A Significant Technology Advancement in High-Speed Link Modeling and Simulation

WP-01212-1.0 White Paper

As high-speed I/O (HSIO) and serial link data rates keep increasing, the requirements for accuracy and advanced simulation and modeling techniques get more stringent. Emerging requirements, such as the ability to model process, voltage, and temperature (PVT) variations at deep submicron process nodes or smaller, to fully account for all the circuit blocks of the link, and to close the gap between modeling and measurements, have become critical. However, conventional simulation and modeling methods cannot meet most, if not all, of the requirements. This paper reviews the techniques used in recent HSIO simulation and modeling, such as statistical behavioral, SPICE, and IBIS-AMI models, outlining the areas where they fall short in comparison with the emerging requirements. This paper also introduces Altera's JNEye transceiver link simulation tool and discusses how the tool enhances HSIO link modeling and simulation. This paper includes simulation and experimental results that demonstrate how the JNEye tool can meet the emerging requirements.

Introduction

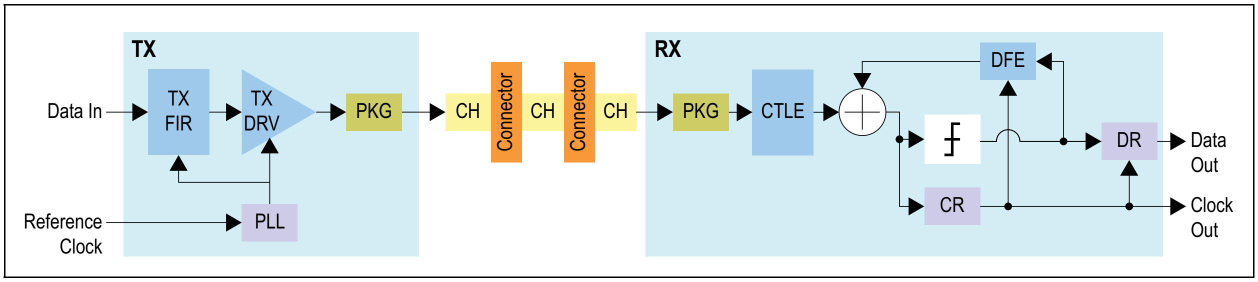
HSIO and serial link speed double every two to three years on average [(1)] to meet the ever increasing bandwidth demands for network and computer systems. In addition to bandwidth benefits, speed doubling also enables density and throughput to double for a device or board. Using PCI Express® (PCIe®) as an example, its speed has increased from 2.5 Gbps (Gen1), to 5.0 Gbps (Gen2), to 8 Gbps (Gen3), and to 16 Gbps (Gen4). Another example is the Optical Internetworking Forum (OIF) Common Electrical I/O (CEI) implementation agreement, which states its Gen1 speed at 6.5 Gbps, Gen2 at 11.3 Gbps, Gen3 at 28 Gbps, and Gen4 at 56 Gbps. At 28 Gbps, the unit interval (UI) is only 35.7 ps for a non-return-to-zero (NRZ) modulated signal, and this timing budget has to be shared by the transmitter (TX), receiver (RX), and channel (CH). This means that the TX and CH will get approximately 30 percent of the UI each, and the RX will get approximately 40 percent [(2)]. When all the timing and jitter impairments are considered, meeting a UI of 35.7 ps at 28 Gbps is challenging. It gets even more challenging at 56 Gbps, where the UI is 17.9 ps. Therefore, accurate and capable simulation and modeling techniques are critical for designing HSIO circuits, devices, and links.

As the data rate increases, the intersymbol interference (ISI) gets worse due to the lossy characteristics of the copper channel. To mitigate the ISI at higher data rates, various equalization circuits have been developed, including TX finite impulse response (FIR), RX continuous linear equalizer (CTLE), and RX decision-feedback equalizer (DFE). Figure 1 shows a typical high-speed link and its subcomponents at 10 Gbps and above for a backplane, typically with an insertion loss of 25 dB or higher, and consisting of PCBs, connectors, and vias. In addition to the equalization circuit blocks, a clock generation (CG) circuit via a phase-locked loop (PLL) for the TX, and a clock recovery (CR) circuit via a PLL for the RX are also shown. A 3-tap to 4-tap FIR

101 Innovation Drive
San Jose, CA 95134
www.altera.com

filter, a fourth-order to eighth-order CTLE, and a 5-tap or higher DFE are commonly needed for high-speed backplane links [3], implying a complex equalization solution space.

Figure 1. A High-Speed Link System



Existing Simulators and Their Limitations

Transistor-level-based circuit simulators, such as SPICE or HSPICE, have not been effective for high-speed links due to their long simulation times. For example, a SPICE simulator may take many hours to simulate a primitive link that consists of only a TX driver, RX buffer, and copper channel for a few hundred bits. If various equalizations and clock generation and clock recovery processes are modeled, the SPICE simulation time would be highly prohibitive. The performance merit of the high-speed link is commonly defined by its bit error rate (BER) which is set at 10-12 or 10-15 by most standards [2]. The performance gap between what a practical SPICE simulator can deliver and the standard requirement is huge.

To overcome the limitations of the SPICE simulator, an analytical worst-case channel eye estimation method, called peak-distortion analysis (PDA), was developed [4]. PDA treats a copper channel as a linear-time invariant (LTI) system, and the analysis is based on single-bit-response (SBR) and associated sampling cursors. While the PDA method helps to alleviate the long simulation time challenge of SPICE, it tends to give overly pessimistic results compared with reality. As such, statistical link simulation methods were developed in the mid 2000s [5] [6]. The statistical link simulation methods build eye diagrams by superpositioning time-shifted SBRs in a probabilistic manner equivalent to the convolution of SBR cursor probability density functions (PDFs) statistically. Statistical link simulation methods can simulate a high-speed link with relatively fast throughput. For this reason, the IBIS-AMI standard [7] adopted the LTI and statistical link simulation methods for high-speed link simulation.

However, a major limitation of the statistical link simulation method is its limited capability in handling jitter and noise. Although improvements have been made to incorporate jitter and noise in statistical link simulation methods [8] [9], those approaches often involve assumptions of noise to jitter conversion that do not apply when the two are independent. While the LTI and statistical link simulation methods are appropriate for copper channels, they introduce inaccuracy, or even error, due to the non-linearities of the TX driver and FIR circuit and the RX buffer and DFE that are often overlooked.

In recent years, system designers and signal integrity engineers often demand a good correlation proof between simulation and measurement before they can adopt a simulation methodology for their link design validation and large sample pre-production simulations. In order to address these requirements, a high-speed link simulator needs to model the device and channel PVT variations. Moreover, the number of possible equalization parameters for a link with FIR, CTLE, and DFE circuits can be a few millions; yet, a time-efficient and consistent optimization method is expected. However, correlation, PVT, and equalization optimization are neither addressed nor lightly touched upon in most of the statistical link simulation methods published.

Statistical Link Simulation Methods: An Overview

In general, statistical link simulation methods treat each circuit block with an equivalent or approximated higher level behavioral model that can be represented mathematically. Figure 2 shows a behavioral block diagram for a statistical link simulation method and the associated mathematical representations corresponding to the functional block diagrams in Figure 2.

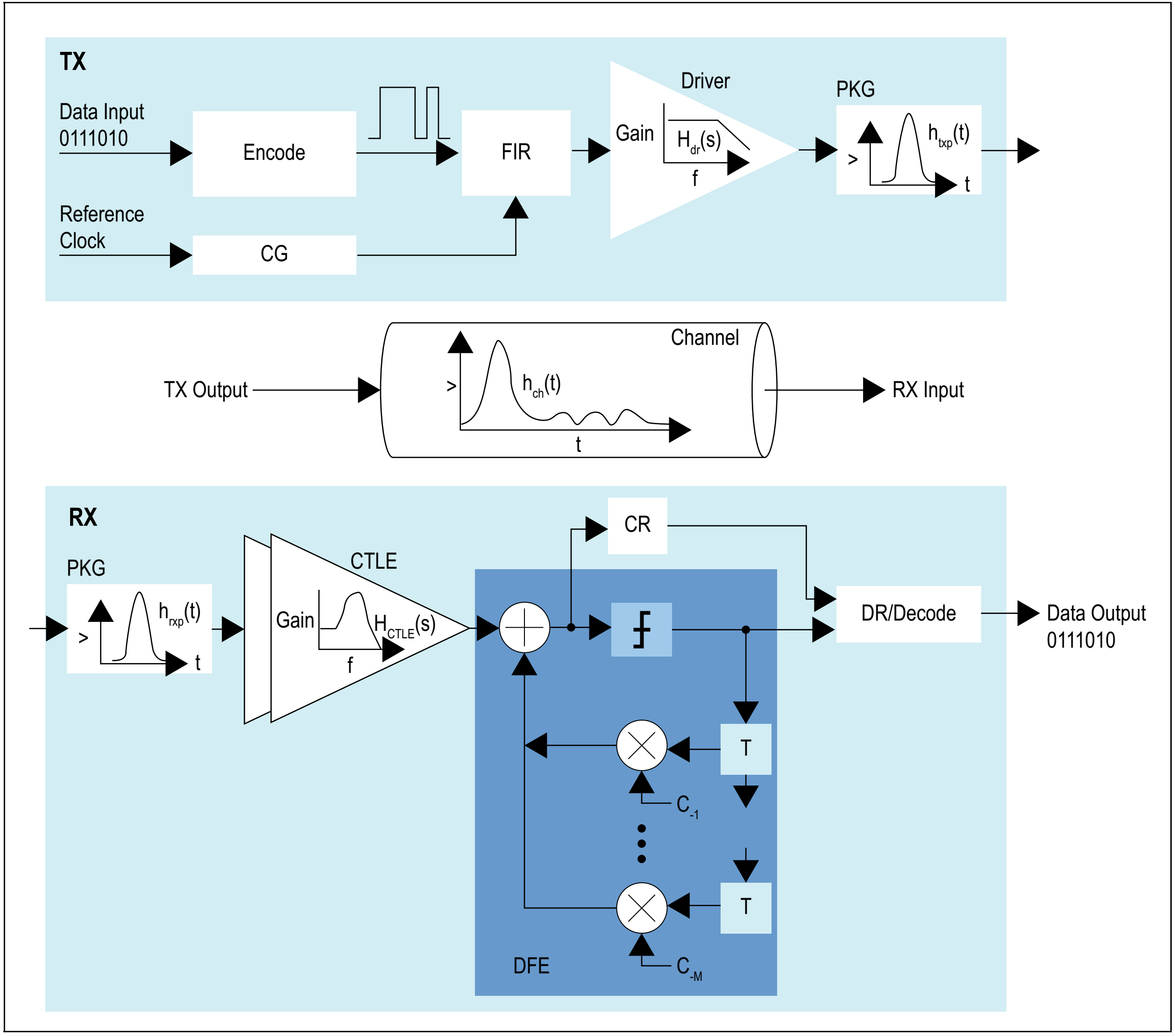
The LTI function is often used for cascading link component blocks and calculating the frequency-domain transfer function (TF) or time-domain impulse response (IR) from point to point. The overall IR from the TX FIR filter, to the RX CTLE, with the TX driver, TX package, CH, RX package, and RX buffer in between, can be calculated via a convolution chain process shown in Equation 1:

Equation 1.

$$h_{sl}(t) = h_{ffe}(t) * h_{dr}(t) * h_{txp}(t) * h_{ch}(t) * h_{rxp}(t) * h_{ctle}(t)$$

where * denotes convolution.

Figure 2. Statistical Modeling for a High-Speed Link



The SBR can be calculated via convolving the IR with a single-bit ideal square wave *S(t)* as shown in Equation 2:

Equation 2.

$$\prod_{s1}(t) = S(t) * h_{sl}(t)$$

The SBR Π_{s1} obtained in Equation 2 does not include the DFE as it does not follow the LTI rule. However, the DFE may be approximately modeled by subtracting the DFE tap weights from the SBR Π_{s1} value as shown in Equation 3:

Equation 3.

$$\prod_{s2}(t) \approx \prod_{s1}(t) - \sum_{i-1}^{N} c_i comb(t - iT)$$

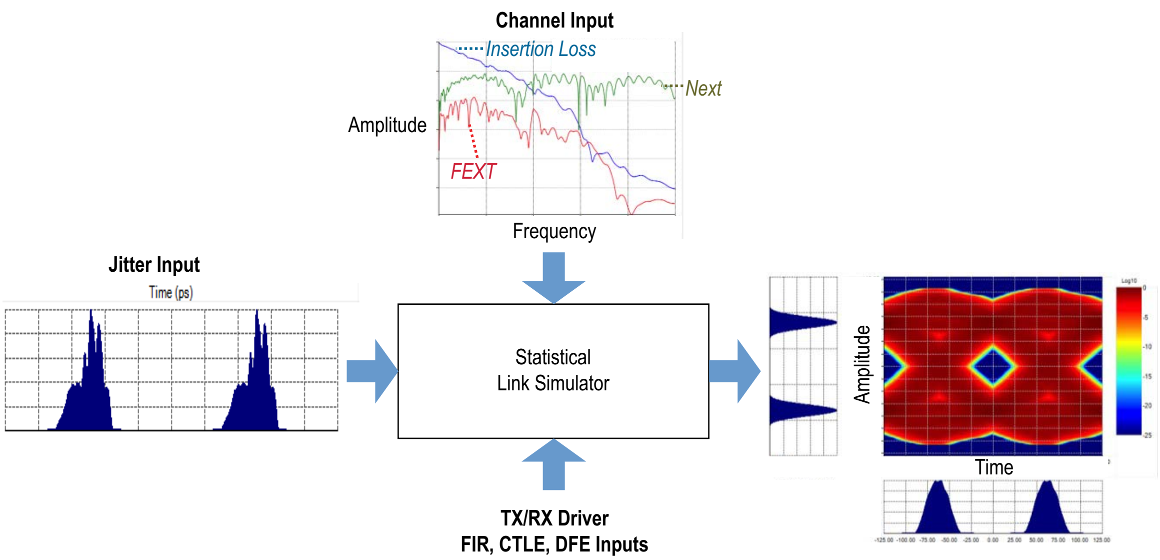
where c_i are the DFE tap coefficients, T is the UI, and *comb(t)* is the comb function.

With the SBR Π_{s2} value, the statistical eye can be constructed by time-shifting Π_{s2} and probabilistic superposition (4) (5). We denote the associated statistical eye PDF as $p_0(t, v)$. The BER cumulative density function (CDF) and "bath-tub" curves can be derived from $p_0(t, v)$ (10).

Statistical link simulation methods may be extended to comprehend the jitter and noise effects, and this is often achieved by convolving the assumed jitter and noise PDFs with the statistical eye PDF, $p_0(t,v)$.

Figure 3 shows an illustrative example of the statistical simulation results for a PCIe Gen3 backplane link running at 8 Gbps, with the TX having an FIR filter and the RX having a CTLE and DFE. The simulation takes about 1.8 minutes to complete using a medium-range personal computer[1].

Figure 3. A Statistical Simulation Example for PCIe Gen3



[1] A personal computer equipped with Intel Core i7 2.7GHz processor, 8GB RAM, and 64 bit Microsoft Windows 7 operating system.

While the statistical link simulation method offers a computationally effective way to estimate the link statistical eye, it is also subject to the following limitations:

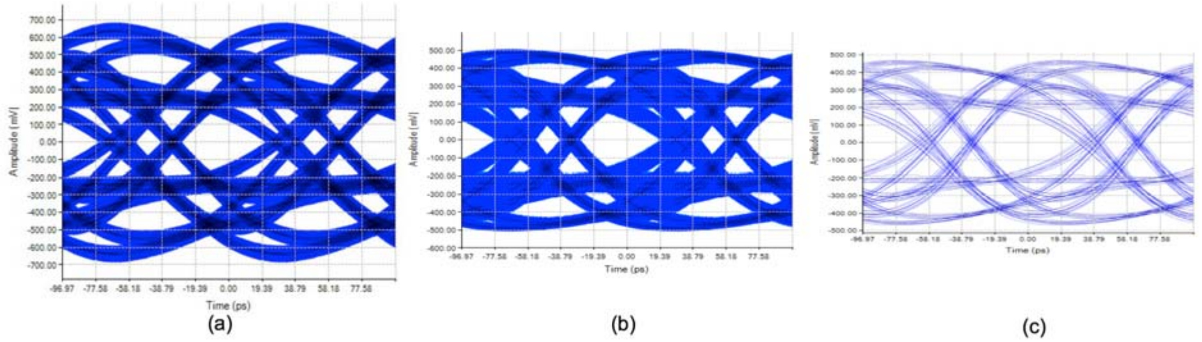
- Hard to build in equalization adaptation, limiting its optimal solution search capability
- Difficult to model interactions of various jitter and noise components from the TX with CH and RX, limiting its accuracy and coverage
- The CG, CR, and data recovery (DR) circuits cannot be accounted for due to their time-domain operation nature, limiting its accuracy and coverage

Although the traditional statistical link simulation method has many drawbacks, this method is useful in developing the high-speed link flow for those who need its speed advantage and are aware of the speed-accuracy tradeoffs. To continue to leverage the advantages of traditional statistical link simulation methods while reducing its limitations as much as possible, the JNEye tool implemented the following enhancements in its statistical link simulation flow:

- Transmitter nonlinear behavioral modeling
- Receiver CTLE gain compression modeling
- Receiver DFE summing node parasitic modeling

These enhancements narrow the gaps between the JNEye tool's statistical link simulation and time-domain full-waveform simulation methods in certain link configurations and conditions. Figure 4 shows an example of the JNEye tool's statistical link simulation enhancement.

Figure 4. Statistical Modeling Improvement in the JNEye Tool



No CTLE Gain Compression Adjustment

JNEye's Statistical Link Simulation Method with CTLE Gain Compression Adjustment

CTLE Output from Time-Domain Full-Waveform Simulation

In Figure 4(b), the "envelope" is reduced for the eye with gain compression adjustment compared with the eye in Figure 4(a) that does not have gain compression adjustment. Even with gain compression, the eye predicted from the statistical link simulation method does not match perfectly with what is generated from the full-waveform in Figure 4(c), suggesting that new methods and advancements are still needed for high-speed simulations beyond what statistical link simulation methods can offer.

JNEye's Advanced Simulation and Modeling Methods

This section discusses the JNEye tool's time-domain full-waveform and hybrid simulation methods, and the associated CG, CR, DR modeling methods that are not possible in statistical link simulation methods. In addition, fast and computationally efficient optimization methods will be discussed. This section also examines how channel modeling is conducted in the JNEye tool and highlights a novel method in modeling channel crosstalk with phase sweeping for all possible crosstalk sources, as well as time/frequency dual-domain handling of reflection and multiple reflections.

Full-Waveform and Hybrid Simulation Methods

The basic concept for a full-waveform method is to simulate the waveform in time domain from the TX, to CH, and to RX, emulating the signal flow and propagation path in an actual link. In this section, we will first discuss the full-waveform method, followed by the hybrid method, and then CG, CR, and DR modeling.

Full-Waveform Method

Let's denote the digital bit sequence or data pattern as $d_i(t) = d_i(iT)$, where d_i is either 1 or 0 for the NRZ modulation, and i is the bit sequence index. The waveform at the TX output can be calculated as shown in Equation 4:

Equation 4.

$$V_{TX0}(t) = V_0 d_i(t) * h_{ffe}(t) * h_{dr}(t) * h_{txp}(t)$$

where V_o is the output voltage and $V_{TX0}(t)$ is the TX deterministic waveform.

Since the simulation is done in time domain, jitter can be introduced via phase modulation (PM), and noise can be introduced via amplitude modulation (AM). Let's denote $\Delta t_{TX}(t)$ as the TX jitter, and $\Delta V_{TX}(t)$ as the TX voltage noise. The TX waveform comprehending both jitter and voltage noise can be expressed in Equation 5:

Equation 5.

$$V_{TX}(t) = (V_0 + \Delta V_{TX}(t))d_i(t) * h_{ffe}(t) * h_{dr}(t + \Delta t_{TX}(t)) * h_{txp}(t)$$

where the $V_{TX}(t)$ is the waveform comprehending the TX deterministic behavior, jitter, and noise. Its characteristics, such as the eye diagram, rise/fall times, and BER contour, can be estimated subsequently [(10)].

The waveform at the channel output can be further calculated using Equation 6:

Equation 6.

$$V_{CH}(t) = V_{TX}(t) * h_{ch}(t)$$

Similarly, the waveform at the RX CTLE output can be calculated with Equation 7:

Equation 7.

$$V_{CTLE}(t) = (V_{CH}(t) + \Delta V_{RX}(t)) * h_{rxp}(t) * h_{ctle}(t)$$

In Equation 7, the RX voltage noise $\Delta V_{RX}(t)$ is introduced and the amplitude modulated in time domain. As mentioned earlier, as the DFE is a nonlinear system, the LTI model does not apply well. The DFE output needs to be modeled in a mixed-signal manner, seen in Equation 8:

Equation 8.

$$V_{DFE}(t) = V_{CTLE}(t) + \sum_{j=1}^{N} c_{-j} V_{SL}(t - jT + \Delta t_{RX})$$

where V_{SL} is the voltage after the DFE slicer, and $\Delta t_{RX}(t)$ is the receiver jitter associated with the recovered clock.

Equations 4 to 8 outline the theoretical frameworks for the full-waveform method in the JNEye tool. With them, waveforms at various observing points within a link can be modeled and estimated, along with the associated characteristics, such as eye diagrams, rise/fall times, jitter and noise PDFs, and BER contour CDFs. In practice, the full-waveform method may take a long time to complete if simulating a long-pattern bit sequence (for example, 10^{12} bits). In this case, extrapolation from the simulated probability level to a target probability level (for example, 10^{-12}) may be carried out [(11)] to alleviate the long simulation time limitation.

Hybrid Method

While the full-waveform method enables accurate and complete time-domain simulations of a high-speed link, it also faces the challenges of long simulation time when the bit sequence is long and when modeling small jitter and noise probability. To overcome this limitation, the hybrid method was developed. The JNEye tool's hybrid method maintains the full-waveform theoretical foundation and framework, but limits the time-domain jitter and noise to bounded or high-probability types, such as periodic jitter (PJ), bounded-uncorrelated jitter (BUJ), duty-cycle distortion (DCD), and ISI. For unbounded jitter and noise, such as random jitter (RJ) and random noise (RN), they are modeled in the statistical domain. In essence, the hybrid method invokes both time and statistical domains.

Let's denote the bounded waveform corresponding PDF as $p_b(t, v)$. The complete PDF modeling for both RJ and RN will be given by Equation 9:

Equation 9.

$$p(t,v) = p_b(t,v) * p_g(t) * p_g(v)$$

where p_g represents a Gaussian distribution.

A hybrid method reduces simulation time compared with a full-waveform method, as its time-domain simulation potion only needs to cover bounded processes or effects that are commonly contained within 1 million bits or less in practice.

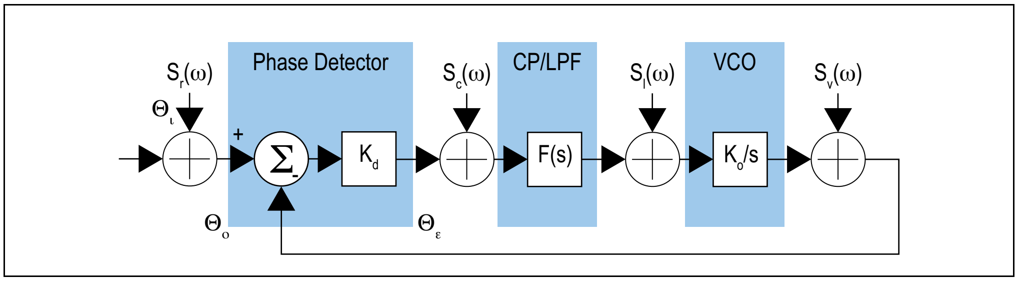
Clock Generation and Recovery

As mentioned in the Introduction section, jitter may be accounted for in a statistical link simulation method via convolution, and by assuming a jitter distribution, rather than modeling the jitter from its root causes. For a high-speed link, the TX and RX clock jitter is determined by the CG and CR circuits respectively, and those circuits are commonly implemented with a PLL circuit. A PLL circuit operates in time domain. Therefore, the clock jitter can be modeled in full-waveform or hybrid methods, rather than being assumed, as done in the statistical link simulation method.

PLL performance and jitter modeling is a well-studied subject. This paper will only give a high-level overview of how this subject is handled in high-speed link simulation. For more information, refer to [(10)] [(12)].

Figure 5 shows a diagram illustrating PLL functional blocks, the corresponding mathematical model in a complex S-domain, and jitter or phase noise (PN) injection and processes.

Figure 5. PLL Components, Associated Transfer Functions, and Jitter or Noise Process Illustration



The phase jitter or PN power-spectrum density (PSD) at the PLL output is determined by Equation 10:

Equation 10.

$$S_o(\omega) == S_r(\omega)\left|\frac{K_dK_oF(s)}{s+K_dK_oF(s)}\right|^2 + S_c(\omega)\left|\frac{K_oF(s)}{s+K_dK_oF(s)}\right|^2$$

$$+ S_l(\omega)\left|\frac{K_o}{s+K_dK_oF(s)}\right|^2 + S_v(\omega)\left|\frac{s}{s+K_dK_oF(s)}\right|^2$$

where $S_r(\omega)$, $Sc(\omega)$, $S_l(\omega)$, and $S_v(\omega)$ are jitter or PN PSD associated with reference to the clock input, charge-bump, low-pass filter (LPF), and voltage control oscillator (VCO) respectively, with $s = j\omega$

Jitter or PN variance can then be estimated by Equation 11 via inverse Fourier transform:

Equation 11.

$$\sigma_t^2(t) = 2(\sigma_0^2 - \xi^{-1}(S_o(\omega)))$$

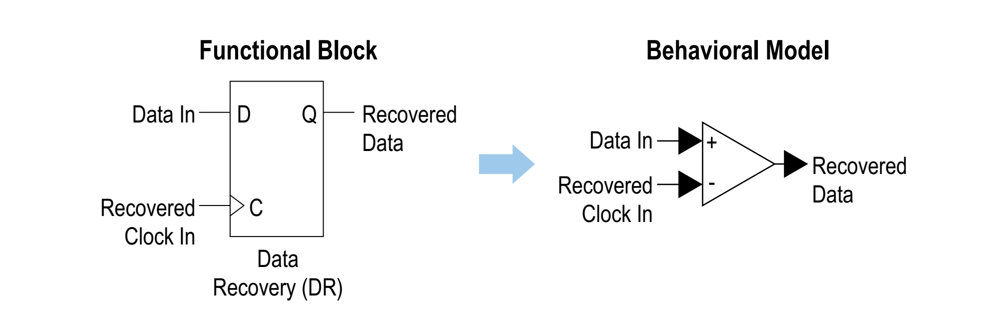
where σ_t^2 is the variance at time t and σ_0^2 is the total variance of the underlining jitter or PN process [(10)].

The σ_t is then used for jitter phase modulation in the full-waveform or hybrid method for the TX CG and RX CR. In the case of the TX CG, σ_t models uncorrelated reference clock jitter and intrinsic PLL jitter. More recent CR circuits use a dual-loop design [(2)] where a reference clock is not used during the mission operation phase. As such, the recovered clock jitter only consists of the PLL intrinsic jitter, and there will be no need to include a reference clock jitter in the σ_t. As for the correlated DCD, it is modeled by controlling the rise/falling edge time transition of the digital bit sequence $d_i(t)$.

Data Recovery

DR is the last analog stage for a high-speed link where the data after the CTLE and DFE stages finally gets recovered by a data latch or sampler as shown on the left side of Figure 6. The clock input to the DR block is the recovered clock that tends to move in phase with data, or equivalently track the jitter on the data. The equivalent behavioral model for the DR block is a difference function in time or phase domain, as shown on right side of Figure 6.

Figure 6. Data Recovery Functional Block Diagram and Corresponding Behavioral Model



Channel Modeling

Serial links, which replace parallel links such as IDE and PCI™ buses, bring the benefit of fewer traces and a smaller PCB area. Furthermore, differential signaling provides immunity to common mode noise, improves voltage margins, and reduces electromagnetic interference (EMI) radiation. Serial links also eliminate the inter-skew and clock-skew issues inherent in parallel link schemes. However, the high-frequency operations also bring the downsides from skin effects and dielectric loss, which are frequency-dependent and lead to a low-pass filter behavior of the channel. The short wavelength nature of the signal also signifies the transmission line effects, where impedance mismatches cause reflections among link components.

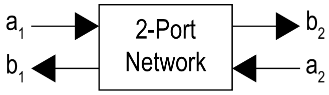
With increasing bandwidth, functionality, and density demands, link aggregations and space limitations force system builders to lay out more complex boards using links with narrower separation distances. Whenever a signal is driven along a wire, a magnetic field is developed around that wire. If two wires are placed adjacent to each other, it is possible that the two magnetic fields will interact with each other, causing a cross-coupling of energy between signals on the two wires, known as crosstalk.

Channel modeling is an essential part of getting accurate high-speed link simulation results. Channel models, which are mathematical representations of the physical communication links, can either come from measurements, for example, using the vector network analyzer (VNA) or time-domain reflectometry/time-domain transmission (TDR/TDT) methods, PCB circuit simulation/design tools, or 3D field solvers. A serial link usually comprises many subcomponents, such as connectors, PCB traces, cable, backplane, and vias. The JNEye tool is an end-to-end link simulation platform, and is capable of handling and manipulating the channel models correctly.

S-Parameters

S-parameter models are widely used in the industry as the standard channel model format. Figure 7 is an illustration of an S-parameter definition for a 2-port network.

Figure 7. A 2-Port Network and Its S-parameter Model



$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \bullet \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

where a_1, a_2 are incident power waves, and b_1, b_2 are outgoing power waves (namely, the square of their values give rise to the associated power), and S_{11}, S_{12}, S_{21}, S_{22} are the S-parameters defined by Equation 12:

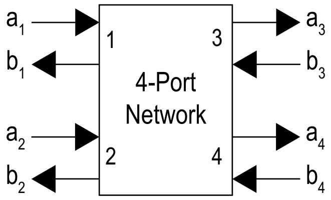
Equation 12.

$$S_{11} = \left.\frac{b_1}{a_1}\right|_{a_2 = 0} \qquad S_{12} = \left.\frac{b_1}{a_2}\right|_{a_1 = 0}$$

$$S_{21} = \left.\frac{b_2}{a_1}\right|_{a_2 = 0} \qquad S_{22} = \left.\frac{b_2}{a_2}\right|_{a_1 = 0}$$

We can now associate the above analytical expressions and derivations of S-parameter elements to channel effects, such as transmission line effects, signal attenuations, reflections, and crosstalk. For instance, S_{21} represents the system response, which is the insertion loss of the channel when a stimulus is applied at port 1 and a measurement is taken at port 2, given that port 2 is terminated with Z_0, and hence there is no reflection from the load side. With a 4-port network shown in Figure 8, we can also associate coupling and crosstalk effects with the S-parameter elements.

Figure 8. A 4-Port Network and Its S-Parameter Model

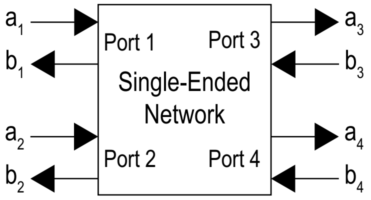


$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix} \bullet \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

- S_{11}—Reflection or return loss. S_{11} is the energy that is reflected back from the stimulus applied on port 1.
- S_{31}—Insertion loss. S_{31} is the amount of energy loss in the channel when the signal is traveling across the channel from port 1 to port 3.
- S_{21}—Near-end coupling. S_{21} is the amount of energy that is coupled to the stimulus-side adjacent port.
- S_{41}—Far-end coupling. S_{41} is the amount of energy that is coupled to the adjacent port across the channel.

The S-parameters from VNA measurements are generally in the single-ended format where they capture the responses between two ports. For differential-signaling systems, mixed-mode S-parameters are appropriate and provide more insights of the channel characteristics.

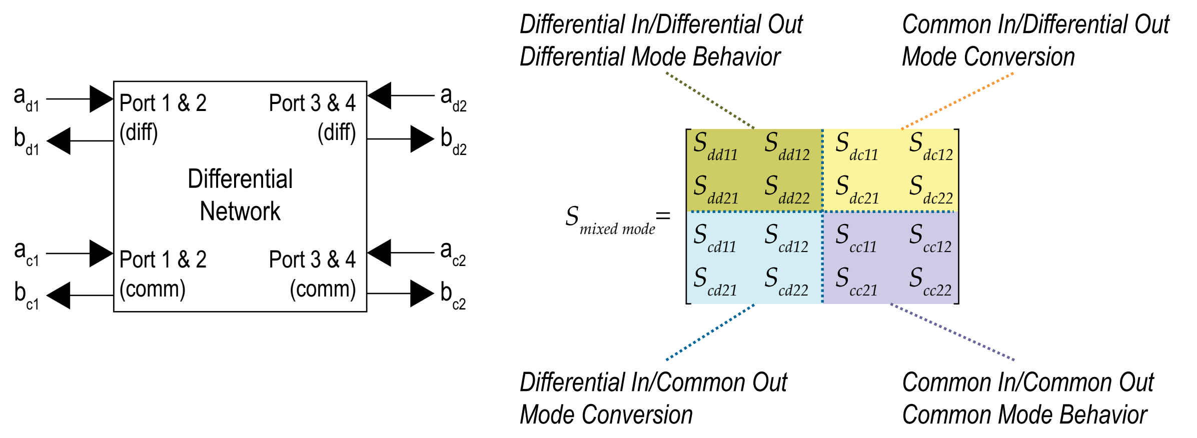
Figure 9. A 4-Port Network and Its Standard-Mode S-Parameter



$$S_{single\text{-}ended\ mode} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix}$$

Mixed-mode S-parameters are actually a transformation of a 4-port differential system into four single-ended subsystems, namely differential-input-differential-out, differential-input-common-output, common-input-differential-output, and common-input-common-output. Standard-mode to mixed-mode conversions can be performed mathematically using the formula seen in Figure 10.

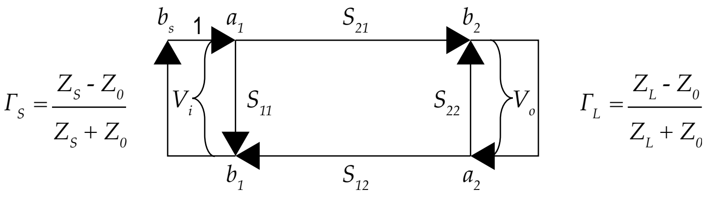
Figure 10. A 4-Port Network and Its Mixed-Mode S-Parameter


a_{d1}
b_{d1}
a_{c1}
b_{c1}
Port 1 & 2 (diff)
Port 3 & 4 (diff)
Differential Network
Port 1 & 2 (comm)
Port 3 & 4 (comm)
a_{d2}
b_{d2}
a_{c2}
b_{c2}
Differential In/Differential Out
Differential Mode Behavior
Common In/Differential Out
Mode Conversion
$S_{mixed\ mode} =$
S_{dd11} S_{dd12} S_{dc11} S_{dc12}
S_{dd21} S_{dd22} S_{dc21} S_{dc22}
S_{cd11} S_{cd12} S_{cc11} S_{cc12}
S_{cd21} S_{cd22} S_{cc21} S_{cc22}
Differential In/Common Out
Mode Conversion
Common In/Common Out
Common Mode Behavior

For high-speed communication systems, differential-input-differential-output responses are the most direct ways for observing the loss and return characteristics. The time-domain transformation of the differential mode S-parameters also gives information about the differential impedance profile. The differential-input-common-out and common-input-differential-output quadrants of mixed-mode S-parameters provide information regarding the mode conversion of the measured channels. The common-to-common quadrant describes the common mode propagation characteristics of the channel.

One classical approach to analyze the behavior of the network utilizes the signal flow graph. You can find numerous references and tutorials on this technique [(14)]. An example of a 2-port network analysis is shown in Figure 11.

Figure 11. A 2-Port Network with Source and Load Reflection Coefficients


$\Gamma_S = \frac{Z_S - Z_0}{Z_S + Z_0}$
b_s
1
a_1
S_{21}
b_2
V_i
S_{11}
S_{22}
V_o
b_1
S_{12}
a_2
$\Gamma_L = \frac{Z_L - Z_0}{Z_L + Z_0}$

The voltage transfer function of this 2-port system can then be derived from Equation 13:

Equation 13.

$$S_{mixed\text{-}mode} = M \cdot S_{single\text{-}ended\text{-}mode} \cdot M^{-1}$$

$$\text{where} \quad M = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Equation 14.

$$H_{VTF} = \frac{Z_s + Z_s}{Z_s} * \frac{S_{21} \cdot (1 + \Gamma_L) \cdot (1 - \Gamma_S)}{2 \cdot (1 - S_{11} \cdot \Gamma_s - S_{22} \cdot \Gamma_L - S_{21} \cdot S_{12} \cdot \Gamma_s \cdot \Gamma_L + S_{11} \cdot S_{22} \cdot \Gamma_s \cdot \Gamma_L)}$$

One usage of Equation 14 is to represent a physical channel between the TX package ball and the RX package ball where both the insertion loss of the channel reflections between the TX package and channel, and RX package and channel are comprehended. The impulse response can be obtained via an inverse Laplace transformation shown in Equation 15:

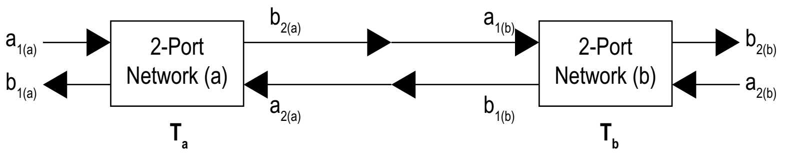
Equation 15.

$$h_{vi} = L^{-1}(H_{VFT}(s))$$

Equation 15 can be plugged in Equation 6 for the waveform calculation.

Furthermore, because the S-parameter is in the linear domain, a channel or device modeled by the S-parameters can be cascaded analytically. In order to do this, the signal flow matrix will need to be rearranged as shown in Figure 12.

Figure 12. Transmission Matrix of 2-Port Networks



$$\begin{bmatrix} b_{1(a)} \\ a_{1(a)} \end{bmatrix} = \begin{bmatrix} T_{11(a)} & T_{12(a)} \\ T_{21(a)} & T_{22(a)} \end{bmatrix} \bullet \begin{bmatrix} a_{2(a)} \\ b_{2(a)} \end{bmatrix} \qquad \begin{bmatrix} b_{1(b)} \\ a_{1(b)} \end{bmatrix} = \begin{bmatrix} T_{11(b)} & T_{12(b)} \\ T_{21(b)} & T_{22(b)} \end{bmatrix} \bullet \begin{bmatrix} a_{2(b)} \\ b_{2(b)} \end{bmatrix}$$

$$\begin{bmatrix} b_{1(a)} \\ a_{1(a)} \end{bmatrix} = \begin{bmatrix} T_a \end{bmatrix} \bullet \begin{bmatrix} T_b \end{bmatrix} \bullet \begin{bmatrix} a_{2(a)} \\ b_{2(b)} \end{bmatrix}$$

Note that $b_1(b) = a_2(a)$, and $a_1(b) = b_2(a)$. The T-parameters can be developed by manipulating the S-parameter equations into the appropriate form, as shown in Equation 16:

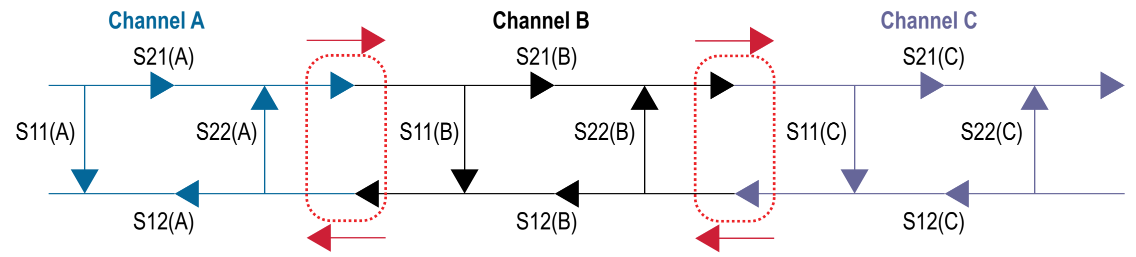
Equation 16.

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} -\dfrac{S_{11} \cdot S_{22} - S_{12} \cdot S_{21}}{S_{21}} & \dfrac{S_{11}}{S_{21}} \\ -\dfrac{S_{22}}{S_{21}} & \dfrac{1}{S_{21}} \end{bmatrix}$$

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{T_{12}}{T_{22}} & \dfrac{T_{11} \cdot T_{22} - T_{12} \cdot T_{21}}{T_{22}} \\ \dfrac{1}{T_{22}} & -\dfrac{T_{21}}{T_{22}} \end{bmatrix}$$

With T-parameters, multiple S-parameters can be combined into a single network, and you can perform network analysis mentioned in the previous sections. For example, you can calculate the S-parameter of a three-segment link (Channel A, Channel B, and Channel C, as shown in Figure 13) by cascading.

Figure 13. S-Parameter Cascading



Cascading $[T_{Combined}] = [T_{Channel\_A}] \bullet [T_{Channel\_B}] \bullet [T_{Channel\_C}]$

S-Parameters Conditioning and Validity Checking

To correctly utilize S-parameters in a link-modeling effort, board system builders, as well as the channel or connector component vendors, must check the validity of the S-parameters that they measured, generated, or received for/from their customers. In many cases, the S-parameters have been post-processed or realigned to suite the modeling needs. We also have to understand the nature of the channel so that the amount of S-parameter clarifications and adjustments can be reduced.

You must go through the following check list before using an S-parameter in the simulation:

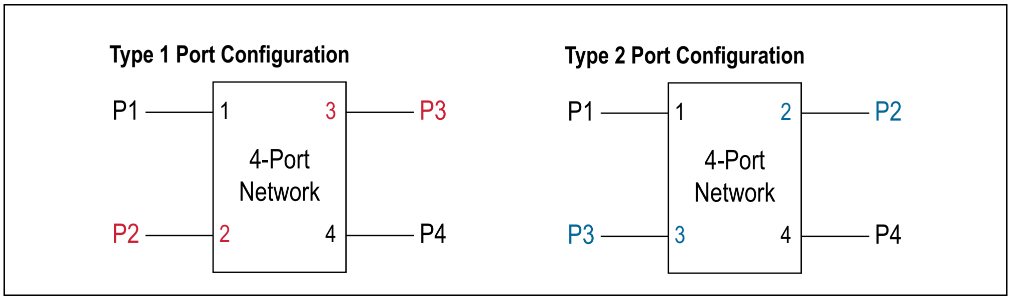
- *Port Configuration*

  The first thing to identify before utilizing S-parameters in your design is the S-parameter port configuration. S-parameters are mostly from channel measurements with VNAs. For 4-port networks, the physical port configurations are usually in one of two types (refer to Figure 14):

  - Type 1
  - Type 2

  You can usually find out the correct port configuration by contacting the vendor or by examining the notes or comments in the S-parameter files.

Figure 14. S-Parameter with Type 1 and Type 2 Port Configurations



- *Excessive Phase Wrapping*

  Another common issue with using S-parameters is phase wrapping. The phase response of a channel is an important factor in high-speed link modeling where it affects not only signal shapes, but also the propagation characteristics at different frequencies. For long channels, the phase changes among different frequencies can be large and beyond the ±180° or ±π range. This will lead to less accurate or even incorrect channel characteristics in a modeling environment. For longer channels, unwrapping is a desired step to maintain the S-parameter validity, and a smaller frequency step makes this easier to achieve.

- *DC Value*

  A common problem with measured S-parameters is missing the DC (0 Hz) value. The DC value is essential in reconstructing the channel model in both time and frequency domains. In general, VNAs do not measure and capture the channel response at 0 Hz and some post processing will be needed to determine the DC value of the S-parameters before using them in system simulations.

- *Frequency Grid and Range*

  The frequency step size of S-parameters can sometimes cause problems in interpreting the channel characteristics in simulations. In addition, the frequency range of the S-parameters should have sufficient margins to cover the operational frequency range. The fifth harmonic is commonly recommended for the maximum frequency to insure the necessary accuracy.

- *Causality and Passivity*

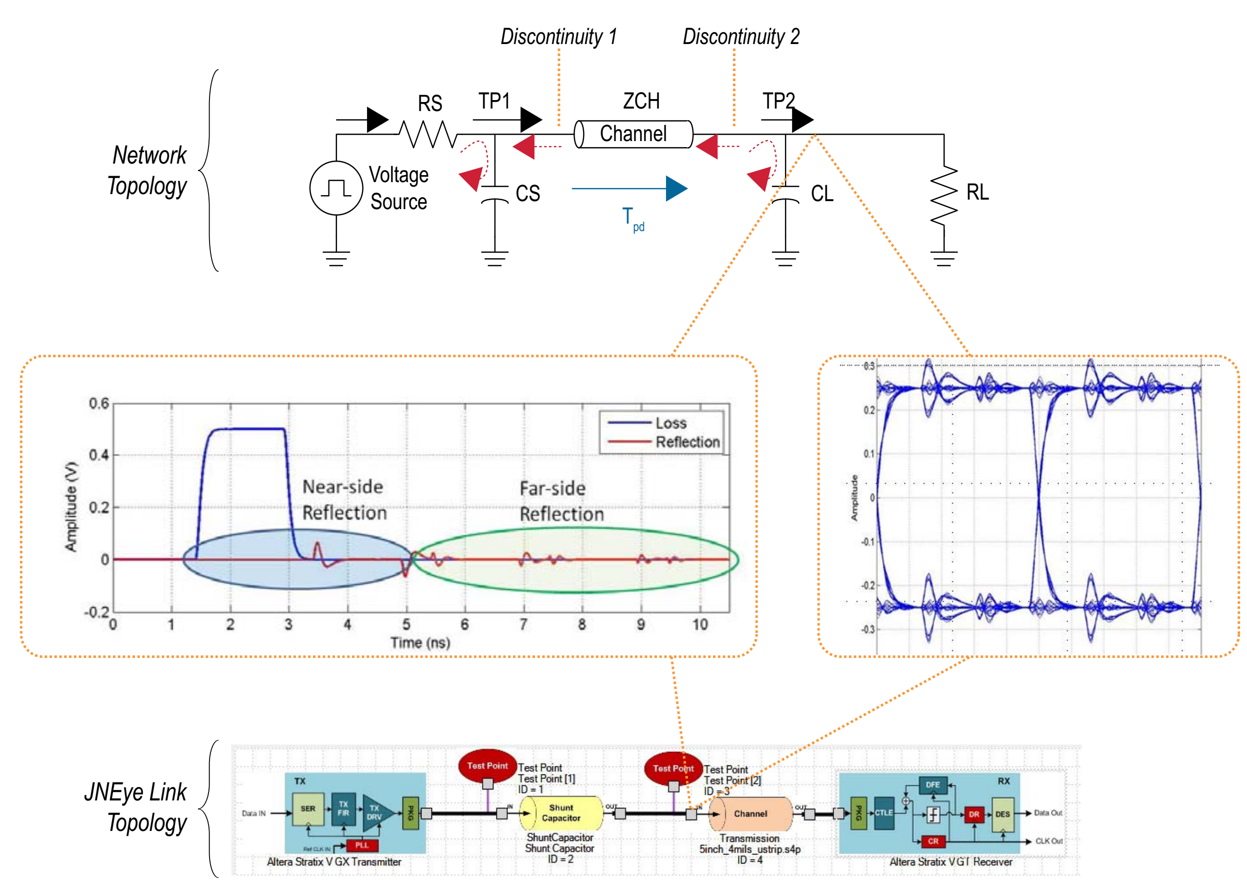
  Regular channel components, such as PCB, cables, and connectors, are passive LTI by nature. Therefore, they should have no gain or amplification, and no output before the signal is input into the component. Before utilizing channel models, one should check if the passivity and causality properties will hold. If a passivity and causality violation is detected, it is recommended to remeasure or regenerate the channel model. There are passive and causality remediation methodologies, but their effectiveness can vary.

The JNEye tool has embedded intelligence in handling, determining, and correcting the above potential issues when it interprets S-parameter models. The JNEye Channel Wizard feature will assist you in determining the S-parameter's port configuration and other settings via a graphical presentation (See Figure 18 for example).

Modeling Insertion Loss, Reflection, and Crosstalk

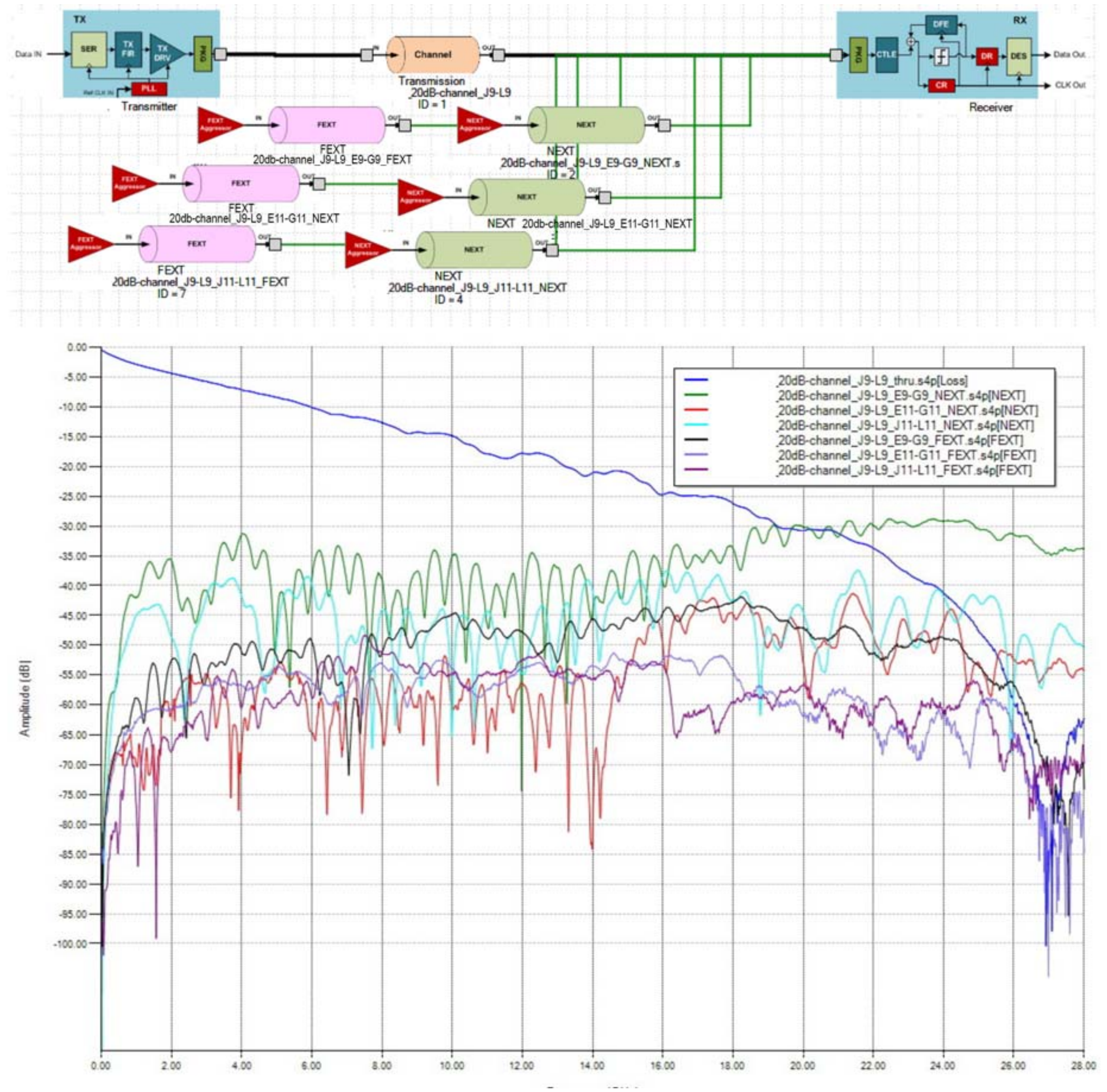
The JNEye tool captures and handles a broad range of channel effects, such as insertion loss, reflections, and crosstalk. Figure 15 shows an extreme condition where significant reflections caused by impedance mismatches were captured in the waveform and eye diagram.

Figure 15. Reflections Caused by Impedance Mismatch



The JNEye tool also provides comprehensive crosstalk modeling capabilities. You can easily and precisely set up a serial link using the JNEye Link Designer with multiple crosstalk channels and their associated programmable NEXT and FEXT aggressors as shown in Figure 16.

Figure 16. Channel Characteristics and Crosstalk Simulation in the JNEye Tool



Automatic Adaptation

In both full-waveform and hybrid methods, automatic equalization and clocking optimization are possible, unlike traditional statistical link simulation methods. There are two types of optimization methodologies in the JNEye tool; the first is with the sequence, FIR => CTLE (and its variations). In this method, a set or sets of TX equalization and clock optimal parameters are found by minimizing the error between the simulated waveform at the link output versus the expected output using methods such as least-mean-square (LMS). Then the best set or sets of TX and RX equalization and clocking parameters are found by minimizing a figure of merit (FOM), such as eye width (EW), eye height (EH), eye area (EA), or signal to noise ratio (SNR). Note that different FOMs will yield different optimal solutions.

The second method is with the sequence, CTLE => FIR (and its variations). In the second method, a set or sets of RX equalization and recovered clock parameters are found. Then each set or sets of parameters will cycle through possible TX equalization and clocking parameters to find the best set or sets of TX and RX equalization and clocking parameters via a FOM.

Based on these two base link adaptation approaches, the JNEye tool provides four link adaptation methods that you can choose from. Table 1 describes the methods and their applications.

Table 1. Link Optimization Methods in the JNEye Tool

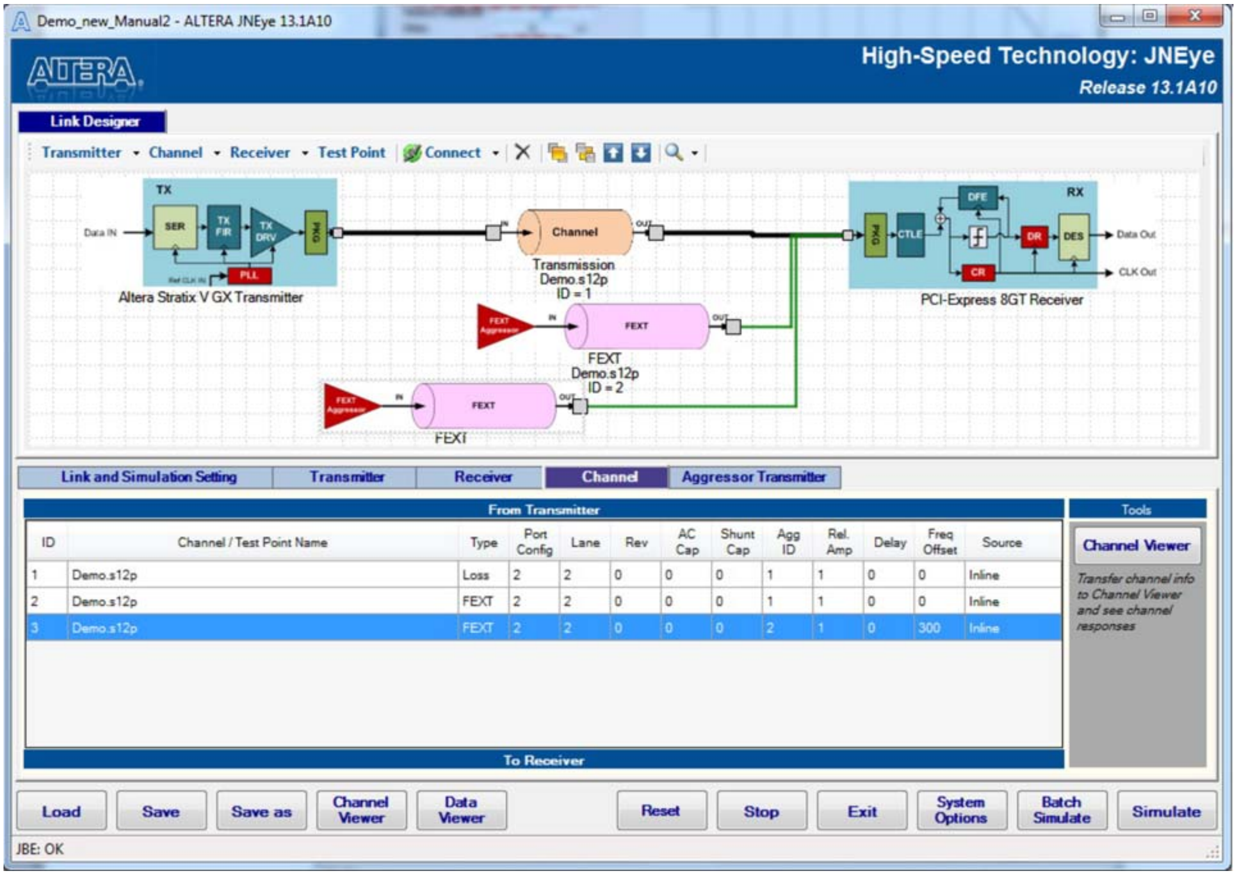
Link Optimization Method	Description	Applications
FIR => CTLE => DFE	This sequence optimizes the link performance by finding the optimal TX and/or RX equalization settings. This method prioritizes TX equalization, such as pre-emphasis, de-emphasis, and FIR, over RX equalization schemes. The RX DFE is adapted at the final stage.	Suitable for most applications or channels for time-efficient link optimizations and heavy insertion loss channels, such as backplanes. It is the default link optimization method in the JNEye tool.
CTLE => FIR => DFE	This sequence prioritizes the RX CTLE over TX equalization. The RX DFE is adapted at the final stage.	Suitable for links with strong impedance discontinuities/reflections and relative low loss.
FIR => CTLE & DFE	This sequence extends the FIR => CTLE => DFE sequence method by enabling the RX DFE when the RX optimization is performed. This method exploits DFE capabilities by possibly reducing the channel compensation from the CTLE.	Suitable for links with large crosstalk noises.
CTLE => FIR & DFE	This sequence extends the CTLE => FIR => DFE sequence method by joint-optimizing TX pre-emphasis/FIR and RX DFE.	

Compared with traditional iteration-based TX and RX equalization optimization schemes, the JNEye tool's link adaptation schemes are time efficient as they usually take about 5 to 20 minutes to find the optimal TX and RX equalization settings.

Using the JNEye Toolset: Case Study Examples

This section shows a link simulation example in the JNEye tool using a PCIe Gen3 channel (shown in Figure 3, Figure 18, and Figure 21). The link can be set up quickly using the JNEye Link Designer editor as shown in Figure 17. The JNEye tool supports several TX, RX, and CH component models that are listed in Table 1. Package models, PVT variations, and characterization data of Altera® devices are included in the JNEye tool, which can be used in simulations. Custom TX and RX types allow you to construct and configure their own transmitters and receivers using basic processing blocks available in the JNEye tool. This feature is useful in performing what-if analysis. The JNEye tool supports link simulations with IBIS-AMI device models where a serial link between an Altera FPGA and other transmitters or receivers can be evaluated.

Figure 17. PCIe Gen3 Link Configuration Using Link Designer in the JNEye Tool



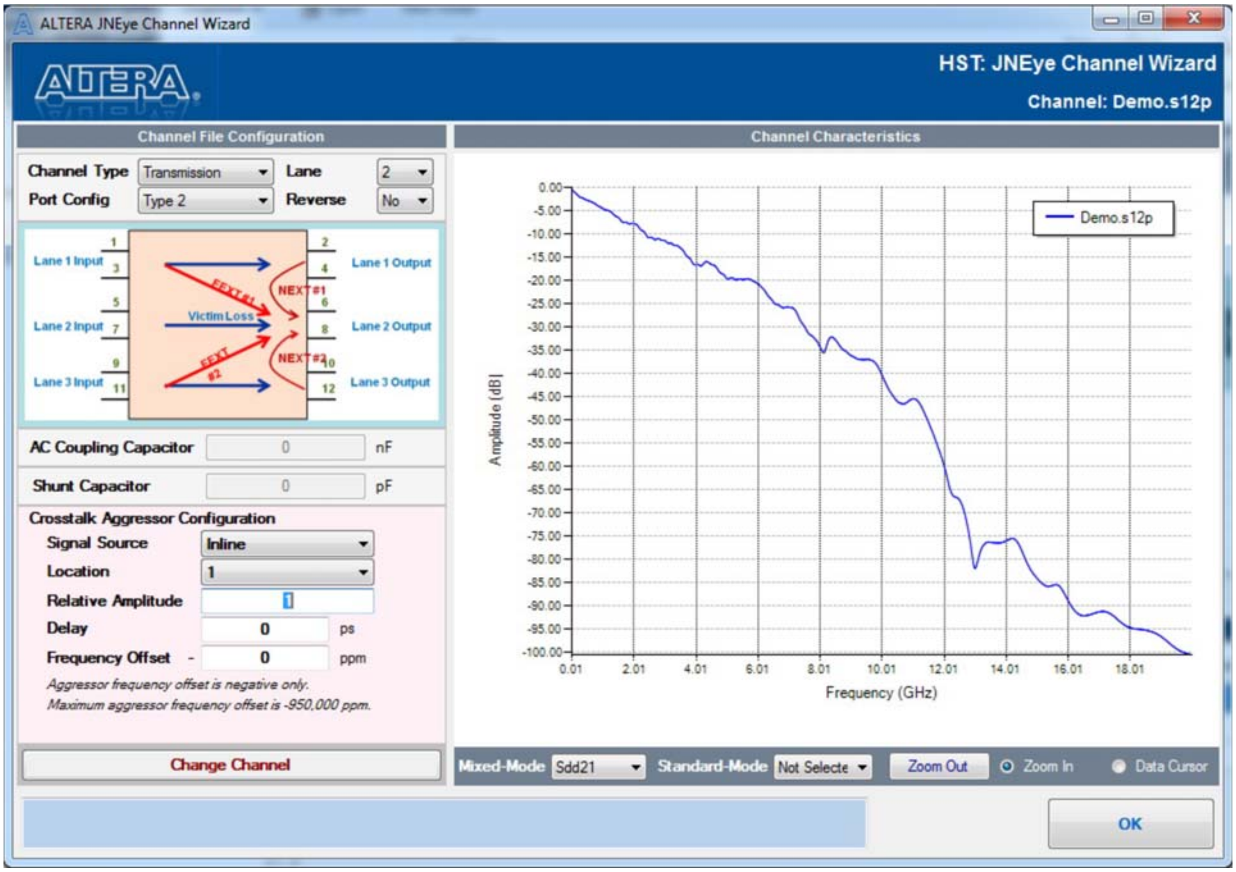
ID	Channel / Test Point Name	Type	Port Config	Lane	Rev	AC Cap	Shunt Cap	Agg ID	Rel. Amp	Delay	Freq Offset	Source
1	Demo.s12p	Loss	2	2	0	0	0	1	1	0	0	Inline
2	Demo.s12p	FEXT	2	2	0	0	0	1	1	0	0	Inline
3	Demo.s12p	FEXT	2	2	0	0	0	2	1	0	300	Inline

Table 2. Supported Transmitter, Receiver, and Channel Types in the JNEye Tool

Link Component	Transmitter	Channel	Receiver
Component Name	Altera Stratix® V GX Altera Arria® V GZ Altera Stratix V GT Altera Arria 10 GX Altera Arria 10 GT IBIS-AMI Custom Standard compliant (e.g., PCIe 8GT)	PCB trace or cable Connector Far-end crosstalk Near-end crosstalk Package AC coupling capacitor Shunt capacitor	Altera Stratix V GX Altera Arria V GZ Altera Stratix V GT Altera Arria 10 GX Altera Arria 10 GT IBIS-AMI Custom Standard compliant (e.g., PCIe 8GT)

Channel components, such as backplanes, cables, and connectors, can be imported via the JNEye Channel Wizard feature. Upon selecting a Touchstone S-parameter channel file, the JNEye Channel Wizard will examine the contents and determine the configuration of the selected channel file. The JNEye tool supports the importing, cascading, and modeling of multilane S-parameters and crosstalk channels. Figure 18 shows the JNEye Channel Wizard in action.

Figure 18. Using Channel Wizard to Import a 12-Port Touchstone S-Parameter File



ALTERA JNEye Channel Wizard

HST: JNEye Channel Wizard

Channel: Demo.s12p

Channel File Configuration

Channel Type: Transmission
Lane: 2
Port Config: Type 2
Reverse: No

AC Coupling Capacitor: 0 nF

Shunt Capacitor: 0 pF

Crosstalk Aggressor Configuration

Signal Source: Inline
Location: 1
Relative Amplitude: 1
Delay: 0 ps
Frequency Offset: - 0 ppm

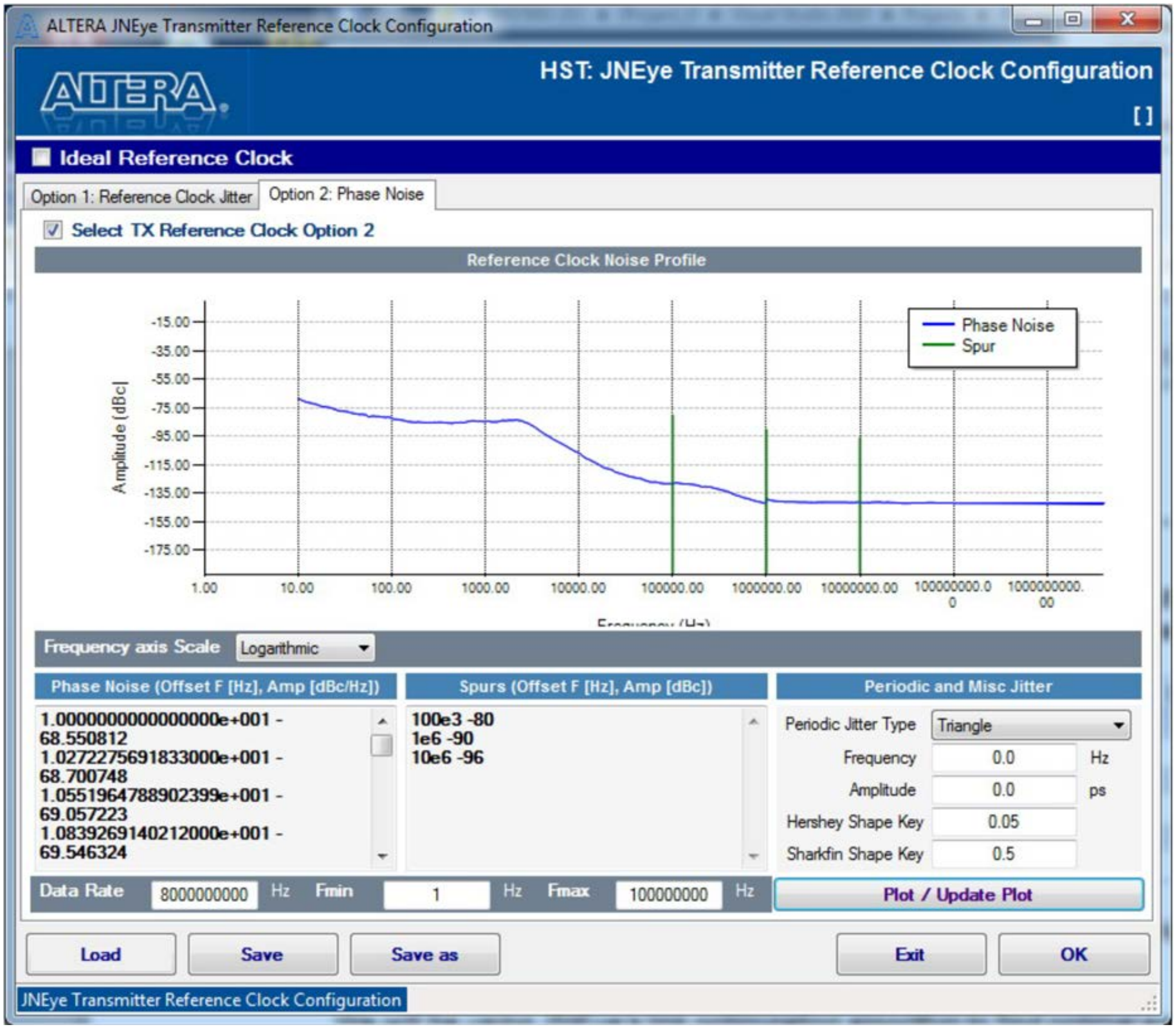
Aggressor frequency offset is negative only.
Maximum aggressor frequency offset is -950,000 ppm.

Channel Characteristics

Mixed-Mode: Sdd21
Standard-Mode: Not Selecte

The JNEye tool facilitates clock jitter or noise modeling with its Transmitter Reference Clock and CG PLL Configuration GUI, and underlining modeling or simulation engine. Figure 19 shows the user interface where you can input or import reference clock PN figures, spurs, and various PJ sources. For instance, this example shows the PN measurement of a typical 100 MHz reference clock. The measurement data can be ported into the JNEye tool and then simulated either in the statistical, full-waveform, or hybrid simulation mode. Figure 20 illustrates the PN characteristics as it travels through the link.

Figure 19. JNEye's Transmitter Reference Clock Configuration GUI



ALTERA JNEye Transmitter Reference Clock Configuration

HST: JNEye Transmitter Reference Clock Configuration

☐ Ideal Reference Clock

Option 1: Reference Clock Jitter | Option 2: Phase Noise

☑ Select TX Reference Clock Option 2

Reference Clock Noise Profile

Frequency axis Scale: Logarithmic

Phase Noise (Offset F [Hz], Amp [dBc/Hz])

```
1.0000000000000000e+001 -
68.550812
1.0272275691833000e+001 -
68.700748
1.0551964788902399e+001 -
69.057223
1.0839269140212000e+001 -
69.546324
```

Spurs (Offset F [Hz], Amp [dBc])

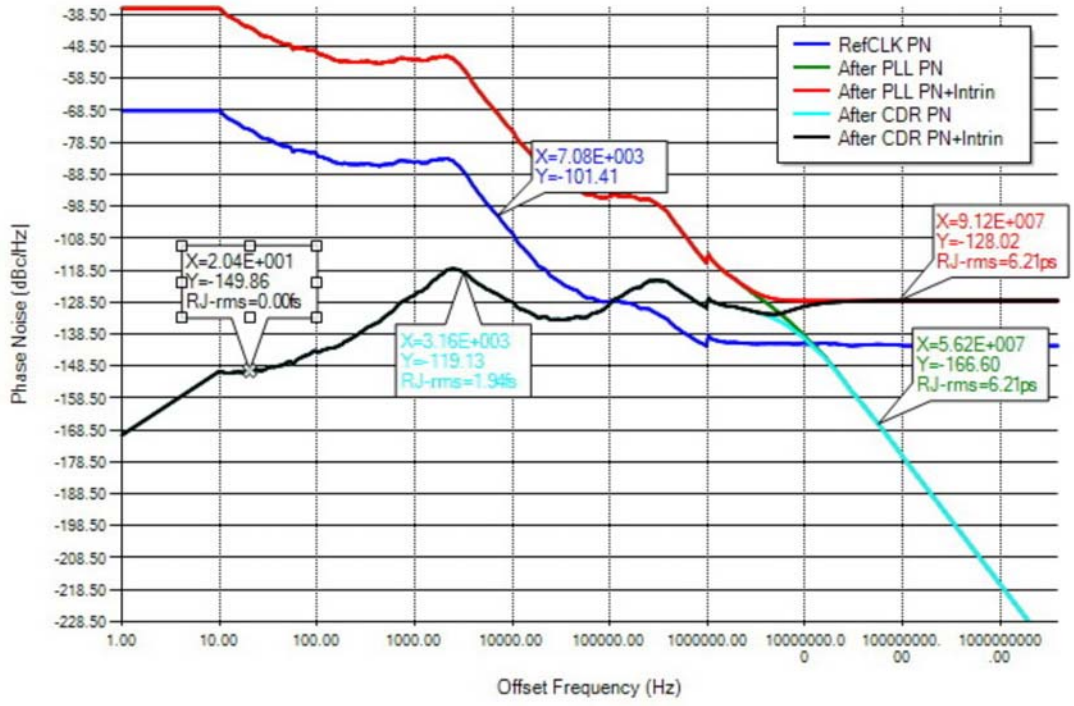
```
100e3 -80
1e6 -90
10e6 -96
```

Periodic and Misc Jitter

Periodic Jitter Type	Triangle	
Frequency	0.0	Hz
Amplitude	0.0	ps
Hershey Shape Key	0.05	
Sharkfin Shape Key	0.5	

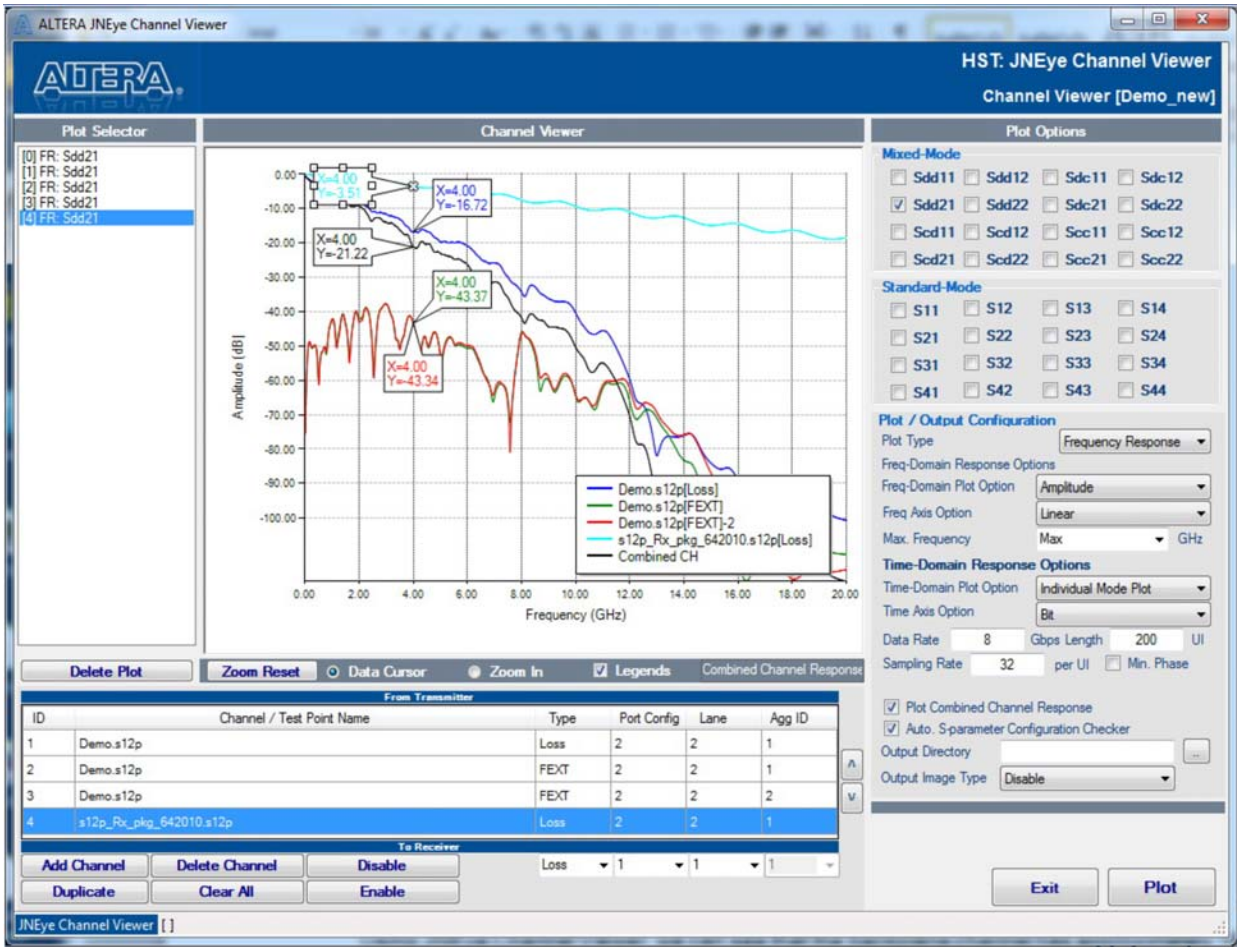
Data Rate 8000000000 Hz Fmin 1 Hz Fmax 100000000 Hz

Figure 20. Phase Noise and Random Jitter Analysis in the JNEye Tool



The JNEye Channel Viewer enables you to observe and analyze channel characteristics in frequency or time domain. Figure 21 shows the frequency responses of the channels used in this example.

Figure 21. View Channel Characteristics with the JNEye Channel Viewer



Mixed-Mode

□ Sdd11	□ Sdd12	□ Sdc11	□ Sdc12
☑ Sdd21	□ Sdd22	□ Sdc21	□ Sdc22
□ Scd11	□ Scd12	□ Scc11	□ Scc12
□ Scd21	□ Scd22	□ Scc21	□ Scc22

Standard-Mode

□ S11	□ S12	□ S13	□ S14
□ S21	□ S22	□ S23	□ S24
□ S31	□ S32	□ S33	□ S34
□ S41	□ S42	□ S43	□ S44

Plot / Output Configuration

Plot Type: Frequency Response

Freq-Domain Response Options

Freq-Domain Plot Option: Amplitude

Freq Axis Option: Linear

Max. Frequency: Max GHz

Time-Domain Response Options

Time-Domain Plot Option: Individual Mode Plot

Time Axis Option: Bit

Data Rate 8 Gbps Length 200 UI

Sampling Rate 32 per UI □ Min. Phase

☑ Plot Combined Channel Response

☑ Auto. S-parameter Configuration Checker

Output Directory

Output Image Type: Disable

From Transmitter

ID	Channel / Test Point Name	Type	Port Config	Lane	Agg ID
1	Demo.s12p	Loss	2	2	1
2	Demo.s12p	FEXT	2	2	1
3	Demo.s12p	FEXT	2	2	2
4	s12p_Rx_pkg_642010.s12p	Loss	2	2	1

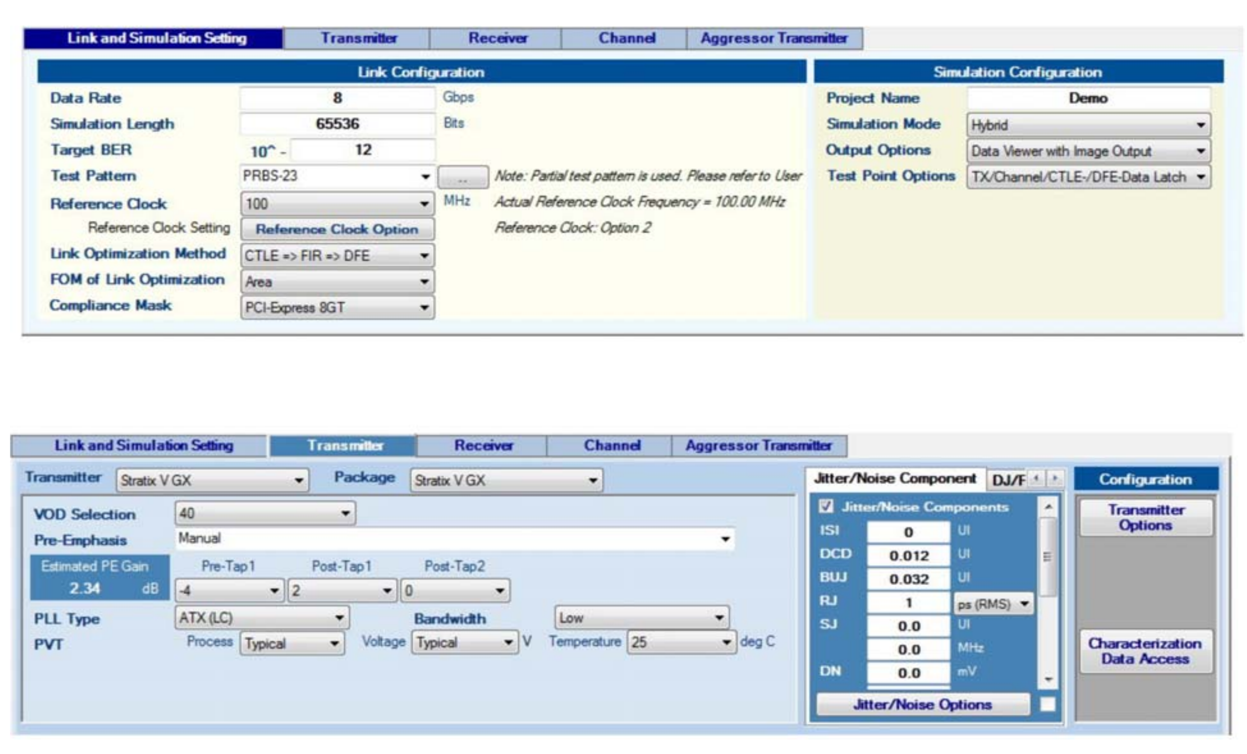
To Receiver

Loss 1 1 1

The JNEye tool includes comprehensive link configurations where you can customize the simulation data rate, test pattern, BER target, simulation mode, and many other settings, as shown in Figure 22. Full-scale jitter and noise configuration and modeling are provided in the GUI and simulation engine, as shown in Figure 23. For Altera FPGAs, the device characterization database is embedded in the JNEye tool. You can access and utilize the data to accurately evaluate link performance.

Refer to "Modeling PVT Variations" on page 29 for further discussions on jitter and noise modeling in the JNEye tool.

Figure 22. Link Configuration in the JNEye Tool



Data Rate 8 Gbps

Simulation Length 65536 Bits

Target BER 10^ - 12

Test Pattern PRBS-23

Note: Partial test pattern is used. Please refer to User

Reference Clock 100 MHz

Actual Reference Clock Frequency = 100.00 MHz

Reference Clock Setting

Reference Clock: Option 2

Link Optimization Method CTLE => FIR => DFE

FOM of Link Optimization Area

Compliance Mask PCI-Express 8GT

Project Name Demo

Simulation Mode Hybrid

Output Options Data Viewer with Image Output

Test Point Options TX/Channel/CTLE-/DFE-Data Latch

Transmitter Stratix V GX

Package Stratix V GX

VOD Selection 40

Pre-Emphasis Manual

Estimated PE Gain 2.34 dB

Pre-Tap1 -4

Post-Tap1 2

Post-Tap2 0

PLL Type ATX (LC)

Bandwidth Low

PVT Process Typical Voltage Typical V Temperature 25 deg C

☑ Jitter/Noise Components

ISI 0 UI

DCD 0.012 UI

BUJ 0.032 UI

RJ 1 ps (RMS)

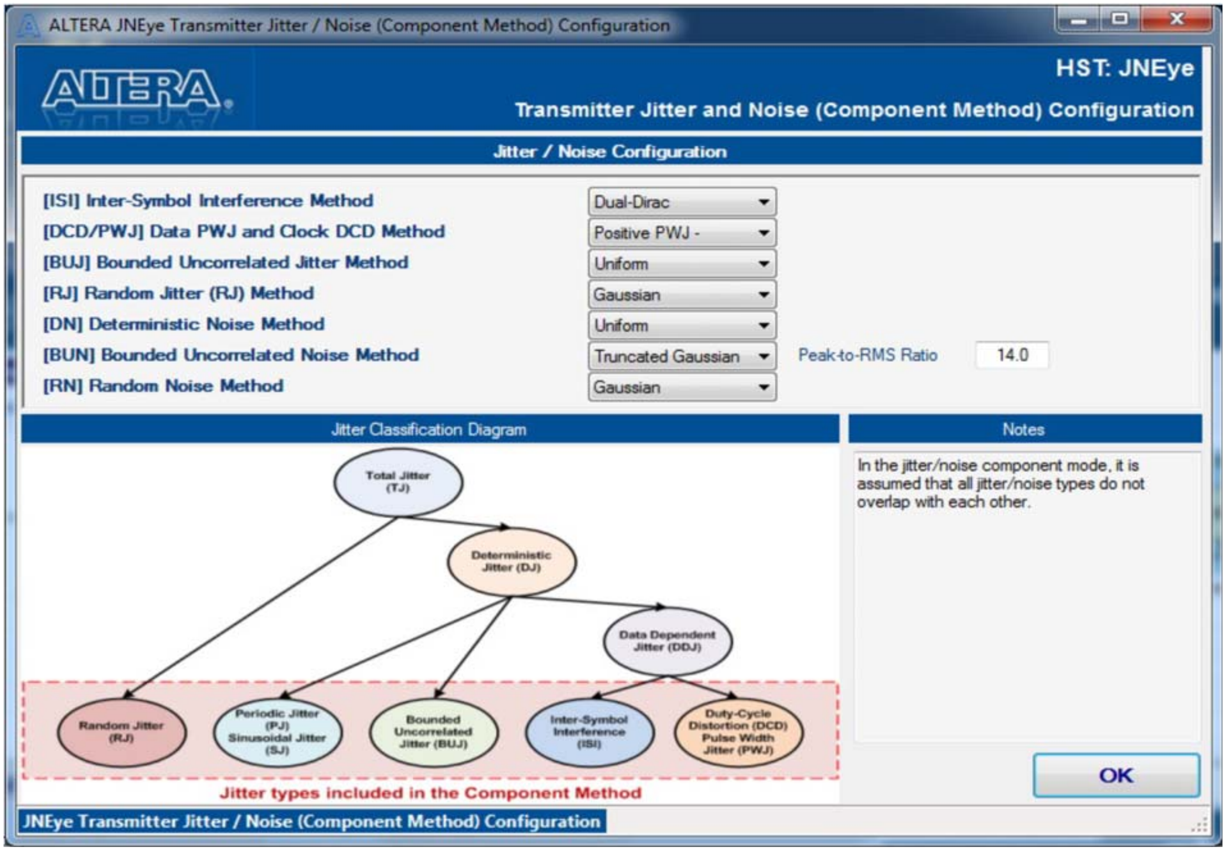
SJ 0.0 UI

0.0 MHz

DN 0.0 mV

□

Figure 23. Transmitter Jitter and Noise Configuration



ALTERA JNEye Transmitter Jitter / Noise (Component Method) Configuration

HST: JNEye

Transmitter Jitter and Noise (Component Method) Configuration

Jitter / Noise Configuration

[ISI] Inter-Symbol Interference Method: Dual-Dirac

[DCD/PWJ] Data PWJ and Clock DCD Method: Positive PWJ -

[BUJ] Bounded Uncorrelated Jitter Method: Uniform

[RJ] Random Jitter (RJ) Method: Gaussian

[DN] Deterministic Noise Method: Uniform

[BUN] Bounded Uncorrelated Noise Method: Truncated Gaussian; Peak-to-RMS Ratio: 14.0

[RN] Random Noise Method: Gaussian

Notes

In the jitter/noise component mode, it is assumed that all jitter/noise types do not overlap with each other.

JNEye Transmitter Jitter / Noise (Component Method) Configuration

ALTERA
Jitter Classification Diagram
Total Jitter (TJ)
Deterministic Jitter (DJ)
Data Dependent Jitter (DDJ)
Random Jitter (RJ)
Periodic Jitter (PJ) Sinusoidal Jitter (SJ)
Bounded Uncorrelated Jitter (BUJ)
Inter-Symbol Interference (ISI)
Duty-Cycle Distortion (DCD) Pulse Width Jitter (PWJ)
Jitter types included in the Component Method
OK

Figure 24 shows an example of a full-waveform method, for the same PCIe Gen3 channel shown in Figure 3, but with a PRBS-223 data pattern (approximately 8 million bits). The CG PLL associated the DCD, BUJ, PJ, RJ, reference clock jitter, deterministic noise (DN), and RN for the TX, the insertion loss, crosstalk, and return loss (RL) for the channel, and the CR PLL associated the DCD, BUJ, PJ, RJ, DN, and RN for the RX, as well as the TX jitter and noise interactions with the channel, are all modeled. This is not possible in a traditional statistical simulation approach, and very challenging to do using SPICE models. The simulation took 3.5 hours to complete, with the same computer system used for the statistical method. Note that the BER CDF at a probability level of 10^{-6} or lower is extrapolated in Q-space using a dual-Dirac jitter model [(10)]. The EW and EH are 0.47 UI, and 69 mv at a BER of 10-12.

Figure 24. Eye Diagram and Associated Jitter/Noise PDFs, BER Eye and Associated Jitter/Noise CDFs, and a Zoomed-In CDF in Q-Space

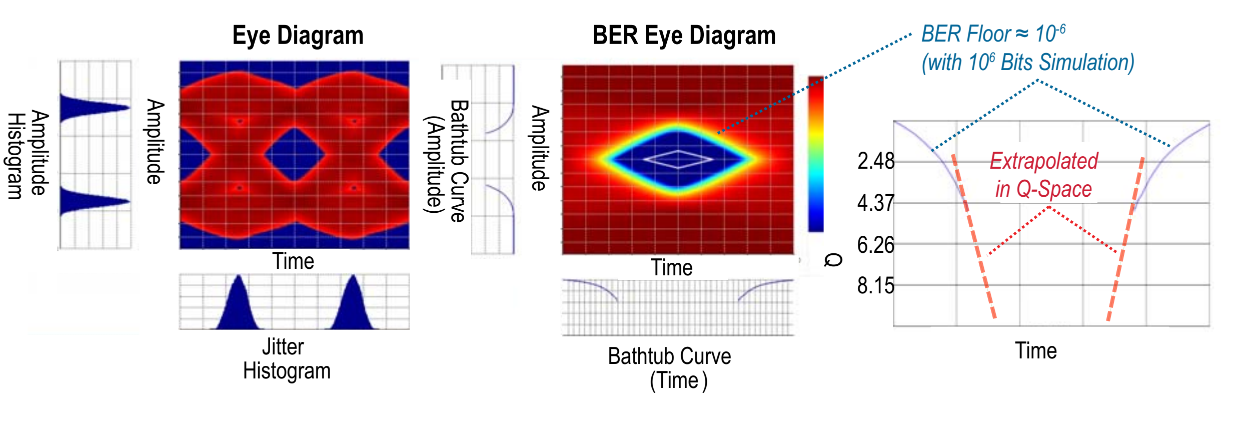


Figure 25. Eye Diagram, Noise vs. Time, TIE, and RJ PDF at Various Points Within the Link

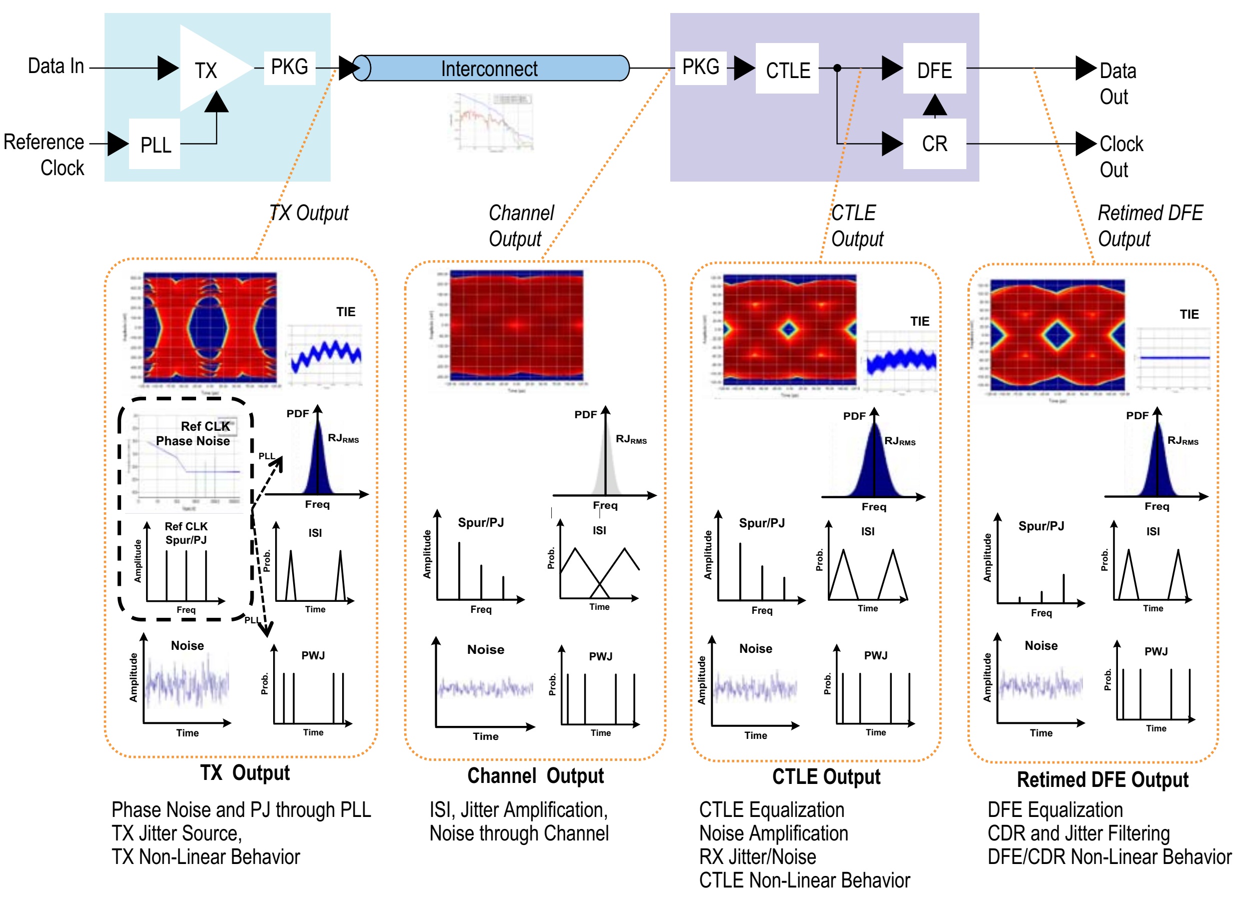


Figure 25 shows a simulation example of the hybrid method using the same PCIe Gen3 channel, where 0.065 million bits are simulated and the worst ISI edge transition is covered, with the same device parameters shown in Figure 24. The simulation time is reduced to ~3.5 minutes. The eye diagram, time-interval error (TIE) vs. time function, spur/PJ amplitude through the PLL and CR, RJ PDF, ISI amplitude, noise amplitude vs. time function, pulse-width jitter, and DCD through the TX, CH, and RX at various observation points within the link are shown.

- At the TX output, the eye diagram is pre-emphasized/de-emphasized and a low-frequency sinusoidal jitter can be seen in the TIE versus time plot.
- At the channel output, the eye is closed due to the lossy channel and associated significant amount of ISI, even with the TX FIR filter turned on. As the eye is closed, the TIE has no meaning and is not shown. The noise amplitude is also reduced due to the lossy channel.
- At the CTLE output, the eye is opened by the CTLE significantly, while the jitter versus time function and RJ PDF widths get larger due to the CTLE intrinsic jitter addition, compared with those at the channel output.
- At the DFE output, the eye gets further opened, and the noise amplitude is reduced, due to the DFE signal only boost. Meanwhile, the RJ PDF width, spur/PJ amplitude, and TIE are significantly reduced due to the CR low-frequency jitter tracking. The EW and EH are 0.45 UI and 66 mv respectively at BER 10-12. In comparison with the full-waveform method, the differences are -0.02 UI (-4.3%) and -3 mv (-4.3%). Note that the hybrid method shows a slightly pessimistic result compared with the full-waveform method. This is because in the full-waveform method, 8 million bits were simulated, corresponding to a “deeper” jitter tracking down to the kHz range, while for the hybrid method, only 0.065 million bits were simulated, corresponding to a not so “deep” jitter tracking down to ~100 kHz.

On the other hand, statistical methods cannot show any of the time-domain capabilities and jitter/noise channel interactions. We did not compare results from the full-waveform and hybrid methods with those from statistical methods, as they would not be apple-to-apple comparisons. This is because statistical methods cannot model CG, CR, DR, and TX jitter/noise interactions with the CH, RX, automatic adaptation, and the nonlinear TX and RX behaviors.

Table 3 lists a relative characteristic comparison of the statistical, full-waveform, and hybrid methods. However, absolute comparison requires specific simulation conditions or assumptions, and the result would be hard to generalize.

Table 3. Comparison Between Different Simulation Methods

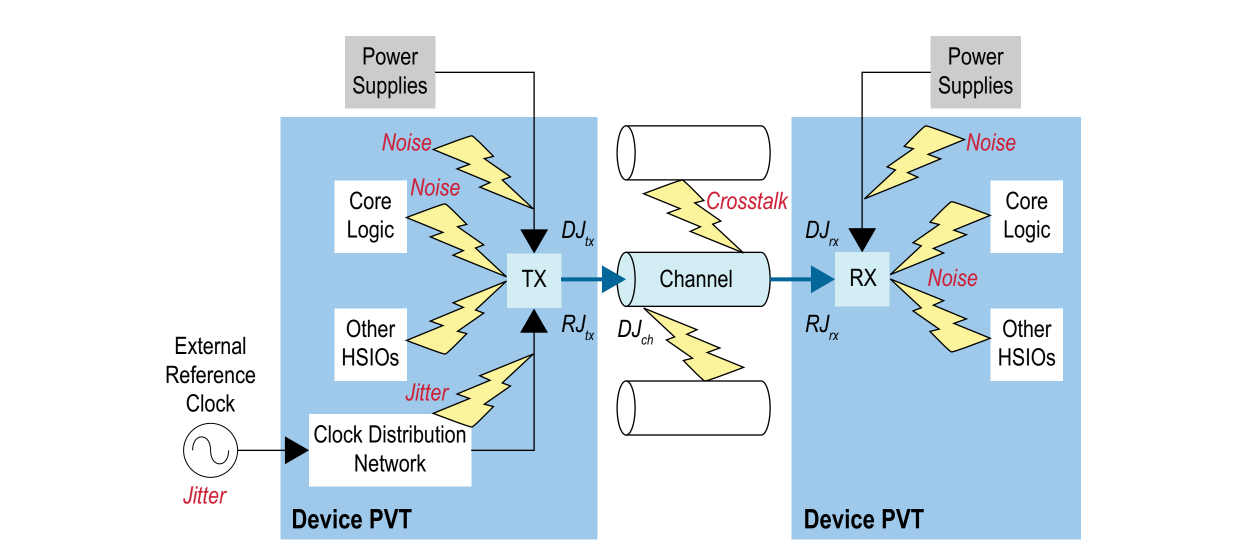
	Statistical	Hybrid	Full-Waveform
Eye-Diagram PDF/BER CDF	Y	Y	Y
Rise/Fall Time	Partial	Y	Y
Waveform/Transient	N	Y	Y
Jitter and Noise Components	Partial	Y	Y
TX Jitter/Noise Interaction with Channel and RX	N	Y	Y
CG, CR, and DR	N	Y	Y
Automatic Adaptation	N	Y	Y
Overall Accuracy (Relative)	1	Better	Best
Simulation Time (Relative)	1	Longer	Longest

Modeling PVT Variations

If behavioral TX and RX models are based on generic LTI mathematical representation (refer to [5]), then they cannot model TX and RX PVT variations, a practical subject that a link design and simulation must address. If, however, the behavioral TX and RX models are developed based on a SPICE model, they are commonly based on typical PVT values, and its variations are not accounted for either.

To model the device PVT variations, behavioral TX and RX models must be derived from a SPICE model or equivalent models that are capable of modeling transistor and IC PVT variations. This can be achieved by creating a bank of TX and RX models covering PVT variation space and corners. However, in practice, SPICE models can only model deterministic behavior such as deterministic waveforms and associated ISI, offering little or no knowledge on the IC/device non-deterministic characteristics, such as jitter, noise, and their PVT variations. Furthermore, the TX and RX jitter and noise depend on the IC/device operating conditions and are often affected by neighboring circuit activities, as shown in Figure 26.

Figure 26. Illustration of Sources of Variability for a High-Speed Link



To account for the jitter and noise variability over operating conditions and PVT, the JNEye tool utilizes a novel measurement-based method that has been developed recently. Early work in this area covers jitter only [(13)]. In this paper we extend the coverage to both jitter and noise. In this method, the jitter and noise components, such as DCD, BUJ, RJ, DN, and RN, are measured over a wide range of TX and RX operating conditions, various devices and channels, and PVT values. The common operating conditions for the TX and RX model may include, but not limited to—data pattern (*pat*), data rate (f_d), logic core, and neighboring channel activity level. The specific operating conditions may include various output voltage (V_o), PLL bandwidth (f_{pll}), reference clock frequency (f_{osc}) for the TX, and input voltage (V_i), clock recovery PLL bandwidth (f_{cr}) for the RX. Multiple-dimensional jitter and noise component look-up tables (LUTs) are created after comprehensive measurements are made, which can be conceptually represented by Equation 17 and Equation 18:

Equation 17. For TX

$$(DCD, BJU, RJ, DN, RN)_{TX} = LUT_{TX}(pat, f_d, v_o, f_{pll}, f_{osc}, PVT)$$

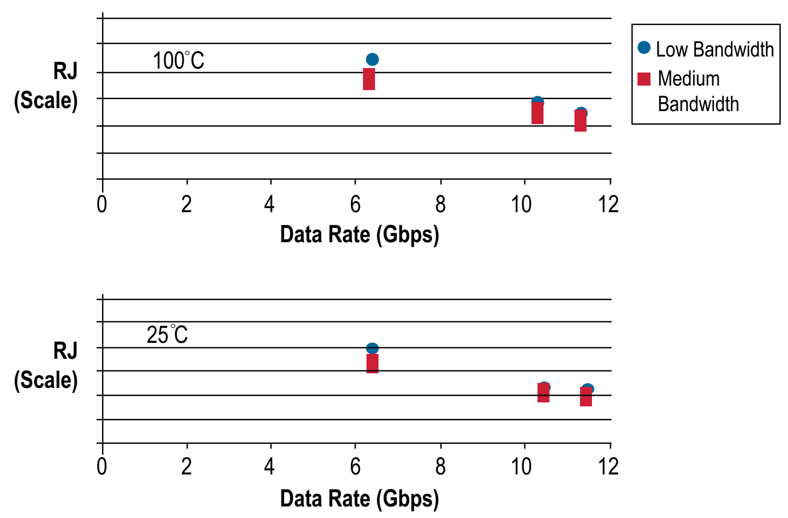
Equation 18. For RX

$$(DJ, RJ, DN, RN)_{RX} = LUT_{RX}(pat, f_d, v_{id}, f_{cr}, PVT)$$

To manage the large amount of possible combinations and corresponding measurement time for the TX and RX LUTs, only carefully selected discrete sets of combinations for the measurements are carried out, and a continuous coverage of LUTs is achieved via interpolation and extrapolation techniques. The environmental, neighboring activities, device-to-device, channel-to-channel variation influences are combined and degenerated to three classes: best, typical, and worst.

Figure 27 is a snapshot example for a subdimensional TX RJ LUT.

Figure 27. A Subdimensional Example of TX RJ LUT



The accuracy of the DCD, BUJ, RJ, DN, and RN in the LUTs are largely determined by the laboratory instruments, and that is ~150 fs for timing, and ~1 mv for voltage.

With PVT and LUTs capable of operating condition variability coverage, deterministic behavioral models, and channel S-parameters all incorporated with the full-waveform, hybrid, or statistical method, the JNEye tool have achieved the goal of modeling variability for the high-speed link simulation.

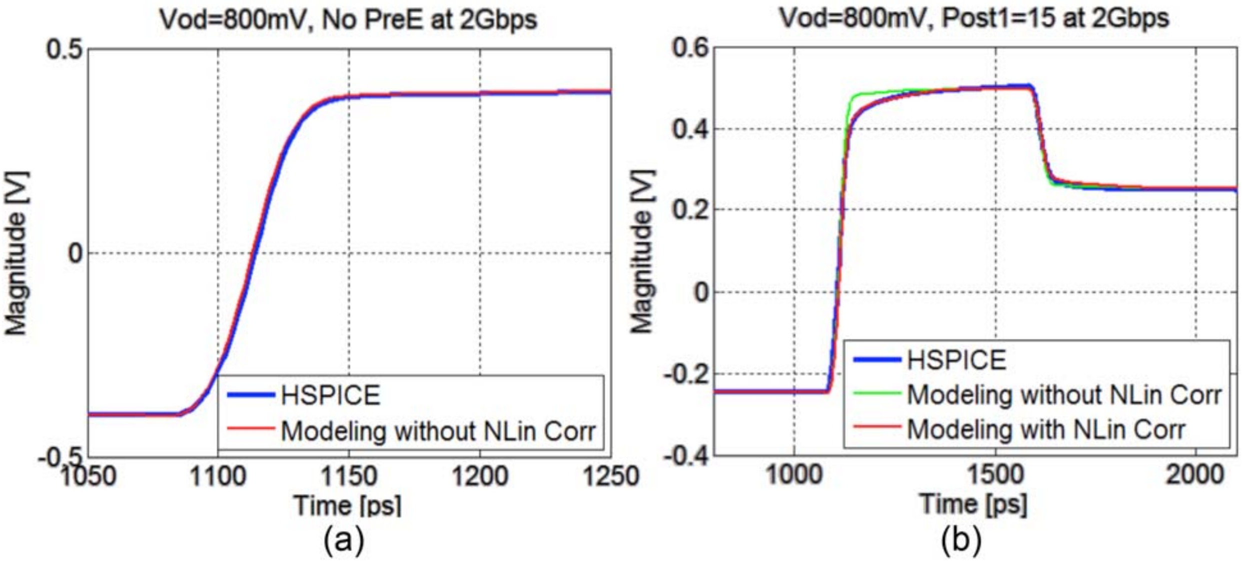
Accuracy and Correlation

An important performance merit for a simulator is its accuracy. For a high-speed simulator, the ultimate golden reference used to determine the accuracy or correlation is the actual measurements. There are two aspects associated with this subject: one is accuracy improvement, and the other is validation accuracy. Both are related and will be discussed in this section.

This section discusses the case study of improving the accuracy for the TX driver and FIR equalization model. In behavioral modeling, the basic TX driver and FIR filter consists of two steps. The first step is to model its edge-shaping linear filter without the FIR filter. The second step is to model the FIR behavior, namely the FIR voltage level settings versus tap coefficients. When the FIR filter is enabled, the behavior of the TX driver plus the FIR filter will have nonlinear effects, such as the peak current limitation of the IC [(2)]. We call this process “static” correction (over LTI) which comprehends the nonlinear effect.

This example is illustrated in Figure 28. Figure 28(a) shows two waveform zoom-ins from a behavioral model versus a HSPICE model without the FIR filter turned on. We get a good correlation between these two, indicating that the LTI system works well in modeling the driver. Figure 28(b) shows three waveform zoom-ins when the FIR filter is turned on with the first post-tap. You will notice that there is a good correlation between the behavioral model, with nonlinearity accounted for, and the HSPICE model. You will also notice a not-so-good correlation between the behavioral model, without nonlinearity accounted for, and the HSPICE model.

Figure 28. TX Waveforms from LTI Behavior vs. HSPICE



TX Waveforms from LTI Behavior (Red) vs. HSPICE

TX Waveforms from LTI Behavior (Red), vs. Behavior Accounts for Nonlinearity (Green), vs. HSPICE

In Figure 28, the accuracy or correlation improvement via nonlinear modeling was done based on a lower data rate of 2 Gbps. It turns out that this nonlinear effect is data-rate dependent, and simple, “static” nonlinear modeling does not work well at a higher data rate. What we need is data-rate dependent or “dynamic” nonlinear modeling. Figure 29 shows an example of how “dynamic” nonlinear modeling improves the accuracy or correlation of the behavioral model waveform prediction versus the reference obtained via measurement by an oscilloscope using the average mode, at a higher data rate of 14.1 Gbps. Notice that the “static” nonlinear modeling developed from a lower data rate does not extend well to a higher data rate.

Figure 29. Waveform Predictions with Behavior Model Using "Static" Nonlinear Modeling, vs. "Dynamic" Nonlinear Modeling, vs. Measurement

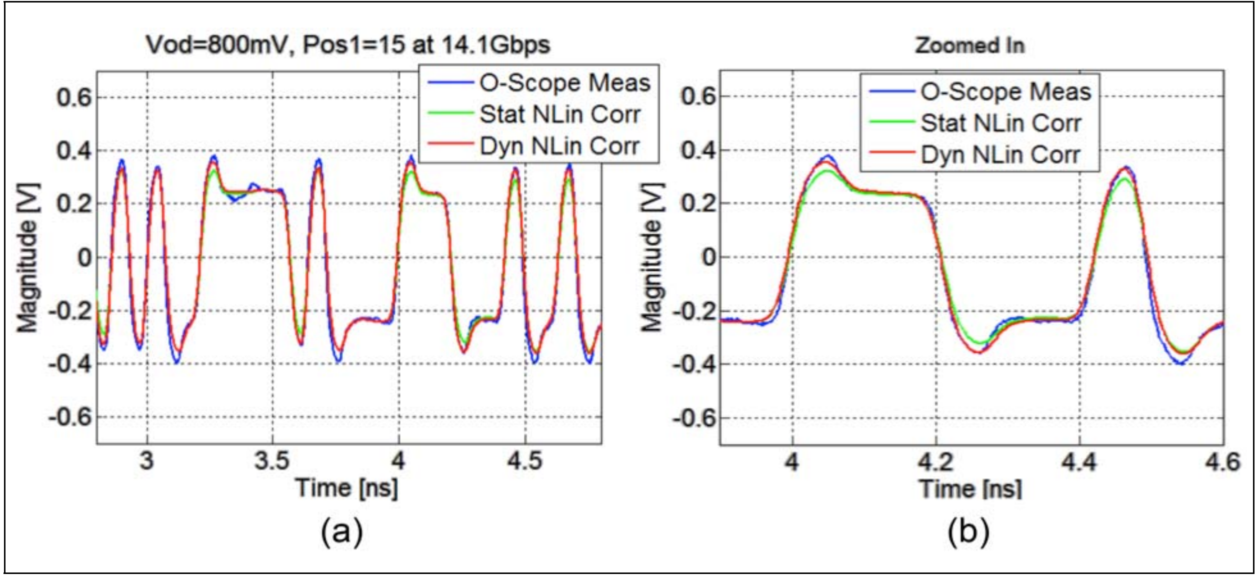


Figure 28 and Figure 29 show two case study examples demonstrating the accuracy or correlation improvement via developing advanced behavioral models for the TX driver and FIR filter, and their comparisons with HSPICE. This procedure needs to be done in an iterative manner. This technique and other new techniques beyond the LTI system need to apply to other circuit blocks of the link in order to achieve good accuracy and correlation.

Figure 30 shows a summary of the general behavioral model and the validation procedure to achieve good accuracy and correlation.

Figure 30. A Flowchart for Accurate or Correlated Behavioral Model or Simulator Development

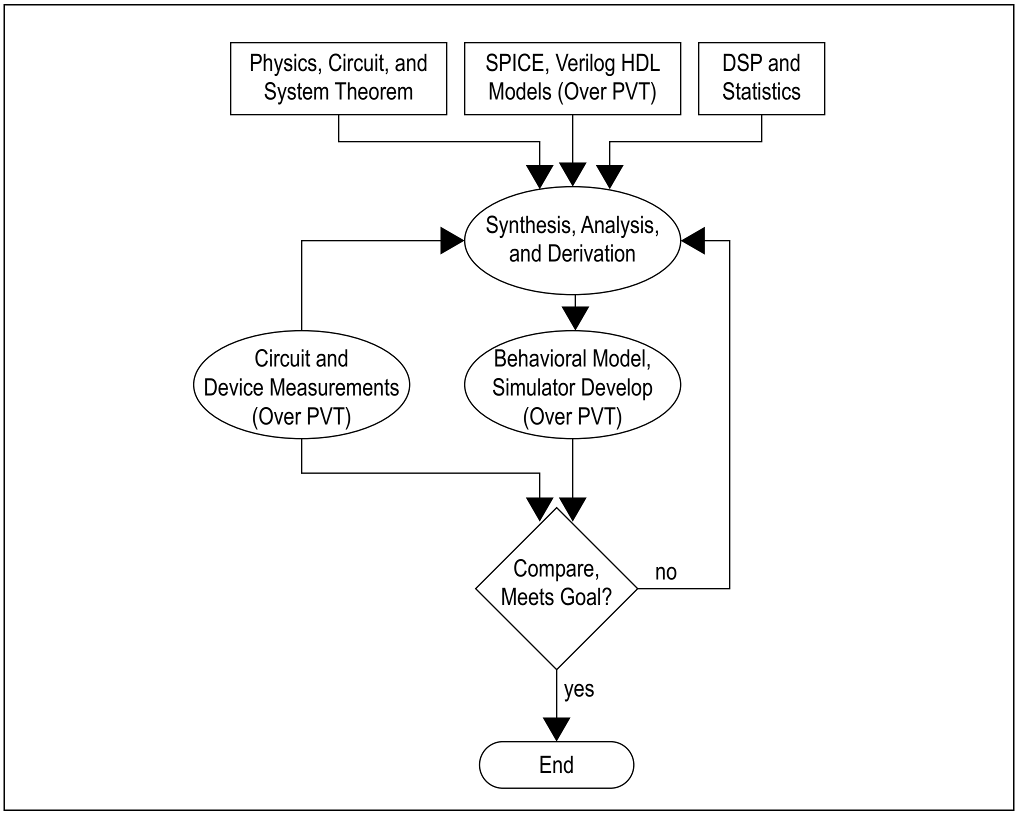

Physics, Circuit, and System Theorem
SPICE, Verilog HDL Models (Over PVT)
DSP and Statistics
Synthesis, Analysis, and Derivation
Circuit and Device Measurements (Over PVT)
Behavioral Model, Simulator Develop (Over PVT)
Compare, Meets Goal?
no
yes
End

Figure 31 and Figure 32 show two measured eye diagrams versus two simulated ones using the hybrid method overlaying on each other, where the TX FIR filter is enabled, with jitter, noise, and nonlinear effects all accounted for. The comparison data clearly shows a good match, indicating good accuracy or correlation.

Figure 31. Measured Eye Diagrams Overlayed with Simulated Ones for 6.5 Gbps and 14.1 Gbps when TX FIR Filter Is Enabled

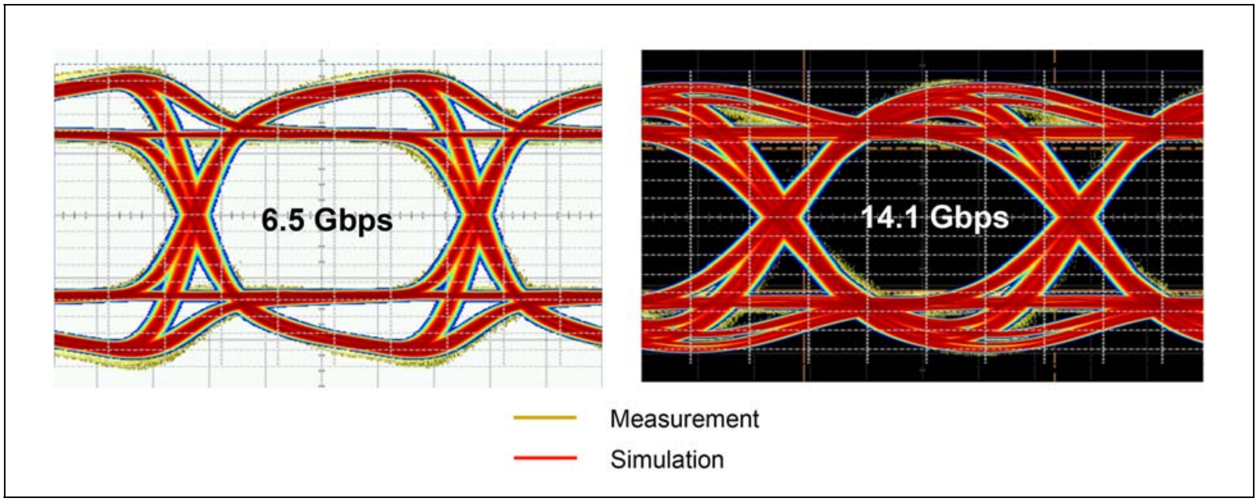
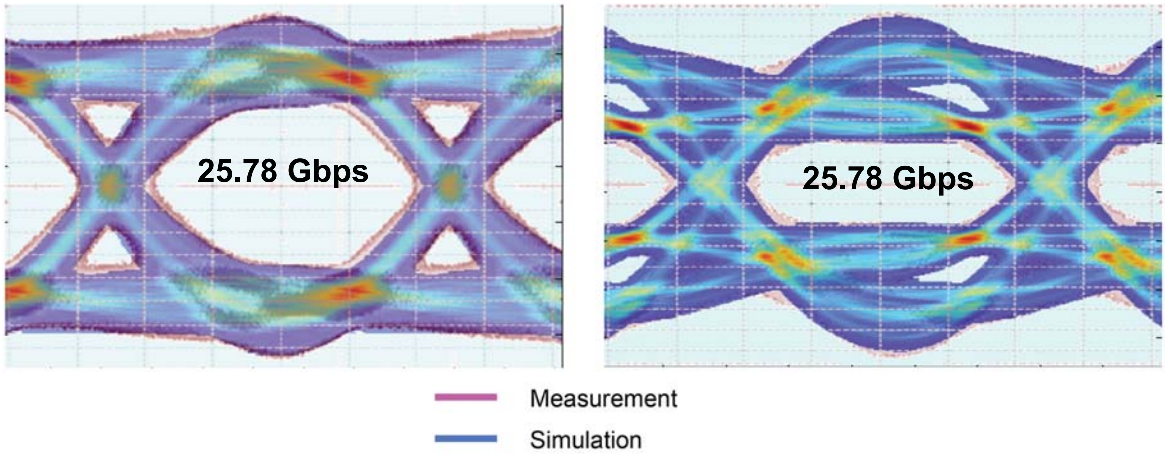


Figure 32. Measured Eye Diagrams Overlayed with Simulated Ones for 25.78 Gbps with Different TX FIR Filter Values



Conclusion

As HSIO and serial link data rates keep increasing, the requirements for accuracy and advanced modeling and simulation techniques have become more demanding. Traditional statistical link simulation methods have limited capabilities in handling these emerging requirements.

With Altera's JNEye tool, we introduced two new methods—full waveform and hybrid—with the full-waveform method in time domain, capable of modeling most of the circuit blocks within a link, and the hybrid method in both time and statistical domains. The JNEye tool also improves the accuracy of statistical link simulations. Time-domain full-waveform and hybrid signaling architecture mimic the actual circuit behavior. As such, the JNEye tool is able to model all the TX and RX subcomponents and their interactions completely and accurately, including the TX FIR filter, driver, package, CG/PLL, reference clock, and associated noise, and RX buffer, CTLE, DFE, CR, and DR. At the link level, the JNEye tool provides various optimization or adaptation methods appropriate for different channel characteristics.

The JNEye tool also provides comprehensive channel modeling and simulation, comprehending all the channel impairments, including insertion loss, crosstalk, and return-loss. Both synchronous and asynchronous crosstalk can be modeled and simulated accurately. Reflections due to return-loss and impedance variations and discontinuities are modeled and simulated to higher orders and warrants the best accuracy. In addition, the JNEye tool also provides a new method that models the deterministic IC/device PVT variations based on a bank of SPICE models, and non-deterministic jitter and noise PVT variations based on LUTs constructed from actual measurements under various operating conditions. Finally, we discussed case study examples demonstrating accuracy and correlation improvement via developing advanced behavioral models.

As the industry starts to deploy Nx25G high-speed links today and Nx50G links in the future, new simulation requirements and challenges will emerge. Continued advancements and innovations are required and expected for high-speed link simulation technology.

Further Information

1. ITRS roadmap, ITRS, 2011 Edition:
   www.itrs.net/Links/2011ITRS/Home2011.htm
2. M. Li and S. Shumarayev, *"Emerging Standards at ~ 10 Gbps for Wireline Communications and Associated Integrated Circuit Design and Validation,"* IEEE Custom Integrated Circuits Conf. Procedings, pp. 105-112, 2009.
3. *"A 28 Gbps 4-Tap FFE/15-Tap DFE Serial Link Transceiver in 32 nm SOI CMOS technology,"* IEEE Int. Solid-State Circuits Conference Digest, pp. 324-325, 2012.
4. B. Casper, M. Haycock, and R. Mooney, *"An Accurate and Efficient Analysis Method for Multi-Gb/s Chip-to-Chip Signaling Schemes,"* IEEE Very Large Scale (VLSI) Circuits Symp, pp. 54–57, 2002.
5. V. Stojanovic and M. Horowitz, *"Modeling and Analysis of High Speed Links,"* IEEE Custom Integrated Circuits Conf. Proceding, pp. 589–594, 2003.
6. A. Sanders, M. Resso, and J. D'Ambrosia, *"Channel Compliance Testing Utilizing Novel Statistical Eye Methodology,"* DesignCon 2004.
7. IBIS-AMI standard website:
   www.vhdl.org/ibis
8. B. Casper et al., *"Future Microprocessor Interfaces: Analysis, Design and Optimization,"* IEEE Custom Integrated Circuits Conf. Procedings, pp. 479–486., 2007.
9. G. Balamurugan, B. Casper, J. Jaussi, M. Mansuri, F. O'Mahony, and J. Kennedy, *"Modeling and analysis of high-speed I/O links,"* IEEE Transactions on Advanced Packaging, vol. 32, no. 2, 237–247., 2009.
10. M. Li, *"Jitter, Noise, and Signal Integrity at High-Speed,"* Prentice Hall, 2007.
11. M. Shimanouchi, M. Li, H. Wu, "*Comparison of Two Statistical Methods for High Speed Serial Link Simulation,"* Designcon, 2013.
12. F. M. Gardner, *"Phaselock Techniques,"* John Wiley & Sons, 2nd Edition, 1979.
13. M. Li, and M. Shimanouchi, *"New Hybrid Simulation Method for Jitter and BER in High-Speed Links,"* Designcon, 2011.
14. S. F. Adam, "*Microwave Theory and Applications,"* Prentice-Hall, 1969.
15. M. Li, M. Shimanouchi, H. Wu, *"Advancements in High-Speed Link Modeling and Simulation,"* IEEE Custom Integrated Circuits Conf., 2013.

Acknowledgements

- Mike Peng Li, Fellow, Altera Corporation
- Hsinho Wu, Principal Engineer, Altera Corporation
- Masashi Shimanouchi, Principal Engineer, Altera Corporation
- Salman Jiva, Sr. Product Marketing Manager, Altera Corporation
- Udayan Sinha, Product Marketing Engineer, Altera Corporation
- Lux Joshi, Product Marketing Manager, Altera Corporation

Document Revision History

Table 4 lists the revision history for this document.

Table 4. Document Revision History

Date	Version	Changes
December 2013	1.0	Initial release.