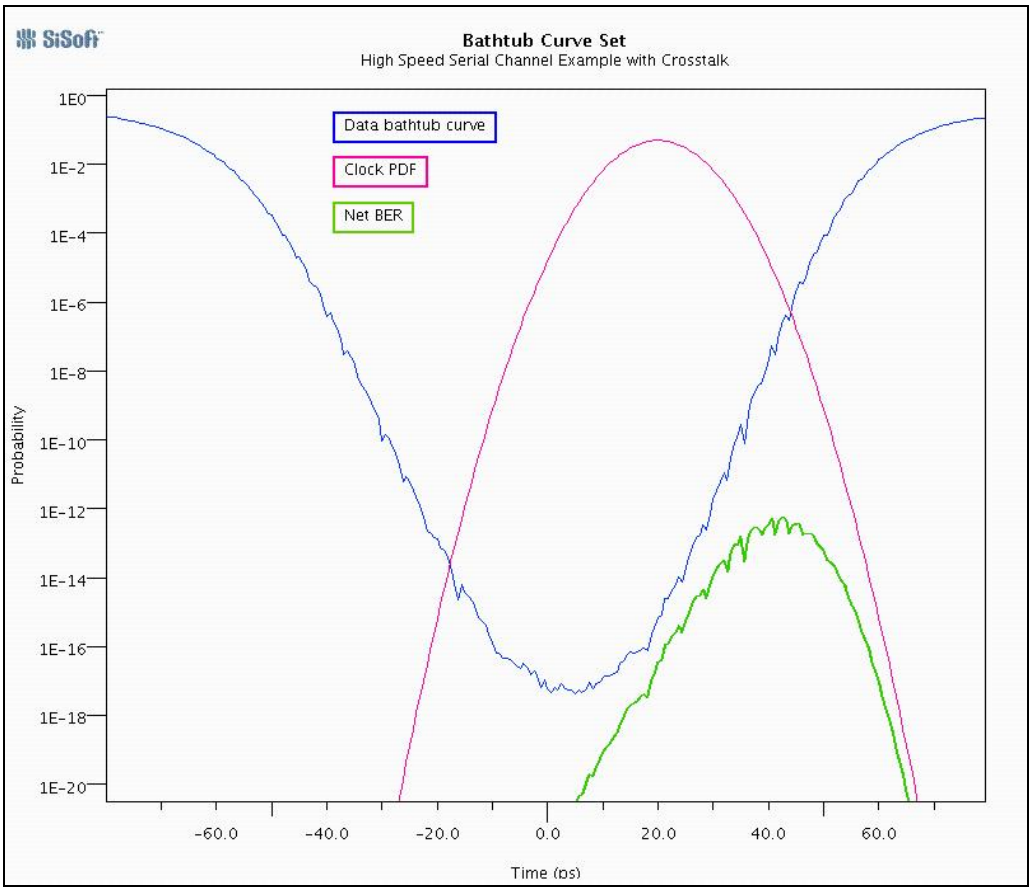

Figure 5: Bathtub curve for high speed serial channel with crosstalk

This method is widely used for digital radio communications, and produces accurate results over a wide range of conditions.

8.0 Equalization Techniques

There are several equalization techniques that are used in various combinations to provide the equalization solution for almost all high speed serial links.

8.1 Transmit De-emphasis

Transmit de-emphasis performs equalization by making the transmitter output equal to the weighted sum of the taps of a tapped delay line, is illustrated in Figure 6. The tapped delay line is usually, but not always, a shift register; which therefore makes the tap spacing equal to the transmit symbol duration. The tap weights for the equalizing taps are usually opposite in sign from that of the primary data output, resulting in an emphasis of the rising and falling edges, as

illustrated in Figure 7.

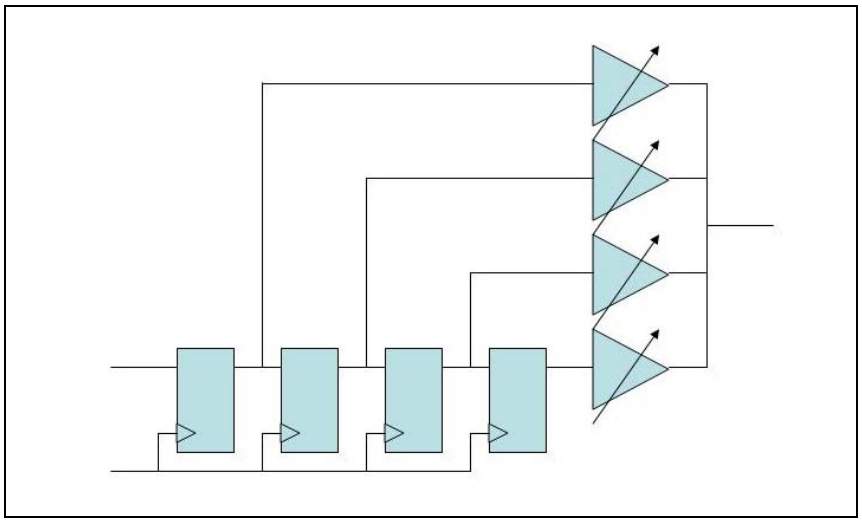

Figure 6: Transmit de-emphasis block diagram

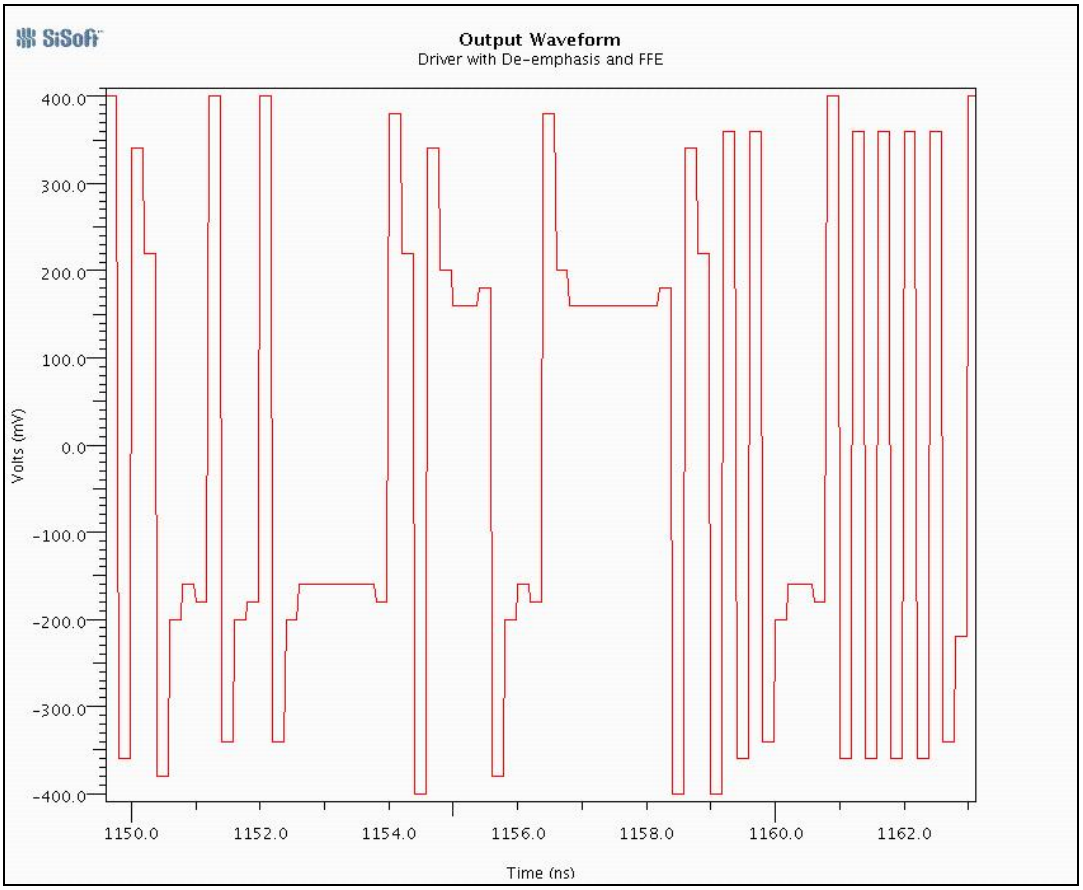

Figure 7: Example transmit output waveform with de-emphasis

Transmit de-emphasis is relatively simple to implement, and can provide the entire equalization solution for paths with losses up to nearly 20 dB. Its disadvantage, and the reason it's called de-

emphasis, is that it is mathematically impossible for transmit de-emphasis to increase the signal amplitude beyond the maximum transmit swing at any frequency, and therefore works by reducing the transmitted signal energy at frequencies which are to be de-emphasized. The inevitable result is a lower amplitude delivered to the receiver, albeit with a data eye that can be quite open.

8.2 Receive Peaking Amplifier

A receive peaking amplifier provides frequency-dependent gain at the input to the receiver, prior to the data decision point. The gain must typically increase with frequency, at least past a frequency equal to one half the data rate, but can then decrease again. It is very desirable for a receive peaking amplifier to provide at least some gain at all frequencies up to the data rate, so as to increase the signal amplitude at the decision point; however, it is common for them to achieve their gain variation by reducing their gain at low frequencies rather than increasing their gain at higher frequencies.

Receive peaking amplifiers are compact, low power, and can be quite effective. It is, however, difficult to design them to maintain their performance across process and temperature, especially if the gain/bandwidth product of the transistors is not large compared to the data rate.

8.3 Decision Feedback Equalization (DFE)

Decision feedback equalization adds a weighted sum of the previously detected bits back into the signal at the decision point. These weights are adjusted to cancel out the intersymbol interference due to those bits.

The primary advantage of DFE is that it can be implemented with very fast, precise adaptive control because the entire control loop is implemented inside the receiver, and the tap weights can be controlled very accurately. This results in a very robust, self-aligning solution. Although DFE is effective by itself, it's even more effective when combined with some other circuit that performs the bulk of the equalization.

A DFE circuit is difficult to design because the previously detected bit must be used to detect the very next bit. Thus, the latency of the feedback path must be much less than one symbol. As a result, receivers with DFE tend to be large and power-hungry. There is, however, an approach to DFE which uses an ultra-fast A/D converter and digital processing to perform the equalization; and this architecture seems to require somewhat less power and area.

8.4 Feed Forward Equalization (FFE)

One of the things that is possible in a transmit de-emphasis equalizer, but not in a peaking amplifier or DFE circuit is to equalize using the data bit which hasn't been sent yet. This so-called precursor bit is therefore an additional degree of freedom that can be used to refine the equalization solution and increase its performance. Use of this precursor bit is called feed forward equalization.

Note that FFE can also be implemented in the ultra-fast A/D architecture mentioned in the previous subsection.

9.0 IBIS AMI Modeling

An IBIS AMI model is a dynamically loadable library (DLL) or shared object library which models the behavior of a transmitter or receiver while conforming to a standardized software interface. Such a model can be run in any software environment or EDA tool designed to match this interface. This approach gives modelers a lot of flexibility while hiding the details of the implementation.

This standardized software interface is defined for three optional functions:

> **AMI_Init:** Initialize the model. For a linear, time-invariant (LTI) channel and circuit block, this function can completely describe the behavior of the circuit block.
> **AMI_GetWave:** Process a sequence of uniformly spaced waveform samples through the model of the circuit block. This is strictly a time domain simulation, and the LTI assumptions are limited to the transmitter and receiver terminating impedances.
> **AMI_Close:** Clean up at the end of the simulation.

This is a departure from traditional IBIS modeling. When IBIS started, HSPICE was the accepted standard for model accuracy and IBIS was a useful approximation. In algorithmic modeling, however, there is currently no accepted standard for model accuracy, and so IP vendors feel that their algorithmic models will be used as the standard, and their IP will be expected to meet or exceed whatever performance their model predicts. The net result is that they will be depending on the time domain simulations, such as those enabled by the AMI_GetWave function, to achieve satisfactory accuracy.

The IBIS AMI standard is currently in draft form, and some models conforming to this draft standard are available.

10.0 Convolution Engine

While a convolution engine can be written to use either a pulse response or a step response, this description will be based on the pulse response.

A convolution engine calculates the statistical eye for a signal by sequentially adding in the contribution of each successive bit offset. For each bit offset, the pulse response for that offset is convolved with the statistical eye that has been accumulated so far. As a result, only N convolutions are required to compute the statistical eye for all messages of length N+1. To achieve the same result in a time domain simulation would require on the order of 2^N bits. Thus, whereas N=20 is practical in the time domain, N=500 is practical with a convolution engine.

The computational method is illustrated in Figure 8. Figure 8a shows an example pulse response, and Figure 8b shows the evolution of the PDF for a single clock to data timing. The only two messages of length one are “1” and “0”, corresponding with a signal value of plus or minus the pulse response at that time position. For all messages of length two, the pulse response at the adjacent bit position can be either added to or subtracted from the signal for the messages of length one. Similarly, for messages of length three, the pulse response at the position two bits away can be either added to or subtracted from the signal for the messages of length two. And so on. The iteration equation is

$$p_{n+1}(x,t) = \frac{p_n(x + r(t + n\tau), t) + p_n(x - r(t + n\tau), t)}{2}$$

where $r(t)$ is the pulse response and τ is the bit time.

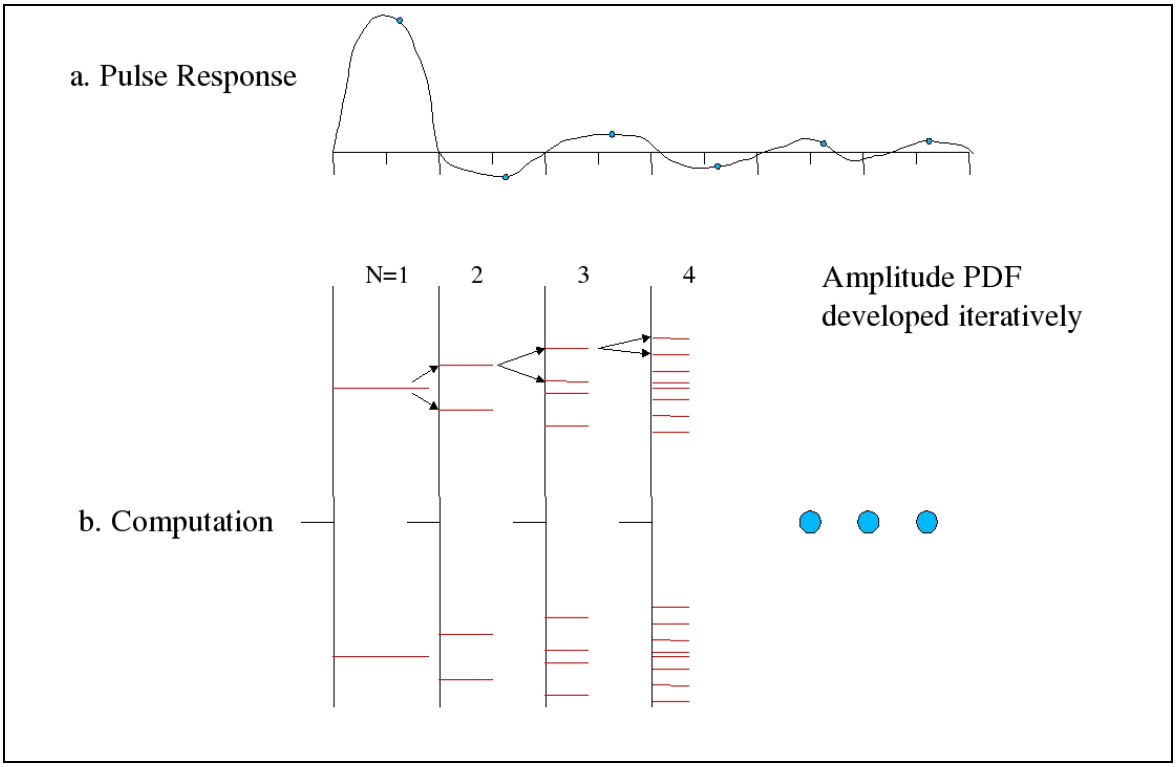

Figure 8: Diagram of convolution engine algorithm

Figure 9 shows a comparison of statistical eyes calculated using time domain simulation and using a convolution engine. Note that whereas the amplitudes/times with the highest probability look the same in both statistical eyes, the eye computed with the convolution engine shows nonzero probabilities for much rarer events, as shown by the blue edges in that eye diagram.

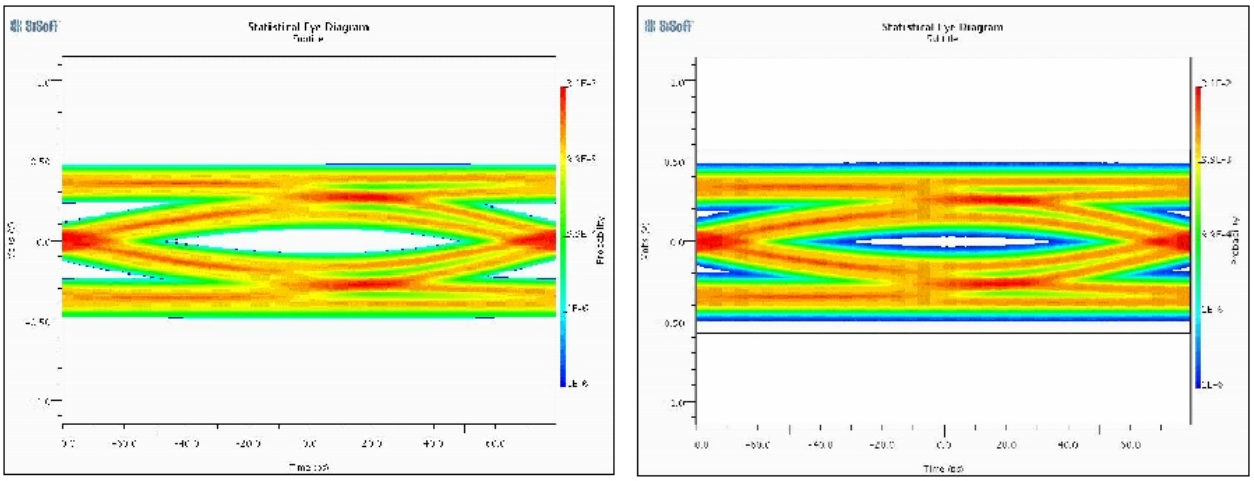

Figure 9: Statistical eyes for time domain simulation and convolution engine

Figure 10 shows statistical eyes with and without crosstalk. As expected, the crosstalk blurs the edges of the eye, in this case to the point where there is no point in the eye where data could be detected error-free.

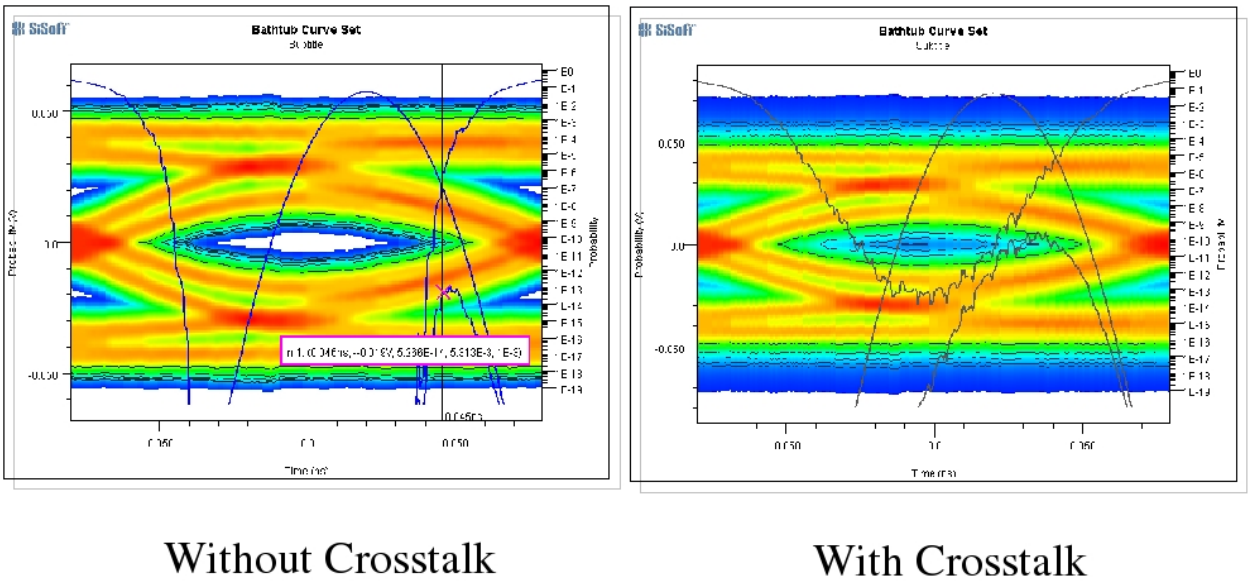

Figure 10: Statistical eyes with and without crosstalk

11.0 Peak Distortion Analysis

Peak distortion analysis takes Figure 9 one step further. Whereas the convolution engine computes the probability density for all possible amplitudes and times, peak distortion only calculates the worst case opening on the inside of the eye; in essence the innermost edge of the blue area in Figure 9. While the organization of the computation is very similar to that for the convolution engine, the computation itself is much simpler because it is only a contour that is being accumulated and not a PDF. If eye opening in the center of the eye is the performance criterion, then the equations are

$$x_0(t) = r(t)$$

$$x_{n+1}(t) = x_n(t) - r(t + n\tau)sign(r(n\tau))$$

$$\forall t \ni -\frac{\tau}{2} \le t \le \frac{\tau}{2}$$

Figure 11 compares the inner eye contour produced by PDA to contours for some finite probabilities, illustrating the conservatism of the PDA result.

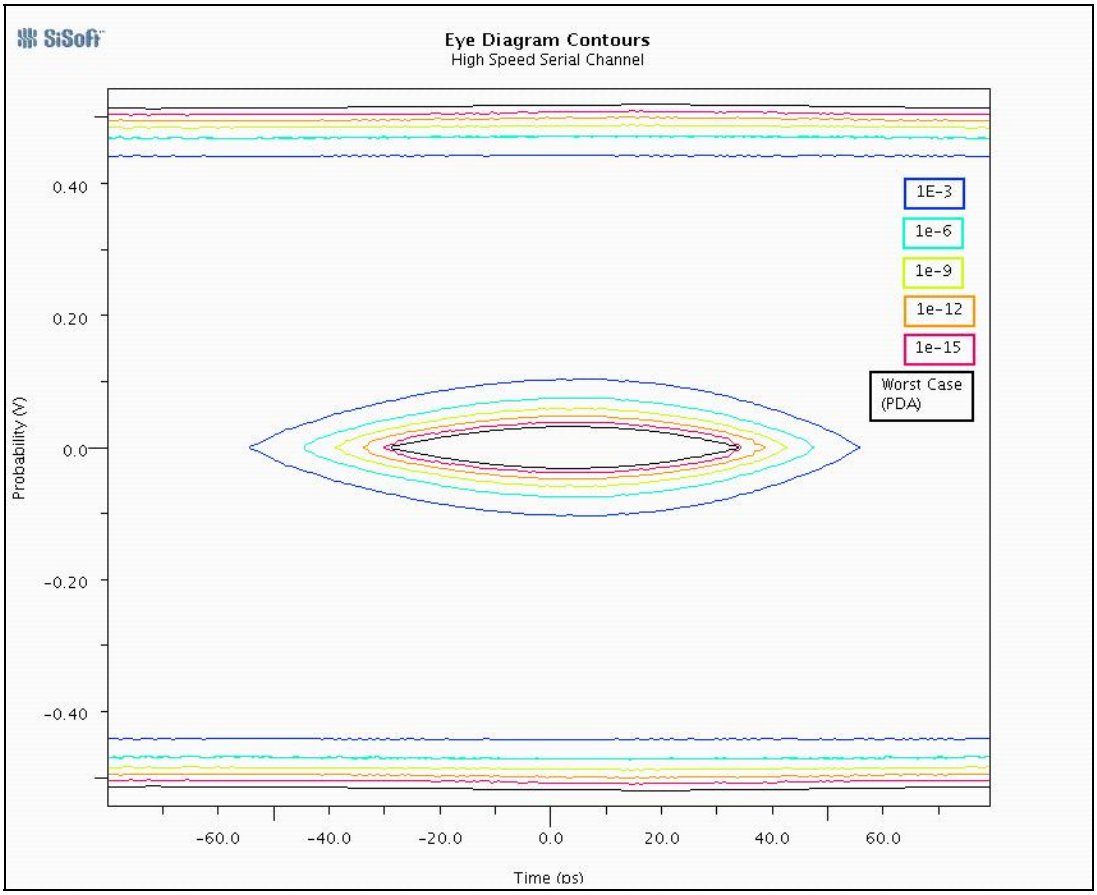

Figure 11: BER contour plots, including PDA contours

12.0 Conclusions and Recommendations

There isn't any one tool, technique, or technical discipline which does the whole job of analyzing serial link behavior. Users must use a collection of tools and techniques to do the whole job, and understanding the advantages and limitations of each technique is essential to doing the job efficiently and well.