Multi-Gigabit Serial Link Analysis using HSPICE and AMI Models

Douglas Burns
Barry Katz
Walter Katz
Mike Steinberger
Todd Westerhoff

Signal Integrity Software, Inc. (SiSoft)
Maynard, MA, USA

www.sisoft.com

ABSTRACT

High-speed serial links are rapidly becoming a primary mechanism used to transfer data between ICs. However, the sophistication of these links is such that predicting their performance using traditional simulation approaches is totally inadequate. Thus, IBIS has adopted a new Algorithmic Modeling Interface (IBIS-AMI) standard to address this challenge. Many vendors are now offering AMI models.

This paper describes how SerDes buffer models can be supplied as HSPICE, AMI, or a combination of both. It also describes how different combinations of HSPICE and AMI Models can be combined into the same analysis. This provides a link between today's HSPICE flow and the future AMI flow.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Multi-gigabit serial link (SerDes) channels are rapidly becoming the primary mechanism used to transfer high-speed data from one IC to another. To date, most serial links have operated up to 3.125 Gb/s and have been simulated using traditional HSPICE time domain simulation, sometimes combined with statistical tools such as StatEye. Serial link data rates are constantly increasing, however, and the analysis techniques that work today will not serve for next generation systems. The important issues from a system simulation point of view are:

1. **Increasing Data Rates and Constant Channel Lengths.** While 3.125 Gb/s was common a year ago, new systems are being designed with data rates of 6.25 Gb/s and higher. System size and interconnect length are remaining fairly constant, while serial link density (the number of serial links per board) is increasing. Higher data rates and constant channel lengths result in increased Inter-Symbol Interference (ISI), where any given data bit is affected by more neighboring data bits. Increased crosstalk due to increased link density exacerbates this effect, driving an exponential growth in the number of data patterns that must be analyzed to provide complete meaningful statistics on design performance. Designers must assess what constitutes statistically significant coverage of the design's behavior, as well as how many patterns can be practically simulated with their simulation tools.
2. **Increasingly complex pin electronics.** As data rates increase, channel distortion becomes severe and requires sophisticated I/O signal processing techniques to achieve reliable data transfer. This increased I/O complexity must be accurately modeled by the simulation environment for design analysis to be useful.

These are conflicting requirements – a need to move from simulating 1,000's of bits to simulating 1,000,000's of bits while ***at the same time*** accurately modeling substantially more complicated pin electronics At first glance, this conflict seems irresolvable; however, there are new assumptions and new analysis techniques that meet these needs. This paper will explore these new possibilities.

It's important to understand which simulation techniques are required to properly analyze a serial link. We believe that a serial link interface can be classified into one of three fundamental speed "bands", each with its own unique set of transmit/receive characteristics and simulation requirements. These three bands are:

- **Up to 1.25 Gb/s** – "Low speed" serial links can be adequately modeled and simulated using traditional non-linear time-domain simulation techniques (e.g. HSPICE). Design standards specify eye masks at both the TX and RX pins; and as long as the simulated or measured behavior conforms to the eye mask, the design is assumed to work.

- **2.5 Gb/s – 5 Gb/s** – In this speed range, transmitter (TX) equalization is required to offset channel losses. Receiver (RX) equalization is optional, generally implemented using a peaking filter. The behavior of the receiver's clock recovery circuit is usually not considered explicitly; as long as the simulated or measured signal at the RX conforms to an associated eye mask, it is expected that a compliant receiver will properly recover the signal.

- **6.25 GB/s and up** – At these speeds, the received eye is usually completely closed at the RX input, and advanced equalization inside the receiver is required to recover a usable signal. The behavior of the RX equalizer (usually including a Decision Feedback Equalizer, or DFE) and the clock recovery circuit are critical to the recovery of a usable signal.

2. Serial Link Analysis Requirements

Each of these three speed bands presents its own unique set of design, modeling and simulation requirements, as summarized in Table 1.

Speed	Example	TX EQ	RX EQ	Simulation Length	Simulation Engine	Compliance metric
< 1.25 Gb/s	SATA 1	None	None	< 5000 bits	HSPICE	Eye @ RX pin
2.5 – 5 Gb/s	PCIe Gen2, XAUI	Two tap FIR filter, scalable output swing	Peaking (Pole/Zero) Filter	100,000 – 1,000,000 Bits	HSPICE, STATEYE, Proprietary	Eye @ RX pin or pad
> 6.25 Gb/s	802.3ap	Multi-tap FIR filter, FFE, scalable output swing	Peaking Filter, DFE, Clock Recovery	>10,000,000 Bits	Proprietary, IBIS-AMI based	Bit Error Rate (BER)

Table 1. Serial Link Analysis Requirements

Low Speed (<1.25 Gb/s) Link Analysis Requirements

Low speed serial links typically don't use equalization at either the transmitting or receiving end of the link. The ratio of the link's electrical length to the bit time is small (typically 5 or less). With signal edge rates of 150ps or more, small structures like vias don't contribute significantly to reflections and ISI. Eye closure due to ISI can be adequately characterized by considering a few hundred or thousand bits and simulation can be performed with traditional time domain simulation techniques (i.e. HSPICE time-domain simulation using transistor models).

Compliance to a standard is demonstrated by showing that the measured or simulated signal at the receiver provides an eye opening larger than a pre-defined eye mask. As long as the signal at the RX input exceeds the defined eye mask, a compliant receiver is expected to recover data to the performance level dictated by the standard. Figure 1 shows typical simulation waveform output, with an associated eye diagram plot in Figure 2.

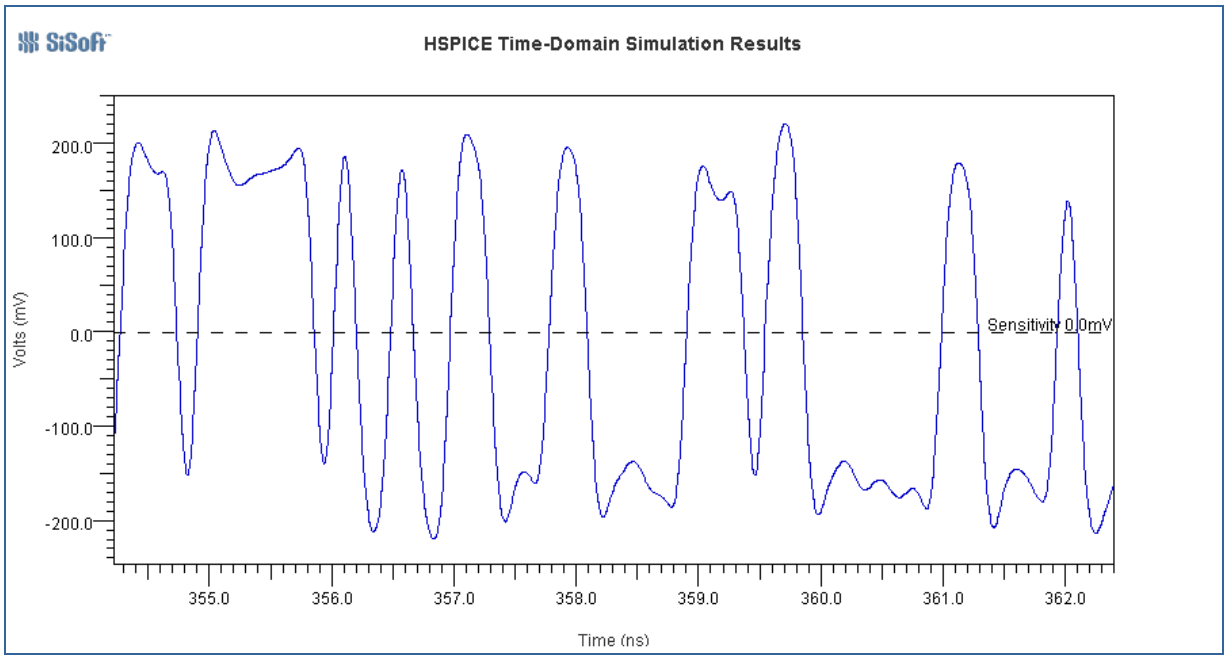

Figure 1. HSPICE waveform output

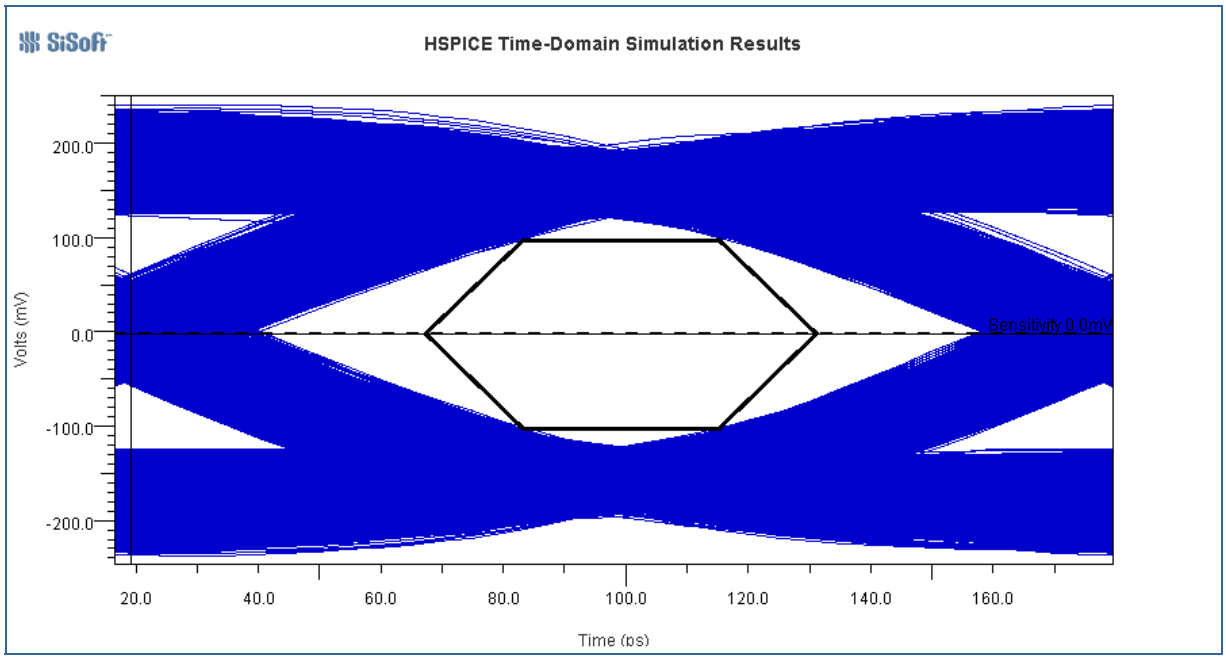

Figure 2. Traditional Eye diagram from HSPICE waveform output

Medium Speed (2.5 Gb/s – 5Gb/s) Link Analysis Requirements

Medium speed links introduce TX equalization and optional RX equalization, requiring I/O models that include equalization behavior. This lets designers use simulation to predict how equalization settings will affect the eye opening seen at the receiver. The ratio of link electrical length to bit time increases to the 10-15 range, and edge rates shrink down to the 75ps range, such that vias begin to contribute to ISI. This causes channel memory (the length of time the effect of single pulse takes to damp out) to increase to 20 bits or more, and can require a million bits worth of analysis to characterize how ISI affects eye closure.

It is difficult to simulate a million bits with a traditional HSPICE simulation, and this is where most existing signal integrity methodologies run into trouble. Fortunately, serial links have operating characteristics that can be leveraged using new analysis techniques to effectively and practically simulate link behavior over millions of bits.

In order to minimize reflections and optimize signal quality on the link, SerDes transmitters and receivers are electrically matched to the channel. Transmitters are typically designed to maintain a differential output impedance of 100 ohms, and receivers include integrated on-die termination, typically with a differential impedance of 100 ohms. Transmitter outputs are often designed to operate in linear fashion as much as possible to maximize signal quality.

Serial link analysis tools (typically called “Channel Simulators”) take advantage of this linear behavior by “characterizing” or “fingerprinting” a link with a circuit simulator (such as HSPICE) and then using that data as a basis for subsequent analysis. The Channel Simulator uses the “fingerprint” obtained from characterizing the channel to predict the link’s behavior over millions or billions of bits. For medium speed serial links, typical Channel Simulation techniques include Peak Distortion Analysis (PDA) and Statistical Channel Simulation. These techniques are described in Section 3.

High–speed (>6.25 Gb/s) Link Analysis Requirements

At these speeds, transmit equalization alone isn’t enough to provide an open eye at the receiver input, and the use of additional equalization within the receiver becomes essential. The ratio of link electrical length to bit time increases to 20 or more, and edge rates shrink below 50 ps. At these edge rates, vias can create reflections that can persist in the channel for 40 bit times or more. Designers grapple with the inherent conflicts between impedance control and anti-pad size for via structures, seeking to optimize the overall frequency response of the channel.

The lack of an open eye at the receiver input is a game-changer for design simulation and lab validation. The open, recoverable eye is now located inside the receiving die, after the equalization circuitry, without direct physical access. Some SerDes IP will "echo" that signal to an external output pin for measurement, which provides useful insight on how RX equalization opens up the received signal. Receivers typically use a Decision Feedback Equalizer (DFE) circuit to provide a form of echo cancellation that helps offset channel reflections. The behavior of the DFE circuit is both non-linear and pattern dependent and must be properly modeled for design simulations. The pattern-dependent (tracking) behavior of the clock recovery circuit is also critical in predicting the link's operating limits.

The basic quality metric for these types of serial links is no longer an eye mask, but how often the link fails to receive the transmitted data correctly. This is known as the Bit Error Rate, commonly abbreviated as BER. A design target for a 6.25Gb/s serial link might be to commit an error in fewer than 1 bit for every 10^{15} bits, or a BER of 10^{-15}. This presents a formidable validation problem, as a 6.25 Gb/s link with a BER of 10^{-15} will only experience a single error, on average, for every 44.44 hours of real-time operation. Accumulating enough real-world measurement data to determine the average error rate is difficult at best, and running a traditional HSPICE simulation of 44.44 hours of real-world operation at 6.25Gb/s is clearly impossible.

To be effective, analysis of these high-speed serial links requires a simulator and models that can account for the non-linear and time varying behavior of the receiver's equalization and clock recovery circuits. The simulator must also be able to take the statistics of equalized data and the recovered clock at the device's sampling latch and use that information to predict the link's overall failure rate, or BER.

The channel simulator is therefore required to "fingerprint" the channel accurately at these very high edge rates, analyze the pattern-dependent behavior of millions of bits worth of stimulus data while accurately emulating the behavior of the receivers DFE and CDR circuits, and finally combine that information to predict the link's overall BER. This is a considerable simulation challenge. Fortunately, there ARE simulators up to this task. Not surprisingly, the first Channel Simulators with these capabilities were created by SerDes vendors that based their simulators on their respective semiconductor devices and processes.

Serial Link Analysis Approaches

As you would expect, the different serial link speed bands require different modeling and simulation strategies. We'll discuss modeling and simulation techniques for each speed band separately.

Channel Simulation has evolved rapidly over the past several years, resulting in a variety of different tools – some public, some commercial, some proprietary – all of which predict serial link behavior over millions or billions of bits. As will happen in any rapidly developing field, each tool has a unique set of features and capabilities; but there are three broad classes of channel simulation tools that have become apparent:

- Peak-Distortion Analysis Tools
- Statistical Channel Simulators
- Time-Domain Channel Simulators

Each of these tools takes in fingerprint data for the channel, performs analysis and outputs results; as will be discussed below, each class of tools has one or more strategies for TX/RX equalization modeling.

Peak Distortion Analysis Tools

Peak Distortion Analysis (PDA) tools don't attempt to predict a link's behavior over many millions of bits – in fact, they do just the opposite. PDA tools take a channel fingerprint and use it to predict the stimulus sequence that will achieve maximum eye closure (minimum open eye height) using as short a stimulus sequence as possible. The usual goal of a PDA tool is to define a stimulus sequence that can be used with traditional time-domain HSPICE simulation to produce an eye with minimum eye height. Since this is supposed to be the worst-case pattern, any other stimulus sequence, no matter how many bits are simulated, should produce an eye with not less than this eye height and usually more.

PDA tools usually assume that any TX and RX equalization is included in the fingerprint provided, which is usually supplied as a pulse response. A typical analysis flow is:

1. Simulate the link in HSPICE with TX & RX equalization configured to their desired settings. Use a single pulse as the input stimulus and run simulation until all transients settle out
2. Reformat HSPICE waveform output for the PDA tool, feed to the PDA engine
3. Reformat PDA engine output (usually a bit sequence) to create appropriate HSPICE stimulus statements
4. Re-simulate HSPICE network from step one using stimulus from step three
5. Plot HSPICE waveform from step four and measure eye height

The challenge with conventional PDA techniques is that they consider only a single measure of eye quality – typically eye height at the 50% point of the Unit Interval; and they only produce the associated worst-case stimulus. This tells us the absolute worst-case height at one specific point in time, but it tells us nothing about the ***probability*** of how often that minimum height occurs; and it tells us nothing about the distribution of signals at any other point in time. What input pattern, for example, would provide an eye with minimum width? What would be the effect on link behavior if instead of sampling at the 50% point of the Unit Interval, the recovered clock sampled at the 55% point? With most PDA tools, we simply have no idea.

PDA is useful because it gives us a quick and deterministic way to define a pattern sequence that can be used with HSPICE to give us an idea of worst-case eye behavior. What it lacks, though, is the ability to give us real insight into probabilities within the eye and how they affect link error rate.

One technique for performing Peak Distortion Analysis is reviewed in [2]. This is just one of many different sets of implementations.

Statistical Channel Simulators

Statistical Channel Simulators take the channel fingerprint and use statistical techniques to predict the distribution of the signal in the receiver eye. These statistical techniques can compute the equivalent of billions of bits worth of simulation within seconds, subject to a set of fundamental assumptions that should be clearly understood.

Conceptually, a statistical channel simulator takes the pulse response for the channel and folds it back on itself to predict the channel's response to all possible combinations of "1" and "0" input bits. The result is an eye diagram that shows the probability of where the signal will be in voltage and time. Because this eye represents the behavior of so many bits, it is better to present the eye diagram with colors used to represent different probability levels. Statistical Channel Simulation assumes the data transmitted through the channel is random and un-encoded; there is no stimulus pattern associated with the simulation, and no time-domain "waveform" that can be plotted after the analysis is complete.

Statistical Channel Simulators make the fundamental assumption that any TX or RX equalization is both Linear and Time-Invariant (LTI). Equalization modeling can be handled in one of three ways:

- Equalization for the TX and RX can be included in the fingerprint provided to the Statistical Channel Simulator
- Equalization for the TX and RX can be omitted from the fingerprint and added explicitly by the Channel Simulator
- The first and second techniques can be combined (TX one way, RX the other)

Because Statistical Channel Simulation effectively simulates an unlimited number of bits, its results offer rich statistical significance. As long as the channel fingerprint and equalization models are reasonably accurate, computing design probabilities that demonstrate a design's predicted Bit Error Rate to be below 10^{-15} is not a problem.

Understanding the fundamental assumptions associated with Statistical Channel Simulation is key to applying this class of tool effectively. Statistical Channel Simulation is not a replacement for other forms of analysis. It is complementary. It allows the designer to simulate billions of bits of channel behavior within seconds, but does not preclude the need for more detailed forms of simulation.

Since the output of a Statistical Channel Simulator is an eye at the receiver coded for probability, this fits well with the compliance metric of choice for low and medium speed links – the eye mask at the receiver. The eye mask can be imposed on the eye predicted by the simulator to determine if the design requirements are met. Two advantages of Statistical Channel Simulation over conventional HSPICE time-domain analysis are:

- The eye represents a MUCH larger sample (usually effectively infinite), so the confidence level about maximum eye closure is much greater
- The eye is color-coded for probability, so it's possible to assess eye height and width at different probability levels – 10^{-9}, 10^{-12} and 10^{-15}, for instance (Eye Contours).

A color-coded eye from Statistical Channel Simulation and its associated eye contours are shown in Figures 3 and 4.

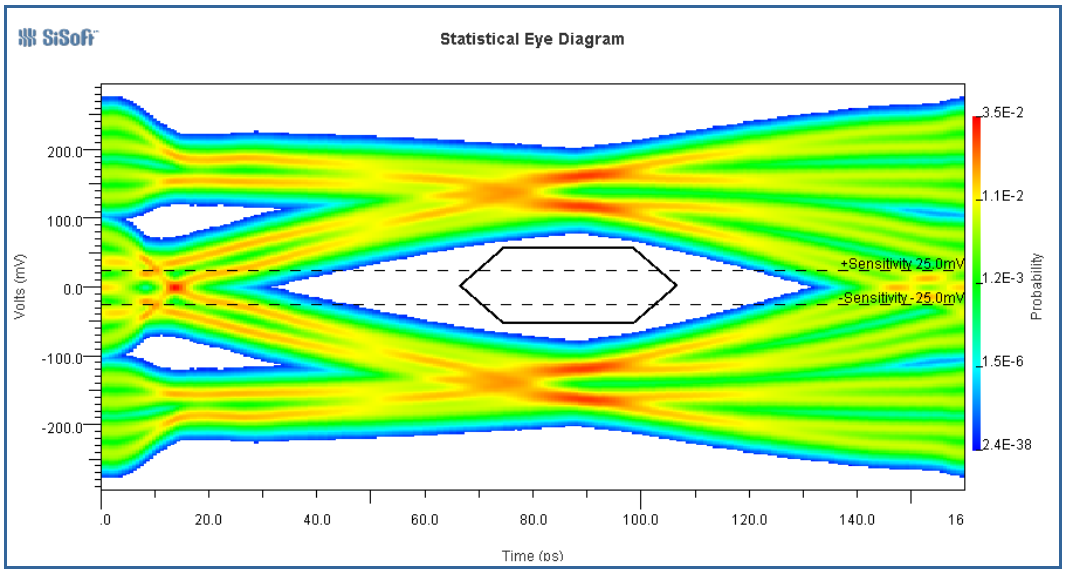

Figure 3. Statistical Eye Plot with Eye Mask

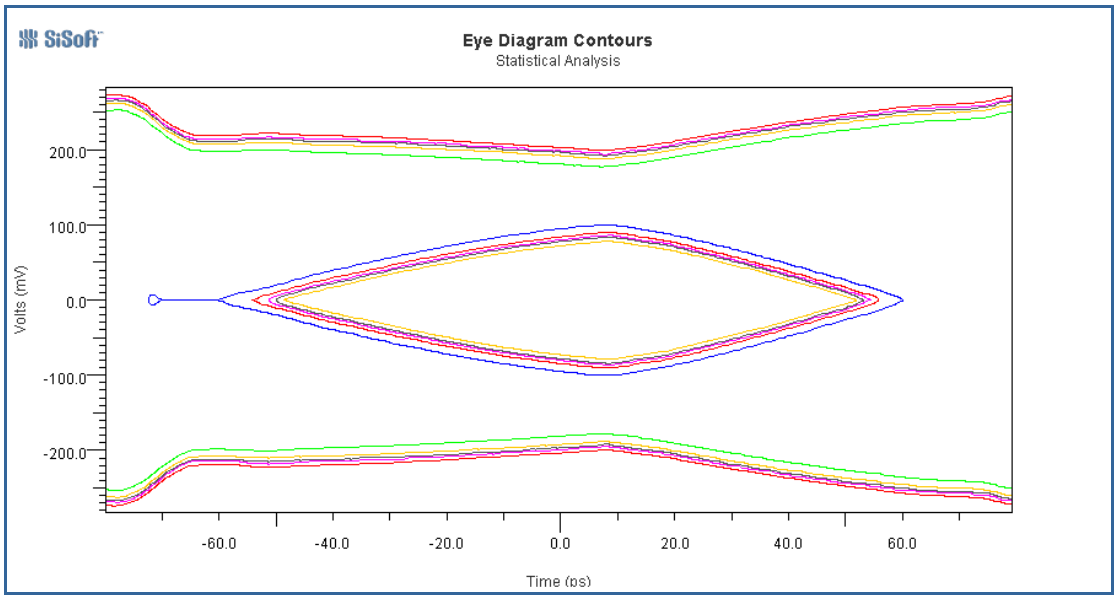

Figure 4. Eye Contours from Statistical Channel Simulation

Statistical Channel Simulation works well for low and medium frequency serial links, but the behavior of DFE and clock recovery circuits in high-speed links violates the Statistical Channel Simulation assumption that all link behavior is linear and time-invariant. Statistical Simulation can be used to ***approximate*** the behavior of DFE and CDR circuits, but these simulations will have to be backed up with more detailed simulations later on in the design cycle.

More information on the concepts behind Statistical Channel Simulation can be found in [3].

Time-Domain Channel Simulators

Where Statistical Channel Simulators assume the input data is random and un-encoded, Time-Domain Channel Simulators allow the use of arbitrary input patterns and encoding schemes. The channel fingerprint is still used in that the combined behavior of the transmitter's analog driver, channel and receiver termination network are still assumed to be linear and time-invariant.

Because a Time-Domain Channel simulator analyzes the link's response to specific bit sequences, SerDes IP models are expanded to include non-linear and time-varying behaviors. This allows Decision Feedback Equalization to be modeled, along with algorithms used by the DFE to dynamically adapt tap coefficients. The details of the clock recovery circuit and its adaptive behavior can be modeled as well. The receiver model in the Time-Domain Channel simulator outputs two pieces of data – the post-equalization waveform provided to the sampling latch, and the clock signal produced by the clock recovery circuit. The simulator combines these two pieces of data to determine how often the link will receive a bit in error.

Remember that the behavior of the analog portion of the channel is still assumed to be LTI, and the Time-Domain Channel Simulator uses a channel fingerprint data in the same way the Statistical Channel Simulator did. Thus, the behavior of the analog circuitry is solved only once

during fingerprinting. Time-Domain Channel simulation evaluates only the signal's voltage as a function of time. This allows processing to occur much faster (typically 1,000,000 bits/minute) than traditional time-domain circuit simulation methods.

The Cloud of Confusing Claims

Working on one's first serial link project is both exciting and frustrating. On the one hand, it offers the opportunity to master a new set of terminology and analysis skills, which is always valuable from a career perspective. On the other hand, almost no one is ever actually ***given*** the time to develop and master these skills, and learning these techniques "on the fly" during the course of an actual project is a huge task.

It doesn't help that the systems designer new to serial links is faced with a large collection of new tools, techniques and jargon. Designers must develop working knowledge of frequency domain and time domain analysis techniques, along with basic skills in statistical analysis. Some tools create models for pieces of the channel (like vias), other tools create electrical models of the entire passive channel (in either S-parameter or Laplacian form), and others (Channel Simulators) are used to simulate how the SerDes IP interacts with the channel to predict the link Bit Error Rate. Even if one just looks at Channel Simulators, the collection of stated modeling and simulation capabilities can be overwhelming.

We'd like to do our part to reduce confusion by pointing out that although there are ***many*** different Channel Simulators, they all fall into one of the three basic categories we outlined at the start of this section. Fancy claims aside, we believe a fundamental issue (and a basis for any serious comparison) is the ability to easily and accurately model the behavior of SerDes IP. Without real models of the SerDes IP's behavior, running a lot of advanced simulation doesn't serve much of a purpose.

3. The Interoperability Problem

The rapid evolution of Channel Simulation tools over the past few years has resulted in a large collection of different (and generally incompatible) simulation tools. Variety is usually a good thing, but not for the systems designer looking for a commercial simulation platform with modeling support for a wide collection of SerDes IP.

Channel Simulators use one of three strategies for modeling SerDes IP:

1. Assume any SerDes Equalization is included in the fingerprint supplied to the Channel Simulator
2. Use SerDes Equalization and Clock Recovery models "built in" to the Channel Simulator
3. Use external SerDes Equalization and Clock Recovery models that "plug in" to the Channel Simulator

The System Designer needs a simulation platform that provides three fundamental capabilities:

1. Supports a variety of different SerDes IP
2. Simulates the design behaviors that need to be analyzed
3. Simulates enough bits in a reasonable time to provide statistically significant results

The first modeling strategy meets two of these three needs. Because the SerDes equalization is included in the channel fingerprint, any one of several modeling and simulation methods can be used to derive the fingerprint. For instance, if the Channel Simulator requires a pulse response as its input fingerprint, transistor level models and HSPICE time-domain simulation can be used to derive that pulse response. This modeling strategy is compatible with both Statistical and Time-Domain channel simulation, either of which could produce a statistically significant simulation size. The trouble with using SerDes Equalization in the Channel fingerprint as a primary modeling strategy is that it makes the assumption that both the TX and RX Equalization are linear and time invariant. This prevents the Channel Simulator from being able to accurately simulate High Performance (>6.25 Gb/s) link behaviors, including Decision Feedback Equalization, Adaptive Equalization and detailed Clock Recovery.

The second modeling strategy brings the SerDes IP model inside the Channel Simulator to meet needs two and three. The SerDes IP model represents the equalizer's nonlinear and time-varying behavior, and includes details of how the clock recovery circuit operates. This allows a SerDes vendor to offer a Channel Simulator with models based on their proprietary device algorithms and processes. The challenge with this strategy is that it can't provide support for multiple SerDes IP vendors. One SerDes vendor's simulator won't model another vendor's devices. A commercial EDA company can offer a Channel Simulator with a set of generic equalization models, but those models won't represent the details of any given SerDes vendor's actual IP.

The third modeling strategy builds upon the second by defining a standard interface between the SerDes IP models and the Channel Simulator, thus making SerDes IP models "pluggable" into the simulator. This strategy satisfies all three of the System Designer's simulation needs.

SerDes IP modeling strategies evolved pretty much in the order listed. The third strategy resulted in the IBIS Algorithmic Modeling Interface (IBIS-AMI), which was standardized as part of IBIS 5.0 in August 2008. IBIS-AMI models include the detailed behaviors of actual SerDes IP and provide the following benefits to the systems designer:

- Fast, efficient execution (Simulation speed)
- Models run in multiple EDA simulators (Transportability)
- Models from different vendors work together (Interoperability)
- Supports Statistical and Time-Domain simulation methods (Flexibility)
- Protects SerDes vendor IP (Vendor Support)

IBIS-AMI defines serial link analysis as a two-step process:

- **Network Characterization** uses **Analog** models for the TX and RX in conjunction with the channel model to fingerprint the analog network. The IBIS-AMI standard defines an impulse response as the fingerprint used for Channel Simulation.
- **Channel Simulation** uses **Algorithmic** models for the TX and RX in conjunction with the impulse response of the analog network to predict end-end link behavior. The Algorithmic models represent the SerDes IP's equalization and clock recovery behavior, and support both Statistical and Time-Domain Channel Simulation.

IBIS-AMI Algorithmic models are supplied as executable object code that is linked into the Channel Simulator at runtime. This provides the highest simulation performance possible. The IBIS-AMI standard explicitly defines the calling interface between the simulator and the SerDes IP model, facilitating model transportability and interoperability.

IBIS-AMI Algorithmic models use an external ASCII control file (called the .AMI file) that tells the Channel Simulator which functions the model supports and how to deal with any model-specific control parameters. This allows the Channel Simulator to expose any user-controllable settings in its GUI (for example, tap equalization coefficients that the user can control).

IBIS-AMI models have both Analog and Algorithmic sections that must be developed to work together when modeling the overall behavior of the device. IBIS-AMI Analog models represent the transmitter's output stage or the receiver's input termination network, and are often based on characteristics extracted from HSPICE transistor-level simulations. Since IBIS-AMI Algorithmic models are supplied as binary code, they can be developed by any method that conforms to the calling interface. C, C++, and wrapped MATLAB code have all been used to create IBIS-AMI Algorithmic models. To help new model developers, SiSoft supplies a free modeling toolkit and sample transmitter source code on its website at www.sisoft.com.

4. IBIS-AMI Based Analysis

When IBIS-AMI models are available for both the TX and RX in a serial link, an IBIS-AMI Compliant Channel Simulator can be used to simulate end-end network behavior and predict a link's BER. A full-featured analysis environment will perform both Network Characterization and Channel Analysis using the IBIS-AMI Analog and IBIS-AMI Algorithmic models, respectively. The details of how the network model is captured will vary from tool to tool, but a full-featured environment will allow the network to be described as an arbitrary collection of lossy transmission lines, subcircuit and S-parameter data, as shown in figure 5.

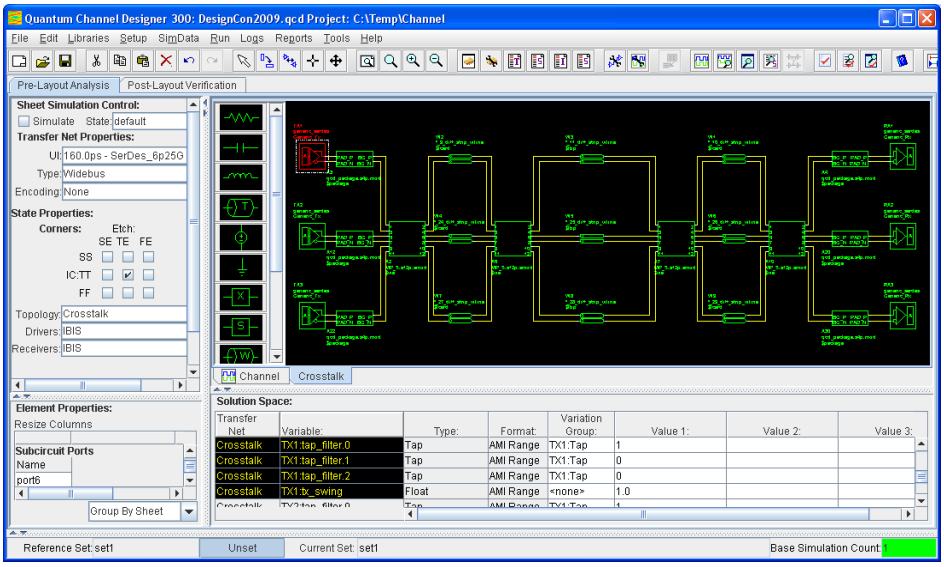

Figure 5. Design capture with IBIS-AMI models

Network Characterization can be performed in any one of several different ways, with the only requirement being that the analysis provides the impulse response of the unequalized analog network to the Channel Simulator. Traditional circuit analysis techniques are used in either the frequency or time-domain, depending on the techniques chosen.

A full featured IBIS-AMI Channel Simulator will support both Statistical and Time-Domain simulation modes. Results from Statistical and Time-Domain simulations should be presented in a variety of formats, shown in Figure 6.

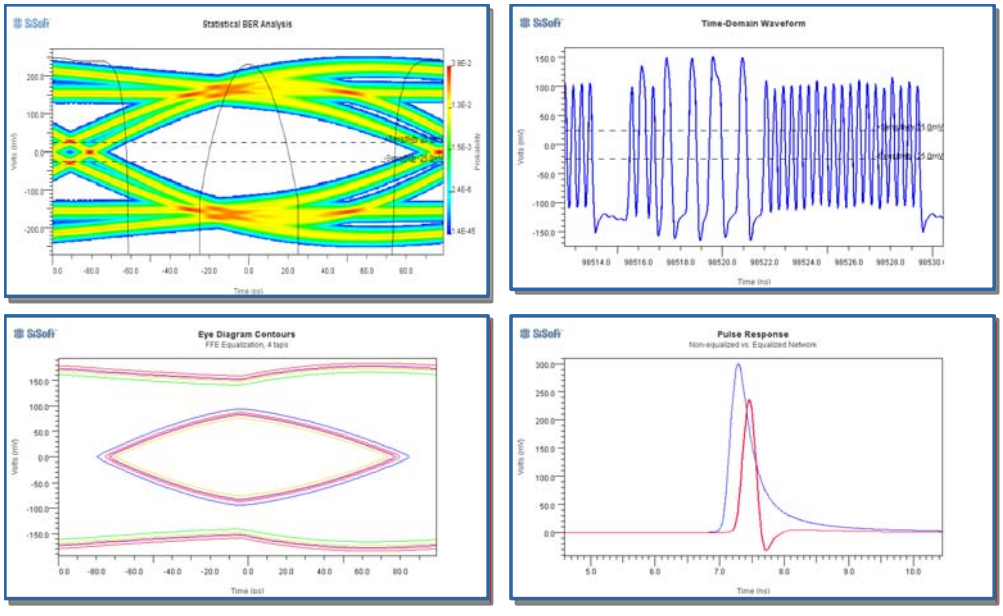

Figure 6. IBIS-AMI simulation results

5. Mixing IBIS-AMI and HSPICE Models

… but what happens when IBIS-AMI models aren't available for the SerDes IP at both ends of the link? In this case, a hybrid modeling and simulation strategy proves useful. If an HSPICE model is available for the end of the link that doesn't have an IBIS-AMI model, that device's equalization can be included in the channel impulse response produced during Network Characterization.

Network Characterization in this case simulates the analog network and at least one SerDes device's equalization behavior in HSPICE. A short time-domain simulation run in HSPICE is used to produce a step or pulse response that is passed to the Channel Simulator. The Channel Simulator then derives the required impulse response. That impulse response includes the equalization behavior of the device being completely represented by the HSPICE model. The impulse response is fed into the Channel Simulator, which uses an IBIS-AMI Algorithmic model to represent the behavior of the SerDes IP at the other end of the link.

This method works with both Statistical and Time-Domain Channel Simulation. The only limitation is that since the equalization behavior for the device being represented by a HSPICE model is included in the impulse response, the equalization behavior of that device is assumed to be Linear and Time-Invariant (LTI).

6. Summary

In this paper, we identified three distinct bands of serial link speeds along with their typical TX/RX configurations. We outlined the simulation and modeling requirements associated with each speed band.

We noted that traditional simulation methodologies run into trouble around 3.125 Gb/s, resulting in the creation of a new class of Channel Simulation tools. We reviewed the most popular Channel Simulation techniques and categorized them into three classes: PDA, Statistical and Time-Domain Channel Simulators. We examined different SerDes IP modeling strategies used by Channel Simulation tools and identified which performance bands were supported by which modeling strategies.

We introduced the IBIS-AMI standard and showed how it uniquely meets the simulation requirements for high-speed serial links, and how IBIS-AMI based Channel Simulators meet the modeling needs of high-speed Systems Designers. Finally, we showed how IBIS-AMI and HSPICE models can be mixed together to simulate serial where IBIS-AMI models are not available for one end of the link.

Many of today's serial link simulations are applied to the middle speed band, where multiple simulation and modeling approaches are possible. As more designs target speeds of 6.25 Gb/s and up, we believe that IBIS-AMI models and IBIS-AMI based simulators will become the prevailing standard.

7. References

[1] Michael Steinberger and Todd Westerhoff, “Multi-Gigabit Serial Link Analysis – Piecing Together a Design and Verification Strategy”, SNUG Boston 2007, September 17, 2007.

[2] Casper, B.K.; Haycock, M.; Mooney, R, “An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes”, VLSI Circuits Digest of Technical Papers, 2002. Digital Object Identifier 10.1109/VLSIC.2002.1015043

[3] Sanders, Anthony, “Statistical signal analysis (SSA) demystified”, EE Times-India, August 2007, http://www.dspdesignline.com/showArticle.jhtml?articleID=201803329