WP382 (v1.0) December 9, 2010

SerDes Channel Simulation in FPGAs Using IBIS-AMI

By: Romi Mayder

The IBIS Algorithmic Modeling Interface (IBIS-AMI) was developed to enable fast, accurate statistical and time-domain simulation of high-speed channels. It combines the ease of use and speed of standard IBIS signal integrity analysis with advanced communications analysis techniques.

As a member of the IBIS Advanced Technology Modeling Group, Xilinx has worked with industry-leading customers and EDA vendors to provide IBIS-AMI models for SerDes channel simulation.

© Copyright 2010 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. PCI, PCI Express, PCIe, and PCI-X are trademarks of PCI-SIG. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. All other trademarks are the property of their respective owners.

Overview

Over the past several years, high-speed FPGA interfaces have moved from fast LVDS at frequencies of up to 1.6 Gb/s, to very-high-speed serial interfaces of up to 3.125 Gb/s (PCIe® and XAUI, for example), to current speeds of up to 11 Gb/s.

Systems designers incorporating high-speed serial links face a significant challenge. As speeds increase and channel lengths remain constant, signal quality decreases to the point where the signal's eye closes at the receiver input pin. Historical methods of signal integrity analysis are no longer sufficient for simulation and analysis of these systems. Standard transistor-level HSPICE simulations and the use of receiver eye mask templates are also no longer suitable. Present-day communications systems analysis requires microwave design techniques and bit error rate (BER) analysis.

SerDes Simulation Requirements

As serial channels first became prevalent and moved to data rates of 1.25 Gb/s and higher (Gigabit Ethernet base-X and SGMII), standard signal integrity simulation methods were initially used. In general, a time domain simulation utilizing HSPICE transmitter and receiver models combined with lossy transmission line parameters were used to generate eye diagrams at a receiver input. Results were compared to a received Eye Mask requirement; if the simulated eye was within the eye opening specification for amplitude and jitter, then a better-than-specified BER (usually 10^{-12}) would be achieved. At low data rates, this worked — but as speeds rapidly increased from 2.5 Gb/s to 28 Gb/s, this method quickly ran into problems. For any reasonably lossy interconnect at data rates approaching 5 Gb/s, the received eye could be nearly closed, yet the signal could still be recovered through the use of equalization techniques at the receiver end. At the same time, lower BER design goals were needed, moving from 10^{-12} to 10^{-15} or 10^{-17}.

Simulation Goals

The ultimate goal for SerDes channel simulation is to be able to quickly and accurately evaluate and optimize the channel for performance. This allows for margin analysis and robustness of the design as well as verification of design implementation. Potential design tradeoffs should be easy to explore. For example: Which connectors have acceptable performance? What impact do board structures like vias have? What trace loss budgets are needed?

Before IBIS-AMI, system designers faced significant limitations when performing serial link simulations:

- Traditional SPICE-based analysis was slow and could not simulate the millions of bits needed to accurately predict link operating margins.
- Open-source statistical analysis tools could simulate many millions of bits, but they could not accurately model a specific semiconductor vendor's device.
- Proprietary semiconductor vendor tools could accurately model their own SerDes devices and simulate millions of bits, but could not be used when devices from different semiconductor vendors were used at each end of the link.

Initially, many SerDes vendors began offering proprietary simulation environments. Many of these simulators were based on MATLAB®, which is well suited for both matrix manipulation and communications modeling. However, each SerDes vendor's simulation environment was proprietary, and the different environments did not work with each other. Proprietary tools are acceptable as long as the SerDes devices at both

ends of the link are from the same supplier, but they are inadequate where device and simulator model interoperability are required.

Performing adequate simulation to sufficiently characterize channel behavior is also inherently difficult. Millions of bits worth of simulation are needed to understand the effects of inter-symbol interference (ISI) and deterministic jitter (DJ), for example. Traditional SPICE-based time domain simulation does not provide the amount of simulation needed in any reasonable timeframe, so a methodology other than brute force time domain simulation (such as statistical analysis) is required.

Some open-source statistical analysis tools provide high-performance analysis, but they fall short because they cannot model vendor-specific SerDes parameters.

Fundamentals of SerDes Links

Serial links are generally designed using differential signaling, with high performance analog drivers and controlled termination networks at the receiver's input. The use of well-designed analog drivers and receivers makes it possible to separate the analysis of the analog network (driver output, interconnect, and receiver input) from the modeling of the equalization and clock recovery circuits. In this two-stage process, the analog circuit is first characterized as an impulse response or equivalent; that characterization data is then used as input to a second stage of modeling and analysis:

1. Stage 1 (analog network characterization) involves traditional circuit modeling techniques. For example, the analog circuit might be modeled and simulated using SPICE to produce a step response, which is then differentiated to produce an impulse response.
2. Stage 2 (communications analysis) combines the analog circuit's characterization with models of TX/RX equalization and clock recovery to predict the overall channel behavior.

Utilizing a two-step approach, a function of the software implementation in Quantum Channel Designer (QCD) by Signal Integrity Software, Inc. (SiSoft), enables performance optimization. Separating characterization of the analog channel from modeling of equalization and clock recovery makes it possible to simulate at rates of about 1,000,000 bits per minute. With this level of performance, reasonable characterization of ISI and DJ becomes practical. The challenge then becomes determining how a particular vendor's SerDes will be modeled, and what simulation engine will be used to produce the simulation results.

The IBIS Algorithmic Modeling Interface (IBIS-AMI)

IBIS-AMI Background

The IBIS Advanced Technology Modeling (IBIS-ATM) working group recognized that most of the proprietary simulation environments were doing essentially the same thing: first characterizing the analog network, and then using communications analysis techniques to predict the behavior of the overall channel. A standardized modeling approach could pave the way for both model interoperability and the development of robust, commercial SerDes EDA software.

The working group established a set of specific requirements for the IBIS-AMI, which was ratified in IBIS 5.0 in August 2008.

The following design goals were met:

- **Interoperability**. Models from different semiconductor vendors work together.
- **Transportability**. The same model runs in different IBIS-AMI simulators.
- **Performance**. 10,000,000-bit simulations run in 10 minutes or less.
- **Flexibility**. Models support both statistical and time-domain simulation.
- **Usability**. Models expose control parameters users can set for simulation.
- **IP Protection**. Models cannot be reverse-engineered; semiconductor vendors control which details are exposed to the user, and their proprietary silicon IP is protected.

Simulation Modes

IBIS-AMI supports modeling at two different levels. The first level of modeling (referred to as "Init") processes impulse responses. An impulse response is passed to the model, which applies its respective equalization and passes back an equalized impulse response. TX and RX models can be chained together to provide the system's end-to-end impulse response, which can then be utilized by a statistical analysis engine to derive eye statistics for the channel.

The second level of modeling (referred to as "Getwave") processes continuous time-domain waveform data. The flow is essentially the same, except a time-domain waveform is passed to the model instead of an impulse response. The model applies its respective equalization function and passes the modified waveform back to the time domain waveform. TX and RX models can be chained together as before. RX models can represent the behavior of the clock recovery loop and pass sampling clock information back to the simulator, which uses the clock information to assess the link's BER.

Two modes of simulation are therefore supported by the IBIS-AMI standard: statistical analysis and time domain simulation. Each mode has its advantages and disadvantages. Statistical analysis makes the assumption that TX/RX equalization is both linear and time invariant. Statistical analysis is quite fast and well suited to exploring a large design space. Time domain simulation allows nonlinear and/or time-varying effects in the TX or RX IP to be represented, and it supports detailed modeling of the clock recovery loop. Time-domain analysis is well suited to detailed analysis of specific stimulus patterns or conditions.

Statistical Analysis

The primary advantage of statistical analysis is that it quickly calculates the average effect of all possible messages of a given length, thus providing comprehensive coverage of all possible conditions. Here are the basic steps:

1. Statistical analysis uses the channel's end to end impulse response to directly compute eye statistics.
2. Convolution techniques are used to generate eye statistics, bathtub curves are derived from that data, and the BER is estimated by combining bathtub curves with estimated recovered clock behavior.
3. Peak distortion analysis can also be used to determine the worst case data pattern and its corresponding eye closure.

For example, a 32-bit message length represents all 2^{32} possible combinations of a 32-bit sequence. Running a time-domain analysis to cover the same set of conditions would require running a 32×2^{32} bit analysis (1.374×10^{11} bits), which is clearly

impractical. Statistical analysis can therefore be used to quickly and efficiently identify conditions that merit further study in the time domain.

Figure 1 shows typical output from a statistical analysis run. The eye diagram is color-coded to show the probabilities associated with the signal's location within the eye. Data bathtub curves are superimposed in black, while the Probability Density Function (PDF) of the recovered clock is shown in the center of the eye in blue. Since the data and clock distributions in this example have very little overlap, the probability of error is small (the predicted BER was less than $1x10^{-20}$).

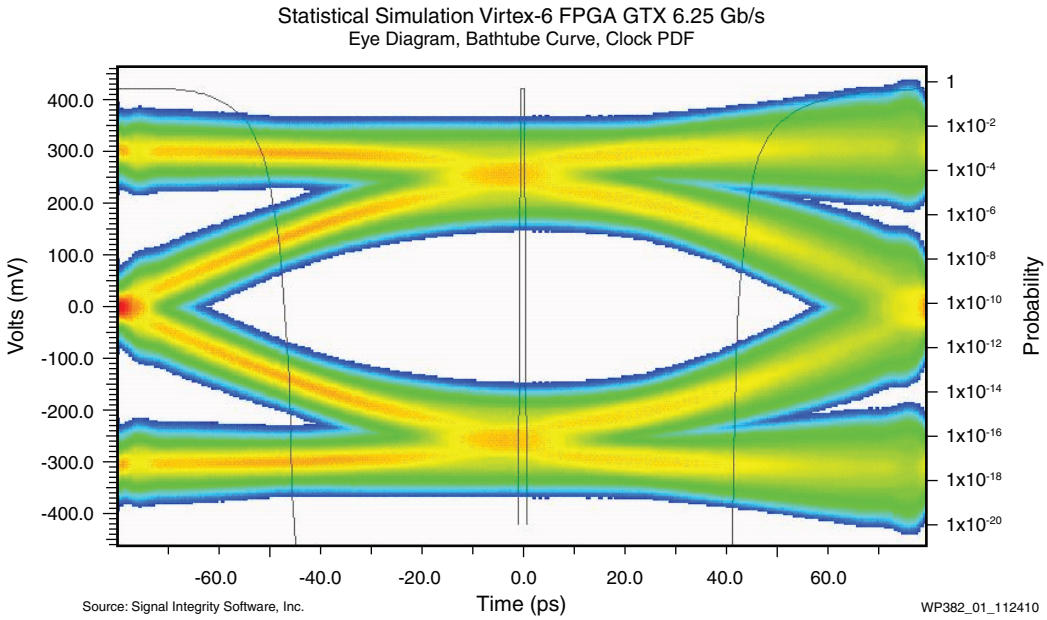


Figure 1: **Statistical Simulation Eye Diagram: Virtex-6 FPGA GTX at 6.25 Gb/s**

Time Domain Simulation

Time-domain simulation allows the channel's behavior to be examined in more detail. The effects of specific stimulus sequences can be explored as well as the effects of non-linear and/or time-varying equalization.

IBIS-AMI time-domain analysis is quite fast compared to traditional SPICE time-domain analysis, typically providing simulation speeds of about 1,000,000 bits per minute. This makes it practical to study how long an adaptive equalizer takes to reach steady state, or how different maximum run lengths might affect DFE taps and the behavior of the clock recovery loop. The high performance of IBIS-AMI time-domain analysis means data and results management becomes quite important. It is easy to generate a million bits worth of simulation results, but it is not practical to review a million bits worth of waveforms. The key to success is having the computer compile waveform statistics directly from the simulation output and selectively choose which data is saved for later viewing. This provides the data necessary to correctly compute BER without overwhelming the user with unnecessary detail. The compiled statistics also ensure that detailed behaviors can be recorded and examined when required.

Figure 2 and Figure 3 show typical time domain eye diagram and waveform data output from a time-domain simulation. This is the traditional way of presenting time-domain simulation output. These plots represent 50,000 bits saved from a 500,000-bit simulation. The waveform is plotted at the output of an RX DFE equalizer and shows the expected horizontal asymmetry for this equalization scheme. The

sample size, 50,000 bits, is actually too small to derive meaningful statistics for the channel's behavior, but it is still clear that detail is being lost in the eye diagram.

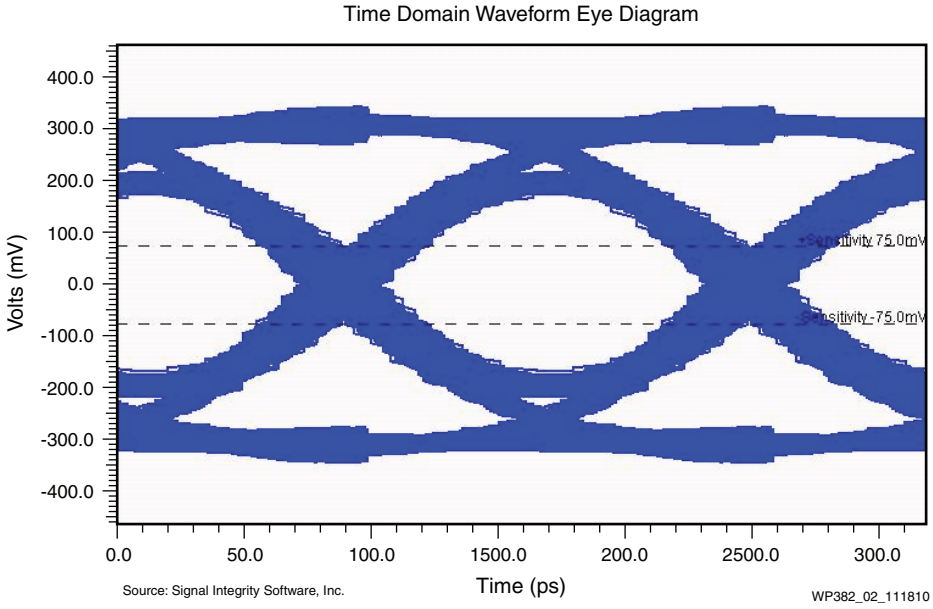


Figure 2: **Time Domain Simulation Eye Diagram: Virtex-6 FPGA GTX at 6.25 Gb/s**

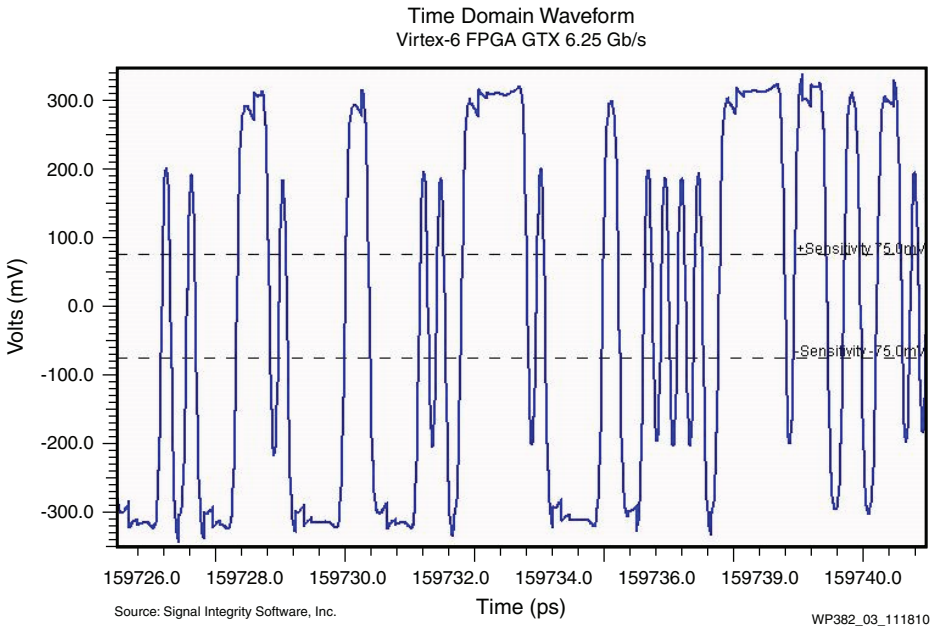


Figure 3: **Time Domain Waveform: Virtex-6 FPGA GTX at 6.25 Gb/s**

A better way to deal with time-domain results is to have the simulator compile statistics on-the-fly and present a summary of the complete run to the user. In Figure 4, the waveform statistics are shown for a 500,000-bit simulation. Since this simulation involved an adaptive DFE that was initialized with its taps set to zero, the model was given 250,000 bits to initialize before the data collection process started.

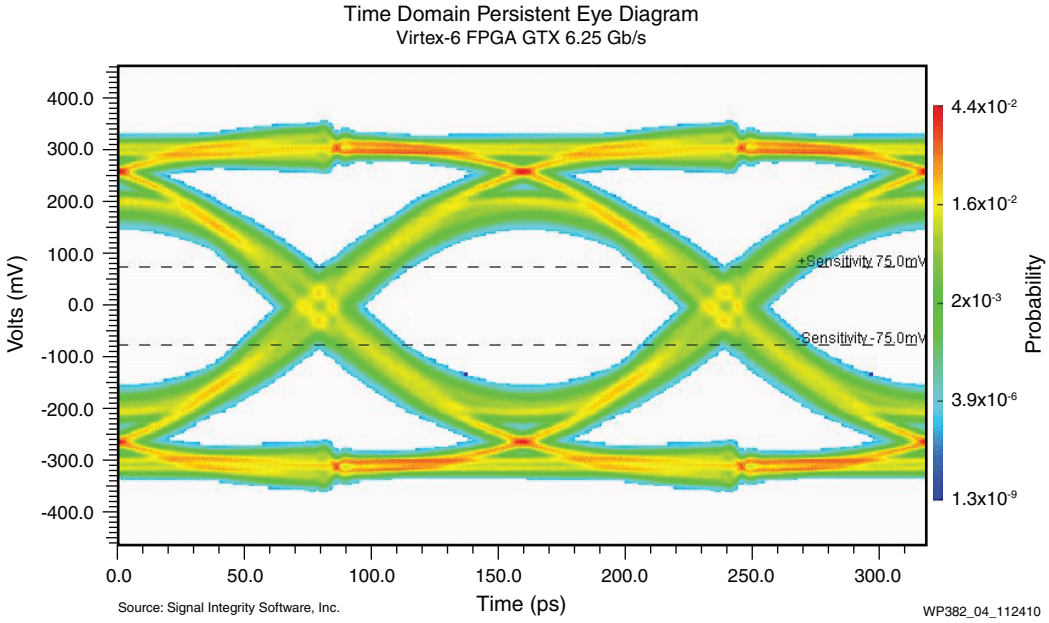


Figure 4: **Time Persistent Eye Showing Statistics: Virtex-6 FPGA GTX at 6.25 Gb/s**

Xilinx IBIS-AMI Models

Xilinx currently provides IBIS-AMI models and design kits for SiSoft's QCD for the following devices:

Table 1: **Availability of SiSoft IBIS-AMI Models and Kits**

Virtex®-5 FPGA GTX	Available
Virtex-5 FPGA GTP	Available
Virtex-6 FPGA GTX	Available
Virtex-6 FPGA GTH	Planned
Spartan®-6 FPGA GTP	Available

Virtex-6 FPGA GTX IBIS-AMI Model

An IBIS-AMI model consists of three parts:

- Analog model (`.ibs`) — `xilinx_v6_gtx.ibs`
- Model interface control file (`.ami`) — `V6_GTX_AMI_Tx.ami`
- Executable model (`.dll` or `.so`) — `V6_GTX_AMI_Tx.dll`

The Xilinx IBIS-AMI models are fully compliant with the IBIS 5.0 specification. Additionally, the Xilinx model supports the proposed features of BIRDs 121–124 for IBIS 5.1. This means the models are fully supported in any simulator that is compliant with IBIS 5.0 as well as any simulator that takes advantage of the proposed IBIS enhancements for IBIS 5.1.

The IBIS Model

The IBIS component of the IBIS-AMI model (`xilinx_v6_gtx.ibs`) is in the format of a standard IBIS model that defines the IBIS component, I/O models for the component signals, analog parameters of the I/O, etc., and uses the `.ami` feature. A partial example from the Xilinx Virtex-6 FPGA GTX `.ibs` file is shown here:

```
|
[Component]        v6_gtx_serdes
[Manufacturer]     Xilinx
|
[Package]
|
R_pkg      .001   NA     NA
L_pkg      1p     NA     NA
C_pkg      1f     NA     NA
|
[Pin]    signal_name          model_name           R_pin    L_pin    C_pin
1p       v6_gtx_tx_p          v6_gtx_ami_tx        NA       NA       NA
1n       v6_gtx_tx_n          v6_gtx_ami_tx        NA       NA       NA
2p       v6_gtx_rx_p          v6_gtx_ami_rx        NA       NA       NA
2n       v6_gtx_rx_n          v6_gtx_ami_rx        NA       NA       NA

[Diff_Pin]    inv_pin    vdiff    tdelay_typ    tdelay_min    tdelay_max
1p            1n         0.1V     NA            NA            NA
2p            2n         0.1V     NA            NA            NA

[Model Selector]          v6_gtx_ami_tx
v6_gtx_ami_tx_opal        Opal
v6_gtx_ami_tx_130mv       0
v6_gtx_ami_tx_210mv       1
v6_gtx_ami_tx_290mv       2
.
.
.
[Algorithmic Model]
Executable Windows_VisualStudio7.1.3088_32 v6_gtx_ami_rx.dll
v6_gtx_ami_rx.ami
Executable Linux_gcc3.2.3_32 v6_gtx_ami_rx.linux.so v6_gtx_ami_rx.ami
[End Algorithmic Model]
|
.
.
.
```

The `.ami` file is the control interface to the executable model. It defines what model options are contained in the executable as well as the control interface (for example, Tx amplitude, pre- and post-cursor controls, and Rx controls). A partial example from the Xilinx Virtex-6 FPGA GTX Tx `.ami` file is shown here:

```
(V6_GTX_AMI_Tx
  (Description "Xilinx transmitter model provided by SiSoft")

  (Reserved_Parameters
    (Ignore_Bits (Usage Info) (Type Integer) (Default 2)
      (Description "Ignore two bits to fill up tapped delay line."))
    (Max_Init_Aggressors (Usage Info) (Type Integer) (Default 25)
      (Description "Number of aggressors is actually unlimited."))
    (Init_Returns_Impulse (Usage Info) (Type Boolean) (Default True)
      (Description "Both impulse and parameters_out returned."))
    (GetWave_Exists (Usage Info) (Type Boolean) (Default True)
      (Description "GetWave is well and truly provided in the module."))
  ) | End Reserved_Parameters

  (Model_Specific
    (A_TXDIFFCTRL (Usage In)(List 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)
      (Type Integer) (Default 12)
.
.
.
```

Simulating Backplane Systems with Virtex-6 Family GTX IBIS-AMI Models and SiSoft Quantum Channel Designer

The key benefit to the system designer is the ability to perform extensive pre-hardware simulation and design—in effect, to build a virtual prototype system that can optimize performance and provide understanding of design tradeoffs and margins. The basic backplane system shown in Figure 5 is used to illustrate the design process. This example is taken from "A Design of Experiments for Gigabit Serial Backplane Channels" by Jack Carrel et al [Ref 1].

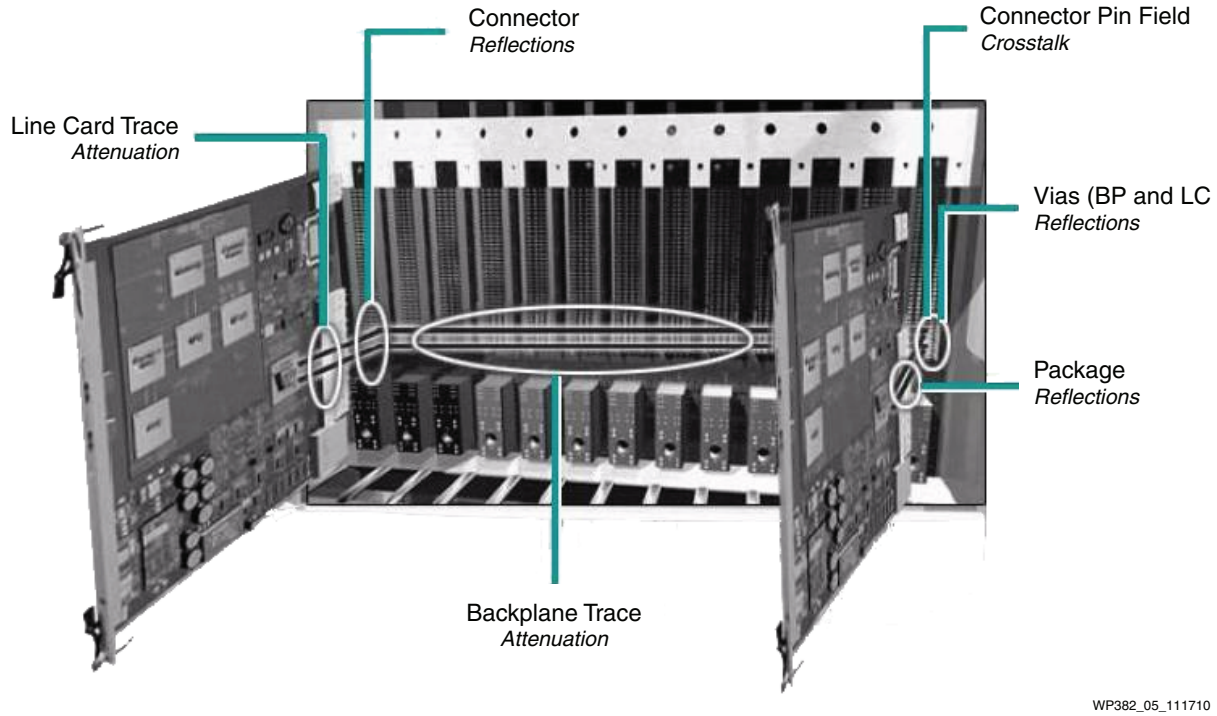
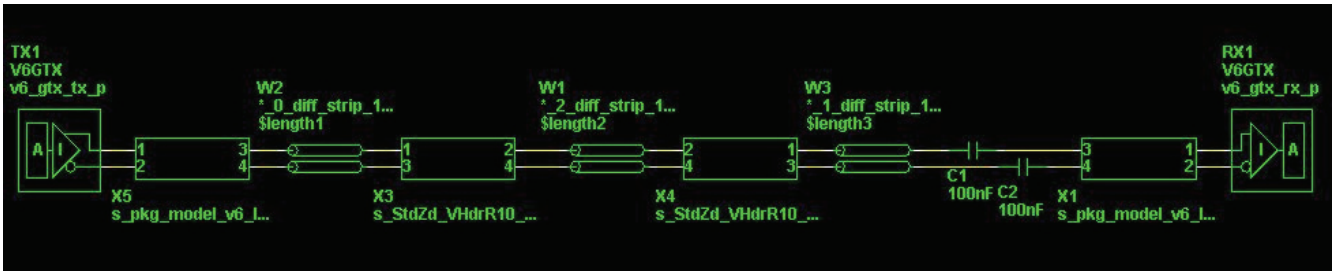


Figure 5: **Example Backplane, Virtex-6 FPGA Driving Virtex-6 FPGA**

Schematically, this can be shown (as illustrated in Figure 6) using Virtex-6 FPGA transmitter and receiver IBIS-AMI models, associated device package models

(s-parameters), and transmission line and connector models for the backplane system. Because the intent is to optimize the design and make design tradeoffs up front, vendor-supplied model data can easily be used as well as model data for channel interconnect structures such as transmission lines, vias, etc. Existing measured s-parameter data obtained from a vector network analyzer or derived from simulation can also be used.

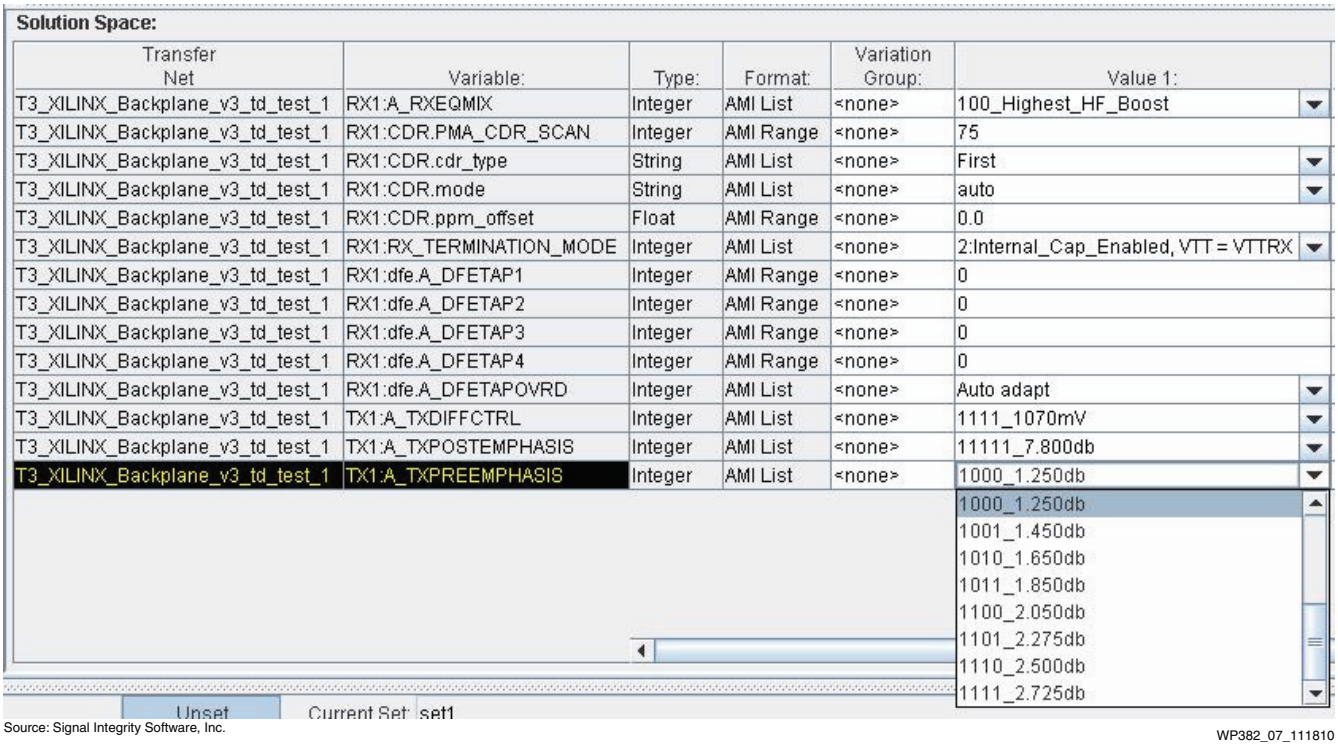


Source: Signal Integrity Software, Inc.

WP382_06_111810

Figure 6: **Schematic Drawing of Backplane Channel Simulation**

The Virtex-6 FPGA GTX high-speed SerDes cores have a number of advanced features for performance optimization in high-speed serial channels. The transmitter has full amplitude control and pre- and post-cursor controls for implementing pre- and post-emphasis. The receiver has an advanced continuous real-time linear equalizer and an adaptive decision feedback equalizer (DFE) to compensate for channel loss and impairments. All of the controls for these features are available through the IBIS-AMI model and are user controllable directly through the Quantum Channel Designer user interface. Figure 7 shows the user interface through the Solution Space pane of Quantum Channel Designer.



Source: Signal Integrity Software, Inc.

WP382_07_111810

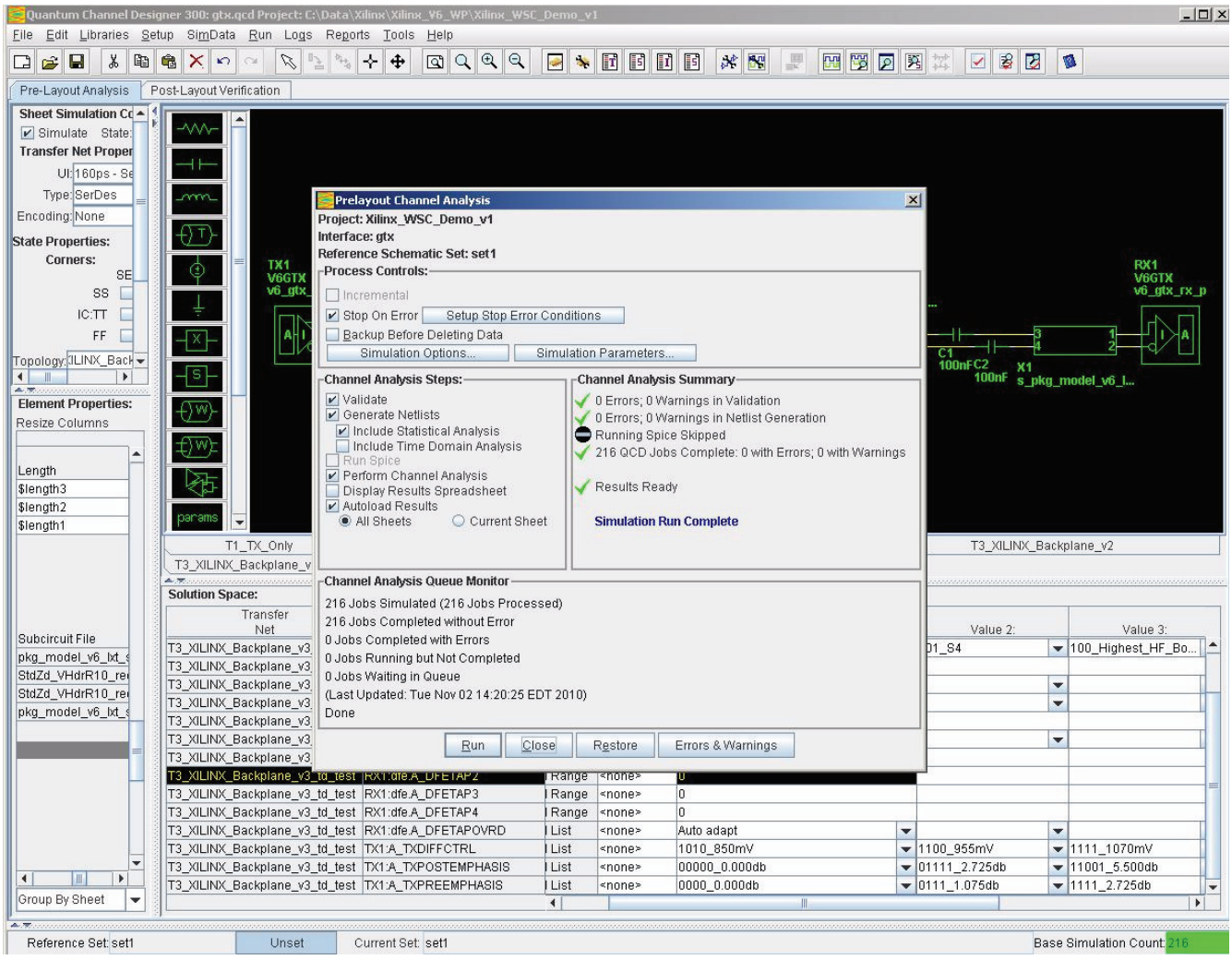
Figure 7: **User Parameter Controls**

Given the number of device parameters, it is possible to easily generate an extremely large number of simulation conditions. For example:

- Transmitter settings = (16 amplitude) x (16 pre-cursor) x (32 post-cursor) = 8,192
- Receiver settings = (8 EQ) x (2 DFE on/off) = 16
- Total (Tx) x (Rx) = 131,072

This is without even considering the number of manual tap settings for each of the four DFE taps or channel parameter values that can be optimized as a variable. Even with fast statistical analysis capabilities, care must be taken to choose a reasonable solution space for simulation.

In this example, a 40-inch total length backplane (consisting of a 3-inch trace on each line card and a 34-inch backplane trace) is examined. The design schematic is drawn using the Quantum Channel Design kit containing the Virtex-6 FPGA GTX transceiver and package models, lossy transmission line models for the trace lengths on both line cards and the backplane, and backplane connector s-parameter models obtained from the connector vendor. The design simulations are configured to sweep a limited number of Tx and Rx parameters to investigate performance at 6.25 Gb/s. Because it is known that the lower values of each setting would be of no use, these values are ignored and the set of simulations required is thus constrained. In this example, 216 simulations are configured and run, as shown in Figure 8.



Source: Signal Integrity Software, Inc.

WP382_08_111810

Figure 8: **User Parameter Controls**

Run and post-processing time for the 216 simulations is approximately 3 minutes, utilizing both cores of an Intel Core 2 Duo 2.4 GHz P8600, with each statistical simulation taking between 1–2 seconds. For larger runs, Simulation Farms can be used to further improve simulation turnaround time.

A large number of simulations also means there will be a large amount of generated data that needs to be analyzed. QCD post-processes the simulation results to derive and present key metrics. The resulting data can be easily sorted by BER performance, eye opening characteristics, and a number of other metrics. A subset of the data from the simulation results is shown in Figure 9.

Row	ID	Transfer Net	State	Transfer		Symbol Rat...	$LENGT...	Stat BER	Stat Eye Hei
4	112	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.16641
5	113	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.175539
6	114	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.171384
7	115	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0.0364139	0.139023
8	116	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0.0114917	0.145764
9	117	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0.21108	0.13616
10	118	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0.249944	0.113079
11	119	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0.250019	0.119388
12	120	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0.563853	0.103793
13	121	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	6.42198e-029	0.176947
14	122	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.196128
15	123	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.219313
16	124	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.164597
17	125	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.199269
18	126	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.195282
19	127	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.161788
20	128	T3_XILINX_Backplane...	default	TX1_to_RX1		6.25	34	0	0.164433

Source: Signal Integrity Software, Inc.

WP382_09_111810

Figure 9: **Simulation Data Sorted for 34-Inch Backplane plus Two 3-Inch Line Cards**

A sampling of the simulated eye diagrams can also be analyzed, as shown in Figure 10 through Figure 13.

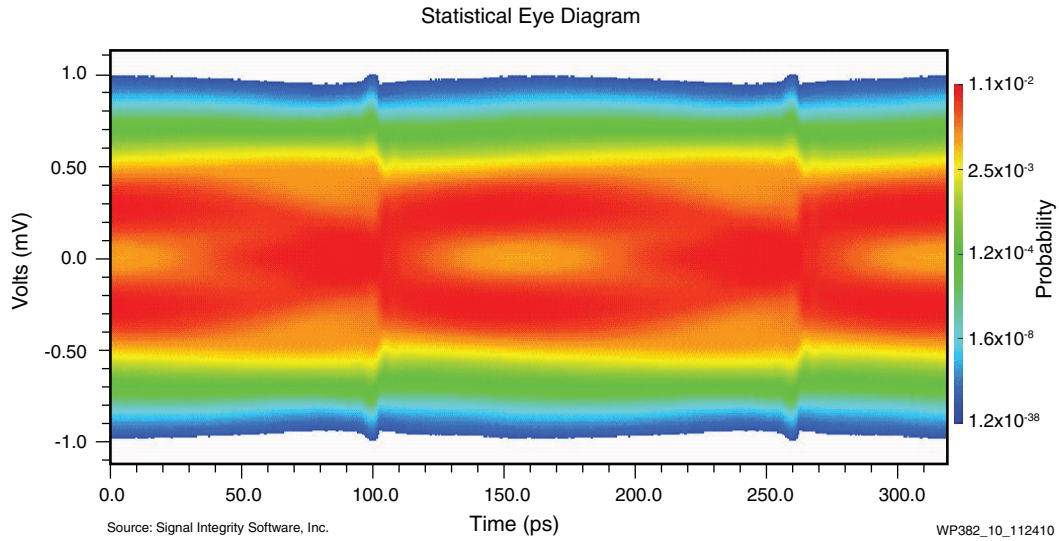


Figure 10: **Unoptimized Settings**

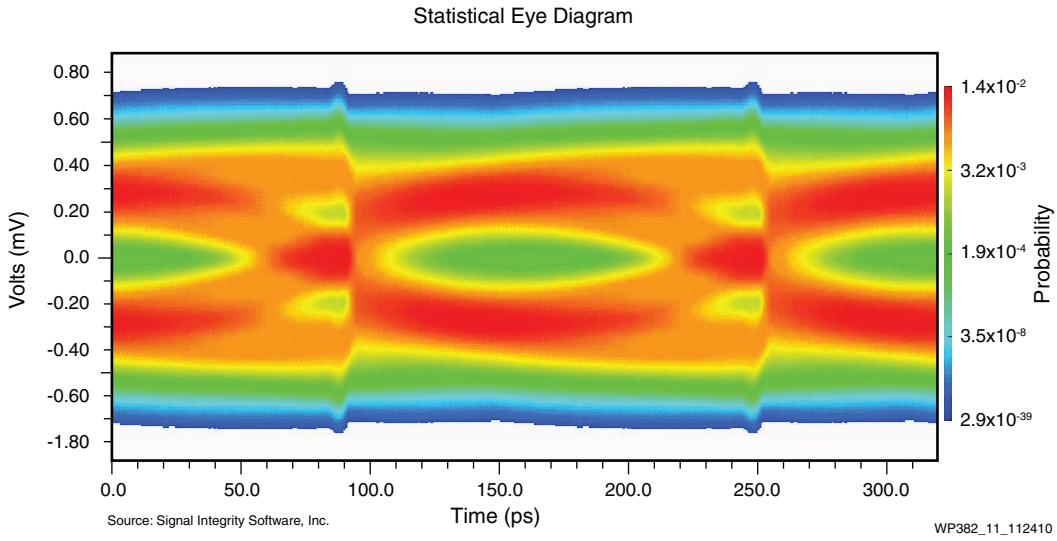


Figure 11: **Increasing Pre-emphasis**

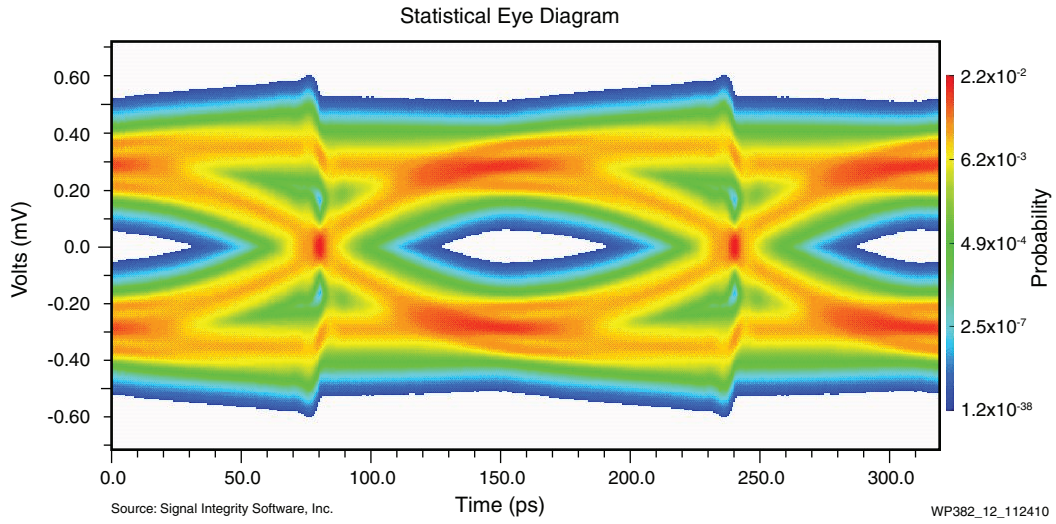


Figure 12: **Further Increasing Pre-emphasis**

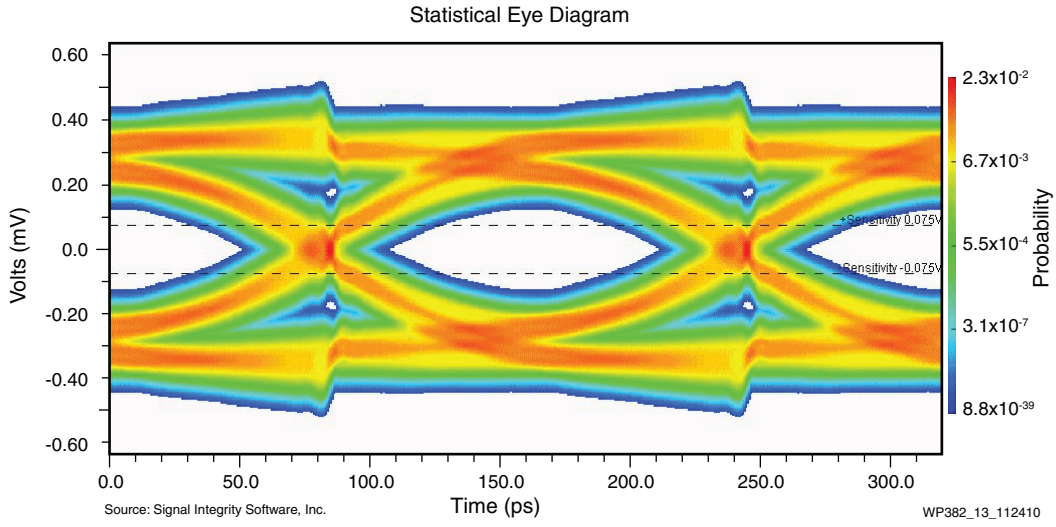
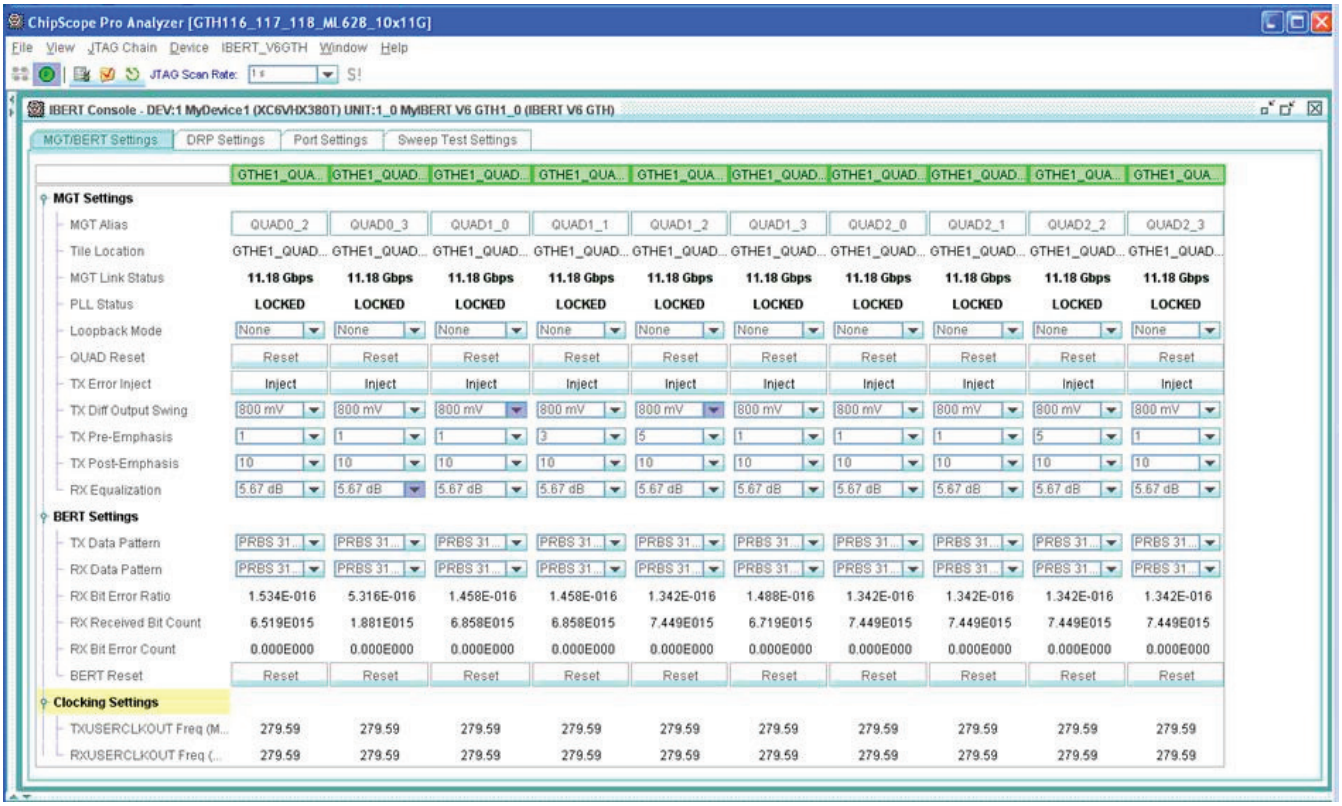


Figure 13: **Optimized Pre-emphasis**

Simulation and Hardware Optimization — IBERT

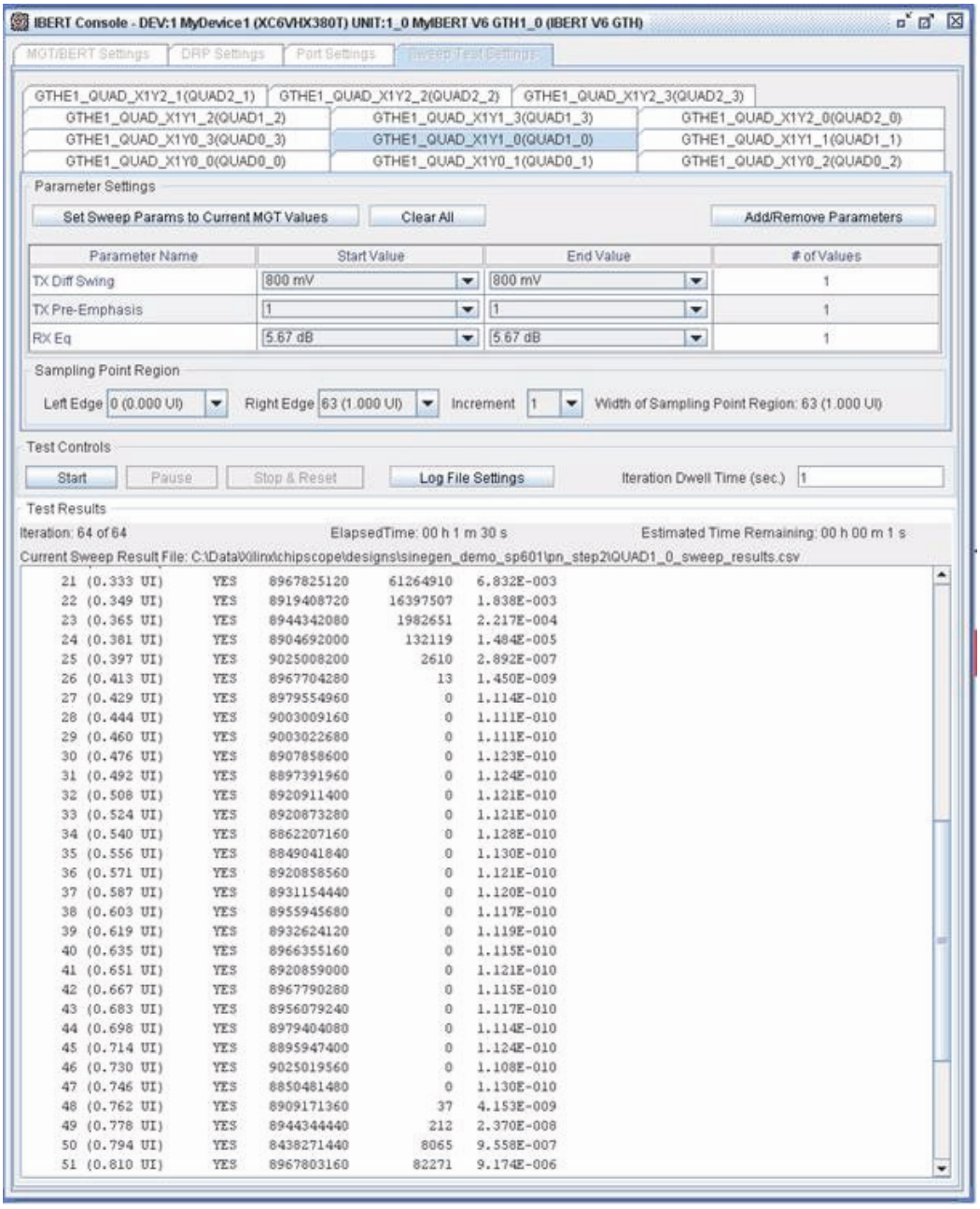
Xilinx provides a design optimization core called an Integrated Bit Error Rate Tester (IBERT) that implements a BER tester in FPGA hardware for hardware system test. IBERT is designed for Xilinx SerDes evaluation and test. All major features of the SerDes are supported and controllable in hardware through IBERT, including Tx differential swing, pre-emphasis, post-emphasis, Rx linear equalization, and DFE and PLL divider settings. Numerous test patterns (generated internally to the FPGA) provide for selectable evaluation of such entities as clock patterns, PRBS 7, 15, 23, 31, and encoded patterns.

The test interface is very similar to the modeling and simulation interface defined in the IBIS-AMI models and made visible through the simulator control panel. This allows for easy configuration of hardware to optimum settings derived from simulation. A sample of the IBERT user interface is shown in Figure 14 along with its output, shown in Figure 15. These are compared to the simulation control interface in Figure 16. Through simulation, near-optimum settings can be derived and then verified in hardware for much faster hardware bring-up and verification.



WP382_14_120310

Figure 14: **IBERT Control Panel**



WP382_15_111110

Figure 15: **IBERT Output — Error Count and BER**

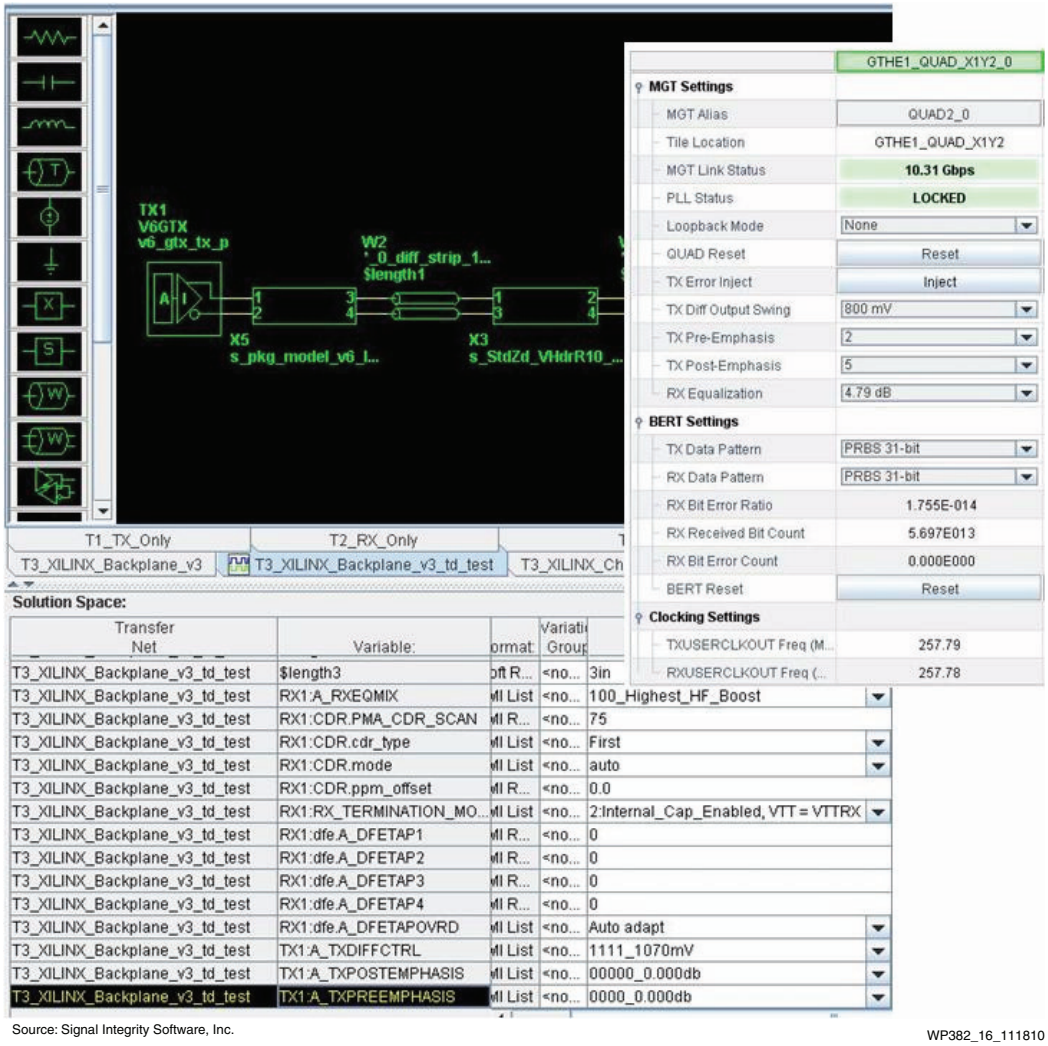


Figure 16: **IBIS-AMI Controls Compared to IBERT Controls**

Summary

Xilinx continues to lead the industry in supporting open standards and providing IBIS-AMI models for high-speed devices. Xilinx is an active member of the IBIS Advanced Technology group and works closely with leading industry vendors such as Signal Integrity Software, Inc. for analysis software and model development. The IBIS-AMI standard enables fast, accurate modeling of high-speed SerDes channels. It provides the interoperability and flexibility required to create statistical and time-domain models. It allows SerDes vendors such as Xilinx to supply models of their devices that run in commercial EDA tools while retaining control of proprietary silicon technology. Xilinx enables customers to simulate, analyze, and verify in hardware high-speed system parameters to provide robust and cost-effective design solutions.

Currently released IBIS-AMI models can be downloaded from the Xilinx Support download website [Ref 2]. Current design kits for SiSoft Quantum Channel Designer can be downloaded from the SiSoft ELearning website [Ref 3].

References

1. DesignCon 2008, A Design of Experiments for Gigabit Serial Backplane Channels by Jack Carrel, Bill Dempsey, and Mike Resso: http://www.google.com/url?q=http://www.iconnect007.net/pdf/DesignCon_08_Resso.pdf&sa=U&ei=YZbtTLL7G4H48AaHjqFu&ved=0CAwQFjAA&usg=AFQjCNFzcW56O2hhIOYC7cgrL2LMvVvF-Q
2. Xilinx Support download website: http://www.xilinx.com/support/download/index.htm
3. SiSoft ELearning website: http://www.sisoft.com/elearning_qcdkits.asp

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
12/09/10	1.0	Initial Xilinx release.

Notice of Disclaimer

The information disclosed to you hereunder (the "Information") is provided "AS-IS" with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.