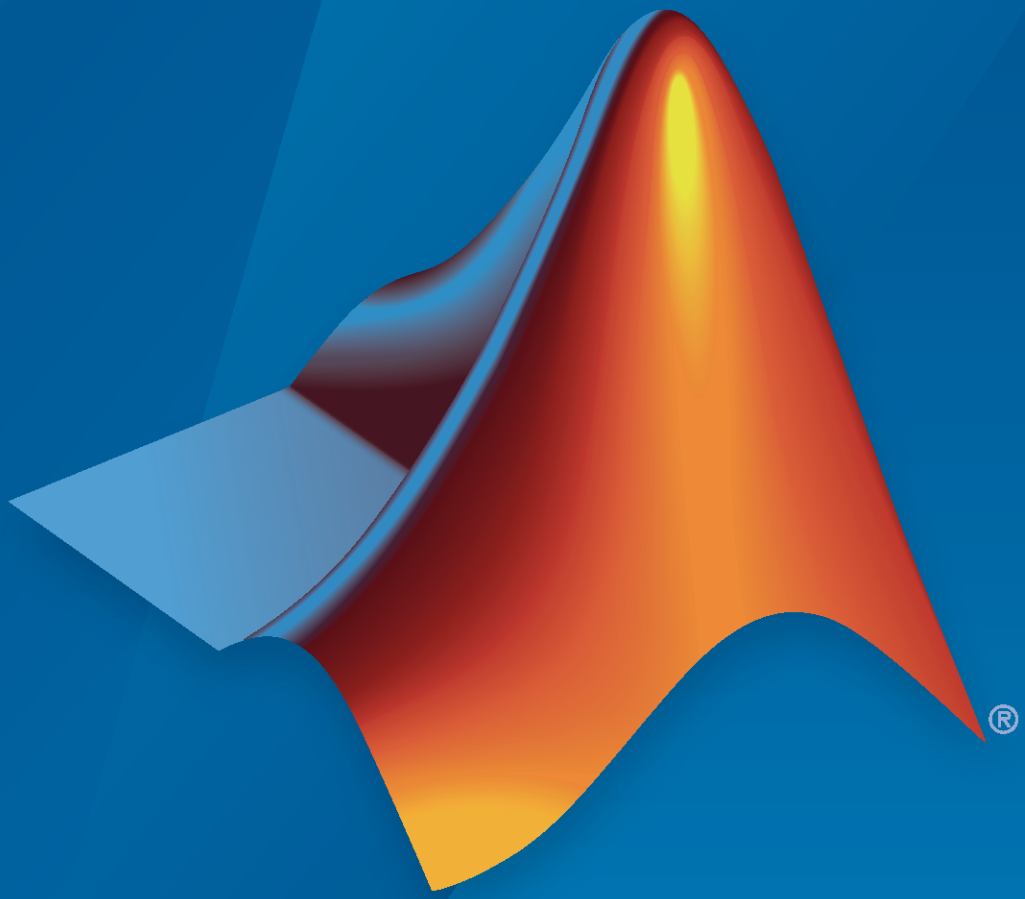
Signal Integrity Toolbox™

Getting Started



MATLAB®

R2022b

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Signal Integrity Toolbox™ Getting Started

Revision History

September 2021	Online only	New for Version 1.0 (Release 2021b)
March 2022	Online only	Revised for Version 1.1 (Release 2022a)
September 2022	Online only	Revised for Version 1.2 (Release 2022b)

Contents

1 Getting Started with Signal Integrity Toolbox

2 Getting Started Examples

1

Getting Started with Signal Integrity Toolbox

Signal Integrity Toolbox Product Description

Simulate and analyze high-speed serial and parallel links

Signal Integrity Toolbox™ provides functions and apps for designing high-speed serial and parallel links. You can generate experiments covering multiple parameters, extract design metrics, and visualize waveforms and results. You can predict operating margins and link performance by analyzing transmitter, receiver, and channel interactions.

The toolbox supports standard-compliant IBIS-AMI models for statistical and time-domain simulation to analyze equalization and clock recovery. You can describe the channel using multiport S-parameter data, IBIS, HSPICE, and analytical models.

Signal Integrity Toolbox lets you analyze waveforms and eye diagrams and measure channel quality while observing effects such as ISI, jitter, and noise. You can analyze the channel in the frequency domain for insertion loss, return loss, and crosstalk, and verify compliance with industry standards including IEEE 802.3, OIF, PCIe, and DDR.

Before layout, you can evaluate tradeoffs and optimize parallel and serial links for cost, performance, reliability, and compliance. You can then perform post-layout verification of the system and correlate simulation results with measurement data.

Get Started with Serial Link Designer

Designers embarking on projects with 3+ Gbps serial links are confronted with a confusing array of tools, techniques and terminology. Traditional time-domain simulation using SPICE alone does not provide an adequate estimate of the Bit Error Rate (BER) of a serial link, which is the key metric used to assess reliability.

When simple transistor level models are used for the active devices, compute time considerations typically limit simulations to a few thousand bits. When more complex models are used that model transmitter and receiver equalization, compute time increases, limiting practical simulations to hundreds of bits at best.

In high-speed serial links there is processing before the I/O stage (transmit equalization) and after the input buffer (receive equalization and clock recovery). The signal of importance is not at the pin or die pad of the receiver, it is the decision point of the receiver, an internal point that is after the receive equalization. In addition, the clock recovery circuitry in the receiver must be taken into account to estimate BER accurately.

In order to estimate BER for such a receiver the equalization and clock recovery need to be modeled, and simulators need to access the waveform at the decision point. The IBIS-AMI (Algorithmic Modeling Interface) standard was developed to provide this in a way that preserves vendor IP and allows inter-operability between EDA tools and vendor models. The IBIS-AMI standard defines an interface to an algorithmic model that is an executable program. The analog portion of a model uses the existing IBIS IV and VT curves. The algorithmic model is used to model equalization, clock recovery and device optimization.

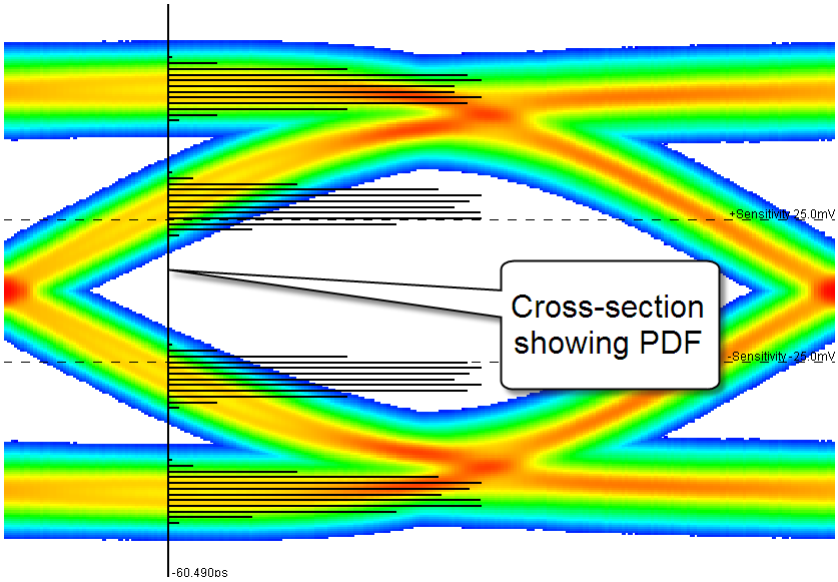
There are three main challenges to estimating BER:

- **ISI (Inter-Symbol Interference)** — looked at in the time domain, more bits interact with each other, so more combinations of bits need to be analyzed. Looked at in the frequency domain, a wider bandwidth with more loss variation means equalization is required.
- **Clock to data timing (at the receiver decision point)** — has less margin, so timing perturbations must be carefully accounted for and analyzed.
- **Crosstalk** — increases with frequency.

The **Serial Link Designer** app uses a combination of statistical and time domain techniques to estimate the BER of a serial link in a fraction of the time it would take using transient non-linear simulation. The app also uses specific project settings and libraries to organize the simulations.

Statistical Analysis of Linear Time Invariant System

Statistical analysis uses a recursive convolution engine to calculate the eye diagram PDF (probability distribution function) from the pulse response of an LTI (linear time invariant) system. This is called a statistical eye.



The statistical eye shows the average effect of all messages of a given length. This is a very fast way to explore a wide range of design options. The major factors affecting performance are handled in the following way:

- **Inter-Symbol Interference** — The effects of inter-symbol interference are reflected directly in the PDF of the eye diagram. This is a very accurate and efficient way to calculate the inter-symbol interference due to all possible combinations of bits, even for very long message lengths.
- **Clock to Data Timing** — The PDF of the clock can be combined with the PDF of the eye diagram to produce a very rigorous BER estimate.
- **Crosstalk** — The PDF of the crosstalk can be convolved with the PDF of the eye diagram to produce a composite PDF which accurately and completely reflects the effects of crosstalk. When combined with the clock to data timing mentioned above, the resulting BER estimate is both rigorous and complete.

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

Peak distortion analysis can be used to derive the worst case data pattern for the system.

Time Domain Analysis of Linear Time Invariant System

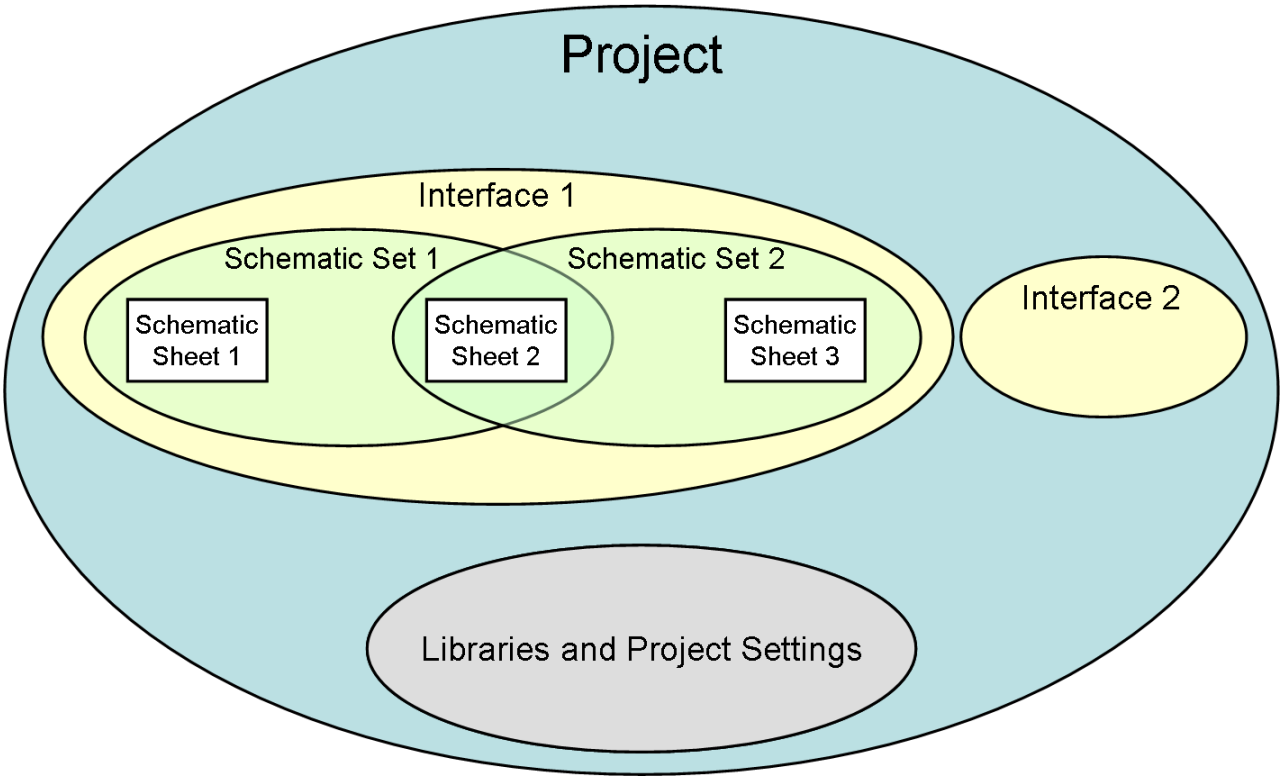
Time Domain analysis is a semi-analytic analysis method in which up to a few million bits of the desired signal are simulated in the time domain, and then the effect of the noise, crosstalk and clock phase noise is estimated using statistical techniques. This has the advantage of allowing time variant and nonlinear effects to be studied, such as pattern dependence in the clock recovery and equalization control loops to be studied while maintaining the efficiency and dynamic range of statistical calculations. The ability to simulate specific data patterns also makes this approach useful for correlating with measured data or determining worst case performance. The major factors affecting performance are handled in the following way:

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

Organization of Simulations

The **Serial Link Designer** app consists of four major project elements:

- **Project** — A project typically represents a full system that may include one or more boards and one or more bus types. For example, a system may have XAUI and PCIe serial links, or serial links and parallel bus memory channels.
- **Interface** — An interface is one subset of the system design, for example the XAUI serial channel or a parallel bus memory. A project can have one or more interfaces. All work in the **Serial Link Designer** app is performed in an interface of a project.
- **Schematic Sheet** — A schematic sheet is a graphical representation of the serial channel, captured using elements such as designators, transmission lines, s-parameters, vias and more. A schematic sheet is always a part of a schematic set.
- **Schematic Set** — A schematic set is a collection of schematic sheets in an interface. Each schematic set may represent a different configuration of the Interface. In the **Serial Link Designer** app an interface can have one or more schematic sets.



The app also uses some specific terms to define certain elements that are useful:

- **Transfer net** — Each signal type in an interface is called a transfer net, and consists of the drivers, receivers, electrical conductors and passive components involved in transmitting that signal type. There may be multiple instances of a transfer net in an interface (example: data lines) or there could be a single instance (example: clock).

  Transfer nets are common to both pre-layout analysis and post-layout verification. In pre-layout analysis, each sheet has an associated transfer net with the same name as the sheet. In post-layout verification, extended nets are assigned to transfer nets.
- **Solution space** — Variations of parameters such as voltage, temperature, or process for each transfer net are organized into a solution space.
- **Designator** — Within a transfer net, drivers and receivers are referred to as designators. In other words, a designator is either a starting point or an end point for a transfer net and is therefore a point where timing analysis is to be performed.

Libraries

The **Serial Link Designer** app comes with libraries of technology and generic models. The technology and generic library models can be assigned to schematic designators. The library elements include I/O buffer models, transmission line models, IBIS models, SPICE package models, and S-Parameter files. By default, new projects and interfaces automatically reference the project's local library as well as the installation library.

See Also

Serial Link Designer | **Signal Integrity Viewer**

Related Examples

- “Analyze Serial Links with Serial Link Designer” on page 2-6

Get Started with Parallel Link Designer

A parallel link interface is usually a bus with many data sources and/or destinations. You need to evaluate clock to data timing for each combination of driver and receiver on the parallel interface. It is also necessary to examine the shape of the waveform for each such combination. Problems such as inadequate amplitude, inadequate slew rate, or ringing can cause unreliable data transfer while problems such as overshoot can cause premature failure.

Each interface design must be evaluated both before and after the system and PC boards are designed. The pre-layout analysis is required if the design is to have any chance of working, and the post-layout analysis is required to make sure nothing was missed in the PC board design process. The net result is that there is usually a massive number of possibilities to analyze. These possibilities are the compounded combination of every combination of driver and receiver, every phase of the interface protocol, semiconductor process corners, PC board process corners separately for each layer of each PC board, passive component value, power supply voltages, temperature, and pre-layout and post-layout analysis. Failure to evaluate each and every one of these combinations increases the probability that problems will be identified much later in the design cycle, when correcting them is much more expensive.

The Parallel Link Designer app automates the running of the simulations by providing a graphical environment to set up the simulations and a standardized way in which to organize the model libraries. You can sift through the mass of data to identify information which is critical to the design in several ways:

- A waveform analyzer evaluates output waveforms and identifies any waveforms which fail to meet predefined requirements.
- A timing analyzer evaluates clock to data timing and identifies cases where the clock to data timing either fails to meet requirements or has minimal timing margin.
- A separate spreadsheet is provided for waveform analyses and timing analyses. Each such spreadsheet organizes the data from summary information to the details of individual cases, thus making it practical to find detailed information about individual failing or marginal cases.
- A waveform viewer makes it convenient to examine results graphically. Separate display panels are provided for waveforms and tabular results. Waveforms can be viewed in either the time domain or the frequency domain. Tabular results are presented in scatter plot format.

Organization of Simulations

To automate the simulation and analysis process, the **Parallel Link Designer** app organizes the work in a generic structure:

- **Project** — A project is the top-level organization in the **Parallel Link Designer** app. All the work related to a particular product or system is usually contained in a project. A project can contain multiple interfaces.
- **Interface** — Interface is the collection signals such as clock, data or strobe used to implement the physical layer protocol from the core one IC to the core of another IC. Example interfaces are: DDR, QDR, RLDRAM, PCIe, or XAUI.
- **Schematic Sheet** — A schematic sheet is a graphical representation of the serial channel, captured using elements such as designators, transmission lines, s-parameters, vias and more. A schematic sheet is always a part of a schematic set.

- **Schematic Set** — A schematic set is a collection of schematic sheets in an interface. Each schematic set may represent a different configuration of the Interface. In the **Serial Link Designer** app an interface can have one or more schematic sets.

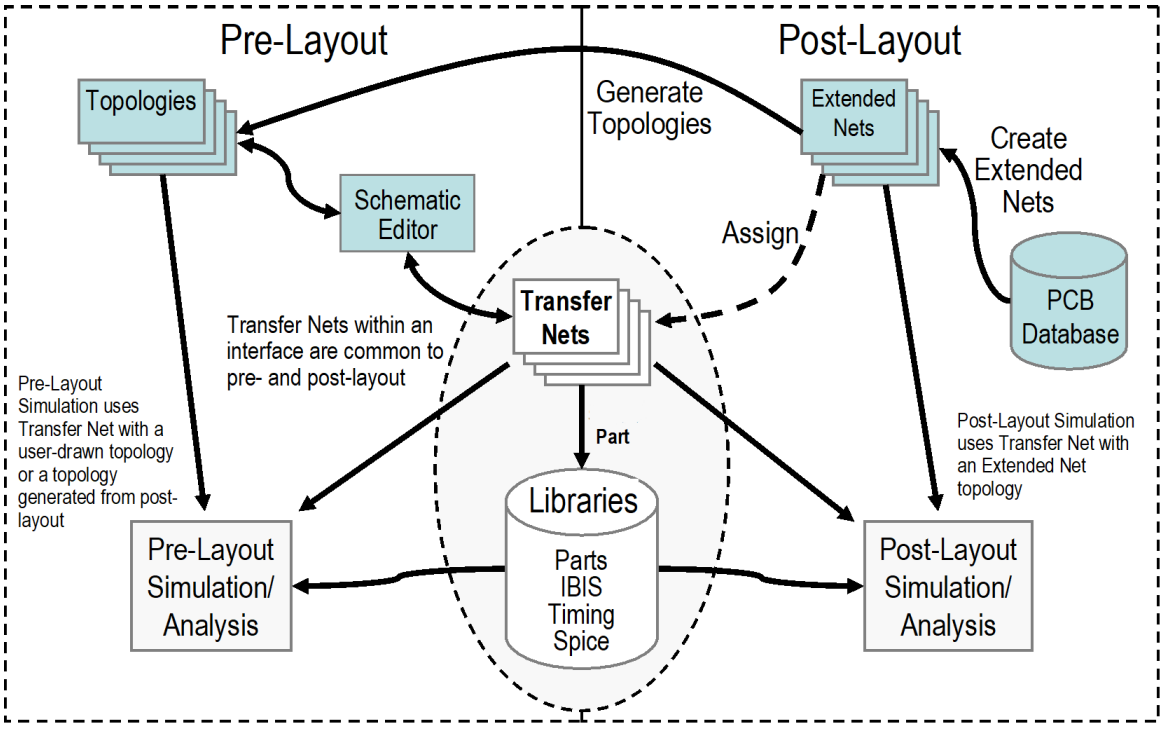
The app also uses some specific terms to define certain elements that are useful:

- **Transfer net** — Each signal type in an interface is called a transfer net, and consists of the drivers, receivers, electrical conductors and passive components involved in transmitting that signal type. There may be multiple instances of a transfer net in an interface (example: data lines) or there could be a single instance (example: clock).

  Transfer nets are common to both pre-layout analysis and post-layout verification. In pre-layout analysis, each sheet has an associated transfer net with the same name as the sheet. In post-layout verification, extended nets are assigned to transfer nets.
- **Solution space** — Variations of parameters such as voltage, temperature, or process for each transfer net are organized into a solution space.
- **Designator** — Within a transfer net, drivers and receivers are referred to as designators. In other words, a designator is either a starting point or an end point for a transfer net and is therefore a point where timing analysis is to be performed.

Pre-Layout and Post-Layout Analysis

In pre-layout analysis, each transfer net is described on a separate schematic sheet and is simulated in a separate simulation run. The circuit element values in a transfer net schematic can be parameterized so that many variations can be evaluated without having to change the schematic. Also, a transfer net schematic can contain crosstalk interferers, so that the analysis of the transfer net can be complete.



The purposes of pre-layout analysis are to:

- evaluate multiple design options to identify proper I/O models, topology and termination,
- identify crosstalk budgets and associated timing penalties,
- and define layout rules for PCB design.

In post-layout analysis, each net on a PCB is called a CAD net. CAD nets can be combined with drivers, receivers, packages, discrete components and connector pins to form an extended net. An extended net contains designators and reaches from one IC to another. Extended nets can also traverse multiple PCBs. The **Parallel Link Designer** app provides convenient mechanisms for identifying extended nets and assigning them to transfer nets for all major PCB flows.

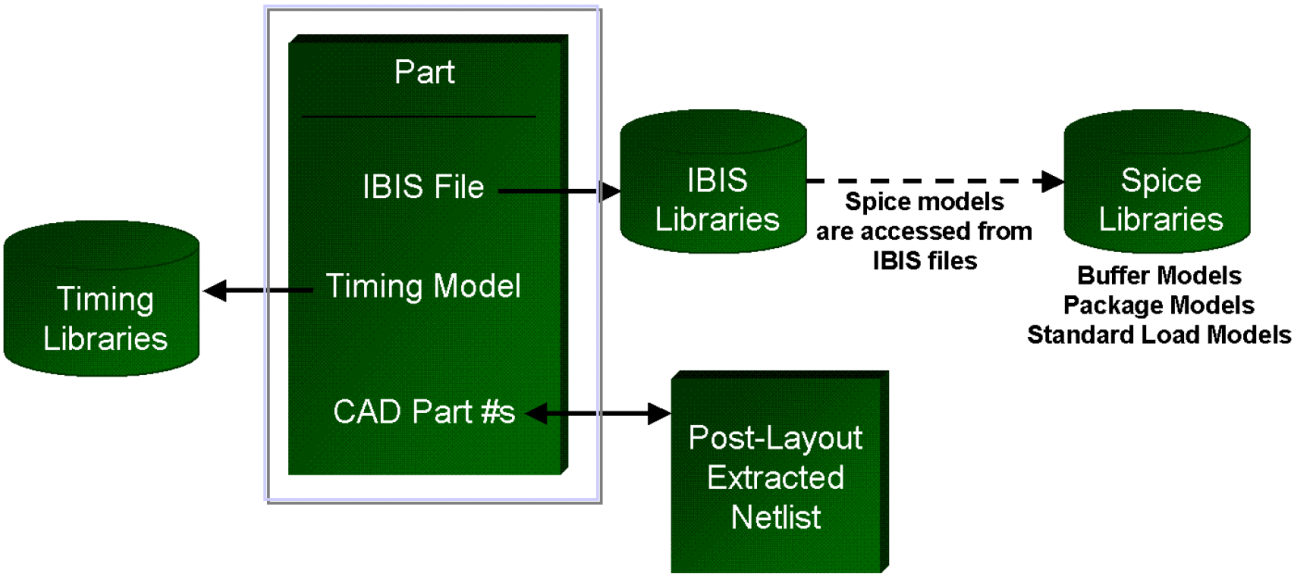
The purposes of post-layout analysis are to:

- validate PCB routing,
- validate PCB signal integrity and timing margins,
- and validate crosstalk margins.

One of the advantages of the transfer net concept is that transfer nets can be re-used without modification from one phase of a project to another, or between projects. For example, if transfer nets were defined during the pre-layout phase of a project, then those definitions can be re-used without modification during the post-layout phase. Conversely, if a transfer net has been defined in the post-layout phase, it can be re-used directly in a pre-layout analysis

Libraries

There are usually a number of different types of data associated with a given component, including electrical data such as HSPICE or IBIS models, timing models, and physical data such as CAD part number and pinout. For each component, all of this data is kept together in a Part, and every designator in a schematic must have a Part.



A Part has a unique identifier and contains references to an IBIS file, a timing model, and a CAD part number. An IBIS file, in turn, may contain:

- mapping information between a part’s logical pin name, physical pin number, and I/O buffer model name,
- behavioral I/O buffer models (for example, IV/VT curves),
- references to HSPICE models and advanced package models,
- and waveform processing levels which are MathWorks proprietary extensions to the IBIS standards.

The **Parallel Link Designer** app includes a 2D field solver which can be used to create accurate models of lossy transmission lines. There are well defined places to put libraries either for a single project or to be shared among many projects. The same library elements are used in pre-layout and post-layout analysis.

See Also

Parallel Link Designer | Signal Integrity Viewer

Related Examples

- “Analyze Parallel Links with Parallel Link Designer” on page 2-2

Get Started with Signal Integrity Kits

Signal integrity kits allows users to understand and implements different interface technologies. Using these kits, you can quickly implement and validate high speed interfaces to meet specified compliance or meet bit error rates (BERs) as a part of a design requirements.

You can download, extract, and open Signal Integrity Toolbox kits by using the `openSignalIntegrityKit` function

There are three types of signal integrity kits.

Architectural Signal Integrity Kits

Use the architectural signal integrity kits and the **Serial Link Designer** or **Parallel Link Designer** app to set up a complete signal integrity environment for a specific interface technology. These kits include technology models for all components, generic topologies for each transfer net, a detailed setup for model characterization, channel compliance, and detailed documentation. You can also reconfigure these kits to meet specific requirements.

Architectural signal integrity kit technology models are set up with a single differential pin-pair representing each of the transfer nets. Furthermore, all topologies are set up as point-to-point, with little or no PCB termination applied. You will need to replace or edit the topologies and IBIS-AMI models with topologies and models that are appropriate to your application. An example of an architectural signal integrity kit is “Low-Power DDR5 Architectural Kit”.

Implementation Signal Integrity Kits

Implementation signal integrity kits are customized versions of the architectural signal integrity kits that uses design-specific simulation models and topologies (such as terminations or configurations.) These kits represent your exact interface implementation. Implementation kits support complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins. An example of an implementation signal integrity kit is “DDR5 Implementation Kit”.

Compliance Signal Integrity Kits

Use the compliance signal integrity kit to test the compliance of simulation models and topologies to a specific industry standard specification. These tests can include transmitter model compliance, receiver model compliance, channel compliance, and end-to-end compliance of a specific implementation as applicable. Each type of compliance may include one or more checks spanning eye masks, frequency domain masks, and other metrics. An example of a compliance signal integrity kit is “PCIe-5 Compliance Kit”.

See Also

`openSignalIntegrityKit`

Related Examples

- “PCIe-5 Compliance Kit”
- “DDR5 Implementation Kit”

- “Low-Power DDR5 Architectural Kit”

Getting Started Examples

Analyze Parallel Links with Parallel Link Designer

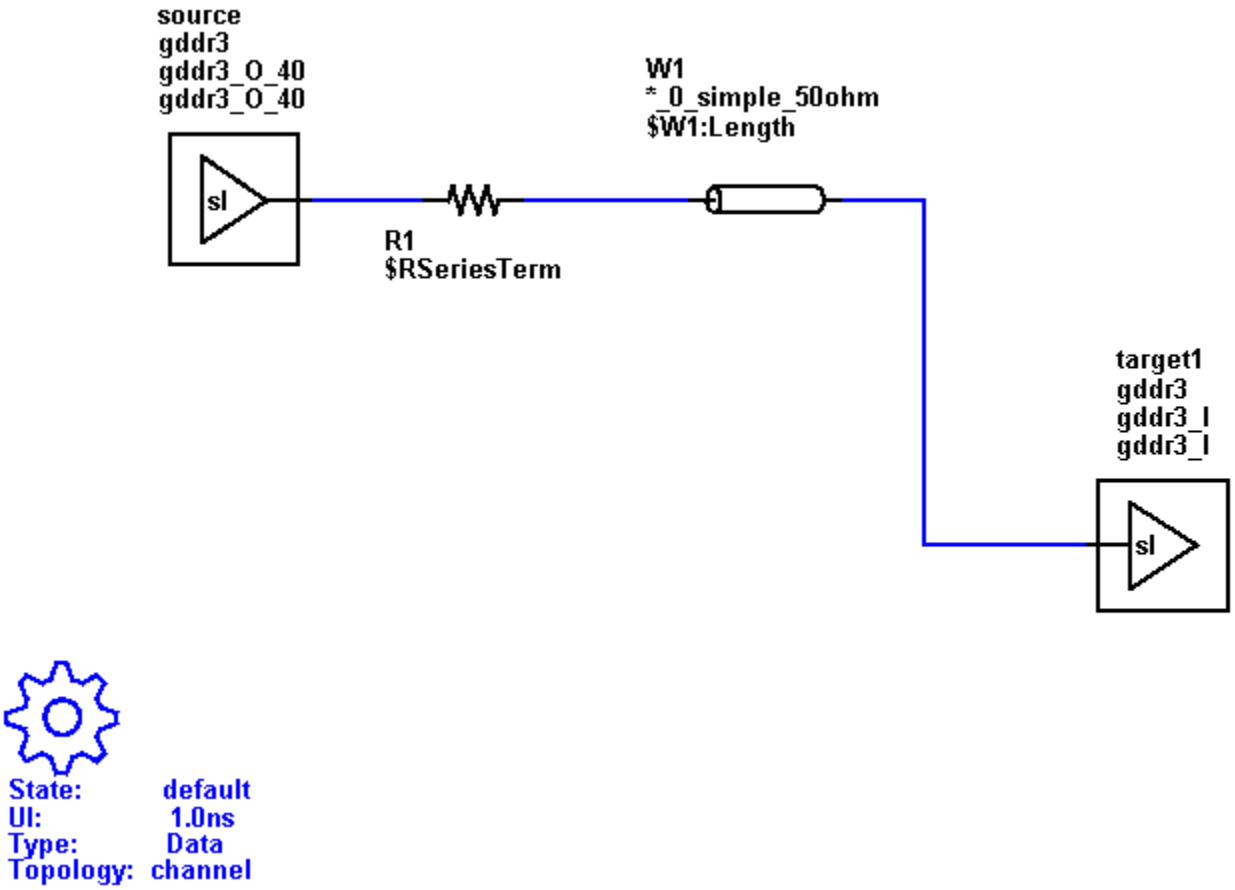
Learn how to configure and explore the design space of a parallel link and analyze the waveforms. You can sweep different parameters and evaluate their impact on your design's performance using the **Parallel Link Designer** app.

Create New Project

Open the **Parallel Link Designer** app.

```
parallelLinkDesigner
```

Create a new project by selecting **File > Project > New Project**. In the newly opened dialog box, name the project as `parallel_link`, the interface as `ddr`, and the schematic sheet as `channel`. Select the **Auto-Generate Topology** checkbox with **Technology Defaults**. Select **Technology Library Models** as `GDDR3` from the dropdown menu. The **Pre-Layout Analysis** tab shows the schematic sheet with a transmitter, a receiver, a resistor, and a transmission line with default values.



Setup Simulation Parameters

Double click on the gear ( ) icon in the **Pre-Layout Analysis** tab to launch the Sheet Simulation Control dialog box. Set the unit interval for the transmitter to 375 ps to by selecting `0.375ns - data_ddr_1333M` from the dropdown menu. Close the Sheet Simulation Control dialog box.

Double click the target1 symbol to open the Designator Element Properties dialog box. Set the **Model** for `target1` **Designator** to `gddr3_I_60u`.

Double click on the resistor symbol to launch the Resistor Element Properties dialog box. Check that **Sweep Resistance** parameter is enabled. Change the name of the **Resistance** parameter to `$tl_res`. Close the Resistor Element Properties dialog box.

Double click on the W-line symbol to launch the Lossy Transmission Line Element Properties dialog box. Check that **Sweep Length** parameter is enabled. Change the name of the **Length** parameter to `$tl_len`. Close the Lossy Transmission Line Element Properties dialog box.

In the Solution Space panel, select the **Variable** `$tl_len`. `$tl_len` already has 2in as **Value1**. Add the values 4in, 6in, 8in, and 10in to sweep the length of the transmission line.

Next, select the **Variable** `$tl_res`. Add the values 75 and 100 to sweep the resistance.

Solution Space: Sheet Options: Case Mode STAT Mode Tx Aggressor Par

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:
channel	Etch	Corner	List	Corners	TE (Typ)					
channel	Process	Corner	List	Corners	TT (Typ)					
channel	$tl_len	W Length	Soft Range	<none>	2.0in	4in	6in	8in	10in	
channel	$tl_res	Resistance	Soft Range	<none>	50ohm	75ohm	100ohm			

At the bottom right corner of the Solution Space panel, the **Base SPICE Simulation Count** should show 15. Save the changes.

Validate the schematic set by selecting **Run > Validate Current Schematic Set**. The validation should run without warning or errors.

Analyze Waveforms

To see the effects of sweeping the transmission line length and resistance, analyze the waveform.

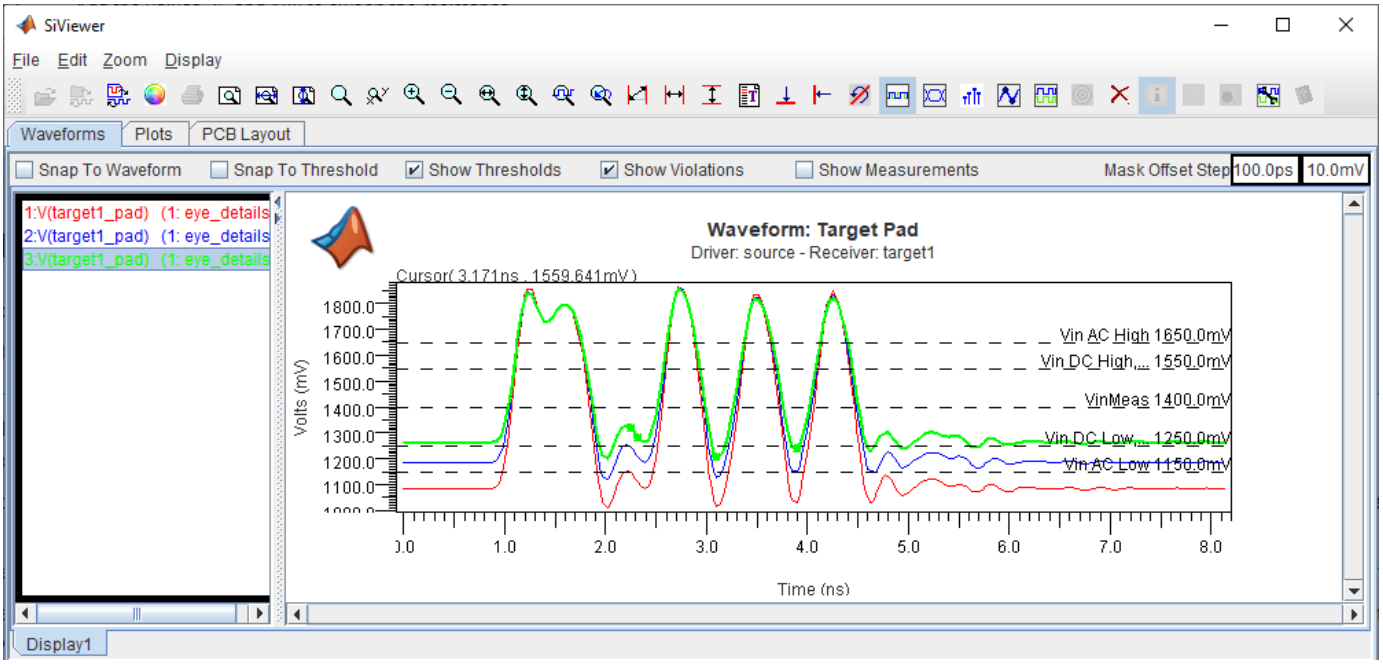
Run the simulation by selecting **Run > Simulate Selected**. In the Prelayout Simualtion dialog box, select **Validate**, **Generate Netlists**, **Run SPICE**, **Analyze Waveforms**, and **Autoload Results**. Click **Run** to start the simulation process.

When the analysis is finished the **SI Viewer** app launches and loads the analysis results. The table has one row per simulation. You can sort by any column by clicking on the column header.

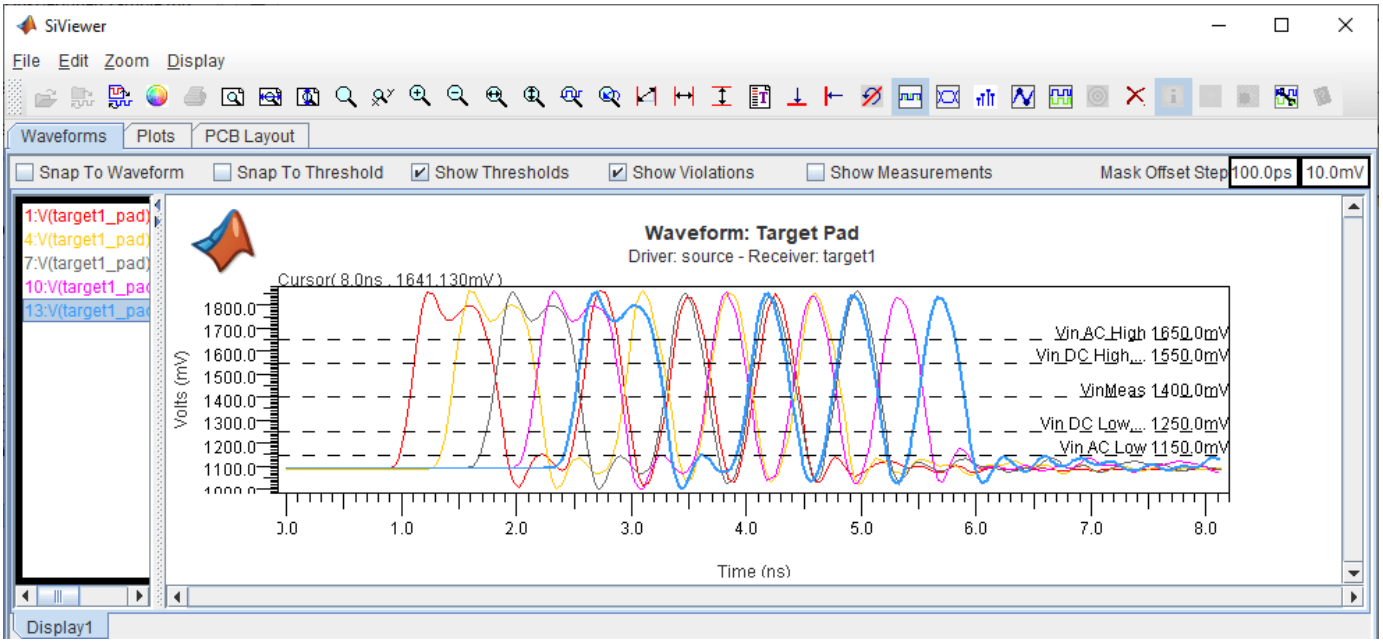
The Prelayout Channel Analysis dialog box shows that 29 warnings occured while running waveform analysis. Click the **Errors and Warnings** button and select the Analyze Waveform tab in the newly

opened Prelayout Errors and Warnings dialog box. The report says there are 29 waveform quality warnings.

To view the effect of changing the resistance, in the **SI Viewer** app, select the rows corresponding to transmission line length 2 in and resistor values 50 ohm, 75 ohm, and 100 ohm. Find the relevant rows by right clicking a row and selecting **Show Solution Space**. Select and right click all three rows and select **Show Waveform > Target Pad**.



Similarly, to view the effect of changing the transmission line length, select the rows corresponding to 50 ohm resistance and 2 in, 4 in, 6 in, 8 in, and 10 in length.



Close the **SI Viewer** app.

See Also

Parallel Link Designer | Signal Integrity Viewer

Related Examples

- “Configure DDR Controller with Two Memory Designators”

Analyze Serial Links with Serial Link Designer

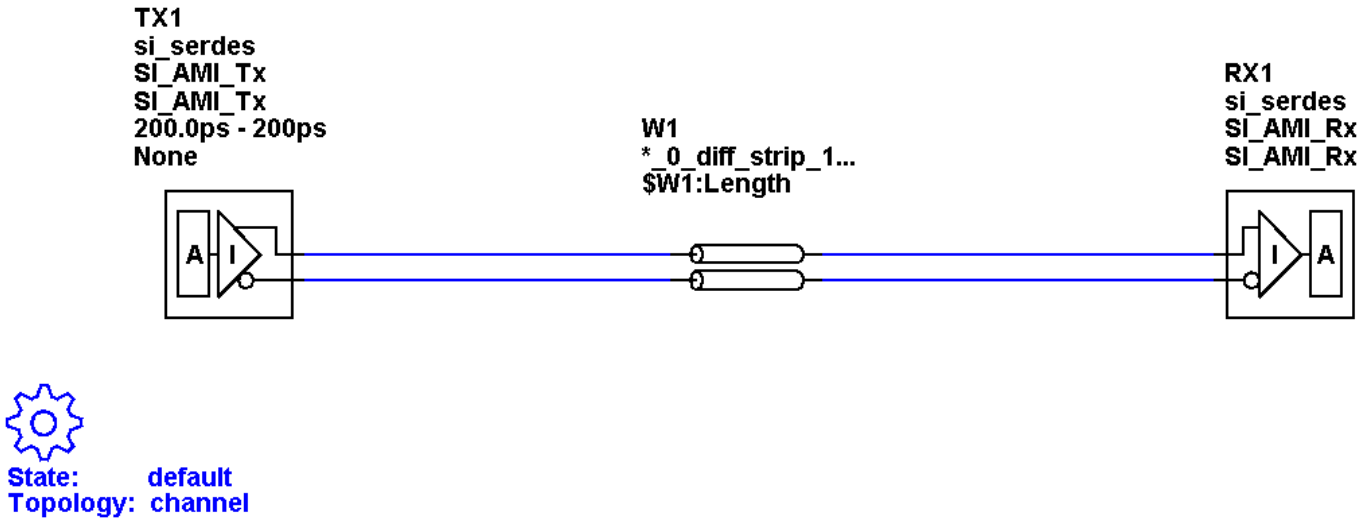
Learn how to configure and explore the design space of a serial link. You can run network characterization, statistical, and time domain analysis using the **Serial Link Designer** app. You can also sweep different parameters and evaluate their impact on your design's performance.

Create New Project

Open the **Serial Link Designer** app.

```
serialLinkDesigner
```

Create a new project by selecting **File > Project > New Project**. In the newly opened dialog box, name the project as `serial_link`, the interface as `serdes`, and the schematic sheet as `channel`. Select the **Auto-Generate Topology** checkbox with **Technology Defaults**. The **Pre-Layout Analysis** tab shows the schematic sheet with a transmitter, a receiver, and a transmission line with default values.



Setup Simulation Parameters

Double click on the TX symbol to launch the Designator Element Properties dialog box. Set the unit interval for the transmitter to `100` ps to by selecting `100.0ps - Serdes_10G` from the dropdown menu. Close the Designator Element Properties dialog box.

Double-click on the W-line symbol to launch the Lossy Transmission Line Element Properties dialog box. Check that **Sweep Length** parameter is enabled. Change the name of the **Length** parameter to `$tl_len`. Close the Lossy Transmission Line Element Properties dialog box.

In the Solution Space panel, select the **Variable** `$tl_len`. `$tl_len` already has `20`in as **Value1**. Add the values `16`in, `12`in, `8`in, and `4`in to sweep the length of the transmission line.

Next, select the **Variable** `$RX1:dfe.mode`. Add the values `fixed` and `adapt` to sweep the DFE mode.

Solution Space: Sheet Options: Case Mode

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:
channel	RX1:dfe.taps.3	Tap	AMI Range	RX1:Tap	0					
channel	RX1:dfe.taps.4	Tap	AMI Range	RX1:Tap	0					
channel	RX1:dfe.taps.5	Tap	AMI Range	RX1:Tap	0					
channel	RX1:dfe.mode	String	AMI List	<none>	off	fixed	adapt			
channel	$tl_len	W Length	Soft Range	<none>	20.0in	16in	12in	8in	4in	
channel	TX1:tap_filter.-1	Tap	AMI Range	TX1:Tap	0					
channel	TX1:tap_filter.0	Tap	AMI Range	TX1:Tap	1					

At the bottom right corner of the Solution Space panel, the **Simulation Count** should show 15. Save the changes.

Validate the schematic set by selecting **Run > Validate Current Schematic Set**. The validation should run without warning or errors.

Network Characterization

To see the effects of sweeping the transmission line length and DFE mode on the physical channel characteristics, run network characterization.

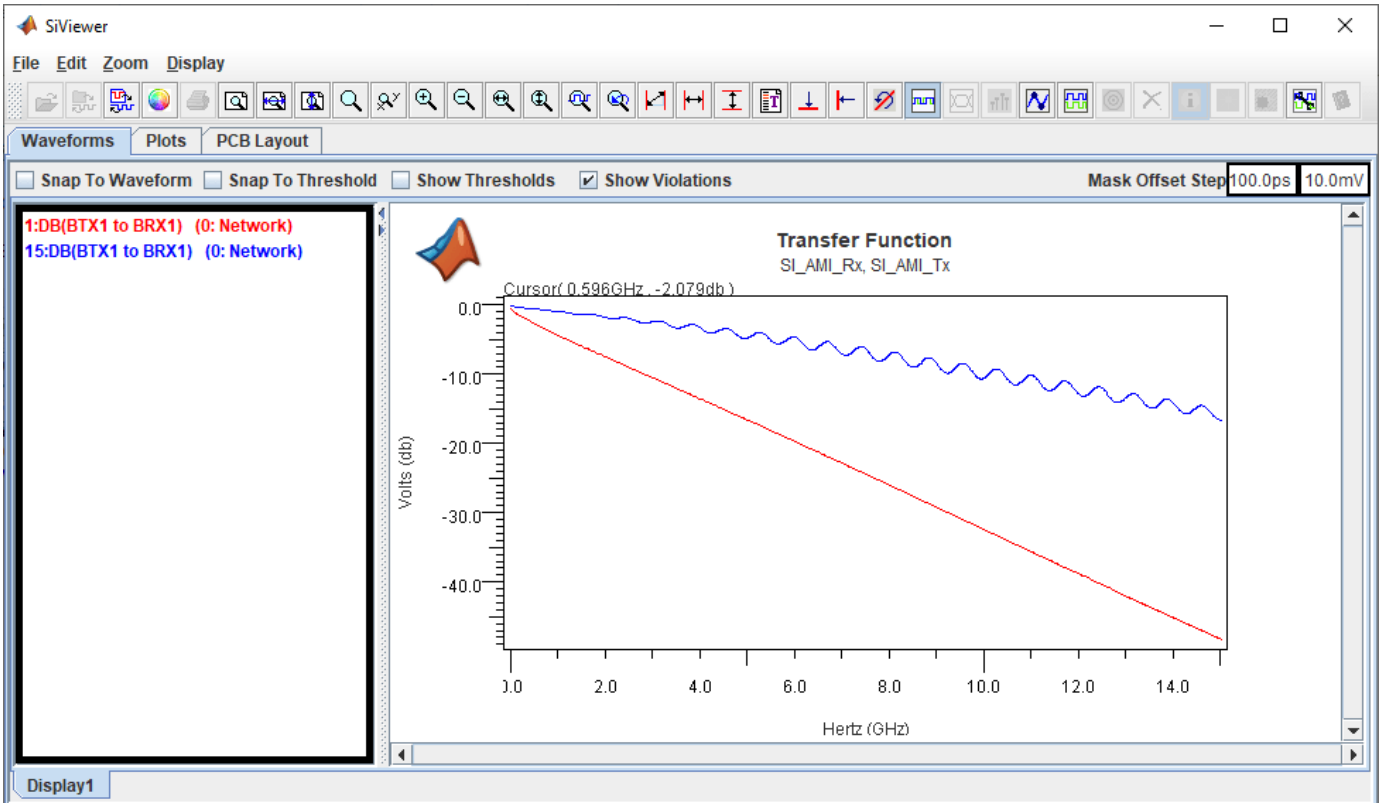
Run the simulation by selecting **Run > Simulate Selected**. In the Prelayout Channel Analysis dialog box, select **Validate**, **Generate Netlists**, **Perform Channel Analysis**, and **Autoload Results**. Make sure **Include Statistical Analysis** and **Include Time Domain Analysis** are unchecked, so network characterization is the only analysis performed. Click **Run** to start the simulation process.

When the analysis is finished the **SI Viewer** app launches and loads the analysis results. The table has one row per simulation. You can sort by any column by clicking on the column header. For this example, the lowest loss is around 4.68 dB and the highest loss is around 16.14 dB.

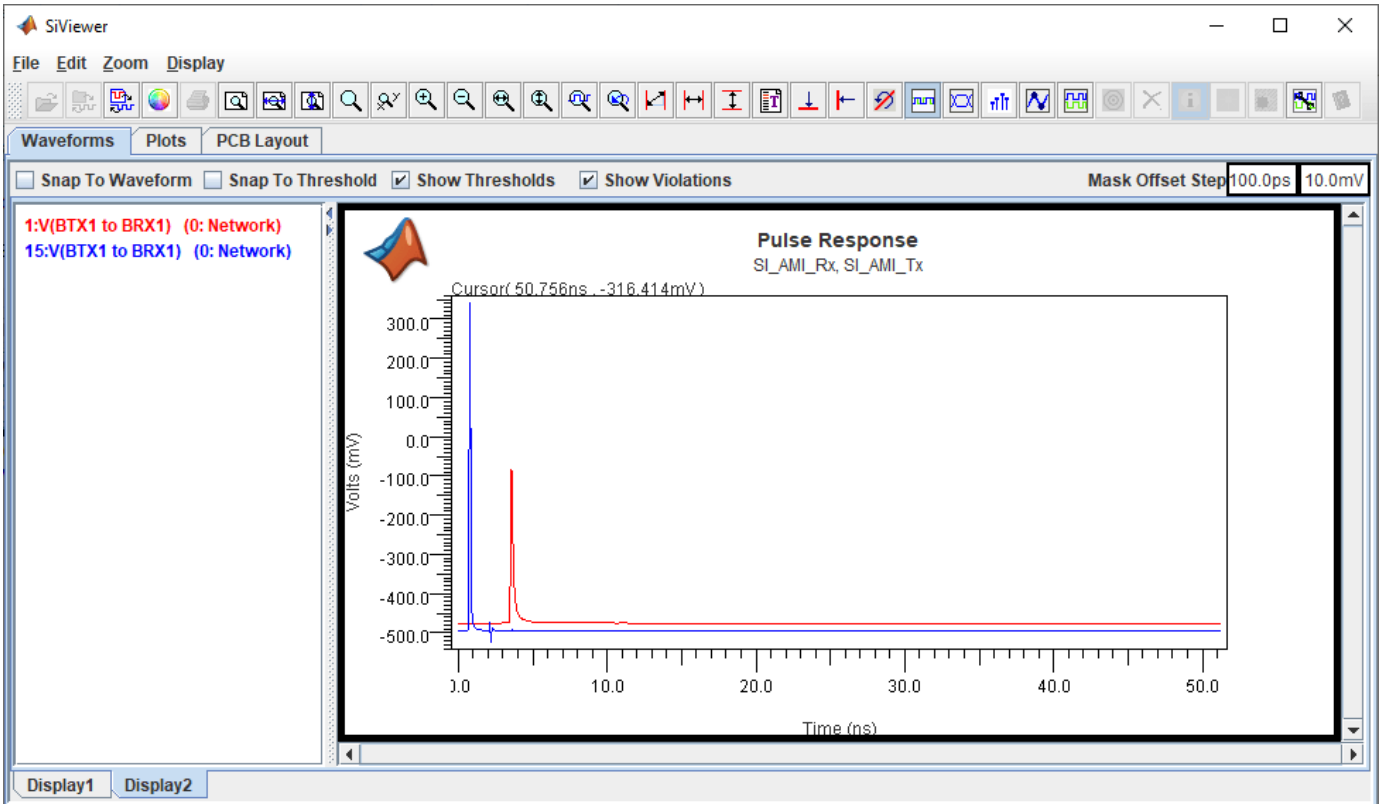
Mode: Results Waveform File Selection: Enable Multi-Select

Row	ID	Transfer Net	State	Transfer	(Gbps)	Symbol Rate (Gbaud)	Loss (dB)	UnEQ Signal/X
1	1	channel	default	TX1_to_RX1		10	16.1357	100
2	6	channel	default	TX1_to_RX1		10	16.1357	100
3	11	channel	default	TX1_to_RX1		10	16.1357	100
4	2	channel	default	TX1_to_RX1		10	13.1987	100
5	7	channel	default	TX1_to_RX1		10	13.1987	100
6	12	channel	default	TX1_to_RX1		10	13.1987	100
7	3	channel	default	TX1_to_RX1		10	10.2926	100
8	8	channel	default	TX1_to_RX1		10	10.2926	100
9	13	channel	default	TX1_to_RX1		10	10.2926	100
10	4	channel	default	TX1_to_RX1		10	7.44129	100
11	9	channel	default	TX1_to_RX1		10	7.44129	100
12	14	channel	default	TX1_to_RX1		10	7.44129	100
13	5	channel	default	TX1_to_RX1		10	4.68441	100
14	10	channel	default	TX1_to_RX1		10	4.68441	100
15	15	channel	default	TX1_to_RX1		10	4.68441	100

To view the transfer function of any data, select the data, right click and select **Show Transfer Function (Unequilized)**.



To add a new display, right-click on the display name tab and select **Add New Display**. Select the **Show Pulse Response (Unequalized)** and view it.



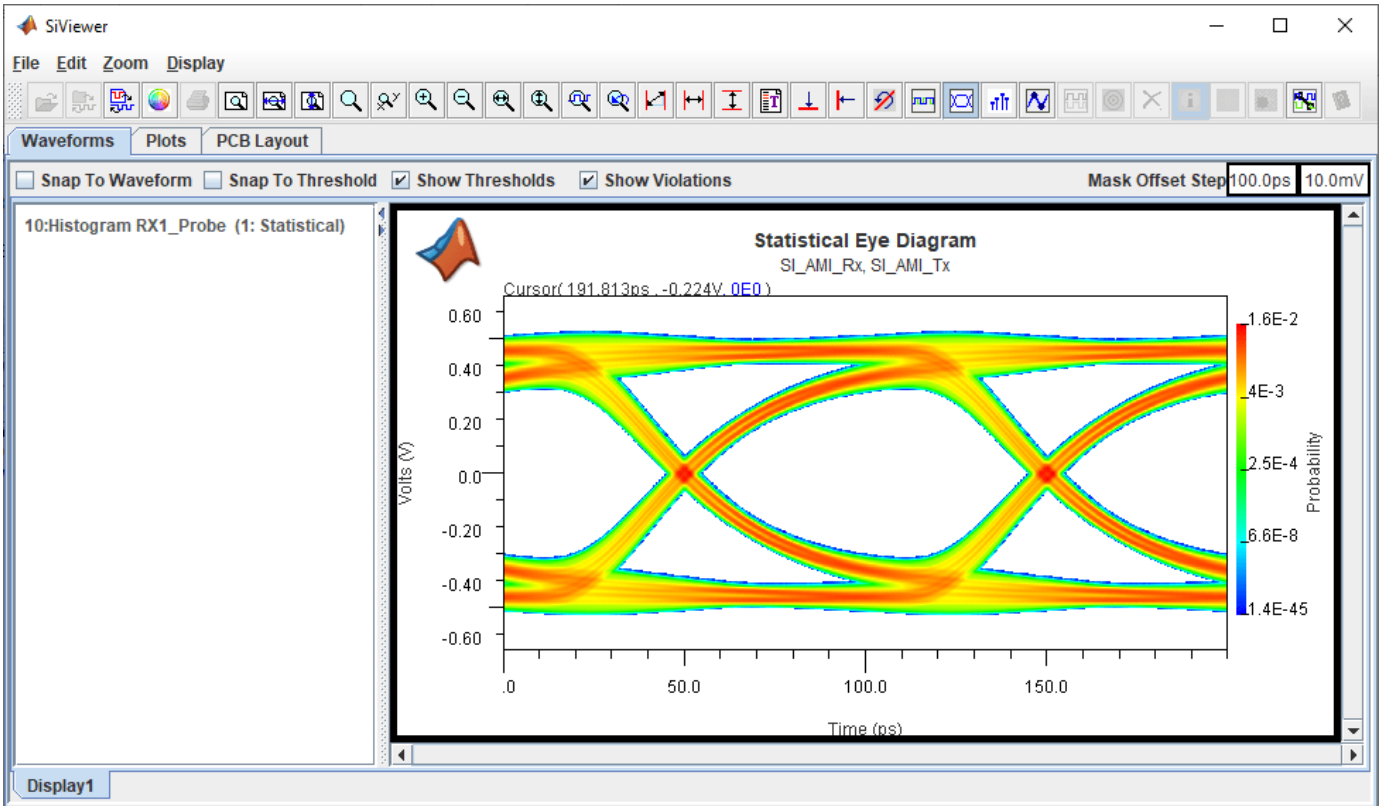
Close the **SI Viewer** app.

Statistical and Time Domain Analysis

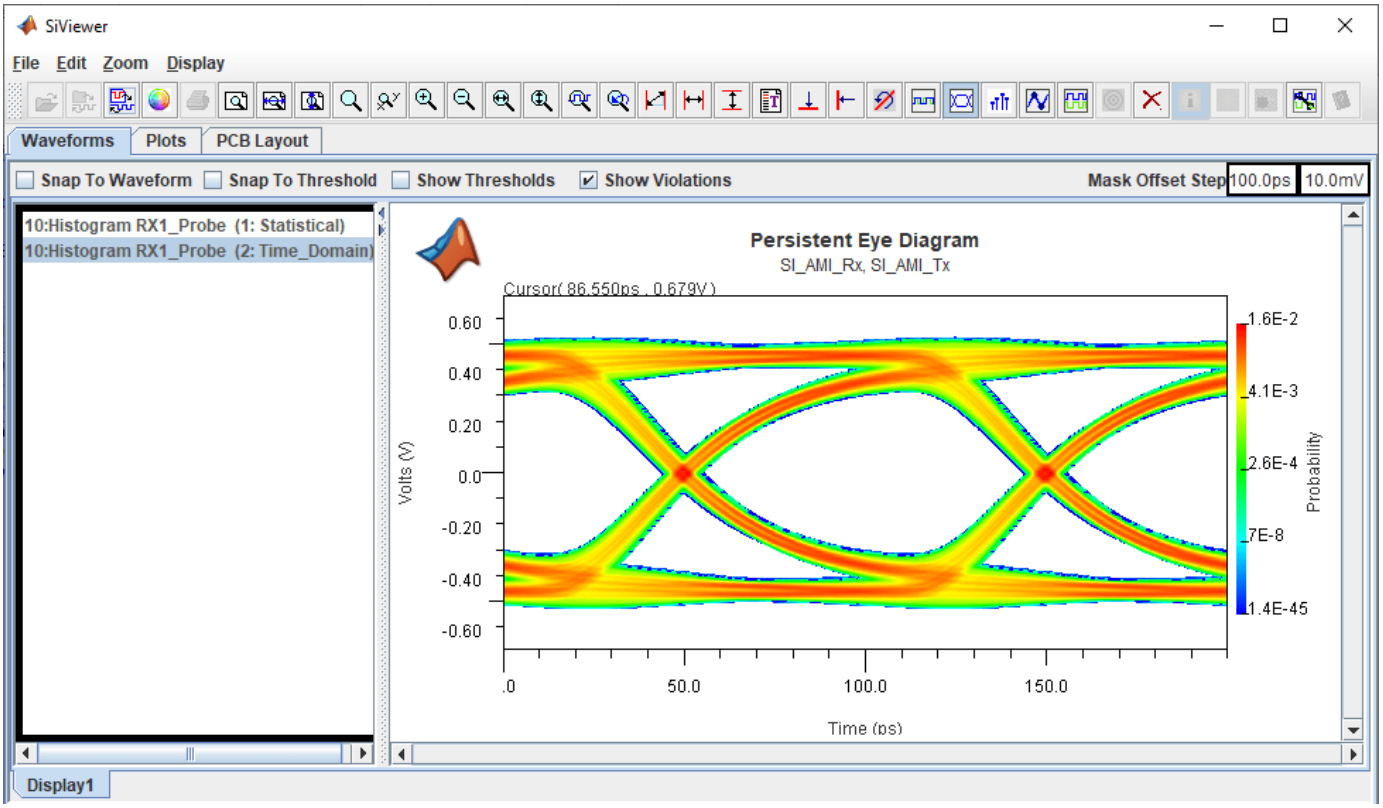
To run the statistical and time domain analysis, select **Include Statistical Analysis** and **Include Time Domain Analysis** in the Prelayout Channel Analysis dialog box. Click **Run** to start the simulation process.

When the analysis is finished the **SI Viewer** app launches and loads the analysis results. There are three tabs in the results table, one for network characterization, one for statistical analysis, and one time domain analysis.

To view the statistical analysis results of any data, select the Statistical tab, select the data, right click and select **Show Statistical Eye**.



To view the time domain analysis results of any data, select the Time_Domain tab, select the data, right click and select **Show Persistent Eye**.



Close the **SI Viewer** app and the Prelayout Channel Analysis dialog box.

See Also

Serial Link Designer | **Signal Integrity Viewer**

Related Examples

- “Analyze Backplane with Line Cards”
- “Edit Imported S-Parameter Data”